

Perceptual Models of Viewpoint Preference

ADRIAN SECORD

New York University

JINGWAN LU and ADAM FINKELSTEIN

Princeton University

and

MANISH SINGH and ANDREW NEALEN

Rutgers University

The question of what are good views of a 3D object has been addressed by numerous researchers in perception, computer vision and computer graphics. This has led to a large variety of measures for the goodness of views as well as some special-case viewpoint selection algorithms. In this paper, we leverage the results of a large user study to optimize the parameters of a general model for viewpoint goodness, such that the fitted model can predict people's preferred views for a broad range of objects. Our model is represented as a combination of attributes known to be important for view selection, such as projected model area and silhouette length. Moreover, this framework can easily incorporate new attributes in the future, based on the data from our existing study. We demonstrate our combined goodness measure in a number of applications, such as automatically selecting a good set of representative views, optimizing camera orbits to pass through good views and avoid bad views, and trackball controls that gently guide the viewer towards better views.

Categories and Subject Descriptors: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—*Perceptual reasoning*; I.3.3 [Computer Graphics]: Picture/Image Generation—*Viewing algorithms*; I.3.4 [Computer Graphics]: Graphics Utilities—*Virtual device interfaces*

General Terms: Experimentation, Human Factors.

Additional Key Words and Phrases: Viewpoint selection, user study, visual perception, user interfaces, camera control.

ACM Reference Format:

Secord, A., Lu, J., Finkelstein, A., Singh, M., Nealen, A. 201X. Perceptual Models of Viewpoint Preference. *ACM Trans. Graph.* XX, X, Article XXX (Month 201X), XX pages.

DOI = 10.1145/XXXXXXXX.YYYYYYY

<http://doi.acm.org/10.1145/XXXXXXXX.YYYYYYY>

Authors' addresses: ajsecord@cs.nyu.edu, jingwanl@princeton.edu, af@cs.princeton.edu, manish@ruccs.rutgers.edu, nealen@cs.rutgers.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© YYYY ACM 0730-0301/YYYY/11-ARTXXX \$10.00

DOI 10.1145/XXXXXXXX.YYYYYYY

<http://doi.acm.org/10.1145/XXXXXXXX.YYYYYYY>

1. INTRODUCTION

Beauty is bought by judgement of the eye.

—Shakespeare, *Loves Labours Lost*, 1588

What makes for a good view of a 3D object? Different views are not equally effective at revealing shape, and people express clear preferences for some views over others [Banz et al. 1999]. Object recognition and understanding depends on both view-independent properties [Biederman 1987] and view-dependent features [Koenderink and Doorn 1979]. Researchers have proposed a variety of measures for view point preference, for example viewpoint entropy [Vázquez et al. 2001], silhouette stability [Gooch et al. 2001], mesh saliency [Lee et al. 2005], and symmetry [Podolak et al. 2006]. We combine attributes like these into an overall measure that we call the *goodness* of views, based on human preference data. While similar approaches based on machine learning algorithms have been previously described, in one case the method only incorporates a single measure [Laga and Nakajima 2008], and in another case the system is designed to produce a very specialized measure trained on a small data set generated by one or few people [Veira et al. 2009].

Our goodness measure relies on weights determined via a large user study, in which, given two views of the same object, hundreds of subjects were asked to select a preferred view. The resulting dataset covers each of 16 models with 120 pairs of views, and each pair was evaluated by 30 to 40 people. From this data we can combine attributes together in various predictive models and reliably predict new view selections not used in the training.

Moreover, our methodology offers several benefits beyond the specific goodness measures recommended herein. First, we gain insight by examining the relative effectiveness of various components of our goodness measure, as well as other methods proposed in the literature. For example, we find that the mesh saliency approach described by Lee *et al.* [2005] is not as effective at describing our data as projected area, a much simpler model. The optimization procedure can incorporate any attribute or combination of attributes *a posteriori* and also evaluate any proposed model for viewpoint preference using only the data acquired in our user study.

Our paper also considers several straightforward applications for the goodness measure for views. Supposedly simple tasks such as orbiting around a 3D shape (with the implied goal of *understanding* it) are often reduced to a “turntable” with the camera rotating above the equator [Google 2010], even where another path might reveal the shape more effectively. We offer several tools motivated by this observation. The first kind of tool automatically selects either good individual viewpoints or an orbit around an object designed to pass

through good views as much as possible. Likewise, a second class of tool helps the user navigate in camera space by gently nudging the camera towards good views and away from bad views.

The main contributions of this paper are:

- an evaluation of 14 attributes from the literature that encode a range of desirable properties w.r.t. understanding and exploring a 3D shape,
- a user study and methodology by which we evaluate and optimize combinations of these attributes,
- a set of simple recommended measures for practical applications,
- a large dataset resulting from the study, publicly available for download by other researchers,
- an optimization tool that finds good individual views of an object or a smooth orbit around an object that passes through good views, and
- an interface that allows users of various skill levels to navigate around 3D shapes using a commodity 2D input device such as a mouse or touch screen, or even 1D input widget such as a scrollbar.

2. RELATED WORK

Our work concentrates on finding good views of a single object. While much work has been done on 3D camera control in scenes (e.g. [Drucker and Zeltzer 1995; Byers *et al.* 2003]), full 3D camera control is outside of our scope, and we refer to the excellent taxonomy of general methods for camera control in 3D by Christie and Olivier [2008].

Several researchers have investigated ways of evaluating the goodness of a view. For example, Kamada and Kawai [1988] attempt to minimize the number of degenerate faces in orthographic projection. Plemenos and Benayada [1996] describe a measure for goodness of views based on projected area of the model, while Roberts and Marshall [1998] compute multiple views that, combined, cover the entire surface of the model as well as possible, using an approximation of the aspect graph [Koenderink and Doorn 1979]. Scene visibility is also used for camera placement in the work of Fleishman *et al.* [1999]. Blanz *et al.* [1999] perform studies that show what attributes are important for determining canonical views for humans, following the seminal work by Palmer *et al.* [1981] that first introduces the *notion* of canonical views as well as the first such study. In the work of Gooch *et al.* [2001], an optimization process adjusts camera parameters to produce more “artistic” compositions by causing silhouette features to match known compositional heuristics such as “the rule of fifths.” Vazquez *et al.* [2001] coin the term *viewpoint entropy*, inspired by Shannon’s information theory [1948] and based on relative area of the projected faces over the sphere of directions centered in the viewpoint. Sokolov and Plemenos [2005] use dihedral angles between faces and discrete gaussian curvature. Also inspired by information theory, Lee *et al.* [2005] describe *mesh saliency* and show how it can be used to optimize viewpoint selection. Yamauchi *et al.* [2006] partition the view sphere based on silhouette stability [Weinshall and Werman 1997], and then use mesh saliency to determine the best view in each partition. Finally, Podolak *et al.* [2006] describe a goodness measure based on object symmetries.

In principle, the tools we present in this paper could make use of any of these goodness measures, and indeed we investigate and combine several of them. To the best of our knowledge, Polonsky *et*

al. [2005] were the first to explore a number of different attributes, which they call *view descriptors*. They suggest the possibility of a combined measure, but leave this as future work. Most closely related to our own is the work of Vieira *et al.* [2009]. They train a support vector machine classifier using a small set of tuples, one for each view, where each consists of a vector containing concatenated goodness values, and a user-provided binary preference for this specific view. They compare to individual goodness measures, and show that using a combination better fits to a wider range of models—an inspiration for our work. Their approach is designed for user interaction on a small set of models with similar objectives, and they leave a validating user study for future work. In contrast, our motivation is to find a goodness measure designed for a broad range of models and applications, and is based on a large user study.

The tools presented in this paper are designed to work for both static and moving cameras, either under guided user control or as a path designed to offer a good overall view of the object. Barral *et al.* [2000] optimize paths for a moving camera so as to provide good coverage of an overall scene, but the resulting paths appear to be unpleasantly jerky. In follow-up work to [Yamauchi *et al.* 2006], Saleem *et al.* [2007] show how to compute smooth animation paths that connect the best viewpoints, and adjust zoom and speed according to the goodness along the path. In comparison, the paths computed by our method are smooth, but additionally optimize the integral of goodness along the path. Recent work by Kwon and Lee [2008] shows how to optimize a camera path given animated character motion as input, where the goal is to optimally cover the space swept out by the motion.

3. MEASURING VIEW GOODNESS

In this section we describe our process for obtaining a measure for the goodness of views. First, we describe a set of view-dependent attributes that we will later combine to form various overall goodness metrics. Next, we present the results of a large user study in which we gather information about subjects’ preferences among pairs of nearby views. Finally, we use the data from the study to optimize the weights of our combined goodness measures, and also evaluate the relative contributions of the different attributes.

