Learning a Human-Perceived Softness Measure of Virtual 3D Objects

Manfred Lau[1] Kapil Dev[1] Julie Dorsey[2] Holly Rushmeier[2]
[1]Lancaster University [2]Yale University

Abstract

We introduce the problem of computing a human-perceived softness measure for virtual 3D objects. As the virtual objects do not exist in the real world, we do not directly consider their physical properties but instead compute the human-perceived softness of the geometric shapes. We collect crowdsourced data where humans rank their perception of the softness of vertex pairs on virtual 3D models. We then compute shape descriptors and use a learning-to-rank approach to learn a softness measure mapping any vertex to a softness value. Finally, we demonstrate our framework with a variety of 3D shapes.

Keywords: 3D modeling, crowdsourcing, learning, fabrication

Concepts: •**Computing methodologies** → *Perception; Shape analysis;*

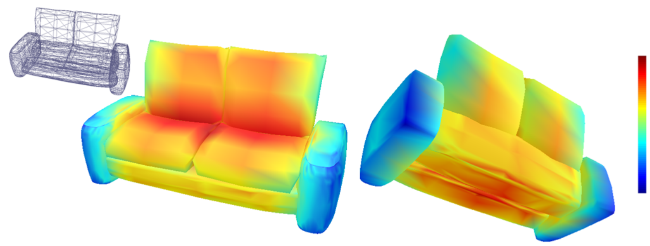

Figure 1: *An input 3D object and softness map computed by our approach (front and back views). The jet colormap corresponds to relative softness values (red is softer).*

1 Introduction

There are many physical properties of real-world objects that are identifiable by humans. Softness or compliance [Tiest 2010] is one such property. Objects in our daily lives frequently have parts with varying softness. For example, a couch, bed, or shoe has varying softness that humans may identify just by observing it. In this paper, we consider the **softness of virtual 3D shapes** as opposed to physical objects. Our definition of "softness" is as follows: a human thinks of the softness of a virtual object given only its geometric shape, and imagines the virtual shape as a real-world object and pressing into points on its outer surface towards the direction of the surface normal; if he/she can imagine pressing into the virtual shape, then the more easily the virtual point can be "pressed", the more soft it is. Computing the spatially varying perceived softness of an object can be useful for both rendering and fabrication. In rendering, we can use this property to convey the sense of softness of an object in a scene. In fabrication, we can use this to manufacture objects that are consistent with user expectations.

The perception of softness when an object is touched depends on physical properties such as stiffness and Young's modulus [Tiest 2010]. However, humans can make judgements on softness without touch, but just by looking at objects. Hence we collect crowdsourced data of human perception which will gather the crowd's consensus to compute softness. We expect humans to recognize the object or parts of it unconsciously as a real object even though the only information provided is a 3D virtual shape (or images of it). This perceived context and recognition of objects will be included in the collected data itself. We are inspired by *Thurstone's law of comparative judgment*, as we study *how humans perceive objects* rather than performing any measurements of physical objects. We also use the concept of *pairwise comparison* when asking users to compare between pairs of points on a shape.

This paper presents a crowdsourcing and learning framework to solve the problem of computing a softness measure for virtual 3D objects. We take as input a 3D shape and automatically compute as output the softness of every vertex on the shape. Our approach is inspired by recent methods for learning similarity measures (e.g. for 2D clip art [Garces et al. 2014]). We collect crowdsourced data where humans compare the softness of pairs of vertices on 3D shapes. For each 3D shape, we compute shape descriptors similar to those used in 3D shape retrieval or classification [Shilane et al. 2004]. We then use a learning-to-rank approach [Chapelle and Keerthi 2010] typically used in machine learning for web page ranking and information retrieval to learn a softness measure mapping each vertex to a relative softness value. After a softness measure is learned, we can use it to compute the human-perceived softness for a new 3D shape of an object type that we have collected data for.

We demonstrate our approach by computing softness maps of 3D shapes from 3D Warehouse and the Princeton Shape Benchmark [Shilane et al. 2004]. We perform a user study to evaluate our approach. We then show the application of fabricating a real-world object with the softness information computed by our approach.

The contributions of this paper are:

- We introduce the problem of computing a *human-perceived* softness measure for virtual 3D shapes.
- We solve this problem with a crowdsourcing + learning-to-rank approach.

