Data-Driven Suggestions for Creativity Support in 3D Modeling

Siddhartha Chaudhuri Vladlen Koltun

Stanford University *

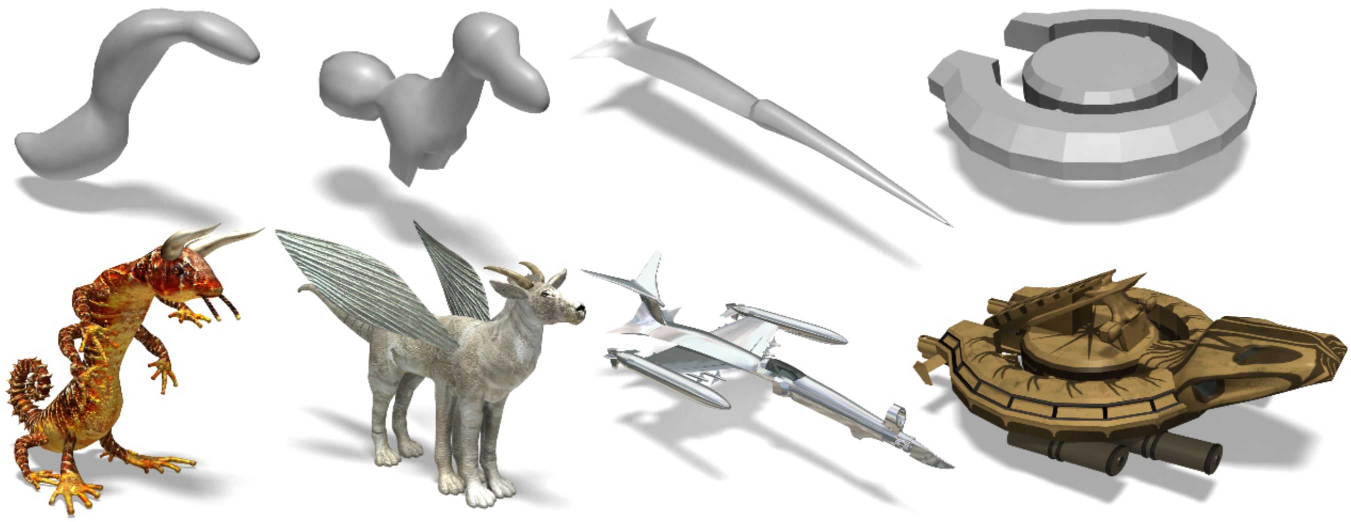

Figure 1: *3D models created using data-driven suggestions, starting from simple shapes for which suggestions were generated.*

Abstract

We introduce data-driven suggestions for 3D modeling. Data-driven suggestions support open-ended stages in the 3D modeling process, when the appearance of the desired model is ill-defined and the artist can benefit from customized examples that stimulate creativity. Our approach computes and presents components that can be added to the artist's current shape. We describe shape retrieval and shape correspondence techniques that support the generation of data-driven suggestions, and report preliminary experiments with a tool for creative prototyping of 3D models.

CR Categories: I.3.5 [Computing Methodologies]: Computer Graphics—Computational Geometry and Object Modeling;

Keywords: data-driven 3D modeling, creativity support, shape analysis, interaction

1 Introduction

The skilled production of a preconceived result is the hallmark of craft. "The craftsman knows what he wants to make before he makes it" [Collingwood 1938]. Computer graphics research has made great strides in supporting the craft of 3D modeling. With appropriate tools, even novices can produce 3D models that match a predefined appearance [Funkhouser et al. 2004].

Yet 3D modeling is also an art. Art is an open-ended creative process, subject to unexpected changes of direction or goal [Collingwood 1938]. In this paper, we present a technique that supports creative discovery in 3D modeling. Our technique is complementary to existing approaches to 3D modeling and falls within the domain of creativity support [Shneiderman 2007].

Our approach to creativity support is motivated by cognitive theories of creativity. Creative cognition can be viewed as the interplay between the generation of solution components and their integration [Finke et al. 1992]. A key role for creativity support tools is to enable easy exploration of possible solution components [Shneiderman et al. 2006].

To support creative discovery in 3D modeling, we introduce data-driven suggestions. Data-driven suggestions present the modeler with possible additional components for the current shape. The suggestions are shown, when requested, in order to stimulate the creative process. The modeler is free to integrate any of the components or disregard the suggestions. The suggestions are automatically generated by comparing the current shape with a large library of existing models. The suggestion generation process is entirely unsupervised – the modeler need not specify the suggested components in any way. This maximizes the potential for unexpected, yet valuable suggestions.

To suggest components for a given shape, our method retrieves models from the library that can be used as sources of suggestions. Such models have similar gross structure to the query but may differ in local detail and contain novel components. The search results are diversified to facilitate rapid exploration of alternatives.

For each retrieved library model, our technique identifies components in the model that are not present in the user's current shape. This is accomplished by computing a correspondence score for each sample point on the library model, indicating the likelihood of it having a counterpart on the user's shape. Segments of the library model that have low average scores are presented as suggestions. High-dimensional nearest-neighbor search techniques are employed to make the computation of a large number of correspondence scores tractable.

