3D Collage: Expressive Non-Realistic Modeling

Ran Gal[1] Olga Sorkine[2] Tiberiu Popa[3] Alla Sheffer[3] Daniel Cohen-Or[1]
[1]Tel Aviv University [2]TU Berlin [3]University of British Columbia

Figure 1: Lazy Boy. (The couch was manually added.)

Abstract

The ability of computer graphics to represent images symbolically has so far been used mostly to render existing models with greater clarity or with greater visual appeal. In this work, we present a method aimed at harnessing this symbolic representation power to increase the expressiveness of the 3D models themselves. We achieve this through modification of the actual representation of 3D shapes rather than their images. In particular, we focus on 3D collage creation, namely, a generation of compound representations of objects. The ability of such representations to convey multiple meanings has been recognized for centuries. At the same time, it has also been acknowledged that for humans, the creation of compound 3D shapes is extremely taxing. Thus, this expressive but technically challenging artistic medium is a particularly good candidate to address using computer graphics methods. We present an algorithm for 3D collage generation that serves as an artistic tool performing the challenging 3D processing tasks, thus enabling the artist to focus on the creative side of the process.

Keywords: non-realistic modeling, expressive imagery, computer-aided art, new media

1 Introduction

For a long time, one of the primary goals of computer graphics research has been the generation of realistic models and photorealistic images. Nowadays, perhaps similarly to the evolution of fine art, more attention can be devoted to abstract and symbolic representations. This shift is apparent in the research field of non-photorealistic rendering (NPR) [Gooch and Gooch 2001; Strothotte and Schlechtweg 2002]. NPR methods typically assume that the geometric model is a given and develop techniques to render the model in a particular style, focusing on the aesthetic aspects or the technical illustration to increase the clarity of the subject. These techniques do not usually change the actual content of the image. In contrast, the expressiveness of traditional art pieces is often increased by modifying the actual shape of the portrayed objects, by abstracting the objects and by introducing symbolic or suggestive content. These approaches became particularly popular in 20th century art. It has been observed that such expressive models better depict our mental view of the world, and are thus more effective in conveying ideas and content [Hanrahan 2005].

Most existing computer graphics work in this area focuses on 2D models, leading to tools such as caricature, cartoon, mosaics and 2D collages [Finkelstein and Range 1998; Hausner 2001; Kerne 2001; Kim and Pellacini 2002; Liang et al. 2002; Klein et al. 2002], or creating stylized 2D images of 3D models [Gooch and Gooch 2001; DeCarlo et al. 2003; Hertzmann 2003]. Akleman [2004] discusses the possibility of generating 3D caricatures using standard 3D modeling packages. Alternatively, in this paper we introduce an expressive modeling method, where the expressiveness is manifested in the 3D shape itself, rather than in its rendering. We call this approach *non-realistic expressive modeling*, where the 3D model is meant to express certain ideas, whose semantics are set by the artist. Our research is inspired by the expressive ability of surrealist art, most notably the “path-blazing” work of Giuseppe Arcimboldo (1527-1593). Arcimboldo noticed that human visual perception can separate shape from content and with his allegorical figures composed of vegetables and fruit, helped to usher in the surrealist era [Janson and Janson 1991]. One particular assertion of the movement is that if a familiar shape is modeled as consisting of multiple objects, humans can simultaneously recognize both. A single image or statue can convey more than one concept, providing a more expressive medium. This concept was embraced by many artists in the 20th century, including Picasso, Miró, Dalí, Chagall, Warhol and Rauschenberg.

However, the technical difficulty of modeling a complex 3D shape from other shapes meant that most composition works dealt only with 2D representations such as images. When 3D compositions were required, artists such as Rauschenberg and Warhol incorporated the actual item into the artwork, thereby sidestepping the is-

Figure 2: Arcimboldo's paintings (1527-1593).

sue of how to work with 3D representational versions of their materials. Thus, this expressive but technically challenging modeling technique is a particularly good candidate to address using computerized geometric methods.

We develop an interactive algorithm for creating expressive 3D compound shapes, or 3D collages. Given a target input shape, these collages convey that shape by accentuating and abstracting its salient features. The collage model is comprised of elements taken from a database of (possibly simple) shapes. The composition of these distinct elements together, when viewed as a whole, resembles the target shape (see Figure 1). We note that collage techniques, such as ours, differ from mosaic, where the parts are typically small and do not attract attention. In contrast, our technique seeks to create expressive models where both the whole, i.e. the target shape, and its parts are recognizable, and where the artist can control the semantic relationship between the whole and its parts.