3.1 Attributes of Views

As reviewed in Section 2, the literature offers many attributes of views that may contribute to the overall goodness. However, previous efforts have typically considered just one attribute in forming a goodness measure. Obviously no single measure taken alone fully characterizes what people consider good, and one would expect different measures to combine with differing relative impact overall. In this section we present a group of attributes with the hope that they may combine to form a more accurate measure than any one or a few measures taken alone. These measures are visualized over the sphere of viewing directions for one model in Figure 1. Each of the attributes we selected is taken directly from the literature or inspired by previously described attributes. In the presentation below, the attributes are organized into five categories relating to different aspects of a view, for example, surface area or silhouettes.

3.1.1 Area attributes. Area attributes are related to the area of the shape as seen from a particular viewpoint.

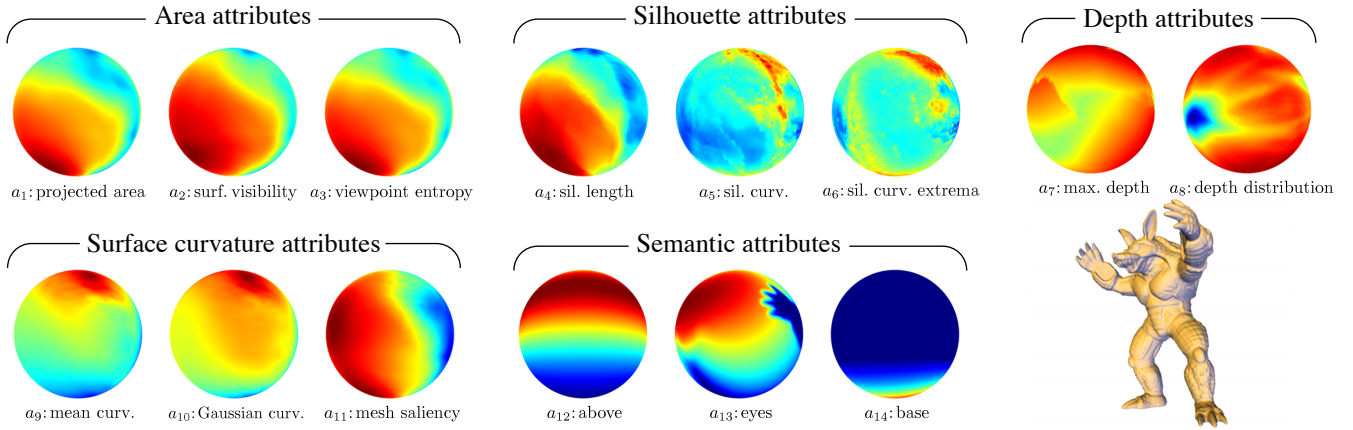


Fig. 1: Five groups of attributes, visualized over the sphere of viewing directions, for the armadillo model shown in the lower right. Color values range from blue (low) to red (high).

a_1 : **Projected area.** This attribute is the projected area of the model in the image plane as a fraction of the overall image area. Introduced by Plemenos and Benayada [1996], this measure is generally maximized by non-degenerate views.

a_2 : **Surface visibility.** Plemenos *et al.* [1996] define the surface visibility as the ratio of visible surface area in a particular view to the total surface area of an object. Maximizing surface visibility should reduce the amount of hidden surface of an object.

a_3 : **Viewpoint entropy.** Introduced by Vázquez *et al.* [2001], this attribute converts projected areas of mesh faces into a probability distribution and measures the entropy of the result. Since the original viewpoint entropy method employed a spherical camera, we use the extension to perspective frustum cameras due to Vázquez and Sbert [2002].

3.1.2 *Silhouette attributes.* Silhouette features are believed to be the first index into the human memory of shapes [Hoffman and Singh 1997], and, as a direct result of Shannon’s information theory, it was known as early as the 1950s that edges and contours contain a wealth of information about a 3D shape [Attneave 1954; Koenderink and Doorn 1979].

a_4 : **Silhouette length.** The overall length a_4 of the object’s silhouettes in the image plane, expressed in units of the average dimension of the image plane. This attribute is correlated with the appearance of holes and protrusions such as arms or legs.

a_5 : **Silhouette curvature.** Vieira *et al.* [2009] introduce silhouette curvature as an attribute, defined as

$$a_5 = \int |\kappa(\ell)| d\ell$$

where the curvature κ is parameterized by arc length ℓ .

a_6 : **Silhouette curvature extrema.** While silhouette curvatures will capture general complexities in the silhouette of an object, we are often interested in sharp features such as creases or the tips of fingers. To that end, we introduce a simple measure that emphasizes high curvatures on the silhouette:

$$a_6 = \int \kappa(\ell)^2 d\ell.$$

However, through experimentation we found it best to drop the curvatures found at depth discontinuities (i.e. T-junctions) as these areas sometimes contribute high curvatures without obvious connection to visual interest or features.

3.1.3 *Depth attributes.* Depth attributes are related to the depth of the shape as seen from a particular viewpoint; similar to area attributes, depth attributes can help avoid degenerate viewpoints.

a_7 : **Max depth.** The maximum depth value of any visible point of the shape is used to avoid degeneracies in [Stoev and Straßer 2002].

a_8 : **Depth distribution.** Since the maximum used in a_7 is noisy, we also introduce an attribute that is designed to encourage a broad, even distribution of depths in the scene:

$$a_8 = 1 - \int H(z)^2 dz$$

where H is the normalized histogram of the depth z of the object, sampled per pixel. This measure becomes small when H is “peaky” (most of the object is at a single depth) and is maximized when H is “equalized” (a range of depths are visible). Thus, a_8 encourages objects with largely planar areas to take oblique rather than head-on views, conforming to a human preference observed by Blanz *et al.* [1999].

3.1.4 *Surface curvature attributes.* Geometric surface curvatures of the shape are assumed to be related to the shape’s semantic features and are easily computed.

a_9 : **Mean curvature.** We compute the mean curvature on the surface of the object using [Meyer *et al.* 2002]. We consider curvature magnitudes to be relevant (not generated by noise) if they could be generated by a feature larger than 1% of the object’s size. We then linearly map the curvature values into $[0, 1]$ and compute the mean value visible at a particular viewpoint:

$$a_9 = \frac{1}{A_p} \int_{x \in A_p} \left[\frac{|h(x)|}{h_{\max}} \right]_{01} dA.$$

Here, A_p is the projected screen area occupied by the object, h_{\max} is the absolute value of largest relevant curvature, and the operator

$[*]_{01}$ clamps its argument to the range $[0, 1]$. We use the absolute value of the curvature to avoid cancellations in the integration.

a_{10} : **Gaussian curvature.** Gaussian curvature is also used in previous work ([Page *et al.* 2003; Polonsky *et al.* 2005]); we compute Gaussian curvature on the surface of the object using Meyer *et al.*'s angle defect formula [2002]. We treat the computed Gaussian curvatures analogously to the mean curvatures:

$$a_{10} = \frac{1}{A_p} \int_{x \in A_p} \left[\frac{|k(x)|}{k_{\max}} \right]_{01} dA.$$

a_{11} : **Mesh saliency.** The final surface curvature attribute is *mesh saliency*, introduced by Lee *et al.* [2005]. Mesh saliency is constructed from the mean curvature of the surface at multiple levels of detail; Lee *et al.* apply this attribute for both mesh simplification and viewpoint selection. As defined by Lee *et al.*, the attribute a_{11} is the *total sum* of mesh saliency visible from a viewpoint. We note that this confounds two factors—average mesh saliency and the projected area measure a_1 described above. While it would probably be wise to decorrelate these two factors, we use a_{11} as described in the literature for easier comparison.