To evaluate the utility of data-driven suggestions for creativity support in 3D modeling, we have developed a tool for creative pro-

*e-mail:{sidch,vladlen}@cs.stanford.edu

totyping during the formative stages of 3D design. In these early stages, the final design is under-specified, and the artist can benefit from rapid exploration of a variety of possible directions [Cross 2001]. Our tool suggests additional components for a rough initial shape given by the user. The artist can interactively try on any combination of suggestions, discard any of them, and request additional ones. Preliminary experiments indicate that data-driven suggestions fit into the workflows of 3D artists.

1.1 Data-Driven Creativity

Creativity is generally defined as the process of bringing something novel and valuable into being [Sternberg 1999]. The novelty criterion should not be misconstrued to require that all components of a creative product are new. Modern accounts of creative cognition stress the central role of combining pre-existing components [Finke et al. 1992]. A significant stream of research suggests that creativity is entirely regulated by ordinary cognitive processes, and greater creativity stems primarily from greater access to solution components, acquired through domain expertise [Boden 1990; Weisberg 2006].

This highlights a key opportunity for computational creativity support. Computational tools can make available a large variety of appropriate solution components to the user, in effect externalizing domain expertise that the user may lack. Experiments suggest that direct incorporation of examples into a creative product does not necessarily decrease the creativity of the output, since people tend to combine existing components in novel ways and build upon them [Marsh et al. 1996]. Even exceptionally creative products, such as Coleridge's poetry, have incorporated pre-existing material [Boden 1990], and creativity enhancement methodologies are often designed as structured search for solution components [Nickerson 1999].

Shneiderman et al. [2006] identify the "freer exploration of alternatives" as a key characteristic of creativity support tools, and advocate for tools that prioritize "easy exploration, rapid experimentation, and fortuitous combinations that lead to innovations." Computational creativity support research has thus sought techniques for generating customized examples that allow the user to explore the space of possibilities and alternative courses of action. This is particularly important during the early, formative stages of design, when mock-ups, prototypes, and other external representations play a pivotal role [Cross 2001].

How can computational tools provide customized examples for a user's creative project? One possibility is to modify the user's current design according to pre-defined rules and present several such alternative designs as suggestions [Terry and Mynatt 2002; Terry et al. 2004]. However, suggestions generated in this way are necessarily limited to components that are already part of the design.

An alternative approach, which we embrace in this work, is to provide *data-driven* suggestions [Hays and Efros 2007; Lalonde et al. 2007; Hartmann et al. 2010]. These are suggestions that are tailored to the user's current design, yet draw from a large set of pre-existing creations. Data-driven suggestions are not the only way to explore existing libraries of relevant solution components: standard search interfaces and catalogues of existing content all provide access to pre-existing examples that can facilitate the creative process. However, data-driven suggestive interfaces have the advantage of providing customized, *in situ* suggestions that are adapted to the user's current design and yet draw from the richness of existing collections of prior content [Lee et al. 2010].

1.2 Prior Work

Igarashi and Hughes [2001] describe a 3D modeling interface that uses a set of rules to analyze the user's actions and suggest subsequent operations. For example, a rectangle is suggested when the user draws two orthogonal line segments emanating from a point. Matejka et al. [2009] use data-driven techniques to recommend under-utilized menu commands to AutoCAD users.

Modeling by example [Funkhouser et al. 2004] allows the user to search a model library and assemble desired models from segments of retrieved shapes. Modeling by example requires the user to search for the specific part they want to add. It is thus less appropriate for open-ended stages of the 3D modeling process, in which the end-goal is ill-defined [Cross 2001].

The Shuffler system [Kraevoy et al. 2007] allows the user to load a hand-selected set of compatible 3D models and quickly interchange similar parts between the models. The system relies on a compatible segmentation of all models and cannot be used to suggest novel, unexpected additions to the user's current model. Pauly et al. [2005] present a technique for completing missing regions of laser range scans by searching a model library. Their work focuses on automated completion of known models rather than creative modeling. Gal et al. [2007] describe a data-driven interface for creating 3D collages that resemble an input shape.

The exploratory modeling system of Talton et al. [2009] applies a data-driven approach to support the open-ended aspects of 3D modeling. Unlike our work, it relies on the existence of a parametric space from which models are drawn. Our work likewise addresses the open-ended nature of early-stage creative 3D modeling, yet is compatible with existing geometric modeling tools.

In summary, no prior work tackles data-driven suggestion of novel components that augment a prototype 3D model. Our method infers suggestions entirely from surrounding context without additional user input, and can present multiple choices, at multiple attachment points, at any single time. The advantage is not only autocomplete-like functionality, but the potential for unexpected, yet suitable suggestions.

2 Shape Retrieval

The first step in generating data-driven suggestions for a user's query model is to identify a diverse set of database models from which suggestions can be drawn. For large databases, we must efficiently evaluate the similarity between query and target shapes. Finding explicit correspondences between shapes is expensive. Hence, prior research has employed statistical signatures that summarize shapes and can be efficiently compared [Osada et al. 2002].

The generation of data-driven suggestions imposes two requirements on shape retrieval. First, since the query shape is unlikely to exactly match any part of a database model, we must support approximate shape retrieval. Second, since suggestions are derived from components of database models that are not present in the query shape, the search must be robust to clutter. We employ a histogram-based signature that is robust to clutter and is easy to generate and compare.

A multidimensional histogram can encode aspects of both a shape's global spatial structure and its local detail [Ling and Soatto 2007; Ovsjanikov et al. 2009]. Our signature captures the distribution of pairwise distances between every pair of descriptor categories. Pairs of samples from the model's surface are quantized to a three-dimensional histogram with two axes corresponding to the descriptor values of the two samples, and the third axis to the distance between them (Figure 2).