1.1 Related Work

Human Perception. When it comes to conveying the physical properties of virtual objects, research shows that human perception can be stimulated in various ways to deliver the desired information. Users can perceive textures [Lécuyer et al. 2004] on the desktop by modifying the cursor's motion on the screen. Garcia et al. [2010] studies the perception of deformations in 3D shapes. They present a new method which adds local deformations to modal analysis simulations in order to improve the realism of the overall deformations. Sanz et al. [2013] allows users to perceive local elasticity in images. When a user interacts with an image with a mouse, their system gives visual feedback and generates deformation effects to induce the perception of stiffness in the image. While a lot of perception related research relies on stimulating human perception for conveying information, we investigate the expectation of the user

SAP 2016, July 22-23, 2016, Anaheim, CA
ISBN: 978-1-4503-4383-1/16/07
DOI: http://dx.doi.org/10.1145/2931002.2931019

(softness in our case) instead of providing a pre-defined feedback to be perceived by the user.

Geometry Modeling. There exists work in analyzing the *virtual properties of 3D shapes* for various purposes. If one can simultaneously segment and label parts in 3D models [Kalogerakis et al. 2010], it may be possible to gather some information about softness from the labels. However, humans may still have to specify such information between labels and softness beforehand. Our work computes the softness of each vertex in one step, which may help to avoid potential errors introduced in an intermediate labeling step.

Physics-based Deformations of 3D Shapes. Early work in this area [Terzopoulos et al. 1987; Kass et al. 1988] described methods for modeling deformable shapes. Modal analysis methods can simulate deformations by defining a deformation energy, and can be used for many purposes such as segmenting a shape into parts [Huang et al. 2009]. Our method does not have an underlying simulation, but we instead target the human perceptual process and extract data from it by taking a combined framework of 3D modeling, crowdsourcing, and learning.

Computational Fabrication. There has been previous work on fabricating objects to resemble virtual shapes. Bickel et al. [2010] fabricate real-world objects with desired softness or deformable properties. Lau et al. [2011] builds real-world furniture by generating parts and connectors that can be fabricated from an input 3D model. Schwartzburg et al. [2013] and Cignoni et al. [2014] generate interlocking planar pieces that can be laser cut and slotted together to resemble the original 3D shape. In the growing field of fabrication, we make a contribution by computing softness information on the surface of a 3D shape from its geometry alone. This allows for the fabrication of objects with softness consistent with user expectations.

2 Human-Perceived Softness Measure

We present the details of our method in this section. First, we compute 3D shape features for each vertex. Second, we describe the process for collecting softness data from humans. Third, the shape features and crowdsourced data are then used to learn a human-perceived softness measure mapping each vertex to a softness value.

2.1 3D Shape Features

To learn the relation from a vertex to a softness value, we need to compute a feature vector ($\mathbf{x}$) for a vertex on a 3D model. We take 3D shape descriptors from previous work that have been used for classification and retrieval. Hence the features are not new on their own. Note that for some features, we use a variant of the commonly used version of the descriptor, as we compute features for a vertex v relative to the whole model M rather than for the whole model.

We build a feature vector with descriptors that are based on geometry alone. The main aspects of the features relate to the 3D shape (D2 Shape Distribution [Osada et al. 2001], Gaussian Image [Horn 1984]), projected 2D shapes (Light Field Descriptor [Chen et al. 2003; Shilane et al. 2004]), and curvatures (Gaussian and Mean curvatures [Surazhsky et al. 2003], Sobel operators on voxels). The input is a 3D model, and we compute a 117-dimensional feature vector $\mathbf{x}$ for any vertex v of a model M. We do not decide a priori the importance of each feature, but we include many possible features and allow the learning method to decide the weight or significance of each feature.

2.2 Crowdsourcing Softness Data

We collected 155 3D models from online datasets: the Princeton Shape Benchmark [Shilane et al. 2004] and Trimble 3D Warehouse. Since it is difficult for humans to provide absolute softness values (e.g. to provide a real number to a single surface point), the key to the data collection is to ask humans to compare the softness between pairs of points. This is similar to [Garces et al. 2014] where humans compare relative styles of 2D clip art.