3.1.5 *Semantic attributes.* Much of the previous work in automatic viewpoint selection has avoided the use of semantic features preferring that view goodness can be fully computed from the geometry of the object. We include semantic features because we believe that they are important in human preference, and we will be able to measure this importance in Section 3.3.

a_{12} : **Above preference.** Blanz *et al.* [1999] observe that people tend to prefer views from slightly above the horizon. Based on their observation, Gooch *et al.* [2001] initialize their optimization for “artistic” compositions from such a view. Thus attribute a_{12} favors these views with a smooth falloff towards the poles:

$$a_{12} = \mathcal{G} \left(\phi; \frac{3\pi}{8}, \frac{\pi}{4} \right),$$

where ϕ is the latitude with 0 at the north pole and $\frac{\pi}{2}$ at the equator, and $\mathcal{G}(x, \mu, \sigma)$ is the non-normalized Gaussian function $\exp(-(x-\mu)^2/\sigma^2)$. The a_{12} attribute peaks at $\frac{\pi}{8}$ above the equator and is minimal at the south pole. For objects with no inherent orientation, such as the heptoroid and rocker arm models (Figure 6), we simply set this term to zero. Nevertheless, typical computer graphics models generated by CAD or acquisition processes do indeed have a stored up-direction, and it is also possible to use techniques like those of Fu *et al.* [2008] to determine the orientation of man-made objects with unknown up directions.

a_{13} : **Eyes.** When the object of interest is a creature with eyes or a face, we observe that people strongly prefer views where the eyes can be seen [Zusne 1970]. Thus, attribute a_{13} measures how well the eyes of a model can be seen, when appropriate. In our system we mark the eyes by hand, by annotating a central vertex on the surface with a tiny radius. We note that just as technology for automatic face detection in images and video has matured, we expect that analogous algorithms for 3D models will become robust in the future and will obviate this manual task. To measure a_{13} , we simply sum this “eyes” surface value for all visible pixels. Most pixels do not contribute, so the behavior of this attribute is roughly that of a delta function for visibility attenuated by a cosine term for oblique views. For objects without eyes, this attribute is set to zero.

Table I. : Time to precompute the 14 attributes for a single viewpoint, averaged over 10242 views, for a range of example models. One of the simpler models, the fish, was fastest at more than 5 FPS, whereas the slowest was Lucy, the most complex of our models, at less than 1 FPS. Timings are reported for a single thread on a 2.26 GHz Intel Core 2 Duo with 4 GB of memory assisted by a NVIDIA GeForce 9400M graphics card.

Model	Faces	Time(ms)
Fish	11K	194
Airplane	24K	260
Dragon	100K	483
Lucy	526K	1256

a_{14} : **Base.** Just as people tend to prefer seeing eyes, they tend to avoid views from directly below for objects that have an obvious base on which they sit. The attribute a_{14} , measures the amount that the hand-marked base is visible, using the same strategy as for eyes. While we mark these features by hand, for many models they could be found using the automatic method of Fu *et al.* [2008]. Note that we distinguish the base from the eyes because we expect their behaviors to be anti-correlated, and because some models will have eyes, some will have base, some will have both, and some neither.

3.1.6 *Implementation.* Our implementation uses an image-based pipeline to avoid dependencies on mesh representation. To compute the attributes described above we render the object into an ID, a depth and several color images, and then use image processing to compute projected area, silhouettes, and so forth. For mesh saliency, we use the implementation of Lee *et al.* to assign a value at every vertex as a preprocess. Similarly, we compute the mean and Gaussian curvatures of the surface using the method of Meyer *et al.* [2002]. For any particular view we render a “color” buffer containing these values interpolated across faces and compute the appropriate quantities on the resulting rendered images. The eyes and base attributes are computed in a similar way.

Table I shows the time required to precompute the 14 attributes for a range of example models in our unoptimized implementation.

3.2 Collecting human preferences

Here we describe a study we performed in order to collect data about the relative goodness of views according to human preferences. In the next section we will use this data to train a model for view goodness that combines the attributes described above. It will also allow us to remark on the relative importance of the individual attributes in forming a combined measure of goodness. In order to design a study to meet these goals, a number of issues need to be addressed.

The first concern is what models to use for the study. Researchers performing perceptual studies often resort to models of abstract shapes like “blobbies” or geons. However, for several reasons we prefer to use models that are more recognizable. First, the resulting data will better characterize the kinds of models that we work with in computer graphics. Second, it is easier for people to express view preferences when they understand what they are seeing, so we believe the data will be both more meaningful and less noisy. Nevertheless, we would like the models to represent a broad range of shapes and objects. We selected 16 models, some scanned from real objects and some modeled via software, and they may be seen in the upper four rows of Figure 6. Eight of the models will be

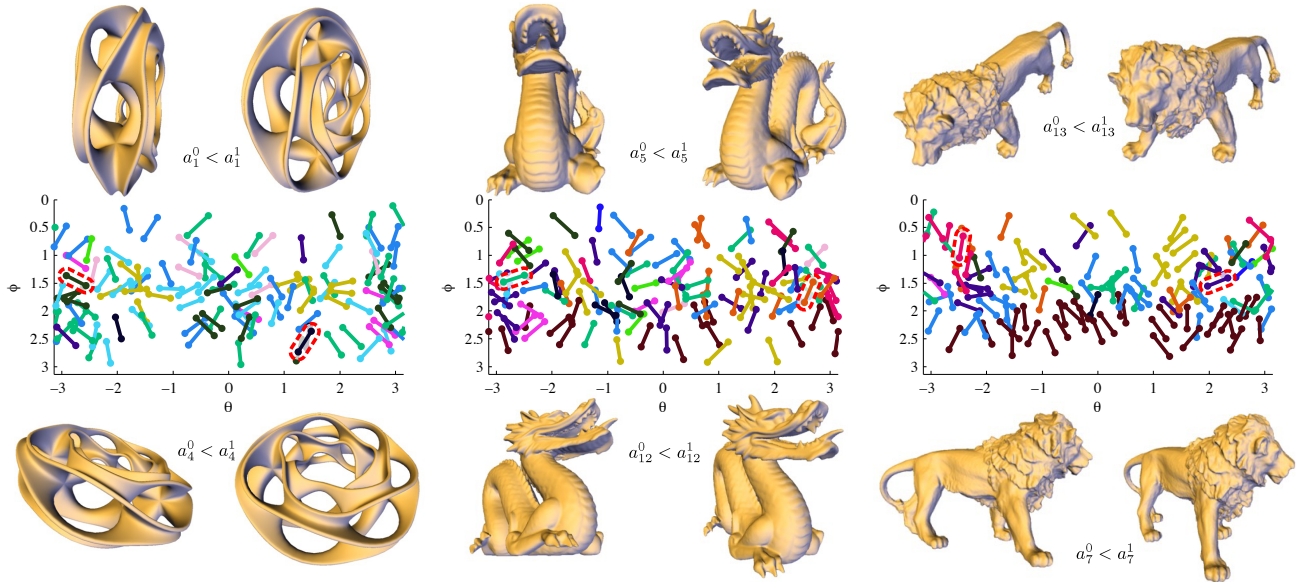


Fig. 2: Distribution of the pairs of selected views for three objects used in our study (120 pairs per object). In the middle the pairs are plotted in the $\theta \times \phi$ domain, and colored by the attribute in which they vary most. To illustrate typical image pairs, we highlight pairs that were particularly dominant in a specific attribute—top-left: projected area, bottom-left: silhouette length, top-middle: silhouette curvature, bottom-middle: above preference, top-right: eyes, bottom-right: max. depth. In each plot, the highlighted pair on the left side of the plot corresponds to the image pair at the top, and the pair on the right corresponds to the image pair on the bottom.

familiar to graphics researchers (armadillo, dragon, etc.) and the other eight are selected from separate categories in the Princeton Shape Benchmark [Shilane et al. 2004]. All but one of the objects are recognizable shapes even for people who have never seen them before. The heptoroid model is more abstract but still easy to understand from most or all views.

Our goal is to learn by asking people which views they prefer and by how much. Unfortunately there is no absolute scale for this kind of preference, since one person’s “pretty nice” is another person’s “okay.” Moreover such judgements are not quantitative. A standard strategy in such scenarios is to use the method of paired comparisons [David 1963], which asks people a simpler question: “Which of these two views do you prefer?”—a two-alternative forced choice experiment (2AFC). In principle, by asking many people this question for many pairs of views it will be possible to establish an overall ranking for all views. Standard practice would be to ask this question for either all pairs or many random pairs. However, in designing our study we found it to be more effective to ask the question only for pairs of *nearby* views. In particular we found that an angular separation of $\frac{\pi}{8}$ radians provides a nice balance in that the views are sufficiently far apart that the difference between the images is obvious, and yet similar enough that it does not feel like an “apples-to-oranges” comparison. We additionally fix the orientation of any rendered view such that the up vector of the model is aligned with the up vector in the image plane, so nearby views typically have similar orientations.

Regardless of how similar or different the views are, the instructions given to the subjects are critical. We want people to consider the shape of the object when choosing a view, but “shape” means different things to different people, especially graphics non-experts. Therefore, inspired by language in the study of Blanz *et al.* [1999], we provide the following instructions to our subjects: *Which of the*

two views of the object shown below reveals its shape better? For example, suppose that you had to choose one of these two pictures to appear in a magazine or product advertisement. Do not worry if neither of them is ideal. Just click on the one that you think is better.