We use the shape diameter function (SDF) as our surface descriptor [Shapira et al. 2008]. SDF captures the local thickness of the shape at a sample point, as measured by a ray along the inward normal. The values are normalized to $[0, 1]$ by dividing by the shape's diameter. In our implementation, each histogram axis is quantized into 32 bins. To reduce quantization artifacts, binned values are treated as low-variance Gaussians and allowed to "bleed" into neighboring bins. The variance is chosen so that the Gaussian integrates to 0.5 within a single histogram bin when its mean coincides with the bin center.

A useful property of this "descriptor-descriptor-distance" (D^3) histogram is its local response to incremental addition or removal

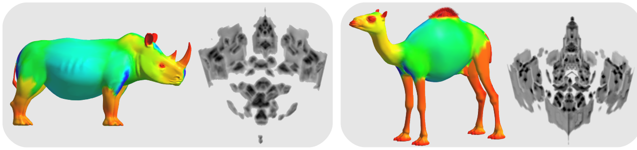

Figure 2: *Shape signatures. Distribution of SDF values over two shapes (blue/green: high; red: low), and volume visualizations of their corresponding* D^3 *histograms. The vertical axis in the histograms corresponds to distance between surface descriptor values.*

of parts of the shape. Since the histogram aggregates $\binom{n}{2}$ pairs of samples, modification of a single sample affects $n-1$ pairs, or $O(n^{-1})$ of the histogram. Thus, a change to a small part of the surface affects only a proportional fraction of the histogram.

Support for partial similarity is completed by the use of the histogram intersection kernel (HIK) [Swain and Ballard 1991], defined as the sum of the pairwise minima of corresponding histogram bins. This kernel has been shown to outperform Lebesgue (L_p) norms in visual classification tasks [Maji et al. 2008]. A notable advantage of the HIK is that it is well suited to matching objects that are only partially similar, since the intersection operation discards unmatched clutter.

In the absence of a standardized scale for all objects, we normalize shapes to the unit cube. Thus if the additional components present in the second shape significantly alter the scale of the object, the above analysis no longer holds. Pyramid matching, described below, is employed in part to compensate for varying scale.

Pyramid matching. The similarity of two shapes can be evaluated by examining corresponding histogram values. Such comparison is sensitive to histogram resolution: histograms with too few subdivisions are not discriminative enough, while excessive refinement eliminates approximate matches. To overcome this problem, we use the pyramid match kernel [Grauman and Darrell 2007].

A histogram pyramid consists of a hierarchy of lower-resolution versions of a base histogram, with each successive level halving the resolution of the previous one. Two pyramids are compared via the sum of pairwise similarities of their corresponding levels, weighted by an exponential attenuation for higher levels. The attenuation reflects the fact that these represent approximate matches, with a coarser discretization of values. Two vaguely similar shapes will have a low but positive matching score from higher-level matches, even if their base (high-resolution) histograms are dissimilar.

Diversification. A key requirement of shape search for creativity support is the diversity of retrieved shapes. If presented suggestions are extremely similar to each other, their utility is diminished and the "wide walls" characteristic of effective creativity support is violated [Resnick et al. 2005].

Given a set S of n search results ranked by similarity to the query Q, we diversify this set using the Maximal Marginal Relevance criterion [Carbonell and Goldstein 1998]. The algorithm reorders the set S, yielding a diversified sequence $D_1, D_2, \ldots, D_n$ with increased spacing between similar results. D_1 is chosen to be the element of S most similar to the query. Each subsequent D_i is chosen according to the formula

$$D_i \equiv \underset{D \in S \setminus S_{i-1}}{\arg\max} \left[\lambda \,\mathrm{sim}(D, Q) - (1-\lambda) \max_{D_j \in S_{i-1}} \mathrm{sim}(D, D_j) \right],$$

where $S_{i-1} \equiv \{D_1, D_2, \ldots, D_{i-1}\}$ and $\mathrm{sim}(\cdot,\cdot)$ is the similarity measure described above, combining the histogram intersection kernel and the pyramid match kernel. $\lambda \in [0,1]$ is a diversity coefficient controlling the degree of reordering. $\lambda = 1$ preserves the original ranking and $\lambda = 0$ yields maximum diversity at the cost of relevance to the query. We used $\lambda = 0.3$ in our implementation.

3 Suggestion Generation

From a diverse set of database models retrieved for the query, our method identifies novel parts that can be used to enhance the initial shape. An interactive application should be able to generate suitable suggestions from a database model, or reject it as unsuitable for suggestions, within a second or two. This requirement renders approaches based on accurate alignment of models – which typically requires computationally expensive global optimization – unsuitable.

We use a statistical approach that does not require pre-alignment. The approach computes a correspondence score for each sample point on the database model, indicating the likelihood of it having a counterpart on the query. The correspondence score is produced by comparing a local signature of the sample point to signatures of sample points on the query shape. Correspondence scores are averaged over segments in the database model, and segments with scores lower than a threshold are presented as suggestions. Duplicates of prior suggestions are detected and discarded by comparing their shape signatures.

The presented approach is fast, robust to the complexity of the underlying shape, and is highly parallelizable. Statistical aggregation over segments stabilizes the approach against individual matching errors. The quality of the local signatures at sample points is still important. We believe, however, that these are easier to develop than general alignment algorithms, and describe an appropriate signature in Section 3.2.

