SWAPART: Synthetic Object Creation by Part Substitution

Ronen Lev
Dept. of Comp. Sc.
University of Haifa
Haifa 31905, Israel
rlev01@cs.haifa.ac.il

Gershon Elber
Dept. of Comp. Sc.
Technion, IIT
Haifa 32000, Israel
gershon@cs.technion.ac.il

Dan Gordon
Dept. of Comp. Sc.
University of Haifa
Haifa 31905, Israel
gordon@cs.haifa.ac.il

Abstract

An artistic, semi-automatic modeling scheme, called SWAPART, is presented. It allows a user to swap parts of existing geometry using similarly-looking geometry. The new scheme, which can be viewed as a computerized analogy of the traditional art technique known as assemblage, *draws heavily on recent results in the area of 3D geometric search engines, and is geared toward novice or artistic end-users. Given a 3D model that the user may obtain from any source, a novice end-user can extract different parts from the model and query a server that holds information about a large database of 3D parts for possible substitutions. The end-user works in an interactive client environment that provides the needed interaction capabilities and the transparent server connections. The output geometry (with the selected substituting parts) is a 3D model that can be further processed by any computer graphics tool, including this client-server system.*

1. Introduction and Background

Geometric modeling is a very challenging task. For several decades now, great effort has been invested in creating intuitive and interactive tools to model geometry using computers. Despite the remarkable success, modeling must still be done by experts. So far, very few tools have been offered to novice users to create geometry. Over the last several years, several attempts were made at sketch-based preliminary design such as [9], where silhouettes are used to construct the geometry or [4, 20] where intuitive and interactive systems were built to allow sketch-based preliminary designs.

In this paper, we propose an alternative modeling approach that utilizes the huge body of models that is accessible via the Web. A graphic modeler can build a whole variety of scenarios by fetching different pieces (i.e., models) from this massive repository. Our approach enables the end-user to extract sub-parts of a given model and replace them with another part that is geometrically similar. We call our technique SWAPART – "swap-part-art". Our method relies on recent results in the area of 3D geometric search engines. It does not propose any new geometry search method, but it can use any current or future method.

A precursor to our approach is the technique used by sculptors called *assemblage* [3], where a sculpture is constructed from other objects. It is a natural 3D extension of the 2D notion of montage. Pablo Picasso (the "Bull's head" sculpture) [18, 19], Julio González [18] and others used this practice in their work.

There are several papers on cut and paste modeling [17, 1, 12], in which a fusion method is proposed between parts cut from one model and pasted into another. Our technique complements these methods by providing the artist with appropriate ways to access and utilize these models. A similar method has been concurrently (and independently) developed by Funkhouser et al. [5].

The rest of this paper is organized as follows. In Section 2, we present the proposed system and then follow with the various details of the different algorithms. In Section 3, some examples are presented, and finally we conclude and discuss future directions in Section 4.

2. The SWAPART Modeling System

The idea of swapping parts in existing geometry with similar parts necessitates a search through a significant body of large-sized geometric objects. Since geometry-exchange is expected to be the bottleneck of any such distributed architecture, any system that supports the ability to select certain parts from a large database of geometric models must find ways to minimize the network communications as well as the computational load on the server. To achieve this, we devised a client-server solution in which the client's end-user submits to the server geometric search queries. It sends certain "signatures" of the sought geometry instead of the geometry itself. The client provides all the user interface services to its end-user: fetching the geometry and displaying and manipulating the SWAPART objects. The computation of a part's signature, which is another computationally expensive task, is also done by the client.

On the server side, all the objects in its geometric database are a priori decomposed into their parts, as explained in Section 2.1. For each part, the server calculates and stores a signature and an ID. When a query signature from a client is received, the server compares it with the signatures in its database. The server also allows clients to search for a model in the database according to additional information, e.g., a category such as "animals". The top-ranked results of the search query are sent back from the server to the client together with their ranking, their URL addresses, sub-part location and auxiliary information such as pictorial snapshots.

A geometry search server can contain references to thousands of sub-parts and models, all with their pre-computed signatures. A combined algorithm is employed here in order to sort through the parts, Section 2.2 addresses this issue. At the client level, the user must prescribe the sub-part he or she wishes to swap. Intuitive and interactive part selection and extraction methods must be offered, and two options are discussed in Section 2.3. The resulting geometry that was identified by the server must be properly placed and oriented at the client site, in place of the selected part; this is discussed in Section 2.4.

2.1. Use of the Inherent Object Hierarchy

Many existing geometric objects already contain an internal structure. This structure is common in most modeling environments as it provides the designer with the freedom to work on different portions of the object and/or tag them differently. For example, in an articulated object, each rigid part might move along a different path in an animation sequence.

An artist might wish to replace a selected sub-part with a model that may be a sub-part of another model. For example, the artist might wish to replace a leg of a chair with a leg of another chair. In many cases the geometry server may not contain isolated legs, but it may contain a model that contains a sub-part that is the desired model, i.e., a chair with the desired legs.

Many geometric file formats that are common over the Web (such as VRML, 3ds, DXF etc.) represent the geometry in a tree structure. This structure is utilized by the geometry search server to specify sub-parts of the model. When the geometry search server initially collects information and the signatures of a new object, it also creates a signature for each node in the tree that contains geometry (some nodes may include other information such as light and material information and are ignored). The geometry search server then adds the nodes' information as well as the nodes' location to the geometry tree structure.