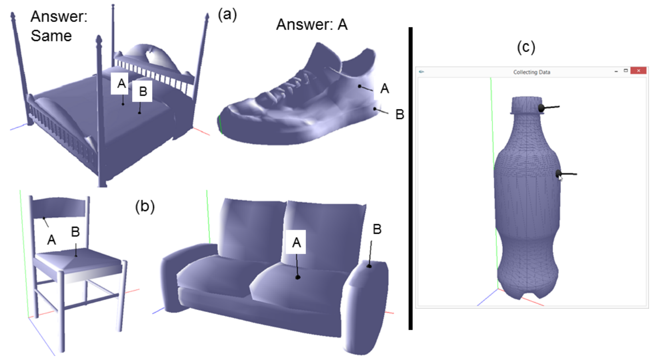

Figure 2: *Data collection: (a) Two examples of images with answers chosen by us given as part of the instructions for Amazon Mechanical Turk HITs. Text instructions were given to users: "Imagine the object as a real-world object and physically pressing into it at points A and B with your finger, in the direction of the small line and towards the object. Choose whether point A or B is softer, or that they have the same softness. Please choose what you believe to be the best answer in each case." (b) Two example HIT images we used. (c) Screenshot of software where user directly selects pairs of vertices and specify which is softer (or same).*

We used two methods to collect data. For the first method, we generated images of pairs of points on the virtual 3D models and asked humans to label them using Amazon Mechanical Turk. For each model, we randomly selected two vertices and we chose a camera viewpoint such that both points can be seen. A human rater is initially given instructions and example images with responses (Figure 2a). Each HIT (a set of tests on Amazon Mechanical Turk) then consists of 50 images (see Figure 2b for some examples). For each image, the human selects either "A" or "B" if one of the labeled points is softer, or "same" if he/she thinks that both points have the same softness. The crowdsourced data may be unreliable. Before a user can work on the HITs, he/she needs to pass a "qualification" test by correctly answering at least 4 of 5 images. In addition, for each HIT, we have 5 control images and if the user incorrectly answers more than 2 of them, we reject the data for that HIT (we rejected 7.3% of HITs). We paid $0.20 for each HIT, and we had 112 users and 6080 samples of data (each sample is one image).

The second method is a software tool we provide to users to select pairs of vertices. The user visualizes a model in 3D space and directly clicks on a vertex with the mouse to select it (Figure 2c). The user then provides the label (i.e. which vertex is softer or same) with keyboard presses. This method provides more reliable data as we can give more guidance to the users from the start, and hence we do not reject any data collected with this method. For this method, we had 15 users and collected 7000 samples.

From the data collection, we have a set of training data $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_h\}$ and the corresponding label sets $\mathcal{I}$ and $\mathcal{E}$ each containing pairs of $(\mathbf{x}_i, \mathbf{x}_j)$. Each pair of vertices labeled as having one vertex being softer is in the inequality set $\mathcal{I}$. Each pair labeled as having the same softness is in the equality set $\mathcal{E}$.

2.3 Learning Softness Measure

We learn a softness measure with the feature vector and crowdsourced data. We use a learning-to-rank method known as RankSVM [Chapelle and Keerthi 2010]. This method takes as in-

