Exploratory Modeling with Collaborative Design Spaces

Jerry O. Talton Daniel Gibson Lingfeng Yang Pat Hanrahan Vladlen Koltun

Stanford University*

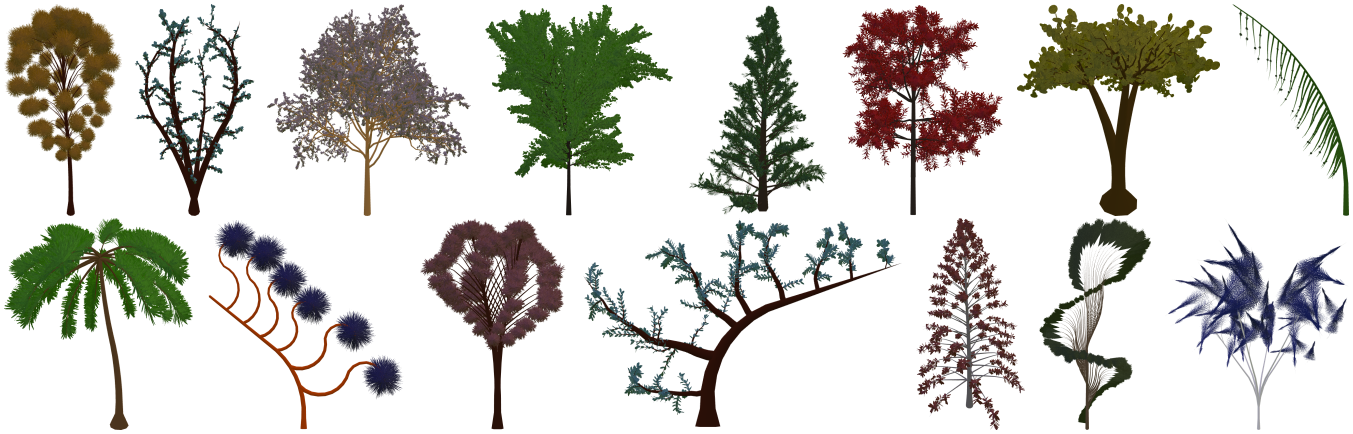

Figure 1: *Trees created by users of our prototype exploratory modeling tool in* ≤ 15 *minutes (top) and* ≤ 1 *hour (bottom).*

Abstract

Enabling ordinary people to create high-quality 3D models is a long-standing problem in computer graphics. In this work, we draw from the literature on design and human cognition to better understand the design processes of novice and casual modelers, whose goals and motivations are often distinct from those of professional artists. The result is a method for creating *exploratory* modeling tools, which are appropriate for casual users who may lack rigidly-specified goals or operational knowledge of modeling techniques.

Our method is based on parametric design spaces, which are often high dimensional and contain wide quality variations. Our system estimates the distribution of good models in a space by tracking the modeling activity of a distributed community of users. These estimates drive intuitive modeling tools, creating a self-reinforcing system that becomes easier to use as more people participate.

We present empirical evidence that the tools developed with our method allow rapid creation of complex, high-quality 3D models by users with no specialized modeling skills or experience. We report analyses of usage patterns garnered throughout the year-long deployment of one such tool, and demonstrate the generality of the method by applying it to several design spaces.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

Keywords: exploration, modeling, collaboration

*e-mail:{jtalton,gibsnson,lyang,hanrahan,vladlen}@cs.stanford.edu

1 Introduction

The increasing popularity of three-dimensional participatory media such as networked virtual worlds [Miller 2007], game mods [Newman 2006], and machinima [Marino 2004] has resulted in unprecedented involvement of ordinary people in 3D modeling activities. For instance, more than one hundred million original 3D models were created in 2008 and 2009 by players of the video game Spore [Maxis Software 2008]. However, despite this surge of interest from the general population, most 3D modeling tools remain notoriously complex and require considerable specialized training to be used effectively. This paper investigates the design of modeling tools for novice and casual users.

When developing such tools, it is important to understand the nature of the design tasks to which they will be applied. In this paper, we draw from the literature on design and human cognition to distinguish between *routine* and *exploratory* modeling processes. Routine design processes focus on the realization of a well-specified goal through a sequence of pragmatic steps. In contrast, exploratory design begins with loosely-specified goals and proceeds in an opportunistic, serendipitous fashion. The application of this distinction to 3D modeling is one of the main contributions of this work.

Although casual users often approach modeling tasks with vague or general goals, the vast majority of prior research in computer graphics has focused on routine modeling. This paper presents a method for creating exploratory modeling tools that can be successfully used by novice and casual users with little or no prior 3D modeling experience.

Given some selected model, our tools automatically construct high-quality alternative models and present these alternatives to the user. To provide this functionality, our method employs a distinct parametric space for each target visual domain. Such a space is defined by a mapping from a set of n real-valued parameters to a renderable entity, and the collection of valid parameter configurations, taken together, form an n-dimensional design space.

High-dimensional parametric spaces provide a compact representation for a wide range of visual models and are ubiquitous in graphics and engineering [Hoschek and Dankwort 1994]. Unfortunately, in many such spaces the probability that a random parameter configuration corresponds to a desirable model is negligible. As a result, it can be difficult to generate high-quality alternatives for a given model. To address this problem, our system uses adaptive kernel density estimation to approximate the distribution of desirable models in the space based on the set of models created by the entire user community. As models are saved by users, their parameter configurations are uploaded to a central server and incorporated into the density estimate. The deployed modeling tools sample from this estimate and other related distributions whenever alternatives are needed. The introduction of large-scale collaboration to 3D modeling is another contribution of our work.

Both the current model and suggested alternatives are visualized on a dynamic map of the design space. Users explore this map with the mouse, and modeling reduces to selecting desirable points from it. To construct the map, our system requires a method for embedding points from the design space into the plane. Unfortunately, traditional dimensionality reduction techniques are of limited utility if the intrinsic dimensionality of the desirable portions of the space is high, or if the full extent of quality regions is not known. We resolve this problem with a visualization algorithm developed specifically for spaces with complex, high-dimensional structure. We obtain a semantically-organized overview of the design space via crowdsourcing, and use this overview in conjunction with the computed density function to dynamically embed regions of the space on demand. These regions are chosen to correspond to the user's indicated areas of interest, guiding navigation towards high-quality models without restricting it to any particular submanifold of the space.

We have developed a modular software system to test these exploratory modeling techniques, and used it to prototype design tools for spaces of trees, humans, and bidirectional reflectance distribution functions (BRDFs). In December of 2007, we released a prototype tree modeling tool to the public. In the twelve months that followed, this software was downloaded by over 17,000 unique users, resulting in the creation of more than 6,000 new models. We report on the results of this deployment, which demonstrate the efficacy of our approach.

2 Related Work

Routine modeling interfaces. Several inventive schemes for routine modeling have been proposed. Funkhouser et al. [2004] describe a system in which the user searches a large database of preexisting models for particular component parts, which are then composited together to form a new object. In the Teddy system [Igarashi et al. 1999], the user sketches the silhouette of a 3D shape with a series of freeform strokes. SmoothSketch [Karpenko and Hughes 2006] interprets these sketched curves as perceptual contours and fills in hidden cusps. FiberMesh [Nealen et al. 2007] extends the sketching metaphor by embedding the defining curves as manipulable handles on the surface of the shape.

