Investigating Human-Perceptual Properties of "Shapes" Using 3D Shapes and 2D Fonts

Luther Power

This dissertation is submitted for the degree of Doctor of Philosophy

April 2020

School of Computing and Communications

Declaration

This thesis has not been submitted in support of an application for another degree at this or any other university. It is the result of my own work and includes nothing that is the outcome of work done in collaboration except where specifically indicated. Many of the ideas in this thesis were the product of discussion with my supervisor Manfred Lau.

Excerpts of this thesis have been published in the following conference manuscripts and academic publications:

Luther Power and Manfred Lau. Schelling Meshes. SAP, Poster, 2017 [1].

Luther Power and Manfred Lau. Schelling Meshes. Eurographics, Short Paper, 2019.

Luther Power and Manfred Lau. Font Specificity. Eurographics, Short Paper, 2019.

Abstract

Shapes are generally used to convey meaning. They are used in video games, films and other multimedia, in diverse ways. 3D shapes may be destined for virtual scenes or represent objects to be constructed in the real-world. Fonts add character to an otherwise plain block of text, allowing the writer to make important points more visually prominent or distinct from other text. They can indicate the structure of a document, at a glance.

Rather than studying shapes through traditional geometric shape descriptors, we provide alternative methods to describe and analyse shapes, from a lens of human perception. This is done via the concepts of Schelling Points and Image Specificity.

Schelling Points are choices people make when they aim to match with what they expect others to choose but cannot communicate with others to determine an answer. We study whole mesh selections in this setting, where Schelling Meshes are the most frequently selected shapes. The key idea behind image Specificity is that different images evoke different descriptions; but 'Specific' images yield more consistent descriptions than others. We apply Specificity to 2D fonts.

We show that each concept can be learned and predict them for fonts and 3D shapes, respectively, using a depth image-based convolutional neural network. Results are shown for a range of fonts and 3D shapes and we demonstrate that font Specificity and the Schelling meshes concept are useful for visualisation, clustering, and search applications. Overall, we find that each concept represents similarities between their respective type of shape, even when there are discontinuities between the shape geometries themselves. The 'context' of these similarities is in some kind of abstract or subjective meaning which is consistent among different people.

Acknowledgements

Firstly, I would like to thank my parents for bringing me into this world, and my Dad especially for inspiring me to become involved in Computing. Luckily my stubbornness didn't impede me from heeding his advice. I also appreciate the timely assistance of my friends: Callum Shields, Tahsin Rahman and James Buckley, for dragging me away from my work and worries from time to time, to relax – although sometimes engaging in heated political arguments, in the case of Callum – helping me to stay on track with my work and keep a clear mind.

I would also like to thank Dr. Kapil Dev and Aluna Everitt for being brilliant lab colleagues. Their work and determination always encouraged me to keep motivated. I must also mention Dr. John Hardy at this point, who helped me feel welcome during the initial stages of my PhD.

In addition, I would like to thank Lancaster University as a whole, for giving me the opportunity to study for a PhD, via a scholarship. In relation to this, I would like to give thanks to Prof. Hans Gellersen and Dr. Gerald Kotonya for their support. Lastly, I want to acknowledge the great supervision I have had under Dr. Manfred Lau and Dr. Abe Karnik in guiding me in both the technical aspects and presentation of the work in this thesis.

Contents

List of Tables

List of Figures

1 Introduction

Shapes in general, are used to convey meaning. 3D shapes are used in video games, films and other multimedia, in diverse ways. Fonts are the most common 2D shapes seen every day. But, understanding how users perceive and interact with shapes is a relatively new area of work. This could allow software to be more adaptable to a user's wants or needs. Examples could include visualisation of shapes according to aspects of human perception, and prediction of shapes which most exhibit these properties. Eventually, people's ideals could be in some way expressed in the software which they use, guiding and supporting further development of these ideas.

The field of 3D modelling focuses on the creation of tools and techniques for computer-aided design of 3D shapes. These shapes may be destined for virtual scenes, such as those of a videogame, or instead represent objects to be constructed in the real-world. 3D shapes designed for physical construction have varied complexity in their geometry. They may exude the simplicity of a ceramic bowl or reach the detailed intricacy of a sports car. Computer-generated animations use 3D shapes to produce majestic environments and the characters within them, expressing a story from the characters' point of view. Sometimes, this can be done simply through a character's facial

expressions and gestures, expressing their emotion as the story progresses. 3D shapes can additionally be used for data visualisation, allowing one to look at slice of a set of high-dimensional data. For a more concrete example, we can visualise tissue samples via MRI (Magnetic Resonance Imaging) scans.

Fonts can be interpreted as additional properties or transformations of a baseline alphabet, which enable different visual expressions of text. Examples include size/scale, font weight and kerning (adjustment of space between characters). Fonts are used across many written and typed works, from letters and notes, to film scripts or posters. They can indicate the structure of a document, at a glance. They add character to an otherwise plain block of text, allowing the writer to make important points more visually prominent or distinct from other text. Through computers, more varied font geometries can be expressed in webpages and typed documents created with word processing packages. These can be changed on a whim, as the user prefers. Typography is a related field of work which aims to discover techniques to style, arrange and change the appearance of text, to make it more legible and visually appealing to a reader. This can be done via modification of font sizes, kerning, character width/length, and the lengths of edges/lines within characters.

1.1 Problem Space

In general, there is a lot of data available from many different sources of multimedia, comprised of many categories. Sources include social media sites such as: *Instagram*, *Flickr*, *Facebook*, and *Twitter* – which predominantly provide images and text. Game asset stores associated with *Unity* and *Unreal Engine*, provide animation data, 3D shapes, textures and more. *Trimble 3D Warehouse* [2] could also be included under this category, for provision of 3D shapes. Audio can be freely obtained and streamed via

Soundcloud. Many wallpapers and fonts are freely available under creative commons licensing. Every day, photographs of moments in people's lives are captured, videos are uploaded, audio tracks created, and 3D shapes modelled, adding to these vast databases of multimedia. There are many types of media and data, from many sources… But how can we organise this data? We may be able to organise it geometrically, but is this what people choose or prefer?

If we wish to organise data from a view of human perception or user preference, it becomes increasingly difficult to organise and interpret this data in a multi-modal manner, so we tried to understand at least a subset of it, in this way. We focused on 3D shapes and 2D fonts, as they are forms of geometry independent of colour, lighting and other attributes, such as colour images. Restricting ourselves in this way would help to narrow down potential variables in our studies. Figure 1.1 provides some examples of the data that we collected and used.

When people study 3D shapes and 2D fonts, they typically use traditional geometric shape descriptors. These are an approximate description of some aspect of a shape's geometry, such as its curvature (see sections 2.1.3, 2.2.2 and 2.3.2 of the *Background* chapter for more details). Curvature has been related to shape aesthetics or beauty in architecture [3]. It has been shown that when people contemplate beauty, viewing contours exclusively activates a region of the brain "strongly responsive to the reward properties and emotional salience of objects" [3].

Figure 1.1 – Examples of 3D shapes and 2D fonts used as part of the thesis' Schelling meshes and font Specificity work.

But, are shape descriptors necessarily enough to understand how people interact with and want to organise 3D shapes and 2D fonts? People have different preferences and provide different interpretations of the same objects, but nevertheless, it has been shown that there can be some level of agreement between our perceptions of geometry [4], which we believe can be exploited for creative applications, such as product design or advertising. More generally, we may be able to use this understanding for search or visualisation applications based on perceived attributes of some geometry, with respect to other geometry of similar function. For example, a single armchair vs. other chairs, or a creative font vs a group of simpler, more legible fonts.

Some typical geometric representations of 2D images and 3D shapes are described in sections 2.2.1 and 2.3.1 of the *Background* chapter, respectively.

1.2 Theme

In this thesis, we introduce new ways to understand 2D fonts and 3D shapes, by measuring and discovering human-perceptual aspects of their geometry, via the concepts of *Schelling Points* [5, 6] and *Specificity* [7].

We define a *human interpretation* of shape as a subjective response to some visual stimuli in the form of a shape, based on the visible geometric structure and/or topology of that shape.

Our approach differs from the traditional approach of saliency detection on individual shapes or fonts, as a basis for understanding them. We instead use a *data-driven* approach where we collect data based on a human interpretation of their geometry relative to other shapes and use this data to better understand them.

1.3 Methodology

Here we provide our methodology for studying and understanding 3D shapes and 2D fonts, which focuses on human perception. We use the concepts of Schelling points or Specificity as tools for this purpose.

1.3.1 Concepts

Schelling Points

An example of human interpretation that we focused on, is the notion of a *Schelling Point*. Schelling Points (or focal points) are a concept invented by Thomas Schelling [5]. They are choices that people make when they aim to match with what they expect others to choose but cannot communicate with others to determine an answer (see section 2.1.1 of the *Background* chapter for more details). Previous work has studied points on 3D meshes selected by people due to their salience in this coordination game

setting [6]. Participants aimed to select vertices that they believed others would also pick.

Existing work had not collected data on whole shape selections from a set of shapes, so we took this approach. When participants are given the task of matching other people's shape selections, we name the most selected shapes under this setting, 'Schelling meshes' (as we intend them to be in the form of polygon meshes). These could be described as the most salient meshes within that set, given the task. Examples of Schelling point distributions on meshes are shown in Figure 1.2, in addition to examples of Schelling mesh selections in Figure 1.3, the latter of which, we collected in our work.

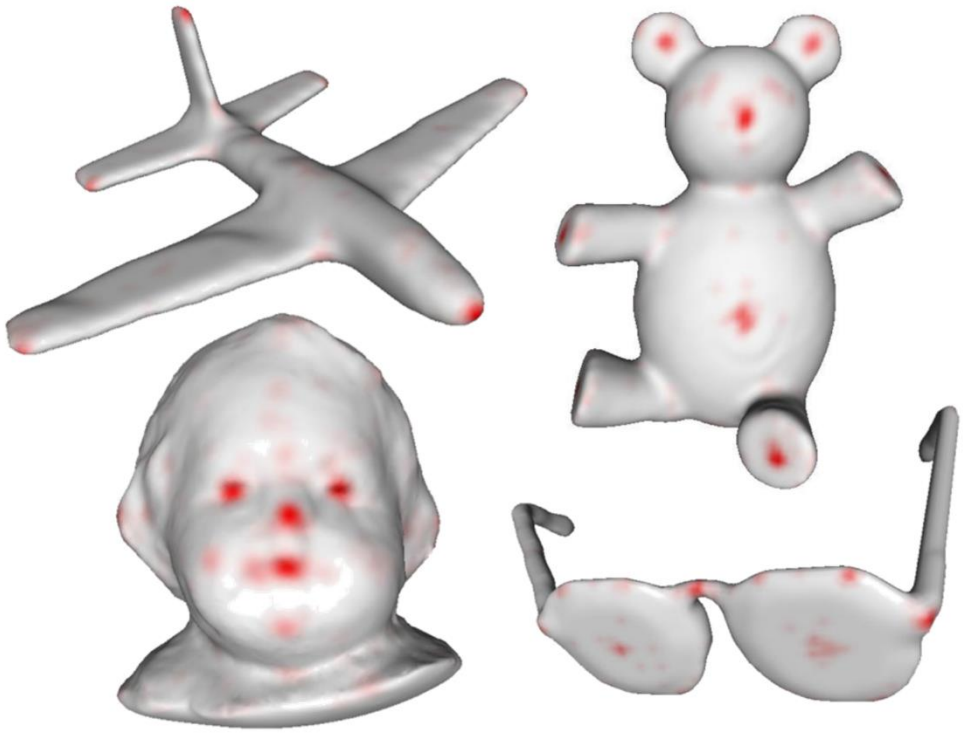


Figure 1.2 – User-chosen Schelling point distributions on polygon meshes [6].

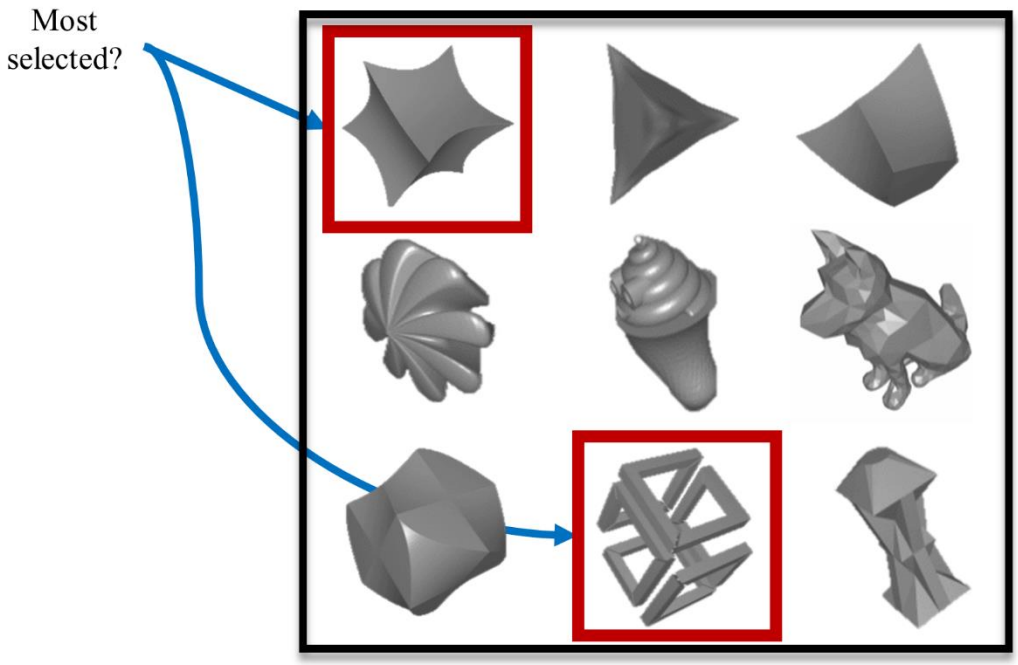


Figure 1.3 – Schelling mesh selections. An example of user-chosen Schelling meshes out of a class of shapes.

Specificity

Another example of human interpretation is the notion of *Specificity*. The origin of the term is from the *Image Specificity* work [7], where the authors asked people to describe images via sentences of text. Each image consisted of a scene of objects and was associated with multiple sentences.

The key idea behind *Specificity* is that different images evoke different descriptions, but '*Specific'* images yield more consistent descriptions than others. These images could be photographs, each depicting a real-world scene, as in the original *Image Specificity* work, or images of individual objects with unique details (see section 2.1.2 of the *Background* chapter for more details).

Existing work had only studied the Specificity of photographic images. Inspired by this, we focused on applying the concept of *Specificity* to 2D fonts. An example image description from the *Image Specificity* paper is provided in Figure 1.4, along with an example from the work in this thesis, in Figure 1.5.

Figure 1.4 – Sentence-level image descriptions from Image Specificity paper [7]. Orange indicates a subject, blue indicates the action, pink indicates nouns and and red, a place.

Figure 1.5 – Word-level descriptions of fonts and associated font images.These were collected as part of the thesis' font Specificity work.

Group-level Saliency

These two measures share some common concepts. Firstly, they require some notion of relative comparison between objects within a group. They also treat shapes as discrete objects (rather than continuous ones). Schelling Points reflect how distinctive an element of a group is, with respect to the other elements of the group. In some way, a Schelling point is a quintessential element of that group. A Specific element of a group is likely to be one which can be represented using the least amount of information, with respect to the other elements of the group. Each measure therefore depends on the distribution of the group's objects. For example, if the group is a class of 3D shapes, the class' distribution could be represented through many potential factors: how varied each shape's surface geometry is; the intended function of each shape, or the familiarity of an observer with each shape's structure. Overall, these approaches allow you to measure complementary aspects of the shapes within a group, that are different to the underlying geometry – e.g. perceived creativity, memorability.

These are *Group-level saliency* approaches. *Unit-level saliency* approaches describe or focus on an individual object at a time, with a goal of understanding which sub-components of a single object are salient. Group-level saliency involves the comparison of whole, discrete objects (a raster image; a polygon mesh) – possibly via derived information such as textual descriptions, whereas unit-level saliency uses approximations to continuous elements (pixels, polygons, voxels, superpixels etc) to describe a single object.

For a group-level saliency example, we might develop a measure of 'distinctiveness' for 3D shapes, which for a chair might suggest how extreme it is relative to traditional designs (armchairs, wooden chairs, stools etc.). By keeping track of this measure over time, we can see how well the individual chairs within a group can be distinguished

from the rest of the group, at regular time periods. It may be possible to make long-term predictions about the group. If we try to predict the ‘distinctiveness’ of new chairs, via a regression model, it can be treated as a population-level statistic for the chairs.

1.3.2 Research Questions

Schelling Meshes

Based on the *Schelling Points* concept, we tried to understand which 3D shapes are selected by people when they want to match shapes that others will pick.

The assumption was that there existed some level of agreement between the shapes that people would select, either exact (‘as-a-whole’ shape choice), or correlational (based on the properties of a subset of the shape collection shown to them), that we could exploit. Overall, we aimed to determine the degree of this agreement, across different classes of 3D shape.

Research Question: Can we understand more about the *Schelling* concept, in the context of 3D shapes, and apply this concept in a useful manner?

Importance: We believed this could be a basis for a group-level saliency of 3D shapes, allowing for relative comparison between shapes, via complementary subjective factors to that of the shape geometry – e.g. creativity, memorability. From this, group-level saliency predictions could then be possible for new meshes in a class, via a machine learning model based on collected data. A measure like this could be used to organise shapes using perceptual information, to attract attention to safety indicators or potential advertising.

Aims: The aim of our work was to understand 3D shapes in the specific "Schelling" context. We wanted to understand Schelling meshes by collecting data on the concept.

We aimed to characterise the notion of Schelling meshes and determine whether it could help us further understand 3D shapes, for applications in search, visualisation and/or clustering.

Potential Applications

If Schelling meshes are consistent among a class, there is likely at least one complementary aspect to those shapes which causes them to be selected. This perceived factor could for example, be a notion of creativity among the shapes. Other potential candidates may include: uniqueness, memorability, and so-on…

This knowledge could enable new methods of shape visualisation which adapt how shapes are viewed, based on some complementary aspect of their geometry. For example, less creative shapes could be given contrasting colours compared to more creative shapes, when compared in a uniform colour case (e.g. all shapes are grey).

Additionally, 3D products and packaging could be designed around automatically generated shapes which exhibit geometry considered to be most memorable, for more effective advertising.

Font Specificity

Using the concept of *Specificity*, we attempted to understand whether 2D fonts can be consistently described, when done in a subjective manner.

We took a similar approach to that of *Image Specificity* [7], in that we asked people to describe images of fonts via text, but we requested that words be provided, instead of sentences. We showed only one font per image, which lacked the scene-level complexity of images being described in the *Image Specificity* work. Each image consisted of a single font and was associated with multiple words. As these words each

conveyed different meanings (word senses), different words influenced the overall subjective response from participants, in different ways. Overall, '*Specific'* fonts yielded more consistent descriptions than others – from measurement based on word frequency, or word co-occurrence probabilities (word embeddings).

Research Question: Can we understand more about the concept of *Specificity*, in the context of 2D fonts, and apply this concept in a useful manner?

Importance: We believed this could be a basis for a group-level saliency of 2D fonts, allowing for relative comparison between fonts, via complementary subjective factors to that of the fonts' geometry – e.g. legibility, creativity, elegance. Given some measure of Specificity, group-level saliency predictions could then be possible for new fonts relative to the original set, via a machine learning model based on collected data. These could be used to organise fonts using perceptual information in a way that is closer to that of natural language concepts which a user of some software may intend to filter or process by. From this, we imagined that applications in search, visualisation and/or clustering could exist, and possibly other areas such as word-processing or syntax highlighting.

Aims: The aim of our work was to understand 2D fonts under the context of Specificity. We wanted to understand font Specificity by collecting data on the concept. We aimed to characterise font Specificity and determine whether it could help us further understand fonts, for applications in search and visualisation.

Potential Applications

Specific fonts are likely to be those that look the most geometrically simple, since people find fewer ways to describe them. These may be the most legible fonts, but least creative fonts. Assuming this is true, Specificity could be helpful in situations where a

user may want to select fonts which complement and contrast text, or possibly fonts which aid in memorability and clarity of text.

Search applications could for example, rank a selection of fonts according to their Specificity, with fonts high in value being used to emphasise important text within a document – and vice-versa. This could be a helpful extension to a word processing package, as it is easy to select a font and never change it – simply due to its common use, or choice as a social standard. A word processor could use this information to automatically find combinations of fonts for a title, document headings and body text, providing potential style suggestions. Additionally, in a programming development environment, we might want to automatically choose more creative fonts for syntax highlighting, relative to the main body of code.

1.3.3 Data Collection

As a basis for measuring which 3D shapes were most likely to be Schelling meshes, and fonts, most Specific, we collected data on 3D shapes and 2D fonts using the Amazon Mechanical Turk crowdsourcing platform. Shapes were shown as *gif* images undergoing looped 360° rotation, and fonts were shown as static *png* images. Cylindrical rotation was provided, as most objects people come across are placed in an upright orientation. Additionally, a single axis of rotation was used to encourage continuity in the rotation, avoiding sudden changes attracting attention. We firstly obtained a dataset of 3D polygon meshes, split into various classes (e.g. tables, chairs, lamps etc.) and additionally, a collection of greyscale fonts. We then collected shape selections made under the Schelling context, for separate classes of 3D shape, in addition to textual data for fonts. Examples are shown in Figure 1.6 and Figure 1.7. Shapes were collected from the *ShapeNet* dataset [8], and fonts from *fontlibrary.org*.

Figure 1.6 – Example shape selections (‘4-choose-1’ approach). Each row is a question, with an example answer highlighted (selected shape out of four).

No examples were shown to participants prior to completing a survey.

A qualification survey was used to filter participants.

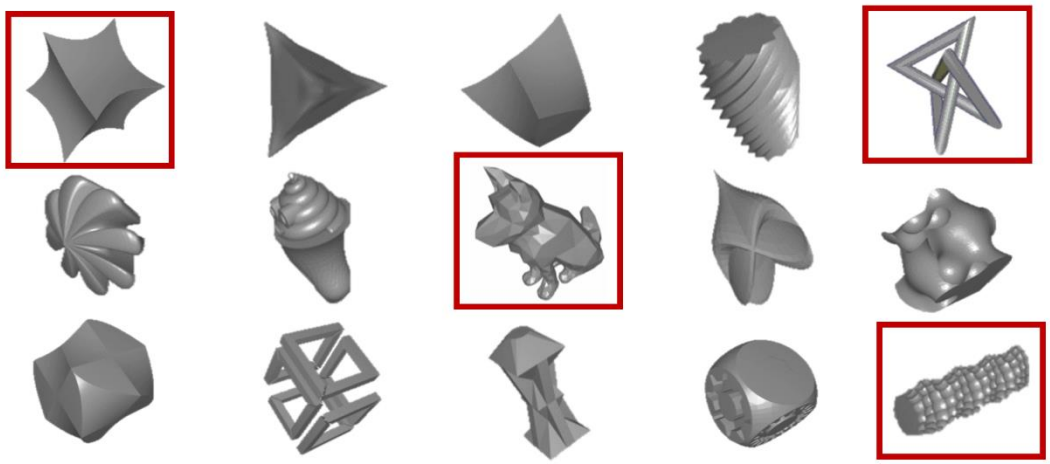


Figure 1.7 – Example shape selections (‘Many-Within-Class’ approach). An example question with example answers highlighted (4 out of 15 abstract shapes).

No examples were shown to participants prior to completing a survey.

For the ‘4-choose-1’ case, selection data consisted of one chosen shape out of four shapes. People selected each shape with the aim of matching with what they expected others to choose, given that permutation of four shapes. Group-level saliency should be

measured across an entire class, but through many permutations of four shapes being shown, relative selection frequencies can be obtained per shape, within the class. In studying these frequencies, we assumed that studying permutations of 4 shapes could lead to the discovery of class-level properties of the shapes. A simple qualification test was carried out, showing questions consisting of four shapes, where was one was unique and the other three were the same. The unique shape was required to be selected across 70% of all questions, for the participant to be allowed to complete a full survey. For the 'Many-Within-Class' case, selections consisted of one or more chosen shapes from a class of shapes, where people selected shapes with the aim of matching with what they expected others to choose. We reiterated this point, with a reminder before each question. We found that this was enough to achieve good results across shape classes, without a qualification test.

For each of the fonts, we collected human descriptions in the form of words. We focused on the consistency of the descriptions, to determine font Specificity. Due to this, we studied per-font word distributions (see sections 6.4.4 and 6.4.5). Figure 1.8 shows a selection of fonts varied in Specificity, grouped with the words used to describe them. These words were collected from people as part of the thesis' font Specificity work. Lines in blue are ambiguous due to overlap with other elements of the diagram, so connections are shown via the terminating circles at the ends of those lines.

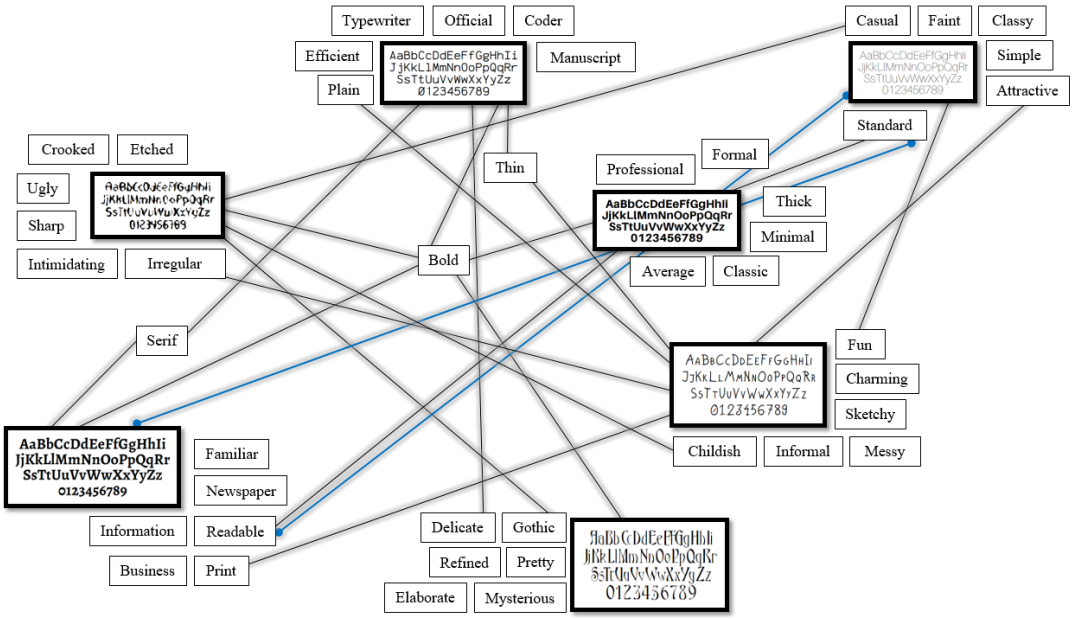


Figure 1.8 – Graph-like diagram of a subset of words used to describe fonts. Fonts are linked to words used to describe them, collected as part of the thesis' font Specificity work. Words unique to each font are placed nearby the font, without any connection. Blue lines are connected at their circular end-points to avoid ambiguity.

1.3.4 Analysis

Using the collected shape selection data, we calculated Schelling frequencies, or the frequency of a shape's selection made under the Schelling context, given how many times it was visible (unless stated, all participants could see all shapes). We could then order the shapes according to these frequencies, allowing us to visualise them and look for initial patterns. We also held separate surveys to determine whether Schelling frequencies correlated with other subjective terms (e.g. 'visual appeal', 'memorability'), and to what degree (these were determined via comments from a small initial survey). Additionally, we determined via statistical tests whether selection frequencies were consistently distributed among different groups of participants, given the same questions (see sections 4.4.4 and 5.4.1). The *Background* chapter summarises a range of statistical tests used throughout the thesis (see section 2.5.2).

Using the collected word-level descriptions of fonts, we computed word frequency-based Specificity scores for each font. Statistical testing was employed to check for consistency in word frequency distributions obtained from different groups of participants (see section 6.4.5). In subjective terms, we held surveys via Amazon Mechanical Turk, to determine whether Specific fonts were considered to be visually appealing, creative, or more legible etc, via Likert score ratings of individual fonts, for each of the terms. The chosen terms for the Likert surveys were sampled from the top-50 most frequent words provided by participants (see Figure 6.3), and topics/subjects associated with those words. We also produced an automated approach to compute font Specificity scores, by representing words assigned to fonts as vectors in a word embedding (see section 2.6.5 of the *Background* chapter for more details). Using these automated scores, we determined the properties of the most Specific fonts – for example, were they mostly bold, italic or of simple geometry? We conducted additional surveys via Amazon Mechanical Turk, to determine whether Specific fonts from a view of the automated approach, were considered to be visually appealing, creative, or more legible etc.

1.3.5 Learning

Using our collected data, we aimed to create models to predict some human-perceptual aspects of our collected 3D shape and 2D font datasets, based on the concepts of Schelling meshes and font Specificity.

To predict Schelling meshes and Specific fonts, we predominately used convolutional neural networks. Regarding Schelling meshes, we created two approaches: 1) a voxel-based convolutional neural network which predicts the relative selection probability of a shape given three other shapes in its class, and 2) a depth image-based convolutional

neural network which predicts a shape's Schelling frequency relative to its class of shape, given a triplet of orthogonal depth images each representing a shape. Data augmentation was employed by sampling different triplets from different initial positions around a shape (via rotations). In comparison to the latter approach, we also tested if geometric descriptions of 3D shapes (represented as individual vectors, or more precisely, histograms over per-vertex shape descriptor values), could be used to predict Schelling frequencies, using a fully-connected neural network. We additionally tested a voxel-based convolutional neural network for Schelling frequency prediction.

Regarding 2D fonts, we created a depth image-based convolutional neural network to predict how Specific a font is. As input, it takes an image representation of a font, mapping it to a single Specificity score, expressing the consistency of words associated with that font. We also tested if geometric descriptions of fonts (represented as individual vectors), could be used to predict font Specificity scores, using a fully-connected neural network.

1.4 Thesis Organisation

1.4.1 Layout and Structure

Following the *Introduction* is a *Background* chapter, covering key concepts needed to understand the work in this thesis, such as information on geometry representations, data collection, statistical tests / analysis, and relevant topics in machine learning.

Afterwards is the *Related Work* chapter, which provides an overview of previous research that is related to the theme of the thesis. It covers four main topics: 1) Saliency + Shape Perception, 2) Understanding of Geometry, 3) Machine Learning, and 4) Crowdsourcing. A summary is provided for each topic. In the conclusion section of the

chapter, research gaps are indicated which are most relevant to the thesis research, highlighting the contributions we provide that complement the existing literature.

In the *Schelling Meshes: '4-choose-1'* chapter, we introduce the notion of 'Schelling meshes', an approach to understanding 3D shapes via a basis of human preference. We study the agreement between participants when they select one out of four shapes, aiming to match other people's selections. We detail our data collection method, interpret and analyse the results, and describe our approach to learning and predicting which shape is most likely to be a Schelling mesh out of a group of four shapes. We also provide potential applications in search and visualisation, using shape selection frequencies given shape visibility by participants. To conclude the chapter, we discuss the approach and report our main findings.

The next chapter (*Schelling Meshes: 'Many-Within-Class'*) introduces an approach to collecting data on Schelling meshes where participants can select multiple shapes within a class, aiming to match others' selections, as before. We interpret and analyse our results and provide a method to predict how likely a shape is to be a Schelling mesh out of a shape class. This is our 'Many-Within-Class' approach. To conclude the chapter, we report and discuss our main findings.

To follow, we study 2D shapes in the *Font Specificity* chapter. This covers our approach to understanding 2D fonts via the concept of Specificity. We detail our data collection approach and show the results of our analysis. Based on these results, we show how per-font word distributions can be used to create a Specificity score and detail an approach to automatically compute Specificity scores with similar properties. We also provide a method to predict font Specificity and introduce potential applications in

search, visualisation and clustering. To conclude, we report and discuss our main findings.

We end the thesis with a *Conclusions* chapter, discussing how the topics of Schelling meshes and font Specificity relate to the thesis' theme of understanding 3D shapes and 2D fonts via human-perceptual aspects of their geometry. We provide potential future applications, and areas of research that could follow from this work.

1.5 List of Main Contributions

- **Data Collection:** Via crowdsourcing, we study what makes a shape more Schelling than others, and a font more Specific than others.
- **Analysis:** We create a scoring approach for meshes, by treating them as Schelling points, and create measures of Specificity for fonts. We determine subjective properties common to Schelling meshes or Specific fonts.
- **Learning:** We show that a function to predict which shapes are likely to be Schelling meshes can be learned for different classes of shape. Such a learned function can then be used to make predictions for any new shape, within the same class. We also show that Specificity can be learned for fonts, which similarly can be used to predict Specificity for any new font.
- **Applications:** We show that 3D shape or 2D font datasets can be clustered or directly visualised using the concept of Schelling meshes, or Specificity for fonts. This data can also be used to search for the most legible fonts, or shapes that stand out most, in a collection.

2 Background

2.1 Concepts

2.1.1 Schelling Points

Schelling Points are those choices people make when they aim to maximise their payoff/reward in a hypothetical game but cannot communicate with others to determine an answer – in other words, they may have imperfect information. When the goal of this game is to "be in agreement with others, as much as possible" (an example of a co-ordination game), they are choices selected by people when they choose to match each other's selections, with no communication beforehand [5]. For example, if two people are driving down a single path, one car travelling down from either end, but they wish to avoid a collision, they need to choose a convention to avoid that collision. If there is enough space to overtake each other's car, they might both signal to only turn left, or only turn right, keeping themselves safe. Without the necessary space, they would likely slow their car down as soon as possible! The Schelling points are either to press the brake or turn in the same direction.

We can take colour selection as another example (see Figure 2.1). When people are asked to select from four colours, three of which are blue and one, red, they are more likely to select red. Given four numbers, three of which are '1' and one, '3', they are more likely to select '3'. In the former case, there is some psychological basis or consistency behind the selection of the red colour. In the latter case, the relative change in the geometry of the fonts representing each number is the stimuli behind the selection of the number '3'. This implies that studying these selections can help to study some

aspect of human visual perception, as a top-down approach. Humans are adept at detecting patterns in a visual scene, due to the accuracy and speed of the human visual system (HVS), so aspects of these patterns should contribute to their selection decisions.

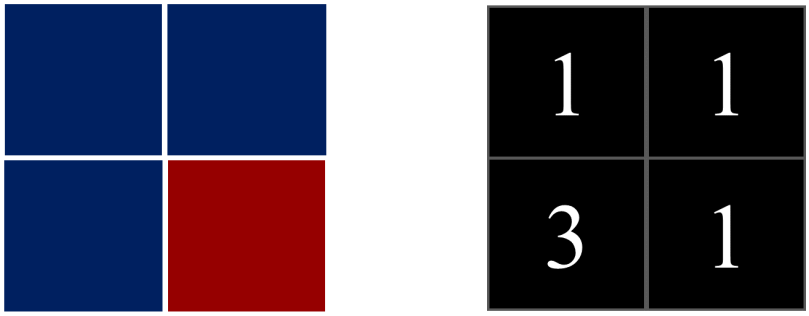


Figure 2.1 – Depiction of a colour or number-based Schelling game.

Works which involve Schelling Points aim to define some unit of selection, which in the most abstract sense is of a collection of objects. A special case of this could be a class of 3D shapes. From these, we may want to refine the selection unit down to a single shape, a region of a shape, or its base elements, such as vertices or voxels (which would be analogous to pixels in 2D images).

Previous work has studied points on 3D meshes selected by people due to their salience in this coordination game setting [6]. Participants of a crowdsourced survey aimed to select vertices on a mesh that they believed others would also pick. These could then be treated as Schelling points. Using the obtained data, a regression model was learned to predict where Schelling points would mostly likely be on a new mesh.

2.1.2 Specificity

Specificity stems from the consistency of descriptions associated with an object [7]. These descriptions might be phrases or sentences made up of unit words, or textural or colour-based symbols representing concepts of the object. If we ask people to describe an object, we expect can varied precision, from precise mathematical definitions or

statements to subjective terminology. But, since people want interpretability with little cost, they describe objects with some accuracy in-between this spectrum. Therefore, humans informally communicate via some natural language, where there is agreement on how to represent common concepts via words and punctuation, structured via a set of grammar rules.

Determining the Specificity of an object requires a set of words which are descriptive of the object, where each word is compared relative to one another within the set. Some value is obtained for each comparison, which for example, might be some occurrence frequency or ratio of occurrence frequencies, but overall it represents how common each element is, with respect to the others in the set. The average or a weighted average of these values results in a measure of Specificity for the object.

The origin of the term ‘Specificity’ comes from the *Image Specificity* work [7], where the authors asked people to describe images via sentences of text. Each image consisted of a scene of objects and was associated with multiple sentences. The key idea behind Specificity is that different images evoke different descriptions, but ‘Specific’ images yield more consistent descriptions than others. These images could be photographs, each depicting a real-world scene, as in the original *Image Specificity* work (see Figure 1.4, in the Introduction chapter, for examples), or images of individual objects with unique details.

The authors introduced two methods for measuring Specificity. One was based on human judgements, where participants rated the similarity of pairs of sentences without being shown the source image. Each sentence corresponded to the same source image. Participants therefore made ratings only on the textual content of each sentence.

The second method was an automated measure of Specificity based on the comparison of word synonyms between pairs of sentences used to describe an image. The similarities between word *synsets* (sets of word synonyms of the same meaning) between sentence pairs, contributed to the final Specificity score for an image.

The authors created a model to predict image Specificity scores, using ground-truth pairs of sentences from humans in the form of 'positive' examples, where both sentences came from the same source image. Pairs which were 'negative', did not come from the same image. The parameters of this prediction model were used to generate Specificity predictions for images not seen in their image database.

2.1.3 Shape Descriptors

Shape descriptors are designed to represent a shape's useful information, reducing the amount of space used to represent its geometry, for a specific task. This can help minimise the amount of computation required to compare or analyse shapes.

For example, we can measure the curvature of the edges/contours of objects in an image, or intensity/colour gradients across an image or texture. Specific to a 3D shape, the distribution of face normals across a polygon mesh can be obtained, or the distances between random pairs of points on a shape's surface, given by the *D2 Distribution* [9, 10].

Shape descriptors are designed to represent at least one property of a shape well, in a geometric or topological sense. This could be *scale-invariance*, where the values of the descriptor are unaffected by the source shape's size in each dimension. Other properties include *translation-invariance*, *rotation-invariance*, or sometimes, invariance to different types of symmetry: *extrinsic* (dependent on the units/co-ordinate system to

measure the shape; has invariance under rigid transformations) or *intrinsic* (inherent to the shape, regardless of co-ordinate system).

We assume that the resolution of the shape's representation that is processed (e.g. a polygon mesh), is high enough to consistently obtain properties of the shape which are close to what an ideal (usually continuous) representation would provide.

Comparison of shape descriptors usually involves a notion of distance, and so in many cases, each descriptor will be a vector of elements. These vectors can be compared using some linear algebraic distance measure, such as *Euclidean Distance*. Another candidate could be *Mahalanobis Distance*, if some notion of probability is involved in the creation of the shape descriptor, and you may want to compare the variance between its elements. A distance measure allows one to measure the (dis-)similarity of shapes, through their descriptors.

Shape descriptors are commonly used for further high-level applications, such as classification. In summary, across all shape representations, many methods of shape comparison involve the use of shape descriptors.

2.2 Understanding 2D Geometry

2.2.1 Typical Representations

2D Images

A 2D image is a contiguous structure of regions placed along two dimensions, known as *pixels*, which are each assigned values. If we index into the image using integer locations, we call the image *discrete*. But, if we use real-valued locations, we call the image *continuous*, as any index into the image is valid that lies within (and includes) the intervals used to define its boundary (width and height).

Scalar images

Pixels in scalar images are assigned a single value. For an intensity image, this indicates brightness at the pixel. A depth image is another example, where single depth value is assigned to each pixel.

Colour Images (RGB)

Instead of a single intensity/brightness value per pixel, we now associate three primary colour values (e.g. red, green and blue) with each pixel, which can be combined or interpolated via barycentric co-ordinates, to produce many different colours.

2.2.2 2D Shape Descriptors

Many shape descriptors exist for 2D shapes. We discuss relevant ones here.

Histogram of Oriented Gradients (HoG)

These are histograms which represent frequencies of gradient orientation across local regions of a 2D image. Images are split into smaller 'cells', by which each pixel votes/contributes its gradient direction (change in angle from a fixed initial direction vector), weighted by its magnitude [11]. Pixels in 2D are analogous to voxels in 3D, so the approach can similarly be applied to 3D shapes, after *voxelisation*. HoG is designed to describe variation throughout an image for example, in terms of intensity/colour change of a 2D image, or surface variation in a 3D voxel grid.

Since gradients have influence in each dimension/axis, we obtain a histogram for each axis. For an image, the resulting descriptor has two histogram dimensions and can be represented via 2D vectors, based on an ideal 2D gradient vector at each image pixel. Similarly, 3D shapes have 3D gradients, leading to a 3D histogram. In both cases, these can be treated as a single vector, via consistently ordered concatenation. See Figure 2.2 for a visualisation based on an image.

Figure 2.2 – Visualisation of Histogram of Oriented Gradients as applied to an image of Lena. (Left) Test image (Right) Dominant gradient orientations of cells throughout the image [12, 11].

Contour Curvatures

Taking the contour curves around objects in an image (indicating change in intensity/colour, analogous to level sets of a geographical map), the curvature of each contour can be computed via derivatives at each point on each contour. This can possibly be done via a forward-backward difference calculation of the derivative, as a discrete approach.

SIFT

SIFT is an algorithm to detect key points within an image at different scales, via a Difference of Gaussian approach [13, 14]. See Figure 2.3 for a visual representation.

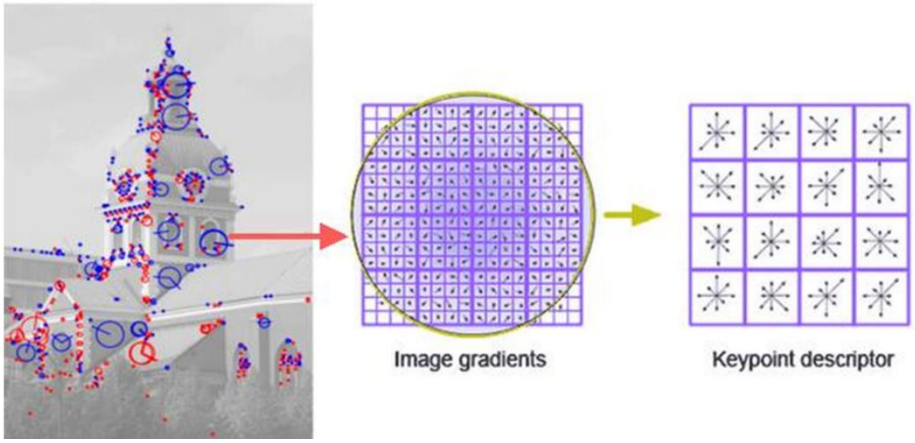


Figure 2.3 – SIFT descriptor representation for a single key point [13, 14].

At each scale/octave, images are blurred at two different values of σ. Across each of these blurred images, local extrema (pixel locations 0 of gradient) are searched for. SIFT scales down the image further and uses larger Gaussian kernels, at each octave. These are potential key points, which are filtered according to an intensity threshold, to remove edges. For each key point, a SIFT descriptor is produced based on the image gradients of a 16×16 window centred on the key point, split into 4x4 cells of 8 orientation bins each. This gives a $4 \times 4 \times 8 = 128$ dimensional descriptor. Since a variable amount of key points can be detected per image, a method to constrain the output descriptor vector to a fixed size is required. This can be done via vector quantization of SIFT keypoint descriptors, by clustering them via k-means. Taking the closest cluster centre to each descriptor as a substitute (also 128 dimensions each) and binning each position into a histogram of n bins, this provides a fixed sized descriptor for classification or regression. The number of bins tends to be high (e.g. 1024), to ensure data is sparsely located throughout each of the resulting descriptors.

SURF

This is a faster method of detecting and describing image key points, than SIFT [15, 14]. Instead of a Difference of Gaussian approach, SURF uses the determinant of the Hessian matrix to detect key points, where convolution of the image via second-derivative Gaussian kernels, is approximated using box filters (which can be calculated using integral images). Larger filters are used with each octave. Key points are searched for using extrema throughout the image at multiple scales.

A descriptor is produced using Haar wavelet responses which can also be calculated using integral images (an algorithm for determining sums of values in a rectangular subset of a grid). For each keypoint, a dominant orientation is determined and the area

surrounding the keypoint is rotated to match its direction. This is set to $20s \times 20s$, where s is the current scale. The area is split into 4x4 sub-regions, for which Haar wavelet responses are taken in the horizontal d_x and vertical directions d_y, weighted by a Gaussian kernel centred at the keypoint, to introduce some robustness to deformations and translations. For each sub-region, a vector: $v = \left(\sum d_x, \sum d_y, \sum|d_x|, \sum|d_y|\right)$ giving a descriptor of $4 \times 4 \times 4 = 64$ dimensions.

An extended 128-dimension version exists, where the sums of d_x and $|d_x|$ are computed separately for $d_y < 0$ and $d_y \geq 0$. This is similarly done for d_y and $|d_y|$, according to whether d_x is negative or non-negative. This doubles the total number of features, per descriptor.

FREAK

FREAK is a shape descriptor based on existing key points (e.g. those extracted via SIFT or BRISK), using overlapping windows around each keypoint [16]. These are structured as multiple concentric circles and are used to detect objects, in a loosely analogous manner to how the human retina works. See Figure 2.4 for a visual representation.

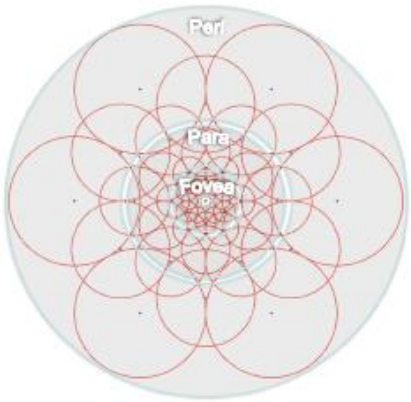


Figure 2.4 – Representation of the retinal-like sampling pattern of the FREAK descriptor [16].

Sobel Filter

The Sobel filter calculates approximations of derivatives/gradient vectors for an image. A filter is computed for each axis, vertical or horizontal [17]. This can be generalised to 3D shapes by treating them as the cells of a 3D cartesian grid (voxel grid). Since these shape representations are discrete, we cannot obtain a derivative at each point, but we can generate a useful approximation based on 3x3 sized regions of an image or 3x3x3 sized volumes of a 3D grid. Convolving an image or 3D grid with a Sobel filter (a weighted average) performs two operations, which can be separately computed: 1) a smoothing/averaging operation, and 2) a central-difference operation as an approximate derivative. See Figure 2.5 for an application of a 2D Sobel filter to an image.

Figure 2.5 – Sobel operator applied to a picture of a landscape (original image overlaid).

2.3 Understanding 3D Geometry

2.3.1 Typical Representations

Multi-view Depth Images

For each pixel in an image (e.g. an image of 3D graphics render), we obtain a different value representing how far away each polygon that is visible on screen, is from the near-projection plane of a 3D scene (objects are culled/not drawn behind this plane, and similarly, objects are culled if they are in front of the far plane).

We do this by shooting out a ray perpendicular to the plane, at the location of each pixel location shown on screen (inverted back from the screen's x, y pixel locations). If the ray hits a polygon, the 'depth' or distance is some value between the near-plane, and far-plane of the scene. We can then squash the depth to lie between 0 and 1 (including) and quantise these depth values per pixel as 8bit integers – values between 0 and 255 (including). Each pixel in the image is represented by an 8bit value. Taking multiple depth images from different positions and orientations between the near and far planes, gives you different views of a shape.

Figure 2.6 – Diagrammatic representation of depth images of a chair, taken by a virtual camera. [18]

Usually a depth image is taken at fixed rotation intervals around the object of interest, around a given axis – e.g. every 30° horizontally via the y-axis. See Figure 2.6 for a visual representation.

Polygon Meshes

A polygon mesh is a collection of vertices, faces and edges that represent a polyhedral 3D shape. Faces commonly consist of triangles and quadrilaterals. The mesh itself representing an embedding of a graph in $\mathbb{R}^3$, and so with small enough faces (e.g. repeated face subdivision), it can approximate a manifold in $\mathbb{R}^3$. See Figure 2.7 for an example render. A polygon mesh can be represented in many ways, including:

- Vertex list and face vertex indices, which each point to a vertex.
- A 'winged-edge' structure: Each edge points to two vertices, faces and the four edges that touch each vertex and face.

Figure 2.7 – Render of a polygon mesh teapot. [19]

Point Clouds

Point clouds are sets of points in $\mathbb{R}^3$, without topological information. For the purposes of shape representation, they are usually taken as the vertices of a polygon mesh, and so represent the surface of a 3D shape. In other cases, they can represent scenes of dense

geometry, potentially obtained via 3D scanners (e.g. time-of-flight, laser triangulation). They are often aligned with other point clouds or existing 3D shapes, in a process called *point set registration* – for example via the *Iterative Closest Point* algorithm, which finds a translation and rotation matrix to move the source point cloud/3D shape to a target 3D shape, or vice-versa. Figure 2.8 provides an example point cloud scan.

For the purposes of processing, point clouds are converted to other shape representations like polygon meshes or NURBS (Non-Uniform Rational Basis-Spline) surfaces.

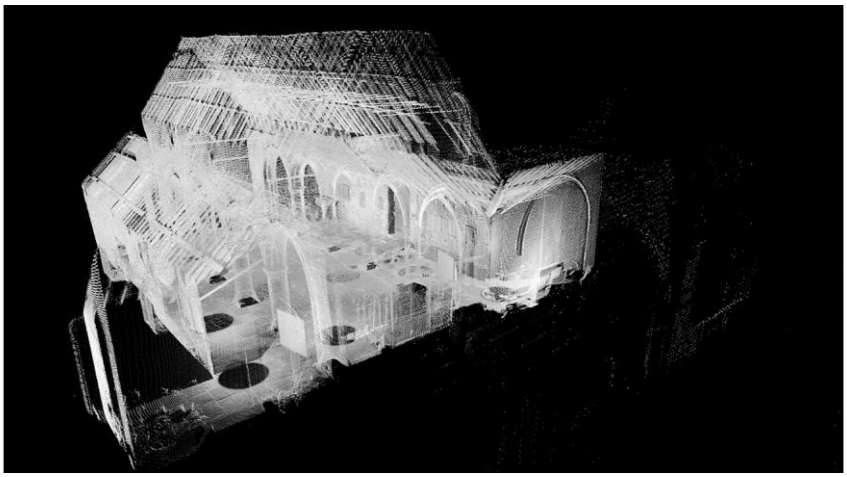


Figure 2.8 – Point cloud scan of a church. [20]

Voxel

A voxel is an element of a 3D regular grid. Each grid element is often represented via a unit cube. These cubes (or parallelotopes, in general) are tessellated together to form the grid. In the case where the grid elements are cubes, the grid is Cartesian. At other times, you may want to use a rectilinear grid, where each element is not necessarily congruent (i.e. not all the same shape). A binary voxel grid is most commonly used, where a value of 1 is stored with a voxel, to denote it occupies a point (e.g. a vertex of a polygon mesh), and 0 otherwise.

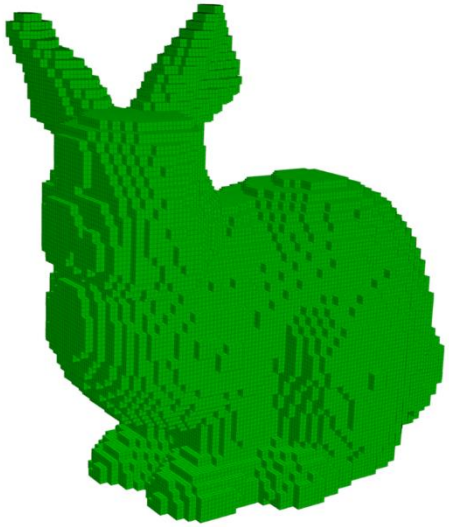


Figure 2.9 – Voxelised form of the Stanford Bunny mesh. [21]

Fixed size voxel grids are often used in discriminative and generative models of 3D shapes, to avoid one having to remodel a shape dataset each time the voxel grid resolution changes. This can be time consuming if training a neural network. Instead, you can anisotropically scale the voxel grid along each of its dimensions (scale each axis differently based on the bounding box of the original shape), to fit it within the target voxel grid dimensions for the network. But, re-sizing a voxel grid to smaller dimensions can result in a loss of information, due to the loss of resolution used to define it. See Figure 2.9 for an example voxelisation of a mesh.

Marching Cubes Algorithm

Marching cubes is an algorithm to extract a polygon mesh representation of an isosurface that lies within a scalar field [22]. A scalar field associates a value to every point in space. A binary voxel grid is a discrete 3D scalar field, which might represent some shape. We can use the algorithm to approximate a shape's surface (isosurface) from a voxel grid, by iteratively constructing a triangular polygon mesh. This is done

by partitioning space into a cube grid, then for each cube's vertices (8 in total), evaluating whether each vertex is above or below a threshold.

The vertices of each cube correspond to 8 neighbour voxels (sample values from the scalar field). If all neighbour voxels are occupied (assigned a value of 1), they are all contained within the surface. This means that no polygons will intersect this cube. Similarly, if all voxels are assigned a value of 0, they are all placed outside of the surface (assigned a value of 0), again indicating that no polygons will intersect the cube. The possible cases where the surface intersects a cube, exist when a cube has some vertices with a value of 1, and some vertices with a value of 0 (some voxels are occupied, and some are unoccupied). The isosurface in some way can be defined via the statement: "all positions of the scalar field with a value greater than some threshold". If we set this threshold to be between the possible cases of 0 or 1, it clearly distinguishes whether a vertex is inside the surface (voxel is occupied as it has a value of 1), or otherwise. For each cube, there are $2^8 = 256$ possible combinations of these vertex assignments (indicating whether vertices are above or below the threshold), which determine how to place polygons within each cube (or not). With certain symmetries, this can be reduced to 15 cases [22].

Hierarchical Structures

More recent neural network structures have been designed to allow prediction using dynamic/hierarchical voxel-space structures as input, either tree or graph-like in structure. These overcome the resolution limitations of having a fixed-size voxel grid, depending on the detail of the generating 3D shape, or method used to produce the structure (e.g. polygon mesh, or point cloud).

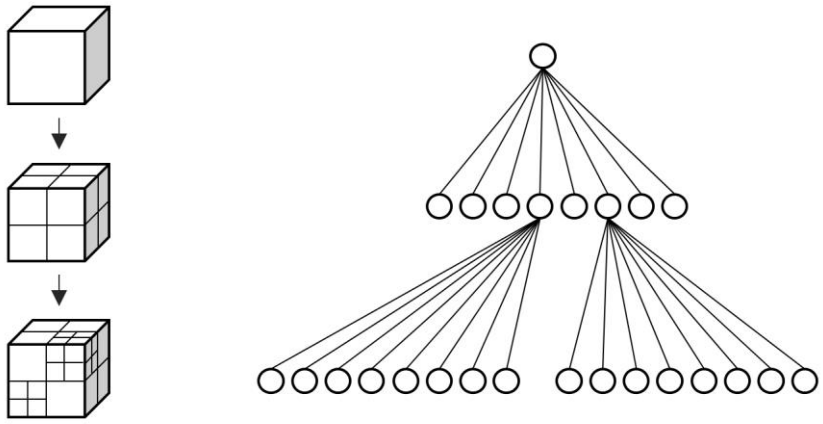


Figure 2.10 - Diagrammatic representation of a voxel octree.

Voxel Octrees are data structures that reduce the memory required to represent 3D shapes, by not storing empty space. A region of 3D space is divided into 8 blocks. If a point exists in one of these regions, then that voxel information is created, and the region is further subdivided into 8 blocks. This process can go on, recursively, if there is available memory. If a point does not exist in a region, further memory is not used up, and processing doesn't take place. Therefore, 3D surfaces can be represented more efficiently and with greater effective resolution via voxel octrees, as most voxels are empty in the fixed-grid case. Volumetric data can be represented efficiently via Sparse Voxel Octrees (SVOs) [23]. See Figure 2.10 for a diagrammatic representation of a voxel octree.

Voxel DAGs (Directed Acyclic Graphs) are an extension of voxel octrees, which also allow more efficient encoding of *identical* regions of space, since nodes in the graph can share pointers to identical subtrees. In one example of this work, a bottom-up (voxel-wise) algorithm was also produced that reduces an SVO to a minimal DAG [24]. Shape 'material' data can also be attached to these DAG structures, via an external data structure [25, 26]. Using reflective symmetries (mirror transformations along the voxel grid dimensions), the required memory to store a Voxel DAG is reduced further [27].

2.3.2 3D Shape Descriptors

Just as 2D shapes have descriptors, 3D shapes also have them. We discuss relevant ones here.

D2 Distribution

The D2 distribution is a distribution over distances between randomly sampled pairs of points in a 3D shape (can be calculated from points sampled on a polygon mesh, manifold, or point cloud) [9].

Curvature

Principal curvatures are the minimum and maximum curvatures of the curve obtained by intersecting the plane containing a surface's normal vector at a point on the surface, and the surface itself. They each measure how curved local regions of a 3D shape are. The directions of the normal vector at this minimum and maximum are known as principal directions. See Figure 2.11 for a representation of these directions on a saddle surface.

Gaussian curvature is the product of the principal curvatures at a point on a 3D surface. If the surface is saddle-shaped, the principal curvatures are both maxima, and the Gaussian curvature is negative. *Mean curvature* is computed as the average of the principal curvatures.

In a discrete or polygon mesh setting, we approximate local or differential properties at some vertex, v_i, as an average over nearby connected vertices, v_j. To do this, we can use connectivity information between the vertices of a mesh (differences between positions), but this alone isn't accurate enough for meshes with irregular triangulations (a triangulated mesh is one where all polygon faces are triangles). See Figure 2.12 for a visual representation.

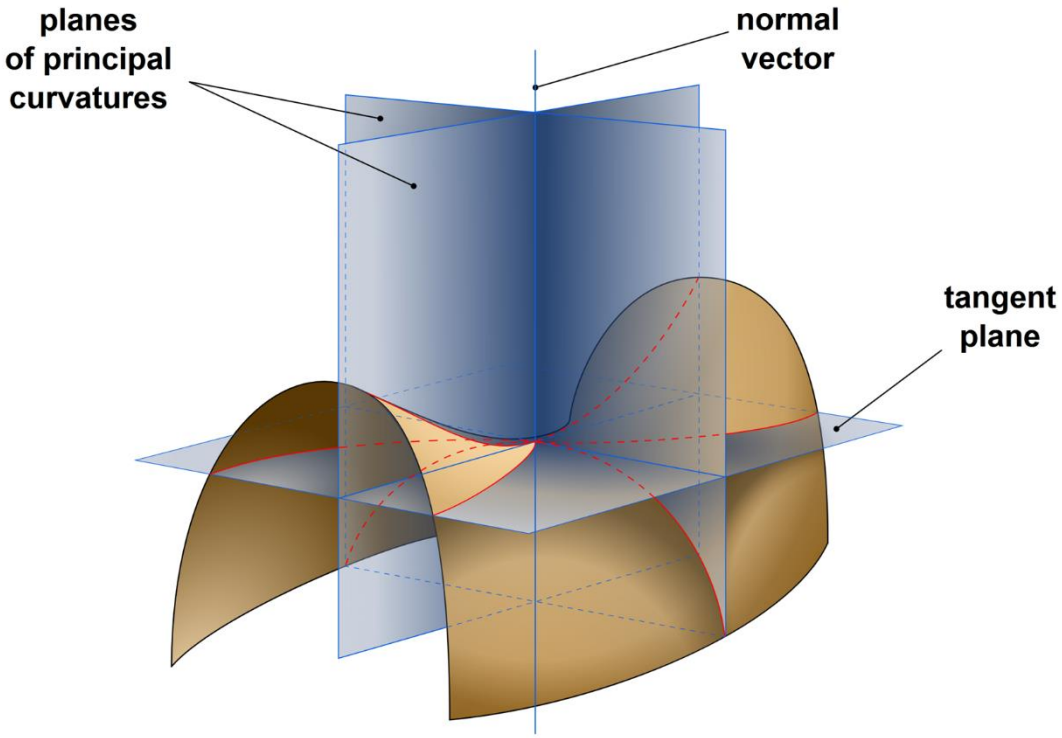


Figure 2.11 – Diagram showing principal curvature directions at a point of a hyperbolic paraboloid (saddle surface). These are obtained by the intersection of normal planes that contain the point's normal vector and the tangent plane perpendicular to them [28].

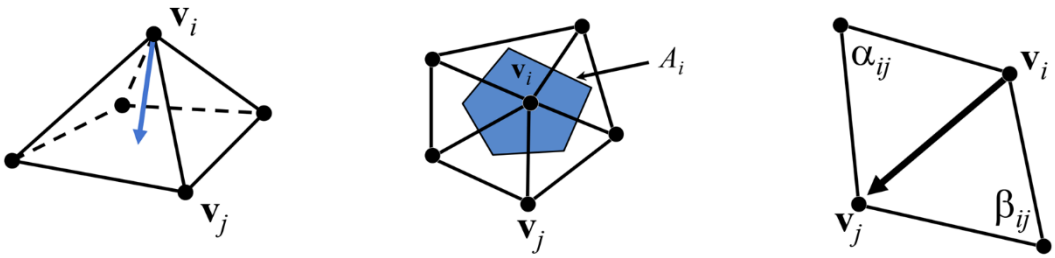


Figure 2.12 – Two views of a triangular polygon mesh and co-tangent angles used to compute discrete curvatures [29].

We augment this connectivity information with the angles between v_i and three other adjacent vertices v_j, given as α_{ij} and β_{ij}, to get a more accurate measurement (see Figure 2.12). This is reflected in the co-tangent formulation of the discrete Laplace-Beltrami operator, $L_c(\boldsymbol{v}_i)$ [30], where $N(i)$ is the set of vertices adjacent to v_i, or its neighbourhood:

$$L_c(\boldsymbol{v}_i) = \frac{1}{A_i} \sum_{j \in N(i)} \frac{1}{2} \left(\cot \alpha_{ij} + \cot \beta_{ij}\right) \left(\boldsymbol{v}_j - \boldsymbol{v}_i\right)$$

From this, we can compute per-vertex mean curvature (see Equation 2.2) and Gaussian curvature values (see Equation 2.3), in addition to the mesh's principal curvatures (see Equation 2.4):

$$H(\boldsymbol{v}_i) = \frac{\| L_c(\boldsymbol{v}_i) \|}{2}$$

Equation 2.2 – Discrete mean curvature (sign defined by vertex normal)

$$K(\boldsymbol{v}_i) = \frac{1}{A_i} \left(2\pi - \sum_j \theta_j \right)$$

Equation 2.3 – Discrete Gaussian curvature

$$\kappa_1 = H + \sqrt{H^2 - K}, \kappa_2 = H - \sqrt{H^2 - K}$$

Equation 2.4 – Principal curvatures

Repeated subdivision of a polygon mesh's faces creates a more accurate approximation of a smooth surface and more accurate curvature calculations.

Shape Diameter Function (SDF)

The shape diameter function is a measure which links a 3D shape's volume to its surface/boundary when the shape is represented as a polygon mesh [31].

The function measures the intersections of rays shot from a vertex of the polygon mesh, through a cone centred at the opposite/inward-facing direction of the vertex's normal (directs rays inside of the shape), towards the opposite side of the mesh. Intersections where the normal at that point is too close to the original vertex's direction are ignored

(< 90 degrees difference in angle). The SDF is the weighted average of ray lengths which are within one standard deviation away from the median ray length. Rays with lengths greater than this, are treated as outliers. See Figure 2.13 for a visual representation of these rays. Computing SDF values for each vertex in a mesh gives a measure of the diameter of the 3D shape's volume in the neighbourhood of each point, on its surface. The SDF is invariant to rigid body transformation of the original polygon mesh; it is "largely pose-oblivious".

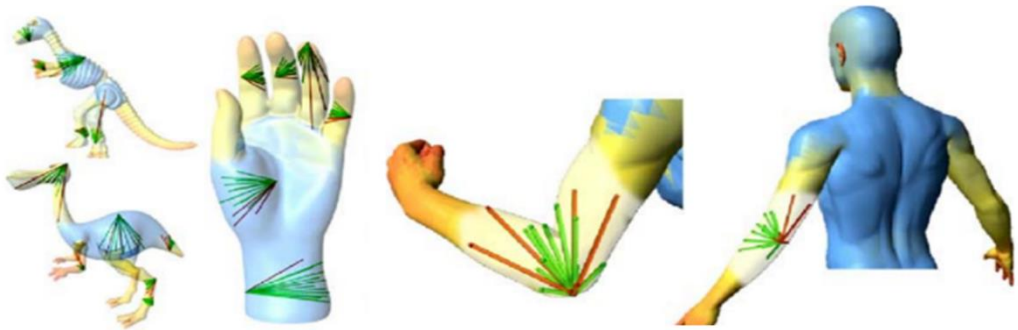


Figure 2.13 - Distribution of cones of rays shot from a mesh vertex. Valid rays used to compute the Shape Diameter Function (SDF) are shown in green. These are rays of length within one standard deviation away from the median ray length [31].

Per-vertex Normal Binning

We can obtain a histogram representing the per-vertex differences in angles *of normal* directions, throughout a polygon mesh, according to deviation from a fixed direction. This acts as a measure of how curved a surface is.

2.4 Data Collection

2.4.1 5-Point Scale (Likert)

A scale that is commonly used in research questionnaires, for participants to rate their level of agreement of a given question that applies to some stimuli. For example, if we were to ask people to rate book covers, we could ask:

"In your opinion, how interesting is the following book cover?"

Least Interesting	-2	-1	0	+1	+2	Most Interesting

Figure 2.14 – Example numerical Likert item

Participants would assign a numerical value to this question (or *Likert item*), as their response (see Figure 2.14). This numerical value would lie within a fixed range for all questions. A *Likert scale* is the sum of responses across many *Likert items* (can also be averaged after data collection). Likert items should also employ symmetry to have equal numbers of potential positive and negative responses, centred at a zero point. 5 or 7 options per scale is reasonable. A numerical value can be implied by some text of similar meaning (see Figure 2.15).

Least Interesting (-2)	Slightly Interesting (-1)	Neutral (0)	Very Interesting (1)	Most Interesting (2)

Figure 2.15 – Example text-based Likert item

2.5 Analysis

2.5.1 Pearson Correlation Coefficient

The Pearson Correlation coefficient, ρ, measures the linear correlation between two random variables, X and Y. It takes a value between -1 and 1 indicating the degree and direction of the correlation. -1 indicates a completely negative correlation, +1 a completely positive correlation, and 0 means no linear correlation. The formulation of ρ depends on the co-variance between X and Y, $cov(X,Y)$, and their standard deviations, σ_X and σ_Y (see Equation 2.5). The co-variance of X and Y is the expectation of the product of the differences between X and its mean, μ_X, and Y and its mean, μ_Y. Expectation in this case represents the average of a large number of measurements of a random variable, as the number of measurements tends to infinity (so here this implies that the earlier mentioned product is also a random variable). In the discrete case, expectation is the weighted average over all states (possible observations) of a random variable, by their probability of occurrence.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y},$$

$$cov(X,Y) = E[(X - \mu_X)(Y - \mu_Y)]$$

Equation 2.5 – Pearson correlation coefficient general formula

If we imagine that we have recorded sensor measurements, corresponding to temperature and pressure over some time period, these two lists of observations can be represented via a pair of vectors x and y. We can compute a correlation coefficient showing how linearly correlated they are to one another. As these observations would be samples (finite data), we compute the sample correlation coefficient, r, as shown in Equation 2.6:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Equation 2.6 – Pearson correlation coefficient sample-specific formula

Within the equation, n is the number of observations, and $\bar{x}$ and $\bar{y}$ correspond to the sample mean for x and y, respectively ($\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$). Throughout this thesis, we refer to Equation 2.6, when discussing correlations.

2.5.2 Statistical Hypothesis Testing

Statistical hypothesis testing can be used to determine whether some sample of data either conforms to or differs from some distribution. A decision is made according to some confidence threshold, representing how likely it is that the data observed occurred due to chance. We start with a null hypothesis (H_0) and the opposite statement (H_1). For example:

H_0: the mean of a sample is equal to another known mean.

H_1: the mean of a sample is different from another known mean.

Here, 'different' indicates less than or greater than as possibilities, indicating a two-tailed test will be performed. It is standard to perform a two-tailed test when first checking for differences as it is unlikely that any prior information is known about which direction to test for.

On selecting a suitable test for our data sample, we choose a *significance level*, α, to denote the probability of incorrectly rejecting H_0 (a type 1 error). Some common examples are: 0.05 (1/20 chance) or 0.01 (1/100 chance). So, you can see that the lower the significance level, the more confidence you have in your result. Each test involves

computing some test statistic in order to produce a *p-value,* which represents the chance of obtaining a sample that is at least as extreme as has been observed, given H_0. This p-value is compared to the chosen significance level, to determine whether we agree with or disprove H_0. If there is a difference more extreme than our chosen alpha (p-value < α), we can reject H_0 (accept H_1) and say that the data follows a different distribution to that of H_0. Otherwise, we accept H_0 (reject H_1).

Fisher's Exact Test

Fisher's exact test is useful for determining whether two groups of categorical data differ by more than chance and is valid even for small sample sizes (e.g. at least one of occurrence of less than 5 examples across measured attributes). It is an exact test, as the p-value can be computed exactly [32, 33]. The hypotheses are as follows:

H_0: There are no associations between two categorical variables that do not occur according to chance.

H_1: At least one association exists between two categorical variables which did not occur according to chance.

	Current year	Previous year	Row total
Geography	4	11	15
Maths	10	3	13
Column Total	14	14	28

Table 2.1 – Example 2x2 contingency table

For example, for a new class of 14 students, we might obtain sign-up counts for two subjects – geography and maths, vs. that of the previous year. We can tally up the results to form a *contingency table*, like that of Table 2.1.

Assuming that students are free to study either of the subjects for their starting year, and choose independently, there should be no deviation from the null hypothesis that across each year, students are equally likely to choose geography or maths as any other year. Algebraically, this can be expressed via the following contingency table:

	Current year	Previous year	Row total
Geography	a	b	$a + b$
Maths	c	d	$c + d$
Column Total	$a + c$	$b + d$	$a + b + c + d$

Table 2.2 – Generalised form of a 2x2 contingency table

Fisher showed that the probability, p, of obtaining a set of values such as this, is given by the hypergeometric distribution. This probability is given in Equation 2.7.

$$p = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}} = \frac{(a+b)!\,(c+d)!\,(a+c)!\,(b+d)!}{a!\,b!\,c!\,d!\,n!},$$

Equation 2.7 – Conditional probability of obtaining a set of values in a 2x2 contingency table, given the null hypothesis and row/column sums.

Within the equation, $\binom{n}{k}$ is the binomial coefficient and ! denotes the factorial operator. If the column and row totals in the table are known, only a, b, c or d respectively are needed to determine other values. We can calculate the probability of observing each of

these 4 values, conditional on the year and subject, with the formula above. We assume that student counts for each subject should be equally likely across years.

To determine how extreme the probability, p, is, we can modify the values of a, b, c and d, while keeping the marginal totals fixed (row and column totals), computing p each time. If we sum together the computed probabilities of all combinations of a, b, c and d which are lower than or equal to what was calculated for the original table – according to H_0 – this produces a p-value for the two-tailed variant of this test. If the p-value is less than some defined threshold such as $\alpha = 0.05$ (for a 95% confidence interval), then we reject H_0. For contingency tables of size greater than 2×2, Pearson's chi-squared test of independence can be employed.

Chi-Squared Goodness-of-Fit Test

This test determines whether a sample of data is derived from a given probability distribution (e.g. Gaussian), by estimating the distribution's parameters from the data [34, 35]. It is suitable for categorical data, where the expected amount of responses per category is at least 5.

H_0: A given frequency distribution does not differ from a specified distribution.

H_1: A given frequency distribution does differ from a specified distribution.