A modeling system aimed at artists should find the right balance between automation and control. A fully automatic assembly process leaves no room for user creativity. At the same time, simply providing a 3D modeling environment and leaving the users to perform the shape fitting on their own is impractical, since partial matching of virtual objects in 3D is quite challenging even for experienced modelers. We achieve the desired tradeoff by providing the users with a number of control mechanisms allowing them to influence the collage assembly, and then performing the assembly process automatically using the user input as guidance. The developed controls are easy to navigate and allow the artist to direct the collage assembly, but do not require her to understand the details of the underlying algorithms.

Partial matching is one of the key ingredients in our collage creation technique, coupled with a score function that evaluates the matches with respect to the global geometry of the collage. Partial matching has been used for several tasks, among those modeling by example [Funkhouser et al. 2004] is the more related to our application. The goal of Funkhouser and colleagues [2004] was to enable interactive design of new models, aided by assembling parts of models found in a database; in this setup users are expected to indicate each individual replaced part, and then careful stitching is performed to ensure that the resulting shape is precisely connected. In contrast, in a collage setup the parts do not connect to each other; instead, the collage creation is influenced by the *interaction* between the different parts, rather than just their individual matching. The algorithm is capable of automatically deciding which parts to fit and where, since no user-defined segmentation of the target shape is expected.

2 Problem Statement

Our modeling creates compound 3D models (collages) of *target* shapes from *elements* or shapes taken from a given database. We devise an algorithm that finds a collection of elements and places them in such a way that both the target shape that we wish to model, and the parts are *recognizable*. The elements may be seen as general "proxies" [Cohen-Steiner et al. 2004; Wu and Kobbelt 2005] with which we wish to represent the given target shape; however, the approximation is meant to be crude since we are not seeking geometric fidelity. Therefore, our collage assembly algorithm is based on the following principles:

1. To make the collage resemble the target shape, each element in the collage should fit some region on the target surface well. The area of collage parts that protrude outside the target shape should be small.
2. Since the elements should be recognizable, the visibility of each element within the collage should be maximized. Thus, we prefer to place elements that approximate a large part of the target shape, and do not overlap the regions approximated by other elements.
3. Collages should resemble physically plausible constructions. Therefore we limit the amount of intersection between the different parts in the collage. While we allow some variation in the scale of the elements during fitting, to facilitate realism we restrict the scale variance, preventing abnormalities like cherries the size of a watermelon.

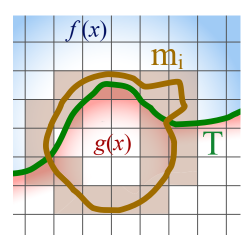
The main building block of our algorithm is a local fitting scheme, based on these principles, that given a point on the surface of the target shape, finds elements in the database that fit that point and its surroundings well. The fitting considers both partial shape similarity between the target and the element, and the interaction between the new element and the previously added parts of the collage.

Using the local fitting, described in more detail in Section 3, we develop a collage assembly framework which allows varying levels of user control, described in Section 4.

3 Element Fitting

Building the collages requires retrieving elements in the database whose parts fit well certain areas around a given point of the target shape. We first find the elements in the database such that some portion of each element *locally* approximates the region around a given point well, employing a partial shape matching technique (described in Section 5); the partial matching algorithm provides a similarity transformation (translation, rotation, scale) that gives the best local fit.

Next, our fitting scheme evaluates the quality of each match based on a score function Q and a number of constraints. The score measures how well the *entire* element matches the target, according to the first principle listed in Section 2. Let us denote the target shape by T and the current matched element by m_i. The score function $Q(T, m_i)$ should favor elements that lie close to the surface of T but do not protrude too far. We define Q by superimposing the surface of m_i against the distance field of T and weighing the overlap against the protrusion. More precisely, we compute a biased version of the signed Euclidean distance transform [Mullikin 1992] of T, discretized on a voxel grid around T. Let us denote the regular distance transform of T by d_T ($d_T < 0$ inside the shape, $d_T = 0$ in voxels that intersect the surface of T and $d_T > 0$ outside); each voxel v is assigned a value $D_T(v)$ as follows:



$$D_T(v) = \begin{cases} 1 & d_T(v) = 0 \\ f(d_T(v)) & d_T(v) > 0 \\ g(d_T(v)) & d_T(v) < 0 \end{cases} .$$

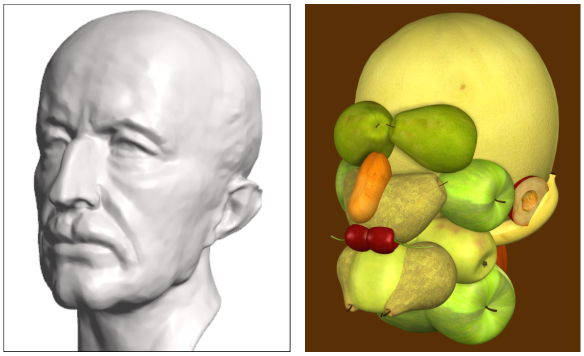


Figure 3: Max Planck – homage to Arcimboldo.