Another prominent technique for reconstructing real-world objects is image- and video-based modeling. A recent highlight in image-based modeling is Furukawa and Ponce [2007]; for a video-based approach see VideoTrace [van den Hengel et al. 2007].

Exploratory modeling interfaces. Although the graphics research community has focused primarily on routine 3D modeling tools, several systems display exploratory components. Igarashi and Hughes [2001] describe Chateau, a CAD tool that suggests possible modeling operations based on geometric relationships specified by the user. Tsang et al. [2004] propose a suggestive interface for 3D sketching, in which curves drawn by the user are matched against a database of previously-created geometry to generate suggestions. In the game industry, both Spore's creation tools [Maxis Software 2008] and a number of avatar creators such as Nintendo's Mii Channel [2006] display some exploratory focus and have seen widespread use. In these systems, models are assembled from prebuilt component parts that can be quickly swapped and reconfigured.

In other visual domains, exploratory design has been more thoroughly investigated. Marks et al. [1997] describe a set of interfaces for parameter setting in graphics and animation that present a discrete set of landmark configurations via sampling and allow users to choose between them. Kovar and Gleicher [2001] propose a simplex-based method that allows a small number of PostScript drawings to be transformed via constrained and unconstrained sampling guided by user feedback. Ngan et al. [2006] describe an interface for parameter setting in a space of bidirectional reflectance distribution functions, making extensive use of an original image-based distance metric. Most recently, Shapira et al. [2009] present an exploratory interface for image recoloring based on Gaussian Mixture Models.

Large-scale collaboration. Collaboration technologies are playing an increasingly prominent role in computing. Since its introduction in the early 90s [Goldberg et al. 1992], collaborative filtering, in which the preferences of many users are aggregated to generate personalized recommendations for individuals, has become a Web mainstay [Linden et al. 2003; Adomavicius and Tuzhilin 2005]. More recently, community-scale self-interested activity has been used for image labeling, object recognition, and the collection of semantic information [von Ahn and Dabbish 2004; Su et al. 2007].

Recent work in computer graphics leverages large image collections sourced from the Web. Snavely et al. [2006] present an interface for interactive exploration of image collections taken at particular geographic locations, leading to an engaging virtual tourism experience. Lalonde et al. [2007] use image collections to seamlessly embed new objects in existing photographs. Finally, Hays and Efros [2007] describe a scene completion algorithm using an Internet-scale image database.

Parametric spaces in graphics. High-dimensional parametric spaces are central to computer graphics, dating back to the landmark face modeling work of Parke [1974]. More recently, Blanz and Vetter [1999] construct a space of 3D faces from 200 examples, in which high-level attributes that correspond to meaningful features like age, weight, and gender can be represented as vectors and learned. Allen et al. [2003] extend this approach to entire human models, reconstructing a space of body shapes from 250 scanned human figures using a sophisticated registration algorithm. Spaces for other natural phenomena abound, in domains as diverse as computational botany [Weber and Penn 1995] and appearance modeling [Matusik et al. 2003].

3 Design and Cognition

In order to develop an effective modeling system for any class of users, it is necessary to consider their goals and motivations. To inform our understanding of the spectrum of modeling processes, we draw extensively from the literature on design and human cognition.

One natural way to classify modeling processes is by the degree of completeness in the goal specification [Visser 2006]. At one end of this spectrum lies what is commonly referred to in the literature as *routine* design. In routine modeling processes, the user begins design with a well-specified goal: generally the replication of a precise mental image or plan [Brown and Chandrasekaran 1989]. Routine processes are highly directed: modeling begins with some initial (usually empty) state, and then proceeds sequentially until

Figure 2: *(Left) A screenshot of our exploratory tree modeling tool, with the display window and map window visible. (Right) The optional routine slider interface.*

the goal is reached [Tweedie 1995]. The process is comprised of a series of well-understood pragmatic steps [Navinchandra 1991], each one chosen to decrease the distance between the current state and the goal [Kirsh and Maglio 1994].

At the other end of the spectrum lie *exploratory* modeling processes. Exploratory modeling is open-ended: the user begins the design process with an under-specified goal, and the precise form of the final model is established through experimentation [Brown and Chandrasekaran 1989]. Exploratory modeling processes are highly nonlinear: candidate models are iteratively considered and rejected as the design space is explored. Since the final product is not comprehended in full detail at the onset, progress is made opportunistically and serendipitously [Tweedie 1995]. Many of the actions taken by the exploratory modeler are epistemic rather than pragmatic in nature [Kirsh and Maglio 1994], serving to refine the user's understanding of the space of possible designs rather than moving directly towards the final product.

Although most real-world design is neither purely routine nor purely exploratory, the majority of professional 3D modeling is widely believed to be routine in nature: the domain knowledge professional designers amass over the course of their careers often allows them to clearly specify the desired product and formulate a well-defined procedure for its design [Brown and Chandrasekaran 1989]. In contrast, casual modelers are less likely to take on modeling tasks with rigidly defined goals, and tend to lack operational knowledge of modeling tools and techniques. Novice and casual users, therefore, are particularly likely to benefit from tools that support exploratory modeling. Furthermore, casual and professional designers alike make extensive use of exploratory design processes during brainstorming and ideation, since the potential for innovation and creativity increases as design becomes more exploratory [Gero 1990]. It is for these reasons that we initiate an investigation of exploratory 3D modeling.

Design is fundamentally about choice, and the manner in which meaningfully distinct choices are presented to users is a central determining factor in the success of any design process [Buxton 2007]. Given a candidate design, our exploratory modeling system constructs high-quality alternatives and presents them to the user. It is well established that the development and evaluation of such alternatives contribute to the continual elaboration and reformulation of the underlying design task [Kolodner and Wills 1993]. This feedback loop is widely held to be an essential part of good design.

To visualize the suggested models, we employ a dynamic map of the design space. Maps are amongst the most ubiquitous graphical representations, and present even novice users with a familiar conceptual model for the design process. Moreover, interactive maps promote experiential cognition, making them ideally suited to exploratory modeling processes [MacEachern 1994].

4 Modeling Interface

When our modeling tool is loaded, it connects to a central server and updates its local copy of the database of parameter configurations corresponding to models created by the user community at large. The tool then constructs a probability distribution over the design space using adaptive kernel density estimation (Section 5). This probability density function approximates the distribution of desirable models in the space and is used to generate high-quality alternatives. Whenever alternatives to a particular model are needed, the tool samples from the density function in the local neighborhood of that model (Section 6).

The modeling interface consists of a map window and a display window, presented side by side (see Figure 2, left). Users are always situated at a point in the space, and there is always a fully-formed model in front of them. At any given time, the map window shows a set of icons corresponding to points from the design space. As the user mouses over the map, the display window updates to a visualization of the corresponding model, smoothly interpolating between the visible icon points. Users may pan the map (by clicking and dragging) and zoom it (by double-clicking, or adjusting a slider). Navigating to an unexplored map region causes new icons to appear. The interface of our tree modeling tool is demonstrated in the video accompanying this paper.