The next issue to address is how to choose the particular pairs of views to be used in the study. The natural goal is to choose pairs randomly but roughly uniformly distributed over the sphere. We use rejection sampling with a probability distribution that includes a term that discourages choosing views that are close to previously selected views. In addition, we include two more criteria for rejection. First, in order to avoid pairs where the model appears to have rotated substantially in image space, we include a term that discourages pairs from varying much in the longitudinal direction near the poles. This term falls off with latitude so that at the horizon pairs vary equally in θ and ϕ , as can be seen for the sets of pairs shown in Figure 2. Second, if we sample the sphere uniformly, it is possible that many pairs will not exhibit strong differences in the attributes and that for a particular attribute a_j we may not get a range of variation among the pairs. Therefore, we also include a rejection policy that favors pairs of views in which attributes are varying. Specifically we use the sum of absolute differences in each attribute between the pairs, where each attribute is scaled in terms of standard deviations (because they have different ranges of values). Using this strategy we selected 120 pairs of views for each of the 16 models (3,840 images in total). Figure 2 shows the pair distributions for three models, colored by the attribute by which each pair varies the most. Of course all attributes vary somewhat for every pair, but it is easy to see that across all pairs every attribute has substantial variation. (See Figure 3 for the color coding.)

We ran our study on the Amazon Mechanical Turk (AMT), a service that allows researchers (and others) to provide small jobs for anonymous workers over the Internet for a small amount of

money. The use of AMT is increasing in computer graphics and human-computer interaction research, see, for example, [Heer and Bostock 2010], [Downs *et al.* 2010] and [Cole *et al.* 2009]. In our study, each job (“HIT” in Mechanical Turk terminology) was to make a choice for each of 30 images. For each person, the pairs were presented in random order, and for each pair the two images were randomly shuffled left-right. To filter out careless subjects, we resort to the strategy of Cole *et al.* [2009] in which every pair is shown to the worker twice (show 30 pairs, shuffle, then repeat) and we only retain the data from HITs where the two answers were reasonably consistent. Specifically, we require that 22 or more of the 30 answers were answered the same way the second time. This affords us reasonable confidence ($p > 0.99$) that the worker was not picking randomly, and also keeps data where the user could not make up their mind a substantial fraction of the time.

After discarding the inconsistent data we have between 30 and 40 people expressing a choice in each pair, for a total of 2,119 HIT assignments and 127,140 choices overall. This data was collected from 524 unique workers. Many subjects did multiple HITs, but they were never presented the same pair in more than one HIT. The most active subject worked on 51 HITs, and the histogram falls off roughly with a power law shape where most workers did just one HIT.

3.3 Modeling Viewpoint Preferences

The study produces a set of data that we can now use to evaluate predictive models of viewpoint preference, and to fit models of viewpoint goodness. The data is as follows: each pair i was shown twice to $n_i/2$ people, meaning there were n_i opportunities to pick one image or the other. Let’s say that view v_i^0 was picked k_i times, and v_i^1 was chosen $n_i - k_i$ times. If we have a probabilistic model that predicts how often a user would choose v^0 over v^1 , call it $P(v^0, v^1) \equiv P(v)$, then we can compute the likelihood that such a model explains our observed data.

3.3.1 Likelihood of observing our data. Out of the n_i people who see pair i , the probability that exactly k_i will choose view v_i^0 is given by the binomial distribution:

$$\binom{n_i}{k_i} P(v)^{k_i} (1 - P(v))^{n_i - k_i}$$

Thus, the likelihood \mathcal{L} of seeing these observations, over all M pairs is:

$$\mathcal{L}[P(v)] = \prod_i^M \binom{n_i}{k_i} P(v)^{k_i} (1 - P(v))^{n_i - k_i}.$$

As usual when dealing with probabilities, the log-likelihood form is more convenient:

$$\begin{aligned} \mathcal{L}^*[P(v)] &= \sum_i^M \ln \binom{n_i}{k_i} + \ln(P(v)^{k_i}) + \ln((1 - P(v))^{n_i - k_i}) \\ &= c + \sum_i^M k_i \ln P(v) + (n_i - k_i) \ln(1 - P(v)) \end{aligned}$$

3.3.2 Interpreting likelihood values. To gain some intuition about the range of likelihood values, we can compute the likelihood of two basic predictive models: a *naïve* model that randomly guesses and an *oracle* that has perfect knowledge. Consider first

the naïve model that predicts either image with equal likelihood; $P_{\text{naïve}} = 1/2$ and its value of $\mathcal{L}_{\text{naïve}}^*$ is -33203 . On the other end of the spectrum, we can consider an oracle with complete knowledge of the users’ selections. When the subjects select view v^0 over v^1 k out of n times, the oracle predicts a probability of k/n for that selection; $P_{\text{oracle}} = k/n$ and $\mathcal{L}_{\text{oracle}}^*$ is -3688 on our data. The naïve predictor and the oracle provide a convenient frame of reference for *any* model’s performance: we express the model *fitness* as $F[P] = (\mathcal{L}^*[P] - \mathcal{L}_{\text{naïve}}^*) / (\mathcal{L}_{\text{oracle}}^* - \mathcal{L}_{\text{naïve}}^*)$. The closer a model’s fitness approaches unity, the better it predicts the response of our users.

3.3.3 Goodness functions for single viewpoints. The preceding discussion focussed on models that operate on *pairs* of viewpoints; while this matches our collected data, for practical applications we would prefer a model that predicts the goodness of a *single* viewpoint. Given goodness values for single viewpoints $G^0 \equiv G(v^0)$ and $G^1 \equiv G(v^1)$, we need a prediction of how often a user would choose v^0 over v^1 . There are many possible models for this response and, generally, this method of paired comparison has been an active area of research since the 1920’s [David 1963]. The well-studied Bradley-Terry model [1952] characterizes the probability P that a person will choose v^0 over v^1 as the sigmoid-shaped logistic function of the difference in their inherent goodnesses:¹

$$P(G^0, G^1) = \frac{1}{1 + e^{-\sigma(G^0 - G^1)}}$$

Examining Figure 3, note that many of the attributes already have sigmoid-like shapes, indicating some weak explanatory power, even individually.

Given a particular model of viewpoint goodness G such as a single attribute, or a weighted combination of various attributes, we can compute the probability $P(G^0, G^1)$ that a user would select the first view, then evaluate the likelihood $\mathcal{L}^*[P(G^0, G^1)]$ that this model explains our data, and finally assign a fitness value $F[G]$ relative to the performance of the oracular and naïve models. Schematically, we have:

$$G \xrightarrow{\text{B-T model}} P \xrightarrow{\text{user data}} \mathcal{L}^* \xrightarrow{\text{oracle \& naïve}} F$$

We now explore various models of viewpoint goodness, with the intent of discovering important attributes and providing the practitioner with practical models.

3.3.4 Single-attribute models of goodness. As described in Section 3, each viewpoint is associated with $N = 14$ attributes. Figure 3 shows raw fitnesses of the individual attributes. We first explore the simplest models of viewpoint goodness: goodness is given by a single weighted attribute: $G_i = a_i$. We can fit the value of σ in the Bradley-Terry model to the data by any convenient 1D optimization procedure. To avoid over-fitting the data we use 100 trials of random sub-sampling validation: in each trial the weights were trained on a random subset of half of the objects and tested against the other half. The final weights are the means of the 100 trials, and are shown in Table II. We can see that surface visibility plays the single strongest predictive role; on the other hand, silhouette curvature does not appear to do as well on its own. However, when

¹Though we refer the reader to [Bradley and Terry 1952] for the details, the Bradley-Terry model follows from the intuitive idea that if view 0 has goodness G^0 and view 1 has goodness G^1 , then the probability that v^0 is picked over v^1 is $G^0 / (G^0 + G^1)$.