The biasing functions are defined as follows:

$$f(x) = 1 - \alpha x, \;\; g(x) = e^{-\frac{x^2}{2(0.01\mu)^2}} \,,$$

where $\mu = \max_v(-d_T(v))$). The element m_i is also rasterized on a voxel grid D_{m_i}, which coincides with the grid of T. The voxels of D_{m_i} that intersect the boundary of m_i attain the value 1 while all others are set to 0. We then define $Q(T,m_i)$ as inner product of D_T with D_{m_i}, i.e.

$$Q(T,m_i) \;\; = \;\; < D_T, D_{m_i} > \,.$$

The exponential drop of D_T inside the shape strongly favors large *surface* overlap between T and m_i, while the linear negative growth of D_T outside T penalizes protrusion. Therefore Q is positive when m_i does not protrude too far, where the tolerance is controlled by the parameter α. Increasing α biases the algorithm towards more accurate (tighter) matches; we usually used $\alpha = 0.6u$, where u is the bounding box diagonal.

In order to obtain meaningful local fits, we enforce a size constraint on the candidate elements requiring that the target surface area portion covered by the new element will be larger than a certain percentage. This restriction prevents the construction of busy, mosaic-like collages with many small elements. In contrast to mosaics, our goal is to build a collage where the elements are recognizable and thus project real content.

The quality score function and the sizing constraint consider only the quality of the fit to the target, disregarding the interaction of the candidate elements with the elements previously added to the collage. As noted in the problem statement (Section 2), we want to minimize the occlusion and intersection between the elements in the collage, and to preserve the relative proportions between the elements. Therefore, when computing the list of candidate elements, we only add those that satisfy the following constraints.

Visibility: The new element does not cover more than $M\%$ of the already covered target surface. This constraint prevents redundancy in the collage and enables distinct fitting of the salient parts of the target shape.

Overlap limit: The new element should have no more than $X\%$ volumetric overlap with the current collage.

Proportions: The scale ratio between the element and the target shape is within ε-tolerance from a proportion value set by the user. The proportion is defined individually for each element as part of the artistic decision.

The last two constraints provide a more physically plausible result. Clearly, it is up to the artist to define the degree of plausibility or realism desired. The elements that satisfy all the constraints are ordered based on their score and comprise the candidate list for the specified target shape region. The candidate lists of all target surface points are used as the basis for the collage assembly.

4 Collage Assembly Framework

Before beginning the collage assembly, the user specifies the desired set of elements by selecting them from any of the existing online databases (e.g., the Princeton Shape Benchmark [Shilane et al. 2004]). Typically the set has a specific semantic meaning and thus can be selected based on the database pre-classification. The parts and the target shape are then pre-processed by computing their local shape descriptors, as described in Section 5. Performing this task beforehand allows efficient partial matching queries during the actual assembly of the collage. After the pre-processing, the algorithm uniformly samples points on the target surface, and computes lists of candidate matches for each point, as described above. If desired, the parameters for the matching can be modified at any stage of the algorithm. If the user chooses to do so, the algorithm recomputes the matches.

Our collage creation algorithm follows the principles described in Section 2. Regrettably, finding the assembly that simultaneously maximizes fitting and visibility while minimizing the intersections size, is infeasible. Instead, our algorithm uses the greedy approach, gradually building the collage, adding the elements one by one. It performs local fitting for each of the pre-computed sample points on the target surface and obtains the list of best matching candidate elements, ordered according to their score (Section 3). In each step the algorithm chooses the best fitting element that satisfies the constraints and adds it to the collage. This element covers some part of the target surface; the algorithm updates the covered area, and proceeds to the next best match. It terminates when the surface area of the target is covered.

The collaging interface is based on an iterative collage updating process. After the user specifies the target model and the database, the algorithm assembles a collage of the entire target model, as described above. If the user modifies any of the algorithm controls, for either the entire target or part of it, the collage is recomputed on the fly. The process is continued, until the user finds the result satisfactory. For instance, in Figure 6, the body of the seahorse was assembled using an initial set of parameters. We then reduced the approximation tolerance in the eye regions, requiring a finer fit and thus causing the algorithm to add the two small shells for the eyes.