3.1 Preprocessing

To improve the coherence of suggested parts, each model in the library is segmented into components during preprocessing. A significant stream of recent work on shape segmentation employs spectral analysis [Huang et al. 2009; Reuter 2010]. Despite its formal elegance, this approach is brittle in the presence of non-manifold topology. Our database contains models that are non-manifold, featuring disconnected faces, seemingly arbitrary disconnected components, and concealed interior structures. This is a common feature of large model databases and calls for segmentation techniques that are robust to such "polygon soups."

Our implementation combines two segmentation algorithms, one based on the shape diameter function (SDF) [Shapira et al. 2008] and the other based on approximate convex decomposition (ACD) [Lien and Amato 2007]. Each has its advantages and disadvantages. SDF isolates parts of a shape that have similar (or smoothly varying) thickness, but is sensitive to internal structure in the shape, such as the seats inside a car. ACD accommodates internal structure, but breaks curved components (such as the tentacles of an octopus) into smaller parts. We found the combination of these approaches to be more robust than each in isolation.

We first partition the shape using the SDF distribution. For efficiency, and to accommodate badly-tessellated models, we operate on a set of surface points sampled uniformly by area rather than the vertices of the mesh itself. A graph is constructed by joining two sample points if they are each among the 40 nearest neighbors of the other, measured using Euclidean distance between 4-dimensional feature vectors of which the first 3 coordinates specify position in 3-space and the 4^{th} is proportional to SDF value. A graph-cut algorithm subdivides this graph into clusters. Adjacent clusters whose union is approximately as convex as the original clusters are recombined. Global reflective symmetry is respected during segmentation [Podolak et al. 2006].

The clustering is converted to a segmentation of the original mesh by assigning to each mesh face the most frequent label of samples in the neighborhood of its centroid. A regularization step smoothes segment boundaries.

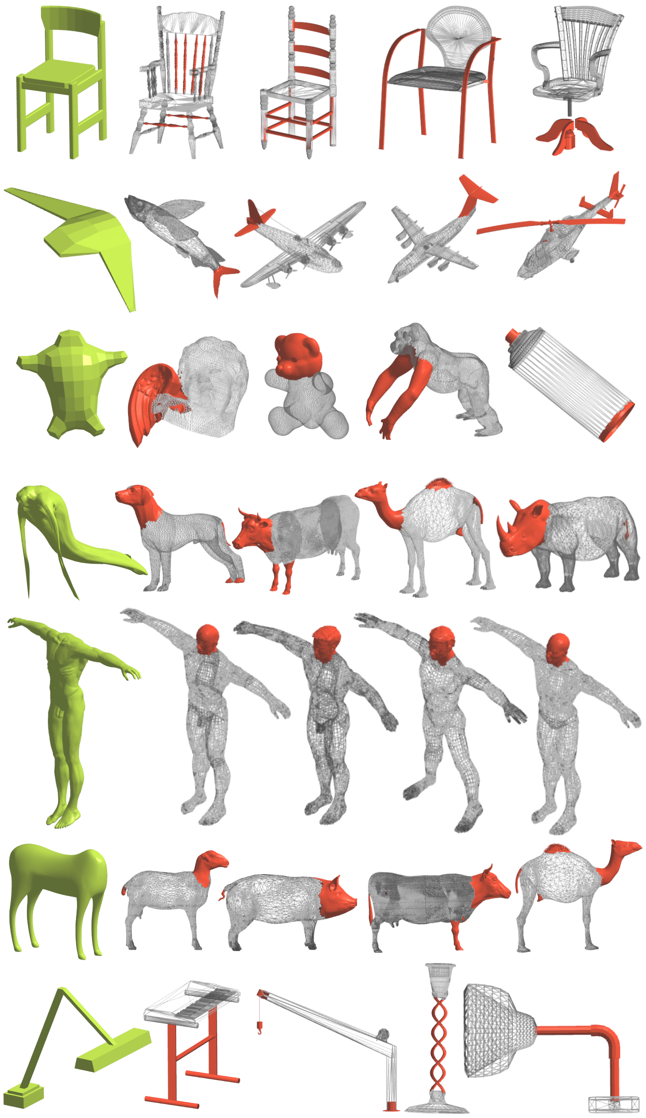

Figure 3: *Top-ranked suggestions. Seven query shapes (green) and the four top-ranked suggestions (red) generated for each by our technique. The automatically retrieved database models that yielded the suggestions are indicated in wireframe.*

3.2 Correspondence Score Computation

Local signatures. In order to measure correspondence scores between sample points on a database model and sample points on the query shape, we compute a local signature for each sample point. Signatures for samples on database shapes are computed during preprocessing. To facilitate accurate correspondence score computation, the local signature must describe both the local neighborhood of the sample point and its global context. To this end, for a sample p on a shape P, we combine a 32×32 spin image $S_P(p)$ [Johnson 1997] with a two-dimensional histogram $D_P(p)$ that bins samples within a local neighborhood by their SDF value and their distance from p.

Spin images have been shown to be effective in robustly capturing the global context of surface points. However, this very robustness makes them insensitive to local detail. On the other hand, the radial distribution of neighbors with varying SDF value is an excellent indicator of local shape, but does not scale well to larger contexts in our experiments. We thus combine these components.