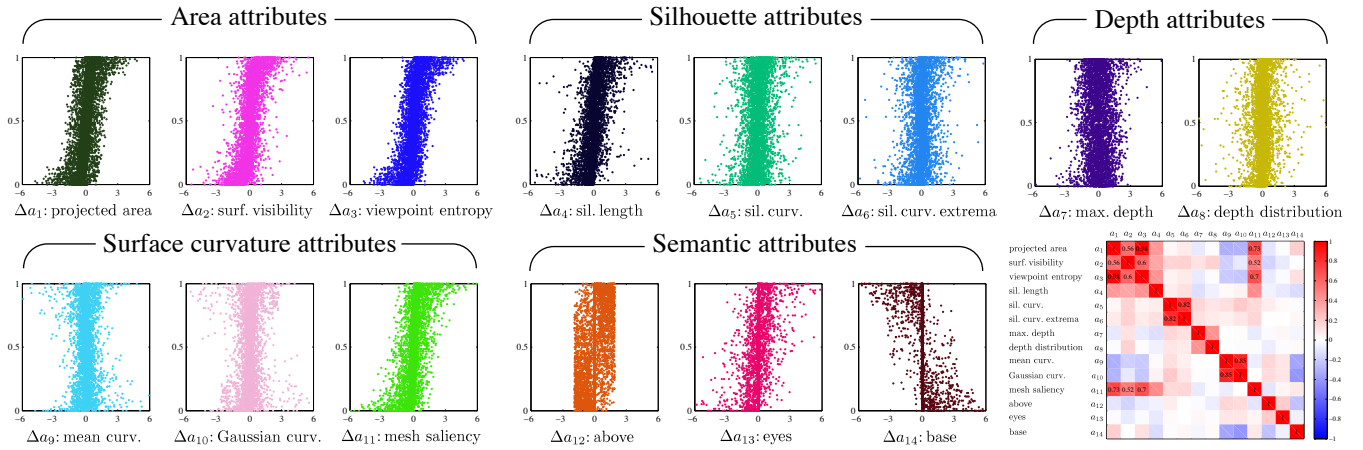


Fig. 3: Plots of the difference in each attribute value in a viewpoint pair versus the user's preference for the first image in the pair. The horizontal axes are in units of standard deviations of the differences in attribute value across all pairs of images. The lower-right shows the linear correlation coefficients between pairs of attributes, with the values of highly-significant correlations marked. Note the three strong clusters: area attributes (projected area, surface visibility, viewpoint entropy and mesh saliency), silhouette curvature attributes (silhouette curvature and silhouette curvature extrema), and surface curvature attributes (Gaussian curvature and mean curvature).

Table II : Fits of individual attributes to our study data. Fitnesses that are listed as exactly zero are statistically indistinguishable from zero at a significance level of $p = 0.05$, the rest are all significant.

Model	F	$\sigma(F)$
Oracle	1	0
Surface visibility	0.38	0.072
Viewpoint entropy	0.37	0.092
Projected area	0.28	0.110
Mesh saliency	0.26	0.100
Sil length	0.20	0.120
Above	0.16	0.064
Base	0.13	0.087
Eyes	0.09	0.033
Sil curvature	0.01	0.040
Sil curvature extrema	0	0.032
Depth distribution	0	0.020
Max depth	-0.02	0.021
Abs mean curvature	-0.15	0.260
Abs Gaussian curvature	-0.15	0.180
Naïve	0	0

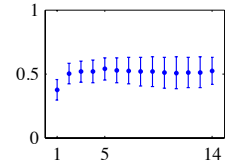
combined with other attributes, it may perform quite differently: see Section 3.3.5.

3.3.5 Linear models of goodness. The natural extension to the single-attribute models are the *linear- K* models that combine K attributes to form a goodness value:

$$G(v) = \sum_{j \in S} w_j a_j$$

where v is a viewpoint and S is the set of indices of attributes used in the particular model ($|S| = K$). Since σ in the Bradley-Terry model is made redundant by w_1 , we fix it to be one. We

can then optimize the values of the weights by maximizing the value of $F[G_i]$, which is nonlinear in the unknown weights w_i and the known attributes a_i of the views. However, the function is quite smooth and the downhill simplex method of Nelder and Mead [1965] find the optimal weights consistently and quickly for our data. There are $\binom{14}{2} = 91$ linear-2 models, $\binom{14}{3} = 364$ linear-3 models, etc., for a total of 16383 possible linear models. We separately trained and tested all 16383 models using, as before, 100 trials of repeated random sub-sampling. The result is a distribution of fitnesses for each potential linear model, sampled 100 times. Given the statistical nature of the sampling, it is inappropriate to, say, simply select the model with the highest mean energy for some particular K : another run of 100 trials of training and testing might result in a slightly different ranking. Instead, we use Tukey's "honestly significant difference" (HSD) method, a multiple-comparison procedure [Hsu 1996], to identify the *pool* of models that perform statistically indistinguishably from the top-performing model. Shown on the right is the mean and standard deviations of the fitnesses of the pool of top-performing models for $K = 1 \dots 14$. Note that using more than five attributes does not improve the performance of the linear models.



Given that there are several top-performing linear models with K attributes, we recommend the following single-attribute, linear-3 and linear-5 models, spanning the useful range of K (Table III). In each class K , these recommended models are in the pool of highest-performing models, and are chosen with an eye towards computational simplicity of the component attributes. The simplest model is simply a_2 , surface visibility, and if more computational resources are possible, then we add a_{12} and a_4 , the above preference and silhouette length, respectively. Still better performance is possible by adding a_1 and a_7 , projected area and max. depth. If it is possible to include marked features such as eyes or base, we also suggest an alternative model, *linear-5b*, which swaps a_7 for

Table III. : Weights of viewpoint attributes for our recommended models of viewpoint goodness. Note that, since each attribute is scaled differently, the absolute values of weights are meaningless and comparing weights across attributes is not possible.

	a_2	a_{12}	a_4	a_1	a_7	a_{13}
Single	23					
Linear-3	18	2.8	0.51			
Linear-5	14	2.7	0.46	14	2.5	
Linear-5b	15	2.6	0.42	13		670

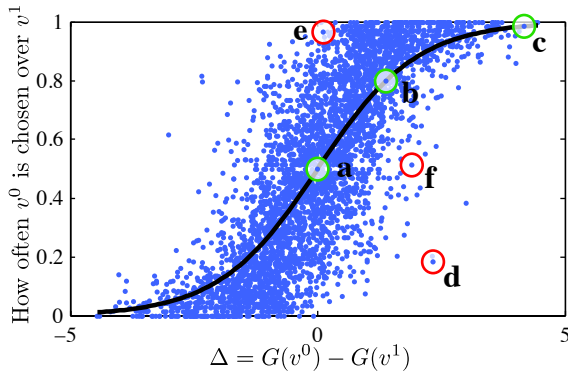


Fig. 4: Fit of differences in goodness values in the linear-5b model to observed user selections for 1920 pairs of viewpoints across 16 models. $G(v)$ is a linear combination of viewpoint measures fit to the probability model of Section 3.3. The points highlighted in green are particularly well-predicted by the model, while the points in red are not; the labels correspond to viewpoints in Figure 5.

a_{13} , the eyes attribute. Table IV lists the performance of our recommended models. We use the linear-5b model for the rest of the discussion that follows and in Figures 4, 6, 7, 8 and 9.

Figure 4 shows how the optimized weights of the linear-5b model fit the user’s preference data. Overall, the sigmoid shape of the logistic curve appears to fit the shape of the data well. While the “slope” of the curve appears to be slightly shallower than that of the data, this can be explained by the nature of the probabilities

Table IV. : Performance of predictive models. All models were tested using 100 trials of random sub-sampling validation: in each trial the models were trained on a random subset of half of the objects and tested against the other half. The resulting mean and standard deviation of test fitnesses are reported.

Model	$\langle F \rangle$	$\sigma(F)$
Oracle	1	0
10-NN	0.77	0.020
Quadratic	0.63	0.090
Linear-5b	0.58	0.060
Linear-5	0.58	0.060
Linear-3	0.55	0.062
Single	0.38	0.072
Naive	0	0

associated with the binomial distribution—that “errors” near $\Delta = 0$ are more easily explained, in a probabilistic sense, than those out in the tails of the curve. Examples of successes and failures of the model are marked in Figure 4 and the corresponding images are shown in Figure 5.

3.3.6 Quadratic models of goodness. We note that there is an obvious gap between the performance of our linear-5b model and that of the oracle. It is natural to wonder whether some other model might perform better. Perhaps the oracle is unapproachable—after all, it is unrealistic because *by definition* the oracle knows the answer for any user preference! One strategy would be to consider attributes other than the 14 that we have evaluated for the views used in our dataset, which we leave for future work. However, we can ask if there are other models that would use our attributes to better fit the data. A potential concern of a linear model is that it cannot characterize correlations between the attributes (see Figure 3 for evidence of their correlations). Therefore we consider a quadratic model as follows:

$$G(v_i^m) = \sum_j^N w_j a_{ij}^m + \sum_{jk}^N w_{jk} a_{ij}^m a_{ik}^m$$

This model has $14 + 105 = 119$ weights, and, when fit using the same procedure as before, has a mean fitness of $F = 0.63$ (Table IV), a moderate improvement over the various linear models.