The assembly framework supports a variety of user controls aimed at users with different levels of modeling expertise:

- At any point during the assembly process the users can indicate which parts of the collage should be kept as-is and which need

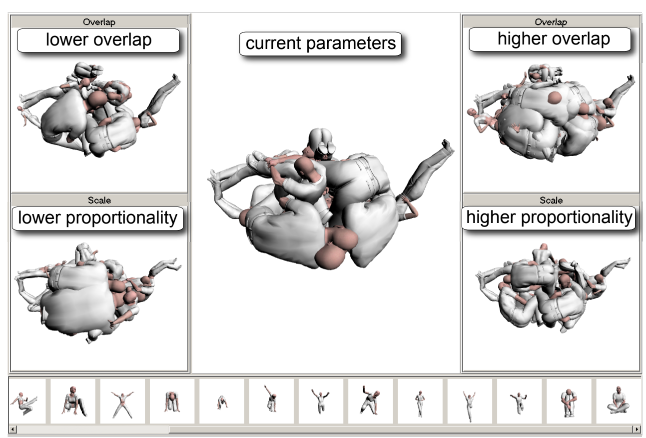


Figure 4: Parameter navigation interface. The result for the current set of parameters is displayed in the middle, while the effects of decreasing and increasing some parameters (in this case: allowed overlap and scale proportionality) are shown on the left and right.

further adjustment. The users can also indicate which parts of the collage should be disregarded during subsequent processing.

- By clicking on icons of the database shapes, the users can control which portion of the database is used for any part of the model. For example, in Figure 1 the hair of the character was restricted to be modeled with cigarettes only.
- The users can control the fitting parameters: level of fitting accuracy and allowed protrusion, amount of overlap and relative scale for any part of the model. The parameters can be set using either a standard menu interface or parameter space navigation. The menu interface is aimed at expert users who understand the algorithms involved. For the non-expert users we developed the parameter space navigation interface (Figure 4) where the users can in addition to the current collage see the collages created when one of the parameters is increased or decreased by a constant increment. By clicking on one of the modified collages, the parameter and the view are updated accordingly. To avoid clutter, only a couple of the parameters are shown in this setup, and the users can easily select which parameters they want to navigate at any given time.
- We provide a greater level of control for expert users, allowing them to directly override some of the greedy algorithm's choices. Given a collage constructed in a greedy manner, the users can click on any of the shapes in the collage and choose from other candidate shapes that the algorithm considered when fitting that spot. This ability to select not necessarily the best match, but one of the best matches, allows the artist to introduce additional expressive and aesthetic considerations into the process.

Figure 5: The human knot.

5 Partial Shape Matching

To find the elements in the database that match a region on the target shape, we developed a simple partial matching algorithm. Partial shape matching has been addressed by a number of papers in Computer Graphics [Huber and Hebert 2003; Funkhouser et al. 2004; Li and Guskov 2005; Gal and Cohen-Or 2006]. It is usually carried out in the following framework: (*i*) the matched regions on the shapes are defined; (*ii*) these regions, also called patches, are transformed into a canonical pose so that they can be described in the same coordinate system, and the aligning transformations (usually rigid or similarity) are recorded; (*iii*) a chosen shape descriptor is computed for pairs of patches and the values are compared to determine the degree of shape similarity. Rotation-invariant shape descriptors, such as the spherical harmonics [Kazhdan et al. 2003], may be computed without pre-aligning the shapes (canonical scaling is still needed), if the sole goal is to decide how similar the shapes are, e.g., in a search and retrieval application. The aligning transformations, also called the canonical transformations, need to be recorded only if one is interested to position the shapes in their matched configuration, as it is in our case.

We define different regions that will be tested for partial matching by intersecting the models with solid spheres of various radii. Formally, we uniformly point-sample the surfaces of the elements and the target model and compute the set of patches $\{S_{r,p}\}$ such that

$$S_{r,p} = S \cap \mathscr{S}(r,p), \;\; p \in S, \;\; r \in \{\frac{1}{n}R(S), \frac{2}{n}R(S), \ldots, \frac{k}{n}R(S)\},$$

where the S is a shape (database element or the target shape), $\mathscr{S}(r,p)$ is a solid sphere of radius r centered at p, $R(S)$ is the diagonal length of the oriented bounding box of S and k,n determine the range of sizes we test (between 0.05 and 0.2 in our examples). We chose this definition over geodesic disks on the surface, because it also implicitly takes the local volume of the model into account. We compute the canonical transformation of each patch using PCA, such that the center of mass of the patch is translated to the origin, and the principal axes coincide with the coordinate axes, in decreasing order of principal components. Canonical scaling is achieved by uniformly scaling all compared patches so that the length of the largest principal component becomes 1.