The map initializes to a set of icons that are representative of the distribution of desirable models in the space, arranged in a semantically meaningful layout. In a typical modeling session, the user explores this overview and picks a particular model to investigate. As the user zooms towards the model, the surrounding map regions are populated with high-quality alternatives. The degree to which these generated alternatives differ from the current selection is proportional to the level of zoom.

Figure 3: *100 landmark points from the parametric space of trees, embedded into the plane via PCA (left), Isomap (center), and our crowdsourced semantic layout algorithm (right). Observe that both linear and nonlinear dimensionality reduction techniques fail to preserve meaningful semantic structure in the embedding. In contrast, our crowdsourced layout provides a good overview of the space.*

As modeling progresses, the user explores the map, iteratively considering candidate models. Settling on a particular model may take anywhere from a few minutes to more than an hour, depending on the user's design goals and time budget. When the user saves a local copy of the model, its parameter configuration is uploaded to the server.

4.1 Semantic Overview

It is well-established that users depend upon landmarks for map navigation [Tversky et al. 2007]. Our method maintains a subset of representative landmark points from the design space, which are used to govern the map layout. These landmarks are chosen from the database of user-created models via the greedy clustering algorithm of Feder and Greene [1988], which computes a 2-approximation to discrete k-center clustering in $O(N \log k)$ time. We pick k to ensure that the resultant clusters provide good coverage for all points in the database using the cluster quality metrics of Raskutti and Leckie [1999]. As new points are submitted, we determine their best-fit cluster and mark points for which this cluster does not provide a good covering. The set of landmarks is recomputed whenever sufficiently many of these poorly-covered points have been accumulated.

Once an initial set of landmarks has been chosen, we must determine a 2D map location for each one. We have experimented extensively with state-of-the-art algorithms for dimensionality reduction including PCA, MDS, Isomap, and LLE. Although any of these unsupervised embedding techniques can be used by our method, they are generally ill-suited for our purposes. Many interesting parametric design spaces have high intrinsic dimensionality; for instance, our analysis of the database of user-created models in the tree space indicates an intrinsic dimensionality of around 50. Techniques that seek to minimize residual variance or distortion cannot produce good 2D embeddings in such conditions (see Figure 3, left, center). Moreover, although these quality measures are appropriate in many circumstances, they do not always promote embeddings that have strong *semantic* structure. This is particularly the case for high-dimensional spaces with complex morphology, in which good perceptual metrics may be difficult to formulate.

Experiments have shown that people naturally organize space into semantically meaningful regions characterized by distinctive landmarks [Tversky et al. 2007]. To produce such semantically meaningful layouts, we employ a crowdsourcing approach. Whenever the set of landmarks is recomputed, our method allows a few designated users to enter a map-editing mode and adjust the landmark positions directly. This requires no particular expertise or time commitment, since the number of landmarks grows slowly relative to the total size of the database. This crowdsourcing approach yields a high-level semantic overview that is both meaningful to modelers and adaptable as the needs of the user community evolve (see Figure 3, right).

4.2 Exploration and Alternatives

Users explore the map by panning and zooming, iteratively recentering the map window on a particular model interpolated from the set of icons. If the region surrounding this model has been insufficiently explored, the tool creates a set of similar alternative models and places them nearby (see Figure 4). These models are generated by sampling from the density function in the local neighborhood of the centered model. Models produced in this manner are kept only for the duration of the session and not propagated to a central database or other modeling tools. Although these sampled models introduce an element of nondeterminism into the system, once an icon is placed on the map it stays there for the remainder of the session.

When the user pans or zooms to a particular region of the map, we determine the set of landmarks and sampled points with map positions that lie within that region. We iterate through these points, attempting to display each corresponding icon without overlap. Strict preference is given to landmark points during this process, and previously-sampled points are displayed in the order in which they were created. This process ends when the number of visible icons reaches a target threshold, or when the set of potential icons is exhausted. In this latter case, new points are sampled from the density estimate and placed in empty regions via Poisson disk sampling.

4.3 Routine Mechanisms

Although our method is fundamentally geared towards exploratory modeling, we expose two mechanisms that are primarily routine in nature. The first is a set of sliders by which the individual parameters of a model can be adjusted (Figure 2, right). This interface is useful for fine-tuning models that have been selected from the map.

The second mechanism is an interface for setting constraints. In our tools, users can select a set of parameters for which desired values are known, set those values via sliders, and then apply them as constraints to the map as a whole (for instance, a user can easily limit her exploration of the space of human avatars to the subset of models with blue eyes). The technical details of this process are described in Section 6.2.

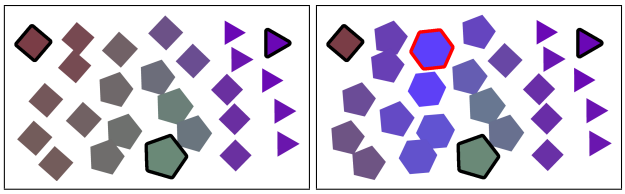

Figure 4: *(Left) Interpolating between landmark points (outlined in black) in a design space of polygons. (Right) Using sampling to introduce local variation. The sampled point is outlined in red.*

Figure 5: *(Left) Typical points sampled from the computed density functions of trees (top) and humans (bottom). (Right) Typical points chosen uniformly at random from these parametric spaces.*

5 Density Estimation

To generate high-quality alternatives to a given parameter configuration, we use the set of models created by the user community to estimate the density of desirable models in the space. This estimate must be computed with some care, since the distribution of quality is highly nonuniform in many interesting parametric spaces (see Figure 5). Let $\{\mathbf{x}_i\}$ be a set of N database points from an n-dimensional design space $\mathbb{D}$, assumed to be drawn from an unknown probability density function $f(\mathbf{x})$. Since we wish to develop a general method that is applicable to a wide variety of distinct spaces, we cannot make any assumptions about the intrinsic structure of $f(\mathbf{x})$: therefore, we are faced with a *nonparametric* density estimation problem.

Multivariate nonparametric density estimation is a well studied problem with applications in numerous fields. One of the most popular techniques is the sample-point kernel estimator, first introduced by Parzen [1962], in which an approximation $\hat{f}(\mathbf{x})$ of $f(\mathbf{x})$ is recovered by centering a smooth kernel $K_i(\mathbf{x})$ at each of the points in the training set:

$$\hat{f}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K_i(\mathbf{x}).$$

Under reasonable assumptions, it can be shown that kernel methods converge at least as quickly as any other nonparametric estimation technique. We employ this approximation with Gaussian kernels:

$$K_i(\mathbf{x}) = \mathcal{G}(\mathbf{x}; \mathbf{x}_i, \mathbf{\Sigma}_i) = \frac{1}{(2\pi)^{n/2}\,|\mathbf{\Sigma}_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{x}_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mathbf{x}_i)\right],$$

where $\mathbf{\Sigma}_i$ is a bandwidth matrix chosen as described in Section 5.1.