The test involves binning the data, obtaining observed counts from the data, O_i (obtaining the frequency distribution), and expected counts, E_i, based on the specified distribution. These are used to compute a test statistic which is distributed similarly to a chi-square distribution, when the counts become large enough (see Equation 2.8). A p-value is obtained for the provided data, indicating how extreme it would be to observe the test statistic value under the specified distribution.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

Equation 2.8 – Chi-Squared test statistic

Using the degrees of freedom of the data (number of possible categories to be counted - 1) and our chosen significance level, α, a critical value, χ^2_α, is computed which indicates the threshold for acceptance of H_0, or rejection. It falls in the right tail of the chi-square distribution. If $\chi^2 > \chi^2_\alpha$, we reject H_0.

Kolmogorov-Smirnov Test (two-sample)

This is a non-parametric hypothesis test to determine whether two datasets are sampled from the same underlying distribution. This distribution is not assumed, so it is valid for the test to be used when the data's distribution is unknown [36].

H_0: A given dataset was sampled from the same distribution as another dataset.

H_1: A given dataset's distribution differs from the distribution of another dataset.

The test statistic, $D_{n,m}$, can be computed via the absolute difference between the CDFs (cumulative density function) of the two samples, or otherwise through estimates of each CDF, via the empirical cumulative distribution function (ECDF), where n and m are the sizes of the first and second sample, respectively. A formulation is described in Equation 2.9.

$$D_{n,m} = \sup_x \left| F_{1,n}(x) - F_{2,m}(x) \right|$$

Equation 2.9 – Test statistic for the two-sample Kolmogorov-Smirnov test

Within this equation, sup is the supremum function (least upper bound) and $F_{1,n}$ and $F_{2,m}$ are the ECDFs of the first and second data sample, respectively. The computed test statistic, $D_{n,m}$, is used to determine how extreme the data are compared to one another, under H_0. At a given significance level, α, the rejection threshold for H_0 is determined by a critical value, $D_{n,m,\alpha}$:

$$D_{n,m,\alpha} = c(\alpha)\sqrt{\frac{m+n}{mn}},$$

where $c(\alpha)$ is the inverse of the Kolmogorov distribution, evaluated at α.

Equation 2.10 – Critical value for the two-sample Kolomogorov-Smirnov test

If $D_{n,m} > D_{n,m,\alpha}$, we reject H_0.

One-Way ANOVA Test

ANOVA (Analysis of Variance), is a method to determine whether data obtained from different groups (or levels) of an independent variable have the same mean [37]. This allows you to determine whether different groups of data have different effects on some response variable. The test assumes that the population of each data sample is normally distributed.

H_0: All group means are equal.

H_1: At least one group mean is not equal to the others.

Now for a short scenario: clustering some photographs of landscapes could yield 7 disjoint groups. Images within a cluster might have similar colour intensity, due to being taken at similar locations or times (desert vs meadow, morning vs sunset)? Images between clusters might differ on these aspects. We could collect subjective ratings on

the visual appeal of each photograph. A candidate response variable for each cluster could be the mean visual appeal rating per cluster. As the clustering would be implicitly performed based on pixel intensities, we would be testing whether different distributions of pixel intensities influence the mean visual appeal of photographs, differently.

ANOVA is formulated as a special case of a linear regression model. It is the sum of a group mean and an error term. Firstly, an independently measured attribute of each image, x_i, is needed. These could be our visual appeal ratings from earlier. α_j would be the mean of visual appeal ratings across all images, x_i, assigned to cluster, c_j. The model is formulated according to Equation 2.11.

$$y_{ij} = \alpha_j + \varepsilon_{ij} \, ,$$

Equation 2.11 – One-way ANOVA model formulation

In this model, y_{ij} is an observation, i is the observation index, j is the group index and ε_{ij} is a normally distributed error term with zero mean and constant variance: $\varepsilon_{ij} \sim N(0, \sigma^2)$. In order for H_0 to be disproved, we can find any pair of groups' means that are not equal. This can be done via an F-test, which uses the F-statistic. The F-statistic is formulated according to Equation 2.12.

$$F = \frac{\text{variation between group means}}{\text{variation within group}}$$

Equation 2.12 – F-statistic formulation

Using a precomputed distribution of F values based on data where H_0 is true (the F-distribution), we can determine how extreme a certain value of F is, given H_0. As F is more extreme under H_0 as it increases, there is a threshold or critical value

(corresponding to a significance level) which when passed, we can reject H_0. For our earlier example, accepting H_0 would indicate it is below the threshold and would indicate that per-cluster mean visual appeal is equal among all clusters.

2.5.3 Clustering

K-Means

K-Means is a clustering method which partitions data into a k clusters without overlap. Usually, this data is a set of vectors. On completion, each vector is assigned an index representing cluster membership. The k-means++ algorithm was used in our work [38]. It begins by randomly assigning a vector to a cluster. It becomes the first centroid, c_1, (centre of the new cluster).

k-means aims to minimise the sum of distances between a cluster centroid, c_j, and all vectors that are members of the cluster. This implies different distance measures can be used, depending on the data. A common metric is Euclidean distance.

After the first centroid is selected, distances from all other vectors, x_i, to c_1, are computed, and a new centroid, c_2, is chosen with probability p_{initial} (see Equation 2.13), creating a new cluster. p_{initial} is proportional to the squared distance between c_1 and the candidate vector, x_i.

$$p_{\text{initial}} = \frac{d^2(x_i, c_1)}{\sum_{j=1}^{n} d^2(x_j, c_1)}$$

Equation 2.13 – Probability of selecting a 2nd centroid in k-means++ algorithm

From now on, the centroid of each new cluster is computed via the distances between each vector and each centroid. Each vector, or observation is firstly assigned to its closest centroid. Next, a new centroid is selected with a probability proportional to the

squared distance between the closest centroid, c_p, to a candidate vector and that vector, x_m (see Equation 2.14).

$$p = \frac{d^2(x_m, c_p)}{\sum_{\{k;x_k \in C_p\}} d^2(x_k, c_p)}, where\ x_m \in C_p$$

Equation 2.14 – Probability of selecting later centroids in kmeans++ algorithm

The previous step is repeated until k centroids have been selected. This involves reassigning observations to their nearest centroid with each iteration, recalculating cluster centroids each time.

The key point is that k-means is non-deterministic, and so not every clustering will be the same. As k-means may also get stuck in a local minimum, it is good to obtain multiple cluster assignments by running the algorithm multiple times, taking the clustering which yields the lowest total sum of distances between observations and cluster centroids. This increases the chance that a clustering will be a global minimum.

2.5.4 Dimensionality Reduction

Principle Components Analysis (PCA)

PCA is a procedure to reduce the number of variables / dimensions used to represent some data, to principal components, or those which explain the most variance in the data [39, 40]. The procedure can be applied to linearly related data. This means that each observation (or vector) can be treated as a linear combination (sum and/or scaling) of a subset of the variables that form the observation.

Principal components are linear combinations of the original variables of each observation and are orthogonal to one another. The components are ordered according to decreasing contribution to variance in the original data. These form a set of basis

vectors with which to represent the data. For example, given a dataset of 15-dimensional observations, it may be that the first 5 principal components explain 85% of the variance in the data, while reducing the dimensionality of the data by 2/3. The transformation is not lossless however, as 15% of the variance is unexplained. Increasing the number of components would increase the variance explained. This is a trade-off to be made, dependent on your accuracy requirements.

t-SNE

If we have observations, X, which are high-dimensional and non-linearly related to one another (e.g. spiral-like or circular trends in a plot), we can use t-SNE to find an embedding or representation of the data within a lower number of dimensions which tries to retain relative distances between observations [41]. Observations are usually represented as vectors. Nearby vectors in the original, high-dimensional space are nearby in the low-dimensional space, and vice-versa. This allows one to visualise trends in the data via a 2D or 3D embedding that is produced.

The algorithm to find a t-SNE embedding takes multiple steps. Firstly, pairwise distances between each vector are computed (e.g. via Euclidean distance). The similarity of each vector, x_i, to other vectors, x_j (j not equal to i), is computed based on P_i, the conditional probability of x_i selecting each x_j, as its neighbour (see Equation 2.15).

$$p_{j|i} = \frac{\exp\left(-d(x_i, x_j)^2 / (2\sigma_i^2)\right)}{\sum_{k \neq i} \exp(-d(x_i, x_k)^2 / (2\sigma_i^2))}, p_{i|i} = 0$$

Equation 2.15 – Conditional probability of a point selecting another as its neighbour in t-SNE algorithm

This is represented via a Gaussian distribution placed at x_i, so if x_j is further away according to the distance measure, it is less likely to be picked as a neighbour, and vice-versa. The *perplexity* of P_i, represents the number of neighbours of x_i. It is based on the Shannon entropy of P_i, $H(P_i)$ (see Equation 2.16). As perplexity increases, the number of neighbours considered nearest to each x_i increases. This may reduce detail in the final embedding, as selecting more neighbours increases the probability of selecting candidates with more varied positions. If the perplexity is too low however, local changes in neighbour position most influence the overall embedding. It is therefore good to repeat of runs of t-SNE and check for consistency between embeddings. Adjusting perplexity for a given dataset can be helpful when locating global patterns.

$$perplexity = 2^{H(P_i)}$$

Equation 2.16 – t-SNE perplexity formulation

How are the vectors shifted into lower dimensions? t-SNE tries to minimise the difference between the Gaussian distributed points in X, and a randomly sampled set of points, Y, that are Student-t distributed in a lower number of dimensions. This is done via gradient descent with respect to the points in Y. Student-t distributions have a fatter tail than that of Gaussian distributions, so close points in X can be closer in Y, and vice-versa. The resulting embedding is produced in a non-linear fashion and is dependent on the initial data, so it may be difficult to analyse further (e.g. finding a shared basis with other data, for regression or classification etc.), but it is useful for visualisation.

2.6 Machine Learning

2.6.1 Artificial Neural Networks (ANNs)

ANNs are computing systems made up of a number of simple, highly interconnected processing elements (neurons), which process information by their dynamic state response to external inputs [42, 43].

These processing elements or *neurons* are mathematical functions, which output a value according to an input value, which can be a simply binary value in {0,1}, using a threshold, otherwise a rational or real-valued output.

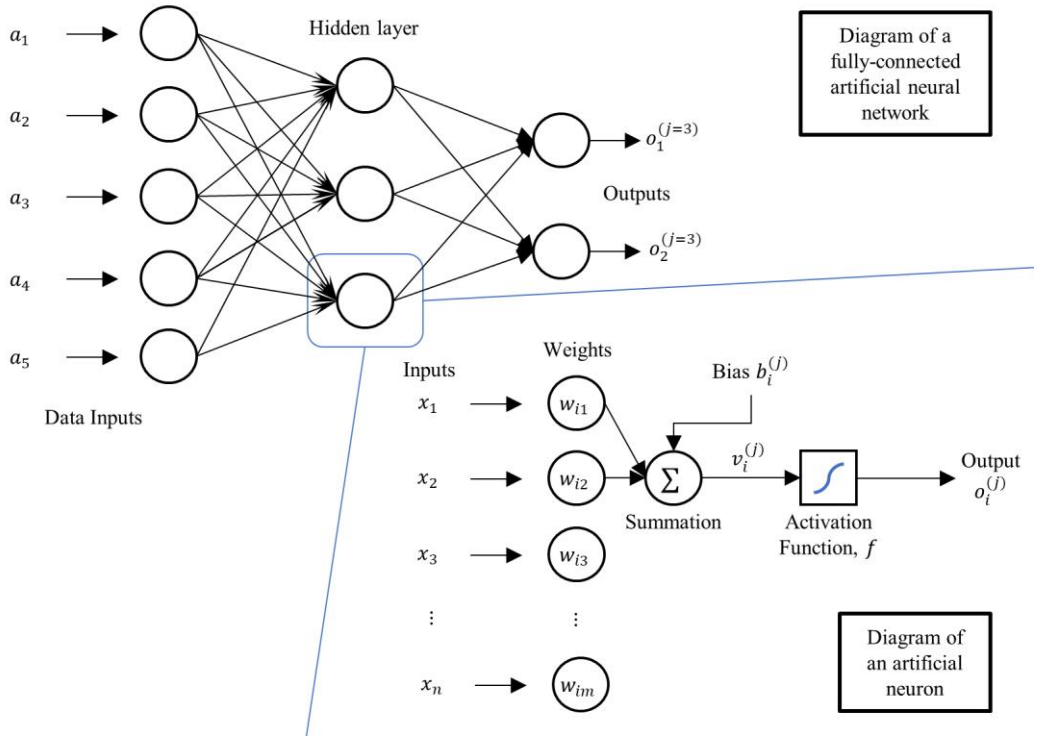


Figure 2.16 – Diagram of a fully-connected artificial neural network

As an example, in a fully-connected network (see Figure 2.16 for a diagram), the input may consist of a vector of 5 numbers, a (corresponding to the frequency of 5 keywords found in a section of text), which we call layer $j = 1$. Each element of a is provided to every neuron, i, in layer j=2. Each neuron multiplies the elements of a by individual weights (a vector of real-valued numbers $w_i^{(j)}$), one corresponding to each element and

then sums the values together (a dot product). The neuron then shifts the resulting sum by some real-valued number $b_i^{(j)}$ (a bias), and transforms it by an *activation function*, f. This is the output value $o_i^{(j)}$ of the neuron i in layer j. Each layer after the input can consist of multiple neurons, each having their own weights and biases, taking all outputs of the previous layer (it is symmetric), processing them to potentially output different values to the next layer. Applying each neuron of layer j to the inputs, gives us a vector of new values $o^{(j)} = f(w^{(j)}x^{(j)} + b^{(j)})$, or outputs for the neurons of the next layer. The number of values output is equal to the number of neurons for that layer. More precisely, the values of a have been transformed into the inputs for the next layer $j + 1$ where the outputs of layer j are the inputs of layer $j + 1$: $o^{(j)} = x^{(j+1)}$. As this can happen in a recursive manner, memory and processing time permitting, a network can consist of more than 2 layers.

An ANN acts as a function mapping a representation of input data, to an output value. Many ANNs have a learning rule or method by which to update the weights or *train* the network to predict existing outputs/responses which correspond to existing inputs – e.g. inputs are instances of the "10 keyword frequencies", and outputs are some "importance score" associated with the text that the frequencies were obtained from.

This learning method is usually the *Backpropagation* approach, which adjusts weights according to derivatives of activation functions with respect to their inputs, starting from the output neuron, multiplied by a *learning rate*, to control the magnitude of weight modification. Since these activation functions must be differentiable, examples are commonly: *tanh*, *sigmoid* or *softmax*. Sometimes, partially differentiable functions can be used, like *ReLU* (Rectified linear unit). This is not differentiable at 0, and so conventionally just outputs a gradient of 0 at that input.

The network should aim for generalisation: it should output a value close to a true/existing output, given by the input data, but also not be so specific that new data is classified incorrectly, or is associated with an output value that deviates too much from a target, in regression (overfitting = high bias; low variance). A *regularisation* method is used to do this, like was mentioned for metric learning. After training, given a new instance of data, neural networks can be used to classify data into one of many discrete options, or a regression model can be determined.

2.6.2 Convolutional Neural Networks

These are biologically-inspired variants of the multi-layer perceptron (MLP). These are neural networks with many hidden layers. A *perceptron* is only a classifier (it uses a threshold function to distinguish between classes), but a *multi-layer perceptron* can be used for regression (uses smoother functions, like *sigmoid*, or *softmax*). But, this is unrelated to the number of layers, and instead the *activation function* of an artificial neuron (what tells it to 'fire' or send a result to further layers).

The difference between an MLP and convolutional neural network stems from early work in the 1960s, on the cat's visual cortex, by Hubel and Wiesel [44]. The cat's visual cortex contains a complex arrangement of cells, which are sensitive to small sub-regions of the visual field – called a receptive field. These sub-regions tile the entire visual field (in sensors, this is known as the field of view, or FOV), and they act as filters local to each tile. Hence, they are well-placed to exploit spatially local correlation found within natural images [45]. At this time, two cell types were identified: simple cells which respond most to specific edge-like patterns within their receptive field. And complex cells which have larger receptive fields that are locally invariant to the exact position of the detected pattern (they are *translation invariant*). Spatially local correlation is

enforced by keeping a 'local connectivity pattern' between neurons of adjacent layers. i.e. the inputs of hidden units in layer j are from a subset of units in layer $j-1$. Each node in the layer acts as a filter. To ensure translation invariance, each layer uses the same parameters: *weights* (multiple/scale), and *biases* (addition/translation) for the activation functions. Many previous works have used convolutional neural networks [46, 47, 48, 49].

2.6.3 Other Types of ANN

Other types of neural network exist, such as *Autoencoders* [50, 51] and *Recurrent Neural Networks (RNNs)* [52, 53], which each have their advantages and disadvantages. If we use a certain type of neural network, it will depend on the source of the data.

Restricted Boltzmann Machines (RBMs) [54, 55] are another type of neural network formed of two layers, one visible and one hidden. Communication only exists between layers. The hidden layer is stochastic, or uses some element of randomness per neuron, to determine how inputs are transmitted. Weights are randomly initialised to provide this effect. Inputs to the network can be reconstructed in an unsupervised manner, by passing data forwards (computing the probability of neuron activations given weighted inputs) and backwards (estimating the probability of some inputs, given hidden layer neuron activations, also dependent on their weights) between layers. Since the weights are shared between layers, this provides a joint probability distribution over the inputs and neuron activations. We can adjust the network weights and therefore neuron activations to minimise the difference between the distribution of inputs vs. network reconstructed inputs. Stacks of RBMs form *Deep Belief Networks* [56, 57]. These generative models have mostly been superseded by variational autoencoders and generative adversarial networks.

Autoencoders are used to compress (encode) an input representation, then reconstruct (decode) it, using a number of hidden layers with less parameters (e.g. weights, biases), than the input size. These are also known as latent variables. But this can be done by just ignoring the parameters, and returning the identity function (doing nothing), still resulting in a perfect reconstruction. To avoid this, a denoising autoencoder introduces noise into the network at the input stage, forcing it to reconstruct a corrupted, or noisy form of its inputs. This acts as a regularisation on the network representation. Using 3D shapes as an example, regularisation causes the autoencoder to not exactly produce the original input shapes, but instead slightly varied versions. This should encourage it to represent a greater population of 3D shapes than its inputs, as it tries to undo the effect of noise on the input shapes, by learning statistical dependencies between them. These occur due to missing, or overcomplete input data, caused by the introduction of randomness. Too much noise can yield inconsistent and unhelpful results, however. If we want to generate new instances of a medium (e.g. images, 3D shapes), we may use an autoencoder to compress the input, and then provide that encoding to another neural network for decoding, classification or regression. Variational autoencoders inherit this structure but make normality assumptions on the distribution of the latent variables.

Generative Adversarial Networks (GANs) can learn to mimic distributions of data. Via a generator network, G, new data instances are generated. A discriminator network, D, is trained to classify/distinguish between 'fake' vs. 'real' instances. The weights of the generator network are adjusted, to make its output match what the discriminator network thinks is valid. Or in other words, fool it into classifying a generated data instance as belonging to the correct class. The final output is achieved via multiple iterations of generation, starting with random noise and comparison to actual input (e.g. images) via a discriminator network's classification. The representation the generator

uses to create images/examples is called the latent space, z. This competition acts as zero-sum game, as the goals of each network differ in an opposing manner.

RNNs are used to learn sequences of data, based on the previous $n-1$ elements in the sequence. If we want to analyse the frames of a video, we may use an RNN, since previous frames may help to determine later ones. We may similarly do so to predict the most probable future words or sentences given previous text. Earlier approaches were difficult to train using Backpropagation, as a sequence's length increases, due to vanishing gradients. So, the Long-term Short-term memory (LSTM) network was introduced [52]. The learning rate is fixed to 1, but this limits control over gradient propagation, so an addition to existing neurons was made to produce an LSTM unit, which uses gating mechanics (input, forget, output) to introduce a notion of memory. The neuron activations of the previous layer are summed with the activations of the current layer. Together with the inner activation of the LSTM unit, the resulting values are provided to a sigmoid function, to make sure values lie in [0, 1]. This value is set to be forgotten, used as future internal input, or set to be an output. Encoding input words according to their location in a vocabulary using 1-of-K coding (e.g. 1:written, 2:tomato, 3:envelop …) and using a *softmax* function for the output layer, provides the conditional probability values of each word given the LSTM network model/hypothesis. Training an LSTM network using a *cross entropy* loss, is equivalent to maximum likelihood estimation of the model.

2.6.4 Metric Learning

This is a type of learning method that computes a metric to measure distances between pairs of items (e.g. between photos, or between points on a 3D mesh). Some examples are: a similarity measure for fonts [58] and illustration style [59].

Formulation

Given a representation of some data, x, in the form of features or attributes, x_i, we can try to automatically find a metric, dependent on a matrix of weights, W, which determine the importance of descriptive or explanatory attributes of the representation [60]. W must be positive semi-definite. This means that the scalar resulting from $x^T W x \geq 0$.

Given two data instances' representations x and x', the metric looks like the following:

$$d(x, x') = \sqrt{(x - x')^T W (x - x')}$$

Equation 2.17 – Weight-based Distance Metric

Equation 2.17 is a weighted distance measure/function, e.g. Euclidean distance (where W is the identity matrix), or Mahalonobis distance. The latter takes into account the covariance between the features of the data instance (In this case, $W = \Sigma$, the covariance matrix). Metric learning approaches aim to adjust the weights to represent the differences between the input data.

We can learn the weights, W, to determine a custom distance measure, using pairs or triplets of constraints of the following form, for each instance of data, x:

- Must-link ($\mathcal{S}$) / Cannot-link ($\mathcal{D}$) constraints

$$\mathcal{S} = \{(x_i, x_j) : x_i \; and \; x_j \; should \; be \; similar\}$$

$$\mathcal{D} = \{(x_i, x_j) : x_i \; and \; x_j \; should \; be \; dissimilar\}$$

- Relative/Training constraints ($\mathcal{R}$)

$$\{\mathcal{R} = (x_i, x_j, x_k) : x_i \; should \; be \; more \; similar \; to \; x_j \; than \; x_k\}$$

The metric learning algorithm therefore tries to find the parameters of the metric, such that they best agree with the above constraints, in a *weakly-supervised* manner. This means that the algorithm has no access to the labels (classification/regression) of individual training instances: "it is only provided with side information in the form of sets of constraints, $\mathcal{S}, \mathcal{D}, \mathcal{R}$" [60].

This is usually formulated as a minimisation problem, with regularisation to avoid overfitting to the data provided. *Regularisation* aims to ensure that the learned dissimilarity/distance function doesn't exactly represent the data but has less variation (or is less complex), allowing for breathing room when new data is to be incorporated into the metric. This increases its utility in the real world, as the data provided to it is very likely to be only a subset of the real population's responses (all possible combinations).

The formulation is below [60]:

$$\min_{W} l(W, \mathcal{S}, \mathcal{D}, \mathcal{R}) + \lambda R(W),$$

Equation 2.18 – Loss function over weights and constraints with a regularisation penalty, scaled according to a parameter, λ.

Where $l(M, \mathcal{S}, \mathcal{D}, \mathcal{R})$ is the loss function (see Equation 2.18), which provides a penalty when training constraints are violated, $R(M)$ is a function of the weights, W, which provides the regularisation, and λ is the regularisation parameter (determines the scale of regularisation).

2.6.5 Vector Space Models

Approaches which use textual descriptions to understand shapes tend to use some method of word sense disambiguation, either using a database of word synonyms or a

vector space model of word semantics. Vector space models represent words as embeddings in a vector space, where words of similar semantics are mapped to nearby points [61]. See Figure 2.17 for a visualisation of a vector space model.

Vector space models can be *count-based* or *context-based*. Count-based models tally frequencies of word co-occurrence throughout a large corpus of text, typically with some matrix factorisation of these counts (e.g. Latent Semantic Analysis [62], *GLoVE* [63]). Context-based models employ the use of a word context or sliding window of words in a sentence, where words occurring in the same contexts have similar meaning (Skip-gram, Continuous Bag-of-Words). These can predict the most likely words to fit or follow the context, or vice-versa. *word2vec* is an example [64, 65]. Figure 2.18 provides a visualisation of these methods.

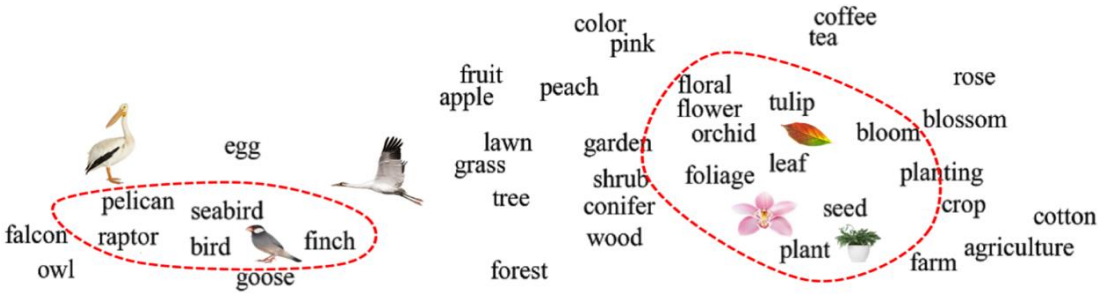


Figure 2.17 – Visualisation of a word vector space model, via t-SNE.

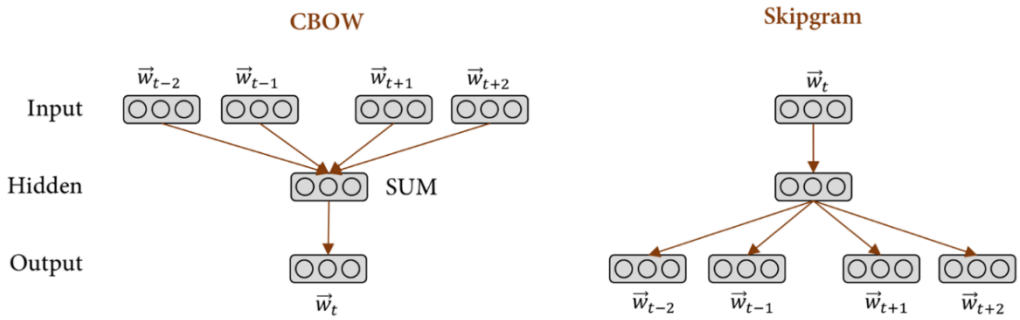


Figure 2.18 - A visual representation of the CBOW and Skip-gram vector space models, from a survey of vector-space representations of word meaning, by Camacho-Collado et al. [61]

The *word2vec* architecture uses a *Skip-gram* or *Continuous Bag-Of-Words (CBOW)* model [64, 65]. The *CBOW* model is based on a feedforward neural network language model, which aims to “predict the current word using its surrounding context”, or the collection of words before and after the current word in the sequence (also including it), by minimising the negative log probability of a target word given the context. The *skip-gram* model is similar, but the goal changes to predicting words in the surrounding context of a given word, rather than predicting the word from a given context. *GLoVE* performs global matrix factorisation of word and context co-occurrence statistics, given large corpora of text.

Since there are many combinations of contexts (word n-gram windows), to be compared against each word, a lower-dimensional representation of the context statistics is obtained by minimizing a "reconstruction loss", which can explain most of the variance in the high-dimensional data. The features of each row are lower-dimensional representations of the statistics, where each row is a vector representing each word in the corpora.

3 Related Work

This chapter provides a review of several works related to human perception of 3D shapes and 2D fonts, and other related fields. It provides potential questions to answer, based on research gaps in these works.

3.1 Overview

- **Saliency + Shape Perception**: The concept of Schelling meshes introduces the idea of an underlying subjective space of agreement between people, indicating which meshes attract attention within a group, given an abstract goal of agreement with others. In this sense, it is a form of task/goal-based saliency for discrete objects, which we call Group-level saliency. This section covers previous approaches in discovering and understanding properties of salient 2D images and 3D shapes, and methods for automatically determining these properties.
- **Understanding of Geometry:** We aimed to understand how people interact with and want to organise 3D shapes and 2D fonts, so we needed an understanding of how 2D images and 3D shapes are represented and processed. This section provides an overview of traditional techniques which compare one object to another, via explicit geometry processing approaches (approx. < 100 models), in addition to *Data-Driven Shape Analysis and Processing* [66]. The latter approach tends to involve the use of machine learning techniques that use implicit descriptions of shapes defined within the learning process and can be scaled to compare 1000s of models.

- **Machine Learning:** To predict which shapes are Schelling meshes, or otherwise, Specific fonts, we used regression models based on neural-networks. This section provides a summary of works which use machine learning for classification, regression and generation of 2D images and 3D shapes.
- **Crowdsourcing**: Crowd-sourcing was our primary method for collection of data on Schelling meshes and font Specificity, so we provide examples of previous crowdsourcing approaches for collecting data on 2D images, 3D shapes and some related topics.

3.2 Problem Statement

In the case where a single 3D shape is studied, 3D shape descriptors can indicate geometric properties of that shape, such as its curvature [67], or a notion of volume [31]. Image descriptors similarly indicate geometric properties of a 2D shape [13, 15]. Traditional saliency methods for images focus on mapping out objects in a scene (such as a photograph) which are most likely to be attended to or focused on. Ground truth data can be obtained by annotation via bounding boxes. Otherwise, regions indicating potential objects can be automatically proposed using convolutional neural networks [68]. Networks like these can classify new objects within an image [69] or indicate salient regions of an image, while simultaneously outlining detected objects [70].

But most existing shape descriptors or methods of saliency detection do not measure properties of shapes which are perceived relative to other shapes within a group (or *Group-level saliency*). So firstly, we wanted to determine how a group-level saliency measure of shape, could be defined. Overall, we aimed for better results in applications such as: search, visualisation or clustering, or to introduce new applications, via such a measure.

We therefore needed to survey current works and subject areas to determine if existing research could provide answers to these problems. Conversely, we also aimed to locate any relevant research gaps which could be investigated. Some of these would form the basis of the studies in this thesis.

3.3 Related Literature

3.3.1 Saliency + Shape Perception

This is a summary of works which have studied how humans perceive shape, with a focus on image and mesh saliency. We highlight existing results and potential research gaps.

Visual Saliency

Saliency is "the quality or fact of being more prominent in a person's awareness or in his memory of past experience" [71]. This is a very broad definition, and so what constitutes a salient element or object in a visual sense, has been a focus in computer vision for many years. Visually, saliency characterises parts of a scene (objects or regions) which appear to stand out relative to neighbouring parts, from an observer's point of view [72]. From now on, we refer to saliency by its visual interpretation.

Attention is a concept related to saliency. It covers all factors which guide or influence visual selection mechanisms from anatomical or computational viewpoints, whether they are 1) bottom-up, scene or data-driven or 2) top-down, goal-based or expectation-driven [72]. The former leads to saliency detection approaches based on differences between local elements of a shape, or *local contrast* methods. The latter leads to saliency methods based on structures located within a shape, or statistics obtained with respect to the whole shape. These are *global contrast* methods.

Local contrast: The greater the difference in intensity/position of a small element (e.g. pixel, patch, vertex, polygon) of a shape from neighbouring small elements, the greater the contrast between them. High contrast indicates a high saliency in the associated region where those elements belong. The concept of 'Centre-surround' differences in response, obtained at multiple scales/resolutions is a general approach to determining local contrast across a shape [73], in a manner similar to that of receptive fields in the human visual system. See Figure 3.1 for a visualisation of local contrast enhancement.

Global contrast: Global contrast methods predominately apply statistical methods across a shape to determine the contrast of regions or structures of a shape, with respect to the entire shape. For an image, this could be pixel intensity/colour deviation from the mean. In general, some distribution over the shape is analysed - e.g. a colour histogram.

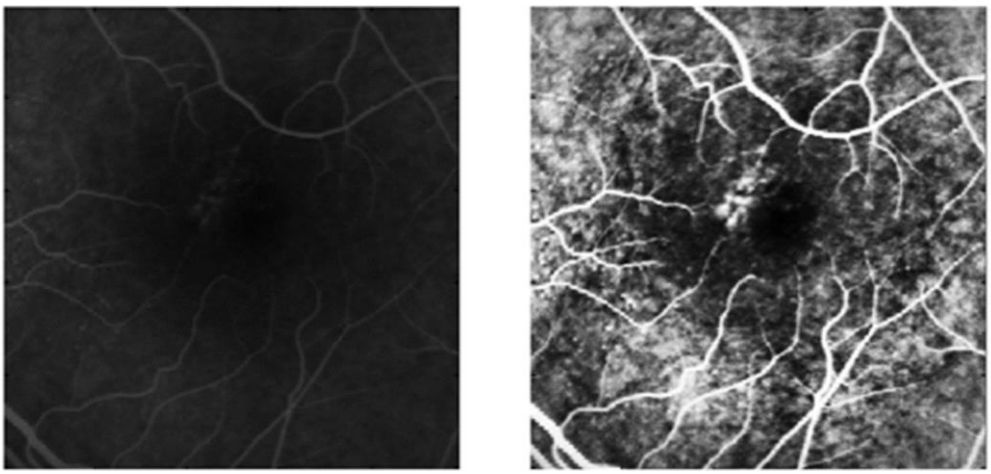


Figure 3.1 – Image of interior surface of the eye with and with contrast enhancement. (Left) original image (Right) enhanced image, showing greater image intensity at regions of higher colour gradient (e.g. veins) [74].

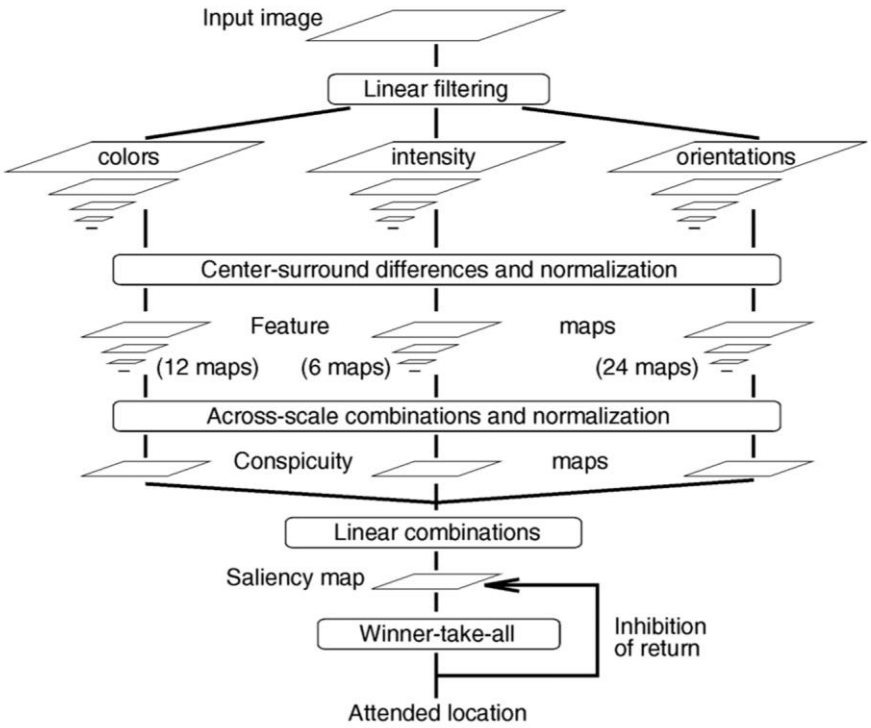


Figure 3.2 – Diagram of the ITTI98 model [73].

Processes of attention affect the responses of neurons within the visual cortex. For example, it is generally believed that cortical cells each respond preferentially to the highest contrast stimulus in their receptive field. Neural systems for attention take advantage of this mechanism, by effectively increasing the contrast of an attended stimulus [75].

Many works have attempted the creation of computational frameworks for visual attention or saliency prediction. A seminal example is *ITTI98* [73] (see Figure 3.2 for a diagram of the model). It is based on the premise that visual attention is dependent on many local features of an image, compared via centre-surround differences, and at multiple scales. The framework was based on the physiologically inspired model of visual attention by Koch and Ullman [76]. This hierarchical, centre-surround structure has been a basis for modern convolutional neural networks, which are used regularly in

computer vision (see *Machine Learning* section) for classification and regression based on shape.

A recent survey summarises the history of visual saliency models starting from early attempts at modelling saliency in a computational manner with the ITTI98 model [73], followed by approaches based on binary image segmentation, and the automatic creation of saliency maps based on hand-crafted image features [77], leading to more modern deep learning approaches based on artificial neural networks [70, 78].

Datasets

Along the way, work has produced a benchmark [79], dataset and baseline model for salient object detection [80]. The authors noted that there was not a standard definition of saliency, and that many datasets were biased, in that pictures: 1) contained only one object of interest, and 2) the object(s) was usually located at the centre of the image. Their focus was on "the relationship between where people look in scenes and what they choose as the most salient object when they are explicitly asked", in scenes with many possible objects of interest.

As a basis for creating a dataset with reduced elements of bias, a larger scale annotation of images was done using the *Judd* dataset (2009) [81]. The resulting dataset was named: *Judd-A* [80]. It contains "eye movements of 15 observers freely viewing 1003 scenes from variety of topics". Approx. 900 images were used. Although larger, the *Judd-A* dataset was more biased towards salient objects placed at image centres, than a dataset created by the authors (*Bruce-A*, 120 images).

Ideally, a large dataset on the scale of *Judd-A*, without this 'centre-bias', would be useful for testing saliency models. As a reasonable compromise, the Judd-A dataset was segmented into 667 and 223 on-centred and off-centred scenes, where the off-centred

scenes had a more varied distribution of salient objects throughout them, but with an empty centre. *Bruce-A*, and the off-centred images from *Judd-A* were found to have reduced centre-bias compared to two other existing datasets: *MSRA-5K* and *CSSD*.

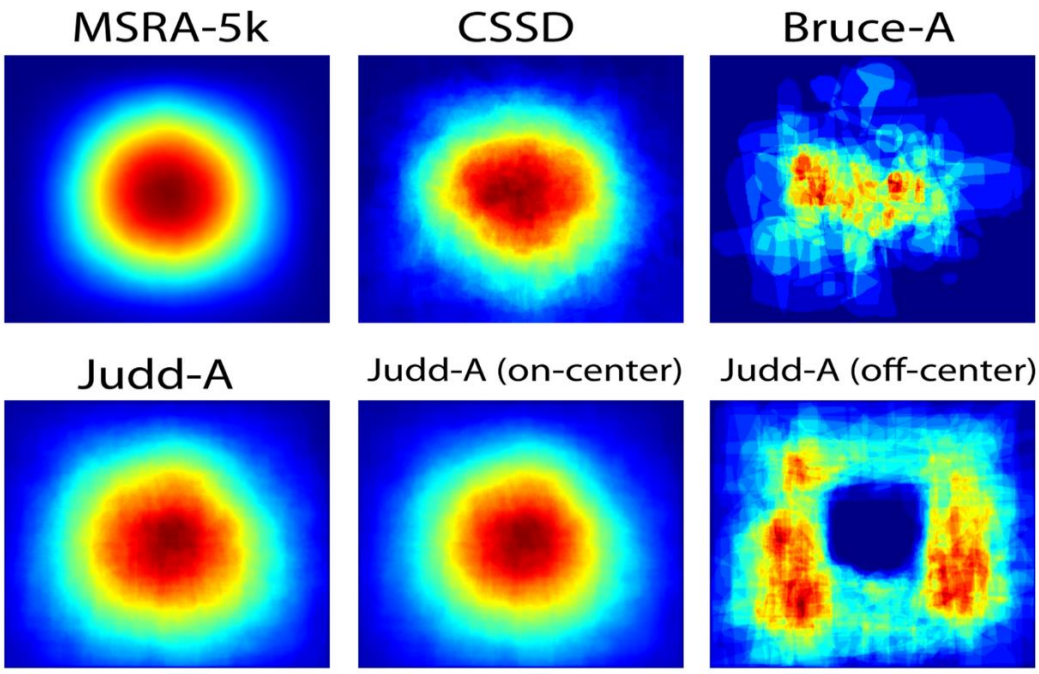


Figure 3.3 – Saliency annotation heatmaps for six image datasets (blue=low density to red=high density) [80].

Object sizes were also smaller for *Bruce-A*, and *Judd-A*, making saliency detection more challenging. The majority of salient objects in those datasets occupied less than 10% of the image. Figure 3.3 provides object annotation heatmaps for each dataset.

On average, the most salient object in *Judd-A* images, contained more superpixels (See *Image Saliency* section for a definition) than salient objects of images in the MSRA-5K and CSSD datasets, even as objects are smaller. The greater the number of superpixels in an image, or by analogy those of an object in an image, the more complex it was considered to be.

To measure and compare the bias and complexity of saliency datasets (including those mentioned above), Borji created a saliency model, which rewarded 1) high detection

rate, 2) high resolution, and 3) high computational efficiency [80]. It was designed to generate a saliency map based on an eye fixation prediction model (which provides a quick estimate of locations people may look at), and an oversegmented map of image regions (which refines the region level estimates of the saliency map). The model was on par with eight existing models on the MSRA-5K dataset but provided better performance over those models using the *Bruce-A*, and *Judd-A* datasets, which had reduced centre-bias. A large drop in performance was noted on eight existing models, given these datasets (40%-70%). One potential reason given was that although humans might sometimes have looked at an object more frequently, the image annotators may have instead chosen a different object, causing increased false positive results.

Properties of Salient Objects

Many properties of salient objects in images have been discovered. Elazary and Itti showed that locations most salient to human observers are likely to fall within the outline of an object (in 76% of the images, "one or more of the top three salient locations fell on an outlined object") [82]. Kathryn et al. discovered that when observers are explicitly asked to click on salient locations in natural scenes, standard saliency models most accurately predict the "explicit saliency selections and eye movements made while performing saliency judgments" [83], but not "eye movements made during image viewing without a specified task (free viewing)".

Borji et al. attempted to determine whether explicit saliency selections could explain free-viewing eye fixations. They asked 70 undergraduate students to draw polygons around the object which stood out most in an image. Through this annotation process, the authors concluded that saliency judgements agree with free-viewing eye fixations, significantly above chance [80, 84]. This result is an empirical reason for the collection of explicit shape selections in our Schelling Meshes work, where we treat them as

saliency judgements. This matched the predictive power of *ITTI98,* a standard saliency model [73]. An annotated dataset of 120 images (*Bruce-A* dataset) was produced as a result. From this conclusion, the authors suggested that the most salient object in a scene attracts the highest fraction of fixations [80]. Are the most salient objects in this sense, Schelling Points? If the objects were 3D shapes in a collection, would this apply? As a distinction, we could term these objects as *Schelling salient*, i.e. those objects which attract the highest fraction of fixations relative to other objects in the scene, when a person is given the task of matching objects that they expect others to focus on (with no communication allowed between participants). If we collect each object's fixation frequencies (or selection frequencies), we can treat Schelling saliency as a form of group-level saliency.

Image Saliency

Image saliency methods aim to automatically determine which are the most prominent or salient regions within 2D images. Traditionally, they were split into *local contrast* (focus on sub-regions of the image – e.g. at a pixel or patch-level) or *global contrast* (process the whole image at a time, based on finding salient structures throughout the image). Sometimes they are a combination of these local/bottom-up, or global/top-down approaches.

Image saliency methods work at a *single scale* or *multiple scales*. Some multiple-scale approaches are based on *Scale-space theory* as a basis for image representation at varying scales, by representing the image as a collection of smoothed images, parametrised by a smoothing kernel, which reduces high frequency / finer-scale components of the image. A Gaussian kernel is commonly used in a pyramid construction, where the image is repeatedly blurred (convolved with the kernel) and sub-sampled to a certain level. An approximate reconstruction of the source image can

be obtained by reversing this process, up-sampling and blurring an image from a higher pyramid level. The SIFT keypoint extractor and image descriptor uses this concept [13, 14]. See Figure 3.4 for a visualisation of the process.

Figure 3.4 - Gaussian and Laplacian Pyramids obtained via a test image of Lena. (Top) Gaussian pyramid obtained by blurring and down-sampling (Bottom) Laplacian pyramid obtained from subtraction of the Gaussian pyramid image at the current level, from the previous image.

An example of a local contrast approach ranks the similarity of image elements (pixels or regions) with foreground or background cues via a graph-based manifold ranking. Each image is represented as a closed-loop graph, with superpixels as nodes. Superpixel algorithms over-segment an image by grouping pixels that belong to the same object within it. Given a node as a query, the remaining nodes are each ranked based on their relevance to the given query, via a function that defines the relevance between unlabelled nodes and queries. This provides the saliency measure [85]. Another approach uses multi-level image segmentation, with over-segmentation being employed to produce superpixels. These superpixels are treated as image regions. A supervised learning method maps per-region feature vectors based on colour and texture information, to a saliency score, then combines the saliency scores across multiple

levels, producing a saliency map (an intensity image denoting salient parts of a source image) [86].

Another global contrast approach uses histograms based on pixel colour separation from other image pixels. Colour smoothing is applied to the histograms, to remove noise in distribution of colour across the image (to create smoother colour gradients), while retaining overall image resolution. The smoothed histograms are then normalised and combined with region-level saliency comparisons, to produce saliency maps and saliency cuts denoting the most salient objects in an image [87].

In the 2012 ImageNet large scale visual recognition challenge [88], results in many computer vision problems were greatly improved beyond the state-of-the-art – largely due to trainable convolutional neural networks. Since then, there has been a large shift towards using artificial neural networks for many purposes, including classification or object detection [89, 69, 90], region proposals [68], and image segmentation [91]. Convolutional neural networks or more generally, methods based on centre-surround differences at multiple scales, have become the predominant method of saliency detection and saliency map generation, spanning a continuum from local contrast to global contrast.

DeepSaliency is an artificial neural network-based model of object saliency in images [70]. The authors focused on modelling the semantic (type/category) properties of salient objects, via a multi-task learning approach, which aims to perform saliency detection and pixel-wise object segmentation (in conjunction) using a convolutional neural network. Weights and layers are shared for each task, allowing correlations between the two tasks to jointly produce features for object perception in images. Saliency maps are output, along with object segmentation maps (pixel-wise outlines of

objects in an image). To help preserve object boundaries in the saliency map, a superpixel representation of the original image is formed. Its adjacency graph is then used to minimise a Laplacian regularised objective function, to smooth over object boundaries in the saliency map.

Another approach aggregates multi-level feature maps captured at multiple resolutions, using a convolutional neural network. The network learns to combine these feature maps at each resolution and predicts saliency maps using them [92].

Other work has attempted to combine higher-level semantic features of images with low-level detail (object structure, with simpler geometric primitives). A feed-forward neural network is augmented with a pyramid-like pooling extension, and a multi-stage refinement mechanism for saliency detection. At first, a coarse prediction map is obtained via the feed-forward network. Another network is then provided with local information to refine the prediction maps, in stages. A pyramid pooling module is later used for aggregation of region-level features [93].

Moving closer to the philosophy of early attempts at visual attention modelling, convolutional neural networks have begun to be trained to predict visual saliency maps [94, 95] and eye fixations [96, 97]. Methods for saliency prediction of 360-degree images have been introduced, which use weighted centre-surround differences of image patches and image features via a colour dictionary representation [98], including eye scan-paths [99, 100] and head and eye movements [101].

More recently, a human fixation dataset has been built using multiple views of 3D printed shapes [102]. The authors tracked binocular movements, mapping pupil positions (4D) to eye fixation positions on 3D shapes. All fixations were modelled as a saliency probability distribution on the surface of each 3D shape – called a gaze density

map. Two prediction models were created using convolutional neural networks. The first model could predict gaze density maps for previously unseen viewing directions of each shape. The second model was trained on data from only a subset of the shapes and was able to predict gaze density maps for the unseen shapes (via cross-validation). In the latter case, using one of the unseen shapes to train the model for new views, improved the prediction accuracy of the model overall, suggesting that "certain shape features cannot be learned from the geometry alone", indicating that they are "higher-level" features of the shapes. They additionally find that stable features across different viewing directions tend to be associated with "semantically meaningful parts".

Some applications of image saliency include: *photo composition optimisation*, by cropping and re-centring an image to include one or more salient targets [103] and *image manipulation*, using previous work on 'patch distinctness' (regions of nearby pixels), combined with an object probability map [104]. The object probability map infers the most probable locations of the subjects of the photograph, according to distinct cues.

Co-saliency

Co-saliency methods aim to define where people look when comparing multiple images [105]. One method models co-saliency between an image pair, as a linear combination of a single-image and multi-image saliency map. The single image saliency map was produced using three existing methods in the literature, at that time. The multi-image map is computed using a multi-layer graph across the pair of images, in a pyramid representation. Each node of the graph is described by two types of visual descriptor based on local image appearance – e.g. colour and texture properties. Similarity between nodes is measured via a normalized single-pair *SimRank* algorithm, to compute a final

similarity score [106]. Clustering techniques have been produced to detect co-saliency [107], in addition to methods which perform hierarchical segmentation of images [108].

More recent works on co-saliency detection have been produced, which employ convolutional neural networks [109]. A metric-learning based framework for co-saliency detection has been introduced [110], in addition to a method to segment objects common to a set of images [111]. A notion of *group saliency* has been introduced, with an algorithm to locate and segment salient objects within image collections, by maximising similarities between images and *distinctness* within images [112].

Schelling saliency in images could be interpreted as a task-based measure of co-saliency between the objects of an individual image, since objects must be selected relative to others in view. Differently to the group saliency approach mentioned earlier [112], there would be no direct, between-image comparisons. To account for this, we could extract the objects from every image, treating them as if they were sampled from the same image. In the 3D shape case, this would be like combining shapes from different classes into one overall class (e.g. combining chairs, tables, plants, bottles etc. into one class). Here, object selections may reflect a form of group-level saliency. In the most general case, a 3D scene or arranged collection of 3D shapes would be analogous to a 2D image, where whole shape selections would be made within the collection. This collection would be representative of a shape population similar in geometry. For example, the population might be a class of 3D shapes which have similar surface detail and volumetric structure relative to each other (e.g. similar curvature distribution, or per-vertex SDF distribution across all shapes), when compared to other classes.

To reduce non-geometric factors from influencing selections however, this requires shapes to be presented with consistent colour and lighting. Without colour or lighting

changes being factors, *global rarity* could be the main distinguishing factor behind Schelling saliency in shape. Global rarity is a concept in visual saliency which states that the rarer, or more unique objects or regions are across an entire scene, the greater overall attention on the scene, and the higher the saliency [113].

Mesh Saliency

Mesh saliency has been defined as: "enabling a machine system to automatically reason about which points, or regions of a 3D polygonal mesh are perceptually important" [114].

As polygon meshes can approximate any 3D shape, this definition can be extended to other 3D shape representations. Approaches include: spectral methods [115, 116, 117, 114]; curvature-based methods [118, 119, 120]; heat diffusion inspired approaches such as the *Heat-Kernel Signature* [121, 122] and other methods, such as the *Shape Diameter Function* [31]. Although not necessarily a cause of saliency in meshes, symmetry detection can inform it, and is a related topic [123, 124, 125] – whether intrinsic (preserves geodesic/curved distances over a 3D surface) [126, 127, 128, 129] or extrinsic (dependent on the units/co-ordinate system to measure the shape; has invariance under rigid transformations) [130].

Liu et al. have introduced a survey on mesh saliency algorithms and their applications [131]. They defined *saliency* as the portion of visual information "that is visually interesting" and "filtered", from the remaining superfluous information. They describe ‘saliency detection’ as imitating “ways of seeing", based on scientific studies of theoretical computer science and human perception.

Approaches were categorised into *local contrast* and *global contrast* methods, as with *image saliency* methods. Approaches of the former category focus on determining the

"most representative salient elements" of a surface based on differences between neighbouring primitive/small elements, whereas approaches of the latter category aim to determine the most salient structures or components (constituted by local elements such as vertices, pixels), via statistics found across a shape.

They noted that most current approaches are *local contrast* based - some of which include: *single scale* approaches or *multi-scale*/hierarchical approaches, which are point-wise, based on clusters, or are based on frequency spectra. *Global contrast* methods can involve clustering [132] or the use of conditional random fields [133]. Other methods use a combination of local and global aspects of shape [134].

There has been much work in computing the visual saliency of meshes, originating with the “Mesh Saliency” work [135]. Examples include the use of salient geometric features for the purpose of partial shape matching [136], the identification of distinctive regions of a mesh’s surface [137] and the consideration of global information from the spectral attributes of a mesh [114]. Related to this topic is the subject of *mesh segmentation*, as in some cases a mesh must be segmented approximately uniformly for a saliency detection algorithm to be used [138, 139, 140, 141]. Mesh saliency can assist with *shape correspondence* [142], where the task is to match points/regions on one shape to points/regions on another, even as its pose changes.

We now detail some examples of *local* and *global* contrast mesh saliency methods. Shtrom et al. have produced an algorithm for detecting salient-regions in 3D point sets [143], using a *distinctness* measure applied at multiple-levels of abstraction in the point set. Their work aimed to take into account the hierarchical nature of the human visual system, by analysing a point set at multiple scales, looking at smaller features, as well as larger regions of the point set. The *Fast Point Feature Histogram* (FPFH) [144] was

employed, to form a type of centre-surround description of variation around each mesh vertex, following Itti's computational framework for modelling human visual attention (*ITTI98*) [73]. The *FPFH* describes the relative angular direction of mesh normals with respect to each other (within a spherical region), centred at each mesh vertex. It was said to cope with “extremely large sets, which may contain tens of millions of points" and was produced without the topological information used by many other mesh-based saliency algorithms. Two applications were provided via "a set of the most informative viewpoints" and the suggestion of "an informative city tour given a city scan".

Later work resulted in a clustering-based approach to salient region detection in point sets. [132]. The *FPFH* was also used in this work. Point sets were segmented into smaller clusters, via fuzzy clustering. Then a measure of each cluster's uniqueness was formed along with a spatial distribution of each cluster, to form a cluster-level saliency function. From this, the probabilities of points being contained in each cluster were used to determine point-wise saliency values. The measure of uniqueness used was based on Shtrom et al.’s work [143]. The authors evaluated their algorithm's saliency predictions, against a 3D interest point detection benchmark [145], showing that their algorithm was better at detecting ground-truth salient points, but at the expense of detecting more "uninteresting" or less extreme points (the emphasis was on recall, over precision), than the *Heat Kernel Signature* (*HKS*) [121]. The earlier mentioned benchmark contains interest points on surfaces, which may not necessarily be perceptually-based, unlike human selected points. If obtaining less false positives is of the essence (i.e. aiming for only salient points) over detecting a large amount of salient points, the *HKS* provides better results.

Additionally, Tasse et al. have provided a quantitative analysis of saliency models [146], including evaluations of the earlier mentioned point set approach [143], and cluster based approach [132], a spectral processing method (using a Laplacian over a mesh) [114] and a PCA-based method, against a dataset of 4800 range scans, with associated ground-truth data. The PCA-based method used the *Fast Point Feature Histogram* (FPFH) which was also employed in other works [143, 132]. The Schelling points dataset of Chen et al. [6] was taken as ground-truth data. Range scans were collected from the SHREC'07 dataset [147].

Evaluation metrics included the *Area under the ROC curve* (AUC), based on the *Receiver Operating Characteristic* (ROC). The ROC curve is formed by separating saliency map points into salient points vs. non-salient points. For different *true positive* thresholds of saliency value, the *true positive* rate vs. the *false positive* rate is produced. This is used to compare saliency models in images, where an AUC of 1.0 is associated with an ideal saliency model. The *Normalised scanpath saliency* (NSS) measure was also computed. It can be used to compare 2D saliency maps to human eye fixations, measuring the saliency values along a user's eye scanpath/trajectory. The authors considered points selected by users in their ground truth data, as fixation points, also agreeing with the visual saliency results obtained by Borji et al. (see *Visual Saliency* section of this chapter) [80]. The *Linear correlation coefficent* (LCC) was also employed to measure the strength of the relationship between ground-truth selections and a saliency map prediction. The ROC, AUC, and LCC measures were noted in the 2016 survey by Liu et al. [131] as not often being applied in mesh saliency evaluations, but often so in visual saliency studies. Other noted evaluation measures of this kind were: *Precision-Recall* and the *F-Measure*.

The results of Tasse et al.'s saliency model evaluation [146] suggested that the point-set method [143] performed better at saliency detection over multiple classes of shape, with the authors' PCA based method, and cluster-based approach [132] close behind, outperforming the spectral processing approach by Song et al. [114].

Shape perception is another topic related to mesh saliency. One can perceive saliency on virtual meshes from a visual [135] or tactile [148] perspective. The visual salience of 3D printed objects has been measured [149]. Other works in shape perception have also predicted and validated shape perception with eye-tracking devices [150, 151], explored the preferred views of 3D objects [152], and developed a perceptually-based preference model for 3D printing orientations [153].

There are properties of shapes that humans can perceive including: the scale of 3D models [154], the relation between shapes and their colours and materials [155], and the depth of 3D objects in a single image [156]. The shape of a virtual object can also influence how the material reflectance is perceived [157]. Many such properties exist [158].

Schelling points selected on 3D meshes have been interpreted as a measure of saliency in polygon meshes. Could this be translated to the meshes themselves, treating them as individual selections within a collection of meshes (See Figure 1.3)?

Summary

We studied existing saliency approaches based on direct processing of geometry to determine what utility they can provide. This is to provide a baseline for comparison to our approach, or to show differences between these approaches and ours based on human perceptual data.

Image saliency techniques aim to automatically determine perceptually salient aspects of images. Mesh saliency techniques similarly aim to automatically “reason about which points, or regions of a 3D polygonal mesh are perceptually important". Within Mesh Saliency, the concept of Schelling saliency has only been applied to 3D meshes in one previous work [6], as Schelling points. This leads us to an open question: can we extend the notion of Schelling points on meshes, from points on 3D shapes, to the shapes themselves? Previous work has concluded that explicit shape selections agree with free-viewing eye fixations [84, 83, 80]. Therefore, to approach this problem, we can collect explicit shape selections, made in the Schelling context. Careful experimental design is required to make sure that experiments cannot be gamed, reducing bias and increasing consistency in obtained results.

Most existing mesh saliency methods to our knowledge, cannot be used as a basis for group-level saliency measures, as they study individual shapes at a time, determining salient regions of those shapes. A point set saliency measure potentially could be used for this purpose, but the placement of shapes within the scene would be arbitrary, and object boundaries would be unclear. Instead, we could potentially use indirect descriptions of shapes, based on their perceived attributes. This would ensure that human perception is directly taken into account. For 3D shapes, this induces a top-down or goal-driven viewpoint of saliency in meshes, closer in principle to global contrast methods, but the space being analysed is complementary to that of the shape geometry – e.g. the space of shape creativity ratings. In the case of fonts, we may be able to understand them via word-level descriptions (rather than image saliency methods), similarly to the Image Specificity work [7].

There are existing works on *Co-saliency* [105] and *group saliency* [112] in images, but not 3D shapes. The former approach studies where people look when comparing multiple images. The latter type of work develops measures of saliency based on maximising both similarities between images and the distinctiveness of objects within images [112].

G*lobal rarity* could be the main concept behind Schelling saliency in shape. But could the Schelling concept of selection while thinking about other people's potential selections, affect this? A mesh saliency approach exists which is based around the concept of global rarity [113].

Despite this focus on 2D images and 3D shapes, at the end of the day, all shapes are geometric in nature. Geometric representations have trade-offs in ease of processing and resolution. Determining which representation is suitable for a given situation, requires an understanding of geometry.

3.3.2 Understanding of Geometry

To determine geometric representations and descriptions of 2D/3D shape which 1) had ideal resolution and properties for later measurement and analysis, and 2) could be used for comparison to our data-driven based approach based on human-perception, we needed to review existing works on shape measurement and description. This yielded shape representations, descriptors and algorithms of varying kinds and uses. We summarise some of these and their applications.

Data-Driven Shape Analysis and Processing

Traditional geometry processing techniques work for shape comparison, retrieval and correspondence, between small groups of shapes (around 2-100 shapes). However, a

different set of methods are required as datasets increase in size. These methods are more *data-driven* and imply the use of larger-scale optimisation and machine learning techniques. They operate on explicit or implicit shape descriptors and can be scaled to compare 100s to 1000s of models. Explicit shape descriptors are manually designed to perform specific tasks, whereas implicit ones are those which are represented through an extraction process defined in the learning process itself. An example is that of translation invariant local descriptors, obtained via convolutional neural networks. We summarise some existing methods and their applications in this section of the chapter.

Shape modelling and geometry analysis are well-developed topics [159, 66], with work done on many different problems. Earlier works focused on topics such as: 3D shape feature computation [9], detection of partial shape matches [136], establishing 3D model benchmarks [139] and performing 3D mesh segmentation and labelling [140].

Structure-aware shape processing [159] is a survey of methods which describe and use the arrangement and relations between shape parts (inter and intra) at a higher-level, rather than local geometric details. *Structure-aware shape processing algorithms* aim to link object function with shape geometry. *Data-driven shape analysis and processing* [66] is another survey, detailing the new approaches being taken to analyse large amounts of 3D shapes, and the new methods of shape understanding that come from this, which include: *shape segmentation; shape reconstruction; scene analysis and synthesis; interactive shape modelling and editing*, and *generative models* for 3D shape creation. However, the use of these is beyond the scope of this thesis.

Given a skeleton model of a body, an algorithm has been developed to determine the pose a person should take to operate a vehicle, based on its geometry [160]. A method for automatic recognition of functional parts of man-made 3D shapes, using part-wise

semantic (category/type) correspondences between those shapes, has also been produced [161].

Later work took this further to attempt to determine how objects interact together in a given 3D scene, dependent on their geometry, to work towards a "geometric functionality descriptor" via a notion of "interaction context" of geometry [162]. The descriptor tries to extract some semantic information about the purpose of an object. An example of a dinner table separates the chairs to sit on, from the table, and from the crockery and plates on the table. When comparing a desk, a trolley/cart and a shopping trolley, a traditional descriptor (LightField [163]) placed the desk much closer to the cart. Whereas, the authors' descriptor places the cart much closer to the shopping trolley instead.

Additionally, the authors attempted to produce a combined description of semantics and geometry, via a continuous approach. The semantics are defined in terms of the parts of a segmented object. E.g. for a horse model, each leg is labelled with the same tag, rather than individually, since they are all similar. This was done in terms of geodesic distances between parts (*geodesic distance* is the curved distance across a surface from a point A to B, both which lie on the surface).

Generative Models

The *ShapeNet* dataset, introduced in 2015 [8], has enabled people to work on shape classification and generation tasks of different kinds. One example involves the use of convolutional neural networks for object recognition, done via voxel grid classification [48]. Using a similar model, the orientation of 3D shapes has also been considered, by treating the optimisation problem as having multiple tasks - predicting both the shape's class and pose [164]. The *ShapeNets* model provides a convolutional neural network-

based representation of voxel grids (within different classes), which can be used to both classify, and generate new voxel grids [47]. Other existing 3D model datasets include: the *Trimble 3D Warehouse* [2] and *Princeton Shape Benchmark* [165].

Work by Kanezaki, has produced a convolutional neural network-based model which attempts to simultaneously classify an object (via an image), and determine the best view of that object, under its predicted category [166]. This 'front' view is obtained via a rotation path from the image coordinates. The goals of the method were to: 1) select the best view for object classification, as the 'front' view of an object category and 2) select the 'front' view which best matches the category of images. Kanezaki argued that the two problems could only be solved by jointly learning a solution to the object classification and pose estimation problems. N*M possible predictions were produced, via N = number of object categories and M = number of views. Probabilities were assigned to each possibility, with the highest probability of a category, taken as the 'front' view.

Liu et al. took a similar approach, using 3D shapes in the form of voxel grids, where their model first learns the shape's class [167]. The shape and its class are then provided to individual regression networks, one for each of the possible shape classes/categories. The output of the each of the networks is a 3x1 vector, corresponding to a predicted orientation of the given shape, for each shape category [167].

Sharma et al. performed an analysis of deep learning methods which aim to learn shape distributions from large scale collections of 3D CAD models [168]. The authors discovered that the training process, as well as the resulting representation of shapes, were “strongly and unnecessarily tied to the notion of object labels”.

From these insights, they worked on a generative approach to 3D shape representation, producing a 3D convolutional denoising autoencoder, *VConv-DAE*, which is trained in an unsupervised fashion, using a binary voxel grid encoding of input shapes [168]. Voxel grids of dimensions $24 \times 24 \times 24$ were used, as in *ShapeNets* [47]. The same dataset as was used to train *ShapeNets*, was also used to produce the authors' model (see 0 *Machine Learning* for more details on autoencoders).

The authors' showed that their model (a stacked convolutional autoencoder) outperformed some other generative methods, such as a standard convolutional autoencoder (CAE), or the *ShapeNets* model, which used a convolutional deep belief network, in most of their shape categorisation and completion tests, with less variance in the output voxel grids' error (proportion of incorrect voxels).

As a test of 'completion' error, the authors evaluated their network based on a more structured form of noise, where the aim was to represent occlusion of objects when sensing them via a camera, for instance. To simulate this, n randomly selected slices of an input voxel grid were removed, and reconstruction error was calculated for an input shape, given a percentage removal of the voxels within it.

Overall, their results suggested that future deep learning approaches to 3D shape reconstruction should involve some form of generative model of 3D shape distribution and introduce noise for regularisation/generalisation purposes. They state this may possibly be useful even for classification tasks, since the parameters/weights of a generative model may be used for other tasks than just reconstruction, transferring knowledge (geometry, topology) of the shape distribution, to other domains.

A different form of generative model: *Generative Adversarial Networks* (GANs) [169], have already been used to generate images, from a high-dimensional ‘latent space’ of

features, learned via convolutional neural networks [170], in addition to 3D shapes [171] (see 0 *Machine Learning* for more details on GANs).

A method for unsupervised learning of image representations has been developed using with convolutional GANs. Arithmetic can be applied to the learned latent space of images, in a similar way to word embeddings – a kind of 'vector arithmetic for image semantics' – where images of similar class are closer together in the latent space [172]. See Figure 3.5 for some examples of generated images. Other work has attempted to make latent spaces more interpretable [173].

Convolutional GANs have been used to generate images of chairs, tables, cars [170], produce images based on perceptual similarity metrics [174] and predict visual saliency maps [94, 95]. Other generative models have been produced to work on more complex problems, such learning approximate inverse projection transforms of an image (via convolutional neural networks), to output a possible 3D shape which produced it, using multiple layers of convolution and de-convolution operators [175]. Models like these can generate new images of an existing object, with variations in pose and lighting.

Rezende et al. have developed an unsupervised learning method for the creation of 3D structure from images [176]. Due to this work, generating 3D shapes from existing voxel grids, or multiple views of a shape to a less accurate degree, became possible. Future work in this area could help aid people that work on 3D modelling, since generation of 3D shape examples via images could save time at the earlier stages of shape design, for example: in furniture design. The *ShapeNet* dataset was used for their training data, as well as an extruded form of the MNIST character image dataset.

Generative models have also been applied to the problems of shape correspondence and segmentation of surface-based representation of 3D shapes, obtaining probabilistic representations of point correspondences and part segmentations of input shapes.

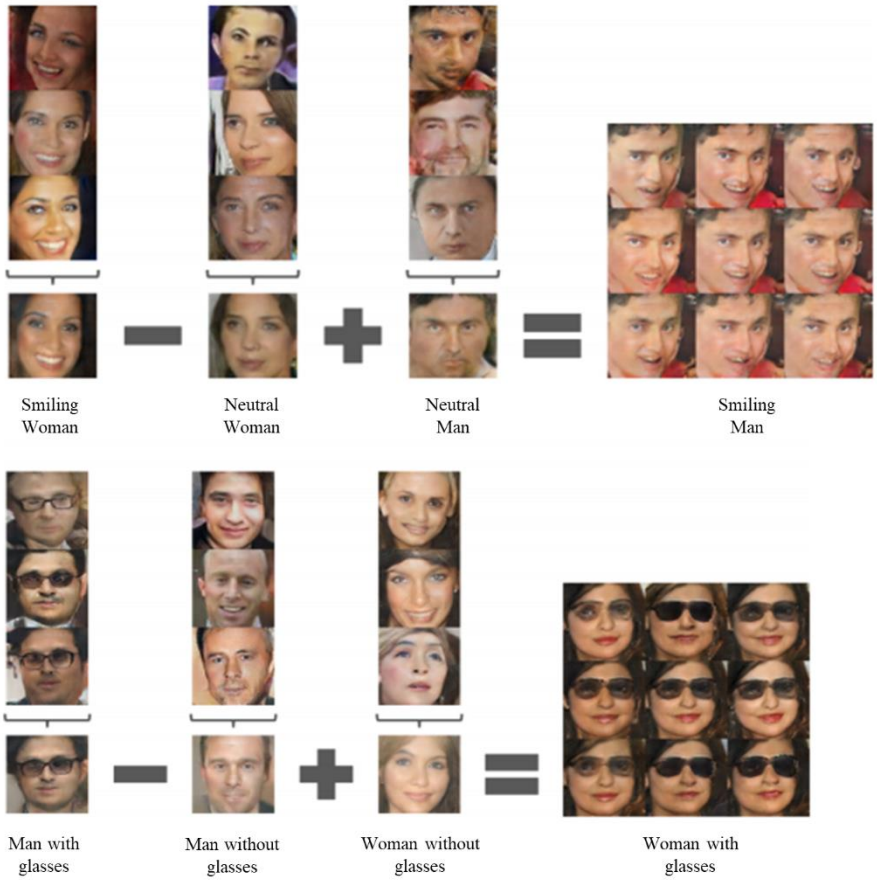


Figure 3.5 – Generated image variations produced via arithmetic in a latent space of image embeddings learned via a convolutional GAN [172]. (Top) Notion of 'smiling' is retained. (Bottom) Notion of 'wearing glasses' is retained.

One work does this via a deformation model, based on a conditional random field over different factors relating to segmentation, correspondence, etc. of part labels, to a given a set of surface points [177].

Other works have used generative models to learn the visual similarity of products or objects [178], classify 3D shapes [49], find dense correspondences between scans of humans [179], compute style similarity functions for 3D shapes [180], hierarchically

segment and label of shapes [181], solve shape classification, retrieval, and segmentation tasks [182] and create character motion synthesis and editing tools [183].

Text-based Description and Analysis of Shape

There exists work in automatically tagging 3D models with labels or words [184], which improves text-based search for 3D models in this case. Work also exists where humans provide text labels. Chaudhuri et al. collect data on the strength of semantic attributes (or mainly adjectives) to describe shape parts, which is then used in an interface for creating novel virtual creatures [185].

Descriptions of 3D shapes are common in terms of geometric features. For a shape, a bag-of-words approach is a set of representative descriptors of the shape, obtained via vector-quantization (e.g. by k-means clustering). Even if a bag-of-words approach is used to represent shapes, the "words" are usually geometric features [186]. Streuber et al. collected data on the ratings of words for describing body shapes, which is then used to generate new body shapes from verbal descriptions [187].

The ESP game [188] collected words that describe an image from human participants. Recently, there has been much work in Computer Vision on the problem of image captioning [189, 190] to generate a description of an image automatically.

The concept of *Image Specificity* [7] inspired our font Specificity work. As part of that work, humans were asked to describe an image in the form of sentences. From these descriptions, a measure of the consistency of sentence descriptions associated with each image, was produced. This was called *Specificity*. Specific images were found to be memorable to some degree (corr.=0.33, $p<0.01$), had less variation in sentence lengths provided as descriptions (corr.=-0.16, $p<0.01$), but sentence length itself was considered

to have no effect on Specificity (corr.=-0.02, $p<0.64$). There were additionally correlations with median object area (corr.=0.16) and mean object area (corr.=0.14).

In 2016, a survey on the automatic generation of descriptions from images, was produced [190]. The authors compared existing models, datasets and evaluation measures on the topic, and classified methods into one of two groups: models which treat description as a generation problem, using meaning extracted from images, or as a retrieval problem over a "visual or multimodal representational space". They note that the goal of answering questions about images, or "*Visual Question Answering*" is a recent one and is still an open challenge. The survey contains many examples of recent work at the time, which used neural networks for this topic.

Recently, work has attempted to take a multi-modal approach to image saliency, based on image annotations and textual descriptions of images at a sentence-level [191]. The authors proposed a word-weighting scheme to extract visual and 'verbal' saliency ranks to compare against each other. They compared the different ways that a human and their saliency model looked at and described images, indicating that this information could provide reliable information to an image captioning model. Some low-level and semantic-level features relevant to visual and verbal saliency consistency are provided, and the authors show that the features can be visualised and integrated into a model to predict the consistency between the two modalities for an image dataset (given both kinds of annotation).

Many image captioning approaches use a convolutional neural network pre-trained for classification to represent image features. Image features are then fed into a LSTM recurrent neural network to model the language word-by-word, considering all previous

words. These networks are trained with a maximum likelihood approach (*softmax* output and cross entropy loss function).