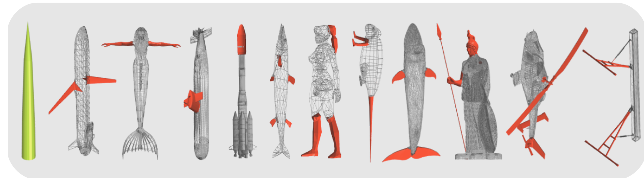

Figure 4: *Suggestions (red) for an ambiguous shape (green).*

Note that both components of the signature are histograms. Given a sample p on a shape P and a sample q on a shape Q, their level of correspondence is defined as

$$\mathrm{cor}(p,q) \;\equiv\; \mathrm{sim}(S_P(p), S_Q(q)) \cdot \mathrm{sim}(D_P(p), D_Q(q)),$$

where $\mathrm{sim}(\cdot,\cdot)$ is the histogram similarity measure described in Section 2, combining the histogram intersection kernel and the pyramid match kernel. The correspondence score of a point q on the query shape Q, with respect to a database model P, is defined as

$$\mathrm{cor}_P(q) \;\equiv\; \max_{p \in \Pi(P)} \mathrm{cor}(p,q),$$

where $\Pi(P)$ is the set of sample points on P.

Efficiency. The complexity of a naïve computation of $\mathrm{cor}_P(q)$ for all sample points q on the query shape grows quadratically, since the correspondence measure $\mathrm{cor}(p,q)$ must be computed for all pairs of samples p, q. To combat this prohibitive performance cost, we use high-dimensional nearest-neighbor search techniques. This is non-trivial, since the distance metric $\mathrm{sim}(\cdot,\cdot)$ is a combination of two histogram similarity measures (the histogram intersection kernel and the pyramid match kernel), and operates on histogram vectors with hundreds of dimensions.

A natural choice for approximate nearest-neighbor search in spaces with hundreds of dimensions is locality sensitive hashing (LSH) [Indyk and Motwani 1998]. Traditional locality sensitive hashes are restricted to L_p metrics or inner products. By Mercer's Theorem, any continuous, symmetric, positive semi-definite kernel function $k(\cdot,\cdot)$ can be expressed as an inner product in a different high-dimensional space, to which traditional LSH can theoretically be applied. However, this mapping is not necessarily of practical value, since the target space can have arbitrarily high (or infinite) dimension.

Our boosted similarity function clearly satisfies the Mercer condition, since it is the product of known Mercer kernels [Odone et al. 2005; Grauman and Darrell 2007]. To leverage this property for efficient evaluation of the correspondence score for a sample point, we wish to construct a hash function that implicitly exploits the Mercer condition without requiring the actual embedding into a higher-dimensional (or infinite-dimensional) space to be constructed. To this end, we employ kernelized locality-sensitive hashing, which extends LSH to Mercer kernels [Kulis and Grauman 2009]. Since the correspondence score for each sample point can be evaluated independently of the others, the computation is trivially parallelizable.

4 Results

4.1 Suggestion Generation

We first present timings for various steps in the suggestion generation pipeline. Performance data was obtained on an 8-core 2.53 GHz workstation. For shape retrieval, it took 0.047 milliseconds to compare two D^3 signatures on a single core. Performance scales linearly: total shape retrieval time is 0.056 seconds (excluding I/O) with a library of 1193 models. Correspondence computation, with precomputed signatures at 3,000 sample points per shape, took 0.3 seconds per pair of shapes, averaged over 1,409 such pairs. The standard deviation was 0.23 seconds. We used three 30-bit hashtables on a kernel matrix of 20 signatures to prune 87% of the search space on average, with an approximate nearest neighbor found 84% of the time with 6% error.

Over the same set of shapes, it took an average of 0.4 seconds to process and reject a retrieved database model when it did not contribute valid suggestions. (This includes the correspondence computation time above.) It took 0.83 seconds to process a retrieved model from which suggestions were accepted. The longer times are due to slower correspondence computation for similar models, and the pruning of similar suggestions. Extrapolating suggestions from the set of surface samples to the underlying mesh (which frequently had >50,000 faces) took an additional 1.62 seconds on average.

Figure 3 shows the highest-ranked suggestions produced for various query shapes. Figure 4 shows suggestions produced for an ambiguous shape that does not naturally lend itself to interpretation.

4.2 Prototyping Tool

To examine the use of data-driven suggestions by 3D artists, we have created a prototyping tool using the techniques presented in this paper. The tool, dubbed InspireMe, lets an artist request suggestions for a rough initial shape. The artist can quickly place and glue any of the suggestions to the query shape, and request new suggestions for the composite shape. This allows the artist to visually mock up a prototype and then export it to a high-end 3D modeling package for refinement and texturing.

The InspireMe interface is shown in Figure 5. InspireMe displays suggestions from 12 automatically retrieved and diversified library models, based on the finding that designers prefer to be shown roughly 10 examples [Lee et al. 2010]. The suggestions are highlighted to increase their visual saliency [Treisman and Gelade 1980]. The artist can add any of the suggested parts to the query shape. The initial placement of the added suggestion on the query shape is computed automatically, as described below. This initial placement can be refined by translating, rotating, and scaling the added suggestion in relation to the query shape. Any number of suggestions can be added and removed, supporting rapid visual exploration of possible designs.

Suggestions that are not useful can be discarded with a mouse click, and a new set of suggestions is automatically generated from a library model with a lower shape retrieval score. An entirely new batch of suggestions can be requested for a newly assembled composite shape. In this way, the user can mock up increasingly refined prototypes.