There are many different shape descriptors, as reviewed and compared in, e.g., [Shilane et al. 2004], developed mostly for the purpose of 3D search engines. In our implementation, we used geometric moments, reviewed below, as our shape descriptor. We have also implemented the spherical harmonics descriptor [Kazhdan et al. 2003], which without doubt outperforms the geometric moments for global shape matching. In particular, spherical harmonics do an excellent job of discerning different *classes* of models, e.g., humans, animals, mechanical models, etc. However, in our particular *partial* matching scenario, we found that there is no significant difference between the results of spherical harmonics and the moments, since the patches being compared are not complete shapes with clear characteristic features but rather smaller, smooth parts. We chose geometric moments primarily because they are faster to compute and in our setting require a smaller signature.

Moments have been used for matching and recognition of 2D shapes in images [Hu 1962; Abo-Zaid et al. 1988; Suk and Flusser 2001]. To the best of our knowledge, Elad et al. [Elad et al. 2000; Elad et al. 2001] were the first to use this method for global shape matching in 3D. The p,q,r-moment of a shape S is

$$M_{p,q,r}(S) = \int_{\partial S} x^p y^q z^r \, \mathrm{d}x\mathrm{d}y\mathrm{d}z\,.$$

For uniformly sampled shapes we replace the integral with a sum over all the surface points. The moments of a shape S up to some order d are stored in a descriptor vector:

$$V(S) = (M_{0,0,0},\, M_{1,0,0}, \ldots,\, M_{i,j,k}, \ldots,\, M_{p,q,r}), \;\; \text{s.t.} \;\; i+j+k \le d.$$

It has been shown that the descriptor vectors $V(S)$ and $V(T)$ of geometrically similar shapes S and T are close, namely the 2-norm $\|V(S) - V(T)\|$ is small. In our implementation we use order $d = 7$, thus the descriptor vectors are 120-dimensional.

To address partial matching queries during the collage assembly, we compute the various patches $\{S_{r,p}\}$ of all database elements S and insert their shape descriptor vectors $V(S_{r,p})$ into a k-nearest-neighbors search data structure. We also store the canonical transformation for each patch, which we later use to correctly orient the elements during the assembly. This preprocessed element database may be used for building collages of various target shapes. The set of descriptors for the target is computed in the same fashion, and during the construction of the collage we can query each point on the target for partial matches with the database elements.

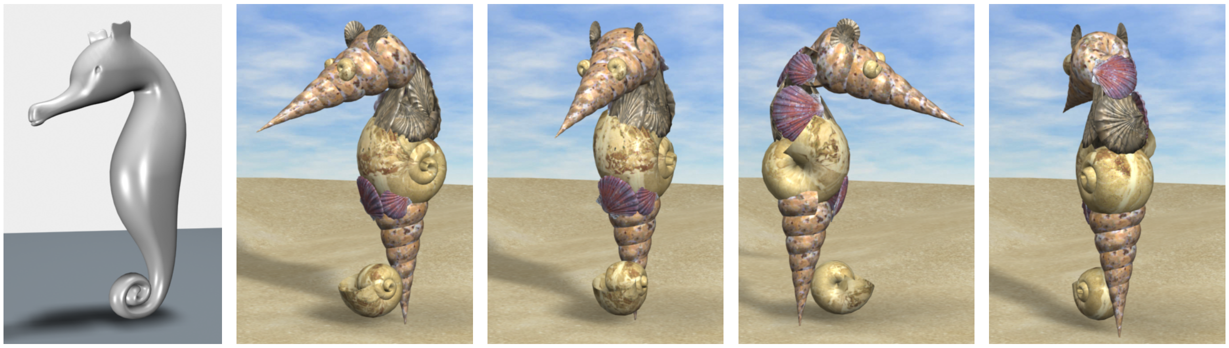


Figure 6: Hippocampus botticelli (*Seahorse*).

6 Implementation and Results

We implemented the collage construction environment in C++ and tested it on a Pentium 4 3.0 GHz computer. The most time-consuming task during the collage construction is the nearest-neighbors search in the space of shape descriptors while looking for locally matching parts; we efficiently implemented it using the ANN library [Mount and Arya 2005]. Given a pre-processed database, the automatic construction of a collage takes a few seconds for all the models we tested, therefore it is easy to interactively modify the various parameters, as demonstrated in the accompanying video. The pre-processing of an element database takes about 2 hours with our unoptimized code (the databases we tried contained several dozens elements of moderate geometric complexity). Note that the preprocessing is trivially parallelizable.

We have created several collages with various element databases, testing the different capabilities of our interactive tool. The results presented in Figures 5, 9, 12 were created using the same set of parameters for the entire model. The *Human Knot* in Figure 5 consists of 52 human models in 16 different poses. Due to the complexity, it is practically impossible to construct such a model manually. We also created mosaic-like examples consisting of a few hundreds of parts (Figure 9). We observe that at this point the elements become harder and harder to recognize.