Recent methods use variational autoencoding or generative adversarial nets (examples shown in the survey), which both generate a latent space of captions to be sampled from. But it is difficult to interpret this vector space, and so it is difficult to control how relevant certain captions are. The captions are also not necessarily based on prior language information or semantics. One approach [192] tries to not only "condition each word on the image, and all previous words in a sentence", but also conditions on a part-of-speech tag sequence, which indicates the allowed structure of topics in the caption. Sampling different part-of-speech tag sequences provides different captions for the same image, allowing a diversified set of descriptions to be generated per image, with some level of control via the part-of-speech tags.

Recently, a survey of methods on the vector space representation of word meaning has been produced [61]. It describes the theoretical background behind vector space models such as *GLoVE* [63] and the relatively low-dimensional word embeddings that they produce. The authors denote the "meaning conflation deficiency" as "the inability to discriminate among different meanings of a word". It is an ambiguity problem which arises when a word's many different meanings are represented with a single vector. Each individual meaning of a word is a called a *word sense*. They state that most words in word sense databases, such as *WordNet* [193] tend to be *monosemous* (have a single meaning), but frequent words tend to have more senses (are *ambiguous*), according to the *Principle of Economical Versatility of Words* [194]. So, they indicate that it is important to capture the semantics of ambiguous words as they are used in text. They categorise methods for learning distributed semantic representations of word meaning

into two main approaches: *unsupervised models* which learn word senses directly from text corpora, or *knowledge-based systems* which employ the use of "sense inventories" or collections of existing word senses put into some form of categorisation or structure.

Summary and Research Gaps

Previous work has studied how people select points on the surfaces of different 3D shapes, treating points/regions where people most agree when they are unable to communicate with others, as Schelling points. [6]. The question of extending the notion of Schelling points to whole 3D shapes, requires us to determine consistent measures of Schelling saliency between shapes. We can do this in many ways (relative shape selection, relative shape comparison or degree of agreement), but would need a method to compare their results. For example, there may be commonalities in Schelling meshes across different classes of shape, which can be exploited for this purpose. Hence, we would need to produce a Schelling saliency-based scoring mechanism for shapes. Using these scores, we can rank the Schelling saliency of shapes. Since we would directly collect human-perceptual data on shapes, that would avoid objects always being closer geometrically rather than in their function, application or meaning. E.g. a desk may be close to a trolley/cart than a shopping trolley, if we study multiple views of the shape (consider some form of rotation-invariance), but a shopping trolley is functionally closer to the cart [162].

There is also an underlying question in the sufficiency of geometric description (e.g. depth images, voxels etc.) or a more direct interpretation/representation of shape based on human perception, for Schelling score prediction, and which approach is best for this. Important geometric attributes for explaining Schelling point distributions over mesh vertices include: min curvature, Gaussian curvature, intrinsic symmetry, Shape

Diameter Function (SDF) values [31] and other attributes [6]. Based on this, curvature, symmetry or SDF based shape descriptions are promising candidates for prediction.

Text-based description of images has been done via the concept of Specificity. Specificity has been applied to images in the form of photographs or scenes of objects [7]. We identify a research gap here. Could we translate the concept of Specificity from images in general, to other forms of shape? These forms could include fonts, a more specialised form of 2D shape, or potentially 3D shapes in general?

This requires a measure of Specificity to be determined for each form of shape. For this, consistency would be needed in the collected data, so it would need to be based on a common truth or common subjective decision agreed on by a population at large. This could be the geometry itself, or to include some aspect of human perception in the measure, it could be based on human language. For example, the consistency of word or sentence-level descriptions of images, as in previous work [7]. If using this approach, we would then need to ask how we could represent word meaning in a quantitative manner. Currently, these word meanings or *word senses* can be described via word co-occurrence frequencies represented as points or *word embeddings* in a vector space. Or otherwise, via a lexical database of words grouped together by their synonyms (as a graph structure), which are closer together if they have similar (and already known) word senses. The Image Specificity [7] work takes the latter approach. Could we represent the Specificity of images through an underlying word embedding?

As was done in the Image Specificity work, it may be possible to automatically compute Specificity scores for shapes, using a measure based around the approaches above. Logistic regression was used to learn parameters for positive and negative sentence classification, with respect to an image using artificial neural networks. Using the

measure, Specificity prediction could be attempted for shapes not yet encountered in a dataset. Prior to our work, there had not been any collection and measurement of word distributions associated with fonts, with a focus on Specificity.

Potentially, a single measure could be applied to different forms of shape/geometry, since the method of description would be separated from the shape data itself. But, the provision of individual measures for each form of shape could result in more accurate or consistent results, based on attributes unique to each of them.

3.3.3 Machine Learning

Machine learning aims to provide the following: given data, you wish to learn a model which represents it, but can generalise to new data, which could be later sampled from the same population (e.g. data class), as the original data.

Classification aims to predict the class of a new data sample from the same population – e.g. Which animal is in this picture? Or, will this company's stock price fall to a certain level, this week? *Regression* aims to predict a real-value for some question – e.g. How likely is it that this animal is a cat? Or, how much could this company's stock price change over the next week?

Discriminative models aim to answer these questions directly, whereas *generative* models aim to learn a joint distribution over inputs and outputs used to predict answers to these questions, allowing new, plausible data to be generated from that distribution.

We use discriminative machine learning methods, to provide us with measures of shape based on human perception, which can be applied to new shapes. This is done using 'ground-truth' data based around the concepts described at the end of the previous chapter (see 0, *Summary and Research Gaps*). Existing works which employ machine

learning include various classification and regression approaches. We describe in detail some specific sub-topics of machine learning that are more relevant to this thesis.

Metric Learning

This is a type of learning method that computes a metric to measure distances between pairs of items (e.g. between photos, or between points on a 3D mesh). Metric learning has been used to compute a similarity measure for fonts [58] and illustration style [59]. The latter approach combines many features/attributes of clip art together, to learn a metric for comparing clip art. An application was created to allow users to make a scene/mashup, of clip art. Search terms are entered in as text, and the closest matches are displayed. As the user places clip art on the screen, new clip art is listed in the search region, which matches the current style of clip art in the scene. Other works have attempted this for 3D shapes [195, 196, 197]. A 2013 survey on metric learning using feature vectors and structured data [60] details some existing metric-learning algorithms [198, 199].

Deep Learning

Deep learning involves multiple layer architectures of neural networks. These neural networks have many hidden layers (potentially up to 10s or 100s), and usually additionally designed properties which make them suited to different kinds of input – e.g. *convolutional neural networks* [200], *LSTM networks* [52] or *residual networks* [90, 201].

LSTM networks can be used for text classification [202, 203], sentiment analysis [204, 205, 206], machine-based language translation [207, 208, 209, 210] and image generation [53]. In cases where 10s to 100s of layers of required for modelling large scale datasets, residual networks can be used to retain information at specific layers if

required (for accuracy improvement). Outputs of certain layers can be skipped, as the network learns a non-trivial identity function between pairs of layers, while outputting its own transformation of the input.

Neural networks in general, have been used to classify 2D images [211] and 3D shapes [55, 212, 48, 47, 49] as well as saliency detection in individual 2D images [46, 70, 211]. Regarding 3D shapes, some classification approaches include: a 3D object classification algorithm, using convolutional, recurrent neural networks [212] and a high-level semantic feature extraction method, for 3D shapes, based on deep belief networks [56]. In this work, multiple views of a 3D shape are first encoded into a 'bag-of-visual-features', extracted via SIFT. Then higher-level shape features are generated from this, via a deep belief network. Deep Boltzmann machines have also been used for 3D model recognition [55] and generation [54].

Geometric Deep Learning

Geometric deep learning is a recent sub-field of machine learning and geometry processing: it is an "umbrella term for emerging techniques attempting to generalize (structured) deep neural models to non-Euclidean domains such as graphs [213] and manifolds". Approaches which learn on graphs aim to automatically learn to encode graph structure into low-dimensional embeddings.

The name of this topic was derived from a 2017 survey which provides examples of geometric deep learning problems and presents potential solutions, key difficulties, applications, and research directions to consider [214]. Since then, many works have begun to appear which move away from using more common 3D (voxel) and 2D (colour/depth image) based Euclidean data, for discriminative and generative models.

A related approach to shape correspondence has been developed using anisotropic (non-uniform scale) convolutional neural networks [215]. Other structures such as point clouds are beginning to be used for learning applications, [216, 217, 218, 219], in addition to B-splines [220] and initial approaches which use polygon meshes [221]. The "graph neural network model" was an early example of this kind of approach [222].

Some models use hierarchical voxel or 3D grid-like structures (octrees; DAGs) for classification [223], representation via autoencoder [224] and generation and reconstruction of 3D shapes [225, 226].

In 2017, a framework for learning dense correspondence between deformable 3D shapes, using learnt functional maps was been produced [227]. Instead of treating shape correspondence as a labelling problem, where each point/vertex of a query shape receives a label identifying that point in a reference domain, with the correspondence being determined by comparing the label predictions of two input shapes, the authors proposed a prediction model in the space of functional maps. These are linear operators that provide a compact representation of the correspondence mapping. They model the learning process via a residual network which takes dense descriptor representations defined on each of the two shapes, as input, and outputs a "soft" (probability) map between the two objects.

A method for approximate reconstruction of a 3D shape from a single image has been defined [228], in addition to a method for deformable shape completion, using graph convolutional autoencoders [229] and generation of 3D surfaces, using a single view of an image or a 3D point cloud, where the network represents a 3D surface as a set of parametric surface elements. It maps a set of squares to the surface of a target 3D shape

to be represented. Surfaces with non-disk topology can be modelled using this approach, which is similar to that of generating an atlas of a manifold [230].

Summary

Artificial neural networks of many kinds have been demonstrated to work for classification [212], regression and generation of shapes [54] and scenes. Other works have produced methods to learn metrics for the comparison of clip art and to measure style in fonts [58], illustrations [59] and 3D shapes [195, 196, 197].

Geometric deep learning approaches aim to learn on non-Euclidean geometric structures directly. These are generally more traditionally analysed shape representations: such as graphs [229, 213], point sets and polygon meshes [221], even in a deformable shape setting [230]. Applications include shape correspondence [227] and shape reconstruction from an image [228].

The focus of our work was not to incorporate sophisticated learning methods, but to have some method to determine whether a Schelling shape function or font Specificity function can be learned.

Can we predict Specificity for fonts through a machine learning approach? We could attempt this in different ways, selecting any of: {shape descriptor(s), image(s), sentence(s)}, mapping them to their respective Specificity score for each font. A fully-connected network or convolutional network could be employed. If collecting word or sentence level descriptions to compute Specificity, a word embedding/vector space representation of the words may be useful for training the network. Previous work has collected and used crowdsourced data to create a font similarity metric [58]. A method to predict Specificity in images has been developed, based on a logistic regression model which is trained on ground-truth pairs of sentences from humans in the form of

‘positive’ examples, where both sentences come from the same source image. Pairs which were ‘negative’, did not come from the same image. The parameters of this prediction model are used to generate Specificity predictions for new images [7].

Can we predict Schelling saliency for whole meshes? This leads to the question of which shape representation(s) to use for prediction (depth images, voxel grids, etc.). The structure of the prediction model may depend on the shape selection approach taken in the data collection phase. That could involve direct mappings to relative shape selections (1-of-n), or prediction based on individual shape selection frequencies across all participants, made within the Schelling context (where people select meshes with the aim of matching with what they expect others to choose).

Regarding 3D shapes, previous work has developed a regression model to predict which vertices of a mesh are most likely to be selected by people, structured as a mapping from a range of per-vertex geometric attributes to a per-vertex indicator function, representing vertex selections. Attributes included curvature, intrinsic symmetry and Shape Diameter Function values [6].

Convolutional neural networks (CNNs) are useful for predicting and analysing human-perceptual properties of shape. A CNN’s structure incorporates some concepts of the ITTI98 saliency model [73]. The concept of centre-surround differences is reflected in local convolutions across an input shape (or that of other layers later in the network). This also indicates a local contrast method of shape analysis. Convolutions are also translation invariant due to filter windows being applied locally across a shape. Detail is represented at multiple scales; visual abstraction is discovered through the flow of information between layers in the network. These visual structures or more complex objects comprised of more primitive curves and shapes can be seen as elements of

global contrast being searched for in a shape. The above points lend to convolutional neural networks acting as a combined local and global contrast approach. This makes it ideal for shape analysis, as previous works have shown. Additionally, through abstraction obtained via multiple layers, deep learning methods in general tend to have greater generalisation potential across various problems, given new data instances.

3.3.4 Crowdsourcing

Our primary methods of data collection are crowdsourcing-based, so we provide a summary of recent approaches and outcomes.

Use of Crowdsourcing for Data Analysis

Crowdsourcing is essentially large-scale data collection via users of the internet, for many purposes such as image annotation and description, user interface testing and other topics. Example providers include *Amazon* and *CrowdFlower* [231]. A lot of work has been done using crowdsourcing, within computer science. Some works include: the summarisation of image collections, according to user preferences [232]. With the help of workers on the *Amazon Mechanical Turk*, the authors obtained a large amount of "manually created visual summaries as well as information about criteria for image inclusion in the summary". From this, an automatic image selection system was produced, using *RankSVM* (a machine learning classifier which can rank its inputs) [198, 199], which jointly analyses "image content, context, popularity, visual aesthetic appeal as well as the sentiment derived from the comments posted on the images".

Other works have focused on modelling human preferences in visual summarization, by attempting to detect semantic concepts [233]; modelling the appeal of photos which portray people [234] and the discovery of 'non-obvious' attributes of social images [235]. In relation to the latter work, three pictures of dogs can imply dogs are friendly

and seem like good companions, or otherwise show dogs to be angry creatures. The same goes for a living room which may seem homely or traditional in one picture or instead be very minimalistic/modern, in another. These biases are not captured in just the definition of the subject of interest.

Crowdsourcing has also been used to discover ‘beautiful’ attributes for the analysis of aesthetic properties of images [236]. Focusing on image preference, the authors aimed to provide accurate and interpretable results, via the automatic learning and discovery of visual appearance from an approx. 250,000 image database, called AVA [237] (images were obtained from: *www.dpchallenge.com*) combined with aesthetics scores and "textual comments given by photography enthusiasts". Discriminative textual attributes were automatically discovered using user comments and preference scores. The learned visual attributes were applied to image classification and retrieval, as well as aesthetic quality prediction. The authors noted that images with high aesthetic score variance were often non-conventional - i.e. "edgy or subject to interpretation", whereas "images with a low variance tend to use conventional styles or depict conventional subject matter".

Using the AVA dataset, Lu et al. produced a system for rating aesthetics in pictures, using a convolutional network which takes two inputs [237, 238].The inputs consist of a global and local set of saliency cues, where each input has its own convolutional layers, leading to a merged single output as a saliency score, based on both inputs.

Many more works have been produced which relate to human computation [239] image/video annotation [240], user interface testing [241], and networking [242]. Some work has been done in crowdsourcing data for saliency. One work studied how humans select “Schelling” points or vertices on 3D polygon meshes, in a coordination game [6].

Another work, *TurkerGaze*, provides a set up for obtaining large-scale eye tracking data for saliency prediction, using eye-tracking games, where a saliency dataset was additionally built for natural images [243]. Other approaches to webcam-based eye-tracking have also been made [244, 245].

Other work has applied crowdsourcing to solve various problems in computer graphics. The idea of collecting data on how humans perform a task and then learning from this data, has been used to determine a similarity measure of style for 2D clip art [59], fonts [58] and 3D shapes [195, 196]. A method for learning visual similarity for product design has been developed using convolutional neural networks [178], which uses pairs of images containing products placed in real-scenes vs. products in their 'iconic' form. The resulting embedding is used for visual search.

Many evaluations of crowdsourcing have been performed over recent years, including a comparison of methodologies for subjectively assessing image aesthetic appeal [246]. The authors assessed four different scoring approaches, on their ability to measure aesthetic appeal in images. 24 people were asked to assess an image set, meant to uniformly represent a "wide range of aesthetic appeal". They suggested that the "Absolute Category Rating (ACR) 5-point scale provides the most consistent ratings across participants". This result reinforced previous work from 2012, on audio-visual subjective tests across six laboratories, from four countries [247]. In the 2012 study, *mean opinion scores* (MOS) were calculated as an average of participants' ratings, across all datasets. Pearson correlations were calculated from this data, against the number of participants and the range of MOS. These two distributions based on 24 participants, yielded narrow distributions/lower variance, at a correlation of 0.96 or greater. Due to this, 24 or more subjects were recommended for *absolute category*

rating (ACR) tests. In public environments, or where there is a narrower range of audio-visual quality, 35 subjects were required for the same Student's t-test sensitivity. A second important aspect mentioned is to remember how opinions differ among subjects. They affirmed that participants drawn from a sole source cannot fully replicate the behaviour of all people and suggest that improved methods are needed for eliminating non-performing subjects - i.e. those that do not understand the task, and those that simply do not perform it. They mentioned that current methods "assume opinions are homogenous". But, their work suggested that factors such as native language, lighting, monitor calibration, viewing distance, translation of ACR labels, or culture/country of origin matter very little, and are likely to be obscured by human factors. The authors however, did note that the "impact of language and culture on subjective scores would be an interesting topic, for further investigation". These studies did not involve viewing 3D shapes, which can have variable/user-defined viewpoints. This may affect the reliability of their guidelines in studies, which do.

In other work, data obtained via crowdsourcing has also been used to extract depth layers and image normals from a photo [248] and to convert low-quality drawings into high-quality ones [249]. Furthermore, human preference data has been collected in terms of semantic attributes (words) to describe body parts [185] and words describing body shape [187].

Crowdsourcing has been used for software engineering support [250] and behavioural sciences research [251]. Amazon Mechanical Turk (MTurk) has been empirically compared to other platforms such as: Crowdflower (CF) and Prolific Academic (ProA). The authors test for honesty and naivety in participants, finding that participants on CF and ProA were "more naïve and less dishonest compared to MTurk participants", and

that participants on CF provide the best response rate of the three platforms. But, CF participants failed more of the authors' attention-checking questions [252].

In 2018, a review of crowdsourcing and suggestions for future research was produced [253], in addition to an analysis of methodologies for conducting interactive experiments online [254].

Summary

There was a need to collect data in addition to shapes since we aimed to understand them from a perspective of human perception. Large scale data collection, or ground-truth data is needed to train neural networks. Due to this, crowdsourcing was a crucial element of our data collection process. Regarding the data collection surveys themselves, it is important to design the questions so that useful data can be gathered. Validation of inputs, qualifications and pre-tests are all potential avenues which can be followed, to prevent results being gamed.

Although data has been collected on fonts for learning a metric based on high-level attributes/adjectives (e.g. legible, formal, friendly) associated with fonts [58], the notion of font similarity is based on triplet-wise comparisons, where participants are asked to decide whether a font B is more similar to a font A, than the remaining font, C. Can meaningful word-level descriptions be collected on fonts? It could be possible for a measure of font Specificity to be produced based on the consistency of these descriptions.

Data has been collected on vertices selected on 3D meshes, given the Schelling concept of attempting to match with others' selections without communication [6]. In contrast, can meaningful data be collected on whole shape selections out of a collection of meshes, given the same Schelling concept?

3.4 Conclusion

There are existing works on *co-saliency* [105] and *group saliency* [112] in images, but not 3D shapes. Therefore, no prior works have created a group-level saliency measure for 3D shapes.

Data has only been collected on Schelling points on 3D meshes, where the points lie on a single mesh. Whole meshes have not been studied relative to other meshes, under this context (where each mesh can be treated as a candidate Schelling point). Part-wise comparisons of shape, under similar constraints, have not been made. Data hasn't been collected on Schelling points in images, audio or video.

Text-based description of images has been done via the concept of Specificity. Specificity has only been applied to photographic images [7], therefore not fonts or 3D shapes. Specificity has been formulated using a lexical database of words / word senses, which is in some way 'supervised', as the data is manually constructed to some degree. But it has not been formulated through an unsupervised approach to representation of word semantics, such as word embeddings.

The previous two points also imply that a discriminative model of Schelling meshes or Specificity in fonts (or 3D shapes) has not yet been developed. Or more generally, given some measure of these concepts, a model to predict them for 2D shapes and/or 3D shapes.

Based on these potential research gaps to follow, the following discussion sets the context for the approaches taken in this thesis.

3.4.1 Potential Approaches

Schelling Meshes

Free-viewing eye fixations have been shown to agree with explicit shape selections in previous work, so we use this as a basis for collecting explicit shape selections as Schelling saliency data, instead of measuring eye fixation trails. This enables us to collect data via a crowdsourcing approach, which would not currently be possible otherwise, as personal eye-trackers are not widespread, even if accurate enough for this purpose.

Given that existing works have shown that humans can perceive the depth of 3D objects in a single image and show that the shape of a virtual object can influence the perception of its material reflectance, it is possible to collect data using rendered images of shapes, with consistent lighting conditions (as animations in the case of 3D shapes). Although it has been shown that relationships between shapes and their colours and materials can be perceived, a consistent colour can be used across all shapes, to remove potential bias that could occur due to contrast in the selected colours of shapes.

Font Specificity

To understand fonts from a derived representation of their geometry, we can collect textual descriptions, as previous works have done before. We then need to disambiguate between word meanings. Currently, these word meanings or *word senses* can be described via word co-occurrence frequencies represented as points or *word embeddings* in a vector space. Or otherwise, via a lexical database of words grouped together by their synonyms (as a graph structure), which are closer together if they have similar (and already known) word senses. To define Specificity in images, previous work has taken the latter approach, but not the former.

General Points

It is possible to use depth images and colour images for 2D shape representation. Depth images of multiple-views can be obtained in the 3D case, in addition to fixed-size or hierarchical representations of voxel grids. Evidence suggests that latter representation would be sufficient for good prediction results.

Previous work considers the various notions that exist in visual saliency, which include: local contrast, global contrast, centre-surround differences at multiple scales and global rarity. These should inform the selection of any methods for prediction of Schelling saliency data and Specificity in fonts. We see that hierarchical prediction methods such as convolutional neural networks are predominately used for saliency detection in images and understanding of 3D shapes.

Existing approaches to regression based on 3D shapes and 2D images use hand-crafted shape descriptors or more recently, deep learning approaches. Determining which approach to use for Schelling saliency or font Specificity prediction, requires knowledge of which methods have been useful for modelling aspects of human perception, or have otherwise been related to the topic. For example, curvature has been related to aesthetics [4]. Some measures of curvature and the Shape Diameter Function have been informative in explaining points selected on 3D meshes, in the Schelling context. Other potential candidates consist of: the distribution of per-vertex normals, D2 Distribution values. In the 2D case, the SIFT, SURF and FREAK descriptors to can be applied to individual images. Some descriptors can be applied to a 2D or 3D shape representation. The Histogram of Oriented Gradients (HoG) descriptor or 3D Sobel filter can be computed over a voxelised form of a 3D shape, whereas the 2D Sobel filter and 2D HoG descriptor can be applied to an image.

Overall, the literature suggests potential avenues to approach the creation of group-level saliency methods for 2D and 3D shapes, even if existing works are uncommon or loosely related. We used this information to form the basis of the studies in the following chapters, which begins with our first Schelling meshes study.

4 Schelling Meshes: ‘4-Choose-1’ Approach

The notion of Schelling points on 3D meshes has been studied [6], where Schelling points are vertices on the surfaces of meshes that people expect will be selected by others. See Figure 1.2 [6] for a visual description. In our approach, the domain of saliency changes: instead of selecting among points on a mesh, people select among multiple meshes.

4.1 Introduction

Schelling Points are choices made by people when they aim to match with what they expect others to choose, with no prior communication between them. Although an abstract concept, people have studied points on 3D meshes selected by people due to their salience, in this coordination game setting. We extended the notion of Schelling points on meshes, from points on 3D shapes, to the shapes themselves.

For this study, our aim was to determine the degree of agreement between people when they are individually asked to select the most salient shape of out of a collection. We collected Schelling-based data for meshes by asking people to choose one of four shapes from a class of shapes (e.g. tables, lamps). They were asked to select shapes that they believed others would also select, given no communication beforehand.

This agreement was reflected in the frequency of shape selections in each context, or ‘Schelling frequencies’. We studied these shape selections and their distributions to determine what makes a shape Schelling salient. As shapes were represented as polygon

meshes, we named the most salient of these in this sense, Schelling meshes. This is a *data-driven* approach to understanding geometry, where we collect data based on a human interpretation of it and use this data to better understand it.

We show that the notion of Schelling salient meshes can be learned and used for Schelling score prediction via a voxel-based convolutional neural network. Results are shown for several classes of 3D shapes. We view the concept of Schelling meshes as another tool for 3D shape analysis, demonstrating that it is useful for the applications of Schelling-based visualisation, clustering, and search.

4.2 Hypotheses

We hypothesized that Schelling meshes are salient in the sense that they stand out in some way, when compared to other meshes. We broke this idea down into multiple hypotheses:

1. A more natural shape is less Schelling frequent, as there is not much special (surface variation) that makes it stand out against other shapes.
2. A stranger shape is more Schelling frequent, as the strangeness will make it stand out against other shapes.
3. A shape which stands out from others or is considered unique, is more Schelling frequent due to global rarity of elements on the shape's surface (related to naturalness and strangeness).
4. A more visually appealing shape is more Schelling frequent, as the appeal/aesthetics of the shape will attract people to select it.
5. A shape may be perceived as more memorable relative to others, as its Schelling frequency increases.

4.3 Methodology

4.3.1 Data Selection and Generation

For shape data, we collected 145 3D shapes mainly from *ShapeNet* [8]. These included man-made objects and a variety of *abstract* shapes (chain-like objects, primitive shapes and others). Specifically, we collected 30 *tables*, 44 *lamps*, 45 *chairs* and 26 abstract shapes. The ‘abstract’ shapes were collected for additional variation, with the aim of increased generality of any collected data, and the results of its analysis.

After obtaining these shapes, we needed to determine the best way to present them to participants. We decided to do so using animated .gif images, since they provided good resolution at reasonable file-sizes. To display the shapes on a 2D screen, we generated a continuously rotating view of each shape, over a period of 3 seconds. The view allowed one to see the top and bottom parts of each shape. Animation frames were produced from renderings of each shape, shaded with consistent colour and lighting, where the shape’s viewpoint was changed via incremental rotations around the horizontal or y-axis, to a full 360°. After a short pause, the rotation would repeat. This happened in a recursive manner, allowing participants to focus longer on a shape if they felt they missed some aspect of the shape after a first rotation. Since direct interaction was not required to initiate any rotation, this minimised the interaction necessary to view each shape. We chose this representation over multiple images, to make visualising each shape easier.

We could have displayed shapes as individual videos, enabling participants to pause a shape’s rotation and adjust their effective viewpoint of the shape (at least along a fixed axis). But we believed it unlikely that the majority of participants would interact with all or even many of the shape videos. This would likely result in many people effectively

viewing static images to make their selections, even when the shapes in question are not symmetric around the video's axis of rotation. This would make the amount of focus on each shape or the amount of each shape viewed, biased to the level on interest the participant had in the survey, introducing unnecessary bias to shape selections.

We decided to show four shapes to participants for each question, where the participant was asked to choose one of these four shapes. We did not take a smaller or larger number of shapes for a few reasons. Showing two shapes is not realistic given a person's field of view, where 10+ objects can be seen at once. Then, why not display 10 or more shapes at once? If displaying shapes on a computer screen, it is difficult to show 5 or more shapes horizontally, without loss of resolution or detail, at a glance of the shape.

Scrolling is a partial solution, but with too many shapes on screen this requires some memorisation of previous shapes. Displaying three shapes is not much different from two, but you can internally rank the shapes shown to you, via triplets (e.g. A > C > B).

Three shapes can force tie-breaking, however. See Figure 4.1 for a pictorial description. With four shapes, you can still allow for dual 'ties' or pair of shapes that are the same or similar, where the two pairs look different.

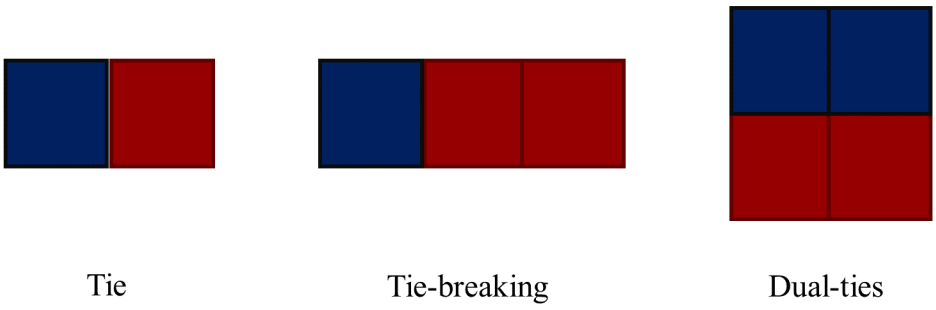


Figure 4.1 - Shows some possible outcomes given 2, 3 or 4 options to choose from, when selecting a Schelling Point.

We also felt that 5+ shapes per question might overwhelm participants, making it difficult for them to focus on the selection task. So, we refer to the current study as our ‘4-choose-1’ setup. Since the item chosen to be *Schelling* was done so relative to the other items, we decided not to compute a Schelling function for each shape, and did not collect Schelling data for each shape, individually.

Although picking four random shapes to generate each question is possible, this may not lead to useful data. Taking colour as an example, if we show humans four squares each with a different colour (e.g. red, green, blue, yellow), it is difficult to choose one of them, and different people may choose different responses. A similar outcome has occurred in previous work, where people have tried to study triplets of shapes. In a study of perception of style in shapes, participants were unable to provide a clear similarity ranking between shapes in the triplets, for over 60% of cases [196].

Because of this, some form of subjective bias was introduced into the process of selecting each triplet for presentation to participants. This ensured that there was always a possible reason for an answer, or that participants could always discriminate between the shapes in some way. For example, the shapes might be members of the same class (e.g. tables) or occupy a shared scene or arrangement [196]. We thereby took the idea of generating a more careful set of questions to produce questions which were more likely to lead to useful data. To sample shapes in a similar manner, we determined categories either according to simple geometric 'primitives', or otherwise, previously collected aesthetic ratings (see Table 4.1 for details).

Based on these categories, we separated the shapes in each class into some high-level groups and then generated survey questions by picking three shapes from one group and one shape from another group. The groups, and shapes in each group were randomly

chosen. Each shape was placed into one or more groups (see Table 4.1). It may seem that each question has a "correct" response, but this is typically not obvious, due to the overlapping of groups and variation of the shapes.

Shape Class	No. of Shapes	High-Level Group/Criterion	Number of possible permutations
Abstracts	30	Sharp (13), Smooth (14), Chain (3), Spikes (4), Shell (3), Sharp and Smooth (7)	657720
Chairs	45	Based on Likert score ratings of shape aesthetics made via Amazon Mechanical Turk, from 15 participants. Low (15), Medium (15), High (15)	3575880
Lamps	44	Unusual (31), Plain (13), Cylindrical Top (33), Rectangular Top (11), Non-Circular Base (9)	3258024
Tables	30	Circular Top (24), Non-Circular Top (6), Multiple Legs (9), Flat Base (12), Round Base (13), Tall (4), Non-Curved (5), Octagonal (1), Pointed Top (1)	657720

Table 4.1 – High-level groups used to sample shapes for presentation to participants.

Before we began to think about high-level groups however, we originally held a small test survey of 5 people, asking them to select one of 4 tables per question that were randomly sampled (where each shape of the 4 was unique), providing them with 10 questions each in total.

Participants consisted of students at Lancaster University, studying different subjects at post-graduate level (2 PhD students, 3 Masters' students). When asked how they made their choices, 4 people stated that the selected shapes were usually "different" or "unique" compared to the remaining 3 shapes, with 2 people mentioning that they sometimes selected "familiar" shapes or "what they were used to in their daily life". People also stated that it was sometimes difficult to make a decision, and sometimes selected randomly. Overall, we didn't find any clear pattern among the most frequently selected shapes. Although at a very small scale, this expressed a similar issue to that of participant uncertainty in previous work [196], when fully randomised shapes were

presented to participants. The number of survey participants was clearly not a valid sample size for a study, but it was an additional small reason behind why we decided to use high-level groups in our actual study. However, as you will see later on, we also collected Schelling saliency data without high-level groups, for comparison.

4.3.2 Data Collection

We created survey webpages using the Amazon Mechanical Turk platform, allowing us to collect Schelling saliency data using a crowdsourcing approach. Participants were first given written instructions: “*For each question, your task is to choose one of four 3D shapes. You should choose the shape that you think will be chosen by other participants. Your only goal is to choose a shape such that it will most likely match with what others will choose.*”

Each HIT (set of questions on Mechanical Turk) had 30 questions where each question gave us one data sample. A participant typically took about 5-10 seconds for each question. We paid $0.10 for each HIT. Separate surveys were held for each shape class; there was no overlap between them (e.g. only tables or only abstract shapes). For each shape class, example questions that we used, are shown in Figure 4.2. Before participants could work on any HITs, they were required to pass a qualification test consisting of 10 control questions and answer at least 8 of them “correctly”. The control questions were designed to test if a user was not paying attention at all. Each control question had an obvious and “correct” answer, as we showed three shapes that were the same, with one different shape. After passing the qualification test, participants could take as many HITs as they liked, but we limited this to about 20 per person, to not bias the complete set of data to a few people. Within each HIT, there were also 5 control

questions. A participant was required to answer at least 3 of these "correctly" for their results to be included in our data.

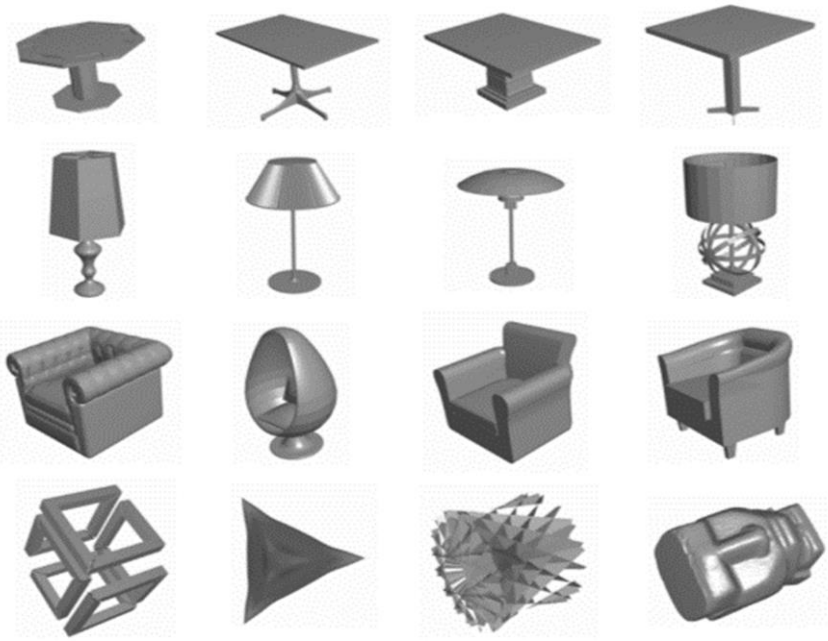


Figure 4.2 – Four examples of questions used to collect Schelling saliency data, one for each shape class (tables, lamps, chairs and abstract shapes). Within each survey, shapes displayed were only sampled from a single class.

Table 4.2 shows the amount of data samples that we collected for each shape class. These were used as training data for Schelling saliency prediction. The data for each class was collected separately, as the four shapes in each question were sampled from the same class. Each data sample, x_i, has the form $(S_A, S_B, S_C, S_D,\ s_A, s_B, s_C, s_D)$ where M represents a 3D shape and s represents the corresponding Schelling score. They each contained four shapes and scores indexed by A, B, C, D. As participants always chose one of the four shapes in each question, only one element of (s_A, s_B, s_C, s_D) was 1 and the others were 0. For the purposes of Schelling score prediction, we wanted to ensure that each permutation of four shapes within a sample was unique (as k-fold cross validation would be employed), but we neglected to enforce this at the beginning of the

data collection process. So, instead we filtered for duplicate permutations after the data was collected. This was done by firstly locating each set of data samples with duplicate permutations, then randomly selecting a sample from each set, and finally substituting each set with the sample selected from it. By doing so, we removed the other samples in each set from the collected data. Therefore, the number of participant selections used in the study was not necessarily a multiple of 30 (the number of questions, and therefore number of shape selections, per survey). We report *Minimum Number of Participants* in Table 4.2, as a lower bound on the number of participants that participated in the study.

Given $rem_{selections} = (\text{Number of participant selections } mod\ 30)$, this is calculated with integer division as: $\frac{\text{Number of participant selections}}{30}$, if $rem_{selections} = 0$ or $\frac{\text{Number of participant selections}}{30} + 1$, otherwise.

Shape Class	Participant Selections	Minimum Number of Participants	Minimum Total Cost	% of permutations of 4 shapes covered by sample size (number of Participant Selections)
Tables	3005	101	$10.10	0.46%
Lamps	4100	137	$13.70	0.11%
Chairs	3800	127	$12.70	0.12%
Abstracts	3800	127	$12.70	0.58%
Total	14705	492	$49.2	0.18%

Table 4.2 – Summary of the collected Schelling selection data, based on high-level groups.

Participant Selections

Figure 4.3 shows some examples of selections made by participants from the Schelling data collection. The selected shape was typically different from the other three and stood out in some way. For example: the selected table is shorter, and the selected lamp is

more planar. After the data collection, we were able to communicate electronically with some participants to gather their comments. One participant mentioned that the shape of one of the chairs was not suitable. Another mentioned that one chair did not look structurally sound. Some participants considered how natural the shapes looked, to make their decisions. These examples show that the participants sometimes thought about the structural and/or functional aspects of a shape, even though they were not instructed to.

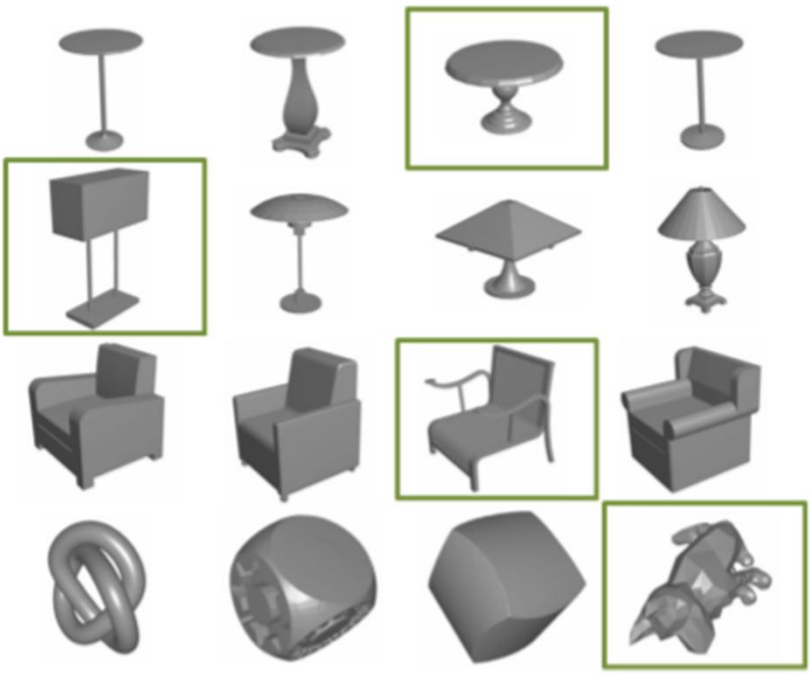


Figure 4.3 - Four examples of "Schelling" questions (one in each row for the tables, lamps, chairs, and abstract shapes) with the participant's selection highlighted.

4.4 Analysis

4.4.1 Validation of Data Consistency

We aimed to test if the collected Schelling saliency data was consistent across different people. We firstly assumed a null hypothesis that there would be no difference in distribution shape between two groups of randomly sampled shape selections from different participants (without replacement, and which were equally sized). To test this,

we gave the same question to different people, to investigate whether their answers would be consistent. We collected the following data: *4 shape classes* × *30 questions per category* × *30 participants*.

There were 120 total questions, each answered by 30 people. We limited each participant to only one HIT, to ensure that we obtained data from different people. For each question, we separated the data into two groups of 15 participants and recorded the distributions of the (A, B, C, D) responses in the two groups. As the values in the distributions were relatively low (some less than five), we compared the distributions by performing a Fisher's exact test. For all the questions that we tested, the test gave p-values >= 0.05, so we could say that the two groups had the same distributions. This provided evidence that there was consistency in the Schelling saliency data.

4.4.2 Schelling Frequencies

The Schelling concept is a relative concept. To capture this, we computed a Schelling frequency for each shape, to give an indication of how likely it will be selected in a Schelling sense. For each shape, we took all data samples for its class, and computed the total number of samples where it was selected, divided by the total number of samples where it appeared (at least once), to compute its *Schelling frequency*. We produced *Schelling frequencies* given a data sample d_i, where M is the number of shapes in the class and s_j is a score for a shape, S_j potentially in d_i:

$$j \in \{1, \dots, M\}$$

$$member(S_j) = \begin{cases} 1 & \text{if } S_j \in d_i \\ 0 & \text{if } S_j \notin d_i \end{cases}, \qquad s_j = 0 \text{ if } S_j \notin P_i$$

$$number\ of\ selections\ of\ S_j = S_{j_{total}} = \sum_i s_j \times member(S_j)$$

$$number\ of\ occurrences\ of\ S_j = S_{j_{occ}} = \sum_i member(S_j)$$

$$s_{f_j} = \frac{S_{j_{total}}}{S_{j_{occ}}}$$

Equation 4.1 – Schelling frequency derivation (4-choose-1 approach).

The mean Schelling frequencies were 0.241 for tables, 0.253 for lamps, 0.251 for chairs, and 0.244 for the abstract shapes. If we randomly sample a shape out of four, this acts as sampling from a uniform distribution of 4 shapes. As each subsequent sample is independent from previous ones, the expected probability of selecting a given shape each time is 0.25. Due to this, we expected a random predictor to be approximately 25% accurate.

Figure 4.4 to Figure 4.6 show Schelling frequency plots for each shape class, with frequency values represented by a picture of each shape. The more common tables are near the middle of the plot with some similar tables clustered together. For the chairs, the more creative and/or strange looking shapes tend to be on the right of the plot, while the more common shapes tend to be on the left. For the abstract shapes, a few shapes which have real-world meaning (human head, dog, and trophy) have the largest Schelling frequencies. Chain-like objects have mid to high-range Schelling frequencies. The shapes are partially clustered into two groups – primitive shapes vs. personified shapes.

For the lamps, those that were more planar, had a thin lamp pole, and/or had a strange looking base, tended to have the largest Schelling frequencies. The lamps that were more common tended to be near the middle (at around 0.25) of the plot. For the tables, the two of them that looked more special (a pointy top and an octagonal top), had the largest Schelling frequencies.

We could infer some visual patterns from the visualisations, but we wanted to determine whether filtering the shapes shown to participants (via per-class high-level groups), was introducing bias into our results (e.g. by introducing confounding factors or introducing a preferred result). We collected additional shape selections where no high-level groups were used to select shape combinations for participants to see and produced Schelling frequencies from them. See Table 4.3 for a summary of the collected data. Like in the high-level groups case, we ensured no data samples had duplicate shape permutations.

As a qualitative visual check, we created additional Schelling frequency plots as above, using this new data. Shapes per question were randomly sampled from each class. These are shown in Figure 4.7 and Figure 4.8. The mean Schelling frequency for the abstract shapes, chairs and lamps was 0.25. The mean for the tables was 0.249.

Shape Class	**Participant Selections**	**Minimum Number of Participants**	**Minimum Total Cost**	**% of permutations of 4 shapes covered by sample size (number of Participant Selections)**
Tables	2160	72	$7.20	0.33%
Lamps	3142	105	$10.50	0.09%
Chairs	2151	72	$7.20	0.07%
Abstracts	4100	137	$13.70	0.62%
Total	11553	386	$38.60	0.14%

Table 4.3 – Summary of the collected Schelling selection data (without high-level groups).

For the chairs, it is difficult to see any clear pattern as to whether certain geometries are more visual appealing, unique, prominent etc. This is unlike the case where high-level groups are in place, and a clear pattern of geometric variation with Schelling frequency, is visible. There is a similar story with the table shapes, but we can see some smaller trends when high-level groups are not in use. Tables with rectangular tops are placed at both the extreme low and nearly highest ends of Schelling frequencies, rather than being distributed around the mid-point of the plot. But tables with a circular base and/or top

tend to be placed near the centre of the Schelling frequency plot, maybe indicating that simpler shapes are likely to have middling or mean Schelling frequencies. The previous two points result in a simple form of clustering shown in the plot.

One small pattern is that the tables with highest Schelling frequency tended to have more complex structures to their bases, in addition to more asymmetric tops (not simply circular or rectangular alone). A similar pattern exists for the table Schelling frequencies based around high-level groups, where the complexity of the shape is the deciding factor.

Looking at the lamps, shapes with lower Schelling frequencies are more varied in structure with respect to one another as compared to other Schelling frequency intervals. But it is not necessarily the case that more unusual lamps have higher Schelling frequencies, potentially indicating that visual appeal is the distinguishing factor. This is different for the lamp Schelling frequencies based around high-level groups, where more complex or unusual lamps had the highest Schelling frequencies.

For the abstract shapes, chain-like objects had the highest Schelling frequencies, not differing extremely from the result obtained via high-level groups. Some sharp or spiky objects (those with extreme curvature) had medium to high Schelling frequencies. This is a different result to the case based on high-level groups, where sharp objects had varied Schelling frequencies in the low to medium range. The shape clustering obtained via high-level groups broke down also, as personified shapes (abstract shapes in the 'Head or Body' category) span the entire Schelling frequency spectrum, rather than only occupying the high-end when high-level groups are employed. The dog-like statue may have been selected highly due to its visual appeal relative to the other shapes in its category.

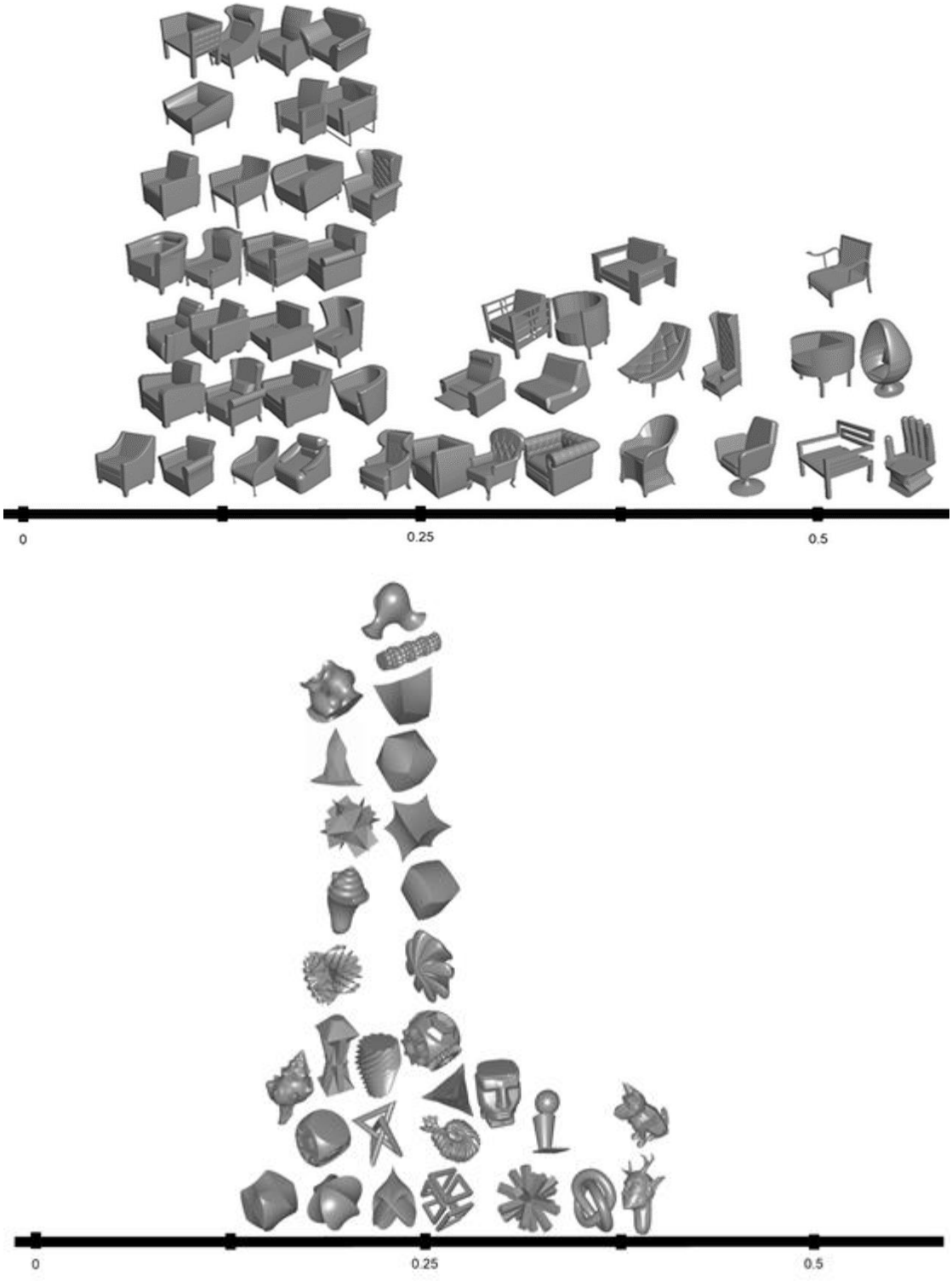


Figure 4.4 – Plots of Schelling frequencies for the chairs and abstract shapes, based on high level groups, indicating how likely each shape will be selected in a Schelling sense.

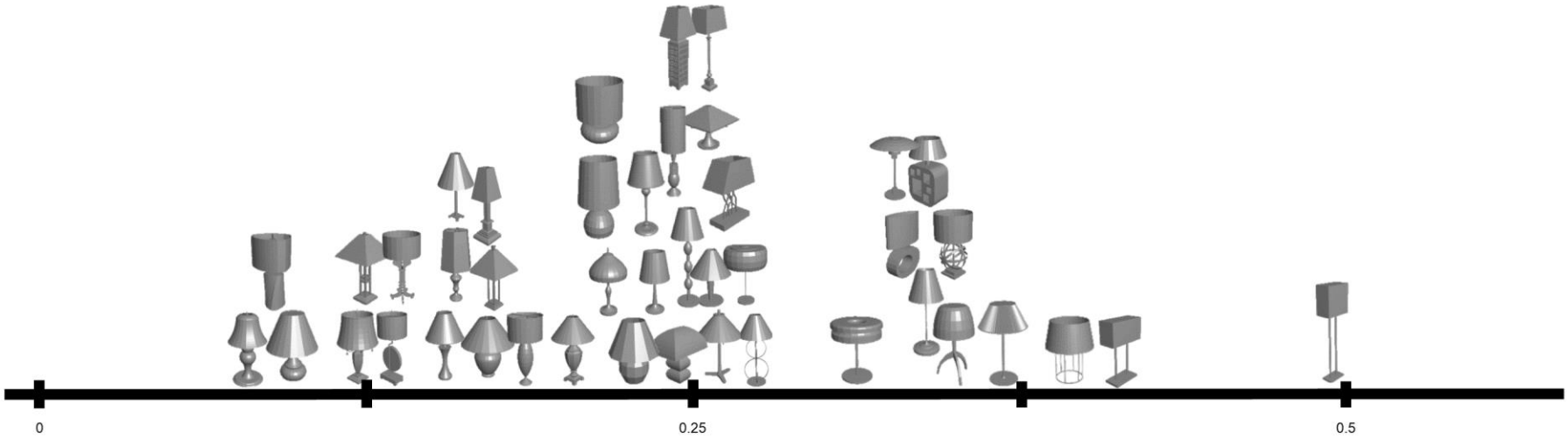


Figure 4.5 – Plots of Schelling frequencies for the lamp shapes, based on high level groups, indicating how likely each shape will be selected in a Schelling sense.

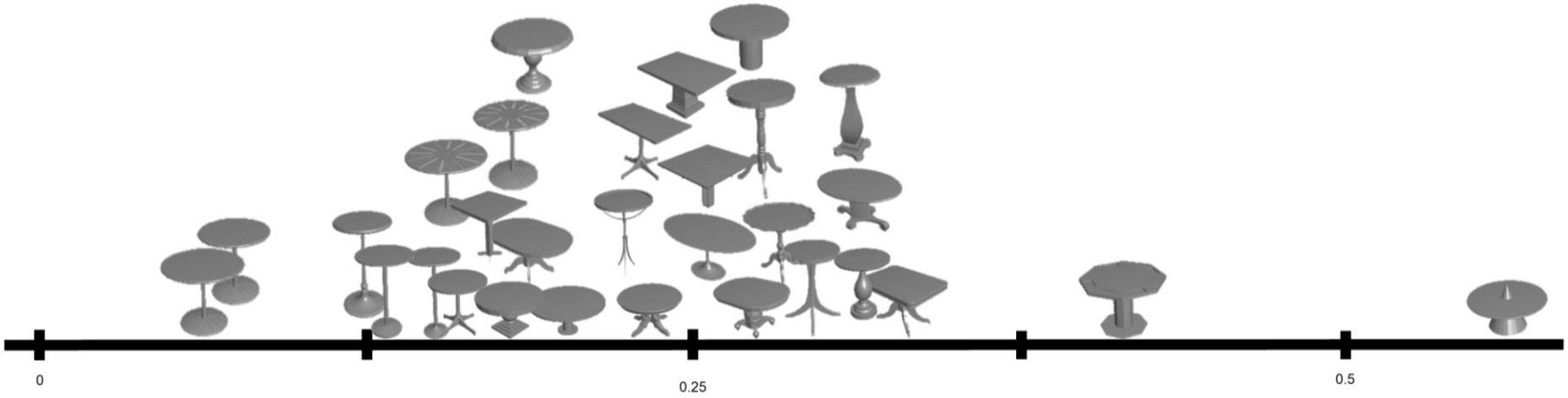


Figure 4.6 – Plots of Schelling frequencies for the table shapes, based on high level groups, indicating how likely each shape will be selected in a Schelling sense.

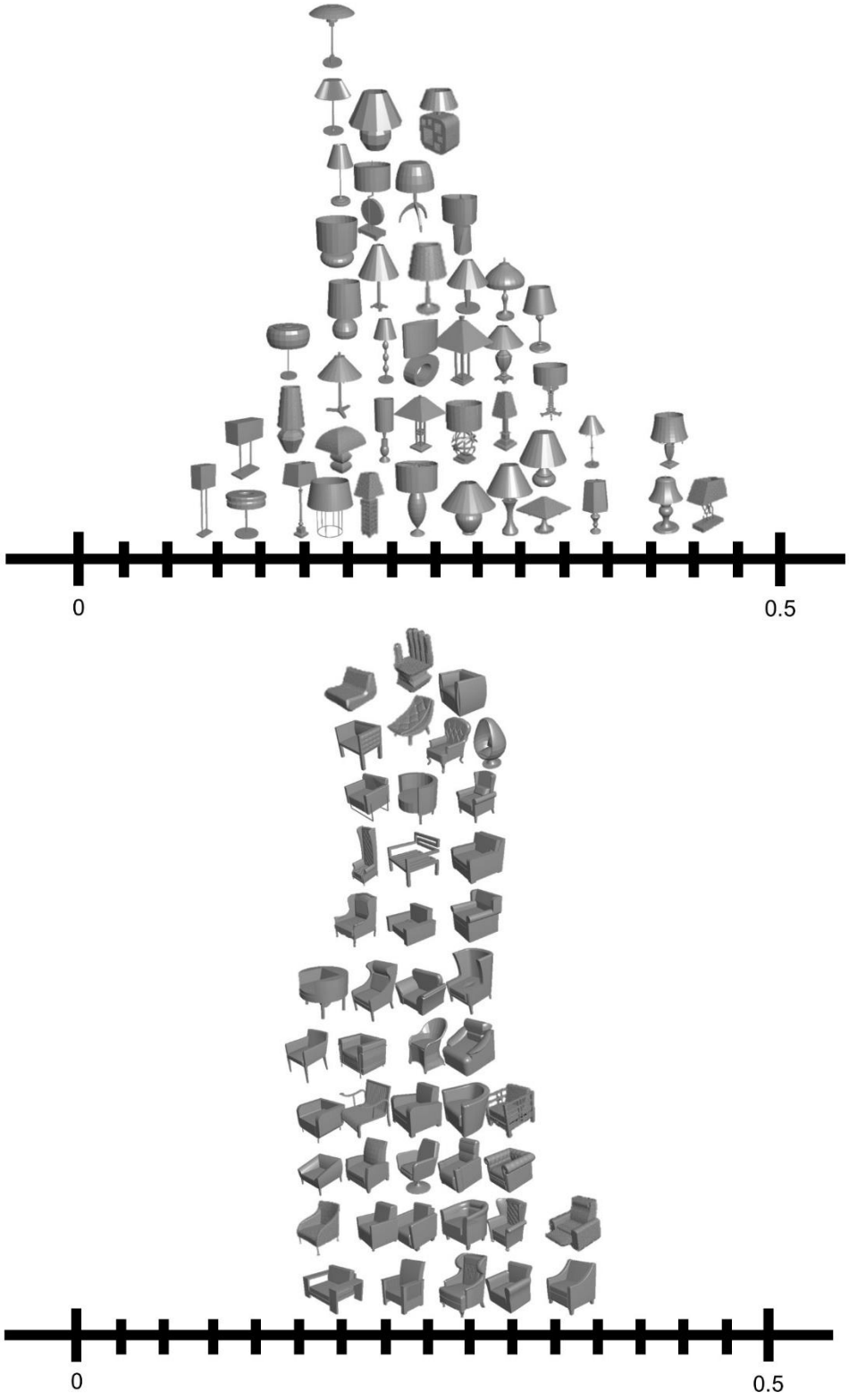


Figure 4.7 – Plots of Schelling frequencies for chair and lamp shapes, indicating how likely each shape will be selected in a Schelling sense (without high-level groups).

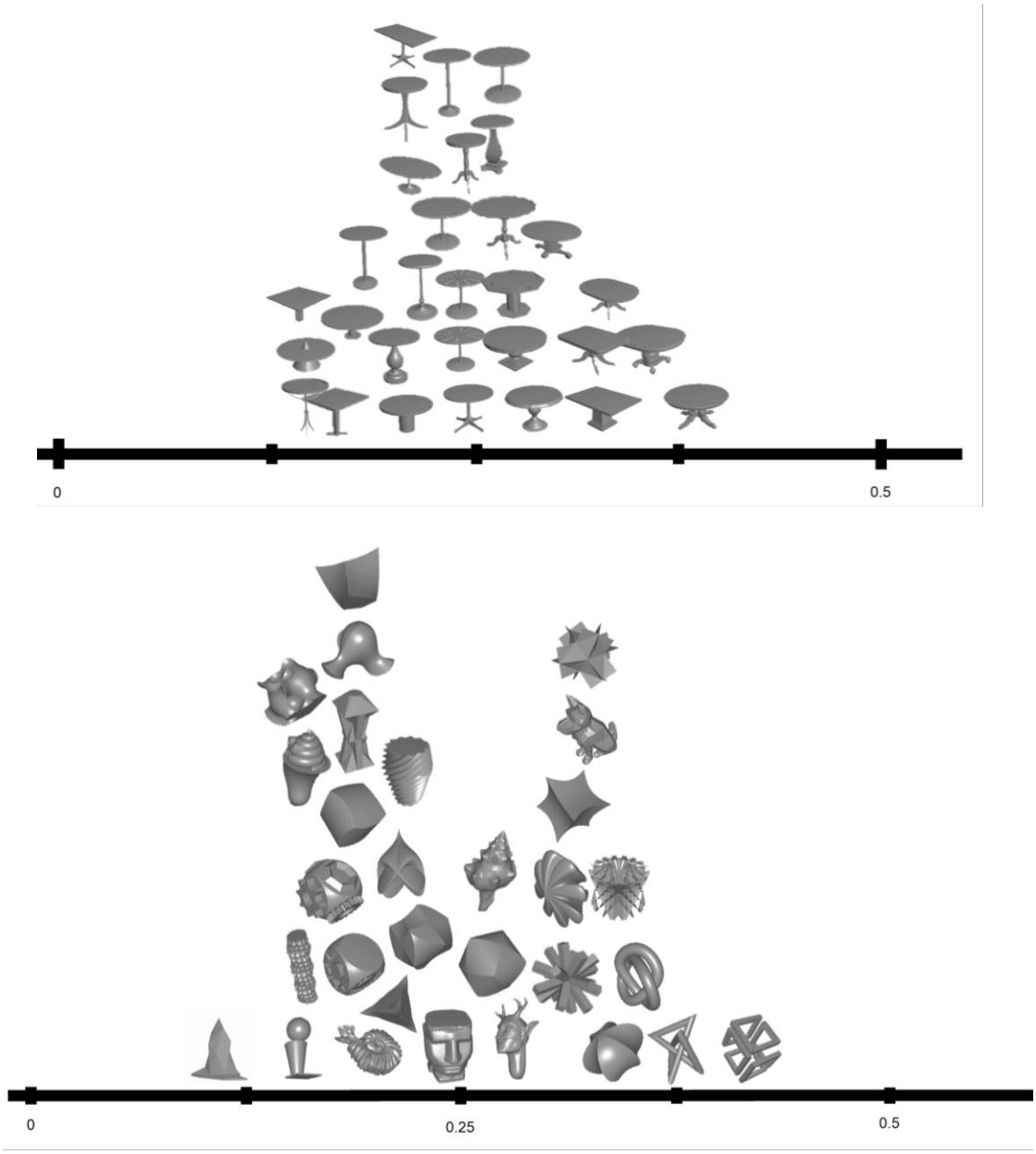


Figure 4.8 – Plots of Schelling frequencies for table and abstract shapes, indicating how likely each shape will be selected in a Schelling sense (without high-level groups).

The overall trend with the Schelling frequency plots based on high-level groups is that of prominence or unusual structure in shapes as Schelling frequency increases, which directly stems from the definition of saliency ("the quality or fact of being more prominent in a person's awareness or in his memory of past experience") [71]. This does not generally apply in the case of Schelling frequencies obtained without them (apart from the abstract class of shapes, in some ways).

4.4.3 What Makes a 3D Shape Schelling Salient?

To understand and characterize what makes a shape Schelling salient, we attempted to discover correlations between some human-understandable, subjective terms and Schelling frequency. These terms consisted of: "*naturalness"*, *"strangeness*", "*memorability*", "*uniqueness*", "*visual appeal*" and a notion of "*standing out*". We compared each term's association with a shape (average Likert score among participants) and its Schelling frequency.

We collected data for these criteria as a separate set of HITs on *Amazon Mechanical Turk*. Each HIT had 30-45 questions. Each question showed one shape only and asked the participant to give a score from 1 to 5 on how natural, strange, or visually appealing they thought the shape was. Each shape was given a score, by 15 participants. We paid $0.10 per HIT. We collected separate Likert data according to these terms and attempted to correlate the average scores in each case, with their corresponding Schelling frequencies, for each shape.

Table 4.4 shows the result of the correlations between these subjective terms and Schelling frequencies based on high-level groups. Values are Pearson correlation coefficients. Bold values are significant ($p < 0.05$).

For the tables and chairs, there was a negative correlation between *naturalness* and Schelling frequency, and a positive correlation between *strangeness*, *uniqueness* and *standing out*, and Schelling frequency (especially in the case of the chairs).

Sample Correlation Coefficient (Corr. Coef.)	**Schelling Frequency (high-level groups)**			
	Tables	**Lamps**	**Chairs**	**Abstracts**
Naturalness	**-0.5653**	-0.083	**-0.6506**	0.1553
Strangeness	**0.5307**	0.2071	**0.8211**	-0.3434
Visual Appeal	0.1894	**-0.3741**	-0.017	0.1002
Memorability	0.3175	0.1327	**0.4872**	**0.6455**
Standing Out	**0.4711**	-0.0749	**0.789**	0.1286
Uniqueness	**0.5304**	-0.1345	**0.616**	-0.0344

Table 4.4 - Correlations between some human understandable terms (naturalness, strangeness, and visual appeal) and Schelling frequencies (based on high-level groups). Significant correlations ($p < 0.05$) are in bold.

For the abstract shapes and chairs, perception of memorability correlated positively with Schelling frequency. Our hypotheses were correct for these cases. The abstract shapes only correlated with perception of *memorability*. They were a special class of shape which did not have a single function to them, potentially indicating that participants imagined and searched for valid concepts or affordances, imposing their own functions or categorisations to the shapes. It may be that the since the shapes had little inherent meaning with respect to one another, the shapes or concepts on display (primitive shapes, chain-like objects or functional objects) were considered to be equally unique, natural or visually appealing compared to one another.

This suggests that the perceived memorability of a shape may not be caused by a shape's surface variation or aesthetics, relative to other shapes. It may instead be based on a shape's alignment with common concepts/meanings/abstractions already learnt by the population being surveyed, the extent of a shape's ability to invoke previous scenarios one has been in, or memories one has experienced. The chairs also correlated positively

with a perception of memorability. We believe that the restricted amount of geometric variation within the lamps and tables, respectively, caused this pattern to not appear within these classes. Shapes within these classes (specifically the datasets being analysed) seemed to be designed around function rather than form, for ergonomic reasons. Collecting additional shapes from these classes, which have more geometric variation could help to verify this association with perceived memorability.

For *visual appeal*, there was no significant correlation with Schelling frequency, across all shape groups except for the lamps, where there was a negative correlation. Our hypothesis for visual appeal was not correct according to these results, as a visually appealing shape can possibly look common or stand out against others depending on the context of the other shapes.

Correlation values were not very high apart from the cases of strangeness and the notion of 'standing out' for the chairs shape class. Naturalness and strangeness are symmetric, and strangeness is related to uniqueness or a notion of standing out, so we can treat these as similar concepts. But memorability was not always associated with these concepts, even though it can be associated with Schelling frequency. Additionally, visual appeal was negatively correlated with the lamp Schelling frequencies, but not those of the other classes. This suggests that a single criterion is likely not enough for understanding and predicting Schelling frequency.

When not using high-level groups to collect data and produce Schelling frequencies, we obtained less correlations, but they more frequently tended positively towards visual appeal (tables, lamps and abstract shapes), indicating that this was the most prominent factor describing Schelling frequencies (see Table 4.5). Perception of memorability was still correlated with the abstract shapes, however. This was in addition to naturalness

for the tables and chairs. So, it is not necessarily the case that visual appeal explains Schelling frequencies completely.

But, since many trends vanish if not using high-level groups (which have variation between those of different classes), this result may indicate that more selection data is required to consistently measure population level differences in Schelling saliency between permutations of four shapes, without high-level groups to filter the many permutations which could be put on display. Additionally, shape variation may have been too narrow across each class of shape (or the number of shapes in each class, too low), and given this narrow variation, collecting selection samples based on 4 shapes per question may have been unlikely to produce population/class level statistics that may appear when larger collections of shapes are shown. Therefore, we found these results to be inconclusive.

Corr. Coef.	**Schelling Frequency (without high-level groups)**			
	Tables	**Lamps**	**Chairs**	**Abstracts**
Naturalness	**0.48**	0.1817	**0.31**	0.05
Strangeness	**-0.49**	-0.1372	-0.24	-0.11
Visual Appeal	**0.53**	**0.49**	0.2854	**0.76**
Memorability	0.2445	0.0707	0.1592	**0.5547**
Standing Out	0.3006	0.2028	-0.0244	0.2843
Uniqueness	0.129	0.1018	-0.1859	-0.112

Table 4.5 - Correlations between some human understandable terms (naturalness, strangeness, and visual appeal) and Schelling frequencies (without high-level groups). Significant correlations ($p<0.05$) are in bold.

4.4.4 Statistical Comparison of Schelling Frequencies Obtained With/Without High Level Groups

We aimed to determine whether the distributions of Schelling frequencies differed whether we employed the use of high-level groups or not.

Firstly, we performed a two-sample Kolmogorov-Smirnov test over the Schelling frequency distributions with/without high-level groups for each shape class. They were each determined to be the same, apart from the chairs shape class (abstract: $p < 0.761$; chairs: $p << 0.01$, lamps: $p < 0.991$ and tables, $p < 0.341$).

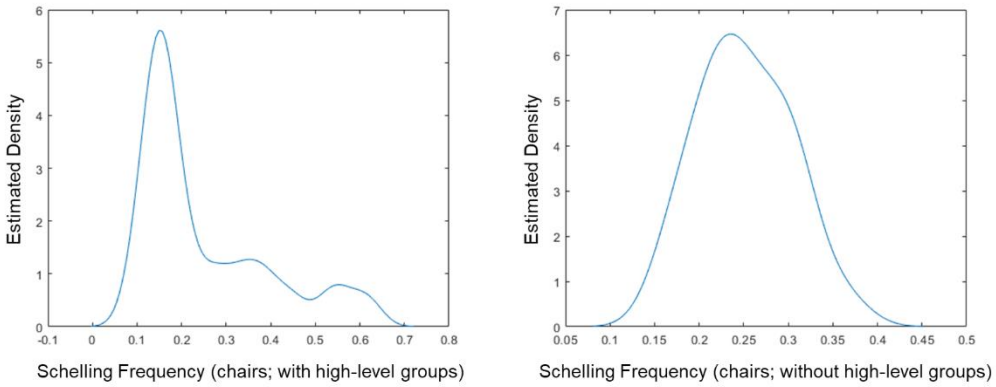


Figure 4.9 – Estimated PDF of Schelling frequencies for the chairs class of shape with/without high-level groups. (Left) With high-level groups. (Right) Without high-level groups.

The difference in shape of the chair-based distributions that was indicated by the two-sample Kolmogorov-Smirnov test is visualised in Figure 4.9. This may be attributed to the type of grouping applied to the chairs (aesthetic ratings). It provides plots of the estimated PDF for the chair Schelling frequencies, with/without high-level groups. Plots of estimated PDFs for the other shape classes are shown in Figure 4.10.

Random choice for each of the PDFs would look like a uniform distribution. As there is a clear deviation from this, it indicates that there is some information in the Schelling frequencies. Additionally, consistency between the distributions of 3 out of 4 shape classes, indicates Schelling frequency distributions are likely to be consistent across different shape classes, whether or not high-level groups are used.

We additionally performed a two-sample t-test on the Schelling frequencies for each shape class produced with/without high-level groups. The distribution means were considered to be the same ($p > 0.05$) for each class of shapes, including the chairs class.

From these statistical results, we focused on using Schelling frequencies and predicting Schelling scores based on data collected with high-level groups, so we refer to these from this point forward.

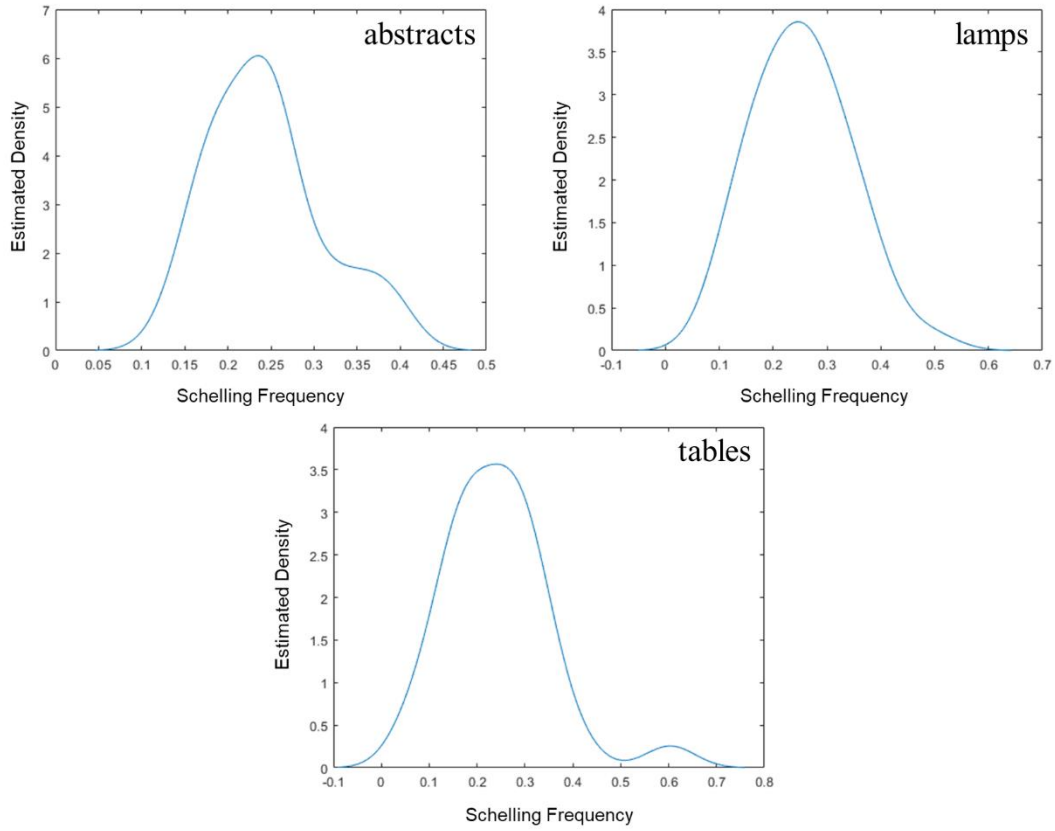


Figure 4.10 – Estimated PDF of Schelling frequencies for the abstract, lamp and table classes of shape based on high-level groups.