5.1 Bandwidth Estimation

For kernel density estimation to perform well in practice, one must choose the bandwidth matrices—which control the size and shape of the individual kernels—with some care. For $n \leq 3$, the optimal structure of $\mathbf{\Sigma}_i$ can be determined analytically, but these results do not generalize to higher dimensions [Scott and Sain 2004]. Likewise, iterative cross-validation techniques that attempt to learn the optimal kernel structure can be prohibitively expensive in higher dimensions.

Approaches based on the sample covariance of the k^{th} nearest neighbors of a given point have been shown to perform well for $n > 4$, but may suffer from sharp discontinuities as the set of nearest neighbors does not vary smoothly across the space. Therefore, we employ a fixed-mean version of the distance-weighted empirical covariance matrix described by Bengio and Vincent [2004]. This approach is predicated on a "soft" weighted-neighborhood notion of locality, thereby ensuring a continuous density estimate. For a kernel centered at a point $\mathbf{x}$, the bandwidth matrix has entries

$$\mathbf{\Sigma}_{s,t} = \sum_{i=1}^{N} \omega_i \left[(\mathbf{x}_i)_s - (\mathbf{x})_s\right]\left[(\mathbf{x}_i)_t - (\mathbf{x})_t\right],$$

where ω_i are normalized weights. We choose

$$\omega_i = \frac{\mathcal{G}(\mathbf{x}_i; \mathbf{x}, \alpha\|\mathbf{x}-\mathbf{x}_{d(k)}\|^2\mathbf{I})}{\sum_{j=1}^{N}\mathcal{G}(\mathbf{x}_j; \mathbf{x}, \alpha\|\mathbf{x}-\mathbf{x}_{d(k)}\|^2\mathbf{I})},$$

where α is a smoothing parameter and $\mathbf{x}_{d(k)}$ is the k^{th} nearest neighbor of $\mathbf{x}$ amongst the $\{\mathbf{x}_i\}$; in our implementation, we take $k = n$. In this manner, the width of each weighting kernel is proportional to the distance from its center to its k^{th} nearest neighbor, which allows each bandwidth matrix to smoothly adapt to the local shape and scale of the underlying distribution. If the database points in some region of the space lie on or near a low-dimensional manifold, the shape of a kernel centered in that region will coincide with the principal directions of that manifold.

5.2 Bandwidth Shrinkage

Unfortunately, all bandwidth estimators based on empirical covariances exhibit serious defects when N is not much larger than n. For instance, unless $N \gg n$, $\mathbf{\Sigma}$ will be ill-conditioned. Worse still, when $N < n$, $\mathbf{\Sigma}$ loses full rank, becomes singular, and is no longer positive definite. These limitations are highly relevant, since one of the key properties of our method is that it can function even with few initial database points.

To overcome these deficiencies, we modify the shrinkage estimator of Schäfer and Strimmer [2005] to apply to weighted covariance matrices, and employ it in all of our bandwidth computations. This involves computing a shrinkage target matrix $\mathbf{\Phi}$ and optimal intensity λ, and then substituting the shrinkage estimator

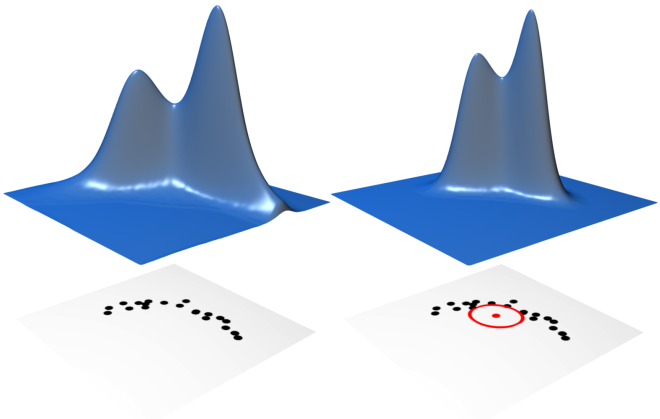

Figure 6: *(Left) The probability distribution $\hat{f}(\mathbf{x})$ for a collection of 2D points, computed with adaptive kernel density estimation. (Right) The local distribution $\frac{1}{\varphi}\mathcal{G}(\mathbf{x};\mathbf{x}_0,\boldsymbol{\Sigma}_0)\hat{f}(\mathbf{x})$, with $\mathbf{x}_0$ and an isocontour of $\mathcal{G}(\mathbf{x};\mathbf{x}_0,\boldsymbol{\Sigma}_0)$ shown in red.*

$\boldsymbol{\Sigma}^\star = \lambda\boldsymbol{\Phi} + (1-\lambda)\boldsymbol{\Sigma}$ everywhere for $\boldsymbol{\Sigma}$. By employing this estimator, we are able to compute accurate bandwidth matrices with favorable numerical properties for arbitrary combinations of N and n. The relevant mathematical details are presented in Appendix A.

6 Sampling

Once the density estimate has been constructed, we sample from it and other related distributions in order to generate alternatives to the current model. Since $\hat{f}(\mathbf{x})$ is a uniform mixture of Gaussians, we can sample from it by choosing an index $i \in [1, 2, \ldots, N]$ uniformly at random and then drawing a sample from the corresponding $\mathcal{G}(\mathbf{x};\mathbf{x}_i,\boldsymbol{\Sigma}_i)$.

In parametric design, each parameter is typically restricted to lie in a compact range of valid values. For example, a parameter representing the joint angle of a human elbow must be appropriately constrained between 0 and 180 degrees. Therefore, we employ the multivariate truncated Gaussian simulator of Robert [1995] in all our sampling schemes. This simulator efficiently produces draws from truncated multivariate Gaussian distributions via a clever rejection sampling technique.

6.1 Local Sampling

To sample from $\hat{f}(\mathbf{x})$ in the local neighborhood of a particular model $\mathbf{x}_0$, we draw samples from probability distributions of the form

$$\frac{1}{\varphi}\,\mathcal{G}\big(\mathbf{x};\mathbf{x}_0,\boldsymbol{\Sigma}_0\big)\cdot\hat{f}(\mathbf{x}), \qquad (\star)$$

where φ is a normalizing constant and $\boldsymbol{\Sigma}_0$ is chosen as in Section 5.1 with k an order of magnitude larger than n. Intuitively, we bias the sample to be close to $\mathbf{x}_0$ while still generally adhering to the probability distribution (see Figure 6). By taking the bandwidth smoothing parameter α to be inversely proportional to the designer's confidence in $\mathbf{x}_0$, we can control the degree of variation produced by the sampling: in our implementation, α is chosen proportional to the current zoom level of the map. To perform this sampling efficiently, we substitute for $\hat{f}(\mathbf{x})$ in $(\star)$ and rearrange terms to obtain

$$\frac{1}{\varphi N}\sum_{i=1}^{N}\Big(\mathcal{G}\big(\mathbf{x};\mathbf{x}_0,\boldsymbol{\Sigma}_0\big)\mathcal{G}\big(\mathbf{x};\mathbf{x}_i,\boldsymbol{\Sigma}_i\big)\Big) \;=$$
$$\frac{1}{\varphi N}\sum_{i=1}^{N}\Big(\mathcal{G}\big(\mathbf{x}_0;\mathbf{x}_i,\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_i\big)\mathcal{G}\big(\mathbf{x};\mathbf{x}'_i,\boldsymbol{\Sigma}'_i\big)\Big),$$

where

$$\begin{aligned}\boldsymbol{\Sigma}'_i &= \big(\boldsymbol{\Sigma}_0^{-1}+\boldsymbol{\Sigma}_i^{-1}\big)^{-1} \qquad \text{and}\\ \mathbf{x}'_i &= \boldsymbol{\Sigma}'_i\big(\boldsymbol{\Sigma}_0^{-1}\mathbf{x}_0+\boldsymbol{\Sigma}_i^{-1}\mathbf{x}_i\big).\end{aligned}$$