Placement and gluing of suggestions. When the user chooses to add a suggested part, InspireMe computes an initial attachment point for it, which the user may subsequently fine-tune. The approximate attachment point is computed by approximately aligning the query shape with the model from which the suggestion was drawn. Our implementation uses the 4-PCS algorithm with 1000 iterations [Aiger et al. 2008]. When both models are detected to have planes of symmetry, we employ an optimized variation of the algorithm that aligns the models based on the symmetry plane and two widely separated points.

Once the part is in place, InspireMe enables simple gluing of proximal surfaces. Each vertex on the suggested part that is within a threshold distance of the query shape is displaced towards its nearest neighbor on the latter. The displacement magnitude falls off with distance, and the displacement field is regularized over local neighborhoods by convolving with a Bartlett filter. A more advanced implementation could utilize high-quality gluing techniques [Sharf et al. 2006].

4.3 Informal Studies

To evaluate the effectiveness of creativity support with data-driven suggestions, we invited twelve 3D artists to test our prototype implementation. Three of the users were university-level art instructors and professional artists, eight were art students, and one was a computer science student and hobbyist 3D modeler. Ten of the participants had to travel to the lab from a different municipality and were compensated for their time. The users were given two open-ended modeling tasks designed to mimic assignments tackled by 3D artists. A few users who were only available for a short time were assigned only one of the tasks. The two tasks were:

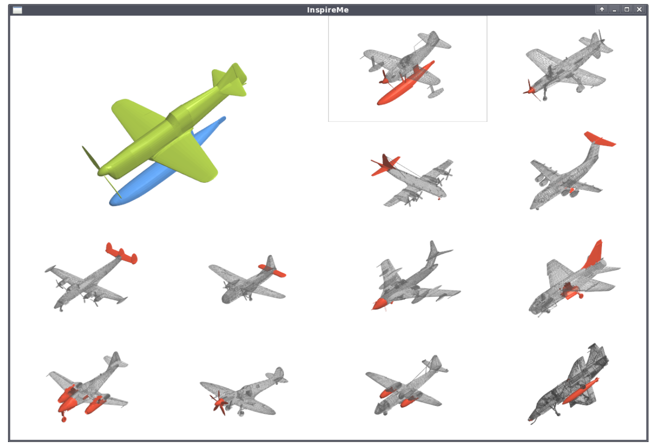

Figure 5: *InspireMe interface, showing a query shape (green) and suggestions for it (red). One suggestion has been selected and added to the mockup (blue).*

Aircraft You are an artist designing futuristic military aircraft for the upcoming sci-fi film *Colony 2130*. These will be used in a large extraterrestrial battle scene. Several types of craft are required, such as a large mothership, attack bombers, low-altitude strafe fighters, search-and-rescue craft for evacuating injured soldiers from alien jungles, etc. Conceptualize and create a mockup design for one such aircraft.

Creature You are a creature designer for the upcoming sci-fi film *Avatar 2*. This sequel will feature Pyrrha, the sister-planet of Pandora from the original film. The dense jungles of Pyrrha teem with fantastical creatures. Conceptualize and create a mockup design for one such creature.

In both cases, the artists were allowed to freely interface between InspireMe and modeling packages of their choice, such as Maya®, 3ds Max® and ZBrush®. In the beginning of each session, we provided a brief overview of the InspireMe interface. We encouraged users to spend no more than 10-20 minutes to construct a rough initial query shape. Models were transferred between InspireMe and other applications using a shared folder. InspireMe simultaneously ran queries on two databases, Digimation ModelBank (1,193 models) and Digimation Archive (11,461 models with >1,500 faces).

Models created by the artists during these sessions are shown in Figure 6. Note the rudimentary starting shapes, and the fleshed-out concepts in the final mockups. The mockups feature little sculpting beyond basic assembly (and occasional duplication) of suggested parts, which constitute by far the bulk of the transition from query to concept.

Figure 7 visualizes the artists' activity during the modeling sessions. Activity time was determined from system logs. The artists frequently worked with modeling programs and InspireMe in parallel, sometimes refining a composite model in Maya while InspireMe was searching for new suggestions. When it was clear from the first page of suggestions that the tool was confused by a particularly ambiguous query model, the artists usually quickly refined it

Figure 6: *Models created by artists for the aircraft task (top) and the creature task (bottom). For each model, the figure shows the initial query shape (top row), the mockup created with InspireMe (middle row), and the final textured model created from the mockup (bottom row). All novel parts in the mockups are derived from data-driven suggestions. For each task, every model was created by a different artist.*

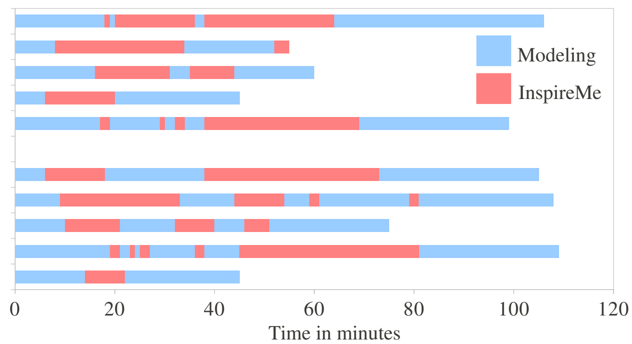

Figure 7: *Activity time in modeling programs (blue) and InspireMe (red) during the creation of the models in Figure 6.*

to add more context (such as adding wings to a hull) and submitted a new query. This accounts for the occasional rapid switches of activity between the modeling software and InspireMe.