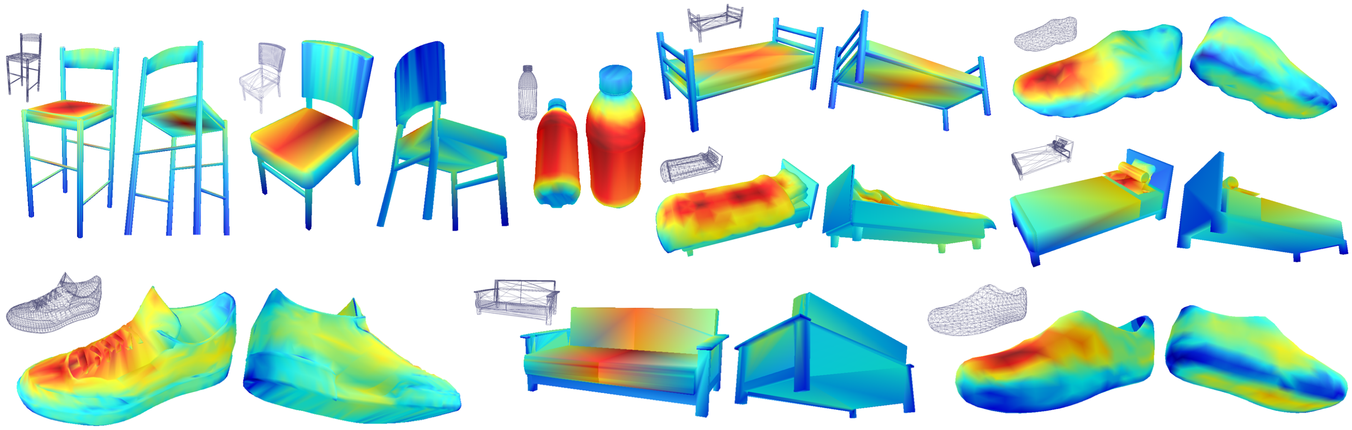

Figure 3: *Results: Input 3D models and corresponding softness maps (front and back/bottom views). Left chair is m813 (Princeton Shape Benchmark), top bed is m944, top shoe is m1746, and bottom left shoe is m1741. The others are from Trimble 3D Warehouse.*

put $\mathcal{X}$, $\mathcal{I}$, and $\mathcal{E}$, and returns the weight vector $\mathbf{w}$. For a given feature vector $\mathbf{x}$, we assume that the corresponding softness $f(\mathbf{x})$ is an output of a linear function: $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$. The weight vector $\mathbf{w}$ is learned by minimizing the rank cost functional:

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{(i,j)\in\mathcal{I}} l_1(d(\mathbf{x}_i, \mathbf{x}_j)) + C \sum_{(i,j)\in\mathcal{E}} l_2(d(\mathbf{x}_i, \mathbf{x}_j)) \quad (1)$$

where the first term is a standard regularizer to prevent over-fitting, C is a hyper-parameter, $l_1(t) = \max(0, 1-t)^2$ and $l_2(t) = t^2$ are suitable loss functions, and $d(\mathbf{x}_i, \mathbf{x}_j) = f(\mathbf{x}_i) - f(\mathbf{x}_j)$. If $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{I}$, vertex i should be softer than vertex j, and $f(\mathbf{x}_i)$ should be greater than $f(\mathbf{x}_j)$. Similarly $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}$ implies equal softness: $f(\mathbf{x}_i)$ should be equal to $f(\mathbf{x}_j)$. The ranking loss function $l_1(t)$ enforces prescribed inequalities in $\mathcal{I}$ with a standard margin of 1, while the equality loss function $l_2(t)$ measures the standard squared deviations from the equality constraints in $\mathcal{E}$.

The energy function $\mathcal{L}$ is minimized using the primal Newton method as originally developed by Chapelle [2010] for inequality constraints and subsequently adapted by Parikh and Grauman [2011] for equality constraints. Since the data and method are ranking-based, the learned measure provides relative softness values and they make more sense when compared to each other.

3 Results

Relevance of Features. The L^2 regularizer ($\|\mathbf{w}\|^2$ term in Equation 1) leads to small weights in the learned $\mathbf{w}$ vector. Comparing the weight values from the learned $\mathbf{w}$ provides information about the relevance of the features. The 2D Light Field Descriptors are the most relevant. The Gaussian curvature, Mean curvature, and Sobel operators have small weights and are the least relevant.

Computing Softness Maps for Whole Shape. We show results of input 3D shapes and computed softness maps (Figures 1 and 3). We generate the results by computing the softness values for each vertex, and then mapping them (while maintaining the ranking) to $[0, 1]$ so each vertex can be assigned a color for visualization purposes. Since our results provide a relative ranking, a single vertex labeled red for example may not necessarily be soft on its own. We also tested our method with new 3D models not used during the data collection (bottom couch and bottom right shoe in Figure 3). It takes a few seconds to generate each result in MATLAB.

4 User Evaluation

We perform a user study to show that the learned softness measure is comparable to the human perception of virtual objects.