4.5 Learning

In this section, we describe the method used to learn and predict Schelling scores using a neural network. We represented a 3D shape with voxels, which the neural network took as input.

We now describe our learned function for Schelling score prediction, and the neural network architecture which represents it. As the Schelling concept is a relative one, the input to the function consisted of multiple shapes. In accordance with each sample of the collected data, we provided four shapes (each in a voxelised form) as input (S_A, S_B, S_C, S_D) to the network, which output four Schelling scores (y_A, y_B, y_C, y_D) in response. We wished to learn the function $(y_A, y_B, y_C, y_D) = h_{W,b}(S_A, S_B, S_C, S_D)$ where W, b are the weights and biases respectively, of a neural network.

y_A is the computed Schelling score, and s_A is the Schelling score from participant provided data (see *Data Collection*). Each of the computed Schelling scores corresponded to how Schelling salient that shape was, relative to the other three shapes. Figure 4.11 shows the neural network used for the input voxel resolution of 32x32x32. A small voxel resolution produced a relatively small number of nodes in the layers, therefore a fully-connected network could be trained for voxel grids of a resolution below this. On the other hand, a large voxel resolution led to a large number of nodes, and convolutional layers were needed. We trained the network using k=10 fold cross-validation, shuffling and splitting the total dataset per class into 90% training data, 10% testing data. For each fold, we correlated the network predictions with actual selection data from participants, to produce our accuracy measure. We averaged these per-fold correlations to achieve our final results.

We produced different neural network architectures for different voxel resolutions, and this was done with the motivation of making the best predictions of Schelling frequencies for each resolution. The network architectures for the other voxel resolutions were similar to that of the diagram in Figure 4.11. Dropout and batch normalisation after convolutions, was employed. For 16x16x16, there was only one

convolutional layer in the blocks on the left. For 12x12x12 and 8x8x8, all the layers were fully-connected with only dropout in use, at the first layer. Rectified Linear Units were used at each layer (after convolutions, or otherwise used to form fully connected layers), apart from the last layer which was a linear combination of weights and previous outputs.

To learn W, b we minimised the following loss function:

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{b}) = \frac{1}{N}\Bigg(\sum_{i \in TrainData} \left(y_{i_A} - s_{i_A}\right)^2 + \sum_{i \in TrainData} \left(y_{i_B} - s_{i_B}\right)^2 + \sum_{i \in TrainData} \left(y_{i_C} - s_{i_C}\right)^2 + \sum_{i \in TrainData} \left(y_{i_D} - s_{i_D}\right)^2 \Bigg) + 0.01\|\boldsymbol{W}\|_2^2 + 0.01\|\boldsymbol{b}\|_2^2$$

Equation 4.2 – Loss function for Schelling frequency prediction (4-choose-1 approach).

Index i denotes the ith training data sample, $d_i = (S_{i_A}, S_{i_B}, S_{i_C}, S_{i_D}, s_{i_A}, s_{i_B}, s_{i_C}, s_{i_D})$, N is the number of samples used for training (in this case, the training batch size), y_{i_A} is the corresponding output in the function $h()$ for S_{i_A} (similarly for S_{i_B}, S_{i_C} and S_{i_D}), and $\|\boldsymbol{W}\|_2^2$ and $\|\boldsymbol{b}\|_2^2$ are L^2 regularizers employed to prevent overfitting.

We minimised Equation 4.2 with the Adam optimisation algorithm and standard backpropagation, where the learning rate decayed by 20% every 300 epochs. Optimisation was executed in batch sizes of 32. Adam uses a weighted combination of an exponential moving average of previous gradients and the current squared gradient, to minimise a loss function.

4.5.1 Neural Network Structure

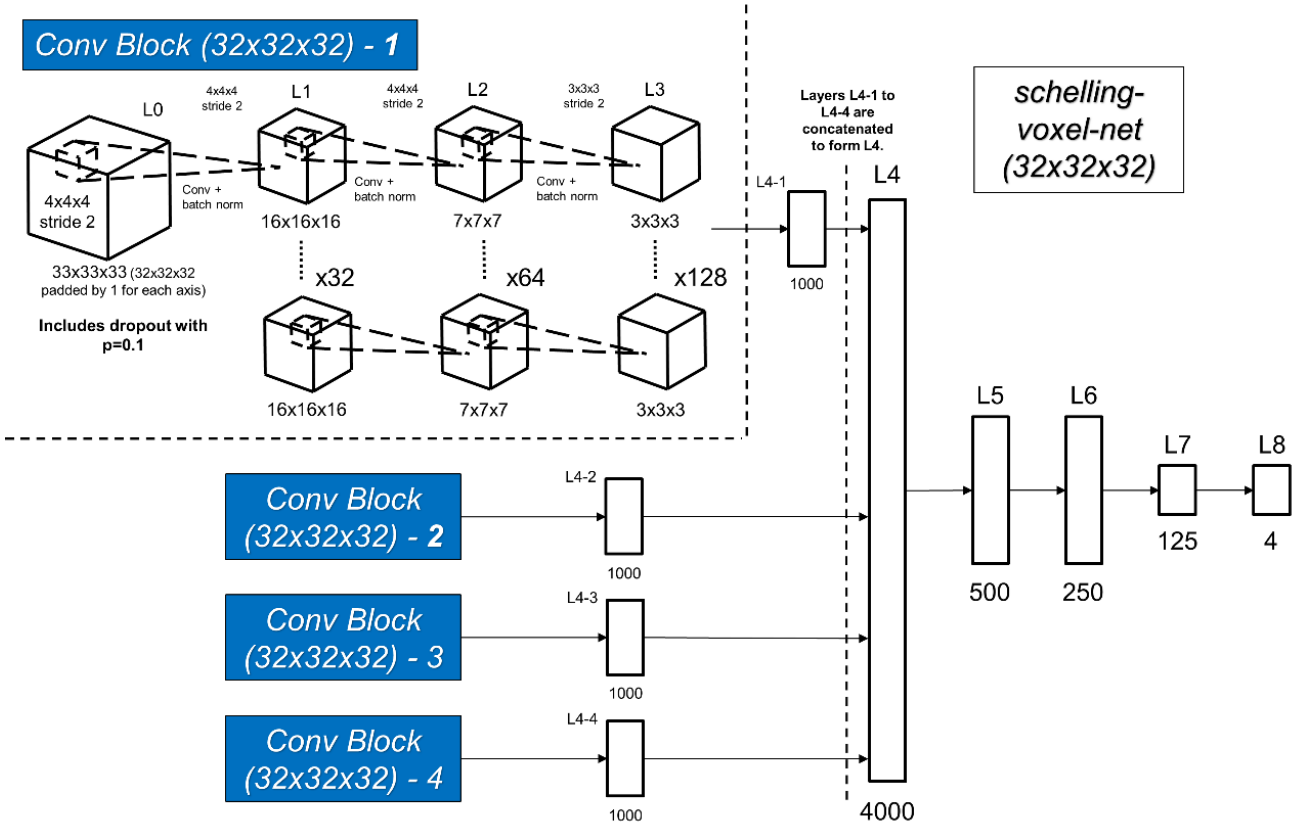


Figure 4.11 – Schelling saliency neural network for an input voxel resolution of 32x32x32. There are 4 input shapes and 4 output Schelling scores. Layers after L4 are fully-connected.

The training time for each shape class was 2000 epochs per fold at approx. 6 seconds each, for 10 folds. This led to an approximate training time of 33.3 hours per shape class. Our implementation used the *Theano* [255] and *Keras* [256] Python libraries. Training was done with a Nvidia GTX 1080Ti graphics card. Refer to Table 4.6 for our results. Also note that a random predictor is 25% accurate.

Shape Class	Number of Shapes	Number of Samples	Number of Samples for Validation	CV Correlation	R^2
Abstracts	30	3420	380	0.42	0.177
Chairs	45	3420	380	0.48	0.226
Lamps	44	3690	410	0.3	0.0089
Tables	30	2704	301	0.467	0.22

Table 4.6 – k=10 fold cross-validation results from training a voxel-based neural network for predicting Schelling saliency via 32x32x32 voxel grids (4-choose-1 approach).

For every shape class in Table 4.6, we achieved significant cross-validation correlations ($p << 0.05$), greater than that of a random predictor, indicating that Schelling scores can be learned. But the correlations were relatively low, especially for the lamps. This suggests that larger amounts of selection data, shapes or computation/time may be needed to improve results further.

4.6 Applications

We demonstrate the use of Schelling salient meshes in various Schelling-based applications.

4.6.1 Visualisation

Here, the idea is to visualize a set of shapes in one larger image or collage, with the Schelling concept influencing their locations in the larger image. We represented each shape as a vector of binary numbers by converting each of them to 15x15x15 voxels.

Afterwards, we used *tSNE* [41] to convert the vector to two dimensions and plot the smaller images of each shape into one larger image. In a second case, we then combined the 2D embedding/location for each shape with its Schelling frequency. These 3D values were then converted into 2D via t-SNE, and the smaller images of each shape were then plotted into one larger image. Figure 4.12 and Figure 4.13 show a comparison of these two cases, for the chairs and tables, respectively. The plots created with Schelling frequencies allow shapes to be grouped which have the smallest and largest Schelling frequencies, and allow comparison of shapes with similar Schelling frequencies to each other.

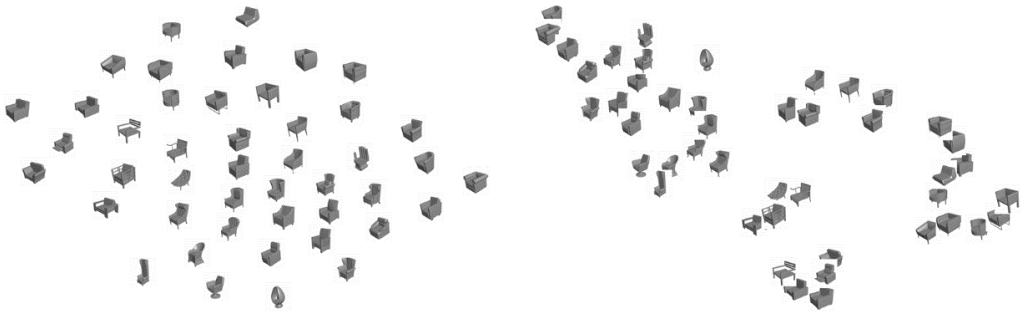


Figure 4.12 - Schelling-based visualizations of chairs obtained using t-SNE. (Left) one without and (Right) one with Schelling frequencies considered before reducing to two dimensions.

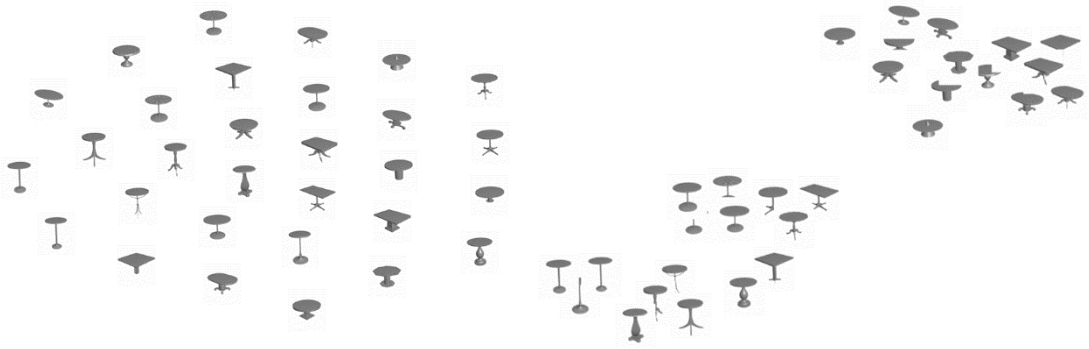


Figure 4.13 - Schelling-based visualizations of tables obtained using t-SNE. (Left) one without and (Right) one with Schelling frequencies considered before reducing to two dimensions.

4.6.2 Clustering

The Schelling concept can be used to cluster a set of shapes into groups based on high-level criteria that would otherwise be difficult using shape geometry alone. For example, the plots in Figure 4.4 to Figure 4.6 show this. For the plots of man-made objects (i.e. lamps, tables, and chairs), the shapes that are near the right end of the spectrum (with higher Schelling frequencies) tend to be more creative, strange, and/or uncommon. The shapes near the middle of the spectrum tend to be more common.

4.6.3 Search

The Schelling concept can be applied for search and retrieval applications of 3D model datasets. The idea is to use the Schelling frequencies or scores as a distance metric such that the distance between two shapes is the difference between the Schelling frequencies or scores. This is effective in placing those shapes that are similar in the Schelling context, to a query shape near the top in a search application. Figure 4.14 shows some examples.

Figure 4.15 and Figure 4.16 show the closest k=5 shapes to a query shape of high Schelling frequency, based on the Euclidean distance between their Schelling frequencies and a range of shape descriptors. The shapes closest to the query in terms of Schelling frequency tend to be more unusual than the shapes closest to the query in terms of each shape descriptor. This suggests that Schelling meshes are the extreme meshes in a shape class/distribution. Or in other words, the higher the Schelling frequency, the more extreme the shape is.

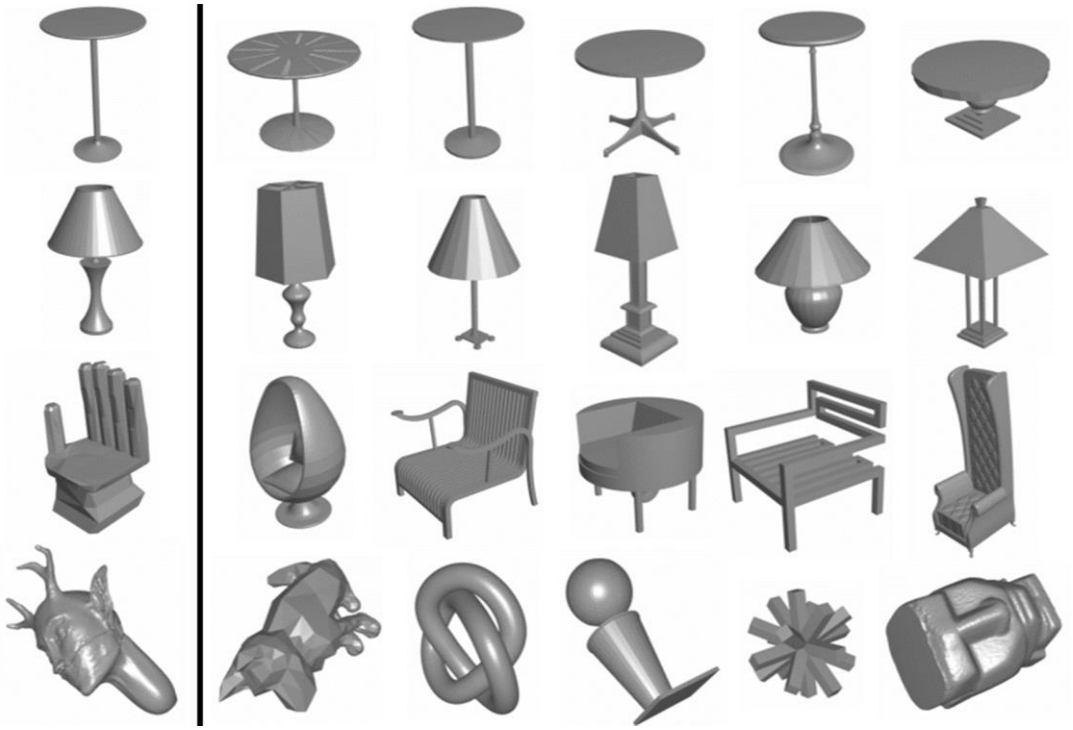


Figure 4.14 - Schelling-based Search. Four examples of searching with a query shape (shown on the left). In each case, the 5 closest shapes based on Schelling frequencies are shown. Moving from left-to-right, indicates increased distance from the shape to the query.

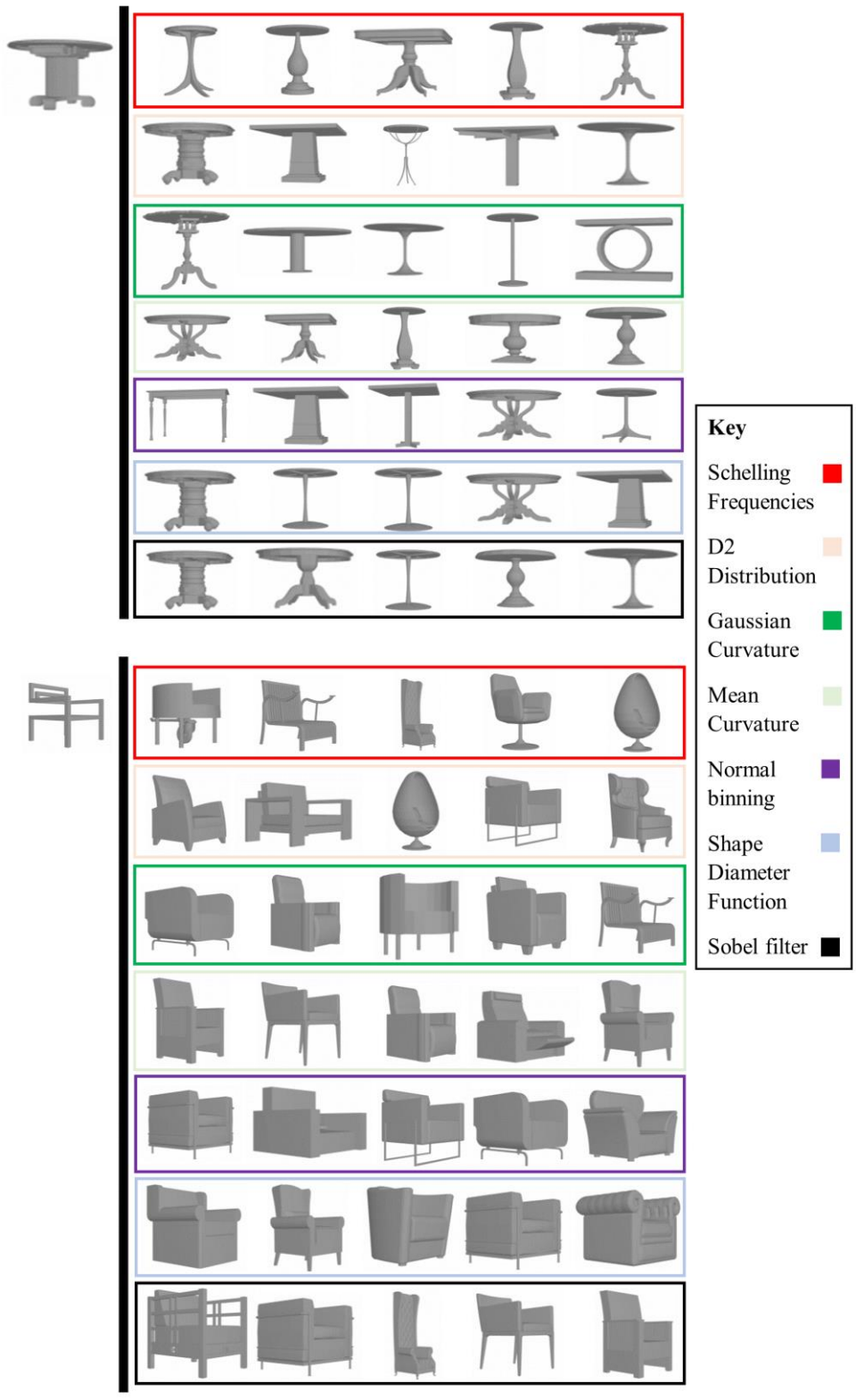


Figure 4.15 - Schelling-based Search 2. Two examples of searching with a query shape (shown on the left) of high Schelling frequency, one each for the chair and table shapes. The 5 closest shapes to the query are shown based on Schelling frequencies and various shape descriptors. Moving from left-to-right, indicates increased distance from the shape to the query.

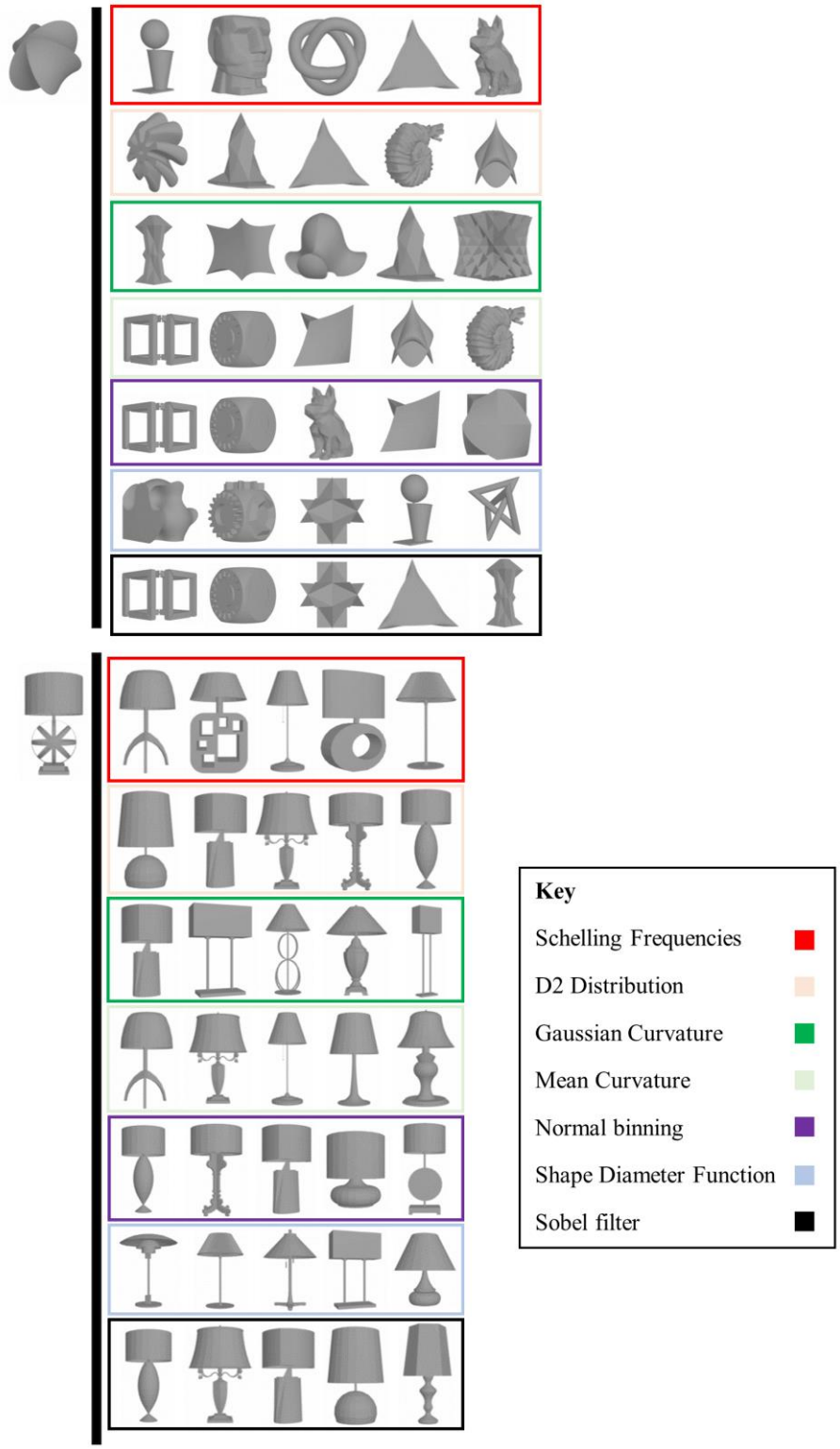


Figure 4.16 - Schelling-based Search 3. Two examples of searching with a query shape (shown on the left) of high Schelling frequency, one each for the abstract and lamp shapes. The 5 closest shapes to the query are shown based on Schelling frequencies and various shape descriptors. Moving from left-to-right, indicates increased distance from the shape to the query.

4.7 Discussion

To begin, we restate our hypotheses below:

1. A more natural shape is less Schelling frequent, as there is not much special (surface variation) that makes it stand out against other shapes.
2. A stranger shape is more Schelling frequent, as the strangeness will make it stand out against other shapes.
3. A shape which stands out from others or is considered unique, is more Schelling frequent due to global rarity of elements on the shape’s surface (related to naturalness and strangeness).
4. A more visually appealing shape is more Schelling frequent, as the appeal/aesthetics of the shape will attract people to select it.
5. A shape may be perceived as more memorable relative to others, as its Schelling frequency increases.

We discuss our results from a viewpoint of high-level groups being necessary to collect Schelling saliency data.

We have found that when using high-level groups to distinguish between shapes to present to participants (from a class), there was only a negative correlation between the Schelling frequencies of the lamp shapes and 'visual appeal', indicating that hypothesis #4 was incorrect.

We also found that natural shapes were less Schelling frequent, for two out of four shape classes (tables and chairs), agreeing with hypothesis #1. But, as this result did not occur for two other shape classes, this is not a general observation.

Additionally, two out of four classes produced positive correlations of 'strangeness' with their shape Schelling frequencies (tables and chairs). This is also not a general trend,

but it provides evidence for hypothesis #2. Strong positive correlations (for the tables and chairs) were also found for notions of 'standing out' and 'uniqueness', which are similar in meaning. These results provide evidence for hypothesis #3, but this also cannot be taken as a general conclusion.

We also found that two out of four shape classes (abstract shapes and chairs) gave positive correlations between their Schelling frequencies and a notion of 'memorability'. As these classes were disjoint from the table and chair classes, this indicates that there are different factors behind understanding Schelling frequencies. This result provides evidence that hypothesis #5 is correct, but not generally so.

Overall, we concluded that potentially: 1) not enough permutations of shapes were presented to participants, to consistently measure class-level patterns, 2) not enough shapes were shown per question, or 3) without predetermined criteria (for example, via the high-level groups), some participants do not understand which choice to make in a Schelling selection task, involving the comparison of a small group of shapes in a class.

4.8 Conclusion

We have explored the notion of Schelling points, where points are the meshes themselves, and we have studied various aspects of this problem. We found that there are many factors behind Schelling frequencies including uniqueness, visual appeal, memorability and other aspects. Whether using high-level groups or not, the Schelling frequencies of all shape classes significantly correlate with the average Likert ratings of at least one subjective term.

Schelling meshes can be predicted at accuracies above chance, using a voxel-based convolutional neural network (correlations of predictions with actual Schelling

frequencies were positive and not near zero – see section 4.5). From our experience, having more shapes and selection data is likely to improve prediction accuracies.

Many shapes in our dataset were quite similar, making it difficult to differentiate among them, based on Schelling frequencies. For example, in the results of the Schelling frequency plots, many shapes can fall within a large group in the middle of the spectrum. Our Schelling-based analysis is less useful for these shapes and more useful for shapes that are in the “extreme” ends of the spectrum.

For data collection, we limited this study to showing participants four shapes, asking them to pick one. We could have instead showed them a larger number of shapes and let them pick any number they wanted to. In addition, we could have mixed shapes from all the available categories in each question. The abstract shapes from this study were mixed already, to some degree, but it was possible to consider all shapes at the same time, without any shape classes. We later attempted this, but participants found it difficult to select shapes consistently. On restricting shapes shown via this approach, by not allowing overlap between classes, we achieved more consistent results. This is the basis of the next chapter.

5 Schelling Meshes: 'Many-Within-Class' approach

5.1 Introduction

Recall that Schelling Points are choices made by people when they aim to match with what they expect others to choose, with no prior communication between them. We can treat explicit shape selections out of a group of shapes as Schelling points within that group. Shapes with high selection frequencies in this context act as Schelling points within their respective shape class.

To remove the restriction on the size of the shape group, which was 4 shapes at a time for the previous study, we designed a new survey setup which allows participants to select as many shapes as they like, with their choices still being represented as binary values. This gave participants more freedom in their answers. Shapes were still restricted to being shown in separate classes at a time, to reduce the difficulty in making selections, as we could not find patterns in Schelling frequencies collected with mixed shape groups.

As before, our aim was to determine the degree of agreement between people when they are individually asked to select a shape of out of a collection that they believe others will also pick. We collected Schelling-based data for meshes by asking people to choose one or more shapes from a class of shapes (e.g. tables, lamps), where people selected shapes that they believed others would also select, given no communication beforehand.

This agreement was reflected in the frequency of shape selections in each context, or ‘Schelling frequencies’. We studied these shape selections and their distributions to determine what makes a shape Schelling salient, given a different selection framework.

We show that the notion of Schelling salient meshes can be learned via a depth image-based convolutional neural network, allowing for Schelling frequency prediction. We compare a selection of traditional shape descriptors to deep-learning approaches for prediction of Schelling frequencies, and achieve better prediction accuracy in each case, using a deep-learning approach. Results are shown for several types of 3D shapes. We demonstrate that the concept of Schelling meshes in this study is useful for the applications of Schelling-based visualisation, clustering, and search.

Overall, we found that Schelling meshes are those that people consider more prominent and stand out with respect to other shapes in a dataset. This suggests that they can represent a dataset’s extremes. They are also perceived as memorable relative to the other members of their class.

5.2 Hypotheses

5.2.1 We retained some of the original hypotheses from the Layout and Structure

Following the *Introduction* is a *Background* chapter, covering key concepts needed to understand the work in this thesis, such as information on geometry representations, data collection, statistical tests / analysis, and relevant topics in machine learning.

Afterwards is the *Related Work* chapter, which provides an overview of previous research that is related to the theme of the thesis. It covers four main topics: 1) Saliency + Shape Perception, 2) Understanding of Geometry, 3) Machine Learning, and 4)

Crowdsourcing. A summary is provided for each topic. In the conclusion section of the chapter, research gaps are indicated which are most relevant to the thesis research, highlighting the contributions we provide that complement the existing literature.

In the *Schelling Meshes: '4-choose-1'* chapter, we introduce the notion of 'Schelling meshes', an approach to understanding 3D shapes via a basis of human preference. We study the agreement between participants when they select one out of four shapes, aiming to match other people's selections. We detail our data collection method, interpret and analyse the results, and describe our approach to learning and predicting which shape is most likely to be a Schelling mesh out of a group of four shapes. We also provide potential applications in search and visualisation, using shape selection frequencies given shape visibility by participants. To conclude the chapter, we discuss the approach and report our main findings.

The next chapter (*Schelling Meshes: 'Many-Within-Class'*) introduces an approach to collecting data on Schelling meshes where participants can select multiple shapes within a class, aiming to match others' selections, as before. We interpret and analyse our results and provide a method to predict how likely a shape is to be a Schelling mesh out of a shape class. This is our 'Many-Within-Class' approach. To conclude the chapter, we report and discuss our main findings.

To follow, we study 2D shapes in the *Font Specificity* chapter. This covers our approach to understanding 2D fonts via the concept of Specificity. We detail our data collection approach and show the results of our analysis. Based on these results, we show how per-font word distributions can be used to create a Specificity score and detail an approach to automatically compute Specificity scores with similar properties. We also provide a method to predict font Specificity and introduce potential applications in

search, visualisation and clustering. To conclude, we report and discuss our main findings.

We end the thesis with a *Conclusions* chapter, discussing how the topics of Schelling meshes and font Specificity relate to the thesis’ theme of understanding 3D shapes and 2D fonts via human-perceptual aspects of their geometry. We provide potential future applications, and areas of research that could follow from this work.. These are listed below:

1. A shape which stands out from others or is considered unique, is more Schelling frequent.
2. A more visually appealing shape is more Schelling frequent.
3. A shape may be perceived as more memorable relative to others, as its Schelling frequency increases.
4. Schelling frequencies convey different information to that of shape descriptors.

5.3 Methodology

5.3.1 Data Selection and Generation

We firstly took the previously collected 145 3D shapes from the ‘4-choose-1’ study (30 abstracts, 45 chairs, 44 lamps and 30 table shapes) and increased this number to 169 shapes, taken from online sources (e.g. *ShapeNet* [8] and *Trimble 3D Warehouse* [2]). Some data collection and analysis was performed with this shape dataset, which will be shown later in the chapter. It consisted of 38 abstract shapes, 49 chairs, 49 lamps and 33 tables.

Further analysis was done using a larger dataset of 387 shapes, which included the previous 4 classes, and another 7: bottles, baskets, cabinets and shelves, cups, plants,

plates and pots. The shape datasets in question are referred to when applicable, throughout the chapter. See Table 5.1 for shape class sizes and per-class mean Schelling frequencies for the dataset of 387 shapes.

Shape Class	Number of Shapes	Mean Schelling Freq.
Abstracts	38	0.1290
Baskets	28	0.1629
Bottles	28	0.1114
Cabinets/Shelves	32	0.1238
Chairs	49	0.1084
Cups	31	0.1077
Lamps	49	0.1228
Plants	31	0.1439
Plates	21	0.1838
Pots	47	0.0957
Tables	33	0.1245

Table 5.1 – Shape class sizes and the mean Schelling frequencies of each class.

As in the '4-choose-1' study, we displayed 3D shapes on a 2D screen, via a continuously rotating view of each shape. All shapes went through a full rotation (taking three seconds) where one could also see the top and bottom parts of the shape. This was followed by a short pause before the shape was rotated again.

For this study, we aimed for shapes to be selectable in an analogous manner to points in the *Schelling Points on 3D Surface Meshes* work [6]. Specifically, participants would need to be shown multiple shapes on screen, potentially all shapes of a given class (e.g. chairs), where they would choose several of the presented shapes, with a goal of matching other participants' responses based on what they believed others would choose. In the previous study, we did not collect data in this way, as we originally believed it difficult to learn a function that takes many shapes as input. But for this study, we allowed many shapes to be selected per question and determined an approach

to learn a function to predict Schelling saliency, which takes depth images of a shape as input. It was possible to mix classes but separating them was already interesting for us to study the concept of Schelling meshes.

5.3.1 Data Collection

In this section, we describe the process of collecting data to study Schelling meshes. The main difference between this study design, and the last, is in the representation of the Schelling concept as applied to 3D shapes. As mentioned before, it requires humans to analyse a shape relative to other shapes.

We used the *Amazon Mechanical Turk* crowdsourcing platform, to collect data. Each question displayed all shapes in a shape class. Each shape had a selection box for the participant to indicate choosing it or not, and each selection box was independent from the others. Unlike the ‘4-choose-1’ approach, we did not restrict shapes shown to participants, based on criteria. By doing this, we aimed to reduce bias in the shape selection process/survey. Since participant selections were subjective, and there were no right or wrong answers (there were no combinations more valid than any other – unlike in the more restricted ‘4-choose-1’ case), we decided to not have any control questions to filter out potentially bad users.

Participants were first given written instructions: *“For each question, your task is to choose from a selection of shapes. Other participants will be given the same task. You should choose shapes that will most likely match with their selections. Note that you will not be able to communicate with other participants, and this is intentional”*. They were also told to choose at least one shape per question.

Each HIT ("Human Intelligence Task" or set of questions on Amazon Mechanical Turk) consisted of several questions, one for each class of shape. Shape order was randomized per question. We provided questions based on 11 shape classes, and so split surveys into 3 or 4 questions at a time, for different participants. At the end of each HIT, we included an optional text box and asked participants to provide a few words describing why they selected the shapes that they did. Figure 8.1 (in the appendix) provides a screenshot of a survey provided to participants. A shape *selection* was a binary choice (like in the '4-choose-1' study), but in this case, participants could choose many shapes until the total amount of shapes per shape group was reached.

Each participant provided a binary-valued vector, s for an ordered set of shapes, S. Associated with each shape S_j was a score, s_j indicating whether it had been selected (1 = selected, 0 = not selected). Therefore, each selection sample was of the form (S_j, s_j), where S_j was the participant's stimuli. Equation 5.1 provides the Schelling frequency, s_{f_j} for a shape, S_j.

We took the number of participants as N, and the number of shapes in a class as M. For a shape S_j, which had a score, s_j, across many participants' selection samples, P_i:

$$j \in \{1, \dots, M\}$$

$$number\ of\ selections\ of\ S_j = S_{j_{total}} = \sum_i s_{j_i}, \qquad \left(S_j, s_j\right)_{\mathrm{i}} \in P_i$$

$$number\ of\ occurrences\ of\ S_j = S_{j_{occ}} = N$$

$$s_{f_j} = \frac{S_{j_{total}}}{S_{j_{occ}}}$$

Equation 5.1 – Schelling frequency derivation ('Many-Within-Class' approach).

However, we attempted to introduce data quality standards. In the provided instructions, we tried to encourage users to carefully work on the questions by specifying to users that if they randomly chose their answers, their HIT responses would not be taken, and they would not be paid. Furthermore, users were only allowed to work on our HITs if their acceptance rate of previously completed HITs, was at least 80%. A participant took about 1 to 5 minutes to complete each HIT. We paid participants $0.10 for each HIT.

Qualitative Characteristics of Schelling Meshes

Next, we attempted to understand the qualitative characteristics of Schelling meshes from participant text responses. We collected responses from 102 participants (made at the end of their surveys), for the abstract shapes, chairs, lamps and tables (169 shapes). 46 out of 102 participants gave comments. 18 participants mentioned they made selections based on appeal, aesthetics, or beauty.

An example user comment was: "I basically selected items that I liked and items which I thought other people would like as well". 16 participants said they selected shapes that stand out, are different, or catchy. For example, one user commented: "I selected the shapes that are unusual and different from others". This suggested that Schelling meshes may be perceived to stand out or be unique from others in a collection. A few participants said that they chose familiar or memorable shapes. For example, one user commented: "I hope that most of the other participants have also gone for similar designs as they are common and easy to remember". This indicated that perception of shape memorability could be a factor behind Schelling mesh selections.

5.4 Analysis

5.4.1 Validation of Data Consistency

At this point, we wanted to determine whether the collected data was consistent among participants. Since the data was subjective, there was no right or wrong answer to compare against. Hence, we checked the consistency within the collected data. The main idea was to split the whole set of data into different groups and check whether the groups had similar distributions.

We performed this test for the abstract shapes, chairs, lamps and tables (169 shapes), given selections from 102 participants. The collected data consisted of binary values indicating whether each of the participants selected each shape. We randomly sampled from this dataset, 10 times, where each time we randomly picked half (51) of the participant responses. Half of the responses still provided us with information about all shapes, giving us 10 vectors of 169 values. Figure 5.1 shows a visual representation. We can see that there is much correspondence in the horizontal rows, where some rows are mostly blue and light blue, and some rows are mostly yellow and orange. This means that across the 10 vectors, the distributions of the values are similar. We found that a minimum of 40-50 participants was required to show a consistent pattern in Schelling frequencies, across all shape classes (see Figure 5.1 or columns 4 and 5 of Figure 5.2).

Quantitatively, we performed a two-sample Kolmogorov-Smirnov test for each pair of 169 values (pairs sampled from the 10 vectors) and found that the p-value >= 0.05 in each case. This provided evidence that these 10 vectors came from the same distribution and that there was consistency in the collected data. So, for the remaining shape classes which contributed to the dataset of 387 shapes, we collected data from 50 participants for each shape class (baskets, bottles, cabinets/shelves, cups, plants, plates and pots).

Using some of the additional shapes, we performed a different test where we asked participants to make selections from a small set of shapes (all from the same class), and separately asked people to select from a large set of shapes (from the same class). Then we determined whether the selections were consistent, by attempting to correlate the Schelling frequencies produced for the separate distributions of small vs. large shape sets.

We showed 100 participants surveys in a very similar setup to what was mentioned previously, but the difference is that we independently showed people a survey of questions with only 12 shapes randomly sampled from a shape class, vs. a survey of questions showing all shapes in that class.

Data was collected on the *pots* shape class in this manner, where we showed participants 95 pots (which contained the original 47 pots) vs. 12 pots (sampled from the original 47 pots). In the former case, shape order was randomised. In the latter case, shapes were randomly sampled with replacement.

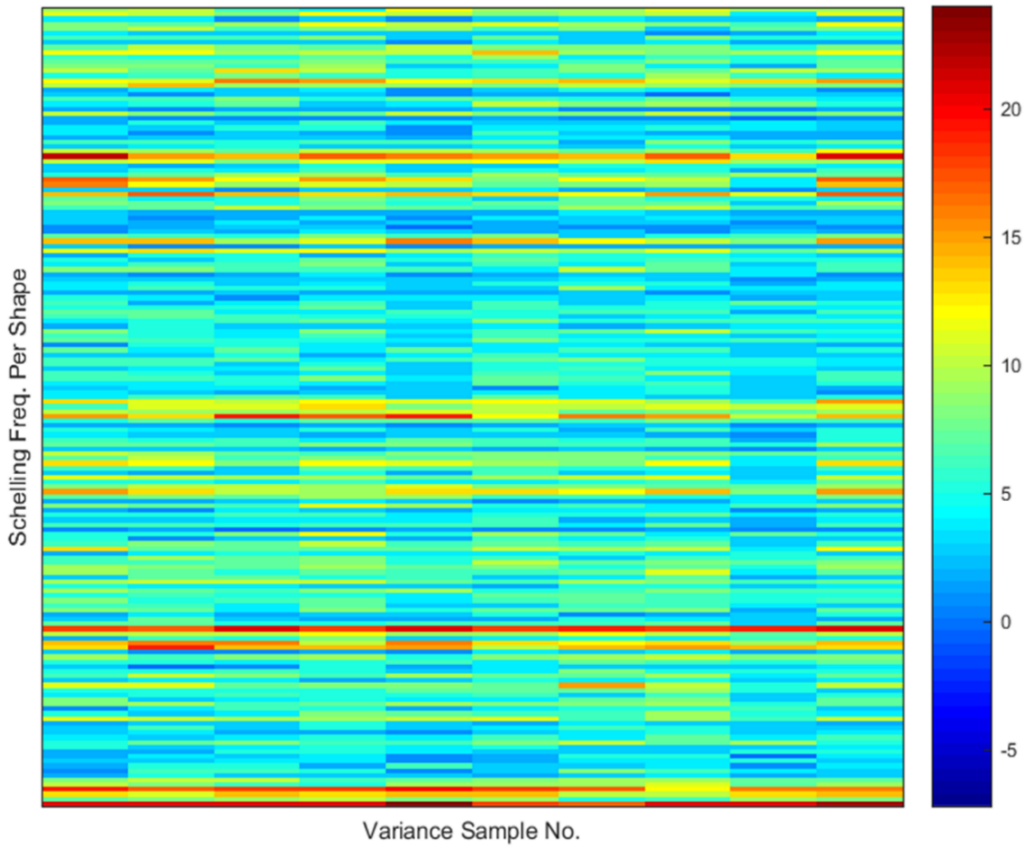


Figure 5.1 – Plot showing variance in Schelling frequency distributions according to shape selections randomly sampled from 51 participants (out of 102) with replacement.

The y-axis indicates each shape (from top to bottom) across the abstract shapes, chairs, lamps and tables (169 shapes). The x-axis/columns represent samples of Schelling frequencies for each shape.

Horizontal lines of the same or similar colour (selection frequency), indicate consistency between the samples.

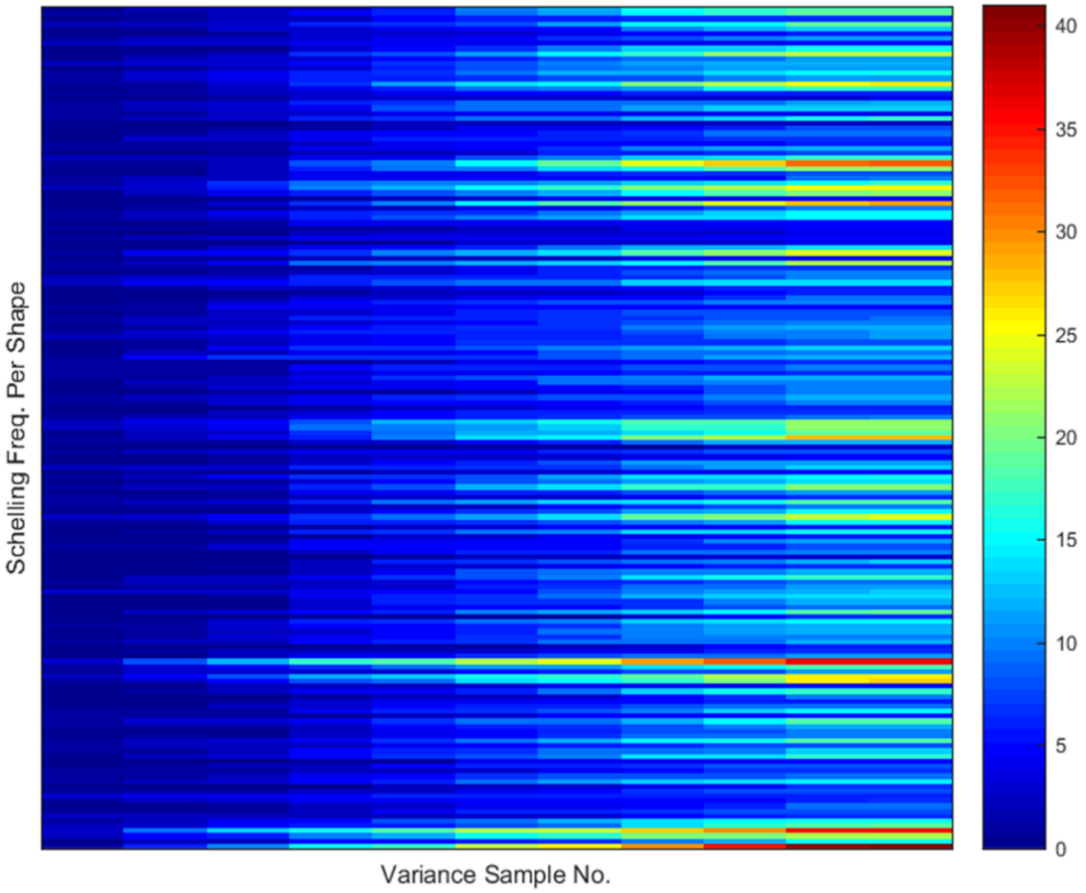


Figure 5.2 – Visualisation of how Schelling frequencies become more stable as more selections are gathered.

The y-axis indicates each shape (from top to bottom) across the abstract shapes, chairs, lamps and tables (169 shapes). The x-axis/columns represent samples of Schelling frequencies for each shape, derived from incrementally summing 10 randomly chosen participants’ shape selections, each sampled without replacement.

Consistently distinct horizontal regions of colour across columns, indicates consistency in the shapes’ Schelling frequencies, from a certain column onwards. This indicates a minimum selection sample size.

Additional shapes were collected from *ShapeNet* [8] in order to hold the survey. For both cases of survey, participants were asked to select a number of shapes. We aimed to compare the Schelling frequencies of the original 47 pots, given that they were mixed within the 95 pots, with Schelling frequencies obtained from incrementally collecting selection data on 12 pots out of the original 47 pots (eventually producing Schelling frequencies for the 47 pots).

Since not all shapes were shown when providing 12 out of a larger total, the Schelling frequency definition changed (see Equation 5.2), since the number of participants, N, was not necessarily the number of times a shape was shown to participants. The number of shapes in the class was M, as before. In this case, for a shape S_j, which has a score, s_j, across many participants' selection samples, P_i, the shape's Schelling frequency, s_{f_j}, is calculated as follows:

$$j \in \{1, \dots, M\}$$

$$member(S_j) = \begin{cases} 1 & \text{if } S_j \in P_i \\ 0 & \text{if } S_j \notin P_i \end{cases}, \qquad s_j = 0 \text{ if } S_j \notin P_i$$

$$number\ of\ selections\ of\ S_j = S_{j_{total}} = \sum_i s_j \times member(S_j)$$

$$number\ of\ occurrences\ of\ S_j = S_{j_{occ}} = \sum_i member(S_j)$$

$$s_{f_j} = \frac{S_{j_{total}}}{S_{j_{occ}}}$$

Equation 5.2 – Schelling frequency derivation (consistency test of 'Many-Within-Class' approach).

Given that the mean number of occurrences (in surveys) of each shape was 50, we obtained results as shown in Table 5.2. This is approximately the target number of times

that we wished participants to see a shape, to get a good distribution of Schelling frequencies (see Figure 5.2).

Consistency Test	**Pots (47 shapes total)**
Corr. Coef.	0.5676
Corr. Coef. (shapes with >= 50 occurrences)	0.6348
R-Squared	0.307
p-value	p << 0.01 (3.17E-05)
Mean	0.2155
Median	0.2
Min	0.0333
Max	0.5
Std. Dev.	0.1047

Table 5.2 – Correlations between Schelling frequencies from the pots class where all 47 shapes were shown (within a larger 95 pots group) vs. Schelling frequencies obtained via showing 12 shapes at a time, incrementally.

Mean, Median, Min, Max, and Std. Dev., refer to statistics obtained on the final Schelling frequencies across each shape in the *pots* shape groups (which were shown 12 shapes at a time), respectively.

Correlations between Schelling frequencies obtained via showing participants 12 *pots* shapes at a time, vs. showing participants all 47 pots (mixed within 95 pots), were significant and positive. The correlations improved if we only included shapes with a high number of occurrences (seen by 50 participants or more).

This suggests that incrementally collecting shape selections for smaller groups of shapes at a time, can be a valid option for consistently obtaining Schelling frequencies vs. doing so by showing all shapes in a class, allowing Schelling frequencies to be collected in more realistic situations. Further validation with additional shape classes would be needed to confirm this, however. Additionally, a greater number of participants than 50

may be required to achieve high correlations to Schelling frequencies obtained from shape selection data collected in bulk.

5.4.2 Observed Patterns

Firstly, we describe the patterns that we observed in the plots of the first 169 shapes that we studied (See the Visualisation section for plots of other shape groups). Figure 5.3 provides 1-D plots of these shapes (abstract shapes, chairs, lamps and tables).

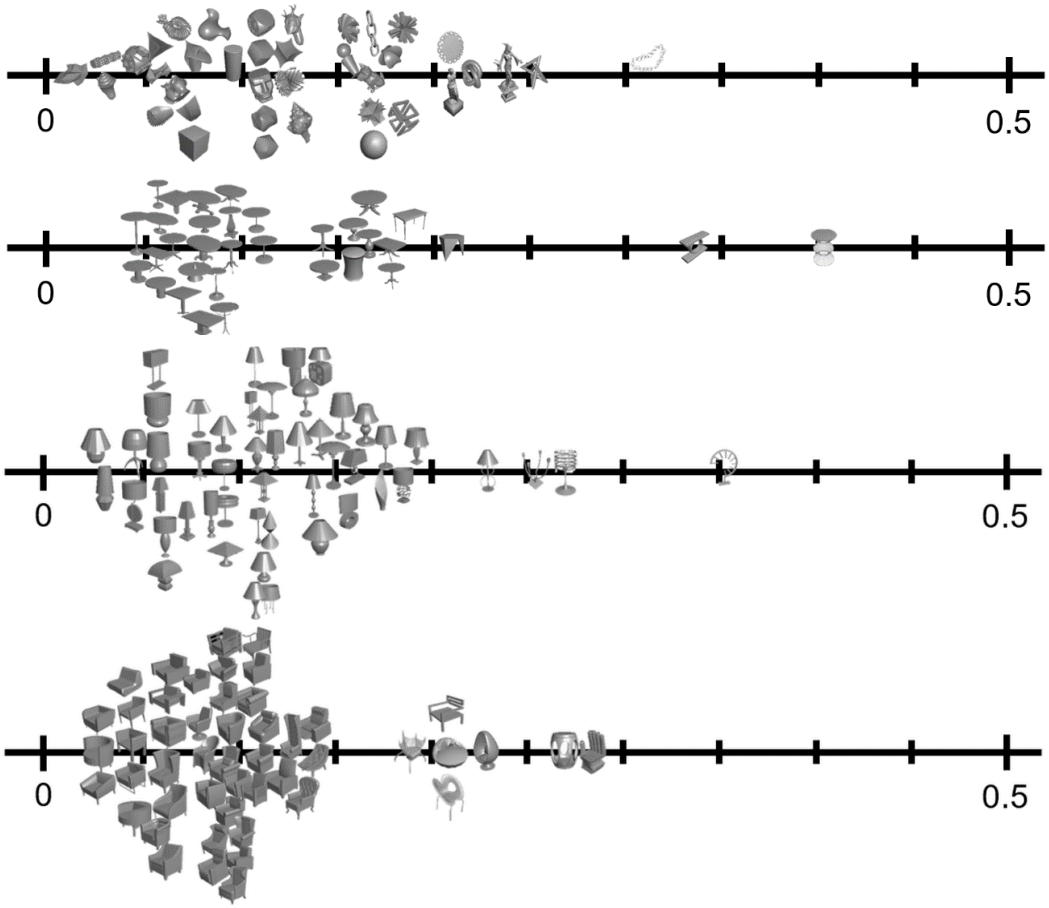


Figure 5.3 – 1-D plots of shapes at their respective participant Schelling frequencies. We show one plot for each of the abstract shapes, tables, lamps and chairs shape classes.

Regarding the chairs, shapes that looked more strange, unusual, and/or different from the others had higher Schelling frequencies. In contrast, the plain or normal-looking chairs were all clustered into one big group near the left side of the 1D plot. Similarly,

for the tables and lamps, the non-typical and/or strange shapes had higher Schelling frequencies, while the more normal-looking shapes were mostly clustered on the left side.

For abstract shapes, chain-like shapes (rings or holes) and the statue shapes had higher Schelling frequencies. The lower Schelling frequency shapes tended to be blobs, primitives, or otherwise abstract or unknown shapes.

Comparing the range of Schelling frequencies for the four shape categories, the category with the smallest range was the chairs. It seemed that there was less variety or creativity in the provided types of chairs, compared to the tables and lamps, even though all 169 shapes were examples of furniture. This led to a smaller range for the chairs.

5.4.3 Comparison with 3D Shape Descriptors

Another characteristic that participants pointed out, is that they chose shapes that were appealing or aesthetically pleasing. We tried to measure the aesthetics of a shape (in the form of a polygon mesh) by computing curvature since it is regarded to be related to aesthetics [4]. We computed the Gaussian curvature and mean curvature [67] for each mesh vertex and then averaged them to get curvature values for each shape. We tried to correlate each of these averaged values for all shapes, with their Schelling frequencies but found that there was no correlation ($p > 0.05$). Therefore, we did not observe any quantitative correlation between aesthetics and Schelling frequency (although curvature is only one possible measure of aesthetics).

In addition, we computed for each shape, histograms of some common 3D shape descriptors: D2 shape distribution [9], Gaussian curvature and mean curvature [67], and the Shape diameter function [31]. Descriptors were computed using MATLAB code,

and C++ via *LibIGL* [257] and *CGAL* [258]. We ranked these descriptors by participant Schelling frequency, in ascending order, and plotted each descriptor as a vertical colour/heatmap. As an example, for the 33 tables, 33 vertical columns are shown in the plot. This allowed us to visualise if there were correlations between individual shape descriptors, and the participant Schelling frequencies. Visually, we found no clear correlation between each descriptor and Schelling frequency. See section 5.5.5 for the mesh processing that we applied to all shapes before computing descriptors on them.

Figure 5.4 and Figure 5.5 show some heatmaps (for the abstract shapes, chairs, tables, plants and all shapes together) based on our larger 387 shape dataset (which consisted of: abstract shapes, lamps, tables, chairs, plants, pots, plants, bottles, baskets, cups and cabinets/shelves).

5.4.4 Understanding Schelling Frequencies through Subjective terms

To determine what a 'high', 'low' or even 'average' Schelling frequency may mean, we collected data on how people ranked shapes based on subjective terminology, as in the '4-choose-1' study. As previously, we used the Amazon Mechanical Turk platform for this. Participants were paid $0.10 per HIT.

Similarly, to the '4-choose-1' case, we provided participants with Likert surveys, with each question showing a single shape and a scale. We asked them to rank how visually appealing, memorable, unique or 'stand out' each shape was, by choosing an option on a scale, between 1 and 5 for each shape, with 1 being "Not at all", and 5 being "Extremely".

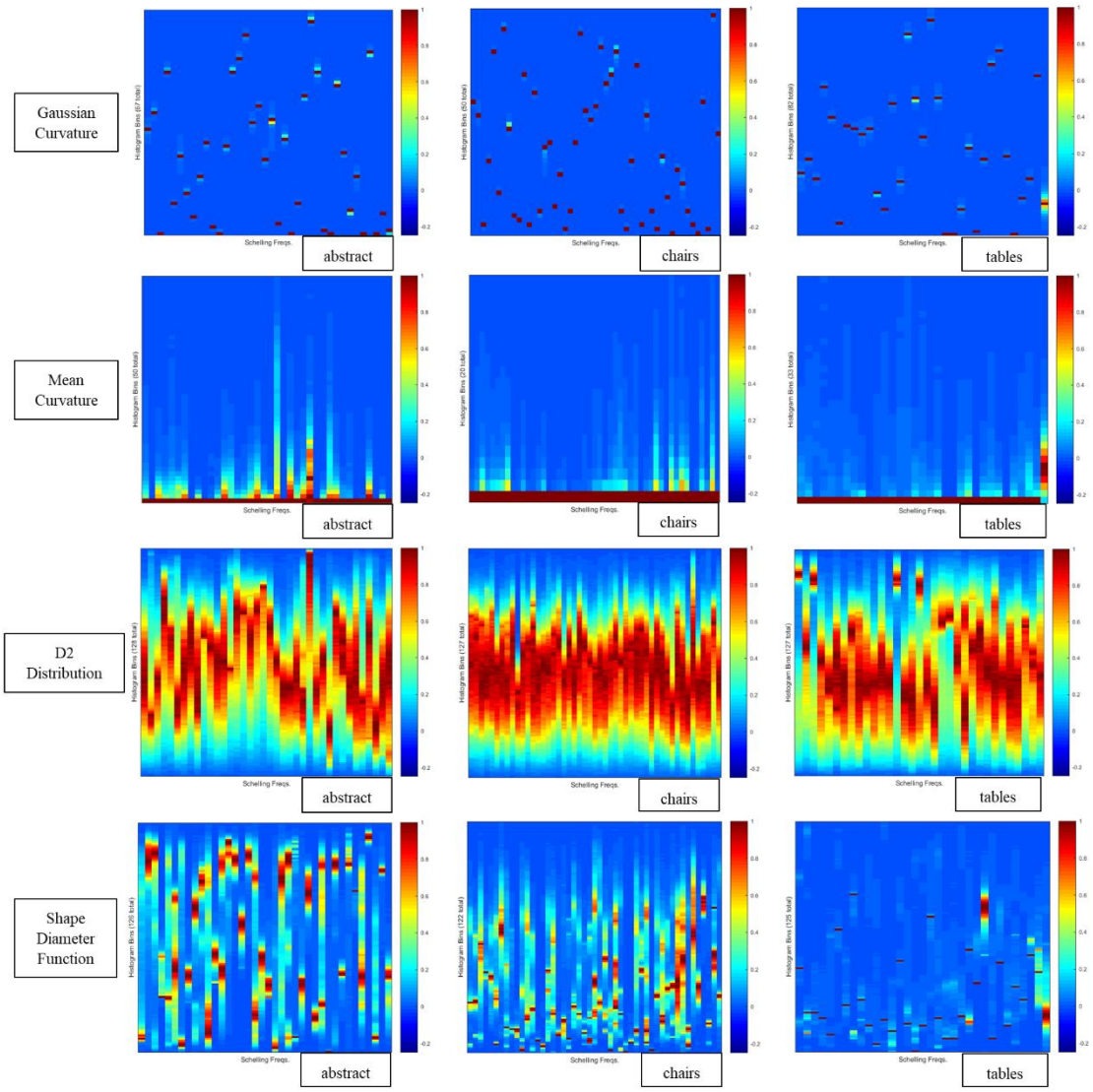


Figure 5.4 – Plots of shape descriptor histograms for the abstracts, chairs, and tables shape groups, with each column representing one shape, where columns are sorted according to increasing Schelling frequency. Bins with values of < 5e-3 were removed.

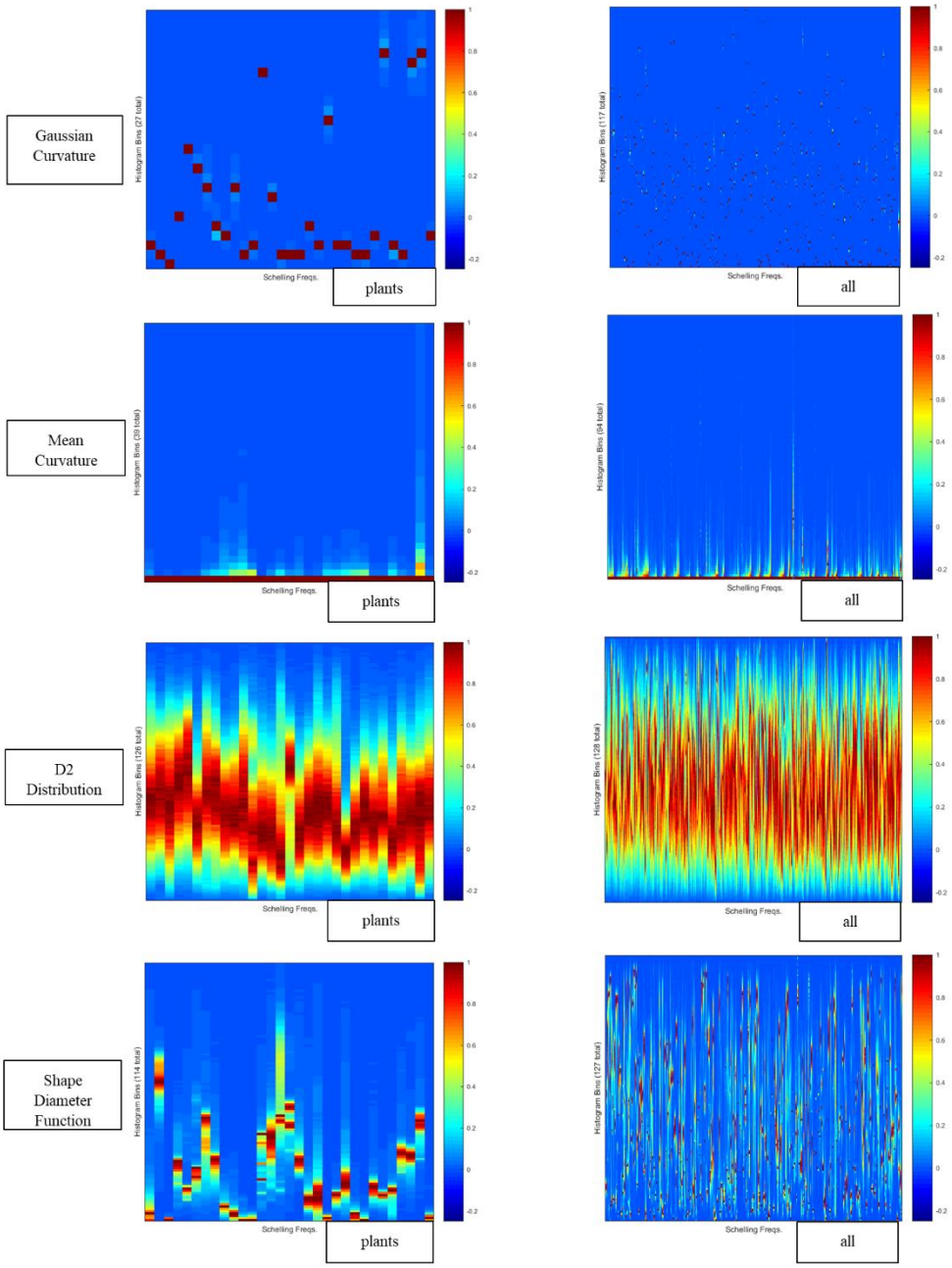


Figure 5.5 – Plots of shape descriptor histograms for the plants shape group, as well as all shape groups combined, with each column representing one shape, where columns are sorted according to increasing Schelling frequency.

Groups of 15 unique participants were each shown a survey on a shape class and criterion combination (e.g. baskets + memorable). Shapes were sampled from a single shape class, without replacement. Table 5.3 shows correlations between the average Likert scores for each shape (from 15 participants) and each shape’s Schelling frequencies as obtained via showing all shapes in each shape class (from 50 participants). Bold values indicate statistical significance (p-values < 0.05).

For 10 out of 11 shape classes, participant perception of shape memorability correlated significantly and positively with the Schelling frequency for each shape. This suggests that a shape’s perceived memorability is likely to be positively correlated with the Schelling frequency of a shape, especially since significance occurred for very different shape groups. For the memorability and ‘standing out’ cases, Schelling frequencies could be treated as a sort of prior for further shape analysis or processing.

Previous results on image memorability [259, 260] have suggested a counter-intuitive result, that the perception of memorability is inversely correlated with actual memorability. The authors created a *Memory Game* in which participants were asked to memorise a sequence of images. Their task was to indicate (via key press) whenever they saw an identical repeat of an image at any time in the sequence. The memorability score for an image was the percentage of correct detections, by participants. People were also asked to estimate whether they were likely to remember an image the following day. They found that the human estimates of image memorability were negatively correlated (corr. = −0.19) with the true memorability of images. However, this is a different class of data (2D images), so the correlation (which spans a large dataset of images but is also weak) may not apply directly to 3D shapes or scenes. A memorability study of Schelling meshes could be informative in determining this.

Corr. Coef.	Abstract	Baskets	Bottles	Cabinets-Shelves	Chairs	Cups	Lamps	Tables	Plants	Plates	Pots
Memorable	**0.5008**	0.3193	**0.545**	**0.4271**	**0.3153**	**0.5184**	**0.4007**	**0.6452**	**0.3838**	**0.7568**	**0.4311**
Stand-out	-0.1026	**0.3875**	0.0144	**0.5059**	**0.5839**	**0.4281**	**0.3015**	**0.635**	0.0582	**0.6813**	**0.2986**
Uniqueness	0.1298	0.3336	**0.4234**	**0.4198**	**0.439**	0.2987	0.2418	**0.6337**	0.0868	**0.6876**	0.1976
Visual Appeal	**0.6005**	**0.3907**	0.052	0.213	0.1512	0.2612	**0.4987**	**0.3703**	0.1764	**0.7263**	-0.0172

Table 5.3 - Correlations between the average Likert scores for each shape (from 15 participants), and each shape's Schelling frequencies as obtained via showing all shapes in each shape class (from 50 participants). Significant correlations ($p < 0.05$) are in bold.

All other criteria had reasonable counts of significance with the Schelling frequencies, indicating that only a single criteria/dimension is not sufficient to model the Schelling frequencies of 3D meshes, with the term "*stand out*", or a shape's prominence, significantly correlating 8/11 times; *uniqueness*, 5/11 times, and *visual appeal*, 5/11 times. This suggests that the hypothesis of positive correlation with visual appeal is true for nearly half of the shape classes, but this is less consistent than the more frequent occurrences/classes where the hypotheses relating to (perception of) memorability and a notion of "standing out" are correct. This leads onto the next section. How can we predict, or model Schelling meshes, given this study setup?

5.5 Learning

To attempt to generalise our Schelling frequencies to shapes we had not yet encountered, we created a convolutional neural network which takes as input a collection of three depth image views of a shape, i, and produces a single output value, $\hat{y}_i$, as a Schelling frequency prediction. Depth images were taken at a resolution of 128 x 128, so each training sample was of the form of: (X, y_i), where $X \in \mathbb{R}^{3\times128\times128}$, and y_i was a Schelling frequency derived from participant selections.

For each depth image in a sample, its pixel values corresponded to depth/intensity values in $[0, 1]$ (although the values were originally quantized as integers that lie within $\{0, \ldots, 255\}$ via OpenGL), except for pixel values corresponding to the background, which were set to -2.

To generate samples for each shape, depth images were taken in groups of 3 orthogonal views. Firstly, a random location is sampled on a sphere (of fixed radius) with a shape at its centre. From this location, 3 orthogonal views of the shape are generated by

rotating 90 degrees around each of a pair of axes that are perpendicular to the sampling camera's direction and each other. For example, if we started looking at the shape along the z-axis (a possible camera direction), from the x-y plane, we could rotate 90 degrees around the y-axis, moving us into the y-z plane. We would then look along the x-axis towards the shape. Similarly, we could now rotate around the z-axis, moving us into the x-z plane, where we would look along the y-axis towards the shape (the rotations can be anti-clockwise or clockwise, but must be fixed before generating samples). Taking a depth image at each stage, generates 3 orthogonal views of a shape (of fixed order). If we associate the Schelling frequency, y_i of the shape with these views, that generates a single data sample for training.

In order to generate enough examples for training, we performed data augmentation by: 1) looking around the entire shape and 2) generating new samples from different random locations. To look around an entire shape, we repeated the earlier sampling process another 5 times, rotating around the shape from the initial random location at 60-degree intervals, from 60 to 300 degrees inclusive (using a fixed axis, such as the y-axis). This produced a total of 6 samples (6 collections of 3 orthogonal views). To obtain different groups of 360-degree views, we repeated the entire sampling process 17 times per shape, to obtain 102 samples per shape, in total. This was done to reach an approximate target of 100 samples per shape. Finally, we associated the Schelling frequency, y_i, of each shape, i, with each of the 102 samples (by duplicating the Schelling frequency). This would indicate to the model that it would need to look for primitive features and more complex structures from multiple viewpoints of the same shape, to generalise well. In other words, the network would be less likely to regress to a Schelling frequency based on a shape's orientation alone. The convolutional aspect of the network also provides translation invariant outputs, given the same input, allowing important features

of a depth image to be searched for across an image, at the scale of the convolution window/filter. For specific parameters, see Figure 5.6.

We trained the convolutional network to learn a hypothesis function: $h(X) = \hat{y}_i$, to predict Schelling frequencies using 3 orthogonal depth images. These can be automatically extracted per shape (with Schelling frequencies duplicated for each sample corresponding to the same shape). The training process aimed to minimise the error between predicted Schelling frequencies, $\hat{y}_i$, and the associated target (or participant derived) Schelling frequencies, y_i (shown in Equation 5.3), using the mean-square error loss function as a measure of accuracy. Weights were optimised via stochastic gradient descent and standard backpropagation. The stochastic gradient descent executed in batch sizes of 24. $\|\boldsymbol{W}\|_2^2$ and $\|\boldsymbol{b}\|_2^2$ were L^2 regularizers employed to prevent overfitting.

In Equation 5.3, index i is the i-th sample in the training data and N is the number of samples used for training (the training batch size).

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{b}) = \frac{1}{N} \sum_{i \in TrainData} (y_i - \hat{y}_i)^2 + 0.01\|\boldsymbol{W}\|_2^2 + 0.01\|\boldsymbol{b}\|_2^2$$

Equation 5.3 – Loss function for Schelling frequency prediction ('Many-Within-Class' approach).

The network structure used LeakyReLU neurons [261, 262] at each layer, excluding the output layer, which was a linear combination of previous outputs. LeakyReLU neurons allow for a small, non-zero gradient, even when they are inactive (or more specifically, negative). This gradient can still help to propagate information throughout the network, unlike ReLU neurons where the gradient can suddenly change to zero, given negative inputs (for ReLU inputs, when x < 0, the gradient is 0).

This helps to avoid the vanishing gradient problem, that can occur while training neural networks with backpropagation, where gradients which are used to update the parameters/weights of the network, tend to zero, due to the repeated small changes occurring through previous layers, to the input.

Figure 5.6 shows a diagram of the neural network that we trained. We produced a separate instance of this network for each shape class, training it using $k = 10$ fold cross-validation ($\frac{1}{10}$ of the data samples per shape class, were separated into a test set), then averaging the prediction accuracy of each of the held-out test data sets.