In one instance, an artist decided to experiment by seeking to enhance a finished, production-quality model that he had constructed earlier. He was pleasantly surprised when InspireMe quickly made a few suggestions that could enhance his model and still maintain a coherent yet unusual shape. The initial and final models are illustrated in Figure 8.

User feedback. A number of art students noted that InspireMe would be useful for early-stage 3D design, with one commenting "I would definitely use a tool like this during an early conceptual phase" and another saying InspireMe "would be great for whiteboxing." The term "whiteboxing" refers to the practice, in the design of game art, to create rough concept models before fully refining them. A professional concept artist suggested that InspireMe would be useful for a local company specializing in the creation of concept art for film and games, further commenting that the tool fits naturally into the production workflow of concept artists.

One suggestion for improvement was to add the ability to select a suggestion and request "more like this," or reject a suggestion and request that similar ones be suppressed. Artists also suggested automatically remeshing the suggested components to match the tessellation characteristics of the query shape.

5 Conclusion

This work leverages statistical geometry processing techniques to enable a data-driven approach to creativity support in 3D modeling. The suggestion generation method introduced in this paper is compatible with traditional geometric modeling tools. Data-driven suggestions could thus be integrated into such tools to assist artists with conceptual 3D design. More broadly, the development of data-driven techniques that support open-ended design tasks is a challenging research direction that can advance the practice of three-dimensional content creation.

The most significant limitation of the presented technique for suggestion generation is its purely geometric nature, which does not take into account the meaning and function of the suggested components. Recent advances indicate the feasibility of semantically labeling components in large collections of three-dimensional content [Kalogerakis et al. 2010]. Utilization of semantic information can substantially increase the effectiveness of suggestion generation. More generally, reasoning about the semantics of shapes and their components can advance the power of data-driven tools for three-dimensional content creation.

Finally, the effectiveness of data-driven content creation tools is closely related to the quality of the available data. Existing 3D model libraries are still limited in their extent, fidelity, and availability. On the other hand, the Web has already been used for collaborative creation of extensive image libraries, with millions of freely available, ranked, labelled, high-quality photographs [Flickr 2010]. Large-scale collaboration can similarly enable the construction of comprehensive and detailed databases of three-dimensional content. Such databases can drive the development of new kinds of tools for three-dimensional content creation.

Acknowledgments

We are grateful to Daniel Cohen-Or for his comments on drafts of this paper, anonymous reviewers for their feedback, and Leonidas Guibas, Scott Klemmer and Fei-Fei Li for helpful discussions. Chris Platz helped organize the informal studies and create the associated images and video segments. We thank all the participants in the studies. This work was supported in part by NSF grants SES-0835601 and CCF-0641402.

References

AIGER, D., MITRA, N. J., AND COHEN-OR, D. 2008. 4-points congruent sets for robust pairwise surface registration. In *Proc. SIGGRAPH*, ACM.

BODEN, M. A. 1990. *The Creative Mind: Myths and Mechanisms*. George Weidenfeld and Nicolson Ltd.

CARBONELL, J., AND GOLDSTEIN, J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. SIGIR Conference on Research and Development in Information Retrieval*, ACM, 335–336.

COLLINGWOOD, R. G. 1938. *The Principles of Art*. Clarendon Press.

CROSS, N. 2001. Design cognition: results from protocol and other empirical studies of design activity. In *Design Knowing and Learning*, C. Eastman, M. McCracken, and M. Newstetter, Eds. Elsevier Science, 79–103.

FINKE, R. A., WARD, T. B., AND SMITH, S. M. 1992. *Creative Cognition: Theory, Research, and Applications*. MIT Press.

FLICKR, 2010. http://www.flickr.com/.

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.

GAL, R., SORKINE, O., POPA, T., SHEFFER, A., AND COHEN-OR, D. 2007. 3D collage: expressive non-realistic modeling. In *Proceedings of 5th International Symposium on Non-Photorealistic Animation and Rendering*.

GRAUMAN, K., AND DARRELL, T. 2007. The pyramid match kernel: efficient learning with sets of features. *Journal of Machine Learning Research 8*, 725–760.

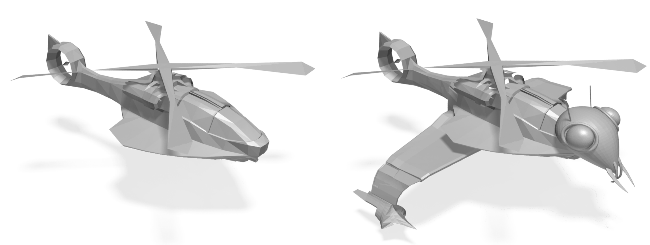

Figure 8: *Enhancing an existing model. A previously completed model (left) and the enhanced model (right), which uses three suggestions made by InspireMe (insect head, flat wings from racing car, curved wings from spacecraft).*

HARTMANN, B., MACDOUGALL, D., BRANDT, J., AND KLEMMER, S. R. 2010. What would other programmers do? Suggesting solutions to error messages. In *Proc. ACM Conference on Human Factors in Computing Systems*.

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

HUANG, Q.-X., WICKE, M., ADAMS, B., AND GUIBAS, L. J. 2009. Shape decomposition using modal analysis. *Computer Graphics Forum 28*, 2, 407–416.