In the *Seahorse* collage (Figure 6) we used one set of parameters for the body. We then increased the fitting parameter α in the eye and ear regions to obtain tighter approximation, and manually selected the fitted elements from the list of candidates. Constructing this model from seashells is quite challenging since many of the shells have a non-convex, almost flat structure. We created the *Lazy Boy* collage (Figure 1) in a similar fashion. First, the algorithm created the body of the character using one set of parameters. Then, the construction was restricted to the hair region and the element database was reduced to one element, a cigarette; the standard algorithm was then used to obtain the hair elements. Finally, the fitting parameter α was increased for the eyes and "bloated" region at the front of the face, and the best fitting elements provided by the algorithm were added for them.

When making the *Running Sportsman* collage (Figure 11), we first used the entire sports dataset to obtain a collage. The algorithm matched the basketball to the body, the helmet to the head, the shoes to the feet, and various balls and helmets to the rest of the shape. It did, however, use the football for far too many regions in the model. We removed the football from the database and reran the method on the regions where we wanted other matches, creating more variety among the matched balls.

The collages in Figures 7 and 13 were obtained from an initial automatic result by replacing some parts by other candidates, to enforce symmetry between the leg elements in the case of the *Handyman* and to create better separation between the shins and the thighs for the *Vegetarian Woman*. The collage of Max Planck (Figure 3) was modeled on one side of the face and then mirrored using an external modeling tool to create perfect symmetry. The collage in Figure 8 was created by choosing each element separately from the candidates list (this is the most user-intensive interaction mode). The main advantage of this mode is the ability to override the algorithm choices without considering which parameters need change.

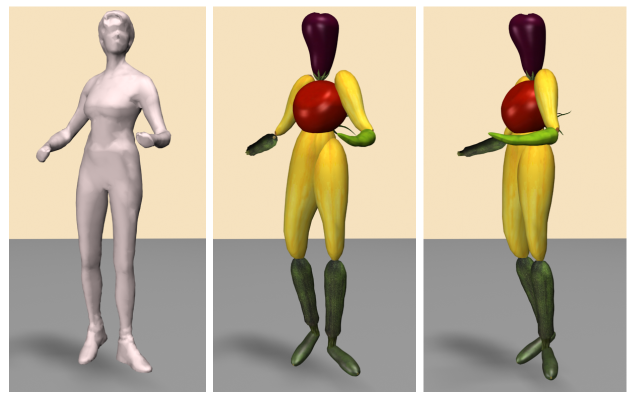


Figure 7: You are what you eat (*Vegetarian Woman*).

Note that since the elements of the collage are matched to the target shape, we can easily "lift" animation from the target to the collage. We demonstrate this in the video with the *Running Sportsman*.

The 3D collages often contain intersecting objects. While it is possible to constrain the tool to completely avoid intersections, some collages would then turn out too "empty" or will contain many overly small elements. It is possible to come up with alternative rendering styles for the intersecting collages. For example, instead of rendering with traditional hidden-surface removal, it is possible to render the collage element by element in back-to-front order. The order of the elements is determined by the nearest depth value of each object. This rendering technique is demonstrated in Figure 10. The fruits appear as if they do not intersect one another, although, of course, the feeling of depth is less realistic. Another option is to render the intersecting parts with "depth-darkening" [Luft et al. 2006], so that they appear cut-out (see Figure 10).

7 Conclusions

Computer-aided artistic tools serve as powerful paintbrushes of the 21st century. Our work adds a new, conceptually different brush to the artist's existing arsenal of tools. While previous tools focused on rendering existing models, our work addresses expressive mod-

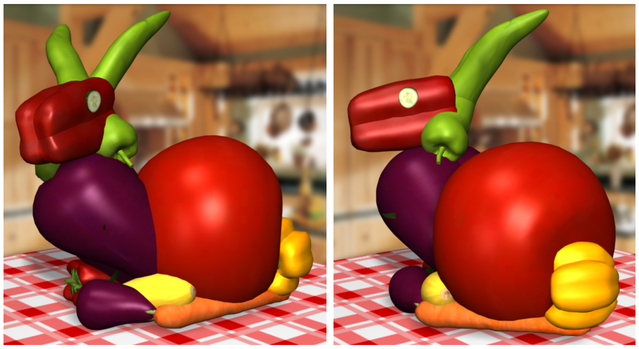


Figure 8: Abundant Bunny.