3.3.7 Non-parametric models of preference. We also experimented with non-parametric fitting of the viewpoint preference data. In particular, we applied the K -nearest-neighbors model for various values of K [Fix and Hodges 1951]. We again randomly split the data into a training set and a test set by partitioning the data associated with a random subset of half the models. To find the likelihood of each viewpoint in the testing set, we first computed the attribute deltas for all 14 attributes, converted to units of standard deviations, then found the K -nearest neighbors in the training set using Euclidean distance. Once the K -nearest neighbors are found, the likelihood is computed in the same way as with the oracle, but averaged over the K -nearest neighbors. The performance of this model is shown for $K = 10$ in Table IV, averaged over 100 trials (similarly to the linear- K models, values of K greater than 10 did not improve performance). While the performance of this model is better than the linear or quadratic models, it is onerous to compute: to predict the preference for one view in a viewpoint pair, all 14 attributes must be computed for the two viewpoints and the minimum distance must be computed to the training set of 960 viewpoint pairs. In addition, the K -nearest-neighbors model does not directly provide the goodness for a single viewpoint, only the preference in a viewpoint pair.

Of course, any number of other models from machine learning might perform even better, but at the expense of both computational complexity and loss of intuition. Thus, for applications and results described in the remainder of the paper, we employ the linear-5b model. For example, based on this goodness model we show in Figure 6 the best view for each of the 16 shapes used in our study, and for 4 shapes not used in our study.

4. APPLICATIONS

In lieu of virtual hands and clay, most complex tasks in three dimensions, such as shape modeling or camera navigation, are

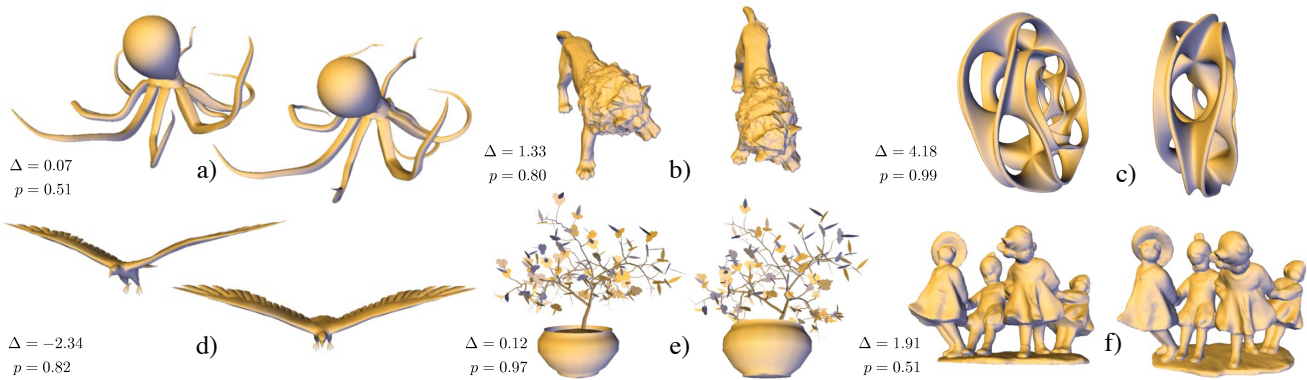


Fig. 5: Selected viewpoint pairs corresponding to the labelled points in Figure 4. In cases a) and f), the users expressed no clear preference, while in the other cases they preferred the right image. The upper row (a, b, c) shows example pairs of images that are well-predicted by our model, ordered by increasing values of predicted and actual goodness. The lower row shows pairs of images that are not well-predicted by our model. Our model predicts that users will select the left-most image of pair d) less than 20% of the time, but it was actually selected in over 80% of the trials; an anti-correlation. Our model predicts no preference for either view of pair e), yet users nearly always selected the left view. Finally, pair f) is overwhelming predicted to choose the left-most image, but users chose both images equally. The right-most image allows the second child's head to be more clearly seen, a quality that our attributes do not capture.



Fig. 6: The best view according to the six attribute model for each of the 16 training models (rows 1–4) and four additional model (last row).

generally performed in two dimensions when using contemporary computer hardware. True 3D input devices [Zhai 1998] are not widely deployed, perhaps because of the simplicity and commercial success of commodity 2D input devices such as the mouse or, more recently, multi-touch panels. At the same time, the complexity of 3D modeling packages [Maya 2010; 3ds Max 2010] has grown to a point where one requires significant training and practice to carry out even the most mundane tasks.

In the following, we propose several applications that use our goodness measure (or any other) to assist a user in finding good views and navigating around a 3D model.

4.1 Finding the N -best views

A straightforward application that uses our goodness measure is finding the N -best views of a model, as proposed and demonstrated in [Polonsky et al. 2005; Yamauchi et al. 2006; Vieira et al. 2009]. Instead of picking a fixed number of best views, we have decided to find those views that are most representative, and therefore employ *mean-shift clustering* [Comaniciu and Meer 2002] to find the dominant peaks in our goodness function. As a result, depending on the model chosen, we obtain varying numbers of best views. As can be seen in the distribution of good views over the viewing sphere, low frequency goodness functions result in only few representative views (Figure 7), whereas high frequency functions require a larger number of views to reveal all important features of the shape.

4.2 Periodic orbits and scrubbing

Closed-loop viewpaths provide a convenient 1D user interface to viewing the salient features of a 3D model, removing the complexities of the standard trackball interface. The user can *scrub* the viewpoint along the viewpath by means of a standard scroll bar widget or by linear mouse drags. This simplified 1D interface is appropriate for applications where the user might want to quickly preview the features of a model, for example, when browsing a large database of 3D models. In this scenario, models are often explored simply by rotating around the equator [Google 2010].

The method presented by Barral *et al.* [2000] computes a greedy path that suffers from visibly jaggy motion. Saleem *et al.* [2007] correct this by smoothly connecting stable and salient viewpoints. We go a step further, and compute a path such that the integral of goodness along the path is maximized. Similar to [Saleem et al. 2007], we initialize a closed loop such that it passes through regions of high goodness using our N -best views algorithm de-

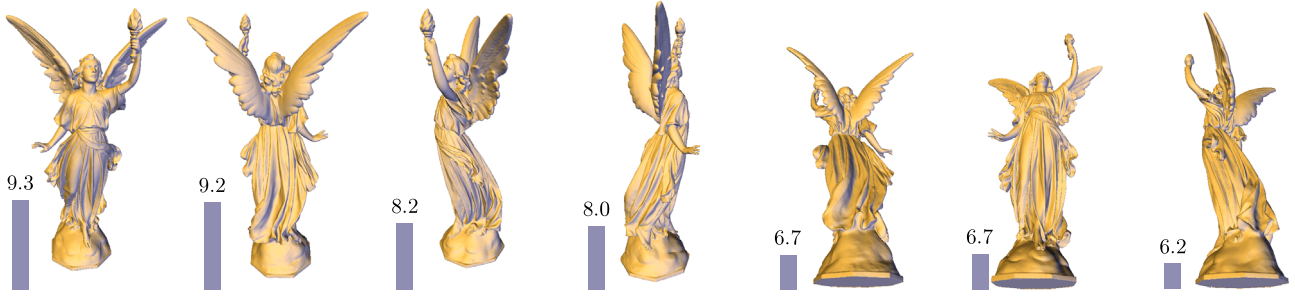


Fig. 7: The seven best views of the Lucy model, selected using the mean shift algorithm on the linear-5b model. Using the mean shift algorithm on the viewing sphere, the number of views do not need to be pre-selected. The goodness value of each view is displayed on the left: note the clear distinction between the various types of views.

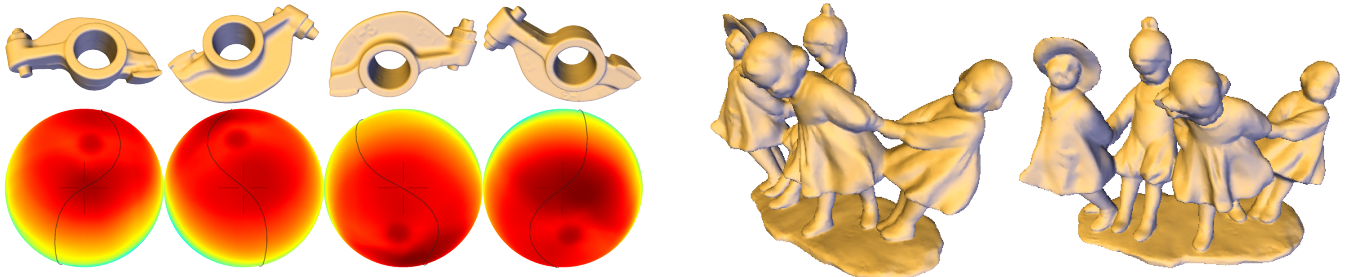


Fig. 8: A closed, periodic camera orbit passing through the best views of the rocker arm object.

scribed above. The path optimization then proceeds by optimizing a snake [Kass et al. 1988] on the view sphere, guided by the gradient of our goodness measure. To ensure that the path is of controllable length and smoothness, we use the common energy terms for stretch and bending. The result is a closed, periodic loop that passes through all good views, and can either be animated, or explored with a 1D scrub bar (Figure 8).