Task. Participants observe virtual shapes with two selected points and choose the softer point (or same). They are given the same instructions as in the crowdsourced HITs, except the shapes are not provided to them on Mechanical Turk but we provide them with a program. We collected a new test set of 10 3D models. Each user was given 5 of the 10 models picked randomly beforehand. We provide the user with a program that randomly selects 16 pairs of vertices for each of the 5 models. The user can see each pair of vertices by rotating the 3D model if needed (the rotation is only for visualizing vertices if they are occluded and does not provide additional information). One pair of vertices for each model is a control sample to check for quality. There are 5 of these and if a user provides 2 or more responses that do not match ours, we reject that user's responses.

Apparatus and Participants. The participants completed the task using a laptop we provided. We recruited 15 participants (6 female) with online and poster advertisements from our university. Participants were between 21 and 45 years old (Mean=31.2, Std=6.2).

Study Results. We have 15 users x 5 models x 15 responses. No user responses were rejected based on the control samples. For each "A" or "B" response (305 and 395 of each respectively), we compute the softness of the vertex pair with our learned measure and see if it matches with the user response. All the results from our learned measure is either "A" or "B" as the measure gives real values. The accuracy of our measure is 91.43%.

For each user response labeled "same" (425 of these), we compute the absolute difference of the softness values from our measure between the two vertices. The mean of these absolute differences is 0.083 (Std=0.096). For comparison purposes, we also compute the differences for the "A" and "B" responses. In this case, if the response is "A", we compute the softness for vertex "A" minus the softness of vertex "B", which can be negative if our measure is incorrect. The mean of these is 0.260 (Std=0.194). We perform a two-sample t-test assuming unequal variances and find a significant effect (t=-5.595; p <0.001) between "same" and "A/B" responses.

5 Application: Fabrication Example

We can fabricate a virtual shape into a real object according to the computed non-uniform human-perceived softness. There are many ways to construct a physical prototype object. Inspired by methods to layer materials to fabricate soft objects [Hudson 2014], we choose to experiment with a layering-based technique with felt material for a simple example of the bottom part of a sandal. Figure 4a shows the model and the softness map. We discretize the soft-

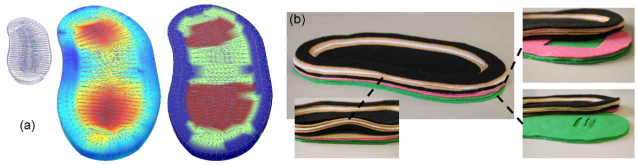

Figure 4: *Application: Fabrication Example. (a) An input 3D model of the bottom part of a sandal, the softness map computed by our method, and the softness values discretized into three types for fabrication purposes. (b) The sandal fabricated with layers of felt material. We also show (small inset images) some simple ways we experimented with to adjust the softness (e.g. adding a harder card piece, cutting a hole, or stripes of holes) if we were to press top-down into the object to match the values our method computed.*

ness values into three types to simplify the fabrication process. To approximate the object shape, we voxelize the 3D shape to create layers of 2D profiles and cut them with felt material (Figure 4b). To approximate the softness, we converted the computed softness to a top-down 2D grid of discretized softness if a human were to press top-down into the real object.

6 Conclusion, Limitations, and Future Work

We have presented a framework for computing a human-perceived softness measure for virtual 3D shapes. Our framework is flexible since we can add more training data or more shape features at any time and then re-learn the weight vector.

Physically-based simulation methods can be used to compute deformation information of 3D shapes. However, our focus is on human perception of virtual shapes which may not match with the deformations computed by physically-based methods. There also exists devices for measuring material stiffness or Young's modulus, but we do not use such devices as we explore the softness of virtual shapes which cannot be directly measured with physical devices.

For many 3D models, humans can perceive softness information well from the geometry alone. A limitation of our method is that there are some models where it can be ambiguous to decide if the object is soft or not soft. For example, a virtual bottle with smooth surfaces may be soft if it is made of plastic or not soft if it is made of metal. One possible future work direction is to add an estimate of reliability to the computed softness map. Another possible direction is to combine our method with other methods for understanding the materials of shapes [Jain et al. 2012].

Acknowledgements

We thank the reviewers for their suggestions and Kwang In Kim for help with machine learning. This work was funded in part by NSF grants IIS-1064412 and IIS-1218515. Kapil Dev was funded by a Microsoft Research PhD scholarship and Manfred Lau was on a sabbatical leave at Yale during this project.