Since each shape corresponded to 102 different predictions, as a measure of network performance, we correlated each of these predictions with a duplicate of the participant provided Schelling frequency, y_i, of that shape, i. In addition, we produced the averages of the 102 predictions, for each shape. We report the correlations between these averages $\bar{y}_i$, and the participant provided y_i. Results are shown in Table 5.4. R-squared values are shown in each case, in addition to the shape dataset sizes. The neural network was trained using a Nvidia GeForce 1080ti graphics card with 16GB RAM.

5.5.1 Neural Network Structure (Depth Image-Based)

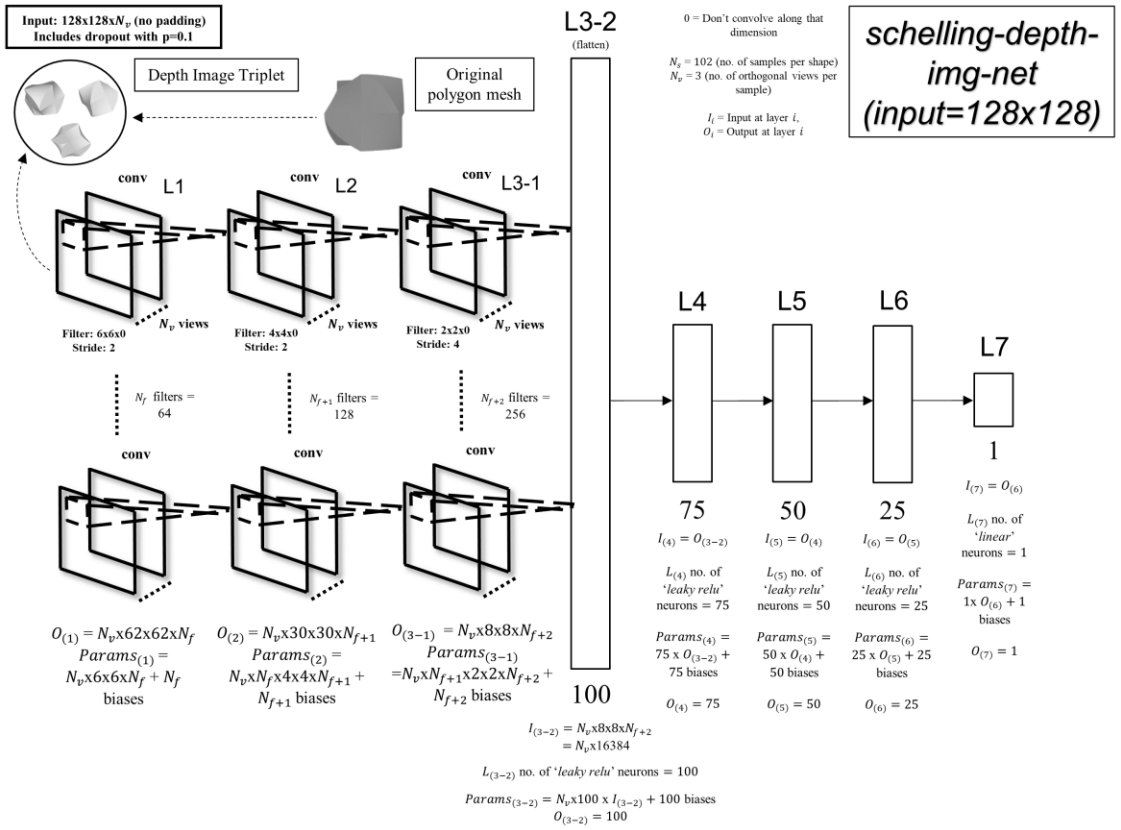


Figure 5.6 – A diagram showing the structure of a convolutional neural network for predicting Schelling frequencies. Its input is a triplet of depth images I_{ij}, with dimensions: $3 \times 128 \times 128$. The output is a single real-valued Schelling frequency prediction, $\hat{y}_i$.

5.5.2 Depth Image-based Results

CV Learning Results	Number of Shapes	Number of Samples	Number of Samples for Validation	CV Correlation	R^2	Average-based CV Correlation	Average-based R^2
Abstracts	38	3488	387	0.91	0.83	0.98	0.96
Baskets	28	2570	285	0.75	0.57	0.93	0.86
Bottles	28	2570	285	0.92	0.84	0.97	0.94
Cabinets/ Shelves	32	2937	326	0.59	0.35	0.87	0.77
Chairs	49	4498	499	0.82	0.67	0.94	0.89
Cups	31	2845	316	0.78	0.61	0.94	0.90
Lamps	49	4498	499	0.71	0.51	0.92	0.84
Plants	31	2845	316	0.72	0.52	0.92	0.84
Plates	21	1927	214	0.75	0.56	0.90	0.81
Pots	47	4314	479	0.77	0.59	0.88	0.78
Tables	33	3029	336	0.77	0.6	0.94	0.89

Table 5.4 – Correlations between Schelling frequency predictions based on depth image triplets and participant provided Schelling frequencies, for each shape class. Additionally, shows the correlation between the average of all predictions across each shape's depth image triplets and each shape's participant Schelling frequency (which was associated with each triplet).

Our implementation used the *Theano* [255] and *Keras* [256] Python libraries. Correlations between the participant-provided Schelling frequencies and the predicted Schelling frequencies, were highly positive, and statistically significant ($p << 0.05$), for every shape class.

In addition, the average correlations derived from $\bar{y}_i$ were always higher than correlations produced from directly mapping participant Schelling frequencies to predictions (by duplicating the Schelling frequencies). We found that if the correlations between prediction averages and participant Schelling frequencies increased as the training process progressed, it showed that the neural network was learning from the data. It was a useful guideline indicator, when training the network.

Since we were successful in predicting Schelling frequencies for individual shape groups, we attempted to train the network using all shapes, given 11x weights at each layer (one multiple for each shape group). Due to memory limitations, we trained the network in batches of approx. 11662 samples, in 3 batches, for 3765 epochs each batch, at a learning rate of 0.01. Samples were randomly collected without replacement for each of the 3 batches, to ensure that all of the training data was seen by the network. The number of training examples taken from each shape class, was weighted according to the number of shapes in each class, relative to the total. A larger proportion of samples was taken from classes with a larger amount of shapes relative to other classes, as a fraction of the entire dataset's shape total (387).

We achieved only a lower than average correlation between the predicted Schelling frequencies and the participant ones. We believe this was either due to computational (memory limitations) or there being contradictions in the reasoning behind selections between some of different shape groups, resulting in the neural network not being able to handle this. Our results are shown in Table 5.5.

CV Learning Results	Number of Shapes	Number of Samples	Number of Samples for Validation	CV Correlation	R^2
All	387	35526	3947	0.50	0.25

Table 5.5 – Correlations between Schelling frequency predictions and participant provided Schelling frequencies across all shapes. Additionally, shows the correlation between the average of all predictions across a shape's depth image triplets and each shape's participant Schelling frequency (which was associated with each triplet).

5.5.3 Voxel-based results

We additionally wanted to compare the results of training via depth images to a voxel-based shape representation. Given an input of a 32x32x32 voxel grid, representing a single shape, the model produced a Schelling frequency prediction for that shape.

Similarly, to the depth image case, we provided 102 rotated versions of each voxel grid (or 102 samples per shape) as potential training examples to a convolutional neural network. Each of these samples was assigned the Schelling frequency of the original shape, as in the depth image case. The network was trained via the Adam optimiser, and k=10 fold cross-validation was employed to achieve our final results. Our results are located in Table 5.6. A diagram of the network is shown in Figure 5.7.

CV Learning Results	Number of Shapes	Number of Samples	Number of Samples for Validation	CV Correlation	R^2	Average-based CV Corr.	Average-based R^2
Abstracts	38	3488	387	0.54	0.297	0.72	0.53
Baskets	28	2570	285	0.099	0.0099	0.55	0.30
Bottles	28	2570	285	0.28	0.08	0.77	0.59
Cabinets/ Shelves	32	2937	326	0.33	0.11	0.55	0.30
Chairs	49	4498	499	0.44	0.19	0.66	0.43
Cups	31	2845	316	0.26	0.07	0.66	0.43
Lamps	49	4498	499	0.36	0.13	0.61	0.37
Plants	31	2845	316	0.22	0.05	0.44	0.19
Plates	21	1927	214	0.05	0.002	0.22	0.05
Pots	47	4314	479	0.11	0.01	0.32	0.10
Tables	33	3029	336	0.52	0.27	0.79	0.62

Table 5.6 - Correlations between Schelling frequency predictions based on voxel grids and participant provided Schelling frequencies, for each shape class. Additionally, shows the correlation between the average of all sample predictions across each shape and each shape's participant Schelling frequency.

We can see that the voxel-based approach performed worse than the depth image-based approach, across all shape classes (correlations are lower, whether from individual predictions or averaged predictions per shape). We believe this was at least partly due to the low resolution of each slice of a voxel grid (32x32) vs that of the depth images (128x128).

Similarly, to the depth-image based results, correlations between a shape's participant-based Schelling frequency and the average of predictions across the samples of each

shape were always higher than the correlation between the list of per-sample predictions and the participant-based Schelling frequency, duplicated for each prediction. This indicated that the model was learning to weight each sample as a contribution to a Schelling frequency, but to a less accurate degree than in the depth image case. The results did not seem to (negatively) correlate with shape class size, as the plates, pots and plants shape classes, which span nearly the full range of class sizes, produced low magnitude correlations, when compared to the rest of the classes. Shape classes with thin structures (e.g. plates, plants) or hollowed-out elements (e.g. baskets, bottles, pots) tended to have the worst results (see *CV Correlation* column of Table 5.6).

Overall, we recommend the use of depth images over voxel grids for the prediction of 3D shape Schelling frequencies, given dataset sizes similar to ours. It may be the case that at higher resolutions, fixed-size voxel grids (e.g. 64x64x64, 128x128x128) provide comparable results or better, but memory would be less efficiently used. For memory efficiency, while retaining increased detail, a dynamic voxel grid representation possibly could be used instead.

5.5.4 A short note on memory usage

The number of bytes in a 128x128 depth image is 128 x 128 x 1 byte = 16,384 bytes, as each value of the OpenGL depth buffer is 8 bits in size. For a fixed-size voxel grid, occupation state can be represented with 1 bit: 32 x 32 x 32 x 1 bit = 32,768 bits, or 4096 bytes. However, 16-bit floating point values are usually used to represent data at training time. This is done to avoid integer overflow. Therefore, memory usage is 32 x 32 x 32 x 2 bytes = 65,536 bytes. This 2-byte, or 16-bit floating point unit is also required in the depth image case. As we used 3 depth images per sample, the actual number of bytes used to represent each sample in this case, is 128 x 128 x 2 bytes x 3

images = 98,304 bytes. The depth image approach requires 1.5x the space, per sample. However, the results were worse when using a 32x32x32 voxel grid representation. Additionally, the required memory for a voxel grid becomes exponentially bigger with each added dimension, indicating that depth images should be tested first, if memory limitations are a factor, before moving to fixed-size voxel grids. Even for a 48x48x48 voxel grid, 48 x 48 x 48 x 2 bytes > 98,304 bytes.

5.5.5 Predicting Schelling frequencies via Shape Descriptors

We asked ourselves: are there any correlations between geometric descriptions and shape Schelling frequencies? To attempt to answer this question, we produced a set of shape descriptor values (binned to histograms) based on the source polygon meshes of the shapes shown to participants. These captured geometric aspects of each shape in our dataset.

Before computing descriptors, we pre-processed meshes using MeshLab [263]. We firstly removed duplicate faces and any vertices detached from the overall mesh structure. Vertices within 1% of the maximum distance between pairs of vertices in each mesh were merged into a single vertex.

Some meshes were still not well-formed, so for consistency, we then voxelised each mesh at a 200x200x200 resolution and reconstructed them via the *marching cubes* algorithm (see section 2.3.1). After this point - in certain cases - we applied *quadric edge collapse decimation* [264] to simplify reconstructed meshes where their face count was above 100000, to reduce their face count to a target of 100000, also attempting to preserve normal directions and topological structure. This reduced unnecessary descriptor computations.

5.5.6 Neural Network Structure (Voxel-Based)

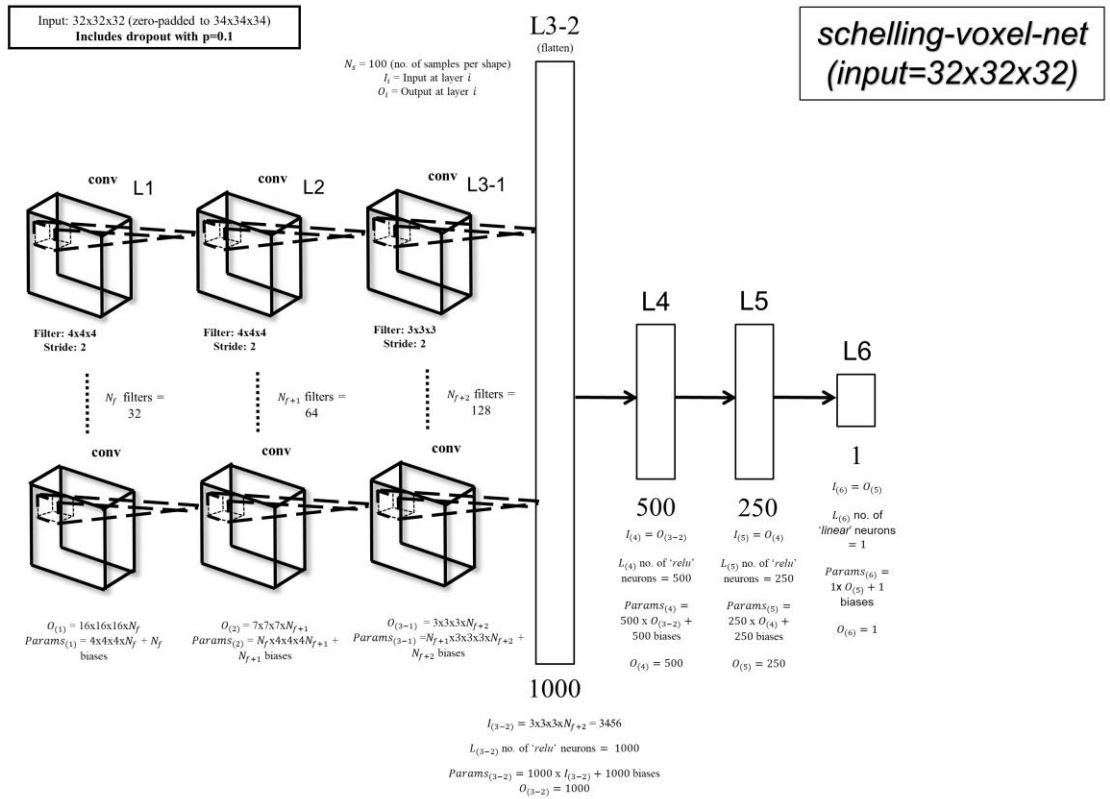


Figure 5.7 – A diagram showing the structure of a convolutional neural network for predicting Schelling frequencies. Its input is a voxel grid with dimensions: $32 \times 32 \times 32$. The output is a single real-valued Schelling frequency prediction, $\hat{y}_i$.

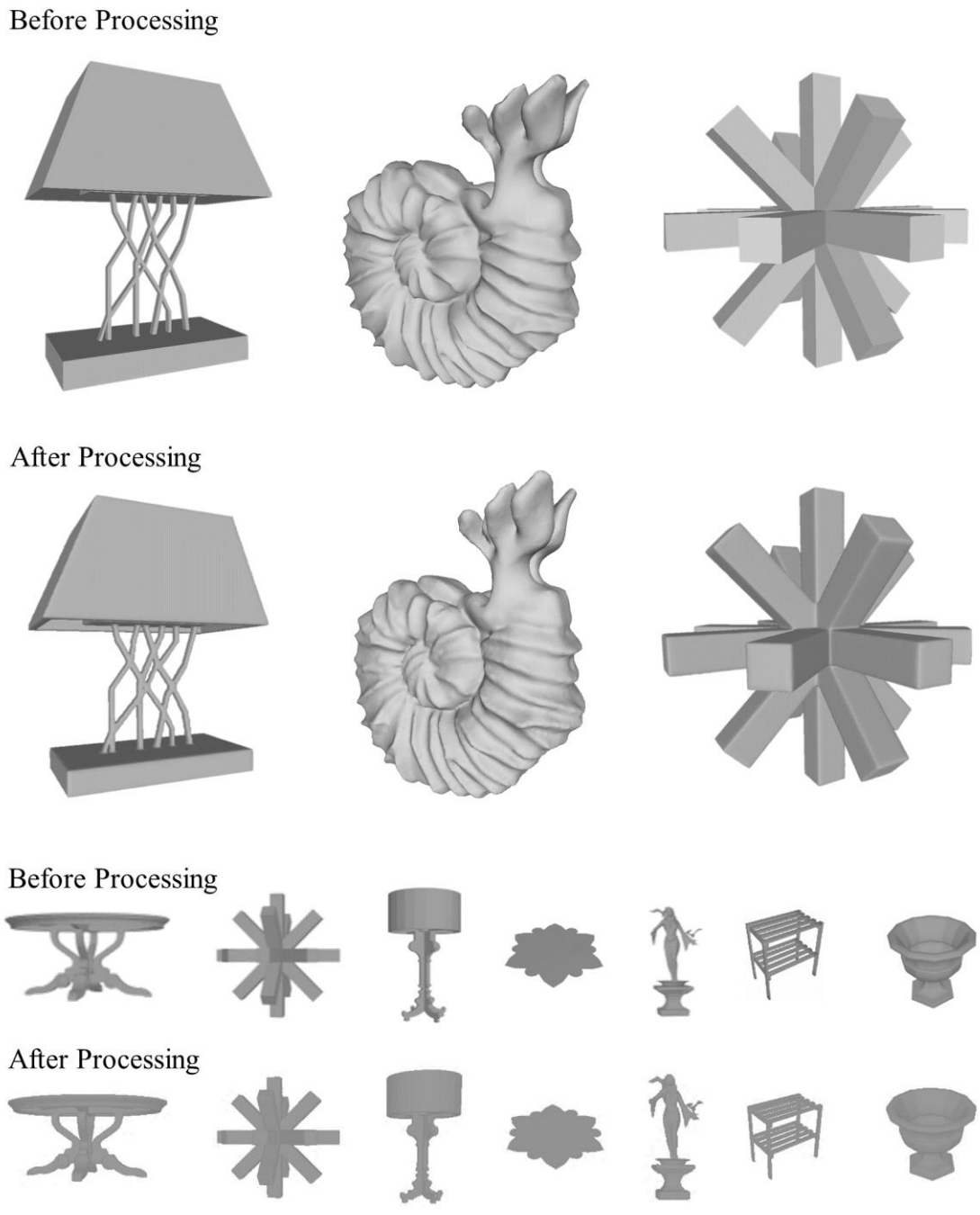


Figure 5.8 – Various meshes before and after the mesh processing required for descriptor computation.

Across all meshes, we then applied 2 steps of *Laplacian smoothing* [265], using a co-tangent weighting scheme (See Curvature section in the Background chapter). This replaced each vertex with a position based on the average of its surrounding vertices,

calculated using angles between edges formed from those vertices and the initial mesh vertex. See Figure 5.8 for renderings of a subset of the meshes, before/after processing.

Descriptors were computed using MATLAB code, and C++ via *LibIGL* [257] and *CGAL* [258]. The chosen descriptors are listed below:

- D2 Distribution
- Shape Diameter Function (SDF)
- Difference in angles of per-vertex normals
- Gaussian curvature
- Mean curvature
- Histogram of Oriented Gradients
- Sobel Filter

Details on these descriptors are provided in the 2D Shape Descriptors and 3D Shape Descriptors sections of the Related Work chapter. We used polygon meshes to represent our 3D shapes for processing, so curvatures and D2 distribution values were weighted by face area. For each descriptor we binned their values into histograms, to produce our shape representation, v_i, per shape, i. In total, each vector v_i had 672 dimensions. Descriptors were concatenated together in a consistent order, to produce each v_i.

The number of bins for each descriptor were as follows:

- D2 Descriptor: 128 bins
- Gaussian Curvature: 128 bins
- Mean Curvature: 128 bins
- Normals: 50 bins
- Shape Diameter Function: 128 bins
- Sobel: 110 bins

We attempted to learn a regression function that maps a single vector of shape descriptor values to Schelling frequencies, in a similar vein to previous work studying Schelling Points on 3D meshes [6], but the authors of this work instead predicted probable vertex selections/regions on polygon meshes. The loss function we minimised was similar to that of the convolutional neural network, but the network was a simpler fully-connected one, in order to minimise the influence of factors regarding the network's design, on the results. The network consisted of 4 layers, taking as input a single $d = 672$ dimensional vector, v_i, representing each shape i, and output a single value, $\hat{y}_i$ - a Schelling frequency prediction. The hypothesis h, is represented by the network: $h(v_i) = \hat{y}_i$.

The number of neurons for each successive hidden layer, were $\lfloor 0.2d \rfloor$, $\lfloor 0.1d \rfloor$, $\lfloor 0.05d \rfloor$ and $\lfloor 0.025d \rfloor$, respectively. We attempted to train the network for each separate shape group (using stochastic gradient descent, and backpropagation, as before), but did not achieve good results in most cases, with predictions varying wildly from their expected values (the participant Schelling frequencies). Increasing the data provided to the network, by training with all shapes, did not greatly change the outcome. Our results are shown in Table 5.7. R-squared values are shown in each case, in addition to the shape dataset sizes. The neural network was trained using a Nvidia GeForce 1080ti graphics card with 16GB RAM.

The main problem was that our shape dataset was too small for meaningful regression (to Schelling frequencies) to be achieved using a fully-connected network. But this suggested that using a multitude of shape descriptors may not necessarily be enough to predict Schelling frequencies for small-to-medium dataset sizes.

A larger shape dataset would also require further Schelling frequency data collection, compared to what is required to train a convolutional neural network, before better results might be obtained.

CV Learning Results	Number of Shapes	Number of Samples	Number of Samples for Validation	CV Correlation	R^2
Abstracts	38	34.2	3.8	-0.08	0.01
Baskets	28	25.2	2.8	-0.28	0.08
Bottles	28	25.2	2.8	0.16	0.026
Cabinets/ Shelves	32	28.8	3.2	-0.065	0.004
Chairs	49	44.1	4.9	-0.34	0.11
Cups	31	27.9	3.1	-0.00026	7.08E-08
Lamps	49	44.1	4.9	-0.117	0.014
Plants	31	27.9	3.1	-0.09	0.009
Plates	21	18.9	2.1	-0.254	0.06
Pots	47	42.3	4.7	-0.025	0.0006
Tables	33	29.7	3.3	0.459	0.21
All	387	348.3	38.7	-0.144	0.021

Table 5.7 – Correlations between Schelling frequency predictions based on shape descriptors and participant provided Schelling frequencies, for each shape class.

This suggests that deep learning methods may be useful for prediction/regression of other aspects of shape perception where a glut of memory or computation time is not available. Neural networks which learn on more complex geometric shape representations, such as polygon meshes, graphs and approximations to continuous manifolds, directly, could be useful here (see the *Geometric Deep Learning*, *Machine Learning* section of the *Related Work* chapter).

5.6 Applications

We show some applications in search, visualisation and clustering.

5.6.1 Search

Schelling frequencies allow you to search for a complementary or contrasting shape to a query shape. Treating a shape as a query, we can take its participant Schelling frequency, and find the closest k shapes. These are the most complementary shapes, in terms of Schelling frequency. The farthest k shapes are the most contrasting, similarly so. We plot search results from a query shape of high Schelling frequency, based on Schelling frequencies and the shape descriptors previously computed (using Euclidean distance as the metric). Some examples for $k = 5$, are shown in Figure 5.9. Each plot is read from left to right, from closest shape to farthest shape. Search results are laid out according to the given key.

As in the '4-choose-1' study, shapes closest to a query shape of high Schelling frequency tend to be more extreme than shapes closest to the query in terms of shape descriptors.

5.6.2 Visualisation

Each shape group for which Schelling frequencies can be obtained, can be visualised as a 1D plot of shapes, according to their ordered Schelling frequencies. The greater the distance between shapes, the more contrasting they are, and vice-versa. The higher a Schelling frequency is, relative to that of other shapes, the more memorable it is perceived to be, and more likely it is to be focused on or selected. Figure 5.10 shows Schelling frequency plots for the pots and cups.

In Figure 5.11, we provide Schelling frequency plots for the abstract shapes, bottles, tables, lamps, chairs and plates. You can see that the more geometrically varied a shape is, relative to the rest of its class, the more likely it is to have a higher Schelling frequency.

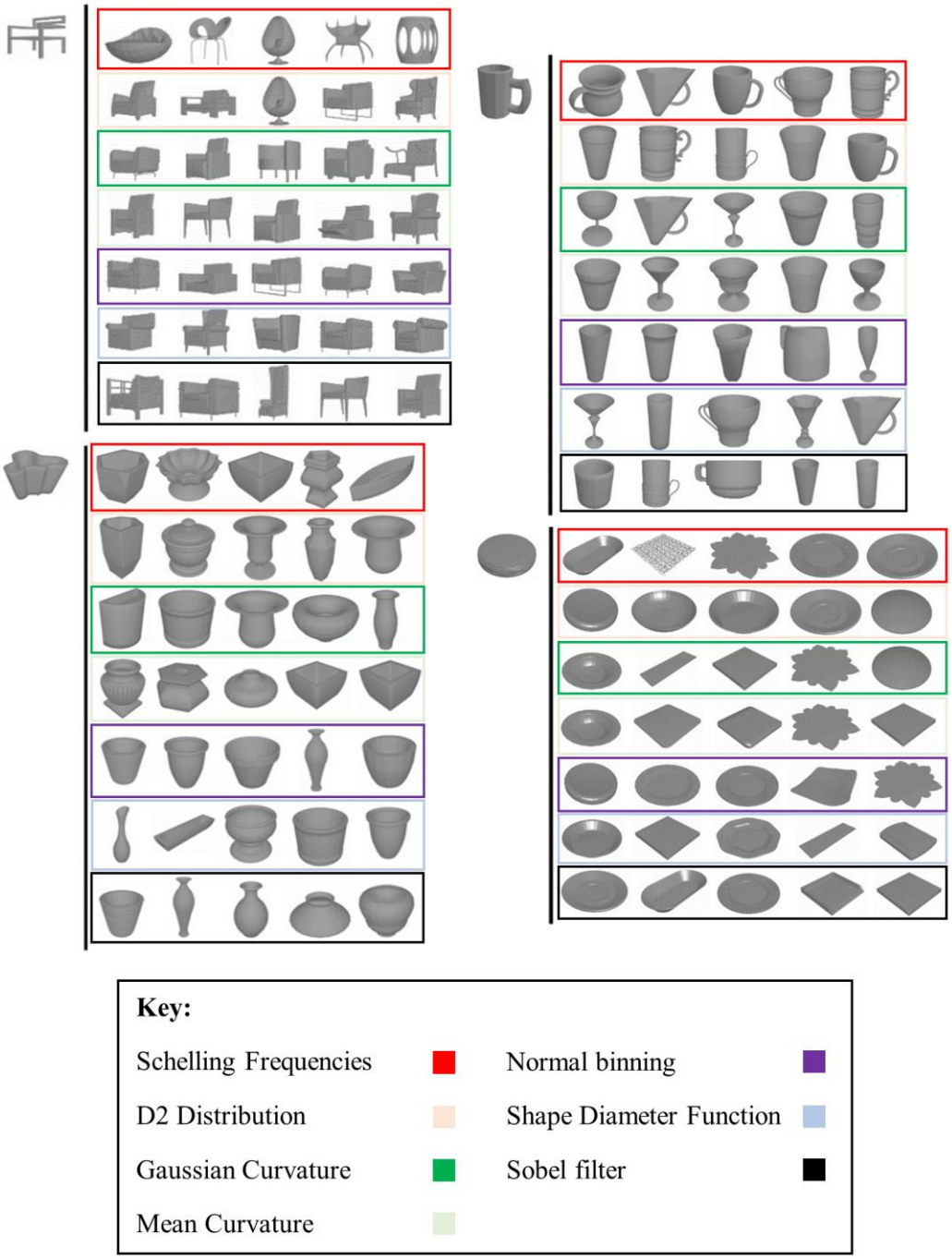


Figure 5.9 – Plots of shapes displayed in rows according to how close they are to a query shape, on the left of each plot. Moving from left-to-right, indicates increased distance from the shape to the query.

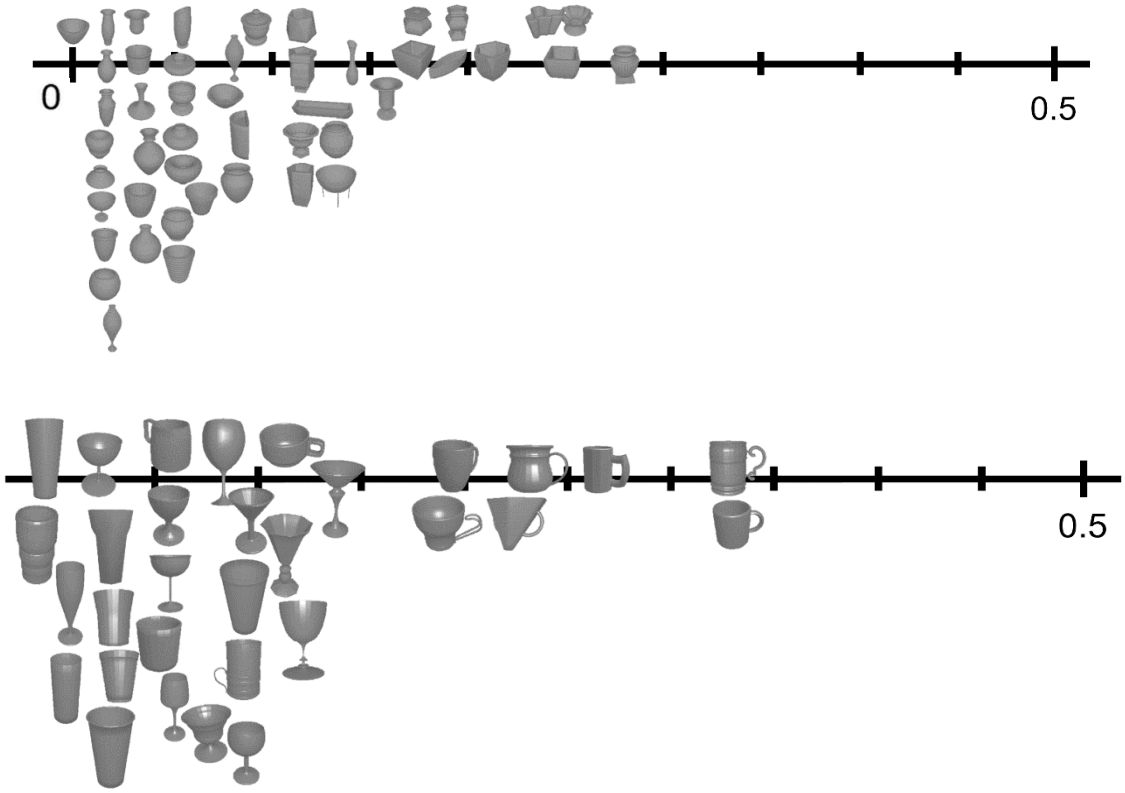


Figure 5.10 - 1-D plots of shapes at their respective participant Schelling frequencies. We show one plot for each of the pots, and cups shape groups.

This is not always the case. For example, with the lamps, some oblong structures which are clearly different to most shapes in their class (curved; oval like structures; with shrouds), do not have high Schelling frequency. But these shapes are not complex.

In a similar manner to the '4-choose-1' case, we also produced t-SNE embeddings of our dataset of 387 shapes, based on the 2nd -to-last output of our convolutional neural network (at layer L6 of the network in section 5.5.1; a 25-dimensional vector), for each shape in a shape class. This output is what is put through a linear combination, to reach a final Schelling frequency prediction. Some example visualisations based on the resulting 2D embeddings, are shown in Figure 5.12 and Figure 5.13.

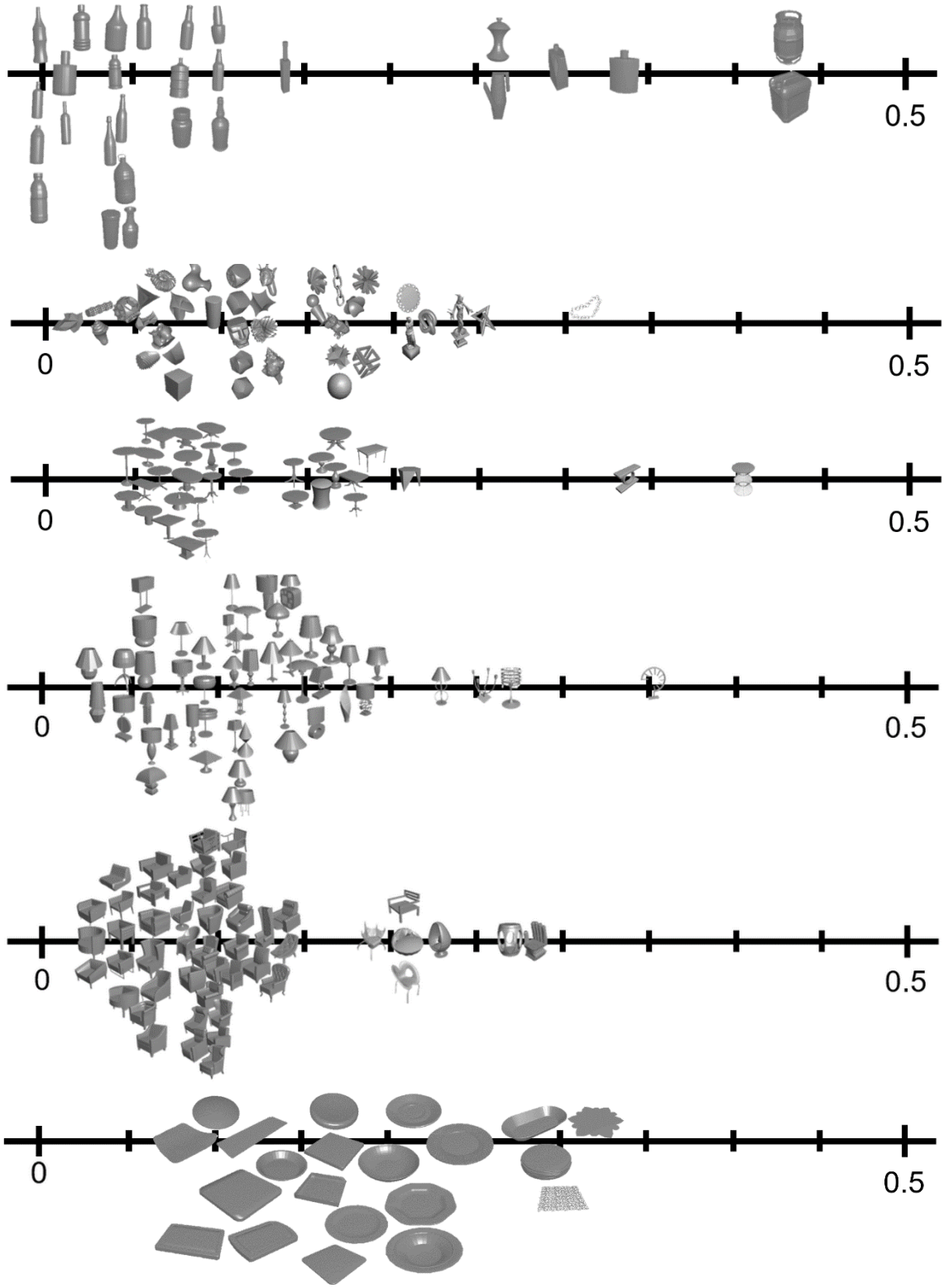


Figure 5.11 - 1-D plots of shapes at their respective participant Schelling frequencies. We show one plot for each of the abstract shapes, tables, lamps, bottles, chairs, and plates shape groups.

You can see that for each plot, the highest Schelling frequency shapes are placed around the boundary of the shape distribution. Figure 5.12 shows this for the plates and tables shape classes. More specifically, Schelling frequency tends to increase with distance/radius from the centre of each plot. In some cases, like the abstract shapes and bottles, distinct clusters appear. Visual patterns also appear when explicitly clustering according to k-means, which we discuss ahead.

5.6.3 Clustering

Visual Patterns

Figure 5.14 provides a visualisation of k-means (k=4) for the chairs, bottles and cups. Figure 5.15 does so for the abstract shapes and plates (k=4). Regarding the abstract shapes, we can see increased shape variation in the top-left and bottom-left clusters, relative to the other clusters of that class. This is reflected in the higher mean Schelling frequency of those clusters. There is also increased detail in the top-right cluster relative to the bottom left. Overall, each cluster has increasing geometric variation according to its mean Schelling frequency (of its constituent shapes). The same trend is visible in the bottles, plates and chairs clusterings, and to a lesser degree, the cups clustering.

Across many shape classes, Gaussian curvature measurements did not provide clear visual differences between clusters. Shapes sometimes varied dramatically in smoothness even within clusters. Mean curvature seemed to dampen this noise and could achieve better results for some shape classes, but this did not result in visual patterns as clear as the Schelling frequencies. Additionally, clusters were unevenly populated. Clustering via the D2 distribution did provide good visual patterns in some cases: such as the abstract shapes, bottles and plates.

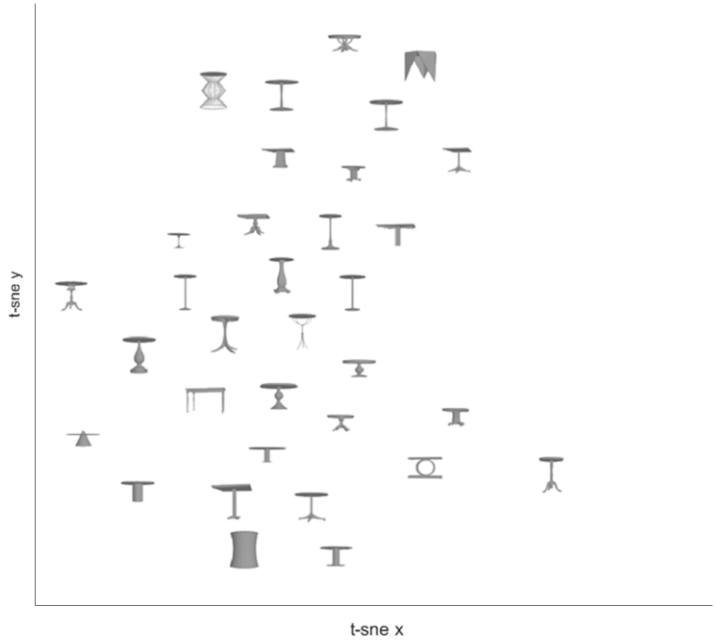
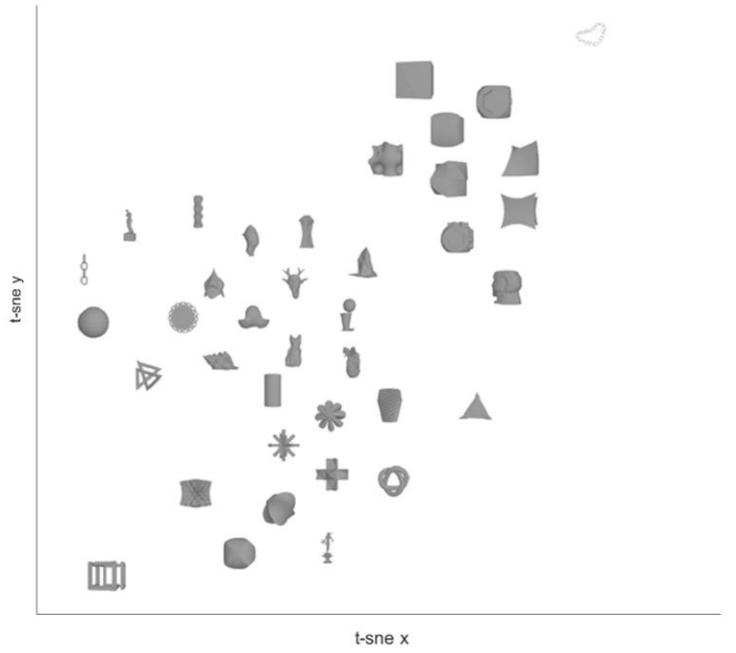


Figure 5.12 - Plots of the plates and tables shapes according to the 2D t-SNE embedding of their neural network outputs from layer n-1 (2nd to last layer), just before they are transformed into a Schelling frequency prediction.



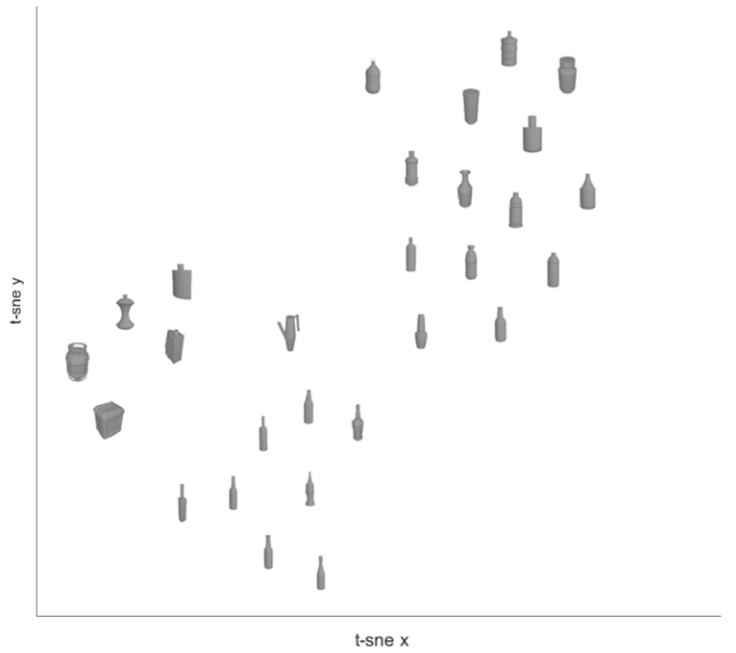


Figure 5.13 - Plots of the abstracts and bottles shapes according to the 2D t-SNE embedding of their neural network outputs from layer n-1 (2nd to last layer), just before they are transformed into a Schelling frequency prediction.

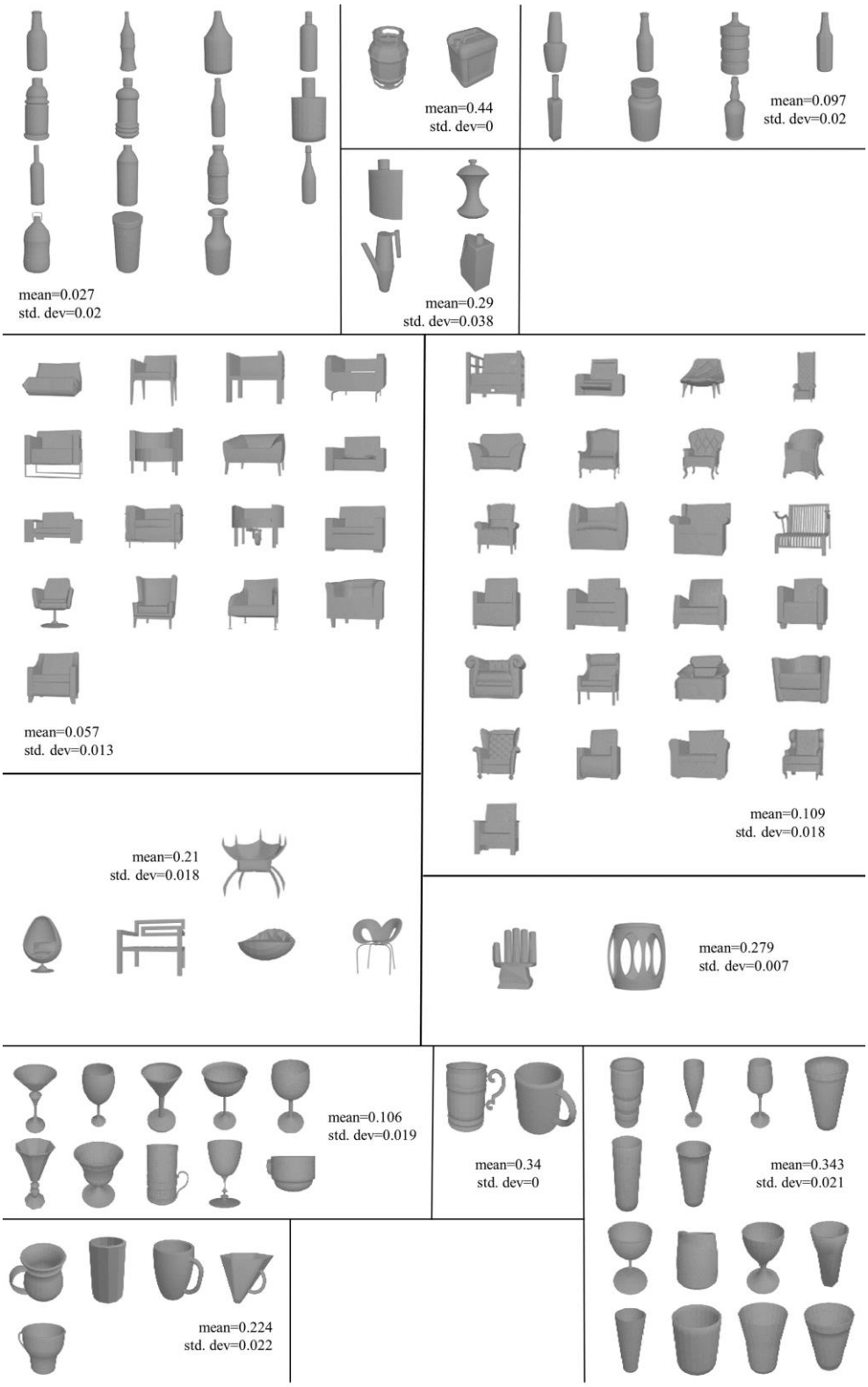


Figure 5.14 – Visualised Schelling frequency based clusterings (k-means) for the bottles, chairs and cups. For each cluster, the mean and standard deviation of the Schelling frequencies of its constituent shapes is displayed.

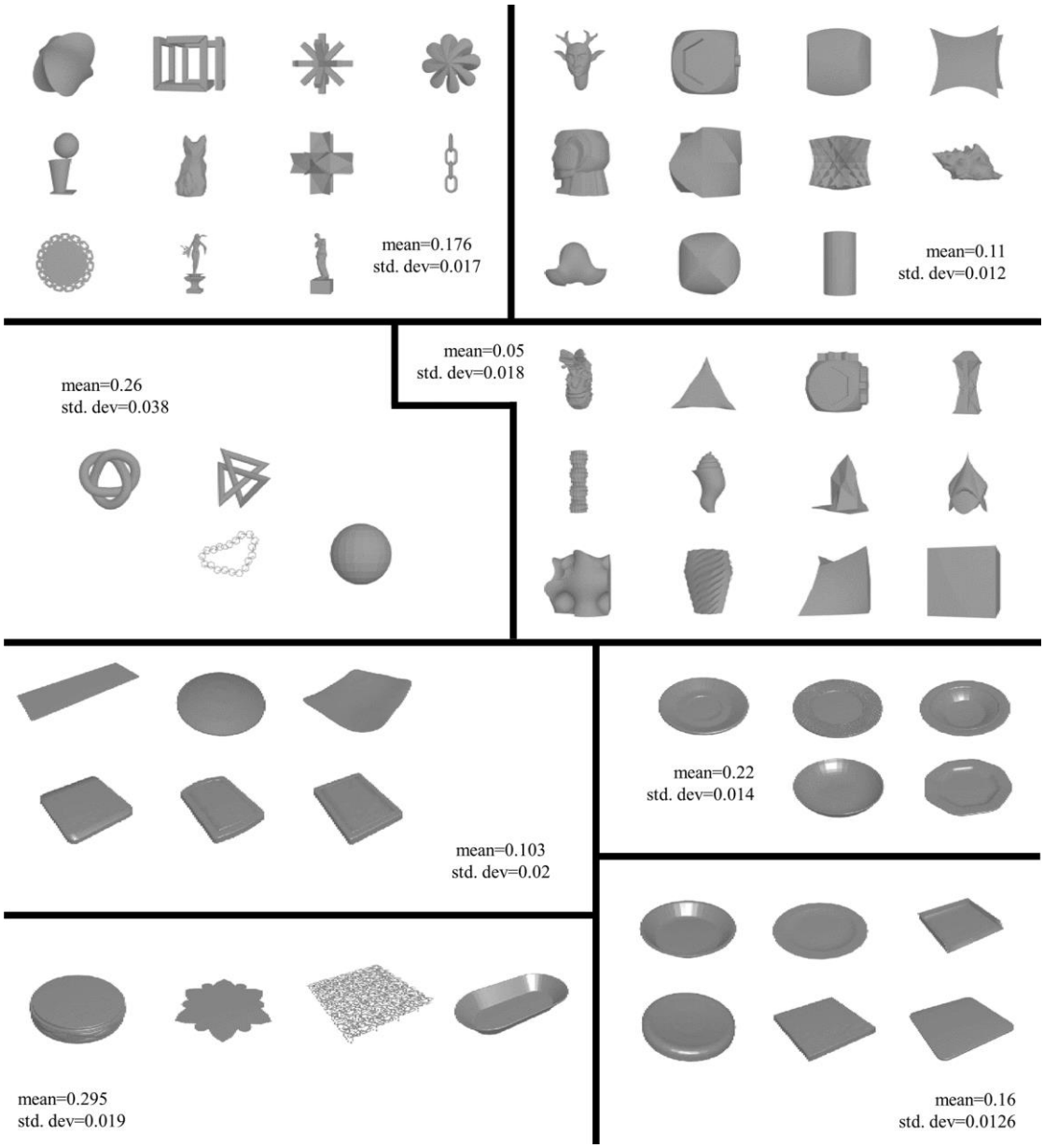


Figure 5.15 – Visualised Schelling frequency based clusterings (k-means) for the abstract shapes and plates. For each cluster, the mean and standard deviation of the Schelling frequencies of its constituent shapes is displayed.

Clustering via Sobel filtering provided good visual patterns for the cups and chairs. Normal-binning rarely provided good clustering results, whether for the simpler classes such as the plates and tables, or more complex classes such as the abstract shapes. Lastly, visual patterns did not appear across most shape classes when clustering according to the Shape Diameter Function (SDF) descriptor. Overall, a single shape descriptor was not sufficient to provide clear visual differences between clusters, across

all shape classes, as was possible via clustering according to Schelling frequencies, or visualisation via 1D shape plots. See Figure 8.2 to Figure 8.4 in the appendix, for visualisations of descriptor-based clusterings.

Statistical Comparison

Similarity Tests

To determine the similarity between clusterings based on Schelling frequencies and those based on shape descriptors, we employed the Adjusted Rand Index (ARI) and Adjusted Mutual Information (AMI) metrics. They each measure the degree of agreement between two partitions of a dataset (such as two clusterings), both taking into account the possibility that cluster assignments occurred due to chance. The ARI is commonly used when it is believed that the underlying or true clustering yields evenly distributed, large clusters. The AMI is used when the underlying clustering is considered to be uneven, where small clusters might exist. As we were uncertain about which method would be most accurate for our data, we computed both measures. For the ARI, a value of 1 indicates the exact same clustering, and values near 0 suggest a uniform random clustering. The AMI results in 0 on average, when the clustering occurred due to chance (so values can be negative), and 1 when the partitions are the same. Our results are shown in Table 5.8 and Table 5.9 (per-class Schelling frequency-based clustering vs. descriptor clusterings). Results which deviated by more than 2 standard deviations from the mean were nearly the same in both cases, apart from the case where all descriptors were combined. These cases tended to be the most positive and were somewhat similar to clustering based on Schelling frequencies. But the degree of similarity was usually low in magnitude (all values were below 0.24) and across disjoint shape classes. This indicates that clusterings based on Schelling frequencies conveyed different information to the tested shape descriptors.

ANOVA Using Likert Scores

We also aimed to test whether shape clusters obtained via Schelling frequencies better encapsulated differences in subjective notions of shape, rather than clusters obtained via shape descriptors. Our earlier collected Likert ratings represented the subjective aspect of this test. We clustered each class of shapes according to their Schelling frequencies, or one of a range of shape descriptors (used earlier in the chapter). These included the D2 distribution; Gaussian curvature, mean curvature, per-vertex normals, per-vertex Shape Diameter Values and Sobel filter values as applied to a 100x100x100 voxelisation of each shape. Each of these were considered *treatments*. We then took the Likert scores corresponding to memorability, 'standing out', uniqueness and visual appeal, and individually assigned the scores to each shape, in each cluster. From there, a one-way ANOVA test was performed for each treatment's clustering and Likert score category combination for a shape class (e.g. abstract shapes: memorability + mean curvature, or lamp shapes: uniqueness + Schelling frequencies).

This was to test whether the means of the Likert scores in each cluster (for each clustering) differed significantly from one another, with p-value < 0.05. That would indicate the degree of utility of a clustering, for separating the shapes according to the criteria of the Likert rating (e.g. visual appeal). We show our results in Table 5.10 to Table 5.13.

Apart from the case of 'uniqueness', Schelling frequencies yielded more, or equal numbers of differences in cluster Likert score means, across classes, than the shape descriptors. When equal, Schelling frequencies tended to yield differences in disjoint shape classes to the shape descriptors. Among the shape descriptors, mean curvature yielded differences in memorability, most (but less than the Schelling frequencies).

Schelling Frequencies	All Descriptors	D2	Gaussian Curvature	Mean Curvature	Normals	Shape Diameter Function	Sobel Filter
Abstract	-0.054	0.002	-0.046	0.017	0.033	-0.053	0.058
Baskets	-0.035	-0.004	-0.032	0.139	-0.033	0.002	-0.02
Bottles	0.088	-0.001	0.055	-0.04	0.001	-0.013	0.071
Cabinets-Shelves	-0.056	0.005	-0.064	-0.051	0.004	-0.024	0.029
Club Chairs	-0.02	-0.015	-0.024	-0.005	0.045	-0.037	0.08
Cups	-0.051	0.106	0.16	0.023	0.168	-0.049	0.043
Lamps	-0.039	0.058	-0.047	0.001	-0.025	-0.003	-0.01
Tables	-0.054	0.088	0.036	0.187	0.021	-0.064	0.149
Plants	0.025	0.236	0.013	0.008	0.029	0.122	-0.021
Plates	0.037	0.138	-0.005	0.007	-0.007	0.004	0.207
Pots	0.033	-0.008	0.01	0.003	0.026	0.007	0.059

Table 5.8 – Adjusted Mutual Information values based on pairing a clustering derived from Schelling frequencies with a clustering based on each shape descriptor (values $> 2\sigma$ away from the mean are in green).

Schelling Frequencies	All Descriptors	D2	Gaussian Curvature	Mean Curvature	Normals	Shape Diameter Function	Sobel Filter
Abstract	-0.018	-0.026	-0.011	-0.003	0.013	-0.004	-0.006
Baskets	-0.053	-0.03	0	0.104	-0.026	-0.034	-0.006
Bottles	-0.041	-0.005	0.024	0.022	0.057	-0.074	0.117
Cabinets-Shelves	-0.035	-0.013	-0.037	-0.047	-0.005	-0.009	-0.009
Club Chairs	0	0.01	-0.018	-0.027	0.023	-0.043	0.031
Cups	-0.035	0.072	0.169	-0.001	0.212	-0.002	0.043
Lamps	-0.018	0.029	-0.034	-0.015	0.008	0.01	0.011
Tables	-0.044	0.056	0.052	0.197	0.038	-0.034	0.156
Plants	-0.005	0.228	0.005	-0.021	0.114	0.06	-0.004
Plates	0	0.109	-0.016	0.008	-0.034	0.007	0.204
Pots	0.012	0.014	0.014	0.001	0.05	-0.016	0.055

Table 5.9 – Adjusted Rand Index values based on pairing a clustering derived from Schelling frequencies with a clustering based on each shape descriptor (values $> 2\sigma$ away from the mean are in green).

Normals did so for ‘standing out’ (equivalent to the Schelling frequencies), and Sobel filter values did so for visual appeal (less than the Schelling frequencies). Regarding the classes where there was significance given the normals-based clustering, there were only 2 cases of overlap with the Schelling frequency-based clustering (the chairs and cups classes of shape). The total number of significant cases for the normals was 5, whereas for the Schelling frequencies it was 4.

Excluding uniqueness, Schelling frequencies yielded differing Likert score means across clusters more often or equally as often across shape classes, compared to the tested shape descriptors. For the Schelling frequencies, memorability yielded the largest frequency of differences. This was greater than the frequencies of each of the shape descriptors, indicating that Schelling frequencies may be preferred when clustering shapes according to perception of their memorability.

ANOVA Using Schelling Frequencies

By clustering each class of shapes in the same manner as above, for each shape descriptor (best of 10 iterations of k-means; k=4), we tried to determine whether shape descriptors could be used to cluster shapes in a way that significantly introduces between cluster differences in terms of their Schelling frequencies. We performed a one-way ANOVA test between the Schelling frequency distributions of clusters obtained per shape class, to determine if there were significant differences between the means of the cluster Schelling frequency distributions. As a sanity test, we found that $p << 0.0001$ when clustering via per-class Schelling frequencies, respectively. Therefore, we considered k-means as a valid option for clustering via the shape descriptors. Overall, we found that five descriptor-based clusterings yielded significant differences in mean Schelling frequencies, but for only 2 classes out of 11 at most. Like the

clustering similarity tests, this suggests that Schelling frequencies contain some information orthogonal to that of the shape descriptors. See Table 5.14 for our results.

5.7 Discussion

To begin, we restate our hypotheses below:

1. A shape which stands out from others or is considered unique, is more Schelling frequent.
2. A more visually appealing shape is more Schelling frequent.
3. A shape may be perceived as more memorable relative to others, as its Schelling frequency increases.
4. Schelling frequencies convey different information to that of shape descriptors.

We found that 8 out of 11 shape classes gave positive correlations between their shape Schelling frequencies and a notion of 'standing out', with 5 out of 11 doing so for a notion of 'uniqueness'. Looking at these results, these notions are likely factors behind Schelling frequencies, with the difference in shape class totals potentially being due to people not agreeing with the definitions of the terms 'stand out' and 'unique'. Due to this, hypothesis #1 may be generally correct.

We found that 5 out of 11 shape classes gave positive correlations between their shape Schelling frequencies and a notion of 'visual appeal', indicating the hypothesis #2 is correct in some cases, but may not generally be so.

We consider hypothesis #3 to likely be correct, as 10 out of 11 shape classes gave positive correlations between their shape Schelling frequencies and a notion of 'memorability'. But, previous results on image memorability [259, 260] have suggested

a counter-intuitive result, that perception of image memorability is inversely correlated with the actual memorability of images, although at a small magnitude. It was found that human estimates of image memorability were negatively correlated (corr. = −0.19) with the true memorability of images. So, given that we showed 3D shapes to participants in the form of animated images, it may be possible that the higher the Schelling frequency of a shape is, the more difficult it is to truly memorise.

We found that when shapes are clustered according to k-means (k=4) via their Schelling frequencies, significant differences are exhibited between mean memorability scores of shape clusters, more often, than clusters obtained via the tested shape descriptors (similar to Likert correlation results). Apart from the case of ‘uniqueness’, Schelling frequencies yielded more, or equal numbers of differences in cluster Likert score means, across classes, than the tested shape descriptors. But even in that case, there was only partial overlap in the results. Also, through comparing each partitioning via the Adjusted Rand Index and Adjusted Mutual Information measures, we see little to no alignment between clusterings based on Schelling frequencies vs. the tested shape descriptors. Visually, we can see some differences between clusters, as well. Additionally, visualising per-shape descriptor values as an intensity heatmap across each shape class, shows no visual patterns when the descriptors are displayed in order of increasing Schelling frequency. These results provide evidence that hypothesis #4 is correct.

Memorability	Schelling Frequencies	All Descriptors	D2	Gaussian Curvature	Mean Curvature	Normals	Shape Diameter Function	Sobel Filter
Abstract	**p < 0.01**	p >= 0.05	p >= 0.05	**p < 0.05**	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05
Baskets	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Bottles	**p < 0.01**	p >= 0.05	p >= 0.05	**p < 0.01**	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05
Cabinets-Shelves	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05
Chairs	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	**p < 0.01**	p >= 0.05	**p < 0.01**
Cups	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05
Lamps	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Tables	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05	p >= 0.05
Plants	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05
Plates	**p < 0.01**	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Pots	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05

Table 5.10 – One-way ANOVA test results for significant differences in mean memorability Likert scores of clusters obtained via k-means (k=4), across all shape classes.

Standing out	Schelling Frequencies	All Descriptors	D2	Gaussian Curvature	Mean Curvature	Normals	Shape Diameter Function	Sobel Filter
Abstract	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Baskets	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Bottles	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Cabinets-Shelves	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Chairs	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05	**p < 0.01**
Cups	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05
Lamps	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Tables	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05	p >= 0.05
Plants	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Plates	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Pots	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05

Table 5.11 – One-way ANOVA test results for significant differences in mean 'standing out' Likert scores of clusters obtained via k-means (k=4), across all shape classes.

Uniqueness	Schelling Frequencies	All Descriptors	D2	Gaussian Curvature	Mean Curvature	Normals	Shape Diameter Function	Sobel Filter
Abstract	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05	p >= 0.05
Baskets	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Bottles	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05
Cabinets-Shelves	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	**p < 0.05**	p >= 0.05
Chairs	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05	p >= 0.05
Cups	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05
Lamps	**p < 0.05**	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Tables	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	**p < 0.01**	p >= 0.05	p >= 0.05
Plants	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**
Plates	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Pots	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05	p >= 0.05

Table 5.12 – One-way ANOVA test results for significant differences in mean uniqueness Likert scores of clusters obtained via k-means (k=4), across all shape classes.

Visual Appeal	Schelling Frequencies	All Descriptors	D2	Gaussian Curvature	Mean Curvature	Normals	Shape Diameter Function	Sobel Filter
Abstract	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Baskets	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Bottles	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Cabinets-Shelves	p >= 0.05	**p < 0.05**	**p < 0.05**	p >= 0.05	**p < 0.05**	p >= 0.05	**p < 0.01**	p >= 0.05
Chairs	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Cups	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05	**p < 0.05**
Lamps	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Tables	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Plants	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Plates	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**
Pots	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	**p < 0.01**	p >= 0.05	**p < 0.01**

Table 5.13 – One-way ANOVA test results for significant differences in mean visual appeal Likert scores of clusters obtained via k-means (k=4), across all shape classes.

Schelling Frequencies vs. Shape Descriptors	All	D2	Gaussian Curvature	Mean Curvature	Normals	Shape Diameter Function	Sobel Filter
Abstract	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Baskets	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Bottles	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Cabinets-Shelves	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Chairs	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05	**p < 0.01**
Cups	p >= 0.05	**p < 0.01**	p >= 0.05	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05
Lamps	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Tables	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	**p < 0.05**	p >= 0.05	p >= 0.05
Plants	p >= 0.05	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Plates	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Pots	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05

Table 5.14 – One-way ANOVA test results for significant differences in mean Schelling frequency of clusters obtained via k-means (k=4) for various shape descriptors, across all shape classes.

It is possible to search for shapes that have complementary or contrasting Schelling frequency to a query shape. We have produced plots, representing a search response given the top k=5 closest shapes, with respect to Schelling frequency and each tested shape descriptor (e.g. D2 distribution, SDF values, Gaussian curvature). Search query response shapes based on Schelling frequencies tend to be more prominent or detailed, compared to those obtained via the tested shape descriptors.

Geometric shape descriptors such as curvatures (mean, Gaussian), per-vertex normals, Sobel filters, the D2 distribution, and per-vertex SDF values consistently do not correlate well, either individually or as a group, with shapes ordered by Schelling frequency. Even without a linear correlation relationship between Schelling frequencies, a combination of 10s or 100s of shape descriptors might provide better prediction results than those in this thesis, possibly with shape dataset sizes near our current amount (387), but it would require more selection and testing of shape descriptor combinations (what is a descriptor’s purpose; what are the dimensions of each descriptor?). Testing combinations of descriptors could quickly become impractical as more shapes are added to the dataset, relative to testing different numbers of depth image samples per shape, and different depth image resolutions.

5.8 Conclusion

Shapes with higher Schelling frequency, or *Schelling meshes*, are more geometrically varied. Schelling meshes can be visually appealing but that is not the most important factor in understanding them. Schelling meshes tend to be those that people consider more prominent and stand out with respect to other shapes in a dataset. They are perceived as memorable, relative to the remainder of their class. This suggests that they can represent a dataset’s extremes. More balance can be achieved via sampling some

shapes from the low and mid Schelling frequency ranges/intervals. We find that Schelling frequencies best distinguish differences between shapes which have more extreme frequencies.

Using the 'Many-Within-Class' methodology, we have found that Schelling frequencies can be obtained via: 1) repeated integration of results from showing people smaller datasets within a larger group/class – e.g. 12 out of 30 shapes total, or 2) results obtained when showing people all shapes within a class. Under case 2), Schelling frequencies can be obtained with approximately 35 people, but ideally 50 or more. Additionally, results can be updated as new shapes are added to each class, over time. However, case 1) requires further testing with additional shape classes, to be a general conclusion.

Schelling frequencies can be learned via depth-image/multiple-view shape representations provided to a visual field-like regression approach, such a convolutional neural network. For new shape classes, a classifier could be used to pre-classify an unknown shape before providing it to a neural network to predict its Schelling frequency (decided via exact match, or synonyms of the class word/category).

At this point, our study of human-interpretations of shape had been limited to 3D shapes. We began to question whether similar approaches could be undertaken to better understand 2D shapes. This leads us to the next chapter, which focuses on a specific category of 2D shapes: fonts.

6 Font Specificity

6.1 Introduction

Inspired by the *Image Specificity* [7] work, we focused on applying the concept of *Specificity* to 2D fonts. We provide two interpretations of Specificity, by asking people to describe fonts using words. One is based on word frequency, and another is an automated method based on pre-trained word embeddings that reflect word co-occurrence probabilities.

We firstly collected font Specificity data by asking people to describe fonts using words. The consistency of these descriptions was represented via the distribution of word frequencies associated with each font. Secondly, we determined consistency via the closeness of word embeddings, where the word embeddings represented word co-occurrence probabilities within a corpus of text. We determined Specificity scores via these approaches. We explored the question of "what makes a font Specific?", via our collected data. We show that Specificity can be learned and used for prediction of Specificity scores using a colour image-based convolutional neural network. We compared a selection of traditional image-based shape descriptors to deep-learning approaches for prediction of Specificity scores, and achieved better prediction accuracy in most cases, using a deep-learning approach. Results are shown for a range of 2D fonts and we demonstrate that Specificity is a useful concept for the applications of Specificity-guided visualisation, clustering, and search.

We have found that fonts of low Specificity score start out as a mixture of many characteristics (texture, curvature, thickness), with thin and italic fonts appearing as

Specificity increases, then mostly bold fonts appearing as Specificity is highest. Highest Specificity fonts tend to have only one or two clear aspects to them – i.e. bold; bold and italic; italic and thin etc.

In addition, Specific fonts tend to be perceived as more memorable (corr.=0.2774, $p<0.05$); they are considered to be more normal (corr.=0.3651, $p<0.05$) and visually-appealing (corr.=0.3471, $p<0.05$), more legible (corr.=0.4676, $p<0.05$) and less creative (corr.=-0.5174, $p<0.05$). For our automated font Specificity scores, the number of unique words provided by participants, per font, reduced as Specificity score increased (corr.=-0.5601, $p << 0.01$). This was also true for our approach based on word frequency (corr.=-0.7544, $p << 0.01$).

6.2 Hypotheses

We aimed to determine whether Specific fonts could be described via unique attributes or properties, such as legibility or creativity. Below is a list of our tested hypotheses:

1. A more legible font is more Specific.
2. A more creative font is less Specific and less legible.
3. Visually appealing fonts are more Specific but potentially not the most Specific fonts.
4. Specificity conveys information different to that of existing image descriptors.

6.3 Methodology

6.3.1 Data Collection

We collected human descriptions of 100 individual fonts in the form of words, from 111 participants (approx. 19-24 people per 20 fonts, avg.=22.4 people per 20 fonts), via the *Amazon Mechanical Turk* platform. Fonts were collected from *fontlibrary.org*. We

paid participants $0.10 per HIT. An example survey is shown in Figure 8.5 (see appendix).

For each question, participants were asked to describe an image of a font, using any words that came to mind. Each image showed the letters A-Z and the numbers 0-9, in a different font style. Some text validation was employed to ensure that participants provided only words, separated by commas.

Pre-processing of User Provided Words

All words provided by participants, were checked for spelling errors and corrected. Otherwise, grammatically incorrect words were removed. Plural words were substituted with their singular form. Words concatenated together were split into individual words. Stop-words, or highly common words (e.g. ‘the’, ‘at’ and ‘on’) were removed, in addition to punctuation. Words of length less than 3 were ignored, and remaining words were lemmatised (group inflected word forms were treated as a single entity – e.g. ‘jumped’ and ‘jumping’ are replaced with ‘jump’).

Participant and word-level statistics

7443 words were provided across all users. The average number of words provided per participant was 67.05, with a standard deviation of 27.96 across all 100 fonts, and median word count of 65. Additionally, the minimum amount of provided words by a participant was 13, and the maximum was 150, showing a large absolute deviation in word frequency among participants.

In Figure 6.1, we provide some plots showing 1) mean per-participant word frequencies across all fonts, and 2) the estimated PDF of mean per-font word frequencies, across all participants. The mode is approximately 4 words per font, on average.

Figure 6.2 shows a plot of the proportions of words provided only once across all participants, for each font. The distribution is mostly random, but nearly all values lie within a confined range of approximately 0.6 to 0.87, or 60-87% of words, per font.

On average, words were provided once approximately 73.7% of the time. See Table 6.1 for other statistics. Here, we can see that median is close to the mean, indicating that the percentage of unique words provided was consistent across participants (suggesting that there were few outliers).

Percentage of words provided once	**100 Fonts**
Min	53.66%
Max	90.77%
Mean	73.69%
Median	74.36%
Std. Dev.	68.1%

Table 6.1 – Statistics of the percentage of words provided only once, by participants.

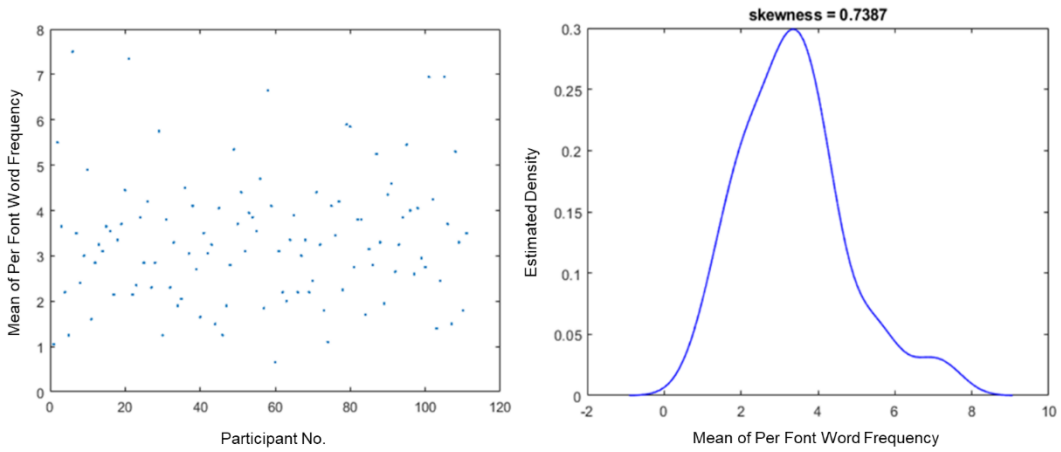


Figure 6.1 – Plots of per-font word frequency statistics across participants. (Left) means (Right) estimated PDF of per-font word frequency means.