4.3 Trackball extensions

Note to the reviewer: the trackball and orbit interfaces are available for direct evaluation in the accompanying executable.

Finally, we can further enhance the traditional trackball interface with a goodness-based force model that gently guides the user towards good viewpoints and tries to keep them away from bad viewpoints. We term the two modes of the trackball *grab*, which is the mode of holding the mouse button down and moving the model, and *throw*, which is the animation path the model camera describes after the mouse has been released. These two modes of interaction are treated separately: while strong guidance can be perceived as a significant disturbance during the grab operation, the throw path can be adjusted with a larger force.

Grab As the user rotates the trackball, we apply a *nudge* force in the direction of and proportional to the gradient of the goodness function, but only use the component which is orthogonal to the current direction of motion. The summed force is scaled down to ensure that the nudged camera travels the same distance than the original camera would without the nudge. This adds a small resistance to motions that would travel to bad viewpoints and eases

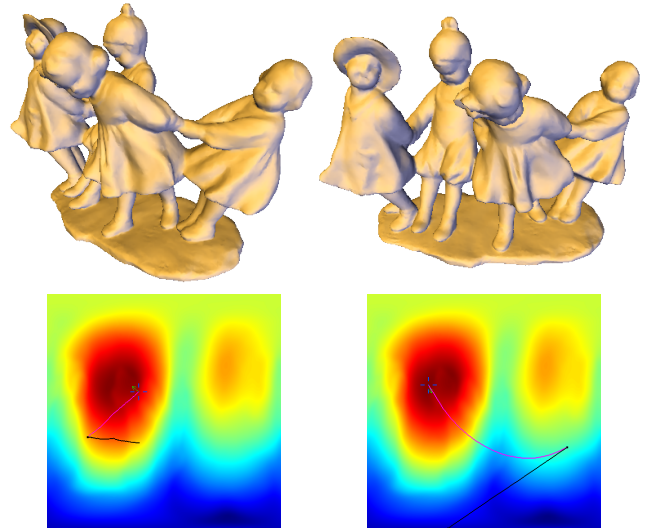


Fig. 9: Top: the trackball gently nudges the viewpoint towards better views while the user is dragging with the mouse. Bottom: if the user “throws” the trackball, the path is attracted by nearby high-quality viewpoints. The black line shows the original path without nudging, while the pink path shows the nudged viewpoint path experienced by the user.

motions towards good viewpoints. The nudge force is small in all cases, but increases with the speed of the user’s input, similar to how mouse acceleration depends on input speed in current desktop systems. The expectation is that when a user applies large, fast, coarse motions, they will end up at a good viewpoint more often than not. The results of this guidance force can be seen in Figure 9 (top). Note that the adjustments are locally subtle, but add up to large displacements in the direction of good viewpoints.

Throw In the standard trackball interface, once the mouse button is released, the animation around the model proceeds linearly, with the last rotation speed applied during grabbing. By adding our nudge force described previously, we can guide the animations towards better views. We furthermore add in some friction force that is inversely proportional to the goodness of the current viewpoint, so as to slow down near good viewpoints. To avoid overshooting good viewpoints, we pre-compute the entire animation path directly after switching from grab to throw, and search for the first point

along that path where the camera would turn π away from the initial throw direction (the turn-around point). From there we perform gradient descent to find the nearest off-path local maximum of our goodness function, and warp the path, from the turn-around point to the end, such that it ends in the best view. The adjustment to the animation path is more pronounced when compared to the modification during grabbing (Figure 9, bottom), but we can guarantee that the animation comes to a stop in good views. We expect that this will especially be beneficial for model exploration on devices with constrained touch input, such as Apple’s iPhone or iPad devices.

5. VALIDATION

Section 3 reports on the effectiveness of several trained goodness models at predicting people’s viewpoint preferences. Those experiments used cross-validation, meaning the training was performed on different set of models than were used for testing. Nevertheless, to further validate the method, we compared our derived goodness model linear-5b (Section 3.3.5) against human preferences for views on an independent set of models than had been used for previous evaluation, and using a different procedure described here.

For each of the four 3D models shown at the bottom of Figure 6, we generated 100 random viewpoints on the viewing sphere, using a rejection sampling procedure similar to that of Section 3.2 (except that each view is independently chosen, rather than in pairs).

In a new Mechanical Turk study similar to that of Section 3.2, we showed subjects random pairs of views of the same model (rather than pairs separated by a specific angle). In each HIT, the worker was shown 30 randomly chosen pairs, shuffled and repeated for consistency as before. After filtering out inconsistent HITs, data were gathered from a total of 1075 HITs performed by 233 subjects. Each of 400 viewpoints was shown between 234 and 370 times.

Since each viewpoint was compared against a random set of other views (as opposed to a single nearby view) we can use the fraction of the time each view was picked as a simple approximation of the “true goodness” of the viewpoint – good views are picked often, and bad views rarely. For comparison, this value is plotted (vertical axis) against the linear-5b goodness measure (horizontal axis) for two of the four models in Figure 10. Of the four models, the bunny model (left) exhibits the strongest correlation, while the city model (right) has the weakest correlation. Also shown is a best-fit logistic regression curve and its deviation from the data.

Table V reports the coefficients of the linear-5b term in the logistic regression for all four models, along with the standard errors and the t -values associated with these coefficients ($df = 98$). The t -values are extremely large ($p < 10^{-65}$) in all cases, indicating that linear-5b is a highly significant predictor of the kinds of views that our subjects preferred in this experiment. We believe that the reason the city model offers the weakest correlation is that it exhibits a number of qualitative differences from the other models in Figure 6, including an overall “boxiness” and a planar arrangement of multiple connected components.

6. CONCLUSIONS, LIMITATIONS AND FUTURE WORK

We have presented a perceptual model of viewpoint preference, and while our model performs well for the task carried out in our user study, we see this as the starting point for a much larger research

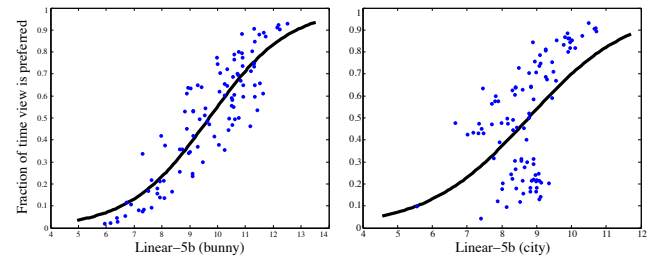


Fig. 10: Comparison of the linear-5b model against human preferences for views of the bunny (left) and the city (right). Each data point represents a single viewpoint; the curve shows the best fit logistic regression.

Table V. : Logistic regression for validation study. Coefficients of the linear-5b term in the logistic regression, and the standard errors and t -values associated with these coefficients ($df = 98$).

Model	Logistic	S.E.	t -value
Bunny	0.699	0.0099	70.6
David	0.822	0.0135	60.9
Buddha	0.481	0.0084	57.5
City	0.684	0.0152	45.0

effort. The method as proposed does have some limitations, and these generally identify interesting areas for future work.

We can only make quantitative claims about the 16 models for which we have collected data. While we picked models that have a range of visual features and qualities, there are certainly more classes of models to be explored. Furthermore, we are only searching over two camera parameters, which surely benefits the optimization procedure of Section 3.3. So, in addition to evaluating over a larger set of models, we also hope to search over more camera parameters, such as field-of-view, up vector, velocity, and the camera position in complex scenes [Christie et al. 2008].

There are many potential attributes of viewpoint goodness described in the literature, and we did not include every such attribute. In ongoing work, we hope to include broader classes of attributes. For example, Podolak *et al.* [2006] observe object symmetries can play an important role in selecting viewpoints. But also attributes such as lighting variation, occluding contours, texture and others could be included. It is important to note though, that the fact that silhouette curvature is surprisingly unimportant after model fitting, and that surface visibility is by far the most influential metric, points to some significant redundancies in *any* combined measure of viewpoint goodness. As a result, we eventually discarded some measures that initially appeared promising, such as “number of disconnected silhouette loops” which are heavily correlated with measures such as silhouette length. Thus, a more robust methodology for investigating of the correlations between various attributes is merited.