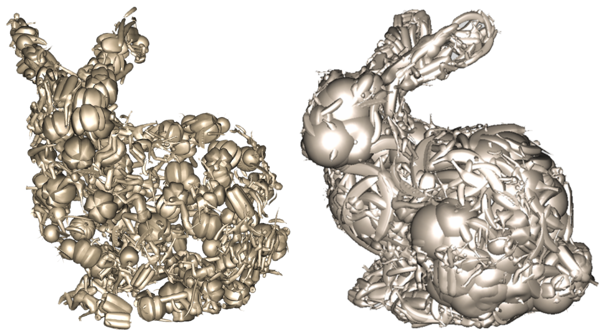


Figure 9: Mosaic-like collages. The database contains 38 elements; the collages are comprised of 351 parts (left) and 607 parts (right).

eling. Our tool does not deal with the way objects are displayed, but actually changes the way they are modeled. In particular, we introduce a computerized tool for modeling 3D collages. The collage is constructed from 3D elements that roughly approximate the target shape. Since both the target and the elements are recognizable, the result can expressively convey layered concepts. The algorithmic core of our technique is a partial matching method, where the local shape descriptors are based on geometric moments.

We believe that our work only skims the surface of the potential of computerized modeling techniques to serve as effective tools for computer-aided art. Another interesting tool to investigate is the ability of geometric techniques to distill the core structure of complex shapes through simplification and skeleton construction. As with our work, the challenge here is to harness the technical ability of computer algorithms, while maintaining the artist's ability to control the creative process.

Acknowledgements

This work was supported in part by the Israeli Ministry of Science, NSERC, MITACS NCE, IBM and the Alexander von Humboldt Foundation. We would like to thank Ciaran Llachlan Leavit for providing help with editing and proofing this paper.

References

ABO-ZAID, A., HINTON, O. R., AND HORNE, E. 1988. About moment normalization and complex moment descriptors. In *Proceedings of the 4th International Conference on Pattern Recognition*, 399–409.

AKLEMAN, E. 2004. Modeling expressive 3D faces. In *Proceedings of ACM SIGGRAPH 2004*, Sketch.

COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape approximation. *ACM Trans. Graph. 23*, 3, 905–914.

DECARLO, D., FINKELSTEIN, A., RUSINKIEWICZ, S., AND SANTELLA, A. 2003. Suggestive contours for conveying shape. *ACM Trans. Graph. 22*, 3, 848–855.

ELAD, M., TAL, A., AND AR, S. 2000. Directed search in a 3D objects database using SVM. Tech. Rep. HPL-2000-20 (R.1), HP Laboratories Israel, August.

ELAD, M., TAL, A., AND AR, S. 2001. Content based retrieval of VRML objects: an iterative and interactive approach. In *Proceedings of the 6th Eurographics workshop on Multimedia*, Springer-Verlag New York, Inc., 107–118.

FINKELSTEIN, A., AND RANGE, M. 1998. Image mosaics. In *Proceedings of RIDT*, 11–22.

FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. *ACM Trans. Graph. 23*, 3, 652–663.

GAL, R., AND COHEN-OR, D. 2006. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph. 25*, 1, 130–150.

GOOCH, B., AND GOOCH, A. 2001. *Non-Photorealistic Rendering*. A. K. Peters, Ltd., Natick, MA, USA.

HANRAHAN, P. 2005. Realistic or abstract imagery: The future of computer graphics?, keynote talk. In *Eurographics 2005*.

HAUSNER, A. 2001. Simulating decorative mosaics. In *Proccedings of ACM SIGGRAPH*, 573–580.

HERTZMANN, A. 2003. A survey of stroke-based rendering. *IEEE Computer Graphics and Applications 23*, 4, 70–81.

HU, M.-K. 1962. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory IT-8* (February), 179–187.

HUBER, D., AND HEBERT, M. 2003. Fully automatic registration of multiple 3D data sets. *Image and Vision Computing 21*, 7 (July), 637–650.

JANSON, H., AND JANSON, A. F. 1991. *A Basic History of Art*. Prentice, Englewood Cliffs, NJ.

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *SGP*, 167–175.

KERNE, A. 2001. *Collagemachine: a model of "interface ecology"*. PhD thesis, NYU.

KIM, J., AND PELLACINI, F. 2002. Jigsaw image mosaics. *ACM Trans. Graph. 21*, 3, 657–664.

KLEIN, A. W., GRANT, T., FINKELSTEIN, A., AND COHEN, M. F. 2002. Video mosaics. In *Proceedings of NPAR*, 21–28.

LI, X., AND GUSKOV, I. 2005. Multi-scale features for approximate alignment of point-based surfaces. In *Proceedings of SGP*, 217–226.

LIANG, L., CHEN, H., XU, Y.-Q., AND SHUM, H.-Y. 2002. Example-based caricature generation with exaggeration. In *Proceedings of Pacific Graphics*, 386–393.

LUFT, T., COLDITZ, C., AND DEUSSEN, O. 2006. Image enhancement by unsharp masking of the depth buffer. *ACM Trans. Graph. 25*, 3, 1206–1213.