Each of the summands is now a scaled Gaussian and the contribution of summand i to the probability mass is proportional to the constant $\mathcal{G}\left(\mathbf{x}_0;\mathbf{x}_i,\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_i\right)$. Thus we can sample from $(\star)$ by choosing an index $1 \le i \le N$ with probability

$$\frac{\mathcal{G}\left(\mathbf{x}_0;\mathbf{x}_i,\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_i\right)}{\sum_i \mathcal{G}\left(\mathbf{x}_0;\mathbf{x}_i,\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_i\right)}$$

and then drawing a sample from $\mathcal{G}\left(\mathbf{x};\mathbf{x}'_i,\boldsymbol{\Sigma}'_i\right)$. Computing these probabilities with floating point arithmetic requires some care: see Appendix B for details.

6.2 Constrained Sampling

For modeling tasks in which desired values for p of the n parameters are known *a priori*, we may, with an appropriate reordering, consider the conditional probability distribution $\hat{f}(\mathbf{x}_1 \mid \mathbf{x}_2)$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ are subvectors of $\mathbf{x}$ of dimensionality $n-p$ and p representing the varying and fixed parameters, respectively. Given matching decompositions

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix}, \quad \mathbf{x}_i = \begin{bmatrix}\mathbf{x}_{i_1}\\ \mathbf{x}_{i_2}\end{bmatrix}, \quad \text{and} \quad \boldsymbol{\Sigma}_i = \begin{bmatrix}\boldsymbol{\Sigma}_{i_{11}} & \boldsymbol{\Sigma}_{i_{12}}\\ \boldsymbol{\Sigma}_{i_{21}} & \boldsymbol{\Sigma}_{i_{22}}\end{bmatrix},$$

we observe that

$$\hat{f}(\mathbf{x}_1 \mid \mathbf{x}_2) = \frac{1}{N}\sum_{i=1}^{N} K_i(\mathbf{x}_1 \mid \mathbf{x}_2) = \frac{1}{N}\sum_{i=1}^{N} G(\mathbf{x}_1;\mathbf{x}_{i_{1|2}},\boldsymbol{\Sigma}_{i_{1|2}}),$$

where

$$\begin{aligned}\mathbf{x}_{i_{1|2}} &= \mathbf{x}_{i_1}+\boldsymbol{\Sigma}_{i_{12}}\boldsymbol{\Sigma}_{i_{22}}^{-1}(\mathbf{x}_2-\mathbf{x}_{i_2}) \qquad \text{and}\\ \boldsymbol{\Sigma}_{i_{1|2}} &= \boldsymbol{\Sigma}_{i_{11}}-\boldsymbol{\Sigma}_{i_{12}}\boldsymbol{\Sigma}_{i_{22}}^{-1}\boldsymbol{\Sigma}_{i_{21}}.\end{aligned}$$

Since the conditional probability of a multivariate Gaussian distribution is another multivariate Gaussian distribution of lower dimensionality, our sampling framework adapts seamlessly to the presence of such constraints.

7 Parametric Spaces

To assess the applicability of our method, we investigated three disparate design spaces: a modified version of the tree space developed by Weber and Penn [1995] ($n = 91$), an extension of the space of human body shapes of Allen et al. [2003] ($n = 130$), and the anisotropic Phong BRDF space of Ashikhmin and Shirley [2000] ($n = 9$). In each case, we seeded the space with a number of real-world examples corresponding to observed trees ($N = 19$), scanned human shapes ($N = 124$), and measured reflectance functions ($N = 42$), respectively.

We focused the bulk of our efforts on trees for a number of reasons: they are familiar to casual users and expert modelers alike, have complex structure, are ubiquitous in multi-user virtual environments, and are difficult for novices to rapidly produce in existing modeling tools. We made relatively minor modifications to Weber and Penn's original space, mostly altering parameters that triggered conditional modes of execution and removing parameters that were visually unimportant or linearly dependent. A thorough evaluation of our tree modeling tool is reported in Section 8.

Realistic human models are among the most sought-after assets in modern graphical applications and are known to be difficult and time-consuming to produce using traditional modeling tools. The

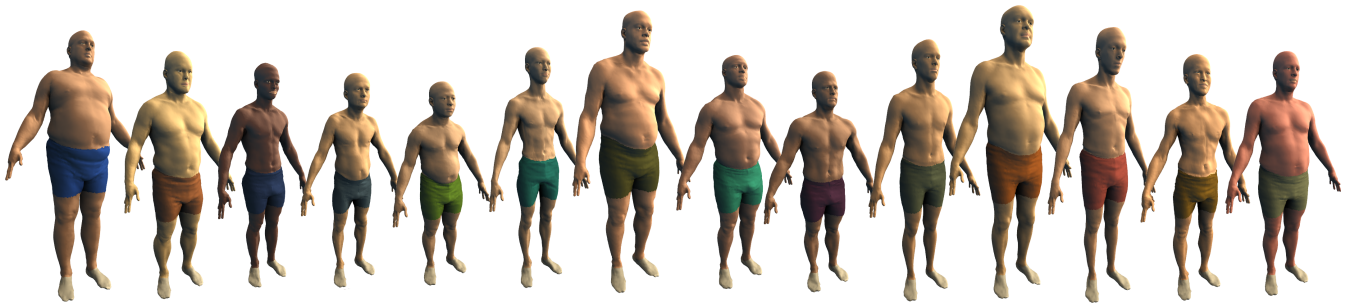

Figure 7: *Typical samples from the computed distribution of human body shapes (zoom in for detail).*

space of human body shapes developed by Allen et al., therefore, was another natural choice. We made slight alterations to the original space, adding a few extra parameters related to skin color and appearance in order to make the produced models more realistic (see Figure 7). Exploratory modeling is especially valuable in this space, since Allen's original parameters defy semantic interpretation and are difficult to manipulate directly.

We also tested our method on the standard anisotropic BRDF model of Ashikhmin and Shirley: a low-dimensional parametric space specifically constructed to contain only good parameter configurations. Somewhat surprisingly, density estimation is valuable even in this space. In particular, materials with unlikely diffuse/specular combinations are properly assigned low probabilities by the estimate and avoided during modeling (see Figure 8).