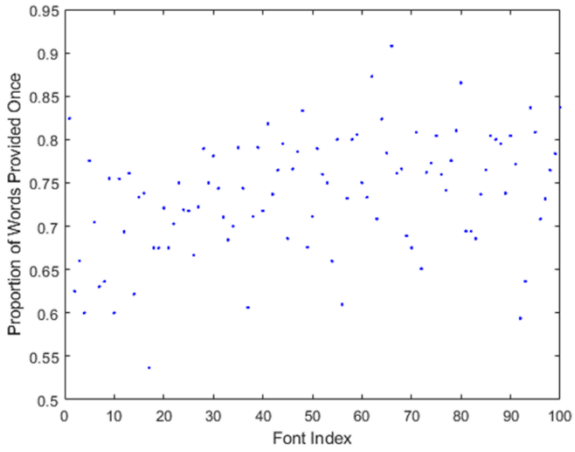


Figure 6.2 – A plot showing the proportion of words provided once for each font, across all 111 participants.

6.4 Analysis

6.4.1 Top-50 words across all fonts

See Figure 6.3 for a plot of the top-50 words among the 100 fonts, and their frequencies. We can see that simpler, geometry-oriented words tend to have higher frequency (bold, italic, thin, thick, thin, slant, narrow etc). In some cases, more subjective terms appear (classic, simple, elegant). As word frequency decreases, the proportion of words related to subjectivity/emotion increases.

6.4.2 Determining Word Categories via Wordnet Synsets

We also categorised participant-provided words by grouping words of similar meaning. For this purpose, we obtained the synonyms of each font's words using *Wordnet* [193], a lexical database of English, which groups words of similar concepts as *synsets*. These are connected via "conceptual-semantic and lexical relations" [193].

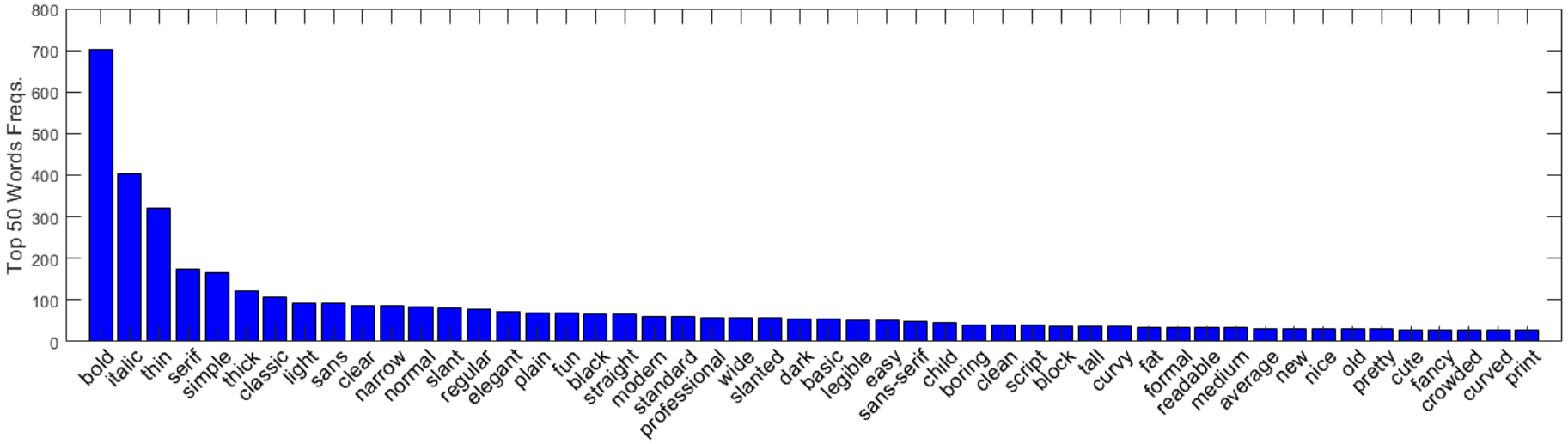


Figure 6.3 – Word frequency plot of the top-50 most frequent words across all fonts.

Each English word in the database (noun, verb, adjective etc.), is associated with a 'synset', or a set of synonym words. Participant provided words were grouped according to the synsets that they shared (at least one of the word's synsets was part of an existing group of synsets). Across 100 fonts, we found 693 unique groups of words, with 191 groups that contained more than one word.

Using the words of these 191 groups, we categorised each of the fonts manually according to more general categories of our own, including words related to 'geometry', words related to 'subjectivity and emotion', and words hinting at 'abstraction'.

Table 8.1 in the appendix provides the words that were associated with each category. Words grouped in brackets were considered to have the same meaning (via their shared synsets), and if associated with a font, were treated as an equivalent word occurrence for the purposes of statistical calculations including: word frequency (either per font, or conditional on a category) and part-of-speech tagging.

There was a large amount of word variation per category even within the reduced amount of 191 groups of words, which we thought was sufficient enough to understand which types of words are associated with Specific fonts, and the meanings of those words. Figure 6.4 shows plots of frequencies of words and word groups found in each category.

6.4.3 Types of Words According to Category?

Across all fonts, we tagged each of their associated words using the Stanford Part-Of-Speech tagger [266], to determine the most common type of words associated with them, and also to check whether the words provided, were mostly descriptive. We additionally tagged words conditionally on a word categorisation such as: geometry-

related terms, subjectivity/emotion-related terms and 'abstraction'-related terms, to determine if word meaning induced any differences.

Part-of-Speech

Across all fonts, we determined the types of words provided by participants, via part-of-speech tags. Each font was predominantly described with adjectives (approx. 60% of the time) and nouns (approx. 20% of the time). Verb/Adverb terms were provided rarely, approx. 2% of the time.

Word Category	Unique Tag Count	Most Frequent Tag	2nd Most Frequent	3rd Most Frequent	4th Most Frequent
All Words	16	Adjective (0.63)	Singular Noun (0.24)	Plural Noun (0.037)	Past-Tense Verb (0.027)
Geometric	11	Adjective (0.59)	Singular Noun (0.2)	Plural Noun (0.06)	Past-Tense Verb (0.04)
Subjective/Emotion	9	Adjective (0.81)	Singular Noun (0.15)	Adverb (0.016)	Past-Participle Verb (0.007)
Abstraction	12	Adjective (0.65)	Singular Noun (0.296)	Plural Noun (0.024)	Present-Tense Verb (0.015)

Table 6.2 – Proportions of Part-Of-Speech associated with word categories.

We can see from Table 6.2 that adjectives and singular nouns made up most of the words provided under each category, matching the overall trend for the whole set of 100 fonts. This was expected, if participants completed surveys to a good standard. But the ratio of adjectives to singular nouns changed with category. Subjective words tended to consist mostly of adjectives (over 80%), with a smaller amount of singular noun terms than abstraction-related words and geometry-related words (approx. 15%). The number of unique part-of-speech tags reduced for subjective words also. Across categories, the fraction of foreign/unknown words was 0.007 or under 1%. As you will see later, the frequency of unique words across fonts, is negatively correlated with font Specificity, possibly suggesting that Specific fonts are described with more subjective words than geometry-related/abstraction words.

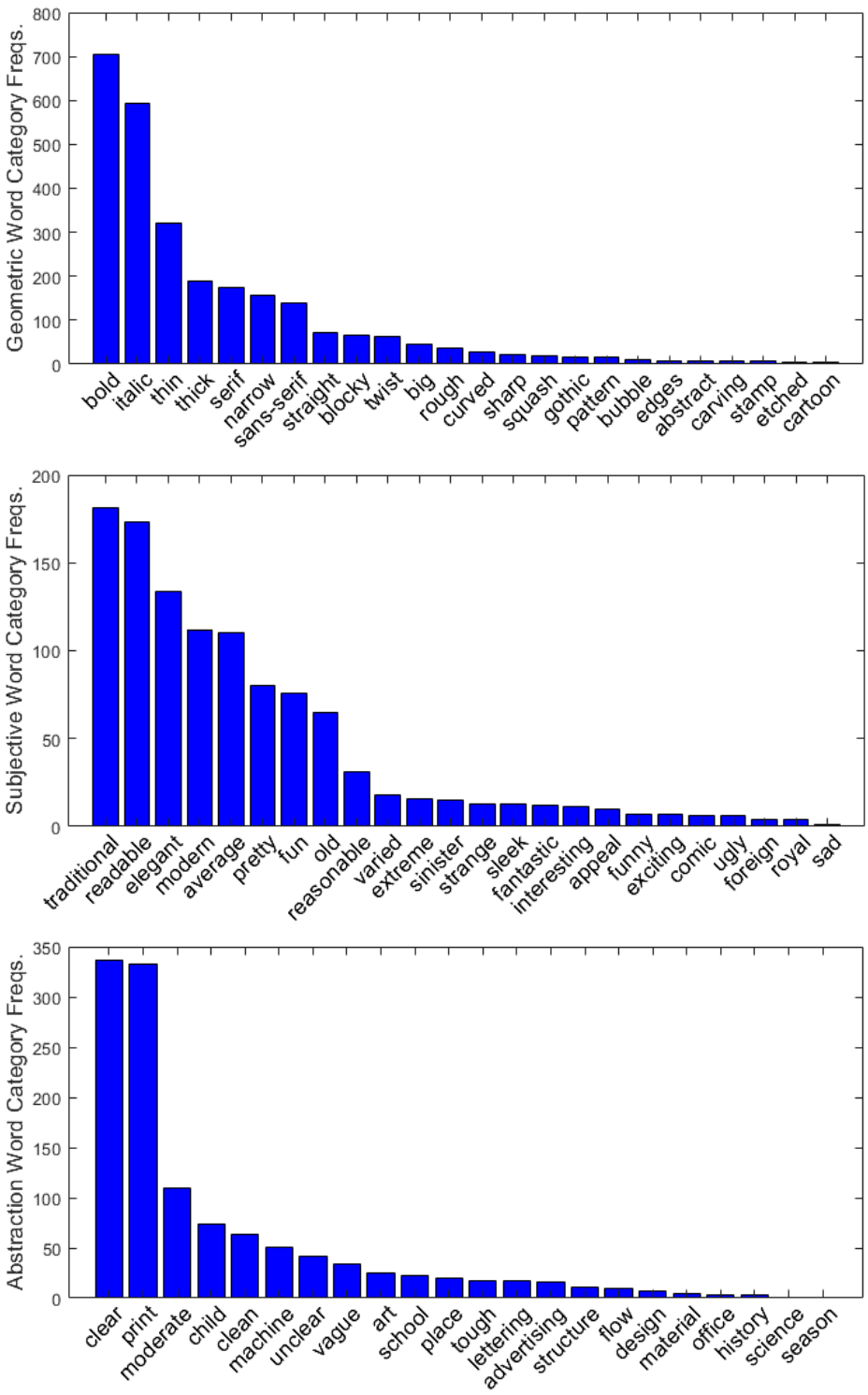


Figure 6.4 – Plots of word and word group frequency, per category. Word groups are represented in the plot by the first word in their group.

Word Category	Bold	Italic	Etched	Sans-Serif	Narrow	Child-like	Elegant
All Words	Adj. 0.63	Adj. 0.62	Adj. 0.511	Adj. 0.61	Adj. 0.63	Adj. 0.594	Adj. 0.64

Table 6.3 – Most frequent PoS tags of seven of the most frequent word categories.

Word Category	Fun	Modern	Pretty	Legible	Moderate	Education
All Words	Adj. 0.65	Adj. 0.63	Adj. 0.66	Adj. 0.65	Adj. 0.63	Adj. 0.59

Table 6.4 – Most frequent PoS tags of six of the most frequent word categories.

In Table 6.3 and Table 6.4, we see a trend of participants mostly providing adjectives in their responses, with similar proportions across words, except in the 'Etched' case, but that category also had a lower absolute number of parts-of-speech associated with it, likely due to it being a more specialised word category.

For determining a measure of font Specificity, we began to focus on the consistency of font descriptions. In this sense, the target data was the distribution of descriptions, or a distribution based on word frequency, per font. From these, we determined a per-font *Specificity* score, which increased as variation in the words used to describe a font, reduced.

6.4.4 Rényi Specificity

Definition

Given a set of participant-provided words, w_i, for each font, f, we counted the occurrence frequency of each word (after being passed through a stemming/lemmatising process, such that words with the same root were treated as the same). These frequencies, f_i, were normalized to obtain a set of probabilities, p_i, for each f. These p_i represent a probability distribution over f. Entropy is a measure of uncertainty in a distribution, so we consider Specificity to be in opposition to entropy.

Therefore, we define a font's Specificity score to be the inverse of the entropy of the word frequency distribution for that font.

Rényi entropy [267] is:

$$H_\alpha(p_i) = \frac{1}{1-\alpha}\log_2\left(\sum_i {p_i}^\alpha\right)$$

Equation 6.1 – Rényi entropy formulation

We take its inverse as:

$2^{-\frac{H_\alpha}{c}}$, where c is a parameter to be chosen.

Shannon entropy [268] is a special case of Equation 6.1, if $\alpha = 1$:

$$H_1(p_i) = -\sum_i p_i \log_2(p_i)$$

Its inverse/Specificity is:

$$S_1(p_i) = \prod_i (p_i)^{\frac{p_i}{c}}$$

Equation 6.2 – Rényi Specificity ($\alpha = 1$)

If $\alpha = 2$:

$$H_2(p_i) = -\log_2\left(\sum_i {p_i}^2\right)$$

Specificity in this case, is:

$$S_2(p_i) = \left[\sum_i {p_i}^2\right]^{-\frac{1}{c}}$$

Equation 6.3 – Rényi Specificity ($\alpha = 2$)

Equation 6.2 and Equation 6.3 capture an intuitive notion of Specificity, in that:

1. The larger the number of descriptions, the more varied they are, and this leads to a higher entropy and smaller Specificity.
2. The more uniform the distribution, the descriptions become more equally likely, and this leads to a higher entropy and smaller Specificity.

For any future references to Rényi Specificity, we use $\alpha = 2$ (Equation 6.3).

We chose to use Rényi entropy at $\alpha = 2$, since the resulting Specificity formula is similar to existing diversity indices used in Physics and Ecology (such as the diversity index derived from the *Simpson concentration* or *Gini-Simpson* index) [269]. Diversity indices account for the number of elements associated with a concept, rather than just the uncertainty in describing the concept. They aim to show equivalence between concepts (*communities*, fonts) consisting of the same base elements – e.g. (*species*, words). Many diversity indices are monotonic functions of $\sum_{i=1}^{S} p_i^q$ (given S species) [269]. Values of $q < 1$ favour rarer species, and values of $q > 1$ favour the most common species. When $q = 0$, the index is independent of species frequency (e.g. independent of word frequency), as $p_i^0 \equiv 1$ (for non-zero probabilities). When $q = 1$, species are only weighted by their frequencies. With $\alpha = 2$ (similar to $q = 2$), we can imagine that when a font is mostly associated with a few, highly frequent/common words, its Specificity should be high.

Visualising Rényi Specificity for Fonts

We can visualise Rényi font Specificity by plotting font images on a 1D plot, according to their scores. To ensure that each image is clearly visible, we show the top-10 and bottom-10 fonts according to Rényi Specificity score, in Figure 6.5. In this figure, we additionally provide a plot consisting of one font sampled from a 10-bin histogram, according to increasing Rényi Specificity score.

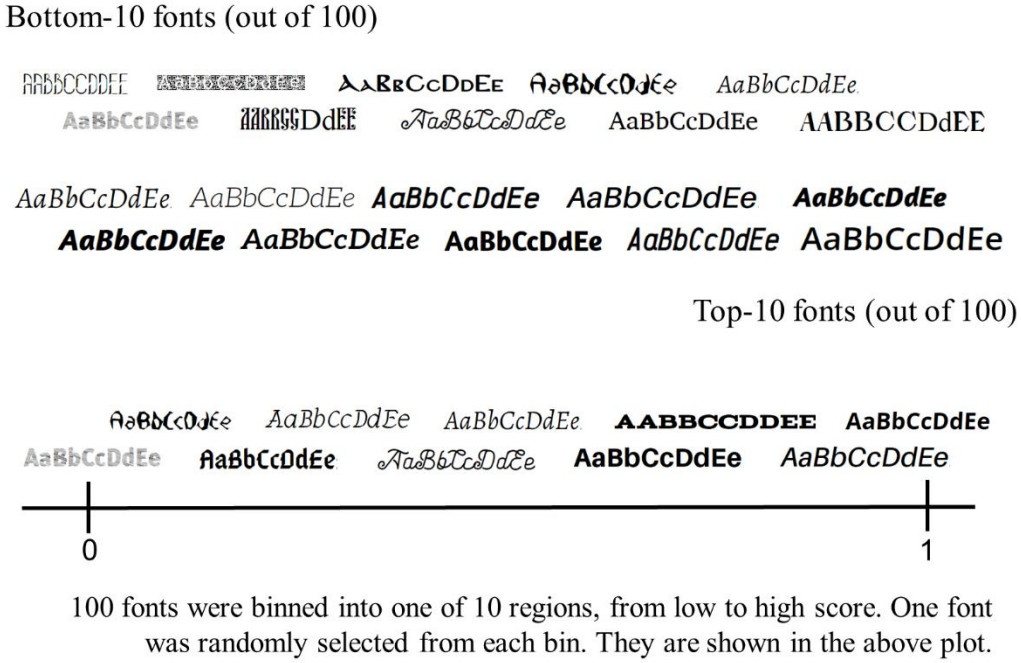


Figure 6.5 – Visualisations of Rényi Specificity. (Top) Plot of the bottom-10 and top-10 fonts according to their Rényi Specificity scores. (Bottom) Plot of a sample of fonts randomly taken from 10 equal Specificity score intervals.

Fonts seem to start out as a mixture of many characteristics (texture, curvature, thickness) if they have low scores, with thinner fonts appearing as score increases (around the mid-score range), in addition to italic fonts, and then finally more bold fonts appearing as Rényi Specificity is highest. We can also see that fonts with high Specificity tend to have only one or two clearly defined features to them – i.e. they are *bold*, *bold+italic*, *thin+italic* etc. This reflects the trend that the number of unique words

per font strongly reduces as Rényi Specificity increases (corr.=-0.7544, $p << 0.05$), as shown in Figure 6.6.

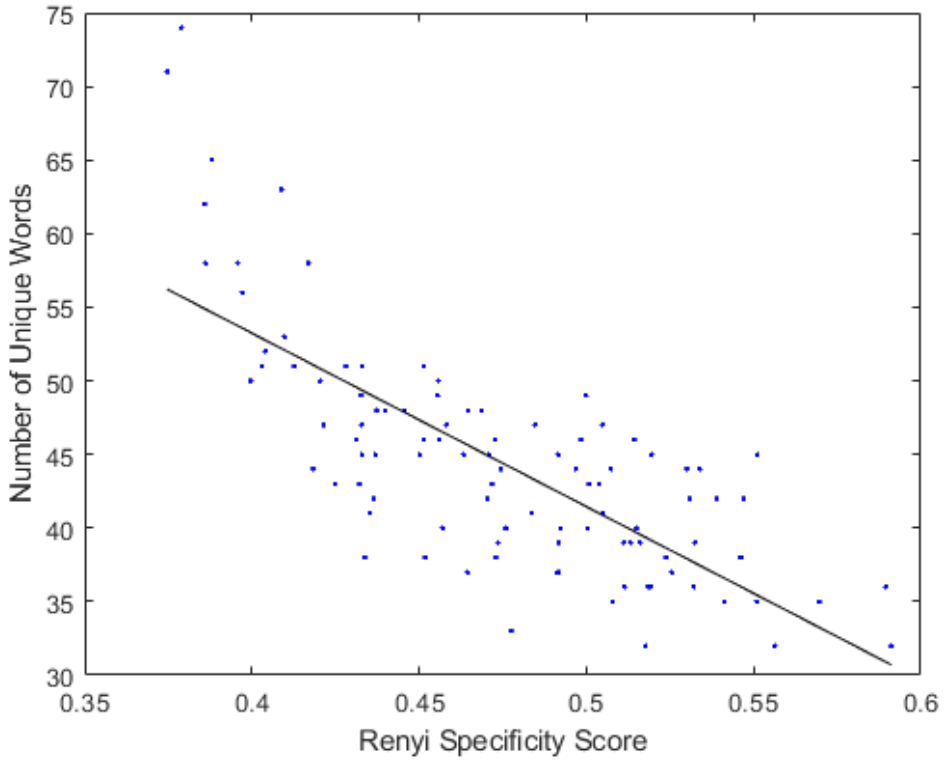


Figure 6.6 – Plot showing Rényi Specificity decreasing with the frequency of unique words associated with a font (corr.=-0.7544, $p << 0.05$).

Understanding Rényi Specificity through Subjective terms

We collected Likert score data on how "visually appealing", "stand out", "memorable", and "unique" people thought each of the 100 fonts were. 15 participants provided results, via *Amazon Mechanical Turk*. We paid participants $0.10 per HIT. For each term, we attempted to correlate the average Likert score per font, across all participants, with our Rényi Specificity scores. Our results are shown in Table 6.5. Correlations in bold are significant ($p < 0.05$). Also see Figure 8.6 in the appendix, for a screenshot of a survey that we distributed to participants. Perception of memorability correlated positively with Rényi Specificity scores. This may imply the opposite with regards to true memorability, based on previous image memorability work [259, 260], although

this was focused on photographic images. Perception of uniqueness was inversely correlated with font Specificity. Visual appeal correlated positively.

Corr. Coef.	100 Fonts
Memorable	**0.2832**
Stand-out	0.1353
Unique	**-0.2913**
Visually Appealing	**0.3544**

Table 6.5 – Correlations of Likert scores of subjective terms with Rényi Specificity scores. Significant correlations ($p < 0.05$) are in bold.

6.4.5 Automated Font Specificity via Cosine Similarity of Word Embeddings

At this point, we believed Rényi Specificity to be a useful measure, which could be collected via crowdsourcing, over time. But it still requires direct collection of word frequencies from participants. So, we wanted to produce a method of computing font Specificity values which had many of the same properties of our Rényi Specificity scores, but could more explicitly use text to do so, rather than in a derived or relative fashion, using word frequencies alone.

At first, we wanted to replicate the *Image Specificity* [7] method of computing Specificity values, using font word data. The method uses *WordNet* 'path similarity' as a basis for producing a Specificity score [193].

But this approach required pairs of sentences per image (or in our case, per font) to compute reasonable Specificity values. The authors pre-processed participant provided words in each sentence and took the base-forms of each word via *lemmatisation*, to produce their dataset (E.g. 'running' becomes 'run', jumped becomes 'jump'). For each pair of these processed sentences, per image, the maximum similarity between pairs of words in each sentence is computed, as defined via the *WordNet* corpus [193]. This is then weighted by the frequency of each word's occurrence in the current sentence and

across all sentences (via *tf-idf* or term-frequency, inverse document frequency). Across all pairs of words in the two sentences, the sum of these maximum similarity values is obtained. Across all pairs of sentences, the average of these values is taken, to produce a Specificity measure for an image.

Since we did not ask participants to provide sentences per font/image, we could not calculate Specificity via pairs of sentences per image. We believed that asking participants to provide sentences would request superfluous information such as connectives and punctuation, unneeded to describe fonts in enough detail for discrimination.

So, we attempted to produce an approximation to the *Image Specificity* approach, which compares only pairs of words per font, rather than pairs of sentences. See Algorithm 6.1 for our formulation.

#	**Algorithm**
1.	$wordlist_f \leftarrow$ List of words *a*ssociated with a font, f
2.	$path\_sims_f \leftarrow$ For each word w_a in $wordlist_f$, obtain the remaining words wl_b and repeat steps 3, 4 and 9.
3.	$tf_{w_a} = term\_frequency(w_a)$
4.	$path\_sim\_list_{w_a} \leftarrow$ For each word w_b in wl_b repeat steps 5 to 8.
5.	$sl_a, sl_b \leftarrow$ Obtain the list of synsets sl_a of word w_a and sl_b of word w_b, using *WordNet.*
6.	$ps_{w_{ab}} \leftarrow$ Repeat steps 7 and 8 for all pairs of synsets (syn_a, syn_b), where syn_a in sl_a, and syn_b in sl_b
7.	$ps_{ab} = path\_similarity(syn_a, syn_b)$ $ps_{ba} = path\_similarity(syn_b, syn_a)$
8.	$ps_{w_{ab}} = ps_{w_{ab}} < \max(ps_{ab}, ps_{ba})\ ?\ \max(ps_{ab}, ps_{ba}) : ps_{w_{ab}}$
9.	$path\_sim\_w_a = tf_{w_a} \times max\ (path\_sim\_list\_w_a)$
10.	**return** avg$(path_sims_f)$ // avg. of contributions is Specificity of a font

Algorithm 6.1 – Path Similarity-based Specificity formulation

Regarding step 7 of Algorithm 6.1, Both ps_{ab} and ps_{ba} are computed since the $path\_similarity$ function is not commutative [7]. For step 9, the term-frequency weighting is calculated via the term/word frequency of w_a (the current word being compared to all others of a font) in its original list of words. Eventually, word w_b will be the first word of a pair, so its term-frequency will be considered.

Within Algorithm 6.1, we still weighted each word's contribution to the similarity, but instead used only term-frequency (TF) to do this (without inverse document frequency (IDF), since each word per document is equally frequent among documents if there is only one document of interest – i.e. one font's list of words). Doing this for all pairs of words associated with a font, and summing those values together gave us per-font scores to compare with our Rényi Specificity scores. The correlation between path similarity-based font Specificity scores and Rényi font Specificity scores was approx. 0.43 ($p << 0.01$).

But, on removal of the term-frequency component of the formulation, we achieved no (significant) correlations using this approach, apart from in the case of the standard deviation of each font's path similarity-based Specificity contributions, correlating positively with Rényi Specificity scores (see Table 6.6). Additionally, these correlations were low in magnitude. This suggested that term-frequency was the main factor behind the correlation between path similarity-based scores and Rényi Specificity scores.

Corr. Coef.	**Correlation with Rényi scores**	**p-value**
Per-font Std. Dev. of path similarity contributions	0.1187	$p < 0.05$
Per-font Std. Dev. of path similarity contributions without outliers (removed according to Cook's distance)	0.2256	$p < 0.01$

Table 6.6 – Correlations between standard deviations of path similarity based font Specificity contributions (without term-frequency weightings) and Rényi Specificity scores.

We believe that this lack of significance in direct correlations of path similarity-based Specificity (without term-frequency) and Rényi Specificity occurred due to one or more of the following points:

1. There was a lack of word variation per font – predominantly adjectives and nouns (approx. 60% of word distribution across all fonts).
2. *WordNet* synsets per font may have also been too similar – either per font, or when compared to fonts of similar score. We thought that there was too little variation in the words between score bins and believed that we'd probably need more font word data before any changes would appear.

Due to this result, we attempted to formulate a similar method which used a different measure of similarity between words, and so we attempted to use distances between word-embedding vectors as a basis for this method. In this approach, words are represented via their word vector's closeness/similarity to other vectors.

Word vectors are modelled to capture the occurrence probability of words, given a context of other words, by the distances between their respective word vectors. They also capture some notion of word semantics/meaning, expressed via arithmetic between word vectors (see Figure 6.7).

For example, word vectors can be added or subtracted from another word vector, to discover words similar in meaning or further away in meaning [270, 271]:

- vector('Paris') - vector('France') + vector('Italy') results in a vector that is very close to vector('Rome')
- vector('king') - vector('man') + vector('woman') is close to vector('queen')

We decided not to train our own *skip-gram* or *bag-of-words* vector space model, but instead used existing/pre-trained word embeddings for our purposes – due to computational efficiency (time required) and accuracy reasons (size of word dataset). To this end, we gathered existing *Word2Vec (skip-gram* model*)* [64, 271], *GloVe* [63] and *FastText* (*skip-gram* model) word embeddings to test with.

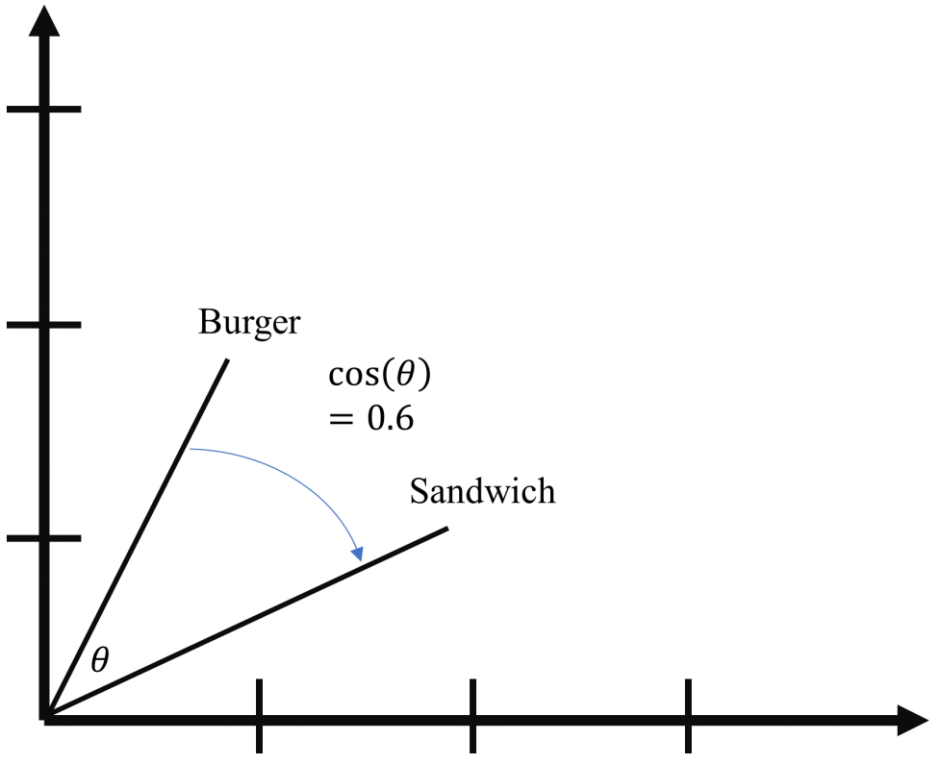


Figure 6.7 – Image visualising cosine similarity between two word vectors; data by Shi et al. [272].

Word Embedding Comparison

Given pairs of words, w_a and w_b, associated with each font, f, we can compute the cosine similarity, $CS(v_a, v_b)$, between their word vectors v_a and v_b, as a contribution to the Specificity of f:

$$CS(v_a, v_b) = \cos(\theta) = \frac{v_a \cdot v_b}{\|v_a\|\|v_b\|} = \frac{\sum_{i=1}^{n} v_{a_i} v_{b_i}}{\sqrt{\sum_{i=1}^{n} {v_{a_i}}^2} \sqrt{\sum_{i=1}^{n} {v_{b_i}}^2}}$$

Equation 6.4 – Cosine similarity between two word vectors.

It is also possible to substitute cosine similarity, $CS(v_a, v_b)$, with the Jensen-Shannon Divergence (JSD) between v_a and v_b, by firstly normalising v_a and v_b, and then computing: $1 - sqrt(JSD(v_a||v_b))$, treating the word vectors v_a and v_b as probability distributions P and Q, over low-dimensional representations of word co-occurrence.

$$JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M)$$
$$= \frac{1}{2}\sum P(\log P - \log M) + \frac{1}{2}\sum Q(\log Q - \log M)$$

Equation 6.5 – Jensen-Shannon divergence

$$JSS(v_a, v_b) = 1 - sqrt(JSD(v_a||v_b))$$

Equation 6.6 – Jensen-Shannon similarity between two word vectors.

We computed Specificity scores using three pre-trained word embeddings [64, 63, 273], where each word vector was of 300 dimensions:

1. **Word2Vec (skip-gram)**: Google News dataset (100 billion tokens, 3 million words/phrases) [64]
2. **GLoVe**: Wikipedia 2014 + Gigaword 5 (6B tokens, 400K words/vocabulary) [63]
3. **fastText (skip-gram)**: Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (approx. 1 million words, 16B tokens) [274, 273]

For each of these word embeddings, we computed per-font Specificity scores based on:

- **Cosine similarity**: $CS(v_a, v_b)$, based on pairs of word vectors, v_a and v_b.
- **Jensen-Shannon similarity**: $JSS(v_a, v_b)$, based on pairs of word vectors, v_a and v_b.

- **L2 Norm of element-wise min + max across word vectors**: Given the word vectors v_i, of words w_i associated with each font, compute the L2 norm of the element-wise minimum values across all v_i concatenated with the element-wise maximums across all v_i:

$$s_i = L2\left(\underset{v_i}{\text{elementwise min.}}\ concat\ \underset{v_i}{\text{elementwise max}}\right)$$

- **L2 Norm of element-wise avg. across word vectors**: Given the word vectors v_i of words w_i associated with each font, compute the L2 norm of element-wise averages of values across all v_i: $s_i = L2(\underset{v_i}{\text{elementwise avg.}})$

For the cosine similarity and Jensen-Shannon similarity cases, to determine the contribution of a word, w_a, to a font's Specificity (that was used to describe the font), w_a is compared to every *other* word associated with that font, in a pairwise manner. This is done by computing one of the similarity measures on the word vectors of each pair of words. The maximum of these similarities is taken and weighted by the term frequency of w_a, to produce w_a's Specificity. The set of Specificity values generated across all possible w_a (all words associated with a font), are averaged, to obtain a font's Specificity. For the L2 norm cases, the Specificity of a font is computed as earlier described. An issue is that any words to be processed must be included in the word vector dictionary, which is why we used existing word embeddings that were pre-trained with many words/tokens.

We attempted to correlate each set of word embedding-derived scores with our original Rényi Specificity scores and found the scores based on cosine similarity and Jensen-Shannon similarity correlated significantly, and positively with Rényi Specificity. The L2 norm-based scores correlated negatively (and nearly always significantly). See

Table 6.7 for our results. However, we could see that all of the correlations based on cosine similarity and Jensen-Shannon similarity Specificity scores were very close to one another…

Correlation Coefficient	Avg. of Cosine Sim.	Avg. of Jensen-Shannon Sim.	L2 Norm of element-wise avg.	L2 Norm of element-wise min + max
Word2Vec	**0.6442**	**0.6228**	**-0.2097**	**-0.2500**
GLoVe	**0.6302**	**0.608**	**-0.2352**	**-0.2664**
fastText	**0.6262**	**0.6133**	**-0.2360**	-0.1360

Table 6.7 – Correlations between Rényi Specificity scores, and scores obtained between various word embeddings and formulations of font Specificity (with term-frequency weighting). Correlations in bold are significant ($p<0.05$).

On removal of the term-frequency component in the Specificity formulation for these cases, all of their correlations had also become negative. After this, when we ordered the fonts by ascending score to visualise them, we saw the opposite trend to what was expected. Simpler, more legible fonts had lower scores, and more creative fonts which were varied in shape and texture, relative to the others, had higher scores. The solution was to invert the resulting scores, simply by subtracting them from 1. We also applied this to the L2 norm-based scores. In the cosine similarity case, this maps scores in the interval [-1 1] to [2 0] (therefore, halving the result will normalise the values). The lowest scores became highest, and highest scores became lowest. This changed the sign of the correlations in all cases, to positive.

After this change, for the cosine similarity and Jensen-Shannon similarity Specificity scores, we obtained only one similar correlation to beforehand, which was also the highest relative to other cases. This was produced using cosine similarity and the word2vec-based embedding, which we selected for our automated font Specificity approach. We refer to Specificity scores derived this way, from this point forward (see Algorithm 6.2).

We also found that when using the cosine similarity formulation, correlations across all word embeddings were higher than the path similarity-based approach. See Table 6.8 for our results. Correlations in bold are significant ($p < 0.05$).

Correlation Coefficient	1 – Avg. of Cosine Sim.	1 – Avg. of Jensen-Shannon Sim.	1 - L2 Norm of element-wise avg.	1 - L2 Norm of element-wise min + max
Word2Vec	**0.6086**	0.1886	**0.2097**	**0.2500**
GLoVe	**0.4850**	**0.4858**	**0.2352**	**0.2664**
fastText	**0.4977**	**0.4119**	**0.2360**	0.1360

Table 6.8 – Correlations between Rényi Specificity scores, and scores obtained between various word embeddings and formulations of font Specificity. Correlations in bold are significant ($p<0.05$).

Given a set of participant words associated with each font, f, Algorithm 6.2 computes a Specificity value for f. For each word, w_a in w$ordlist_f$, a Specificity contribution is obtained for the font f, by comparing w_a to every *other* word w_b in w$ordlist_f$, via the cosine similarity between their word vectors. The maximum of these similarities is taken to produce w_a's Specificity. The set of Specificity values generated across all w_a, are averaged and subtracted from 1, to obtain a font's Specificity.

It may be that fonts with more varied curvature, texture, font weight, etc. tend to have mainly similar word vectors, as even though there is more word variation, it is specialised to a certain style or concept (resulting in weakly to strongly positive cosine similarities between word vectors). For example, many words might relate to entities and scenery found at a beach, such as: 'sea', 'sandcastle', 'shell', 'jellyfish'. Whereas fonts which are plainer may have less word variation, but more variation between the concepts or meanings of those words (negative, orthogonal or weakly positive cosine similarities between word vectors).

Word Embedding-based Font Specificity Definition

#	Algorithm
	1. $syn_f \leftarrow$ Given a list of words associated with a font, f, obtain the synsets of each word, using *WordNet*. Concatenate them into a list. 2. $rem_f \leftarrow$ All words of f, where synsets were not found. 3. $wordlist_f \leftarrow$ Extract lemma words for every synset in syn_f and the lemma words of words in rem_f. Concatenate them into a list. 4. $cos\_sims_f \leftarrow$ For each word w_a in $wordlist_f$, obtain the remaining words wl_b and repeat steps 5 and 8. 5. $cos\_sim\_list_{w_a} \leftarrow$ For each word w_b in wl_b repeat steps 6 to 7. 6. $wv[w_a], wv[w_b] \leftarrow$ Obtain the word-vectors of word w_a and word w_b, using a word embedding dictionary. // Given that w_a and w_b must be in the wordvector dictionary to be computed, at most $length(wordlist_f) - 1$ similarity values will be computed. 7. $cs_{ab} = cosine\_similarity(wv[w_a], wv[w_b])$ 8. $cos\_sim\_w_a = max\ (cos\_sim\_list\_w_a)$ 9. **return** $1 - \text{avg}(cos\_sims_f)$ // 1 – avg. of contributions is Specificity of a font

Algorithm 6.2 – Cosine Similarity-based Specificity formulation

Statistical Tests

To check for consistency in the distributions of words between sub-groups of fonts, we randomly sampled two disjoint groups of 50 fonts from the total of 100 and computed a two-sample Kolmogorov-Smirnov test between the two groups' Specificity scores. The resulting p-value was greater than 0.05, indicating that the two groups of scores were considered to be from the same distribution. Since approximately every 20 fonts were described by a different group of participants (avg. of 22.4 participants per 20 fonts), we considered this result to imply that font Specificity scores obtained by different groups of participants are also likely to have the same distribution.

We also checked to see whether the font Specificity scores were Gaussian distributed, and we found this to be the case, using a chi-squared goodness of fit test ($p > 0.05$). The distribution of scores was skewed however (skewness: -0.2651).

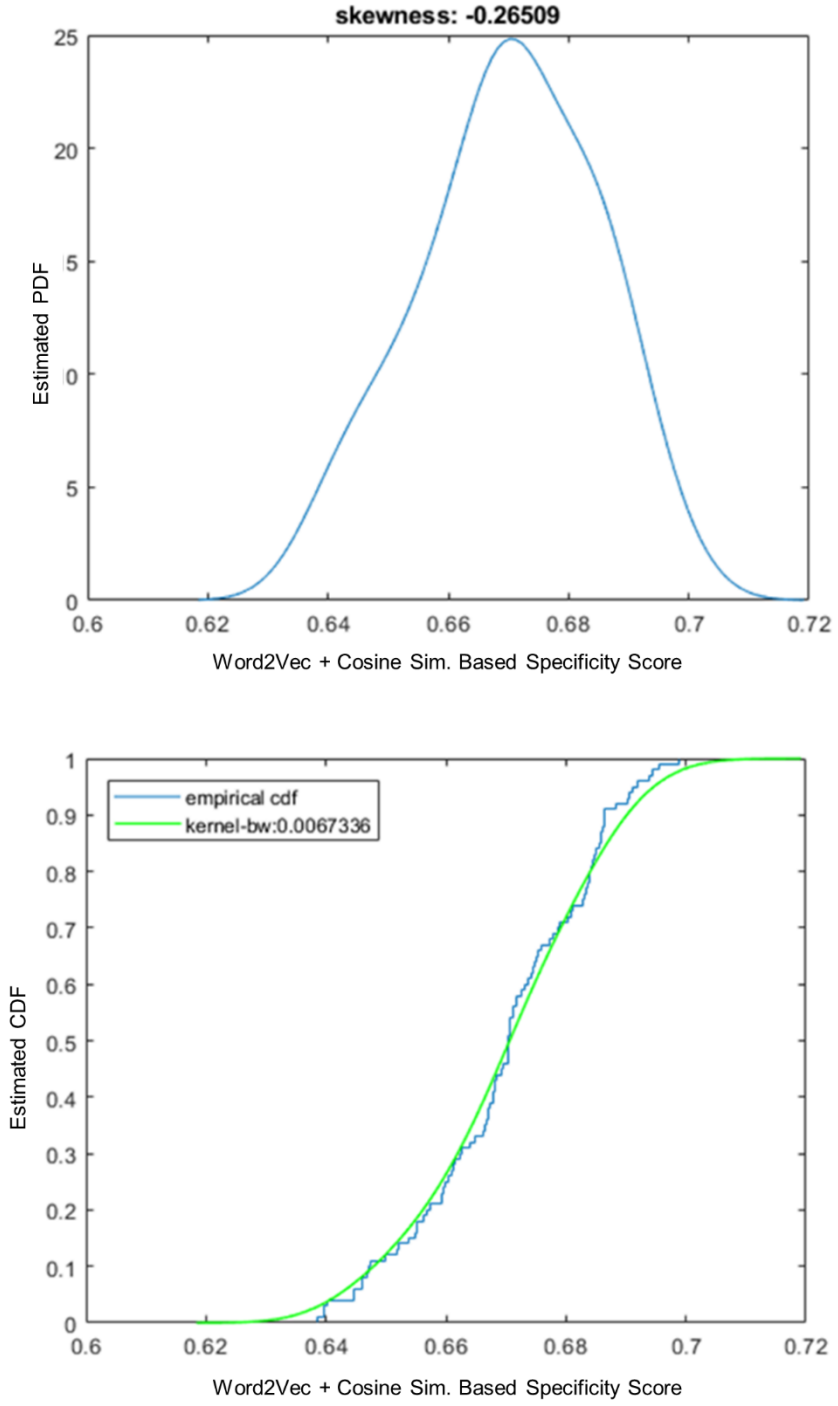


Figure 6.8 – (Top) Estimated PDF of word embedding-based Specificity scores and (Bottom) their empirical CDF (blue), with a fitted curve overlaid (green).

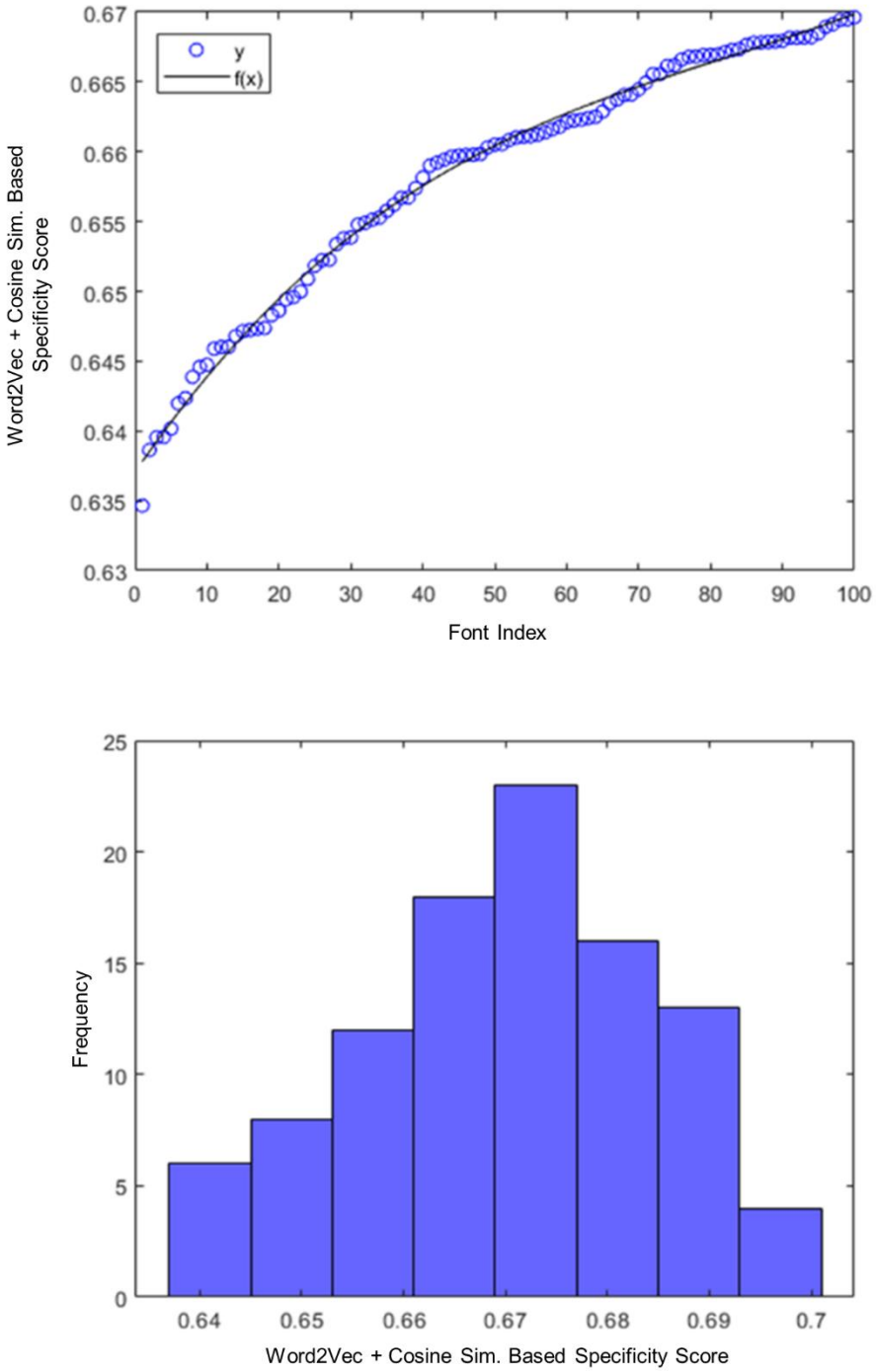


Figure 6.9 – Plots created via word embedding-based font Specificity scores. (Top) Plot of data sorted from lowest to highest score, with a fitted curved overlaid. (Bottom) 8-bin histogram of those scores.

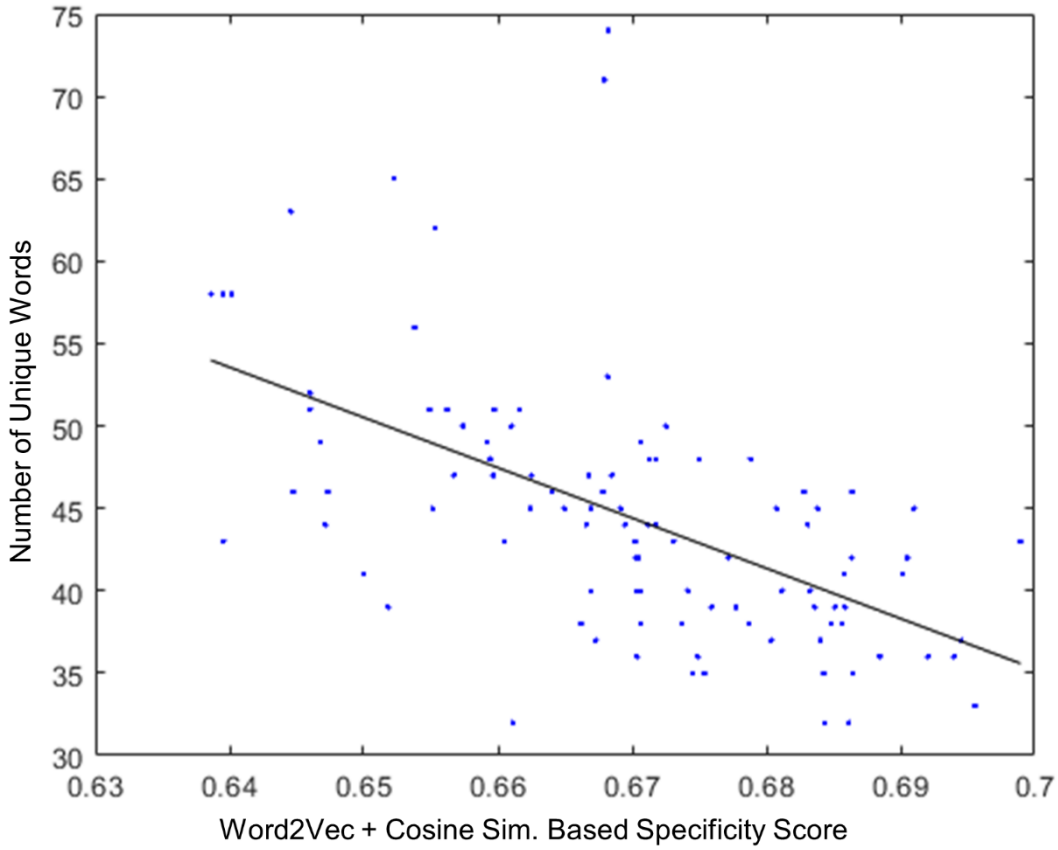


Figure 6.10 – Plot showing word embedding-based Specificity decreasing with the frequency of unique words associated with a font.

We show the PDF and empirical CDF of the Specificity scores, in Figure 6.8, where the data is left-skewed. A plot of the scores ordered from lowest to highest, along with a histogram, is also shown in Figure 6.9. The modal score is approximately 0.675.

Visualising Word Embedding-based Font Specificity

Figure 6.11 provides a visualisation of the bottom-10 and top-10 fonts according to their word embedding-based Specificity scores. The number of unique words provided per font correlated significantly and negatively with these Specificity scores (corr.=-0.5601, $p << 0.01$), similarly to Rényi Specificity. See Figure 6.10 for a visual representation.

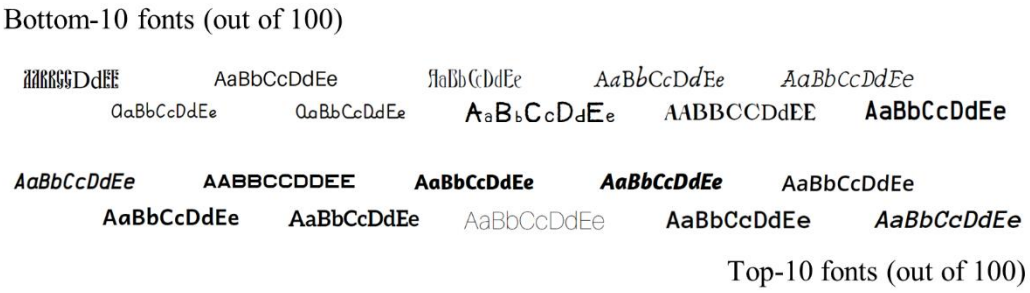


Figure 6.11 – Plot of the bottom-10 and top-10 fonts according to their word embedding-based Specificity scores.

Understanding Word Embedding-based Font Specificity through Subjective terms

We additionally collected Likert score data on several subjective terms, including those collected in the Rényi Specificity case, for each of the 100 fonts. 15 participants provided results, via *Amazon Mechanical Turk*. As before, we paid $0.10 per HIT. For each term, we attempted to correlate the average Likert score per font, across all participants, with our word embedding-based Specificity scores. The chosen terms were sampled from the top-50 most frequent words provided by participants (see Figure 6.3), and topics/subjects associated with those words. See Table 6.9 for our results. Correlations in bold are significant ($p < 0.05$).

Corr. Coef.	100 Fonts
Memorable	**0.2774**
Stand-out	-0.0381
Unique	**-0.3591**
Visually Appealing	**0.3471**
Legible	**0.4676**
Creative	**-0.5174**
Boring	**0.4405**
Fun	-0.1841
Elegant	**0.2257**
Modern	-0.1962
Normal	**0.3651**

Table 6.9 - Correlations of Likert scores of subjective terms with word embedding-based Specificity scores. Correlations in bold are significant ($p<0.05$).

Perception of memorability correlated positively with these Specificity scores, as in the Rényi Specificity case. Uniqueness was inversely correlated with font Specificity as before, in addition to creativity. Visual appeal or a notion of 'elegance', correlated positively, as did perception of legibility and the perception of how boring or normal each font was.

We further studied the statistically significant terms, to determine whether some of the terms were associated with fonts of high, low or medium score, or covered the entire distribution of fonts.

For each of these terms, we attempted to correlate their font Likert score averages with word embedding-based Specificity scores obtained from fonts of low, medium and high score. Table 6.10 shows our results.

Corr. Coef.	Low scores (33 fonts)	Medium scores (33 fonts)	High scores (34 fonts)
Memorable	0.12	0.31	-0.09
Unique	-0.22	-0.07	-0.0118
Visually Appealing	**0.40**	0.19	0.03
Legible	**0.40**	0.09	0.11
Creative	**-0.37**	-0.15	-0.01
Boring	0.34	-0.0064	0.02
Elegant	0.23	-0.02	-0.02
Normal	0.22	0.06	-0.17

Table 6.10 – Correlations of Likert scores of subjective terms with word embedding-based Specificity scores, given fonts with low, medium and high Specificity. Significant correlations ($p < 0.05$) are in bold.

We highlight correlations based on perception of memorability and uniqueness, in addition to a notion of normality or elegance in fonts, as belonging to the entire distribution of fonts, since there was no significant correlation based on certain font Specificity score ranges.

Reduced visual appeal, reduced legibility or higher creativity was associated with fonts of low Specificity score. This suggested that medium to high Specificity score fonts may be less creative or more visually appealing or legible, compared to fonts of low score, but not compared to each other, since there was not a significant trend occurring between fonts of medium score, and separately, fonts of high score, in these cases.

When comparing fonts of low score to the whole font dataset, correlation due to perception of creativity increased in magnitude from corr.=-0.37 to corr.=-0.5174, indicating that this trend spans the entire font dataset, but is mostly concentrated in fonts of low score. Perception of legibility also followed this trend, but to a lesser degree.

For the lower score fonts, corr.=0.40, vs. 0.4676, for all fonts. Correlation due to visual appeal reduced from corr.=0.40 (low score fonts) to corr.=0.3471 (all fonts), indicating that this trend may be relegated to lower score fonts.

We can also see changes in word and word frequency distribution as word embedding-based Specificity increases. Figure 8.9 to Figure 8.11 in the appendix provide plots of the top-50 words' frequencies for 5 groups of 20 fonts sampled according to increasing word embedding-based Specificity score (without replacement), shown from top (lowest scores) to bottom (highest scores). Note that the word frequency axis varies per plot.

6.5 Learning

To generalise our font Specificity scores to fonts we had not yet encountered, we created a convolutional neural network model, which takes as input an RGB image, I_{rgb} of size 200×200, where $I_{rgb} \in \mathbb{R}^{3\times200\times200}$ and outputs a single font Specificity score prediction, $\hat{y}_i$. Each font was represented as an image, I_f of approximately 512×233

in dimensions, containing A-Z characters presented in upper and lower case, with the numbers 0-9 placed beneath them. These were centred over a white background.

Training data came in the form of (I_{rgb}, y_i) pairs. Multiple pairs were sampled from each font image, I_f via 100 randomly positioned windows of pixels, of size 200×200, which were each associated with the font Specificity score, y_i of font image, I_f. Given the 100-font dataset, this generated 10000 (I_{rgb}, y_i) training examples. As the network was trained using k=10 fold cross validation, each fold used 9000 examples for training, and 1000 for testing.

Number of Fonts	CV Correlation	R^2	Number of Epochs (training was split into 120 epochs at a time)	Training Time
100	0.709875312	0.5039229	2280	19 hours

Table 6.11 – Correlation between k=10 cross-validation Specificity score predictions and actual Specificity scores of fonts (based on word embeddings), based on training examples created via font image sub-samples.

Specificity scores y_i, were based on per-font word distributions wd_f, based on the stimuli of I_f, that we wanted to map to s_f (we assumed that there exists an invertible function mapping $wd_f \rightarrow s_f$ and $I_f \rightarrow wd_f$, for this to be possible): $h(I_f) = s_f$.

The trained convolutional neural network represented, h. Overall, a function was learned to minimise the error between predicted font Specificity scores, $\hat{y}_i$, and the target font Specificity scores, y_i, computed via word-embeddings. To do this, we minimised the mean-square error loss function (shown in Equation 6.7), via stochastic gradient descent, and standard backpropagation. The stochastic gradient descent executed in batch sizes of 24. $\|\boldsymbol{W}\|_2^2$ and $\|b\|_2^2$ are L^2 regularizers that were employed

to prevent overfitting. Additionally, N is the number of samples used for training (the training batch size).

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{b}) = \frac{1}{N} \sum_{i \in TrainData} (y_i - \hat{y}_i)^2 + 0.01\|\boldsymbol{W}\|_2^2 + 0.01\|\boldsymbol{b}\|_2^2$$

Equation 6.7 – Loss function for font Specificity prediction.

Similarly, to the Schelling meshes case, the network structure used *LeakyReLU* neurons at each layer, excluding the output layer, which was a linear combination of the previous layer's outputs. Figure 6.12 shows a diagram of the neural network. It was trained using $k = 10$ fold cross-validation ($\frac{1}{10}$ of the I_{rgb}, were separated into a test set). After training, we averaged the prediction accuracy of each of the held-out test data sets. But, since each font corresponded to 100 different predictions, we instead correlated each prediction $\hat{y}_i$, from I_{rgb}, with the word embedding-based font Specificity scores, y_i, of each font I_f, that each I_{rgb} was sampled from. We achieved a significant correlation ($p<0.05$) of 0.71, after approx. 2280 epochs of training across the 10 folds. For a random predictor, correlations would hover around 0. Results are shown in Table 6.11. Our implementation used the *Theano* [255] and *Keras* [256] Python libraries.

6.5.1 Predicting Font Specificity with Image-Based Shape Descriptors

To attempt to answer the question of whether shape descriptors could predict font Specificity, we trained a descriptor-based fully-connected neural network to learn a regression function that mapped a single vector of shape descriptor values, to word embedding-based font Specificity values.

We computed shape descriptors such as: Sobel filters (magnitudes and orientations), k-means vector-quantised SIFT and SURF descriptors, Histogram of Oriented Gradients

(HoG) values and contour curvatures, to see if they were useful for this. The network consisted of 4 layers. It accepted a single d dimensional vector, v and output a single value, $\hat{y}_i$, a font Specificity prediction.

Each font image I_f, was used to generate multiple vectors, v_j, by randomly sampling 200 x 200 windows of I_f, $j = 100$ times. Each v_j was subsequently paired with their associated font's Specificity score, y_i. Given the 100 font dataset, this generated 10000 (v_j, y_i) training examples. As the network was trained using k=10 fold cross validation, each fold used 9000 examples for training, and 1000 for testing. The hypothesis h, was represented by the network: $h(v_j) = \hat{y}_i$.

The SIFT, SURF and Histogram of Oriented Gradients vectors, were 512 dimensions each, so in that case, $d = 512$. The Sobel filter values were 64 dimensions each, so for these, $d = 64$. The number of neurons for each successive hidden layer, were $\lfloor 0.5d \rfloor$, $\lfloor 0.25d \rfloor$, $\lfloor 0.125d \rfloor$ and $\lfloor 0.06125d \rfloor$, respectively. The loss function we minimised was similarly to that of the convolutional neural network, but we chose a simpler fully-connected network in order to not allow other factors, such as network design, to influence the results.

We trained the network using stochastic gradient descent and backpropagation, but achieved worse results in most cases, with predictions sometimes varying largely from the word embedding-based font Specificity scores. We highlight the Histogram of Oriented Gradients descriptor, as a good predictor of Specificity. Our results are shown in Table 6.12.

CV Learning Results	CV Correlation	R^2	Number of Epochs (training was split into 150 epochs at a time)
Histogram of Oriented Gradients (PCA applied – 512 dimensions)	0.52	0.27	2250
Sobel filter (64 bins)	0.43	0.19	2250
Vector-Quantised SIFT (512 bins)	0.25	0.06	1500
Vector-Quantised SURF (512 bins)	0.06	0.004	2250

Table 6.12 – Correlations between k=10 cross-validation Specificity score predictions and actual Specificity scores of fonts associated with training examples based on image descriptors.

6.5.2 Neural Network Structure (Colour Image-Based)

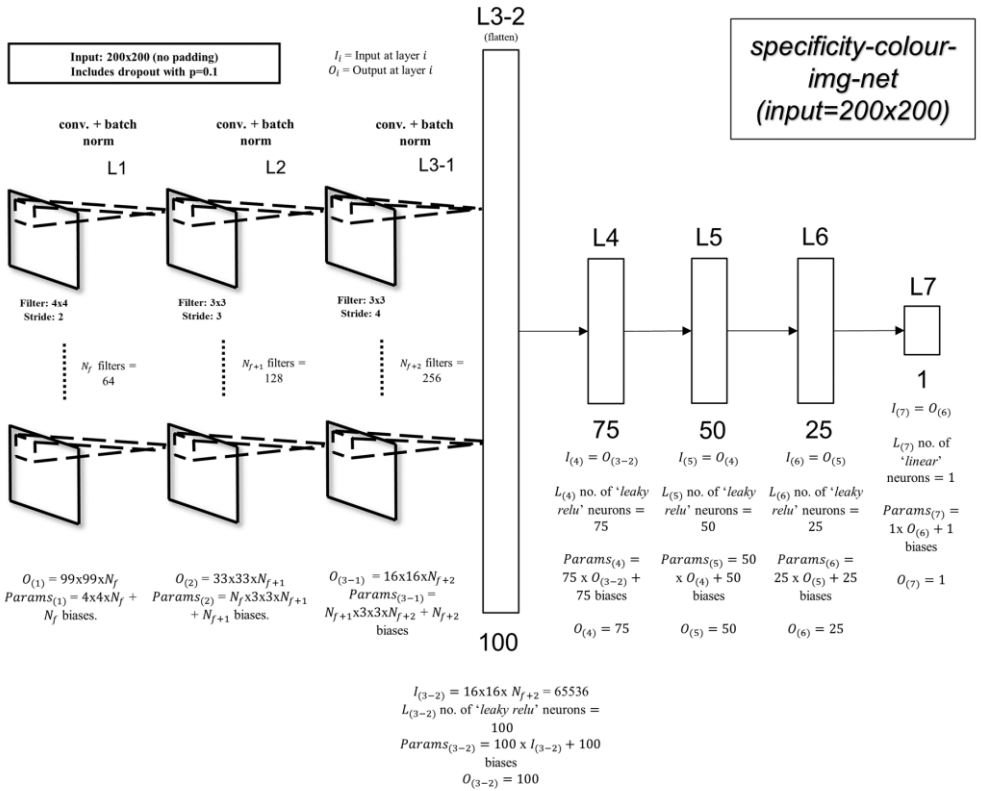


Figure 6.12 – Diagram of a convolutional neural network for word embedding-based font Specificity prediction. Takes as input a 200x200 sub-image of a font, and outputs a font Specificity prediction, $\hat{y}_i$.

6.6 Applications

We can search for and visualise fonts via their Specificity scores. They can be searched for and compared according to their Specificity, to find complementary fonts (similar Specificity values) and contrasting fonts (large relative distance between Specificity values). Fonts can be clustered via their Specificity, with visual differences expressed between clusters.

6.6.1 Visualisation

Fonts sorted according to word embedding-based Specificity scores (in ascending order), show a trend of increased simplicity as Specificity increases. See the appendix for a list of the entire 100 fonts, ordered in this way, with their font names (Figure 8.8).

We can additionally visualise the font dataset using t-SNE embeddings based on the weights of the convolutional neural network trained for font Specificity prediction. This is done using the outputs at layer L6 of the network (see section 6.5.2). An example visualisation is shown in Figure 6.13. The picture shows a trend of increasing contrast from the top-left to the bottom right of the plot. This indicates that the network may be using contrast to predict Specificity, as was shown to be possible via the reasonable cross-validation learning results of the PCA-HoG descriptor and Sobel descriptor to some degree. The network seems to have clustered thinner fonts of similar geometry together; a similar pattern appears for the bold fonts.

Via Amazon Mechanical Turk, we showed 15 participants a visualisation of all 100 fonts, created via t-SNE embeddings derived from our Specificity prediction model (see Figure 6.13) alongside another visualisation based on t-SNE embeddings of a shape descriptor applied to the fonts.

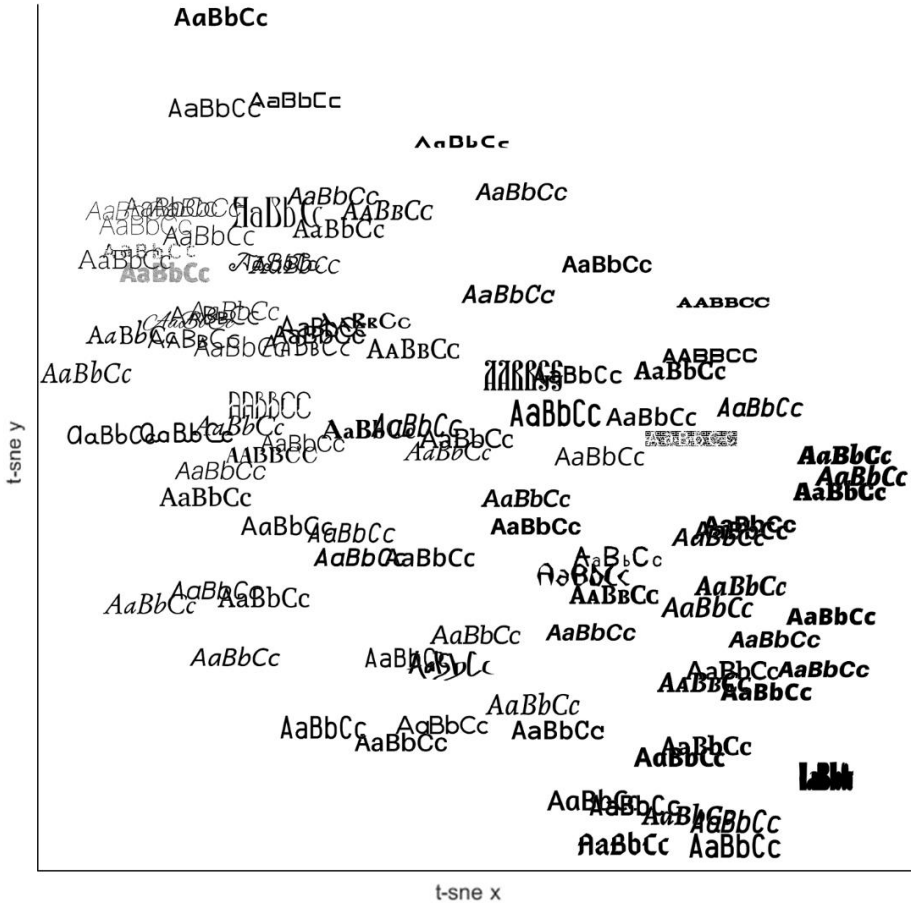


Figure 6.13 – Visualisation of 100 font t-SNE embedding based on outputs of our font Specificity prediction model.

We repeated this for each shape descriptor. Each group of 15 participants was not necessarily unique for each descriptor. In two separate cases, we asked participants to select which set of fonts (visualisation image) they found 1) most legible or 2) most creative. Regarding legibility, participants found the Specificity score-based plot to be preferred over those of the Sobel and SIFT descriptors, but not the curvature and PCA-HoG descriptors. We believe this was due to an experimental design error, in that participants focused on the shape of the visualisation image (due to the contrast between the overlapping black text of the fonts and the white background), rather than the fonts themselves to make their decision. We believe that the majority of participants effectively gave their preference on the entire shape of each visualisation image vs

another, based on the shape boundary defined by the change in contrast in the image, and did not mention similarities/differences between clusters of fonts, contrast etc. Due to this, we held a more focused survey on creativity and legibility of fonts at extreme Specificity levels, where the layout of fonts on display was fixed. See Figure 6.14 for some images of fonts that were shown to participants.

6.6.2 Search via Extremes: Specificity vs. Shape Descriptors

Via Amazon Mechanical Turk, we held a survey to determine whether fonts with low Specificity score were considered to be more creative (but still legible), compared to fonts close to these from a perspective of shape descriptors. Specifically, we took the closest 5 fonts to a query font of high creativity from the earlier Likert survey results, based on word embedding-based Specificity scores and a range of shape descriptors.

We asked 30 participants on Amazon Mechanical Turk to look at 7 rows of fonts, with each row corresponding to the closest 5 fonts in terms of: 1) word embedding-based Specificity scores, 2) Pixel-wise Euclidean Distance, 3) Contour curvature, 4) Sobel filter magnitudes and orientations, 5) Vector-quantised SIFT descriptors and 6) PCA-reduced Histogram of Oriented Gradients (to 256 dimensions) and 7) SURF descriptors. Their task was to select the row of fonts that was most creative, but still legible. See Figure 8.7 for a screenshot of a survey that we provided to participants (refer to appendix).

By shuffling rows, different orderings were shown per 10 participants. Participants were also required to explain why they made their choice, with a minimum 50 characters. The average creativity Likert score for the query font was approximately 3.87 out of 5 (96^{th} percentile; from earlier collected data). Average Likert scores for the fonts in each row indicated that the font Specificity-based row would be most selected – see column

7 of Table 6.13. Row selection frequencies were collated, which indicated that the row of fonts based on Specificity scores was selected most often. Totals are shown in Table 6.16.

Participants that selected the Specificity row of fonts, mentioned that the fonts were different, yet clear and legible. For example, one participant mentioned that these fonts were “all easily readable” and contained “several different and creative fonts”. Another participant explicitly indicated some row comparisons, stating that “row 1 had the most unique variety of fonts that were legible”, additionally saying that “the other rows such as 7 and 2 contained illegible fonts” and “rows 3-6 contained fonts I felt were too similar to one another and were not very creative” (see creativity section of Figure 6.14). A third participant stated that the row “has the widest array of fonts but you can still read them all”.

Other participants that selected the SIFT-based row of fonts, indicated that they were clear and easy to read, or were not as boring as the fonts of other rows. For example, one participant stated that the fonts were the “clearest and easiest to read”. Another stated that all the fonts in the SIFT-based row were legible (row 3 in creativity section of Figure 6.14), saying that they considered the Specificity row of fonts boring: “none of the font sets are boring like in row 1”. Overall, they mentioned that they “just [liked] the looks of the fonts in row 3 the best”. In these two cases, the participants did not actually answer the question correctly of selecting the most creative row of fonts (which are also still legible). Additionally, a participant that selected the curvature-based row mentioned that they felt “this font type is the most legible while it also looks interesting to me”. Another participant selected the same row stating that it was “[clear] and visible when compared to other fonts”.