MOUNT, D. M., AND ARYA, S. 2005. *ANN: A Library for Approximate Nearest Neighbor Searching*. http://www.cs.umd.edu/~mount/ANN/.

MULLIKIN, J. C. 1992. The vector distance transform in two and three dimensions. *CVGIP: Graphical Models and Image Processing 54*, 6, 526–535.

SHILANE, P., MIN, P., KAZHDAN, M., AND FUNKHOUSER, T. 2004. The Princeton shape benchmark. In *Proceedings of Shape Modeling International*, 167–178.

STROTHOTTE, T., AND SCHLECHTWEG, S. 2002. *Non-Photorealistic Computer Graphics. Modelling, Rendering, and Animation*. Morgan Kaufmann.

SUK, T., AND FLUSSER, J. 2001. Features invariant simultaneously to convolution and affine transformation. In *Proceedings of CAIP*, 183–190.

WU, J., AND KOBBELT, L. 2005. Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum 24*, 3, 277–284.

Figure 10: Back-to-front rendering and "cut" rendering.

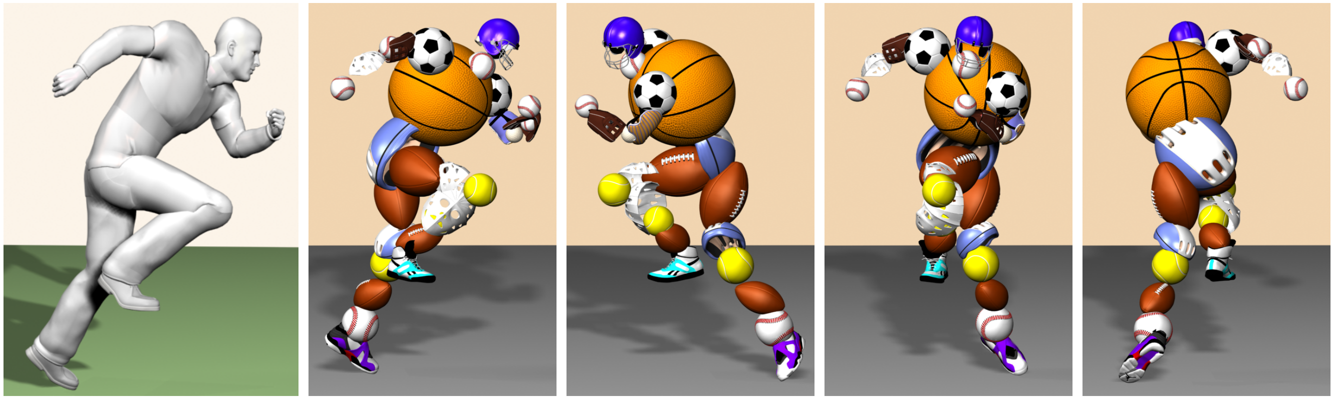


Figure 11: Collage of a running sportsman.

Figure 12: *Stanford Bunny* and *Camel* a-la Arcimboldo.

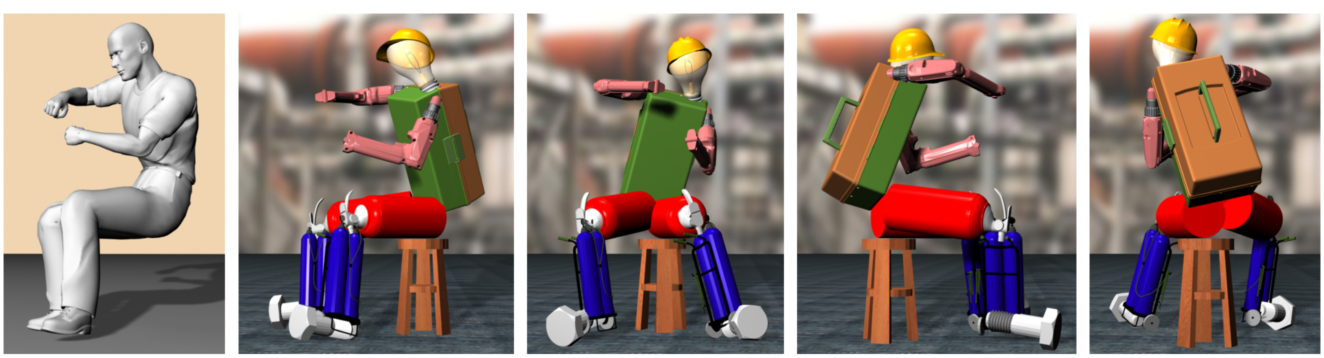


Figure 13: Handyman. (The stool was manually added.)