8 Implementation and Evaluation

We have developed a modular software toolkit for creating exploratory modeling tools, consisting of a client modeling application written in C++ and a server framework written in Ruby with an SQL backend. In December of 2007, we released a prototype tree modeling tool to the public. In the twelve months that followed, a burgeoning community of users expanded the initial database of 19 seed trees (Figure 10) to more than 6,200 new models. Several representative examples of these unique and diverse trees are presented in Figure 11.

To analyze the effectiveness of the tool, all user interactions were anonymously logged and periodically uploaded to a central server. We report statistical analyses of the collected user data and responses to a short user survey.

8.1 Log Analysis

We analyzed 6,901 user logs, detailing the creation of 4,865 tree models. The average creation time per tree was 15.1 minutes, while 63.2% of trees were created in less than 10 minutes, and the longest 1% of modeling sessions took between 1 and 3 hours. These results demonstrate that the tool enables users to create high-quality models in relatively short periods of time: it would be difficult even for professional modelers to create many of the trees shown in Figure 11 so quickly using existing modeling software.

We also investigated the relative impact of the exploratory and routine mechanisms exposed by the tool. On average, users spent only 22.7% of each modeling session adjusting parameters via sliders or applying constraints.

8.2 User Survey

After the number of downloads exceeded 5000, we conducted a short survey of registered users. These users were presented with a series of statements about the software and asked to evaluate them on the standard 5-point Likert scale. Completed surveys from 72 respondents were received. Only 15% of the respondents rated their knowledge of 3D modeling as "fluent" or "expert". The survey results are summarized in Figure 9.

The responses seem to largely validate our method, with a majority of users indicating that the prototype allowed them to create compelling models and was less cumbersome than existing tools. Especially encouraging was the degree to which users found that the software allowed them to express their personal sense of aesthetics and provided an engaging and even inspiring experience.

9 Discussion and Future Work

Further investigation of exploratory 3D modeling techniques has great potential to produce more accessible tools for casual and novice modelers, and support creativity and innovation in novices and experts alike. While our work takes a step in this direction, the presented method is not without limitations. It is, for instance, fundamentally ill-suited for routine design tasks, such as the replication of a particular, precise model specification. It is also appropriate only for domains with good parametric representations. Developing parametric spaces for new visual domains (in the spirit of [Weber and Penn 1995; Allen et al. 2003]) remains an interesting challenge. More generally, developing exploratory 3D modeling tools that do not rely on parametric spaces is a key avenue for future research. Particularly promising is the potential for integrating exploratory modeling techniques with traditional geometry processing methods and grammar-based procedural modeling tools.

There is also much more to be learned about the cognitive aspects of 3D modeling. In particular, there is a pronounced need for ethnographic research in this area: we are aware of no formal comprehensive studies of the behavior and motivations of 3D modelers. We believe that a more complete understanding of the spectrum of design processes employed by both casual and professional modelers will eventually lead to significantly improved tools for 3D content creation.

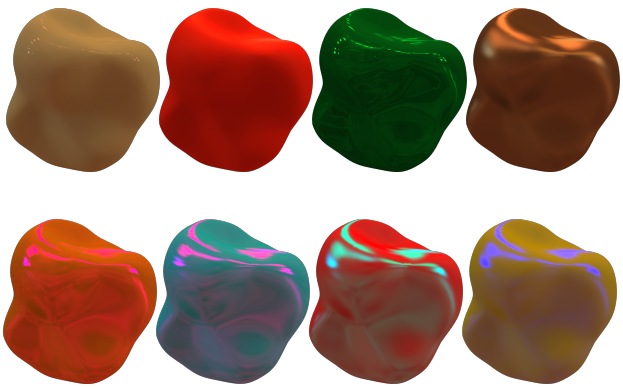

Figure 8: *Even in the nine-dimensional space of BRDFs, sampling from the computed density function (top) is more likely to produce visually-appealing points than random sampling (bottom).*

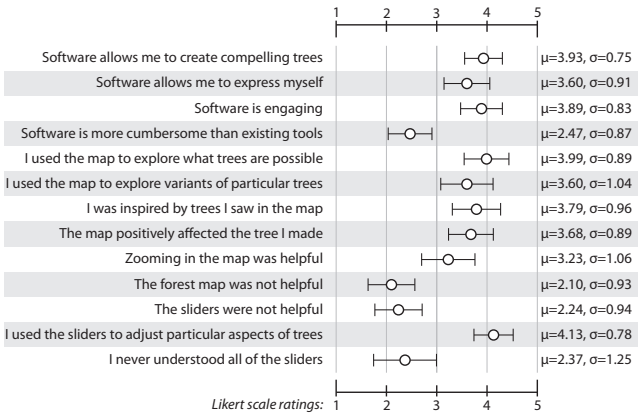

Figure 9: *Survey results from 72 users of the tree modeling tool. 1 = 'strongly disagree', 3 = 'neutral', and 5 = 'strongly agree'. Error bars indicate one standard deviation.*

Acknowledgements

We thank Brett Allen and Zoran Popovic for providing the human body shape data, Ewen Cheslack-Postava for help with BRDF rendering, and Chris Platz for help with art assets. This work was supported by NSF grants SES-0835601 and CCF-0641402, an NVIDIA Fellowship, and an Intel Foundation Graduate Fellowship.

A Computing the Shrinkage Estimate

Given a bandwidth matrix $\mathbf{\Sigma}$ as defined in Section 5.1, we compute a modified version of the "diagonal, unequal variance" shrinkage estimator from Schäfer and Strimmer [2005]. The target matrix $\mathbf{\Phi}$ and optimal shrinkage intensity λ are given by

$$\mathbf{\Phi}_{s,t} = \begin{cases} \mathbf{\Sigma}_{s,s} & \text{if } s = t \\ 0 & \text{if } s \neq t \end{cases} \quad \text{and} \quad \lambda = \frac{\sum_{s \neq t} \widehat{Var}\left(\mathbf{\Sigma}_{s,t}\right)}{\sum_{s \neq t} \mathbf{\Sigma}_{s,t}^2}.$$

Here $\widehat{Var}\left(\mathbf{\Sigma}_{s,t}\right)$ is an estimate of the variance of the *individual* elements of $\mathbf{\Sigma}$. Setting $w_{i_{s,t}} = [(\mathbf{x}_i)_s - (\mathbf{x})_s][(\mathbf{x}_i)_t - (\mathbf{x})_t]$, $\bar{w}_{s,t} = \sum_{i=1}^N \omega_i w_{i_{s,t}}$, and $\bar{\omega} = \frac{1}{N}\sum_{i=1}^N \omega_i$, it is clear that each individual $\mathbf{\Sigma}_{s,t}$ is a weighted mean of the $w_{i_{s,t}}$. With this observation, we may employ the variance estimate from Gatz and Smith [1995] to the elements of our distance-weighted covariance matrices:

$$\begin{aligned} \widehat{Var}(\mathbf{\Sigma}_{s,t}) = \frac{N}{N-1} \Bigg[& \sum_{i=1}^N \left(\omega_i w_{i_{s,t}} - \bar{\omega}\bar{w}_{s,t}\right)^2 \\ & - 2\bar{w}_{s,t} \sum_{i=1}^N \left(\omega_i - \bar{\omega}\right)\left(\omega_i w_{i_{s,t}} - \bar{\omega}\bar{w}_{s,t}\right) \\ & + \bar{w}_{s,t}^2 \sum_{i=1}^N \left(\omega_i - \bar{\omega}\right)^2 \Bigg]. \end{aligned}$$

Note that, in practice, λ must be clamped to the range $[0, 1]$.