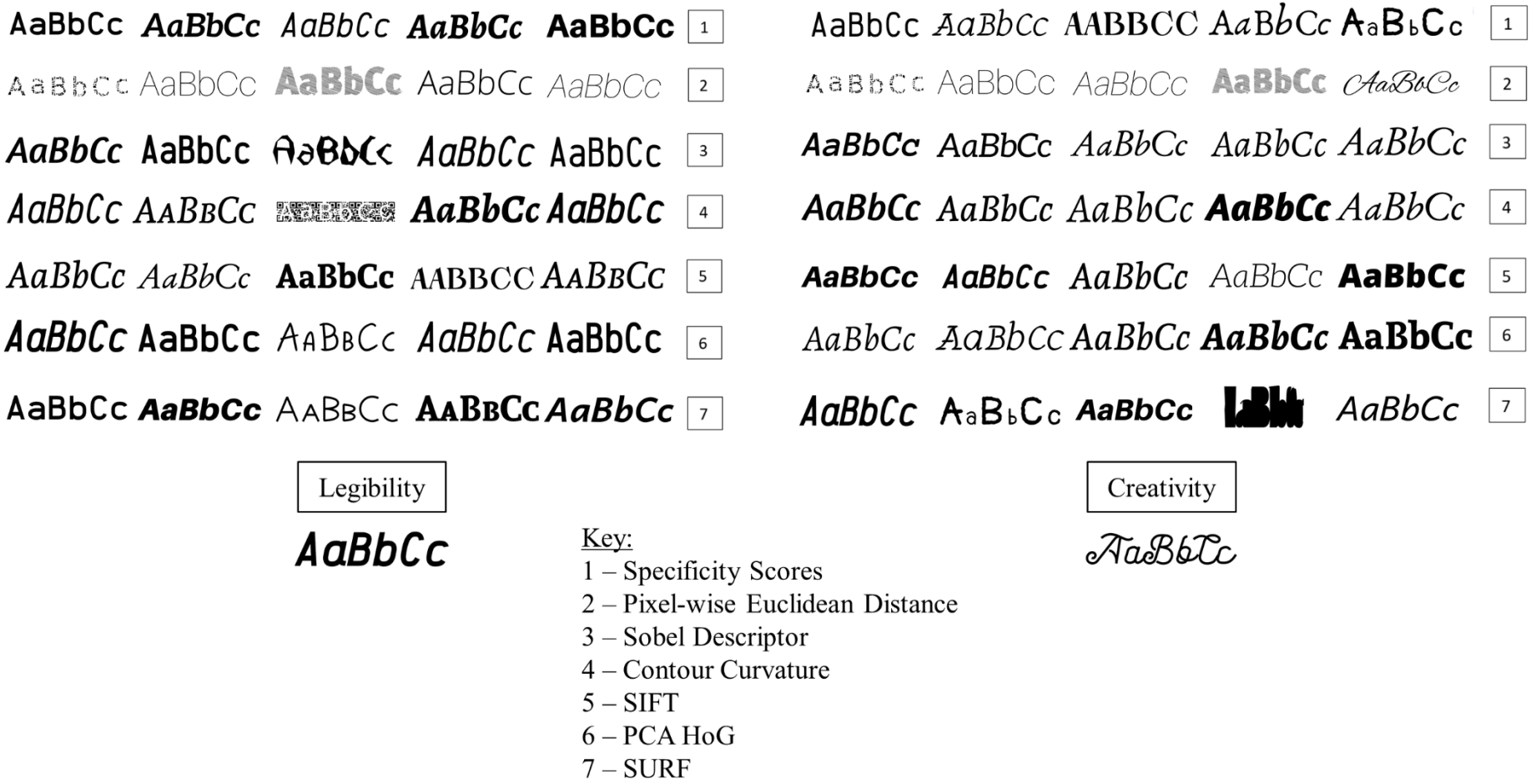


Figure 6.14 – Rows of fonts shown to participants in font creativity and legibility surveys for comparison of Specificity to shape descriptors. Each query font is located under each category (this was hidden from participants).

Creativity	1st	2nd	3rd	4th	5th	Avg. of Font Scores	Std. Dev. of Font Scores	Percentile of Scores Avg.
Font Specificity	2.13	3.53	3.47	3.33	3.27	3.27	0.59	80th percentile
Euclidean Distance	3.47	2.33	2.00	3.00	3.87	2.93	0.77	71st percentile
Sobel Values	2.27	2.53	2.67	2.60	3.13	2.64	0.31	Approx. 60th percentile
Curvature	2.60	2.93	2.60	2.93	3.13	2.84	0.23	Approx. 68th percentile
SIFT	2.20	2.07	2.93	2.27	2.07	2.31	0.36	Approx. 35th percentile
HOG	2.60	3.53	2.93	2.87	2.67	2.92	0.37	Approx. 70th percentile
SURF	2.00	3.27	2.20	3.73	2.40	2.72	0.74	Approx. 65th percentile

Table 6.13 – Average creativity Likert scores of fonts in each row (1st=closest, 5th=farthest), based on 15 participants, in addition to the average score of each row and its approximate percentile relative to the entire 100 font dataset.

	Spec Scores	L1/L2 Distance	Contour Curvature	PCA-HoG	SIFT	Sobel	SURF
Creativity	9	2	6	3	6	3	1

Table 6.14 – Most creative (while still legible) font row selection frequencies among 30 participants.

We cannot be certain whether ‘interesting’ or ‘clear’ corresponds to ‘creativity’ or similar terms, like uniqueness. Discounting these participants responses would effectively increase the proportion of participants that selected the Specificity-based row of fonts. Regardless, all selection frequencies (including those of the two participants mentioned) are shown in Table 6.14. As a counterpoint, one participant mentioned that the SIFT based row was “the only row that the letters were totally readable and that it had some unique fonts”, indicating that they gave a valid answer. Overall, participants that selected the Specificity-based row tended to provide less ambiguity in their answers and more often showed that they accurately followed the survey instructions.

We additionally held another survey asking participants which row of fonts was most legible, given rows of fonts closest to a query font with high legibility. As before, row shuffling was employed per 10 participants. The row frequency for closest fonts via Specificity scores was selected most often, with a greater magnitude compared to the shape descriptors. The average legibility Likert score for the query font was approximately 4.47 out of 5 (80^{th} percentile; from earlier collected data). As with the creativity case, average Likert scores for each row of fonts indicated that the font Specificity-based row would be most selected – see column 7 of Table 6.15. Row selection frequencies are shown in Table 6.16.

Participants that selected the Specificity-based row of fonts, indicated simplicity, font boldness, consistent spacing, scale and orientation and good spacing between letters as the reasoning behind their selections. For example, one participant stated that the fonts were not "too close together or too far apart", indicating that font spacing was a factor.

Legibility	**1st**	**2nd**	**3rd**	**4th**	**5th**	**Avg. of Font Scores**	**Std. Dev. of Font Scores**	**Percentile of Scores Avg.**
Font Specificity	4.20	4.40	3.87	4.07	4.73	4.25	0.33	Approx. 52nd percentile
Euclidean Distance	2.33	4.60	3.53	4.07	3.93	3.69	0.85	Approx. 22nd percentile
Sobel Values	3.67	4.40	2.00	4.33	4.00	3.68	0.98	Approx. 22nd percentile
Curvature	4.33	4.13	1.40	4.07	4.00	3.59	1.23	Approx. 21st percentile
SIFT	4.27	4.13	4.33	3.87	4.13	4.15	0.18	Approx. 45th percentile
HOG	4.00	4.40	3.33	4.33	4.40	4.09	0.46	Approx. 40th percentile
SURF	4.20	4.27	3.93	4.53	4.27	4.24	0.21	Approx. 52nd percentile

Table 6.15 – Average legibility Likert scores of fonts in each row (1^{st}=closest, 5^{th}=farthest), based on 15 participants, in addition to the average score of each row and its approximate percentile relative to the entire 100 font dataset.

	Spec Scores	L1/L2 Distance	Contour Curvature	PCA-HoG	SIFT	Sobel	SURF
Legibility	14	0	0	6	4	4	1

Table 6.16 – Most legible font row selection frequencies among 30 participants.

Another mentioned that “the font designs were fairly straightforward with no weird spacing or flourishes making them harder to read”, citing consistency. A third participant mentioned that font boldness was important, in addition to consistent font scale and orientation. They stated that fonts were “all in bold” and that there was “no distortion in any [of] the lettering”, where the lettering was “basically the same size and easy to read compared to the other rows”, with each font being of a “very similar type” and “not at all confusing in any way". Similarly, another participant stated that “most fonts were bolded” and that they were the “clearest of the others”.

These results suggest that searching for fonts in the extreme of font Specificity scores, can yield creative (but still legible), or most legible fonts which are preferred to closest fonts obtained via many shape descriptors.

6.6.3 Clustering

Visual Patterns

We also attempted to determine whether clustering on the font Specificity scores (both word embedding-based Specificity and Rényi Specificity) provided visual differences between clusters, and whether those differed to clusters obtained via shape descriptors. We used the best of 10 iterations of k-means, to obtain cluster assignments for each font. The shape descriptors employed were those used earlier in the chapter. All clusterings were produced with k=4, unless specified. Ahead, we may also refer to word embedding-based Specificity as word2vec Specificity, on occasion.

Visually, the clearest patterns were visible in the Rényi and word2vec Specificity-based clusterings, and the SIFT clustering. We provide visualisations of cluster assignments via images of the fonts themselves shown in Figure 6.15 to Figure 6.18. On repeated clustering attempts, we see similar visual patterns, indicating that there is some stability in the results.

Rényi and word2vec Specificity-based clusters show reduced variation or greater simplicity in font edges, according to the increasing mean Specificity score of each cluster. The clusters with the top-2 highest mean scores contain simple fonts which tend to be very thin or bold. SIFT best reflects contrast/boldness and thinness of fonts between clusters. This is somewhat like clustering according to the contrast of each font against the background.

Contour curvature sometimes shows contrast and texture variation between clusters, but some of the simplest fonts are blended within the extreme fonts in those clusters. Also, cluster assignments are uneven. This unevenness is also true for the PCA-HoG and Sobel-based clustering. Sobel filtering produces cluster assignments determined via contrast or thickness.

Statistical Comparison

Similarity Tests

We used the Adjusted Rand Index (ARI) and Adjusted Mutual Information (AMI) metrics to determine the similarity between clusterings based on word2vec font Specificity and those based on shape descriptors. They each measure the degree of agreement between two partitions of a dataset, both taking into account the possibility that cluster assignments occurred due to chance. For the ARI, a value of 1 indicates the exact same clustering, and values near 0 suggest a uniform random clustering.

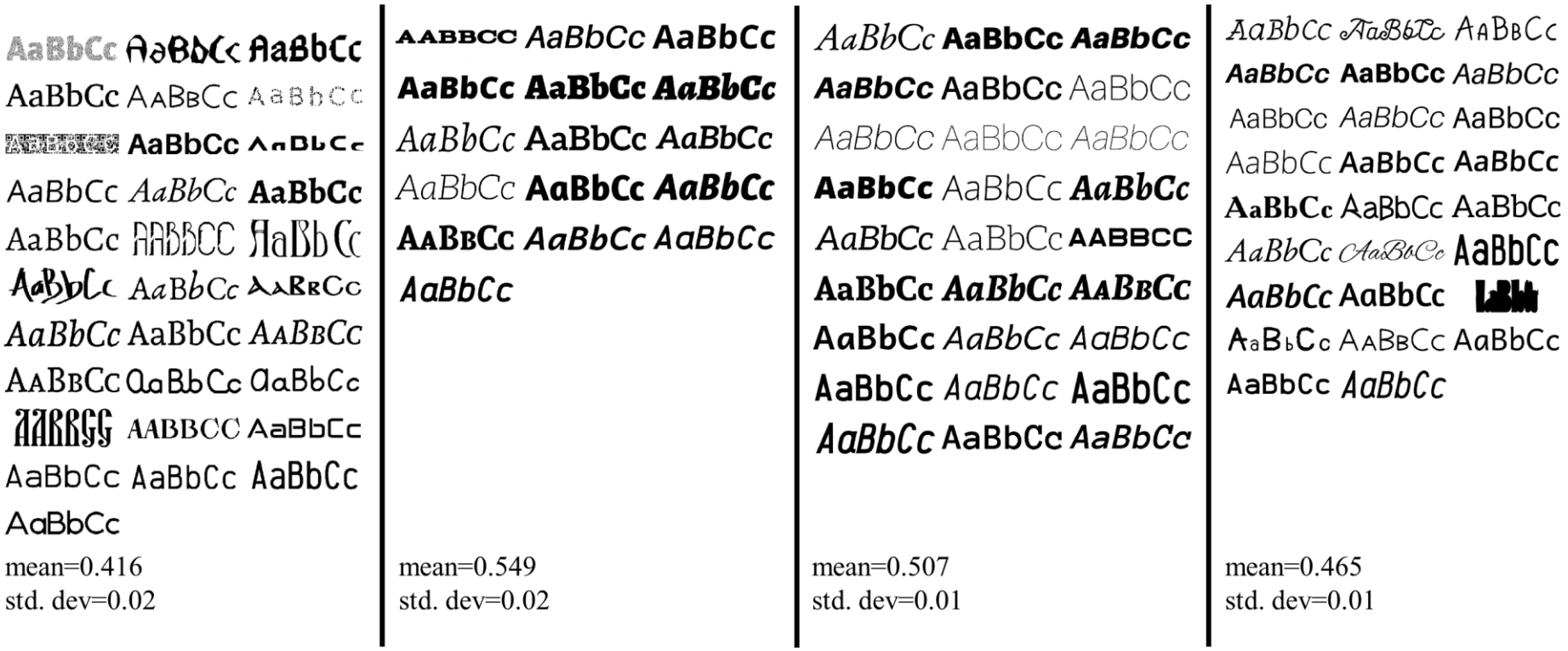


Figure 6.15 – Visualised Rényi Specificity-based clustering (k-means; k=4) for all 100 fonts. For each cluster, the mean and standard deviation of the Specificity scores of its constituent fonts is displayed.

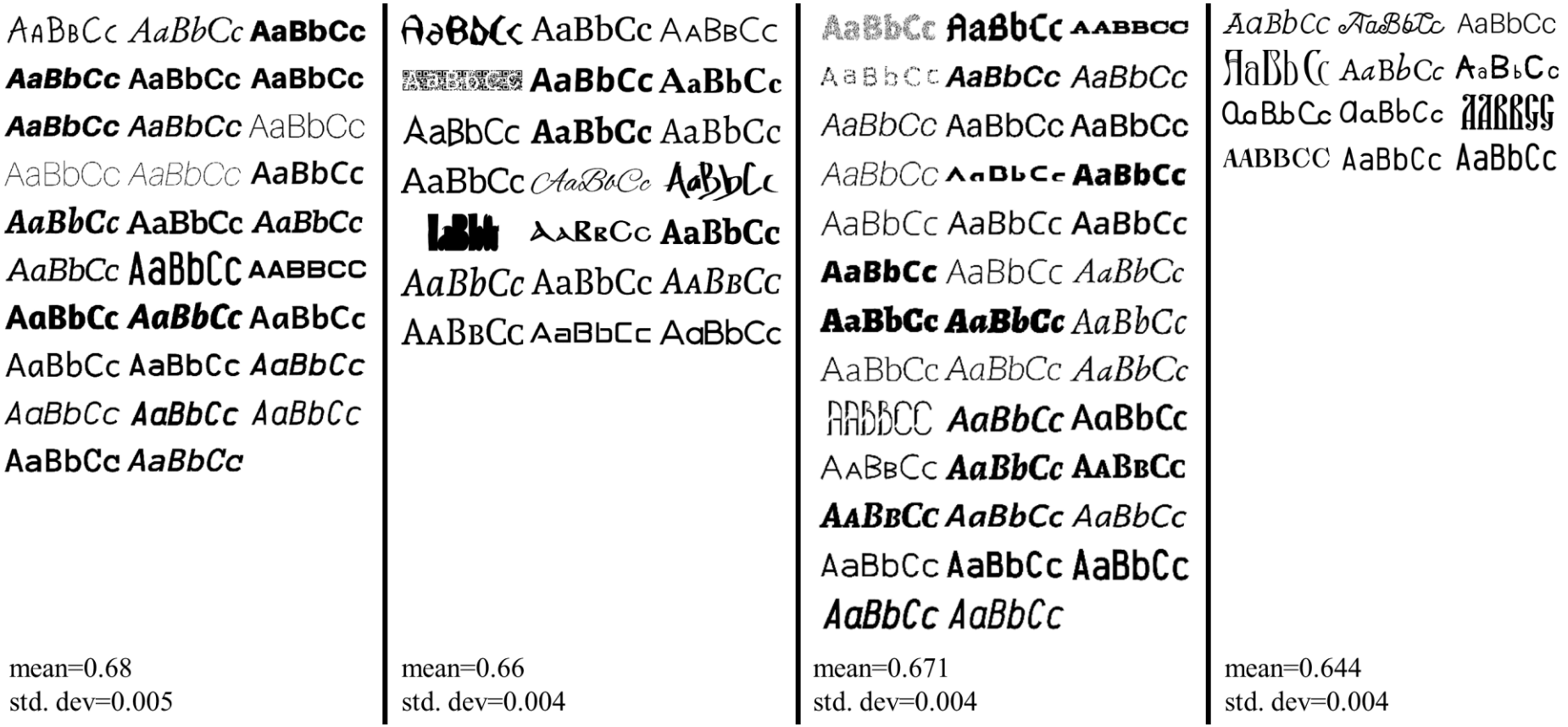


Figure 6.16 – Visualised word2vec Specificity-based clustering (k-means; k=4) for all 100 fonts. For each cluster, the mean and standard deviation of the Specificity scores of its constituent fonts is displayed.

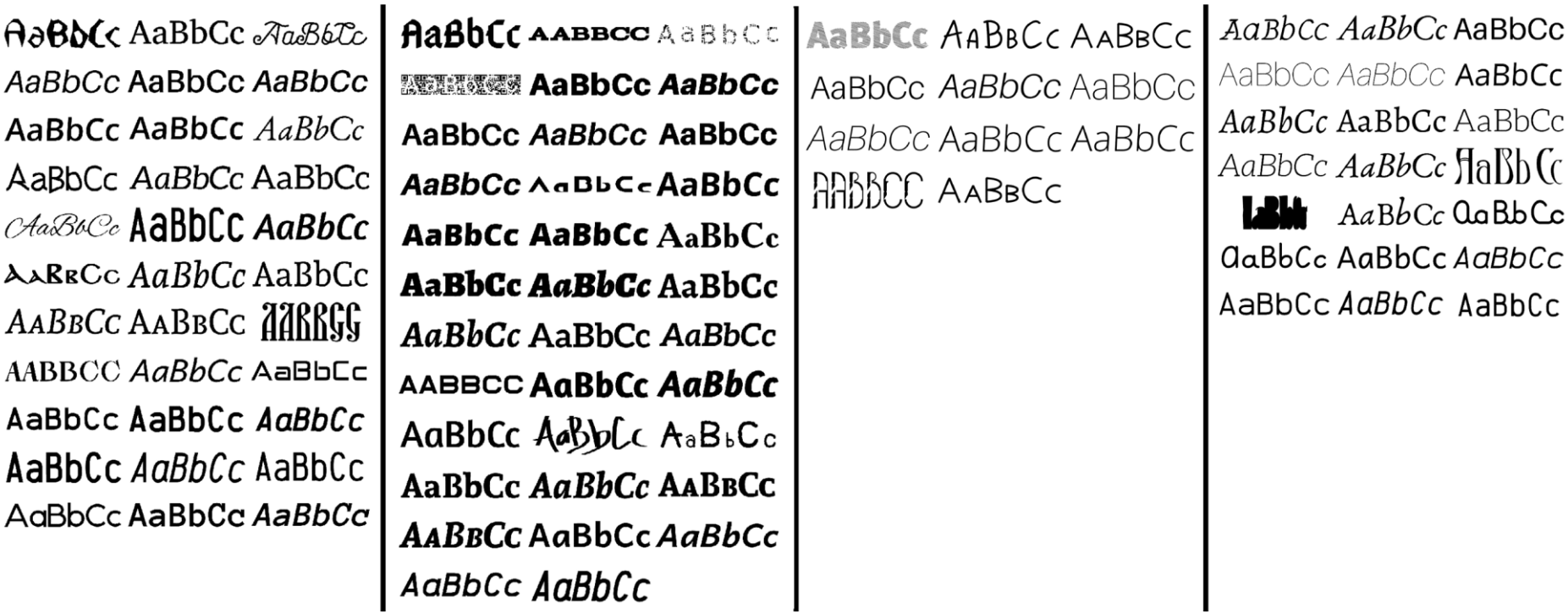


Figure 6.17 – Visualised SIFT-based clustering (k-means; k=4) for all 100 fonts.

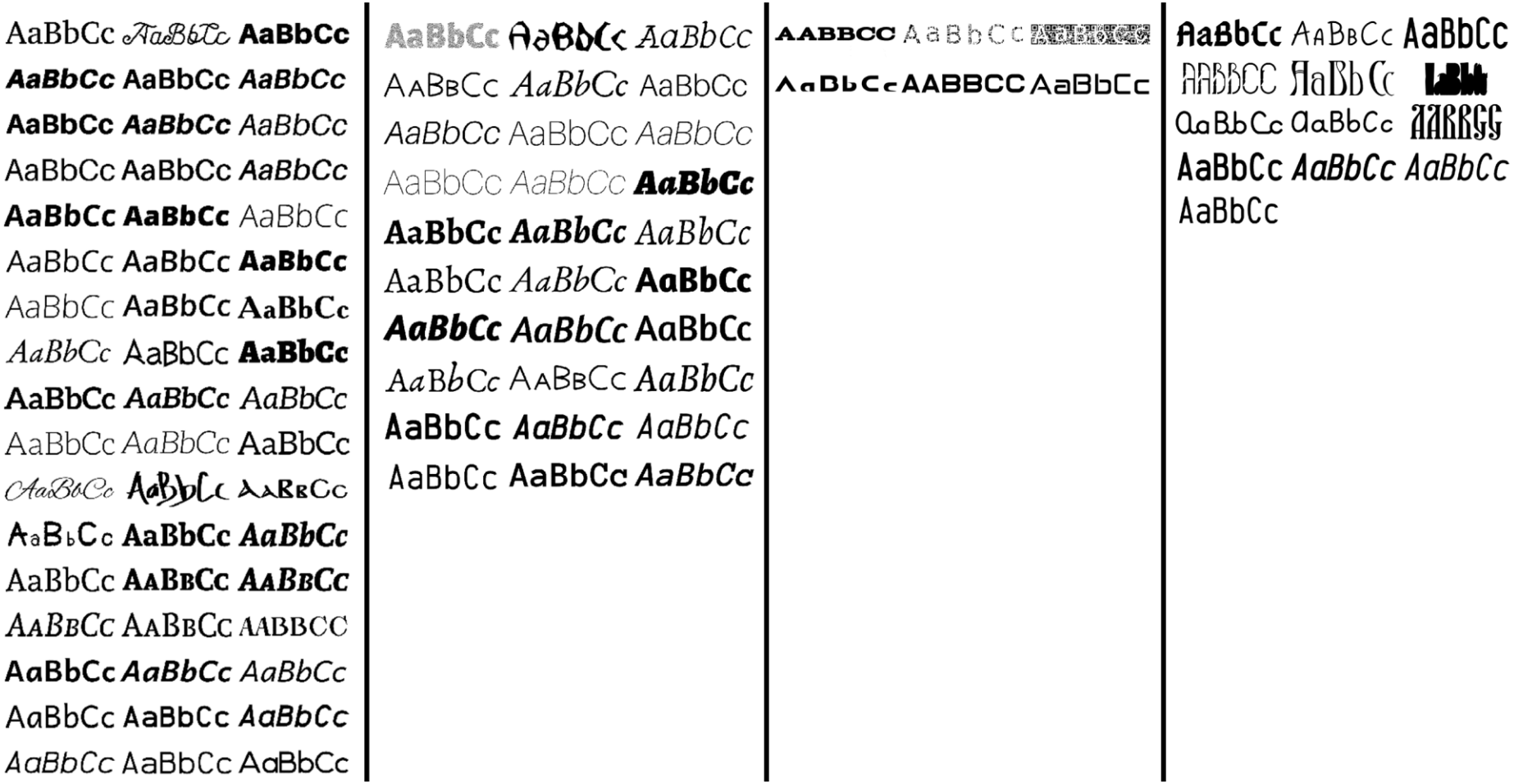


Figure 6.18 – Visualised PCA-HoG-based clustering (k-means; k=4) for all 100 fonts.

The AMI results in 0 on average, when the clustering occurred due to chance (so values can be negative), and 1 when the partitions are the same. Our results in Table 6.17 and Table 6.18 show that cluster assignments based on word2vec or Rényi Specificity vs. those of the tested shape descriptors do not align, even in the most extreme cases of the Sobel descriptor.

Adjusted Rand Index	Contour Curvature	PCA-HoG	SIFT	Sobel	SURF
Rényi (k=4)	0.002	0.025	0.024	0.043	-0.022
Rényi (k=8)	-0.007	0.005	0.026	-0.008	-0.016
Word2vec (k=4)	0.008	0.006	0.041	0.028	0.005
Word2vec (k=8)	-0.005	0.00035	0.00032	0.007	-0.01

Table 6.17 – Adjusted Rand Index values based on pairing k-means clusterings derived from Specificity scores with a clustering based on each shape descriptor (values $> 2\sigma$ away from the mean are in green).

Mutual Information	Contour Curvature	PCA-HoG	SIFT	Sobel	SURF
Rényi (k=4)	0.007	0.019	0.042	0.078	-0.019
Rényi (k=8)	0.006	0.01	0.021	0.003	-0.04
Word2vec (k=4)	-0.005	0.014	0.042	0.026	-0.012
Word2vec (k=8)	0.011	0.022	0.01	0.033	-0.003

Table 6.18 – Adjusted Mutual Information values based on pairing k-means clusterings derived from Specificity scores with a clustering based on each shape descriptor (values $> 2\sigma$ away from the mean are in green).

ANOVA Using Specificity Scores

For a statistical test of the difference in clusterings obtained via shape descriptors vs. font Specificity, we performed a series of one-way ANOVA tests, to find significant differences, if any, between mean cluster Specificity scores in each case (each descriptor, or font Specificity measure). As a sanity test, we found that $p << 0.0001$ when clustering via the Rényi Specificity and word2vec-based Specificity scores, respectively. Therefore, we considered k-means as a valid option for clustering via the shape descriptors.

Table 6.19 provides results for k=4 and k=8 clusters, respectively. For k=4, we can see that clustering via contour curvatures, or the Sobel descriptor, provides significant differences in mean Rényi Specificity. Only the PCA-reduced HoG does so for the word2vec case. For k=8, the PCA-HoG descriptor provides significant differences in mean Specificity across both the word2vec and Rényi variants. SIFT and Sobel also do so in the Rényi case. Although there are differences here, both the Sobel descriptor (k=4) and PCA-HoG descriptor (k=8) measure gradients of an image at differing levels of detail – and with scale invariance in the case of PCA-HoG. This result, in addition to the earlier cross-validation learning results based on descriptors, indicates that some information regarding image gradients is reflected in a font's Specificity score – whether in word frequency, or variation in co-occurrence of words associated with a font (as defined by the word2vec embedding).

Specificity Scores vs. Shape Descriptors	**Contour Curvature**	**PCA-HoG**	**SIFT**	**Sobel**	**SURF**
Word2vec (k=4)	**p < 0.05**	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05
Word2vec (k=8)	p >= 0.05	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05
Rényi (k=4)	p >= 0.05	p >= 0.05	**p < 0.01**	**p < 0.01**	p >= 0.05
Rényi (k=8)	p >= 0.05	**p < 0.01**	**p < 0.05**	**p < 0.01**	p >= 0.05

Table 6.19 – One-way ANOVA test results for significant differences in mean Specificity score of clusters obtained via k-means (k=4, k=8), across all 100 fonts.

ANOVA Using Likert Scores

We additionally held a series of one-way ANOVA tests to determine whether there were significant between cluster differences in the Likert score means of the fonts. Table 6.20 and Table 6.21 show that significant differences occur across many Likert criteria and image-based descriptors, whether clustering via k=4 or k=8 clusters. Overall, we see that for k=4, both the word2vec and Rényi Specificity variants showed the most significant difference in creativity Likert score means. For the k=8 case, Rényi Specificity was best. In terms of legibility, Rényi Specificity and the Sobel descriptor

yielded the most significant differences. For k=4, Rényi Specificity and the PCA-HoG descriptor did so for legibility, whereas word2vec Specificity did so for creativity.

For uniqueness, visual appeal and notions of normality or boringness, both Specificity measures were similar in significance to other descriptors at $p < 0.05$. The Sobel descriptor yielded most differences with respect to memorability and a notion of 'standing out' for both the k=4 and k=8 cases. Finally, for notions of elegance, fun and modernity, shape descriptors tended to yield better results than Specificity – apart from a notion of fun, in the k=4 case, where word2vec Specificity yielded a significant result, in addition to the Sobel descriptor.

6.7 Discussion

To begin, we restate our hypotheses below:

1. A more legible font is more Specific.
2. A more creative font is less Specific and less legible.
3. Visually appealing fonts are more Specific but potentially not the most Specific fonts.
4. Specificity conveys information different to that of existing image descriptors.

We have found that more Specific fonts tend to be more legible, as fewer words or words with a more consistent meaning are used to describe them. This can be visualised by ordering the font dataset by increasing Rényi or word2vec-based Specificity score. We also observe negative correlations with per-font word frequency and font Specificity scores (Rényi and word2vec variants).

	Word2vec	Rényi	Contour Curvature	PCA-HoG	SIFT	Sobel	SURF
Legible	**p < 0.01**	**p < 0.01**	**p < 0.01**	**p < 0.01**	p >= 0.05	**p < 0.01**	p >= 0.05
Creative	**p < 0.01**	**p < 0.01**	**p < 0.05**	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05
Memorable	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05
'Stand Out'	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05
Unique	**p < 0.01**	**p < 0.05**	p >= 0.05	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05
Visual Appeal	**p < 0.05**	**p < 0.01**	**p < 0.05**	**p < 0.05**	p >= 0.05	**p < 0.01**	**p < 0.01**
Boring	**p < 0.01**	**p < 0.01**	**p < 0.05**	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05
Normal	**p < 0.01**	**p < 0.01**	p >= 0.05	**p < 0.05**	p >= 0.05	**p < 0.01**	**p < 0.05**
Elegant	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05
Fun	**p < 0.05**	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05
Modern	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	**p < 0.01**	p >= 0.05

Table 6.20 – One-way ANOVA test results for significant differences in mean Likert score of clusters obtained via k-means (k=4), across all fonts.

	Word2vec	Rényi	Contour Curvature	PCA-HoG	SIFT	Sobel	SURF
Legible	**p < 0.01**	**p < 0.01**	**p < 0.01**	**p < 0.01**	p >= 0.05	**p < 0.01**	p >= 0.05
Creative	**p < 0.01**	**p < 0.01**	**p < 0.05**	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05
Memorable	p >= 0.05	**p < 0.05**	**p < 0.05**	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05
'Stand Out'	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.01**	p >= 0.05
Unique	**p < 0.01**	**p < 0.05**	**p < 0.05**	**p < 0.01**	p >= 0.05	p >= 0.05	p >= 0.05
Visual Appeal	**p < 0.01**	**p < 0.01**	**p < 0.05**	**p < 0.01**	p >= 0.05	**p < 0.01**	p >= 0.05
Boring	**p < 0.01**	**p < 0.01**	p >= 0.05	**p < 0.01**	**p < 0.05**	**p < 0.05**	p >= 0.05
Normal	**p < 0.01**	**p < 0.01**	p >= 0.05	**p < 0.01**	p >= 0.05	**p < 0.05**	p >= 0.05
Elegant	p >= 0.05	p >= 0.05	p >= 0.05	**p < 0.05**	p >= 0.05	**p < 0.01**	p >= 0.05
Fun	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05
Modern	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05	p >= 0.05

Table 6.21 – One-way ANOVA test results for significant differences in mean Likert score of clusters obtained via k-means (k=8), across all fonts.

Additionally, we have found that fonts high in Specificity are considered more legible than fonts close to them in terms of the tested image-based shape descriptors. These results indicate that hypothesis #1 may be correct, but do not directly confirm it.

Visual appeal correlates with Specificity (Rényi and word2vec variants), but not as strongly as notions of legibility (positive corr.) or creativity (negative corr.). This suggests that visual appeal can be a factor behind Specificity, but it is not the most important one. We conclude that hypothesis #3 is correct. We have also found that fonts low in Specificity are considered more creative (given some legibility) than fonts close to them in terms of the tested image-based shape descriptors. Given that only a subset of image-based shape descriptors have been tested, and there was a requirement for fonts to be legible to some degree, this result does not imply that hypothesis #2 is correct, but provides evidence towards it.

Through comparing clusterings (k-means) based on word2vec-based font Specificity with the tested shape descriptors, we see little to no alignment between the clusterings, when using the Adjusted Rand Index and Adjusted Mutual Information measures. Visually, we can see some differences in the clusterings, as well. Additionally, when fonts are clustered via Specificity (k-means), Specificity can more significantly distinguish between a notion of ‘creativity’ in fonts, on average, compared to the tested image-based shape descriptors. This was based on a one-way ANOVA test of the clusterings (see section 6.6.3 – “ANOVA Using Likert Scores”). These results give evidence that hypothesis #4 is correct.

6.8 Conclusion

Specific fonts can be perceived as more memorable relative to other fonts. They are considered to be more common and visually-appealing, more legible, less creative and more mundane or normal.

Fonts seem to display a mixture of many characteristics (texture, curvature, thickness) if they have low Specificity. Thinner and italic fonts tend to have higher Specificity scores, with a greater proportion of bold fonts appearing as Specificity is highest. The highest Specificity fonts tend to have only one or two clear aspects to them. For example, they are bold, bold and italic, italic and thin etc.

People have also found fonts high in Specificity to be more legible than similarly close fonts in terms of geometric shape descriptors. Specificity can be used to search for creative or legible fonts, based on these properties. Clustering using Specificity clearly shows differences between fonts which are not only based on contrast, or closest font shape or pixel alignment. Specificity-based clustering assignments can more significantly distinguish between a notion of ‘creativity’ in fonts, on average, compared to a range of shape descriptors.

We have also shown that font Specificity can be learned using an image-based convolutional network, by sub-sampling 200x200 images from a higher-resolution source image of a font’s alphanumeric characters. The Histogram of Oriented Gradients and Sobel (magnitudes and orientations) descriptors were also found to be suitable for prediction of font Specificity.

7 Conclusions

We set out to understand shapes from a lens of human perception, rather than via geometric shape descriptors. The argument was that this approach would lead to a better understanding of "shapes" or provide a different outlook on them, which geometric shape descriptors could not provide. By different, we mean a measure of *group-level saliency* which reflects how distinctive each discrete or whole element of a group is, with respect to the other elements of the group. This differs from *unit-level saliency* measures which describe or focus on an individual object at a time, with a goal of understanding which sub-components of a single object are salient. Group-level saliency approaches allow you to measure complementary aspects of the shapes within a group (i.e. perceived elements of the shapes), that are different to the underlying geometry – e.g. creativity, memorability.

We studied 3D shapes from a new point of view, treating whole shapes as Schelling Points, or *Schelling meshes*. The concept was previously only applied to the study of vertices on 3D meshes. 2D fonts were studied from a new perspective of *Specificity*, which was previously only applied to photographic images. We researched both concepts via the use of crowdsourcing platforms and machine learning, holding online surveys to gather data for further analysis, comparison and prediction of shapes/fonts more aligned with each concept, respectively. From this, we introduced potential applications of each concept.

In this chapter, we look back at the discussed approaches, summarise the research outcomes and indicate future directions.

7.1 Contributions

The alternative approaches to understanding 3D shapes and 2D fonts described in this thesis – introduced via the concepts of *Schelling Points* and *Specificity*, are our main contributions to the field of shape perception. Each approach provides a measure of some subjective elements of shape, which are consistent across many people.

Our contributions are four-fold:

- Methodologies for data collection via crowdsourcing – the '4-choose-1' study of Schelling meshes in Chapter 4, the 'Many-Within-Class' study of Schelling meshes in Chapter 5, and the font Specificity study in Chapter 6.
- Analysis of what makes a shape 'Schelling' and a font 'Specific'. As part of this, we created a scoring approach for Schelling saliency in shapes, measures of Specificity for fonts, and determined subjective properties common to Schelling meshes or Specific fonts.
- An approach to learning a function that can predict which shapes are likely to be Schelling salient, or fonts, Specific. Such a learned function can then be used to make predictions for any new font, or shape within the same class.
- Applications of Schelling meshes in visualisation and clustering, and font Specificity in visualisation, search and clustering.

7.2 Summary of Studies

7.2.1 Schelling meshes: '4-Choose-1' Study

In this preliminary study, selections under the Schelling context corresponded to people choosing one of four shapes, each randomly sampled from a single class of shape (e.g. baskets, bottles), according to various geometric high-level groups, described in the

associated chapter (see section 4.3.1 and 4.3.2). These selections, or Schelling scores were represented as a four-dimensional, binary-valued vector (e.g. 0100), with a one denoting the selected shape. From these scores, we derived per-shape Schelling frequencies, or the fraction of times a shape is selected, given its visibility in a combination of 4 shapes (see section 4.4.2). The question of "what makes a shape Schelling salient?" was explored with the collected data. As an example, we collected data on subjective terms (e.g. visual appeal, naturalness, strangeness), and attempted to correlate Likert score ratings of shapes according to these terms, with shape Schelling frequencies (see section 4.4.3 and 4.4.4). We showed that the notion of Schelling salient meshes can be learned and we predicted Schelling scores with a voxel-based convolutional neural network, mapping 4 shapes to their Schelling score predictions (see section 4.5). Predictions can be interpreted as a weighted average of all Schelling scores for a shape (e.g. table #6 out of 25), given another 3 shapes shown nearby. We show results for several types of 3D shapes and demonstrate that the notion of Schelling saliency of meshes is useful for the applications of Schelling-based visualisation, clustering, and search (see section 4.6).

7.2.2 Schelling meshes: 'Many-Within-Class' Study

In this study, we continued our exploration of the Schelling meshes concept by collecting shape selection data in a setup where multiple shapes can be selected from a class of shapes. This generalises the group-level saliency aspect of the study, in that there are now direct comparisons between a shape and all of shapes of a class (1-to-n, instead of 1-to-m). From this data, we computed per-shape Schelling frequencies, or the fraction of times a shape is selected, given its visibility to participants (see section 5.3.1). As before, we explored what makes a shape Schelling salient. As in the previous study, we collected data on subjective terms (e.g. memorability, uniqueness, visual

appeal), attempting to correlate Likert score ratings of shapes based on these terms, with shape Schelling frequencies (see section 5.4.2 and 5.4.4). We showed that the notion of Schelling saliency of meshes can be learned via a depth image-based convolutional neural network and used to predict Schelling frequencies (see sections 5.5.1 to 5.5.3). We then attempted to predict Schelling frequencies using a range of traditional shape descriptors, but we achieved better prediction accuracy in each case, using a deep-learning approach (see section 5.5.5).

We show results for several types of 3D shapes and demonstrate that this study's interpretation of mesh Schelling saliency is useful for the applications of Schelling-based visualisation, and search. Schelling meshes are perceived as memorable and can best distinguish between the extreme shapes of a dataset. Simple dataset visualisations of shapes can be produced by plotting pictures of them, on a 1-D line according to their Schelling frequencies. t-SNE embeddings based on shape Schelling frequencies can be used to visualise 3D shapes (see section 5.4.2 and 5.6.2). Schelling frequencies can be used to search through a dataset for shapes which are considered to be memorable and/or are likely to stand out relative to the class. This is done through the absolute difference of their Schelling frequencies (see section 5.6.1). Predictive models as described above, can help people to find similar shapes, which are not necessarily within our shape datasets, but are from the same shape class – e.g. different lamps, tables etc.

7.2.3 Font Specificity Study

This study aimed to determine a measure of some aspect of human perception of fonts, via the consistency of textual descriptions associated with greyscale 2D fonts. This consistency encapsulates how Specific a font is. We collected word-level descriptions of these fonts and used them to determine two approaches to measuring Specificity: 1)

based on word frequency (see section 6.4.4), and 2) an automated approach based on pre-trained word embeddings, which represent word co-occurrence frequency in a corpus of text, or a notion of word meaning (see section 6.4.5). To explore what makes a font Specific, we collected data on subjective terms (e.g. creativity, legibility, visual appeal), attempting to correlate Likert score ratings of fonts based on these terms, with font Specificity (see section 6.4.5). Additionally, we analysed word frequencies and the types of words used to describe fonts, via their Part-Of-Speech (see section 6.4). We developed a method to predict word-embedding based font Specificity and introduce potential applications in search, visualisation and clustering (see section 6.5 and 6.6).

7.3 Summary of Findings

The findings listed below have been presented in previous chapters. Here they are displayed in a summarised form.

7.3.1 Schelling Meshes

People consider Schelling meshes to be those meshes which are more prominent and stand out with respect to other shapes in a dataset. This suggests that they can represent a dataset's extremes. They are perceived as the most memorable shapes relative to their class. Differently to that of previous work which has studied points on 3D meshes selected under a Schelling coordination game setting [6], we found that Gaussian curvature and Shape Diameter Function (SDF) values do not correlate with our per-shape Schelling saliency score – its Schelling frequency. This was also the case for the other tested descriptors (see section 5.4.3).

Overall, we found that Schelling frequencies convey different information to that of the shape descriptors. For example, visualising per-shape descriptor values as an intensity heatmap across each shape class, shows no visual patterns when the descriptors are

displayed in order of increasing Schelling frequency (see section 5.4.3). Additionally, Schelling frequencies tend to encapsulate subjective concepts associated with a 3D shape more accurately than the shape descriptors held in comparison to them.

We found that a notion of Schelling saliency in shapes can be learned, using a voxel-based convolutional neural network (see section 4.5 or 5.5.3), or a depth image-based convolutional neural network (see section 5.5.2). Significantly better prediction results were achieved in the latter case, however. We compared a selection of geometric shape descriptors to deep-learning approaches for prediction of Schelling scores, and achieved better prediction accuracy, using a deep-learning approach (see section 5.5.5).

When shape Schelling frequencies are clustered according to k-means, there is greater between-cluster variation in shape memorability ratings across a number of shape classes, than that of clusters obtained via a range of shape descriptors (see section 5.6.3). Additionally, partition agreement tests (Adjusted Rand Index and Adjusted Mutual Information) show large differences between clusterings obtained via Schelling frequencies vs. the tested shape descriptors. Geometric differences in shape also exist between clusters, given per-class clusterings based on their Schelling frequencies (see section 5.6.3).

Measures of Schelling saliency for shapes could be used to select which of a collection of mock-up 3D products or packaging is most prominent or memorable, for more effective advertising. In a more consumer-oriented example, a furniture store could provide a service which ranks a set of chairs a customer is interested in, by their perceived memorability or uniqueness. Going further, if a 3D shape of an existing chair were to be provided by the customer, it could be ranked relative to the in-store options.

7.3.2 Font Specificity

Specific fonts are the simplest and most legible fonts in a collection. Fonts start out as a mixture of many characteristics (texture, curvature, thickness) when they have low Specificity, with thinner and/or italic fonts appearing as Specificity increases. A large proportion of bold fonts have the highest Specificity. The highest Specificity fonts tend to have only one or two clear aspects to them – i.e. bold; bold and italic; italic and thin etc. Specific fonts are perceived as legible and memorable. They are also considered to be more visually-appealing, common/normal, and less creative compared to other fonts (see latter parts of section 6.4.5).

We found that font Specificity can be learned, and we predicted Specificity scores using a depth image-based convolutional neural network. The PCA-HoG shape descriptor was also shown to be a reasonably good predictor of font Specificity (see section 6.5.1).

There is some alignment in the information represented by font Specificity vs. that of the tested shape descriptors, as we reasonably found that contrast is a useful indicator for Specificity prediction (see section 6.6.1). It is not a direct indicator of Specificity however, as many fonts, both low and high in Specificity have consistent colour throughout them.

Simple dataset visualisations of fonts can be produced by plotting pictures of them, on a 1-D line according to their font Specificity scores (see section 6.4.5 and Appendix A6). t-SNE embeddings based on font Specificity, can also be used to visualise fonts (see section 6.6.1). Fonts can be searched for and compared according to the absolute difference of their Specificity scores (see section 6.6.2), to find legible fonts (high Specificity value) and creative fonts (low Specificity value).

When font Specificity scores are clustered via k-means, the clusters show greater or equal variation in subjective notions of creativity, when compared to a 512-bin PCA-compressed, Histogram of Oriented Gradients descriptor. In terms of legibility, some shape descriptors achieve similar between cluster variation (for both points, see section 6.6.3). Geometric differences in fonts also exist between clusters, given per-class clusterings based on their Specificity (see section 6.6.3 for a visual representation of the clusters).

Measures of Specificity for fonts can enable the automatic selection of readable fonts for a website or a word-processed document or assist in the selection of creative fonts for advertising, posters and billboards.

7.4 Discussion

7.4.1 Schelling Meshes – Comparison of Methodologies: Bias and Generality

From our two approaches, '4-choose-1' and 'Many-Within-Class', we collected data on different classes of 3D shape and noticed some differences, similar patterns, and clear trade-offs. Both studies were designed to collect Schelling saliency data, but different levels of bias and data quality were achieved.

The 'Many-Within-Class' study brought up a common problem in collecting survey results without some form of qualifying test. The methodology was good for reducing bias in our results, in that participants were not primed to expect a certain situation, but since there was no pre-defined goal/target, participants could not be easily guided via controls. This was different to the '4-choose-1' approach which allowed for controls, making it more experimental. But the associated bias in that study (shape high-level groups; only 4 shapes at a time) was removed in the 'Many-Within-Class' setup, making

it more exploratory (although the number of shapes to be shown to a participant at any one time, must be pre-determined).

Some bias could potentially be removed from shape high-level group selection in the '4-choose-1' approach by determining groups only based on words provided by survey participants which would describe each shape (in comments, or via a separate survey). But the '4 shapes at a time' restriction, remains.

Now that existing results have been gathered using the 'Many-Within-Class' approach, further comparison with additional shape classes is possible. Regarding Schelling saliency, we believe that this is the best approach to take for any future work involving the study of whole shapes. This is due to the consistency of the obtained results (regarding analysis and learning), relative to the '4-choose-1' approach, even though the setup is less restrictive in terms of data collection. Additionally, if shape class sizes become much larger than those in this thesis, it will likely become infeasible to collect informative data under the '4-choose-1' approach due to the increased number of possible shape permutations. The generality of the 'Many-Within-Class' approach stems from the fact that incremental results can be obtained one person at a time, given existing Schelling frequencies.

7.4.2 Modelling Approaches

Since our purpose was to create a discriminative model of Schelling saliency in 3D shapes or Specificity in fonts, we decided not to use autoencoders or generative adversarial networks in our work. Additionally, there were accuracy and computational cost reasons. Without these limitations, generation of shapes with the properties of these concepts could be an interesting research topic. We also decided to not use a residual network due to shape data not needing a very deep/abstract representation to be learned.

Lastly, we decided to not use a recurrent neural network such as an LSTM, since our shape data does not deform or transform over time. To add to this, we collected word-level descriptions of fonts rather than sentence-level descriptions. An LSTM may be useful for modelling the latter case of data, but not the former. But, our collected shape selections and font word-level descriptions can change over time, so a time-series interpretation of that data could lend to the use of recurrent networks/models for analysis. For example, given a context of the previous 5 selected shapes, present a Schelling salient shape with respect to the context.

7.5 Future Work

7.5.1 Schelling Meshes

Doing it Differently

With additional time and resources, we could produce Schelling frequencies and associated plots for new shapes in each shape class population, completely outside of the presently collected data, via predicted Schelling frequencies obtained by our convolutional network. Analysis of similarities and differences in the Schelling frequencies produced via ground-truth human selections vs. predictions, could encourage the detection of quirks or irregularities in the Schelling frequency predictions of individual shapes.

We would collect more shapes across different shape classes for finer precision and accuracy in Schelling mesh predictions (e.g. 1000+ shapes per class) and attempt to explicitly combine shape classes that are commonly found together (e.g. chairs + tables, bottles + cups) when collecting data, to see how Schelling frequencies change with the mixed stimuli.

It may be interesting to handle the case where shape classes are mixed, as in this situation, between shape class factors could have affected our results. But we have already seen commonalities between disparate shape classes such as the *abstract* and *cabinets/shelves* shapes, which other shape classes share.

Impact of Colour/Texture on Schelling Saliency

Speaking of colour, it may be the case that the colour and/or texture of a shape affects its Schelling saliency. How significant would the effect(s) be? We might be able to measure this in some way, by modifying the colour of half of a class of shapes, keeping it fixed for the remainder. On shuffling the shapes, would participants select the shapes with modified colour more or less often, than when all shapes are of the same colour? We could repeat this test for a range of colours, via crowdsourcing. The colour resolution being measured would depend on cost, however. Would certain colours exhibit larger changes in Schelling frequency? If significant differences appear, a similar test could be held for a range of colour textures. We might separately study shapes with surface-level engravings, or even embossed shapes, produced via displacement maps. Assuming that there is some significant effect, adjusting the colour or texture of a shape could be used to enhance or reduce a shape's Schelling saliency in a scene, potentially affecting how memorable it is. This could be useful in advertising applications or possibly educational scenarios for young students.

Impact of Proximity on Schelling Saliency

When looking at the '4-choose-1' and 'Many-Within-Class' studies, we can see that the former assumes that shapes are close to an observer, taking up much of their field of view. This limits the amount of shapes in view, while increasing their detail. The latter assumes the opposite, in that shapes are further away, increasing the number of shapes, while decreasing their individual detail. This detail does not seem to be necessary – it

may be that there is too much detail in the former case, to collect the best quality Schelling saliency data. But is Schelling saliency applicable at short focal distances? That is, does it work at close distances where most objects or parts of objects are in the periphery of the visual field? If not, is Schelling saliency only applicable at longer focal distances from a group of objects, where none or a minimum of objects are in the periphery? Determining an answer to this would suggest whether explicit focus on shapes within a relatively narrow field of view is required to collect Schelling saliency data, or whether some quick glances over a larger field of view, is enough.

Internal vs. External Validation of Schelling Context?

In the Schelling meshes studies presented in the thesis, we requested that participants understand that their responses needed to be given under a Schelling 'context' for them to be valid. This context was satisfied if a participant was 1) imagining potential choices of other participants, and 2) weighting and filtering these choices to select those which they believed matched with other people's selections. But, how can we know whether each participant is internally agreeing to perform these tasks, without using external comparison and analysis of data to do so? Ethical considerations must be taken into account to determine whether this should be allowed, at all. For example, is the selection device invasive or non-invasive? Is it possible to measure information about a participant that is not explicitly requested?

7.5.2 Font Specificity

Doing it Differently

Collecting more fonts (e.g. 1000+) would enhance the precision of training data used for font Specificity predictions, likely improving prediction accuracy for unseen fonts.

We would also attempt to better handle continuity in Specificity scores, as a few fonts of similar geometry have Specificity scores which are quite far apart (see Appendix for all 100 fonts ordered by word2vec based Specificity). This occurs with both the Rényi and word2vec-based Specificity measures. Doing so could reduce the number of effective outliers in the Specificity score distribution of a set of fonts.

Group-level Specificity

How can we determine the Specificity of a group of fonts? Would it be based on a uniform weighting of their Specificity scores, or would the weighting be adjusted dependent on whether a reading, writing or advertising task is at hand?

Specificity-based Font Selection for a Document

A user might want a set of fonts to be automatically selected for an existing document (e.g. a word-processed article, or a website). This would require an algorithm for the automatic determination of a set of fonts for a document, given task keywords or the meaning behind a word context (phrases or sentences in the document). For a word context, we could train a LSTM network to predict words likely associated with the context, and automatically compute their Specificity via a word embedding, taking a weighted sum of the Specificity scores (or their average), as the Specificity of the word context. A fixed context size would need to be determined, but if large enough, it could provide information on how consistent a sentence is, with consistent sentences being associated with a simpler legible font (high in Specificity). A relatively inconsistent sentence could potentially be displayed in a font with low Specificity, to make it stand out from the main body of text. Accounting for words which co-occur often (are close in the word embeddings) but have different word senses/meanings could help to improve the accuracy of results.

Impact of Colour/Texture on Specificity

The colour and texture of a font may affect its Specificity. Analysis of the types and meanings of the words associated with each font, given gradual changes in colour or texture may be interesting to see. It would be reasonable to expect increased variation in the words provided, increasing the entropy of per-font word frequency distributions, reducing each font's Specificity. But would this scale equally across all fonts, even those with different geometry? Also, would this reduction in Specificity apply to scores computed with a word embedding-based approach?

As font Specificity conveys some different information to that of shape descriptors, it may be interesting to form a multimodal approach to font/image search, using both image geometry/features and a Specificity score, each weighted by their importance to the task at hand. Clustering SIFT features of fonts tends to cluster fonts by their contrast with the background. Weighting Specificity slightly higher than SIFT features and selecting descriptors with values in a certain bound, could ensure that legible fonts of specific contrast are selected. A similar multimodal approach to image saliency based on image annotations and textual descriptions has been attempted [191].

Processing Fluency and Text Memorisation

Processing fluency is the ease with which information is processed. *Perceptual fluency* is the ease of processing stimuli based on manipulations to perceptual quality, whereas *retrieval fluency* is the ease with which information can be retrieved from memory [275].

Judgements of learning (JOL) are a person's estimate as to how well they have learned something. In 1991, Nelson and Dunlosky discovered that JOLs were highly accurate when delayed slightly, rather than being immediately made. This suggested that it took

time for new concepts to move from short-term memory to long-term memory. If tested immediately, a person may perceive to have learned some concept, but quickly forget some aspects of it when moving onto a new task. Testing their longer-term memory can therefore be more accurate [276]. Similarly, if we repeat some text in a paragraph, quickly after the initial text, we are inclined to skip it as we think we have acquired the information already. So, we do not learn anything new.

When showing repeated text to a subject, without delay or with minimal delay, we could ask whether the inconsistency of fonts used to show each piece of repeated text encourages an accurate judgement of learning to be made more quickly, relative to displaying all text in a single font. Could Specificity be a measure of the perceptual fluency of each piece of text? If so, some distance measure based on the text's Specificity could reflect how easy the text is to learn and internalise.

For example, "The quick brown fox jumps over the lazy dog" could be shown in two different fonts (keeping one font fixed, while varying the other), repeated with only a short gap between the sentences. But each font would be different, and some distance apart with respect to Specificity:

E.g.

"The quick brown fox jumps over the lazy dog"

"The quick brown fox jumps over the lazy dog"

Would a subject recall the initial text more accurately, as the Specificity-based distance between the two fonts increases? If true, this may be useful for learning natural language-based concepts, in an educational setting. Given a fixed initial font, we could vary the latter font, to determine how recall times vary with Specificity, if at all.

We might also attempt to increase the space between sentences or request a specified time delay before a response to the font stimuli, to see if recall in each case differs – possibly according to the delayed JOL effect [276].

Many works have attempted to adjust the perceptual fluency of text. One study of children's learning and comprehension found that modifying the perceptual properties of words (font sizes, line lengths and line spacing) provided to children, affects their comprehension, but differently across different ages [277]. Other work has discovered that larger font sizes can affect JOLs, which in some way increases perceptual fluency, even though font size does not affect retention of the information provided [278].

Another study looked at whether disfluency of text influences learning, either perceptually (by making the text less legible), or lexically (via scrambling letters in the text) [279]. The authors tested the recall performance of 134 subjects, immediately after viewing the text, and when delayed by 2 weeks. They found that an illegible font "improved long-term recall by decreasing forgetting", whereas scrambled letters "reduced short-term recall but tended to aid remembering" [279]. From this, we can hypothesize that low Specificity fonts may be useful in aiding text memorisation. If this is the case, it may be worthwhile to see if this still holds when fonts are pre-categorised into plain text (original font), bold and italic, as these are common manipulations of fonts used in word processing. Other work has shown that displaying text in a disfluent manner can reduce confirmation bias, as it promotes more careful analysis of the text [280].

7.6 Conclusion

Overall, we have found that the Schelling meshes and font Specificity concepts both represent similarities between their respective type of shape, even when there are

discontinuities between the shape geometries themselves. This agrees with recent results obtained via the collection, analysis and modelling of human fixations on 3D printed shapes [102]. In this case, the authors found that the accuracy of saliency predictions increased across all shapes, given an unseen shape. This indicated some kind of high-level feature(s) common across many shapes. For our human-perceptual approach, the ‘context’ of these similarities is in some kind of abstract or subjective meaning which is consistent among different people. Some simple examples are that of the Likert criteria tested for in the thesis. Statistical comparisons of cluster assignments relative to shape descriptors for each form of shape, show little agreement, indicating the presence of different information. It may be the case that other consistent factors beyond those discovered, influence Schelling saliency in 3D shapes and Specificity in fonts. Looking ahead, it may also be interesting to apply these concepts to other domains, such as Schelling saliency to images, or Specificity to 3D shapes.

References

[1] L. Power and M. C. M. Lau, “Schelling Meshes,” 2017.

[2] “Trimble 3D Warehouse,” [Online]. Available: https://3dwarehouse.sketchup.com/index.html. [Accessed Jul 2016].

[3] O. Vartanian, G. Navarrete, A. Chatterjee, L. B. Fich, H. Leder, C. Modroño, M. Nadal, N. Rostrup and M. Skov, “Impact of contour on aesthetic judgments and approach-avoidance decisions in architecture,” *Proceedings of the National Academy of Sciences,* vol. 110, pp. 10446-10453, 2013.

[4] M. Gambino, “Do Our Brains Find Certain Shapes More Attractive Than Others,” *Smithsonian. com,* 2013.

[5] T. C. Schelling, The strategy of conflict, Harvard university press, 1980.

[6] X. Chen, A. Saparov, B. Pang and T. Funkhouser, “Schelling points on 3D surface meshes,” *ACM Transactions on Graphics (TOG),* vol. 31, p. 29, 2012.

[7] M. Jas and D. Parikh, “Image specificity,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[8] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su and others, “Shapenet: An information-rich 3d model repository,” *arXiv preprint arXiv:1512.03012,* 2015.

[9] R. Osada, T. Funkhouser, B. Chazelle and D. Dobkin, “Matching 3D models with shape distributions,” in *Shape Modeling and Applications, SMI 2001 International Conference On.*, 2001.

[10] R. Osada, T. Funkhouser, B. Chazelle and D. Dobkin, “Shape distributions,” *ACM Transactions on Graphics (TOG),* vol. 21, pp. 807-832, 2002.

[11] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005.

[12] “Scikit-Image,” May 2016. [Online]. Available: http://scikit-image.org/docs/0.10.x/auto_examples/plot_hog.html. [Accessed September 2018].

[13] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision,* vol. 60, pp. 91-110, 2004.

[14] J. T. Pedersen, “Study group SURF: Feature detection & description,” *Department of Computer Science, Aarhus University,* 2011.

[15] H. Bay, T. Tuytelaars and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*, 2006.

[16] A. Alahi, R. Ortiz and P. Vandergheynst, “Freak: Fast retina keypoint,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[17] I. Sobel and G. Feldman, “A 3x3 isotropic gradient operator for image processing,” *a talk at the Stanford Artificial Project in,* pp. 271-272, 1968.

[18] H. Su, S. Maji, E. Kalogerakis and E. G. Learned-Miller, “Multi-view convolutional neural networks for 3d shape recognition,” in *Proc. ICCV*, 2015.

[19] R. Hull, “Wireframes Gallery,” April 2014. [Online]. Available: https://github.com/rm-hull/wireframes/ blob/master/GALLERY.md. [Accessed September 2018].

[20] “Point Cloud Interiors,” January 2013. [Online]. Available: https://theinteriorprospect.blogspot.com/ 2013/01/point-cloud-interiors.html). [Accessed September 2018].

[21] J. J. Hasbestan and I. Senocak, “Binarized-octree generation for Cartesian adaptive mesh refinement around immersed geometries,” *Journal of Computational Physics,* vol. 368, pp. 179-195, 2018.

[22] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3D surface construction algorithm,” in *ACM siggraph computer graphics*, 1987.

[23] S. Laine and T. Karras, “Efficient sparse voxel octrees,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 17, pp. 1048-1059, 2011.

[24] V. Kämpe, E. Sintorn and U. Assarsson, “High resolution sparse voxel DAGs,” *ACM Transactions on Graphics (TOG),* vol. 32, p. 101, 2013.

[25] B. R. Williams, “Moxel DAGs: Connecting material information to high resolution sparse voxel DAGs,” 2015.

[26] B. Dado, T. R. Kol, P. Bauszat, J.-M. Thiery and E. Eisemann, “Geometry and Attribute Compression for Voxel Scenes,” in *Computer Graphics Forum*, 2016.

[27] A. J. Villanueva, F. Marton and E. Gobbetti, “SSVDAGs: Symmetry-aware sparse voxel DAGs,” in *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2016.

[28] “Principal Curvature,” March 2018. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/e/eb/Minimal_surface_curvature_planes-en.svg. [Accessed June 2018].

[29] G. E. Sorkine Olga, “Discrete Differential Geometry (CS 6501) Slides,” February 2014. [Online]. Available: http://www.connellybarnes.com/work/class/2013/shape/06_ddg_surfaces.pptx. [Accessed June 2018].

[30] K. Crane, F. De Goes, M. Desbrun and P. Schröder, “Digital geometry processing with discrete exterior calculus,” in *ACM SIGGRAPH 2013 Courses*, 2013.

[31] L. Shapira, A. Shamir and D. Cohen-Or, “Consistent mesh partitioning and skeletonisation using the shape diameter function,” *The Visual Computer,* vol. 24, p. 249, 2008.

[32] R. A. Fisher, “On the interpretation of χ 2 from contingency tables, and the calculation of P,” *Journal of the Royal Statistical Society,* vol. 85, pp. 87-94, 1922.

[33] A. Agresti and others, “A survey of exact inference for contingency tables,” *Statistical science,* vol. 7, pp. 131-153, 1992.

[34] K. Pearson, “X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,* vol. 50, pp. 157-175, 1900.

[35] P. E. Greenwood and M. S. Nikulin, A guide to chi-squared testing, vol. 280, John Wiley & Sons, 1996.

[36] F. J. Massey Jr, “The Kolmogorov-Smirnov test for goodness of fit,” *Journal of the American statistical Association,* vol. 46, pp. 68-78, 1951.

[37] R. A. Fisher, “Statistical methods for research workers,” in *Breakthroughs in statistics*, Springer, 1992, pp. 66-70.

[38] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007.

[39] K. Pearson, “LIII. On lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,* vol. 2, pp. 559-572, 1901.

[40] H. Hotelling, “Analysis of a complex of statistical variables into principal components.,” *Journal of educational psychology,* vol. 24, p. 417, 1933.

[41] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of machine learning research,* vol. 9, pp. 2579-2605, 2008.

[42] R. Hecht-Nielsen, “Neural network primer: part i,” *AI Expert,* pp. 4-51, 1989.

[43] M. Caudill, “Neural networks primer, part I,” *AI expert,* vol. 2, pp. 46-52, 1987.

[44] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat's visual cortex,” *The Journal of physiology,* vol. 160, pp. 106-154, 1962.

[45] LISA. [Online]. Available: http://deeplearning.net/ tutorial/lenet.html. [Accessed Jan 2016].

[46] G. Li and Y. Yu, “Visual saliency based on multiscale deep features,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[47] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang and J. Xiao, “3D shapenets: A deep representation for volumetric shapes,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[48] D. Maturana and S. Scherer, “Voxnet: A 3d convolutional neural network for real-time object recognition,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, 2015.

[49] H. Su, S. Maji, E. Kalogerakis and E. Learned-Miller, “Multi-view convolutional neural networks for 3d shape recognition,” in *Proceedings of the IEEE international conference on computer vision*, 2015.

[50] Z. Zhu, X. Wang, S. Bai, C. Yao and X. Bai, “Deep learning representation using autoencoder for 3D shape retrieval,” *Neurocomputing,* vol. 204, pp. 41-50, 2016.

[51] B. Leng, S. Guo, X. Zhang and Z. Xiong, “3D object retrieval with stacked local convolutional autoencoder,” *Signal Processing,* vol. 112, pp. 119-128, 2015.

[52] F. A. Gers, J. Schmidhuber and F. Cummins, “Learning to forget: Continual prediction with LSTM,” 1999.

[53] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende and D. Wierstra, “Draw: A recurrent neural network for image generation,” *arXiv preprint arXiv:1502.04623,* 2015.

[54] S. M. A. Eslami, N. Heess, C. K. I. Williams and J. Winn, “The shape boltzmann machine: a strong model of object shape,” *International Journal of Computer Vision,* vol. 107, pp. 155-176, 2014.

[55] B. Leng, X. Zhang, M. Yao and Z. Xiong, “A 3D model recognition mechanism based on deep Boltzmann machines,” *Neurocomputing,* vol. 151, pp. 593-602, 2015.

[56] Z. Liu, S. Chen, S. Bu and K. Li, “High-level semantic feature for 3D shape based on deep belief networks,” in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, 2014.

[57] B. Leng, X. Zhang, M. Yao and Z. Xiong, “3D object classification using deep belief networks,” in *International Conference on Multimedia Modeling*, 2014.

[58] P. O'Donovan, J. Lībeks, A. Agarwala and A. Hertzmann, "Exploratory font selection using crowdsourced attributes," *ACM Transactions on Graphics (TOG),* vol. 33, p. 92, 2014.

[59] E. Garces, A. Agarwala, D. Gutierrez and A. Hertzmann, "A similarity measure for illustration style," *ACM Transactions on Graphics (TOG),* vol. 33, p. 93, 2014.

[60] A. Bellet, A. Habrard and M. Sebban, "A survey on metric learning for feature vectors and structured data," *arXiv preprint arXiv:1306.6709,* 2013.

[61] J. Camacho-Collados and T. Pilehvar, "From Word to Sense Embeddings: A Survey on Vector Representations of Meaning," *arXiv preprint arXiv:1805.04032,* 2018.

[62] S. T. Dumais, "Latent semantic analysis," *Annual review of information science and technology,* vol. 38, pp. 188-230, 2004.

[63] J. Pennington, R. Socher and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[64] T. Mikolov, K. Chen, G. Corrado, J. Dean, L. Sutskever and G. Zweig, "word2vec," *URL https://code. google. com/p/word2vec,* 2013.

[65] Y. Goldberg and O. Levy, "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722,* 2014.

[66] K. Xu, V. G. Kim, Q. Huang, N. Mitra and E. Kalogerakis, "Data-driven shape analysis and processing," in *SIGGRAPH ASIA 2016 Courses*, 2016.

[67] T. Surazhsky, E. Magid, O. Soldea, G. Elber and E. Rivlin, "A comparison of gaussian and mean curvatures estimation methods on triangular meshes," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, 2003.

[68] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015.

[69] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[70] X. Li, L. Zhao, L. Wei, M.-H. Yang, F. Wu, Y. Zhuang, H. Ling and J. Wang, "Deepsaliency: Multi-task deep neural network model for salient object detection," *IEEE Transactions on Image Processing,* vol. 25, pp. 3919-3930, 2016.

[71] O. E. Dictionary, "OED online," *Oxford University Press. http://www. oed. com, Accessed Nov,* vol. 30, p. 2006, 1989.

[72] A. Borji and L. Itti, “State-of-the-art in visual attention modeling,” *IEEE transactions on pattern analysis and machine intelligence,* vol. 35, pp. 185-207, 2013.

[73] L. Itti, C. Koch and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Transactions on pattern analysis and machine intelligence,* vol. 20, pp. 1254-1259, 1998.

[74] R. Bernardes, J. Dias and J. Cunha-Vaz, “Mapping the human blood-retinal barrier function,” *IEEE Transactions on Biomedical Engineering,* vol. 52, pp. 106-116, 2005.

[75] J. H. Reynolds and R. Desimone, “Interacting roles of attention and visual salience in V4,” *Neuron,* vol. 37, pp. 853-863, 2003.

[76] C. Koch and S. Ullman, “Shifts in selective visual attention: towards the underlying neural circuitry,” in *Matters of intelligence*, Springer, 1987, pp. 115-141.

[77] T. Liu, J. Sun, N.-N. Zheng, X. Tang and H.-Y. Shum, “Learning to detect a salient object,” in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007.

[78] A. Borji, M.-M. Cheng, H. Jiang and J. Li, “Salient Object Detection: A Survey,” *CoRR,* vol. abs/1411.5878, 2014.

[79] A. Borji, M.-M. Cheng, H. Jiang and J. Li, “Salient object detection: A benchmark,” *IEEE transactions on image processing,* vol. 24, pp. 5706-5722, 2015.

[80] A. Borji, “What is a salient object? A dataset and a baseline model for salient object detection,” *IEEE Transactions on Image Processing,* vol. 24, pp. 742-756, 2015.

[81] T. Judd, K. Ehinger, F. Durand and A. Torralba, “Learning to predict where humans look,” in *Computer Vision, 2009 IEEE 12th international conference on*, 2009.

[82] L. Elazary and L. Itti, “Interesting objects are visually salient,” *Journal of vision,* vol. 8, pp. 3-3, 2008.

[83] K. Koehler, F. Guo, S. Zhang and M. P. Eckstein, “What do saliency models predict?,” *Journal of vision,* vol. 14, pp. 14-14, 2014.

[84] A. Borji, D. N. Sihite and L. Itti, “What stands out in a scene? A study of human explicit saliency judgment,” *Vision research,* vol. 91, pp. 62-77, 2013.

[85] C. Yang, L. Zhang, H. Lu, X. Ruan and M.-H. Yang, “Saliency detection via graph-based manifold ranking,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013.

[86] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013.

[87] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 37, pp. 569-582, 2015.

[88] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012.

[89] K. Simonyan, A. Vedaldi and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034,* 2013.

[90] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[91] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence,* vol. 40, pp. 834-848, 2018.

[92] P. Zhang, D. Wang, H. Lu, H. Wang and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[93] T. Wang, A. Borji, L. Zhang, P. Zhang and H. Lu, "A stagewise refinement model for detecting salient objects in images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[94] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol and X. Giro-i-Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," *arXiv preprint arXiv:1701.01081,* 2017.

[95] R. Monroy, S. Lutz, T. Chalasani and A. Smolic, "SalNet360: Saliency Maps for omni-directional images with CNN," *Signal Processing: Image Communication,* 2018.

[96] S. S. S. Kruthiventi, K. Ayush and R. V. Babu, "Deepfix: A fully convolutional neural network for predicting human eye fixations," *IEEE Transactions on Image Processing,* vol. 26, pp. 4446-4456, 2017.

[97] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Trans. Image Process,* vol. 27, pp. 2368-2378, 2018.

[98] J. Ling, K. Zhang, Y. Zhang, D. Yang and Z. Chen, "A saliency prediction model on 360 degree images using color dictionary based sparse representation," *Signal Processing: Image Communication,* 2018.

[99] M. Assens, X. Giro-i-Nieto, K. McGuinness and N. E. O'Connor, "Saltinet: Scan-path prediction on 360 degree images using saliency volumes," in *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, 2017.

[100] J. Gutiérrez, E. David, Y. Rai and P. Le Callet, "Toolbox and dataset for the development of saliency and scanpath models for omnidirectional/360° still images," *Signal Processing: Image Communication,* 2018.

[101] Y. Zhu, G. Zhai and X. Min, "The prediction of head and eye movement for 360 degree images," *Signal Processing: Image Communication,* 2018.

[102] X. Wang, S. Koch, K. Holmqvist and M. Alexa, "Tracking the gaze on objects in 3D: how do people really look at the bunny?," in *SIGGRAPH Asia 2018 Technical Papers*, 2018.

[103] L. Liu, R. Chen, L. Wolf and D. Cohen-Or, "Optimizing photo composition," in *Computer Graphics Forum*, 2010.

[104] R. Margolin, L. Zelnik-Manor and A. Tal, "Saliency for image manipulation," *The Visual Computer,* vol. 29, pp. 381-392, 2013.

[105] D. E. Jacobs, D. B. Goldman and E. Shechtman, "Cosaliency: Where people look when comparing images," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 2010.

[106] H. Li and K. N. Ngan, "A co-saliency model of image pairs," *IEEE Transactions on Image Processing,* vol. 20, pp. 3365-3375, 2011.

[107] H. Fu, X. Cao and Z. Tu, "Cluster-based co-saliency detection," *IEEE Transactions on Image Processing,* vol. 22, pp. 3766-3778, 2013.

[108] Z. Liu, W. Zou, L. Li, L. Shen and O. Le Meur, "Co-saliency detection based on hierarchical segmentation," *IEEE Signal Process. Lett,* vol. 21, pp. 88-92, 2014.

[109] D. Zhang, J. Han, C. Li, J. Wang and X. Li, "Detection of co-salient objects by looking deep and wide," *International Journal of Computer Vision,* vol. 120, pp. 215-232, 2016.

[110] J. Han, G. Cheng, Z. Li and D. Zhang, "A unified metric learning-based framework for co-saliency detection," *IEEE Transactions on Circuits and Systems for Video Technology,* 2017.

[111] J. Han, R. Quan, D. Zhang and F. Nie, "Robust object co-segmentation using background prior," *IEEE Transactions on Image Processing,* vol. 27, pp. 1639-1651, 2018.

[112] M.-M. Cheng, N. J. Mitra, X. Huang and S.-M. Hu, "Salientshape: Group saliency in image collections," *The Visual Computer,* vol. 30, pp. 443-453, 2014.

[113] J. Wu, X. Shen, W. Zhu and L. Liu, "Mesh saliency with global rarity," *Graphical Models,* vol. 75, pp. 255-264, 2013.

[114] R. Song, Y. Liu, R. R. Martin and P. L. Rosin, "Mesh saliency via spectral processing," *ACM Transactions on Graphics (TOG),* vol. 33, p. 6, 2014.

[115] J. Hu and J. Hua, "Salient spectral geometric features for shape matching and retrieval," *The visual computer,* vol. 25, pp. 667-675, 2009.