References

BICKEL, B., BÄCHER, M., OTADUY, M. A., LEE, H. R., PFISTER, H., GROSS, M., AND MATUSIK, W. 2010. Design and Fabrication of Materials with Desired Deformation Behavior. *ACM Trans. Graph. 29*, 4 (July), 63:1–63:10.

CHAPELLE, O., AND KEERTHI, S. S. 2010. Efficient Algorithms for Ranking with SVMs. *Information Retrieval Journal 13*, 3 (June), 201–215.

CHEN, D. Y., TIAN, X.-P., SHEN, Y.-T., AND OUHYOUNG, M. 2003. On Visual Similarity Based 3D Model Retrieval. *Computer Graphics Forum 22*, 3, 223–232.

CIGNONI, P., PIETRONI, N., MALOMO, L., AND SCOPIGNO, R. 2014. Field-aligned Mesh Joinery. *ACM Trans. Graph. 33*, 1, 11:1–11:12.

GARCES, E., AGARWALA, A., GUTIERREZ, D., AND HERTZMANN, A. 2014. A Similarity Measure for Illustration Style. *ACM Trans. Graph. 33*, 4 (July), 93:1–93:9.

GARCÍA, M., OTADUY, M. A., AND O'SULLIVAN, C. 2010. Perceptually Validated Global-local Deformations. *Computer Animation and Virtual Worlds 21*, 3-4 (May), 245–254.

HORN, B. 1984. Extended Gaussian images. *Proceedings of the IEEE 72*, 12, 1671–1686.

HUANG, Q.-X., WICKE, M., ADAMS, B., AND GUIBAS, L. 2009. Shape Decomposition using Modal Analysis. *Eurographics 28*, 2, 407–416.

HUDSON, S. E. 2014. Printing Teddy Bears: A Technique for 3D Printing of Soft Interactive Objects. *SIGCHI*, 459–468.

JAIN, A., THORMÄHLEN, T., RITSCHEL, T., AND SEIDEL, H.-P. 2012. Material Memex: Automatic Material Suggestions for 3D Objects. *ACM Trans. Graph. 31*, 6 (Nov.), 143:1–143:8.

KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3D Mesh Segmentation and Labeling. *ACM Trans. Graph. 29*, 4 (July), 102:1–102:12.

KASS, M., WITKIN, A., AND TERZOPOULOS, D. 1988. Snakes: Active Contour Models. *IJCV 1*, 4 (Jan.), 321–331.

LAU, M., OHGAWARA, A., MITANI, J., AND IGARASHI, T. 2011. Converting 3D Furniture Models to Fabricatable Parts and Connectors. *ACM Trans. Graph. 30*, 4 (July), 85:1–85:6.

LÉCUYER, A., BURKHARDT, J.-M., AND ETIENNE, L. 2004. Feeling Bumps and Holes Without a Haptic Interface: The Perception of Pseudo-haptic Textures. *SIGCHI*, 239–246.

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2001. Matching 3D Models with Shape Distributions. *Shape Modeling International*, 154–166.

PARIKH, D., AND GRAUMAN, K. 2011. Relative Attributes. *International Conference on Computer Vision (ICCV)*, 503–510.

SANZ, F. A., JÁUREGUI, D. A. G., MARCHAL, M., AND LÉCUYER, A. 2013. Elastic Images: Perceiving Local Elasticity of Images Through a Novel Pseudo-Haptic Deformation Effect. *ACM Transactions on Applied Perception 10*, 3, 17:1–17:14.

SCHWARTZBURG, Y., AND PAULY, M. 2013. Fabrication-aware Design with Intersecting Planar Pieces. *CGF 32*, 2pt3, 317–326.

SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. The Princeton Shape Benchmark. *SMI*, 167–178.

SURAZHSKY, T., MAGID, E., SOLDEA, O., ELBER, G., AND RIVLIN, E. 2003. A Comparison of Gaussian and Mean Curvatures Estimation Methods on Triangular Meshes. *ICRA*, 1021–1026.

TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically Deformable Models. In *Computer Graphics*, vol. 21, 205–214.

TIEST, W. M. B. 2010. Tactual Perception of Material Properties. *Vision Research 50*, 24, 2775–2782.