This estimator possesses a number of favorable properties. It is roughly as efficient to compute as $\mathbf{\Sigma}$ and has guaranteed minimum mean-squared error. It is also fully automatic and nonparametric, requiring no parameter tuning and relying on no assumptions about the structure of the underlying distribution except the existence of the first moment. Furthermore, the resultant matrix is *always* well-conditioned and positive definite, and can therefore be efficiently inverted. Moreover, for spaces where n is so large as to make even a single realtime matrix inversion intractable, the estimator admits a principled simplification. By forcing $\lambda = 1$, we obtain a set of kernels in which all of the parameters are assumed to be independent and the resultant bandwidth matrices are all diagonal, making inversion (and sampling) trivial.

B Log-Likelihood Kernels

To compute the discrete probability distribution that is central to the local sampling of Section 6.1 without under- or over-flow, we use a standard technique from machine learning folklore and apply a log transformation to each kernel. Observe that

$$\begin{aligned} \mathcal{L}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{\Sigma}) = \log\left(\mathcal{G}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{\Sigma})\right) = \\ -\frac{1}{2}\left(n\log(2\pi) + \log|\mathbf{\Sigma}| + (\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right), \end{aligned}$$

where

$$\log|\mathbf{\Sigma}| = \log|\boldsymbol{L}||\boldsymbol{L}^*| = 2\log\prod_{s=1}^n \boldsymbol{L}_{s,s} = 2\sum_{s=1}^n \log \boldsymbol{L}_{s,s}.$$

Each $\mathcal{L}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{\Sigma})$ can be computed even for large n, but naïvely inverting the log transformation may still result in underflow. Instead, we set $\gamma = \exp[-\mathcal{L}_m]$, where $\mathcal{L}_m = \max_j\left[\mathcal{L}(\mathbf{x}_0; \mathbf{x}_j, \mathbf{\Sigma}_0 + \mathbf{\Sigma}_j)\right]$, and observe that

$$\frac{\mathcal{G}_i}{\sum_j \mathcal{G}_j} = \frac{\gamma}{\gamma}\frac{\exp[\mathcal{L}_i]}{\sum_j \exp[\mathcal{L}_j]} = \frac{\exp[\mathcal{L}_i - \mathcal{L}_m]}{\sum_j \exp[\mathcal{L}_j - \mathcal{L}_m]},$$

where $\mathcal{G}_i = \mathcal{G}\left(\mathbf{x}_0; \mathbf{x}_i, \mathbf{\Sigma}_0 + \mathbf{\Sigma}_i\right)$ and $\mathcal{L}_i = \log \mathcal{G}_i$. This rescaling forces the unnormalized weight of the most-likely index in the resultant distribution to one, yielding a stable sampling even when the relative weights of the remaining indices all underflow themselves.

References

ADOMAVICIUS, G., AND TUZHILIN, A. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Trans. on Knowledge and Data Engineering 17*, 6, 734–749.

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: reconstruction and parameterization from range scans. In *Proc. SIGGRAPH*, ACM Press, 587–594.

ASHIKHMIN, M., AND SHIRLEY, P. 2000. An anisotropic Phong BRDF model. *Journal of Graphics Tools 5*, 25–32.

BENGIO, Y., AND VINCENT, P. 2004. Locally weighted full covariance gaussian density estimation. Cirano working papers, CIRANO, May.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*, ACM Press, 187–194.

BROWN, D., AND CHANDRASEKARAN, B. 1989. *Design problem solving: knowledge structures and control strategies*. Morgan Kaufmann, San Francisco, CA, USA.

BUXTON, B. 2007. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, April.

FEDER, T., AND GREENE, D. 1988. Optimal algorithms for approximate clustering. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, ACM, New York, NY, USA, 434–444.

FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. In *Proc. SIGGRAPH*, ACM, 652–663.

Figure 10: *The nineteen trees that were used as initial landmarks for the parametric tree space.*

FURUKAWA, Y., AND PONCE, J. 2007. Accurate, dense, and robust multi-view stereopsis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

GATZ, D. F., AND SMITH, L. 1995. The standard error of a weighted mean concentration. *Atmospheric Environment 29*, 11, 1185–1193.

GERO, J. 1990. Design prototypes: A knowledge representation schema for design. *AI Magazine 11*, 4.

GOLDBERG, D., NICHOLS, D., OKI, B. M., AND TERRY, D. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM 35*, 12, 61–70.

HAYS, J., AND EFROS, A. A. 2007. Scene completion using millions of photographs. In *Proc. SIGGRAPH*, ACM Press, 4.

HOSCHEK, J., AND DANKWORT, W. 1994. *Parametric and Variational Design*. B. G. Teubner.

IGARASHI, T., AND HUGHES, J. F. 2001. A suggestive interface for 3D drawing. In *Proc. UIST*, ACM Press, 173–181.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: a sketching interface for 3D freeform design. In *Proc. SIGGRAPH*, ACM Press, 409–416.

KARPENKO, O. A., AND HUGHES, J. F. 2006. Smoothsketch: 3D free-form shapes from complex sketches. In *Proc. SIGGRAPH*, ACM Press, 589–598.

KIRSH, D., AND MAGLIO, P. 1994. On distinguishing epistemic from pragmatic action. *Cognitive Science 18*, 4, 513–549.

KOLODNER, J. L., AND WILLS, L. M. 1993. Case-based creative design. In *In AAAI Spring Symposium on AI and Creativity*, 50–57.

KOVAR, L., AND GLEICHER, M. 2001. Simplicial families of drawings. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, 163–172.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2007. Photo clip art. In *Proc. SIGGRAPH*, ACM, New York, NY, USA, 3–13.

LINDEN, G., SMITH, B., AND YORK, J. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing 7*, 1, 76–80.

MACEACHERN, A. 1994. *Visualization in modern cartography: setting the agenda*. Pergamon.

MARINO, P. 2004. *3D Game-Based Filmmaking: The Art of Machinima*. Paraglyph.

MARKS, J., ANDALMAN, B., BEARDSLEY, P. A., FREEMAN, W., GIBSON, S., HODGINS, J., KANG, T., MIRTICH, B., PFISTER, H., RUML, W., RYALL, K., SEIMS, J., AND SHIEBER, S. 1997. Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proc. SIGGRAPH*, ACM Press, 389–400.

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. A data-driven reflectance model. In *Proc. SIGGRAPH*, ACM Press, 759–769.

MAXIS SOFTWARE. 2008. *Spore*. `http://www.spore.com`.

MILLER, G. 2007. The promise of parallel universes. *Science 317*, 1341–1343.