[116] M. R. Ruggeri, G. Patanè, M. Spagnuolo and D. Saupe, "Spectral-driven isometry-invariant matching of 3D shapes," *International Journal of Computer Vision,* vol. 89, pp. 248-265, 2010.

[117] R. Song, Y. Liu, R. R. Martin and P. L. Rosin, "3D point of interest detection via spectral irregularity diffusion," *The Visual Computer,* vol. 29, pp. 695-705, 2013.

[118] M. Limper, A. Kuijper and D. W. Fellner, "Mesh saliency analysis via local curvature entropy," in *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Short Papers*, 2016.

[119] H. Fadaifard and G. Wolberg, "Multiscale 3D feature extraction and matching," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, 2011.

[120] S. Jia, C. Zhang, X. Li and Y. Zhou, "Mesh resizing based on hierarchical saliency detection," *Graphical Models,* vol. 76, pp. 355-362, 2014.

[121] J. Sun, M. Ovsjanikov and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Computer graphics forum*, 2009.

[122] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010.

[123] N. J. Mitra, L. J. Guibas and M. Pauly, "Symmetrization," in *ACM Transactions on Graphics (TOG)*, 2007.

[124] D. Raviv, A. M. Bronstein, M. M. Bronstein and R. Kimmel, "Full and partial symmetries of non-rigid shapes," *International journal of computer vision,* vol. 89, pp. 18-39, 2010.

[125] N. J. Mitra, M. Pauly, M. Wand and D. Ceylan, "Symmetry in 3d geometry: Extraction and applications," in *Computer Graphics Forum*, 2013.

[126] M. Ovsjanikov, J. Sun and L. Guibas, "Global intrinsic symmetries of shapes," in *Computer graphics forum*, 2008.

[127] A. M. Bronstein, M. M. Bronstein and R. Kimmel, "Topology-invariant similarity of nonrigid shapes," *International journal of computer vision,* vol. 81, p. 281, 2009.

[128] K. Xu, H. Zhang, A. Tagliasacchi, L. Liu, G. Li, M. Meng and Y. Xiong, "Partial intrinsic reflectional symmetry of 3D shapes," in *ACM Transactions on Graphics (TOG)*, 2009.

[129] K. Xu, H. Zhang, W. Jiang, R. Dyer, Z. Cheng, L. Liu and B. Chen, "Multi-scale partial intrinsic symmetry detection," *ACM Transactions on Graphics (TOG),* vol. 31, p. 181, 2012.

[130] N. J. Mitra, L. J. Guibas and M. Pauly, "Partial and approximate symmetry detection for 3D geometry," *ACM Transactions on Graphics (TOG),* vol. 25, pp. 560-568, 2006.

[131] X. Liu, L. Liu, W. Song, Y. Liu and L. Ma, "Shape context based mesh saliency detection and its applications: A survey," *Computers & Graphics,* vol. 57, pp. 12-30, 2016.

[132] F. Ponjou Tasse, J. Kosinka and N. Dodgson, "Cluster-based point set saliency," in *Proceedings of the IEEE international conference on computer vision*, 2015.

[133] R. Song, Y. Liu, Y. Zhao, R. R. Martin, P. L. Rosin and others, "Conditional random field-based mesh saliency.," in *ICIP*, 2012.

[134] S. Wang, N. Li, S. Li, Z. Luo, Z. Su and H. Qin, "Multi-scale mesh saliency based on low-rank and sparse analysis in shape feature space," *Computer Aided Geometric Design,* vol. 35, pp. 206-214, 2015.

[135] C. H. Lee, A. Varshney and D. W. Jacobs, "Mesh saliency," in *ACM transactions on graphics (TOG)*, 2005.

[136] R. Gal and D. Cohen-Or, "Salient geometric features for partial shape matching and similarity," *ACM Transactions on Graphics (TOG),* vol. 25, pp. 130-150, 2006.

[137] P. Shilane and T. Funkhouser, "Distinctive regions of 3D surfaces," *ACM Transactions on Graphics (TOG),* vol. 26, p. 7, 2007.

[138] A. Shamir, "A survey on mesh segmentation techniques," in *Computer graphics forum*, 2008.

[139] X. Chen, A. Golovinskiy and T. Funkhouser, "A benchmark for 3D mesh segmentation," in *Acm transactions on graphics (tog)*, 2009.

[140] E. Kalogerakis, A. Hertzmann and K. Singh, "Learning 3D mesh segmentation and labeling," *ACM Transactions on Graphics (TOG),* vol. 29, p. 102, 2010.

[141] W. Benjamin, A. W. Polk, S. V. N. Vishwanathan and K. Ramani, "Heat walk: Robust salient segmentation of non-rigid shapes," in *Computer Graphics Forum*, 2011.

[142] O. Van Kaick, H. Zhang, G. Hamarneh and D. Cohen-Or, "A survey on shape correspondence," in *Computer Graphics Forum*, 2011.

[143] E. Shtrom, G. Leifman and A. Tal, "Saliency detection in large point sets," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[144] R. B. Rusu, N. Blodow and M. Beetz, “Fast point feature histograms (FPFH) for 3D registration,” in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 2009.

[145] H. Dutagaci, C. P. Cheung and A. Godil, “Evaluation of 3D interest point detection techniques via human-generated ground truth,” *The Visual Computer,* vol. 28, pp. 901-917, 2012.

[146] F. P. Tasse, J. Kosinka and N. A. Dodgson, “Quantitative analysis of saliency models,” in *SIGGRAPH ASIA 2016 Technical Briefs*, 2016.

[147] D. Giorgi, S. Biasotti and L. Paraboschi, “Shape retrieval contest 2007: Watertight models track,” *SHREC competition,* vol. 8, 2007.

[148] M. Lau, K. Dev, W. Shi, J. Dorsey and H. Rushmeier, “Tactile mesh saliency,” *ACM Transactions on Graphics (TOG),* vol. 35, p. 52, 2016.

[149] X. Wang, D. Lindlbauer, C. Lessig, M. Maertens and M. Alexa, “Measuring the visual salience of 3d printed objects,” *IEEE computer graphics and applications,* vol. 36, pp. 46-55, 2016.

[150] S. Howlett, J. Hamill and C. O'Sullivan, “Predicting and evaluating saliency for simplified polygonal models,” *ACM Transactions on Applied Perception (TAP),* vol. 2, pp. 286-308, 2005.

[151] Y. Kim, A. Varshney, D. W. Jacobs and F. Guimbretière, “Mesh saliency and human eye fixations,” *ACM Transactions on Applied Perception (TAP),* vol. 7, p. 12, 2010.

[152] A. Secord, J. Lu, A. Finkelstein, M. Singh and A. Nealen, “Perceptual models of viewpoint preference,” *ACM Transactions on Graphics (TOG),* vol. 30, p. 109, 2011.

[153] X. Zhang, X. Le, A. Panotopoulou, E. Whiting and C. C. L. Wang, “Perceptual models of preference in 3d printing direction,” *ACM Transactions on Graphics (TOG),* vol. 34, p. 215, 2015.

[154] M. Savva, A. X. Chang, G. Bernstein, C. D. Manning and P. Hanrahan, “On being the right scale: Sizing large collections of 3D models,” in *SIGGRAPH Asia 2014 Indoor Scene Understanding Where Graphics Meets Vision*, 2014.

[155] A. Jain, T. Thormählen, T. Ritschel and H.-P. Seidel, “Material memex: Automatic material suggestions for 3d objects,” *ACM Transactions on Graphics (TOG),* vol. 31, p. 143, 2012.

[156] A. Saxena, S. H. Chung and A. Y. Ng, “Learning depth from single monocular images,” in *Advances in neural information processing systems*, 2006.

[157] P. Vangorp, J. Laurijssen and P. Dutré, “The influence of shape on the perception of material reflectance,” in *ACM Transactions on Graphics (TOG)*, 2007.

[158] Z. Pizlo, 3D shape: Its unique place in visual perception, Mit Press, 2010.

[159] N. J. Mitra, M. Wand, H. Zhang, D. Cohen-Or, V. Kim and Q.-X. Huang, "Structure-aware shape processing," in *ACM SIGGRAPH 2014 Courses*, 2014.

[160] V. G. Kim, S. Chaudhuri, L. Guibas and T. Funkhouser, "Shape2pose: Human-centric shape analysis," *ACM Transactions on Graphics (TOG),* vol. 33, p. 120, 2014.

[161] H. Laga, M. Mortara and M. Spagnuolo, "Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes," *ACM Transactions on Graphics (TOG),* vol. 32, p. 150, 2013.

[162] R. Hu, C. Zhu, O. Kaick, L. Liu, A. Shamir and H. Zhang, "Interaction context (ICON): towards a geometric functionality descriptor," *ACM Transactions on Graphics (TOG),* vol. 34, p. 83, 2015.

[163] D.-Y. Chen, X.-P. Tian, Y.-T. Shen and M. Ouhyoung, "On visual similarity based 3D model retrieval," in *Computer graphics forum*, 2003.

[164] N. Sedaghat, M. Zolfaghari, E. Amiri and T. Brox, "Orientation-boosted voxel nets for 3D object recognition," *arXiv preprint arXiv:1604.03351,* 2016.

[165] P. Shilane, P. Min, M. Kazhdan and T. Funkhouser, "The princeton shape benchmark," in *Proceedings Shape Modeling Applications, 2004.*, 2004.

[166] A. Kanezaki, "Rotationnet: Learning object classification using unsupervised viewpoint estimation," *arXiv preprint arXiv:1603.06208,* 2016.

[167] Z. Liu, J. Zhang and L. Liu, "Upright orientation of 3D shapes with Convolutional Networks," *Graphical Models,* vol. 85, pp. 22-29, 2016.

[168] A. Sharma, O. Grau and M. Fritz, "Vconv-dae: Deep volumetric shape learning without object labels," in *European Conference on Computer Vision*, 2016.

[169] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014.

[170] A. Dosovitskiy, J. Tobias Springenberg and T. Brox, "Learning to generate chairs with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[171] J. Wu, C. Zhang, T. Xue, B. Freeman and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *Advances in Neural Information Processing Systems*, 2016.

[172] A. Radford, L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434,* 2015.

[173] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing

generative adversarial nets," in *Advances in neural information processing systems*, 2016.

[174] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in Neural Information Processing Systems*, 2016.

[175] T. D. Kulkarni, W. F. Whitney, P. Kohli and J. Tenenbaum, "Deep convolutional inverse graphics network," in *Advances in neural information processing systems*, 2015.

[176] D. J. Rezende, S. M. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg and N. Heess, "Unsupervised learning of 3d structure from images," in *Advances in Neural Information Processing Systems*, 2016.

[177] H. Huang, E. Kalogerakis and B. Marlin, "Analysis and synthesis of 3D shape families via deep-learned generative models of surfaces," in *Computer Graphics Forum*, 2015.

[178] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Transactions on Graphics (TOG),* vol. 34, p. 98, 2015.

[179] L. Wei, Q. Huang, D. Ceylan, E. Vouga and H. Li, "Dense human body correspondences using convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[180] I. Lim, A. Gehre and L. Kobbelt, "Identifying style of 3D shapes using deep metric learning," in *Computer Graphics Forum*, 2016.

[181] L. Yi, L. Guibas, A. Hertzmann, V. G. Kim, H. Su and E. Yumer, "Learning Hierarchical Shape Segmentation and Labeling from Online Repositories," *SIGGRAPH,* 2017.

[182] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun and X. Tong, "O-cnn: Octree-based convolutional neural networks for 3d shape analysis," *ACM Transactions on Graphics (TOG),* vol. 36, p. 72, 2017.

[183] D. Holden, J. Saito and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Transactions on Graphics (TOG),* vol. 35, p. 138, 2016.

[184] C. Goldfeder and P. Allen, "Autotagging to improve text search for 3d models," in *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, 2008.

[185] S. Chaudhuri, E. Kalogerakis, S. Giguere and T. Funkhouser, "Attribit: content creation with semantic attributes," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, 2013.

[186] A. M. Bronstein, M. M. Bronstein, L. J. Guibas and M. Ovsjanikov, “Shape google: Geometric words and expressions for invariant shape retrieval,” *ACM Transactions on Graphics (TOG),* vol. 30, p. 1, 2011.

[187] S. Streuber, M. A. Quiros-Ramirez, M. Q. Hill, C. A. Hahn, S. Zuffi, A. O'Toole and M. J. Black, “Body talk: crowdshaping realistic 3D avatars with words,” *ACM Transactions on Graphics (TOG),* vol. 35, p. 54, 2016.

[188] L. Von Ahn and L. Dabbish, “Labeling images with a computer game,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004.

[189] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[190] R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat and B. Plank, “Automatic description generation from images: A survey of models, datasets, and evaluation measures,” *Journal of Artificial Intelligence Research,* vol. 55, pp. 409-442, 2016.

[191] H. Liang, M. Jiang, R. Liang and Q. Zhao, “CapVis: Toward Better Understanding of Visual-Verbal Saliency Consistency,” *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 10, p. 10, 2018.

[192] A. Deshpande, J. Aneja, L. Wang, A. Schwing and D. A. Forsyth, “Diverse and Controllable Image Captioning with Part-of-Speech Guidance,” *arXiv preprint arXiv:1805.12589,* 2018.

[193] T. Pedersen, S. Patwardhan and J. Michelizzi, “WordNet:: Similarity: measuring the relatedness of concepts,” in *Demonstration papers at HLT-NAACL 2004*, 2004.

[194] G. K. Zipf, Human behavior and the principle of least effort: An introduction to human ecology, Ravenio Books, 2016.

[195] T. Liu, A. Hertzmann, W. Li and T. Funkhouser, “Style compatibility for 3D furniture models,” *ACM Transactions on Graphics (TOG),* vol. 34, p. 85, 2015.

[196] Z. Lun, E. Kalogerakis and A. Sheffer, “Elements of style: learning perceptual shape style similarity,” *ACM Transactions on Graphics (TOG),* vol. 34, p. 84, 2015.

[197] B. Saleh, M. Dontcheva, A. Hertzmann and Z. Liu, “Learning style similarity for searching infographics,” in *Proceedings of the 41st graphics interface conference*, 2015.

[198] T. Joachims, “Training linear SVMs in linear time,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006.

[199] O. Chapelle and S. S. Keerthi, "Efficient algorithms for ranking with SVMs," *Information retrieval,* vol. 13, pp. 201-215, 2010.

[200] Y. LeCun, Y. Bengio and others, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks,* vol. 3361, p. 1995, 1995.

[201] C. Szegedy, S. Ioffe, V. Vanhoucke and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, 2017.

[202] C. Zhou, C. Sun, Z. Liu and F. Lau, "A C-LSTM neural network for text classification," *arXiv preprint arXiv:1511.08630,* 2015.

[203] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

[204] D. Tang, B. Qin, X. Feng and T. Liu, "Effective LSTMs for target-dependent sentiment classification," *arXiv preprint arXiv:1512.01100,* 2015.

[205] H. Chen, M. Sun, C. Tu, Y. Lin and Z. Liu, "Neural sentiment classification with user and product attention," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[206] D. Tang, B. Qin and T. Liu, "Aspect level sentiment classification with deep memory network," *arXiv preprint arXiv:1605.08900,* 2016.

[207] I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014.

[208] D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473,* 2014.

[209] G. Klein, Y. Kim, Y. Deng, J. Senellart and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:1701.02810,* 2017.

[210] D. Britz, A. Goldie, M.-T. Luong and Q. Le, "Massive exploration of neural machine translation architectures," *arXiv preprint arXiv:1703.03906,* 2017.

[211] F. Zhang, B. Du and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 53, pp. 2175-2184, 2015.

[212] R. Socher, B. Huval, B. Bath, C. D. Manning and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Advances in neural information processing systems*, 2012.

[213] Y. Hechtlinger, P. Chakravarti and J. Qin, "A generalization of convolutional neural networks to graph-structured data," *arXiv preprint arXiv:1704.08165,* 2017.

[214] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam and P. Vandergheynst, "Geometric deep learning: going beyond Euclidean data," *CoRR,* vol. abs/1611.08097, 2016.

[215] D. Boscaini, J. Masci, E. Rodolà and M. M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," *CoRR,* vol. abs/1605.06437, 2016.

[216] C. R. Qi, H. Su, K. Mo and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE,* vol. 1, p. 4, 2017.

[217] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[218] P. Guerrero, Y. Kleiman, M. Ovsjanikov and N. J. Mitra, "PCPNet Learning Local Shape Properties from Raw Point Clouds," in *Computer Graphics Forum*, 2018.

[219] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," *CoRR,* vol. abs/1801.07829, 2018.

[220] M. Fey, J. E. Lenssen, F. Weichert and H. Müller, "SplineCNN: Fast geometric deep learning with continuous B-spline kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[221] M. Dominguez, F. P. Such, S. Sah and R. Ptucha, "Towards 3D convolutional neural networks with meshes," in *Image Processing (ICIP), 2017 IEEE International Conference on*, 2017.

[222] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks,* vol. 20, pp. 61-80, 2009.

[223] B. Graham, "Sparse 3D convolutional neural networks," *arXiv preprint arXiv:1505.02890,* 2015.

[224] G. Riegler, A. O. Ulusoy and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[225] M. Tatarchenko, A. Dosovitskiy and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017.

[226] C. Häne, S. Tulsiani and J. Malik, "Hierarchical surface prediction for 3D object reconstruction," in *3D Vision (3DV), 2017 International Conference on*, 2017.

[227] O. Litany, T. Remez, E. Rodolà, A. M. Bronstein and M. M. Bronstein, "Deep Functional Maps: Structured Prediction for Dense Shape Correspondence.," in *ICCV*, 2017.

[228] J. K. Pontes, C. Kong, S. Sridharan, S. Lucey, A. Eriksson and C. Fookes, "Image2Mesh: A Learning Framework for Single Image 3D Reconstruction," *arXiv preprint arXiv:1711.10669,* 2017.

[229] O. Litany, A. Bronstein, M. Bronstein and A. Makadia, "Deformable Shape Completion with Graph Convolutional Autoencoders," *arXiv preprint arXiv:1712.00268,* 2017.

[230] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell and M. Aubry, "AtlasNet: A Papier-Mˆ ach'e Approach to Learning 3D Surface Generation," *arXiv preprint arXiv:1802.05384,* 2018.

[231] C. Van Pelt and A. Sorokin, "Designing a scalable crowdsourcing platform," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012.

[232] S. Rudinac, M. Larson and A. Hanjalic, "Learning crowdsourced user preferences for visual summarization of image collections," *IEEE Transactions on Multimedia,* vol. 15, pp. 1231-1243, 2013.

[233] S. Rudinac and M. Worring, "Making use of Semantic Concept Detection for Modelling Human Preferences in Visual Summarization," in *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia*, 2014.

[234] V. Vonikakis, R. Subramanian, J. Arnfred and S. Winkler, "Modeling image appeal based on crowd preferences for automated person-centric collage creation," in *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia*, 2014.

[235] M. Melenhorst, M. Menéndez Blanco and M. Larson, "A crowdsourcing procedure for the discovery of non-obvious attributes of social images," in *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia*, 2014.

[236] L. Marchesotti, N. Murray and F. Perronnin, "Discovering beautiful attributes for aesthetic image analysis," *International journal of computer vision,* vol. 113, pp. 246-266, 2015.

[237] N. Murray, L. Marchesotti and F. Perronnin, "AVA: A large-scale database for aesthetic visual analysis," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012.

[238] X. Lu, Z. Lin, H. Jin, J. Yang and J. Z. Wang, "Rapid: Rating pictorial aesthetics using deep learning," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014.

[239] G. Little, L. B. Chilton, M. Goldman and R. C. Miller, "Turkit: human computation algorithms on mechanical turk," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 2010.

[240] F. Sulser, I. Giangreco and H. Schuldt, “Crowd-based semantic event detection and video annotation for sports videos,” in *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia*, 2014.

[241] R. Vliegendhart, E. Dolstra and J. Pouwelse, “Crowdsourced user interface testing for multimedia applications,” in *Proceedings of the ACM multimedia 2012 workshop on Crowdsourcing for multimedia*, 2012.

[242] G. Huz, S. Bauer, R. Beverly and others, “Experience in using MTurk for Network Measurement,” in *Proceedings of the 2015 ACM SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data*, 2015.

[243] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni and J. Xiao, “TurkerGaze: Crowdsourcing Saliency with Webcam based Eye Tracking,” *CoRR,* vol. abs/1504.06755, 2015.

[244] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang and J. Hays, “Webgazer: Scalable webcam eye tracking using user interactions,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence-IJCAI 2016*, 2016.

[245] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik and A. Torralba, “Eye tracking for everyone,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[246] E. Siahaan, J. A. Redi and A. Hanjalic, “Beauty is in the scale of the beholder: Comparison of methodologies for the subjective assessment of image aesthetic appeal,” in *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, 2014.

[247] M. H. Pinson, L. Janowski, R. Pépion, Q. Huynh-Thu, C. Schmidmer, P. Corriveau, A. Younkin, P. Le Callet, M. Barkowsky and W. Ingram, “The influence of subjects and environment on audiovisual subjective tests: An international study,” *IEEE Journal of Selected Topics in Signal Processing,* vol. 6, pp. 640-651, 2012.

[248] Y. Gingold, A. Shamir and D. Cohen-Or, “Micro perceptual human computation for visual tasks,” *ACM Transactions on Graphics (TOG),* vol. 31, p. 119, 2012.

[249] Y. Gingold, E. Vouga, E. Grinspun and H. Hirsh, “Diamonds from the rough: Improving drawing, painting, and singing via crowdsourcing,” in *Proceedings of the AAAI Workshop on Human Computation (HCOMP)*, 2012.

[250] K. Mao, L. Capra, M. Harman and Y. Jia, “A survey of the use of crowdsourcing in software engineering,” *Journal of Systems and Software,* vol. 126, pp. 57-84, 2017.

[251] L. Litman, J. Robinson and T. Abberbock, “TurkPrime. com: A versatile crowdsourcing data acquisition platform for the behavioral sciences,” *Behavior research methods,* vol. 49, pp. 433-442, 2017.

[252] E. Peer, L. Brandimarte, S. Samat and A. Acquisti, “Beyond the Turk: Alternative platforms for crowdsourcing behavioral research,” *Journal of Experimental Social Psychology,* vol. 70, pp. 153-163, 2017.

[253] A. Ghezzi, D. Gabelloni, A. Martini and A. Natalicchio, “Crowdsourcing: a review and suggestions for future research,” *International Journal of Management Reviews,* vol. 20, pp. 343-363, 2018.

[254] A. A. Arechar, S. Gächter and L. Molleman, “Conducting interactive experiments online,” *Experimental economics,* vol. 21, pp. 99-131, 2018.

[255] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio, “Theano: A CPU and GPU math compiler in Python,” in *Proc. 9th Python in Science Conf*, 2010.

[256] F. Chollet and others, “Keras: Deep learning library for theano and tensorflow,” *URL: https://keras. io/k,* vol. 7, 2015.

[257] A. Jacobson, D. Panozzo, C. Schüller, O. Diamanti, Q. Zhou, N. Pietroni and others, “libigl: A simple C++ geometry processing library,” *Google Scholar,* 2013.

[258] A. Fabri and S. Pion, “CGAL: The computational geometry algorithms library,” in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009.

[259] P. Isola, J. Xiao, A. Torralba and A. Oliva, “What makes an image memorable?,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011.

[260] A. Khosla, J. Xiao, P. Isola, A. Torralba and A. Oliva, “Image memorability and visual inception,” in *SIGGRAPH Asia 2012 Technical Briefs*, 2012.

[261] A. L. Maas, A. Y. Hannun and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*, 2013.

[262] K. He, X. Zhang, S. Ren and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015.

[263] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli and G. Ranzuglia, “MeshLab: an Open-Source Mesh Processing Tool,” in *Eurographics Italian Chapter Conference*, 2008.

[264] M. Garland and P. S. Heckbert, “Surface simplification using quadric error metrics,” in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997.

[265] G. TaubinÝ, “Geometric signal processing on polygonal meshes,” 2000.

[266] K. Toutanova, D. Klein, C. D. Manning and Y. Singer, “Feature-rich part-of-speech tagging with a cyclic dependency network,” in *Proceedings of the 2003*

Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, 2003.

[267] A. Rényi, "On measures of entropy and information," 1961.

[268] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review,* vol. 5, pp. 3-55, 1948, 2001.

[269] L. Jost, "Entropy and diversity," *Oikos,* vol. 113, pp. 363-375, 2006.

[270] T. Mikolov, W.-t. Yih and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013.

[271] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.

[272] "A Hybrid Recommender with Yelp Challenge Data -- Part I," [Online]. Available: https://nycdatascience.com/blog/ student-works/yelp-recommender-part-1/. [Accessed June 2018].

[273] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch and A. Joulin, "Advances in Pre-Training Distributed Word Representations," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[274] Facebook, "Facebook Research - FastText," August 2016. [Online]. Available: https://research.fb.com/fasttext/.

[275] A. L. Alter and D. M. Oppenheimer, "Uniting the tribes of fluency to form a metacognitive nation," *Personality and social psychology review,* vol. 13, pp. 219-235, 2009.

[276] T. O. Nelson and J. Dunlosky, "When people's judgments of learning (JOLs) are extremely accurate at predicting subsequent recall: The "delayed-JOL effect"," *Psychological Science,* vol. 2, pp. 267-271, 1991.

[277] T. Katzir, S. Hershko and V. Halamish, "The effect of font size on reading comprehension on second and fifth grade children: Bigger is not always better," *PloS one,* vol. 8, p. e74061, 2013.

[278] C. Yang, T. S.-T. Huang and D. R. Shanks, "Perceptual fluency affects judgments of learning: The font size effect," *Journal of Memory and Language,* vol. 99, pp. 99-110, 2018.

[279] S. C. Weissgerber and M.-A. Reinhard, "Is disfluency desirable for learning?," *Learning and instruction,* vol. 49, pp. 199-217, 2017.

[280] I. Hernandez and J. L. Preston, "Disfluency disrupts the confirmation bias," *Journal of Experimental Social Psychology,* vol. 49, pp. 178-182, 2013.

[281] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky and A. Courville, "Adversarially learned inference," *arXiv preprint arXiv:1606.00704,* 2016.

[282] J. Donahue, P. Krähenbühl and T. Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782,* 2016.

[283] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644,* 2015.

8 Appendices

A1 – Schelling Meshes (Many-Within-Class): Shape Selection Survey

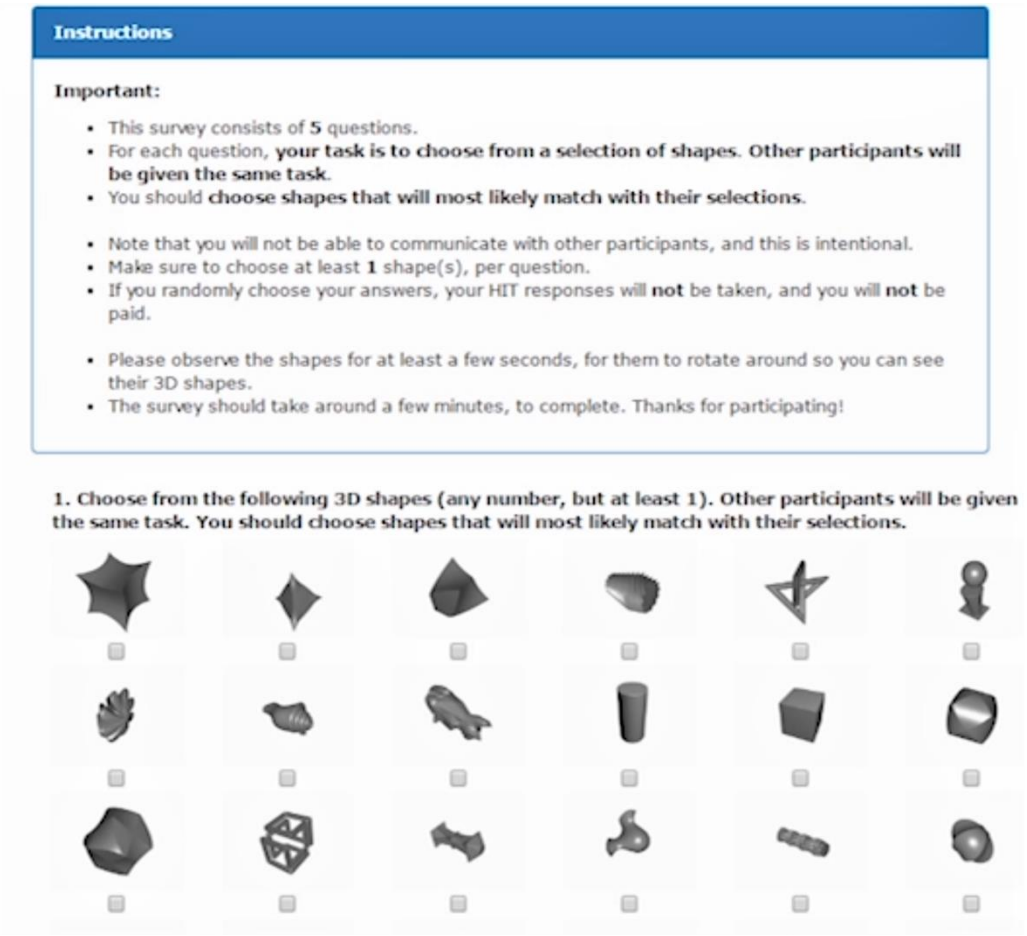


Figure 8.1 - Screenshot of an Amazon Mechanical Turk hosted survey that we provided to participants.

A2 – Schelling Meshes (Many-Within-Class): Descriptor-Based Clustering

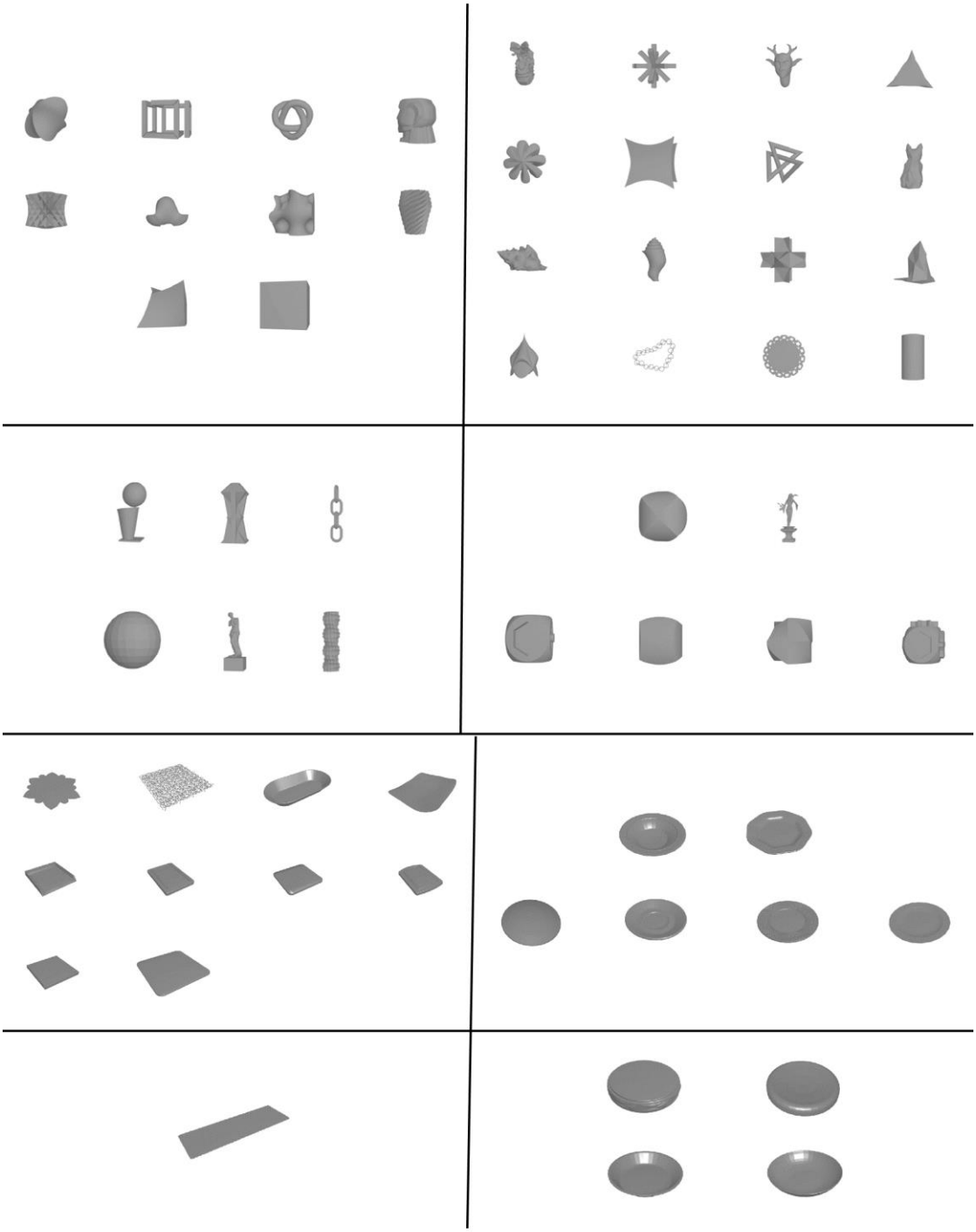


Figure 8.2 – Visualised D2 descriptor based clusterings (k-means) for the abstract shapes and plates.

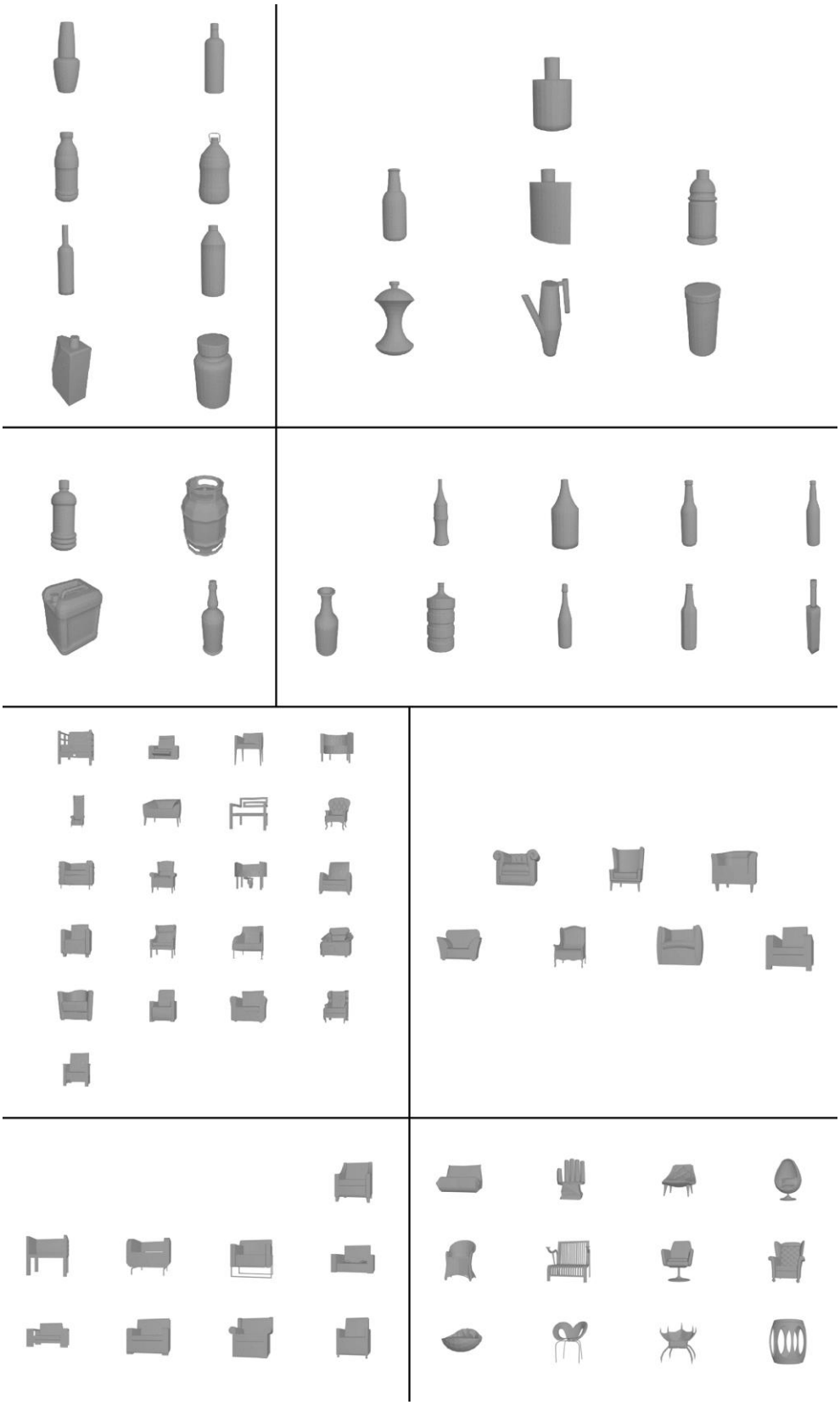


Figure 8.3 – Visualised D2 descriptor based clustering for the bottles and Sobel-based clustering for the chairs (k-means).

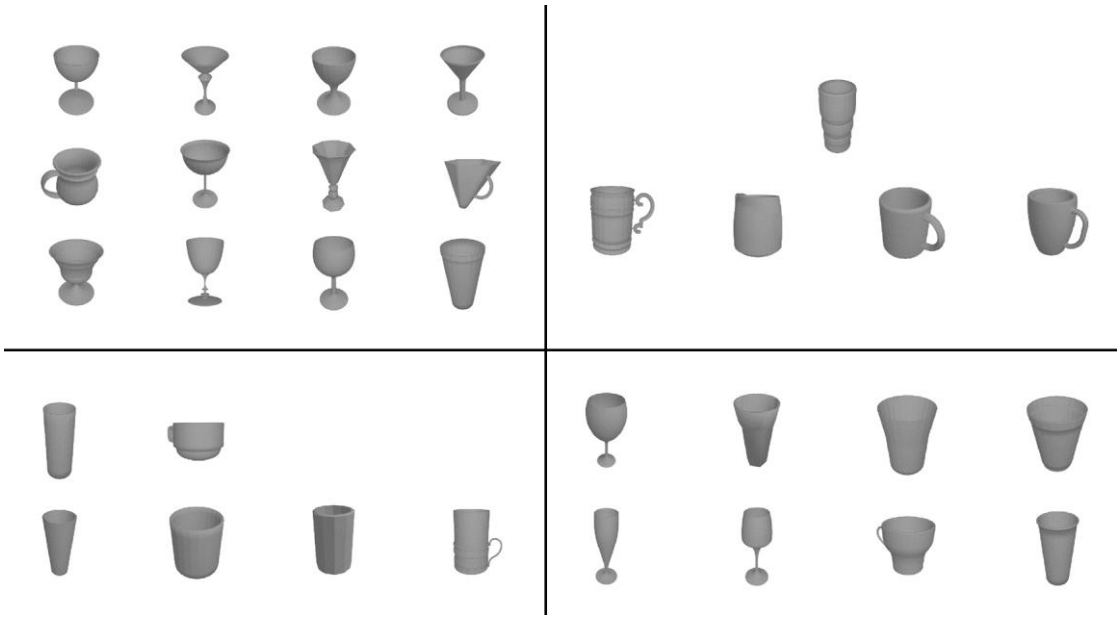


Figure 8.4 – Visualised Sobel-based clustering for the cups (k-means).

A3 – Font Specificity: Word Collection Survey

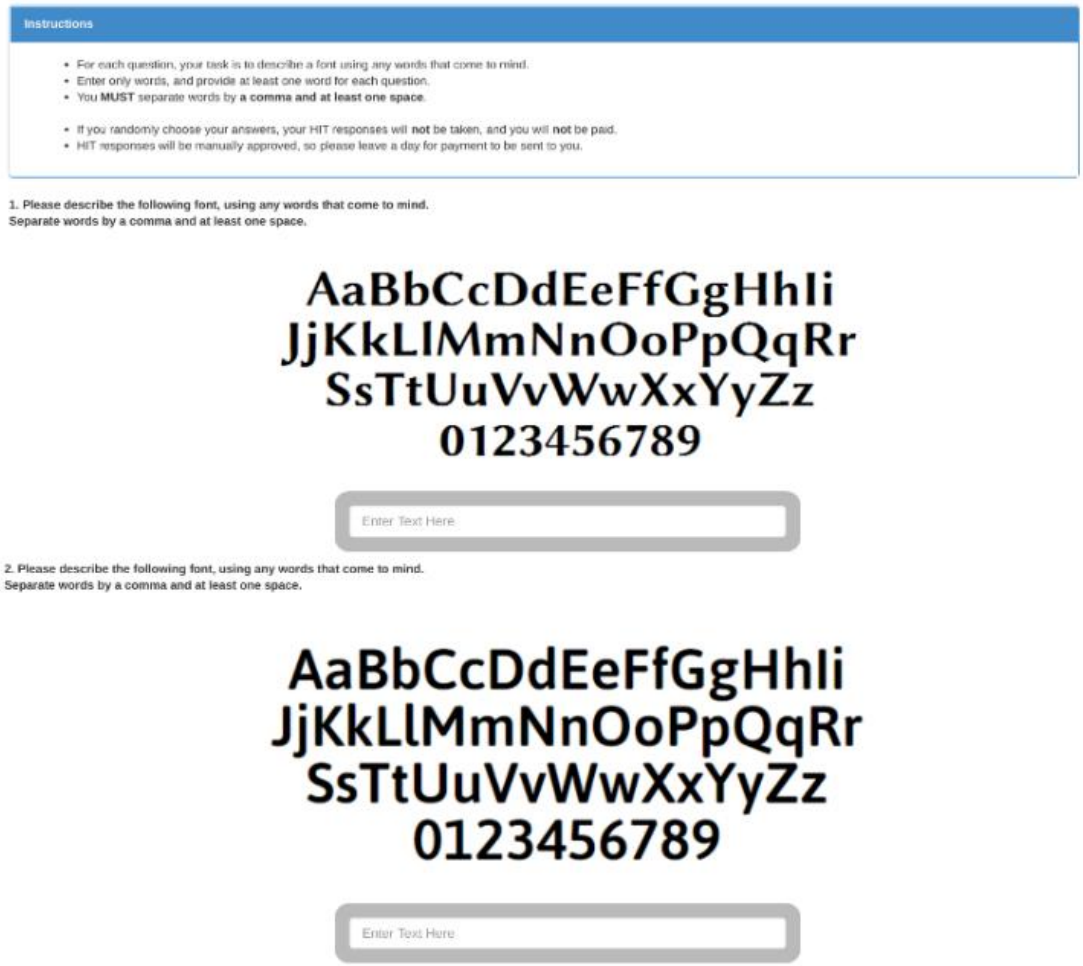


Figure 8.5 – An example of a survey that we provided to participants. They were asked to describe several fonts, using individual words.

A4 – Font Specificity: Subjective Terms Likert Survey

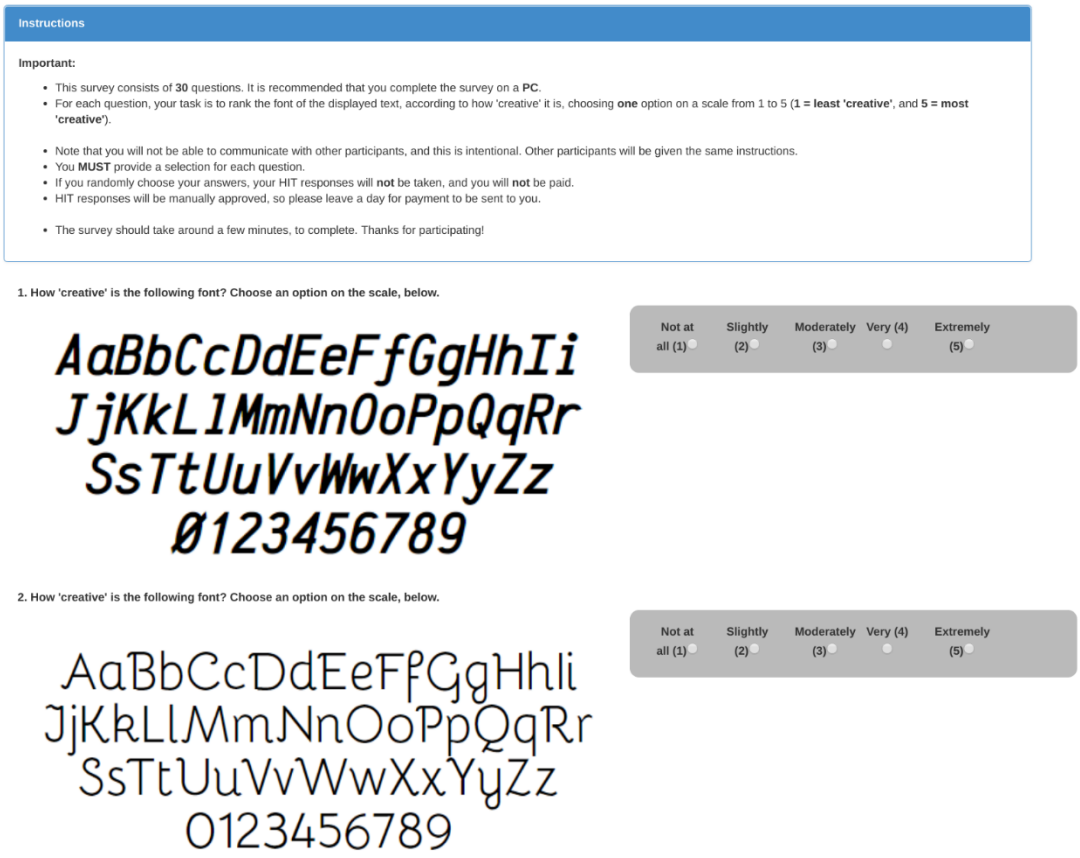


Figure 8.6 – Example Likert survey shown to participants, for data collection of subjective terms.

A5 – Font Specificity: Creativity Likert survey

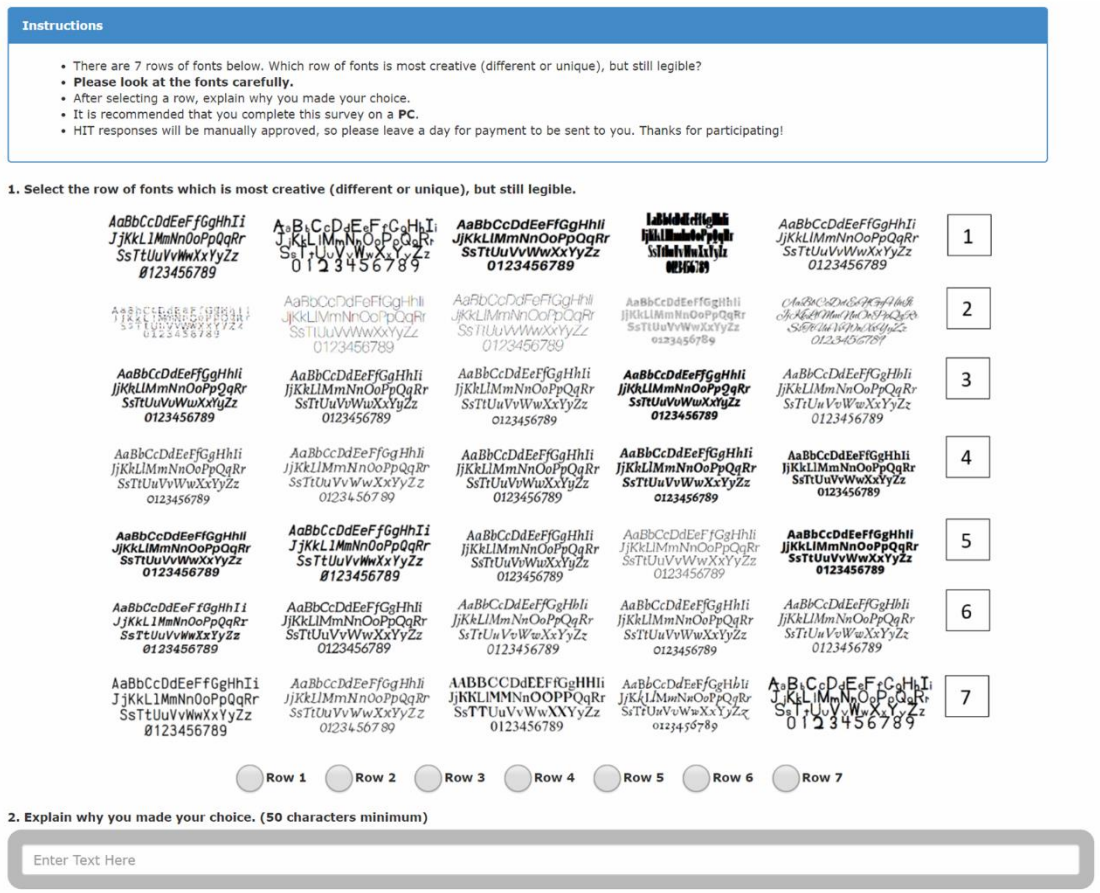
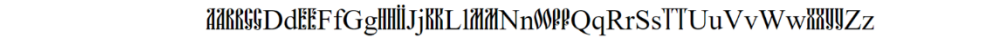


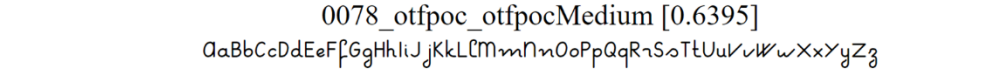
Figure 8.7 – Screenshot of survey on font creativity held via Amazon Mechanical Turk.

A6 – Font Specificity: 100 fonts ordered by word embedding-based Specificity scores

0079_ukrainian-old_AncientKyivRegular [0.6386]



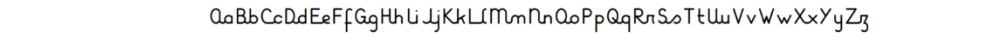
0078_otfpoc_otfpocMedium [0.6395]



0020_aileron_AileronLight [0.6396]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

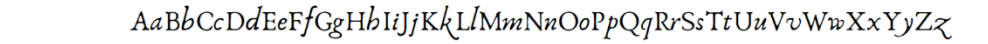
0077_otfpoc_AghjaMedium [0.6402]



0062_ambrosia_AmbrosiaMedium [0.6446]

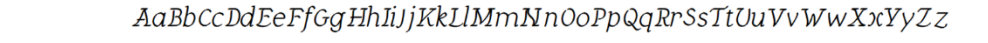
0067_anarchy-sans_AnarchySansRegular [0.6447]

0065_anaktoria-textfonts_AnaktoriaRegular [0.6460]



0080_ukrainian-old_NarbutAbetkaRegular [0.6460]

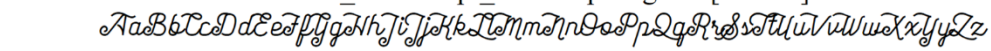
0005_ackermann_AckermannMedium [0.6468]



0093_anka-coder-condensed_AnkaCoderCondensedRegular [0.6471]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

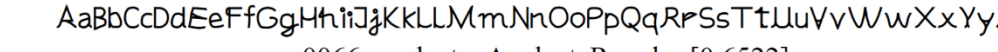
0007_acro-script_acroscriptRegular [0.6473]



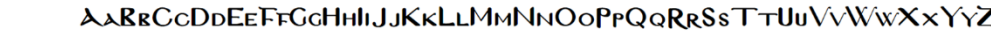
0097_anka-coder-narrow_AnkaCoderNarrowRegular [0.6500]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

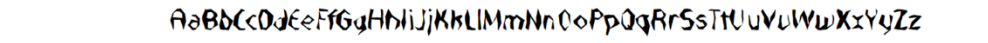
0040_alanesiana_AlanesianaRegular [0.6518]



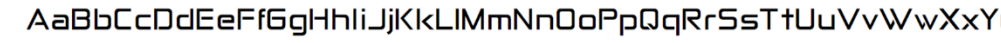
0066_analecta_AnalectaRegular [0.6522]



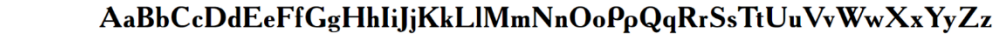
0002_a-bebedera_NabateaRegular [0.6538]



0085_anita-semi-square_AnitaSemisquareNormal [0.6549]



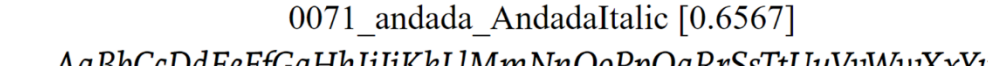
0038_akademie-alte_AkademieAlteRegular [0.6551]



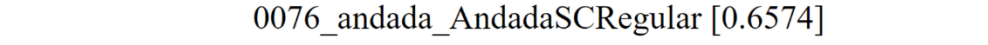
0012_aglry_AGLRYRegular [0.6553]

0064_amstelvar_AmstelvarAlphaDefault [0.6562]

0071_andada_AndadaItalic [0.6567]



0076_andada_AndadaSCRegular [0.6574]



0009_aegean_AegeanRegular [0.6592]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0063_amburegul_AmburegulRegular [0.6594]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0046_alegreya_AlegreyaRegular [0.6596]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0043_alegreya_AlegreyaBold [0.6597]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0072_andada_AndadaRegular [0.6605]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0052_aleo_AleoRegular [0.6610]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0030_ajar-sans_AjarSansBold [0.6611]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0098_anke_AnkeRegular [0.6616]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0004_abyssinica_AbyssinicaSILRegular [0.6624]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0054_alex-brush_AlexBrushRegular [0.6625]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0075_andada_AndadaSCItalic [0.6641]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0069_andada_AndadaBold [0.6649]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0032_ajar-sans_AjarSansLight [0.6662]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0006_acknowledgement_AcknowledgementMedium [0.6666]

AABBCCDDEEFFGGHHIIJJKKLLMMNNOOPPQQRRSSTTUUVVWWXXYYZZ

0095_anka-coder-narrow_AnkaCoderNarrowBoldItalic [0.6668]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0061_amaranth_AmaranthRegular [0.6669]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0070_andada_AndadaBoldItalic [0.6669]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0022_aileron_AileronRegular [0.6673]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0090_anka-coder-condensed_AnkaCoderCondensedBold [0.6678]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0057_amalgame_AmalgameRegular [0.6679]

0011_aglry_AGLRY1Regular [0.6682]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0001_a-bebedera_ABebederaHeavy [0.6682]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0068_anatolian_AnatolianRegular [0.6685]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0050_aleo_AleoLight [0.6691]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0074_andada_AndadaSCBoldItalic [0.6695]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0039_akkadian_AkkadianRegular [0.6702]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0073_andada_AndadaSCBold [0.6703]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0034_ajar-sans_AjarSansSemibold [0.6703]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0026_aileron_AileronThinItalic [0.6704]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0089_anka-coder_AnkaCoderRegular [0.6704]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0042_alegreya_AlegreyaBlackItalic [0.6706]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0019_aileron_AileronItalic [0.6707]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0094_anka-coder-narrow_AnkaCoderNarrowBold [0.6707]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0041_alegreya_AlegreyaBlack [0.6712]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0029_airborne_AIRBORNEGPBold [0.6713]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0023_aileron_AileronSemiBold [0.6717]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0096_anka-coder-narrow_AnkaCoderNarrowItalic [0.6717]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0003_abibas_AbibasMedium [0.6725]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0035_ajar-sans_OpenSansExtrabold [0.6731]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0051_aleo_AleoLightItalic [0.6737]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0021_aileron_AileronLightItalic [0.6742]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0045_alegreya_AlegreyaItalic [0.6745]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0082_andika_AndikaNewBasicBoldItalic [0.6749]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0053_alexander-textfonts_AlexanderRegular [0.6750]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0083_andika_AndikaNewBasicItalic [0.6754]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0031_ajar-sans_AjarSansExtrabold [0.6759]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0016_aileron_AileronBoldItalic [0.6772]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0036_ajar-sans_OpenSansLight [0.6777]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0033_ajar-sans_AjarSansRegular [0.6787]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0060_amaranth_AmaranthItalic [0.6788]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0049_aleo_AleoItalic [0.6804]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0055_alfphabet_AlfphabetCondensed [0.6808]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0010_aegyptus-nilus_NilusRegular [0.6812]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0013_aileron_AileronBlack [0.6828]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0008_addex_AddeXMedium [0.6831]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0018_aileron_AileronHeavyItalic [0.6833]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0088_anka-coder_AnkaCoderItalic [0.6836]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0015_aileron_AileronBold [0.6838]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

0014_aileron_AileronBlackItalic [0.6840]

AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz

Figure 8.8 – All 100 fonts sorted according to word2vec-based Specificity scores (in ascending order).

A7 – Font Specificity: Top-50 words of disjoint groups of 20 fonts, sorted according to word embedding-based Specificity scores

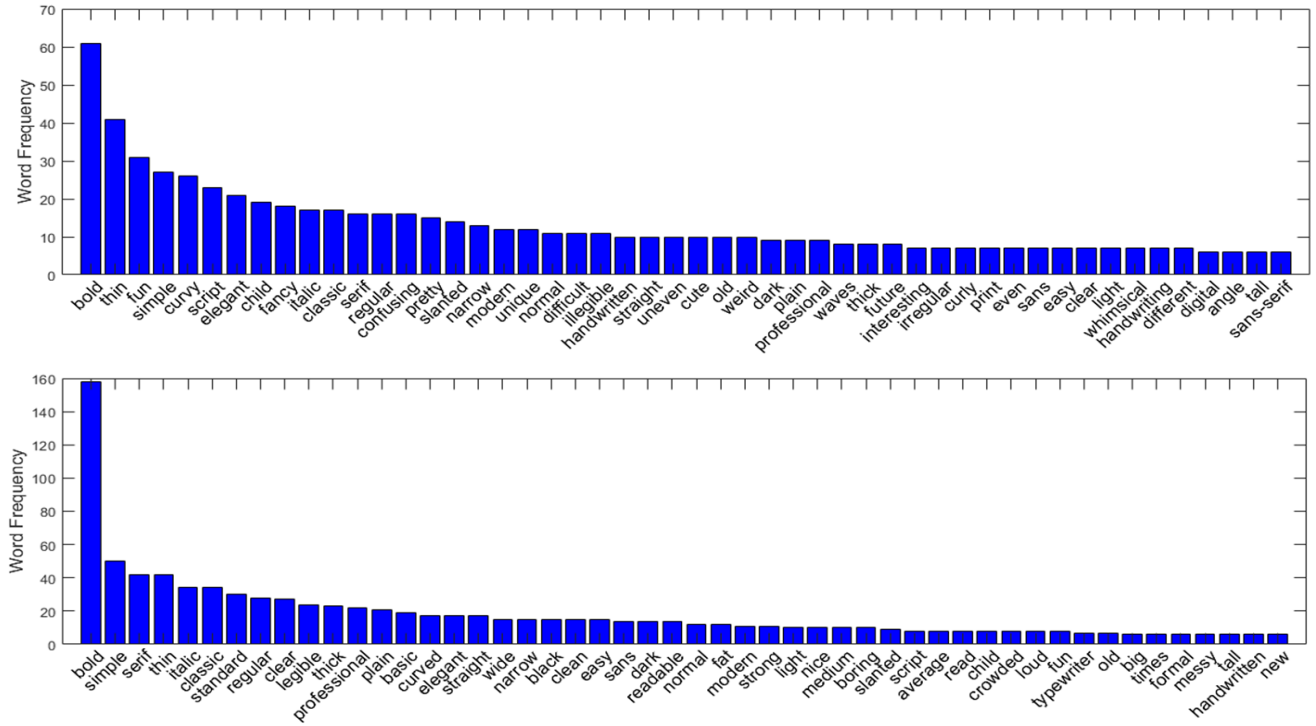


Figure 8.9 – Plots of the top-50 words' frequencies for the bottom 2 groups of 20 fonts sampled according to increasing word embedding-based Specificity score (without replacement) - top (font #1 to #20), bottom (font #21 to #40).

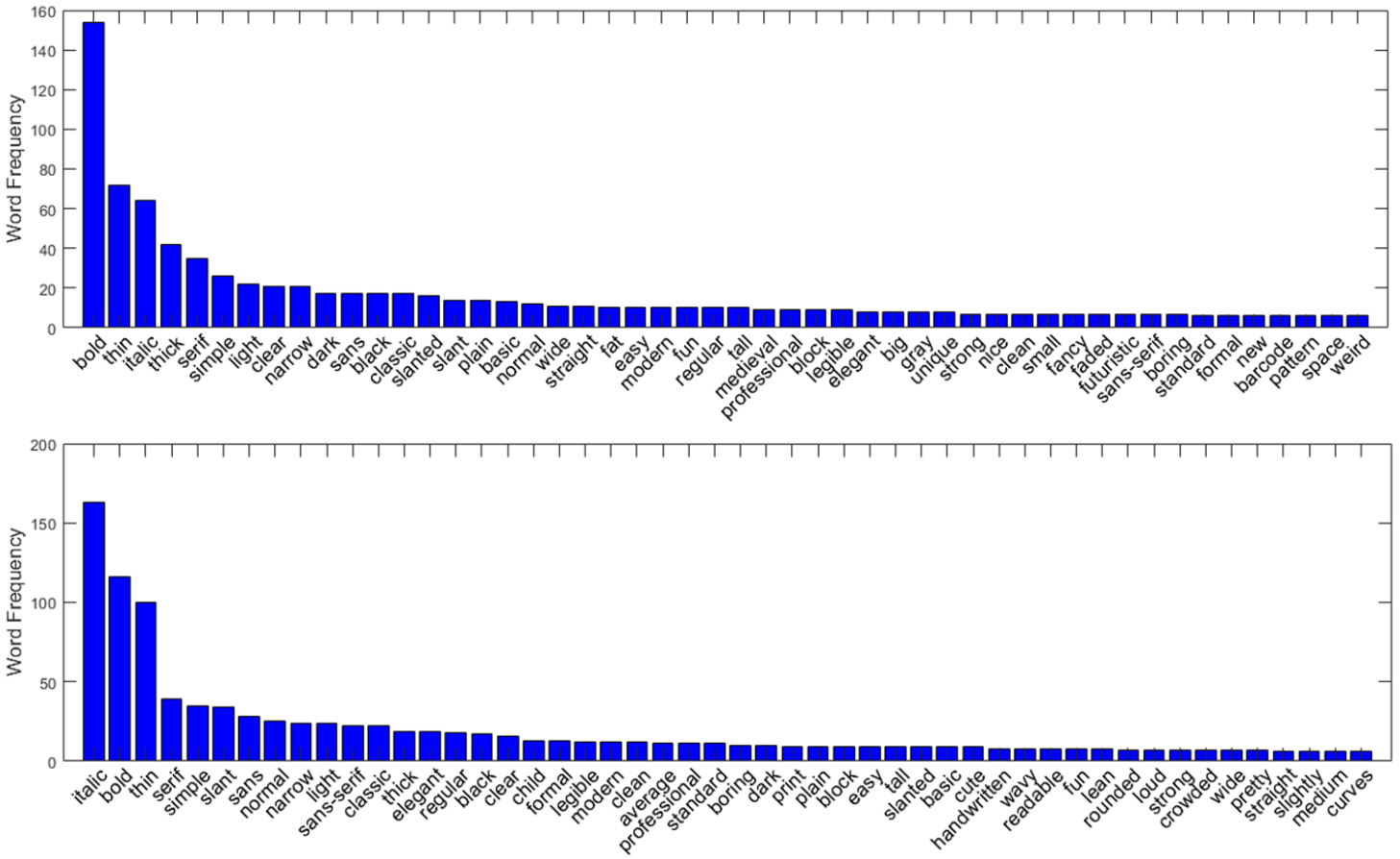


Figure 8.10 – Plots of the top-50 words' frequencies for the mid-to-high score groups of 20 fonts sampled according to increasing word embedding-based Specificity score (without replacement) - top (font #41 to #60), bottom (font #61 to #80).

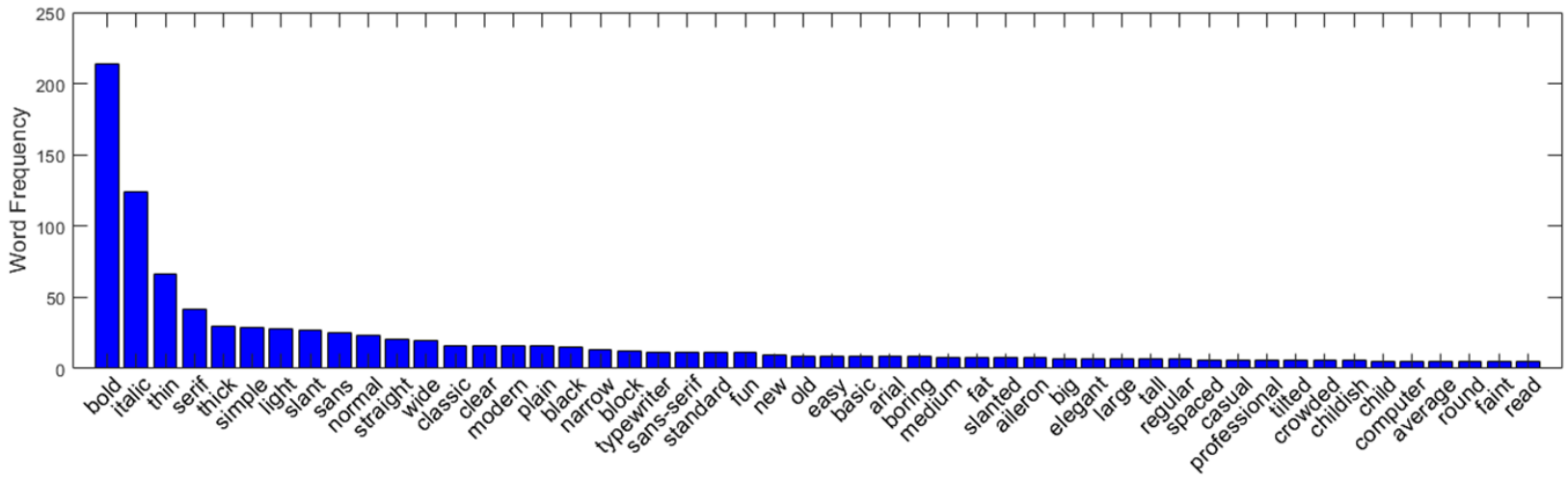


Figure 8.11 – Plots of the top-50 words' frequencies for the highest score groups of 20 fonts sampled according to increasing word embedding-based Specificity score (without replacement; font #81 to #100).

A8 – Font Specificity: Participant words split into categories

Word Category	Groups of words with equivalent meaning, per category
Geometry	(bold, boldface, emphasis, emphasize), (edges, edge, border, borders), (twist, curve, curved, curves, swerving, move, movement, swoosh, swish, curl, loop, looping, waves, wave, sway, swayed, zig, zag, zig-zag, zigzag), (italic, slant, slanted, diagonal, tilt, lean, crook, crooked, leaning, incline, sway, swayed, taper, weight, angled, angle), thin, serif, (thick, wide, broad, full, spacious, wider), (sans-serif, sans, sans serif), (straight, line), (narrow, tall, strain, crowded, elongate, elongated, stretch, stretched, tense), (cartoon, toon), (squash, stress, squeeze, constrict, crush, cramp, cramped, squashed, compress, flat, compact), (etched, etch, scratch, scratched, rubbed, engrave, hatch, hatched, scrape, scraped, scrap), (abstract, strange, mysterious, mystery), (sharp, point, pointy, spike, spiky, sharpen, pointed, precise, crisp, astute, abrupt, scrunch), (curved, curved, twist), (rough, rush, rushed, hurried, harsh, sketch, unfinished, outline, rustic, unsophisticated, doodle, doodled, wobbly, shaky, wonky, jerky, ragged, colouring, colour, color, choppy, uneven, spotty), (bubble, bubbles), (carving, jagged, jag, slash, cut, carve), (blocky, block, blockish, stumpy, blunt, chunky, gaunt), (pattern, patterned, embellish, embellished, repeating, copy, copying, copies, recurrent, double), (stamp, stamped, emboss, boss), (gothic, medieval), (big, great, large)
Subjectivity	(old, dated, old-fashioned, antique, antiquated, older, age, mature, aged), (traditional, classic, tradition, authoritative, standard, reliable, honest, secure, safe, dependable, practical, pragmatic), (modern, new, young, youthful, innovative, novel, immature, fresh, smart, sassy), (funny, peculiar, odd), (readable, legible, clear), (average, boring, dull, tiresome, ordinary, mean, norm, everyday, casual, mundane, common, usual, daily, uninspired, unimaginative), (elegant, fancy, fashion, flair, style, sophisticate, sophisticated, classy, swish, royal, majestic, regal), (royal, majestic, regal), (pretty, cute, lovely, attractive, desire, want), (sleek, slick, crafty), (sinister, bleak, spooky, flighty, nervous, unsettling, unnerving, confuse, disconcert, illogical, disjointed), (fantastic, fantastical, wonderful, lofty, grand, striking, dramatic, luxurious, special), (varied, eccentric, distant, aloof, wacky, goofy, woozy, dizzy, vary, variation, barbaric, fruity, elastic, flexible, dynamic, active, mix, hybrid, potpourri, mixture, mixed, combine, combination, complex, complicated), (sad, sorry), (strange, unusual, alien, unknown, extra-terrestrial, funny, fishy,), (foreign, ethnic, cultural, foreign, exotic), (comic, comical), (interesting, interest), (exciting, excite, excitement, yelling, shouting, screaming, shout, scream, wow, bright, hopeful, promising, hope), (extreme, overpower, overwhelm, overconfident, overpowering, overwhelming, exaggerate, exaggerated, overdone, overdo, extremely, magnify), (ugly, horrible, repulse, repulsive, abrasive, disgusting), (fun, play, playful, playfulness, toy, sport), (appeal, appealing, attention-grabbing, witch, bewitching, enchanting, capture, charm, spell, allure, tempt, entice, invite, inviting, catchy, attention-getting, eye-catching), (reasonable, good, respectable, proper, dependable, honest, ok, fair, alright, fine)
Abstraction-like	(vague, shadow, shadowed, shadowy, obscure, shade, jumble, blur, smudge, smudged, clutter, cluttered, muddle, hide, unknown, hidden, secret, mysterious, mystical, cryptic, deep, fuzzy, faint, wispy, weak, undefined, trail, cloud, fog, trace, blurry, blot, hazy), (flow, flows, flowing, silver, fluent, fluid, unstable), (clean, neat, bare, barren, bleak, fair), (tough, denser, dense, stout, sturdy, dense, denser, heavy, heavier), (unclear, unreadable, indecipherable, hard, harder, difficult), (clear, light,

	feather, readable, neat, easy, minimal, minimum, approachable, accessible, effortless, direct, straightforward, straight-forward, clean, smart, easily), (school, educated, grade, examination, testing, essay, class, education, teaching, instructions, command, commanding), (office, work), (moderate, soften, relax, informal, loose, docile, mild, comfort, relaxation, smooth, gentle, kind, soft, delicate, lenient, easy-going, easy-going, easy, ease, still, quiet, static, still, calm), (material, chalk, glass, ice, metallic, metal), (advertising, advertise, advert, business, sales, sale, presentation, display, banner), (child, childish, infantile, infant, kid, children, kids, juvenile, adolescent), (structure, family, pops, pop, dad, parents, mum, mom, parent, value, logical, coherent, level, orderly, order, regulate, coordinate, construction, organise, moral, morals, structured, stern, strict), (lettering, letters, postage, post, letter, content, message), (history, historical, memory, evocative, reminiscent, memories), (print, professional, headline, headlines, pro, brochure, pamphlet, telephone, phone, sign, signboard, signs, board, menu, formal, stately, imposing, handwriting, diary, journal, script, textbook, text, book, books, hand, scripture, bible, newspaper, intro, presentation, display, paper, wallpaper, composition, news, newsprint, typography, typeface, typeset, typesetting, define, calligraphy, penmanship, scribe, published, scribed, banner, standard, printing, essay, write, writing, pen), (place, pub, school, saloon, city, metropolis), (design, designer, architect, construction, construct), (art, artistic, aesthetic), (machine, computer, calculator, mac, retro, tech, technology, technological, technical, hackers, hacker, television, TV, video, robots, robot, artificial), (science, sciences, experiment, testing, test), (season, time, day, week, month, year, summertime, summer, wintertime, winter, springtime, spring, autumntime, autumn)

Table 8.1 – Table of word groups used to analyse the collected word data.