IGARASHI, T., AND HUGHES, J. F. 2001. A suggestive interface for 3D drawing. In *Proc. ACM Symposium on User Interface Software and Technology*.

INDYK, P., AND MOTWANI, R. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. ACM Symposium on Theory of Computing*.

JOHNSON, A. E. 1997. *Spin-Images: A Representation for 3D Surface Matching*. PhD thesis, Carnegie Mellon University.

KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3D mesh segmentation and labeling. In *Proc. SIGGRAPH*, ACM.

KRAEVOY, V., JULIUS, D., AND SHEFFER, A. 2007. Model composition from interchangeable components. In *Proc. Pacific Graphics*, IEEE Computer Society.

KULIS, B., AND GRAUMAN, K. 2009. Kernelized locality-sensitive hashing for scalable image search. In *Proc. International Conference on Computer Vision*, IEEE Computer Society.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2007. Photo clip art. In *Proc. SIGGRAPH*, ACM.

LEE, B., SRIVASTAVA, S., KUMAR, R., BRAFMAN, R., AND KLEMMER, S. R. 2010. Designing with interactive example galleries. In *Proc. ACM Conference on Human Factors in Computing Systems*.

LIEN, J.-M., AND AMATO, N. M. 2007. Approximate convex decomposition of polyhedra. In *Proc. ACM Symposium on Solid and Physical Modeling*.

LING, H., AND SOATTO, S. 2007. Proximity distribution kernels for geometric context in category recognition. In *Proc. International Conference on Computer Vision*, IEEE Computer Society.

MAJI, S., BERG, A. C., AND MALIK, J. 2008. Classification using intersection kernel support vector machines is efficient. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society.

MARSH, R. L., LANDAU, J. D., AND HICKS, J. L. 1996. How examples may (and may not) constrain creativity. *Memory and Cognition 24*, 5, 669–680.

MATEJKA, J., LI, W., GROSSMAN, T., AND FITZMAURICE, G. 2009. CommunityCommands: command recommendations for software applications. In *Proc. ACM Symposium on User Interface Software and Technology*.

NICKERSON, R. S. 1999. Enhancing creativity. In *Handbook of Creativity*, R. J. Sternberg, Ed. Cambridge University Press, 392–430.

ODONE, F., BARLA, A., AND VERRI, A. 2005. Building kernels from binary strings for image matching. *IEEE Transactions on Image Processing 14*, 2, 169–180.

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *ACM Transactions on Graphics 21*, 4, 807–832.

OVSJANIKOV, M., BRONSTEIN, A. M., BRONSTEIN, M. M., AND GUIBAS, L. 2009. ShapeGoogle: a computer vision approach for invariant shape retrieval. In *Proc. ICCV Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*.

PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. J. 2005. Example-based 3D scan completion. In *Proc. Symposium on Geometry Processing*.

PODOLAK, J., SHILANE, P., GOLOVINSKIY, A., RUSINKIEWICZ, S., AND FUNKHOUSER, T. 2006. A planar-reflective symmetry transform for 3D shapes. In *Proc. SIGGRAPH*, ACM.

RESNICK, M., MYERS, B., NAKAKOJI, K., SHNEIDERMAN, B., PAUSCH, R., SELKER, T., AND EISENBERG, M. 2005. Design principles for tools to support creative thinking. In *NSF Workshop Report on Creativity Support Tools*. 25–35.

REUTER, M. 2010. Hierarchical shape segmentation and registration via topological features of Laplace-Beltrami eigenfunctions. *International Journal of Computer Vision 89*, 2, 287–308.

SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Visual Computer 24*, 4, 249–259.

SHARF, A., BLUMENKRANTS, M., SHAMIR, A., AND COHEN-OR, D. 2006. SnapPaste: an interactive technique for easy mesh composition. *Visual Computer 22*, 9, 835–844.

SHNEIDERMAN, B., ET AL. 2006. Creativity support tools: report from a U.S. National Science Foundation sponsored workshop. *International Journal of Human-Computer Interaction 20*, 2, 61–77.

SHNEIDERMAN, B. 2007. Creativity support tools: accelerating discovery and innovation. *Communications of the ACM 50*, 12, 20–32.

STERNBERG, R. J. 1999. *Handbook of Creativity*. Cambridge University Press.

SWAIN, M. J., AND BALLARD, D. H. 1991. Color indexing. *International Journal of Computer Vision 7*, 1, 11–32.

TALTON, J. O., GIBSON, D., YANG, L., HANRAHAN, P., AND KOLTUN, V. 2009. Exploratory modeling with collaborative design spaces. In *Proc. SIGGRAPH Asia*, ACM.

TERRY, M. A., AND MYNATT, E. D. 2002. Side views: persistent, on-demand previews for open-ended tasks. In *Proc. ACM Symposium on User Interface Software and Technology*.

TERRY, M. A., MYNATT, E. D., NAKAKOJI, K., AND YAMAMOTO, Y. 2004. Variation in element and action: supporting simultaneous development of alternative solutions. In *Proc. ACM Conference on Human Factors in Computing Systems*.

TREISMAN, A. M., AND GELADE, G. 1980. A feature-integration theory of attention. *Cognitive Psychology 12*, 97–136.

WEISBERG, R. W. 2006. *Creativity: Understanding Innovation in Problem Solving, Science, Invention, and the Arts*. John Wiley & Sons, Inc.