NAVINCHANDRA, D. 1991. *Exploration and innovation in design: towards a computational model*. Springer-Verlag, New York, NY, USA.

NEALEN, A., IGARASHI, T., SORKINE, O., AND ALEXA, M. 2007. Fibermesh: designing freeform surfaces with 3D curves. In *Proc. SIGGRAPH*, ACM Press, 41–49.

NEWMAN, G. 2006. *Garry's Mod software, version 10*. `http://www.garrysmod.com`.

NGAN, A., DURAND, F., AND MATUSIK, W. 2006. Image-driven navigation of analytical BRDF models. In *Proceedings of the Eurographics Symposium on Rendering*, 389–400.

NINTENDO COMPANY LTD. 2006. *Mii Channel*. `http://us.wii.com/`.

PARKE, F. I. 1974. A parametric model for human faces. Tech. rep., University of Utah.

PARZEN, E. 1962. On estimation of a probability density function and mode. *Annals of mathematical statistics 33*, 1065–1076.

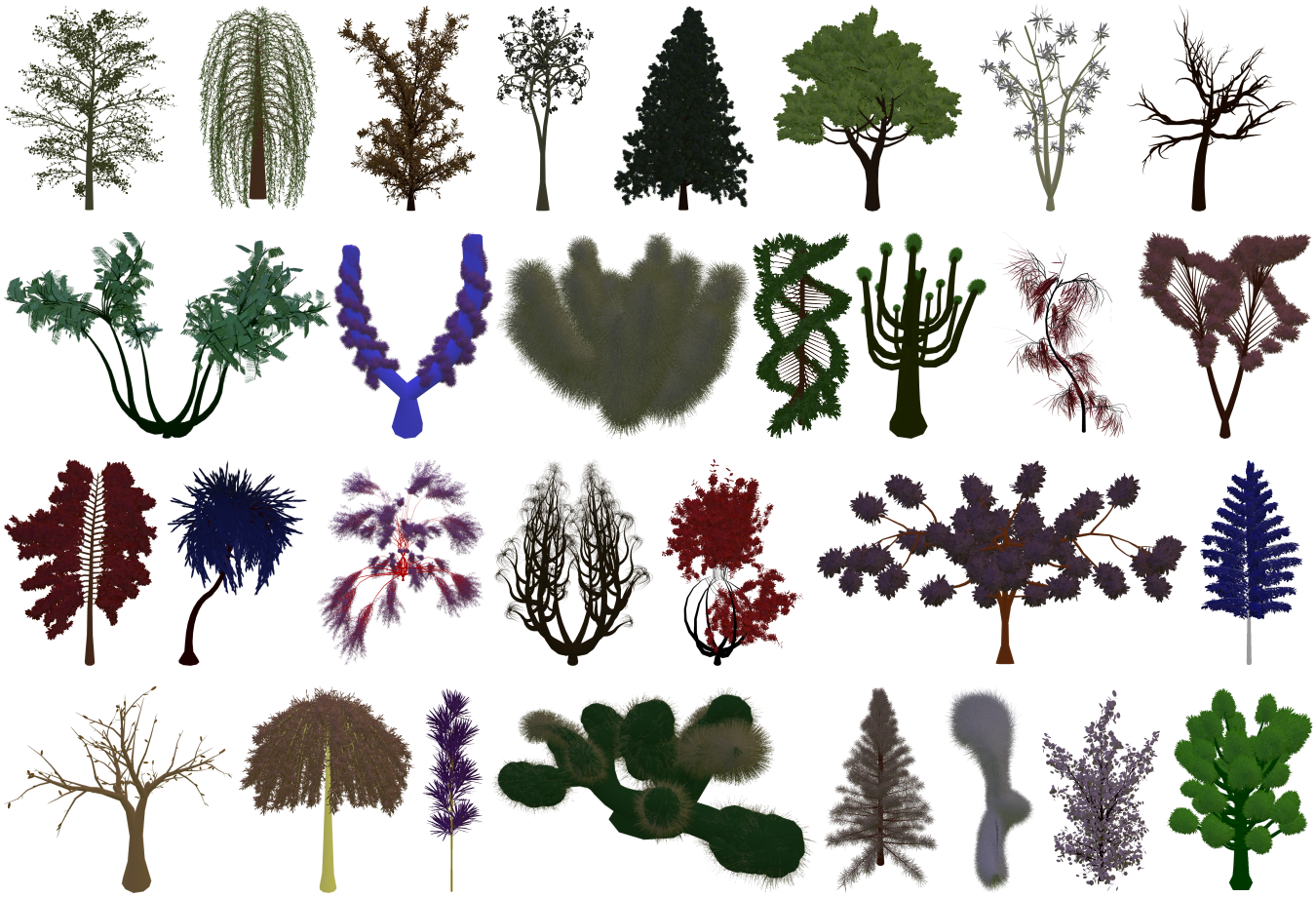

Figure 11: *Trees created by users of our prototype exploratory modeling software.*

RASKUTTI, B., AND LECKIE, C. 1999. An evaluation of criteria for measuring the quality of clusters. In *Proc. International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, 905–910.

ROBERT, C. P. 1995. Simulation of truncated normal variables. *Statistics and Computing 5*, 121–125.

SCHÄFER, J., AND STRIMMER, K. 2005. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology 4*, 1, 1–32.

SCOTT, D. W., AND SAIN, S. R. 2004. Multi-dimensional density estimation. *Data Mining and Computational Statistics 23*.

SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2009. Image appearance exploration by model-based navigation. *Computer Graphics Forum 28*, 2, 629–638.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3D. In *Proc. SIGGRAPH*, ACM Press, 835–846.

SU, Q., PAVLOV, D., CHOW, J.-H., AND BAKER, W. C. 2007. Internet-scale collection of human-reviewed data. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, ACM, New York, NY, USA, 231–240.

TSANG, S., BALAKRISHNAN, R., SINGH, K., AND RANJAN, A. 2004. A suggestive interface for image guided 3d sketching. In *Proc. of CHI*, ACM Press, 591–598.

TVERSKY, B., AGRAWALA, M., HEISER, J., LEE, P., HANRAHAN, P., PHAN, D., STOLTE, C., AND DANIEL, M.-P. 2007. Cognitive design principles for generating visualizations. In *Applied spatial cognition: from research to cognitive technology*, 55–73.

TWEEDIE, L. 1995. Interactive visualisation artifacts: how can abstractions inform design? In *HCI '95: Proceedings of the HCI'95 conference on People and Computers*, Cambridge University Press, 247–265.

VAN DEN HENGEL, A., DICK, A., THORMÄHLEN, T., WARD, B., AND TORR, P. H. S. 2007. Videotrace: rapid interactive scene modelling from video. In *Proc. SIGGRAPH*, ACM Press, 86–91.

VISSER, W. 2006. *The Cognitive Artifacts of Designing*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

VON AHN, L., AND DABBISH, L. 2004. Labeling images with a computer game. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, 319–326.

WEBER, J., AND PENN, J. 1995. Creation and rendering of realistic trees. In *Proc. SIGGRAPH*, ACM Press, 119–128.