And finally, while our fitted model appears to be reasonable, it is error-prone and we have identified some of the most egregious error modes in Figures 4 and 5. This does point to the possibility that special-case combinations of measures will create better combined goodness measures for classes of models.

REFERENCES

3DS MAX. 2010. Autodesk, <http://www.autodesk.com/>

- 3dsmax.
- ATTNEAVE, F. 1954. Some informational aspects of visual perception. *Psychological Review* 61, 3, 183–193.
- BARRAL, P., DORME, G., AND PLEMENOS, D. 2000. Visual understanding of a scene by automatic movement of a camera, short paper. *Proc. Eurographics 2000*.
- BIEDERMAN, I. 1987. Recognition-by-components: A theory of human image understanding. *Psychological Review* 94, 115–147.
- BLANZ, V., VETTER, T., BÜLTHOFF, H., AND TARR, M. 1999. What object attributes determine canonical views? *Perception* 24, 575–599.
- BRADLEY, R. AND TERRY, M. 1952. Rank analysis of incomplete block designs, i. the method of paired comparisons. *Biometrika* 39, 324–345.
- BYERS, Z., DIXON, M., GOODIER, K., GRIMM, C., AND SMART, W. 2003. An autonomous robot photographer. *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on* 3, 2636–2641 vol. 3.
- CHRISTIE, M., OLIVIER, P., AND NORMAND, J.-M. 2008. Camera control in computer graphics. *Computer Graphics Forum* 27, 8, 2197–2218.
- COLE, F., SANIK, K., DECARLO, D., FINKELSTEIN, A., FUNKHOUSER, T., RUSINKIEWICZ, S., AND SINGH, M. 2009. How well do line drawings depict shape? In *ACM Transactions on Graphics (Proc. SIGGRAPH)*. Vol. 28.
- COMANICIU, D. AND MEER, P. 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 5, 603–619.
- DAVID, H. A. 1963. *The Method of Paired Comparison*. Hafner Publishing.
- DOWNS, J. S., HOLBROOK, M. B., SHENG, S., AND CRANOR, L. F. 2010. Are your participants gaming the system?: screening mechanical turk workers. *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, 2399–2402.
- DRUCKER, S. AND ZELTZER, D. 1995. Camdroid: A system for implementing intelligent camera control. *Proceedings of the 1995 symposium on Interactive 3D graphics*, 139–144.
- FIX, E. AND HODGES, J. 1951. Discriminatory analysis, nonparametric discrimination: Consistency properties. Tech. Rep. 4, USAF School of Aviation Medicine, Randolph Field, Texas. Unpublished technical report; see [Silverman and Jones 1989] for a re-printed version with commentary.
- FLEISHMAN, S., COHEN-OR, D., AND LISCHINSKI, D. 1999. Automatic camera placement for image-based modeling. *Computer Graphics Forum* 19, 12–20.
- FU, H., COHEN-OR, D., DROR, G., AND SHEFFER, A. 2008. Upright orientation of man-made objects. *ACM Transactions on Graphics* 27, 3 (Aug.), 42:1–42:7.
- GOOCH, B., REINHARD, E., MOULDING, C., AND SHIRLEY, P. 2001. Artistic composition for image creation. *Eurographics Workshop on Rendering*, 83–88.
- GOOGLE. 2010. Google 3D warehouse and SketchUp, <http://sketchup.google.com/3dwarehouse/>.
- HEER, J. AND BOSTOCK, M. 2010. Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. *Proceedings of Computer Human Interaction (CHI 2010)*. ACM CHI 2010 Best Paper Nominee.
- HOFFMAN, D. D. AND SINGH, M. 1997. Saliency of visual parts. In *Cognition*. Vol. 63(1). 29–78.
- HSU, J. 1996. *Multiple Comparisons: Theory and Methods*. Chapman and Hall/CRC.
- KAMADA, T. AND KAWAI, S. 1988. A simple method for computing general position in displaying three-dimensional objects. *Comput. Vision Graph. Image Process.* 41, 1, 43–56.
- KASS, M., WITKIN, A., AND TERZOPOULOS, D. 1988. Snakes: Active contour models. *International Journal of Computer Vision* 1, 4, 321–331.
- KOENDERINK, J. AND DOORN, A. V. 1979. The internal representation of solid shape with respect to vision. *Biol. Cybern.* 32, 211–216.
- KWON, J. AND LEE, I. 2008. Determination of camera parameters for character motions using motion area. *The Visual Computer* 24, 7 (July), 475–483.
- LAGA, H. AND NAKAJIMA, M. 2008. Supervised learning of salient 2D views of 3D models. *The Journal of the Society for Art and Science* 7, 4, 124–131.
- LEE, C. H., VARSHNEY, A., AND JACOBS, D. W. 2005. Mesh saliency. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*. 659–666.
- MAYA. 2010. Autodesk, <http://www.autodesk.com/maya>.
- MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. 2002. Discrete differential-geometry operators for triangulated 2-manifolds. *VisMath '02 Proceedings*.
- NELDER, J. A. AND MEAD, R. 1965. A simplex method for function minimization. *Computer Journal* 7, 308–313.
- PAGE, D., KOSCHAN, A., SUKUMAR, S., ROUI-ABIDI, B., AND ABIDI, M. 2003. Shape analysis algorithm based on information theory. *International Conference on Image Processing (ICIP 2003)* 1, 29–32.
- PALMER, S., ROSCH, E., AND CHASE, P. 1981. Canonical perspective and the perception of objects. *Attention and Performance IX*, 135–151.
- PLEMENOS, D. AND BENAYADA, M. 1996. Intelligent display in scene modeling: New techniques to automatically compute good views. *Proceedings of GraphiCon (1996)*.
- PODOLAK, J., SHILANE, P., GOLOVINSKIY, A., RUSINKIEWICZ, S., AND FUNKHOUSER, T. 2006. A planar-reflective symmetry transform for 3D shapes. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 25, 3 (July).
- POLONSKY, O., PATANE, G., BIASOTTI, S., GOTSMAN, C., AND SPAGNUOLO, M. 2005. What's in an image: Towards the computation of the best view of an object. *The Visual Computer* 21, 8-10, 840–847.
- ROBERTS, D. AND MARSHALL, A. 1998. Viewpoint selection for complete surface coverage of three dimensional objects. In *In Proc. of the British Machine Vision Conference*. 740–750.
- SALEEM, W., SONG, W., BELYAEV, A., AND SEIDEL, H.-P. 2007. On computing best fly. In *Proceedings of the 23rd Spring Conference on Computer Graphics*. ACM SIGGRAPH, Comenius University, 143–149.
- SHANNON, C. E. 1948. A mathematical theory of communication. *Bell System Technical Journal* 27, 379423.
- SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. The princeton shape benchmark. In *Shape Modeling International*.
- SILVERMAN, B. AND JONES, M. 1989. E. Fix and JL Hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on Fix and Hodges (1951). *International Statistical Review/Revue Internationale de Statistique* 57, 3, 233–238.
- SOKOLOV, D. AND PLEMENOS, D. 2005. Viewpoint quality and scene understanding. In *VAST, Eurographics Symposium Proceedings*. 67–73.
- STOEV, S. AND STRASSER, W. 2002. A case study on automatic camera placement and motion for visualizing historical data. *Proceedings of the conference on Visualization '02*.
- VÁZQUEZ, P. AND SBERT, M. 2002. Automatic keyframe selection for high-quality image-based walkthrough animation using viewpoint entropy. *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG '02)*.
- VÁZQUEZ, P.-P., FEIXAS, M., SBERT, M., AND HEIDRICH, W. 2001. Viewpoint selection using viewpoint entropy. In *VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001*. 273–280.
- VIEIRA, T., BORDIGNON, A., PEIXOTO, A., TAVARES, G., LOPES, H., VELHO, L., AND LEWINER, T. 2009. Learning good views through in-

- telligent galleries. *Eurographics 2009 (Computer Graphics Forum)* 28, 2, 717–726.
- WEINSHALL, D. AND WERMAN, M. 1997. On view likelihood and stability. *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 2, 97–108.
- YAMAUCHI, H., SALEEM, W., YOSHIKAWA, S., KARNI, Z., BELYAEV, A., AND SEIDEL, H.-P. 2006. Towards stable and salient multi-view representation of 3d shapes. In *SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006*. 265–270.
- ZHAI, S. 1998. User performance in relation to 3d input device design. *SIGGRAPH Computer Graphics* 32, 4, 50–54.
- ZUSNE, L. 1970. *Visual perception of form*. Academic Press.