2.2. Shape Similarity

Our SWAPART approach requires a semi-automatic mechanism for finding similar objects in a prescribed database or collection of objects. Quite a few new algorithms, in the area of 3D object similarity, were proposed in recent years. One class of algorithms is based on the principle of histogram comparison, where the histogram reflects measurements done on certain random points [15]. This histogram could be formed out of the Euclidean distances between random points, the areas of triangles between three random points on the surface, the volumes of tetrahedra between four random points on the surface, and so on.

Another noteworthy algorithm uses spherical harmonics [6]. The user determines a paramether R and the method converts the object into a volumetric data set of $(2R)^3$ voxels by assigning each voxel a value of one if it is within one voxel width of a polygonal surface, and zero otherwise. We denote by $Voxel(x, y, z)$ the value of a voxel at (integer) coordinates (x, y, z), and define a binary-valued function $f(r, \theta, \phi)$ (for $r \leq R$) as the volume of a voxel function in spherical coordinates. For every integer r, $1 \leq r \leq R$, we define a function f_r as $f_r(\theta, \phi) = f(r, \theta, \phi)$, and take the collection of spherical functions $\{f_1, f_2, \cdots, f_R\}$ as the starting point for the object's signature. In order to make the signature rotation-invariant, each function f_r is decomposed using spherical harmonics basis functions [8, 14]. The coefficients of the decomposition serve as the signature.

In this work for the shape similarity search, we combine the histogram method of Osada et al. [15], with the spherical harmonics of Funkhouser et al. [6]. We selected these methods as they are simple and efficient to compute and require moderate resources even for large models. These methods are also fairly robust to model errors such as self-intersections and missing triangles. Our experience has shown that their combination produces reasonable results.

As the purpose of the resulting models is mostly of an artistic nature, we are only interested in similarity of the exterior of the objects. Since both of the above methods take into account both the visible and the invisible parts of the model, it is necessary to remove the invisible portions, as explained in Section 2.2.1

During a database search we compare signatures by calculating the L_2 distance between the signatures. In order to increase the speed we have adopted a two-level hierarchical process for searching the database. First, Osada's method is applied in the server to all the objects in the database, in order to select a set of objects that are similar. Then, the spherical harmonics comparison algorithm is applied only to the preliminary set. Every object in the final list is assigned a value that is a weighted average of the values computed by the two algorithms. The final list of matching parts is sent back from the geometry search server to the end-user client, who can then request the geometry of the part(s).

2.2.1. Invisible Geometry Removal In this work, we consider only visible polygons in our search for similar objects. For example, the shape of the outer shell of a computer mouse is similar to (half of) an ellipsoid. Yet, considering all the internal machinery and electronics in a computer mouse, chances are small that the fully detailed computer mouse will be considered as a similar alternative to (half of) an ellipsoid.

While finding the hidden subset of polygons from a given *specific* point of view is a well understood problem, finding the polygonal set which is invisible from *any* point of view that is outside the (bounding box of the) object is a much more difficult task. This is a topic of very active research nowadays. Hoever, since we are only interested in objects that are externally similar, we can tolerate errors and employ the following heuristic.

Tessellate the object so that the area of each triangle is smaller than some predetermined threshold, and assign to it a unique color. Then, scan convert the scene using a regular Z-buffer, possibly in hardware, with no shading. The detected color in every non-background pixel uniquely identifies a polygon. The set of visible colors serves to indicate the polygons visible from this view. This process is repeated from multiple views and the sets of visible polygons are combined. We have found that several dozens views provide sufficient accuracy.

2.3. Selecting a Sub-Part

A major interactive task for the end-user is the selection and isolation of the part, G, to be swapped. In this work, we use two different approaches for this task:

- Selection of G by exploiting the hierarchy of the internal parts in the object.
- Clipping out G from the object, as the portion that is inside a user-defined 3D prism.

As previously mentioned in Section 2.1, some models contain an inherent hierarchy. This inherent hierarchy is used to provide the end-user with an ability to select sub-parts that are different leaves in the original object's hierarchy. Our system displays all the subparts of the object in a transparent view, providing the end-user with a general understanding of the structure of the object. Any combination of sub-parts can then be selected (by displaying them as solid). Once the sub-parts have been selected, they are merged into one model, clipped away from the object and processed for the query.

Other geometric models have no parts hierarchy so we provide a simple manual clipping mechanism to cut a prismatic region out of the current object. The user draws a simple planar polygon to define the base of the prism and then defines the extent of the prism. The polygons inside the prism are then clipped using the 'in' and 'out' operators as specified in [2].

2.4. Calculating the Position of the Swapped Geometries

Calculation of the center of mass and principal axes requires a proper physical object. As mentioned, this does not hold for many objects on the Web. We therefore regard the object as an envelope of some geometry (with an infinitesimal width). This approach allows us to calculate integrals over the object's mass as the sum of uniform random distributions of samples over the model, as in [16].

We calculate the position of the part by collecting a large sampled set of uniformly distributed random points. Now, we compute an approximation to the object's center of mass as a simple average of these points. We take as the part's "size" the average of the distance of these sample points from the center of mass.

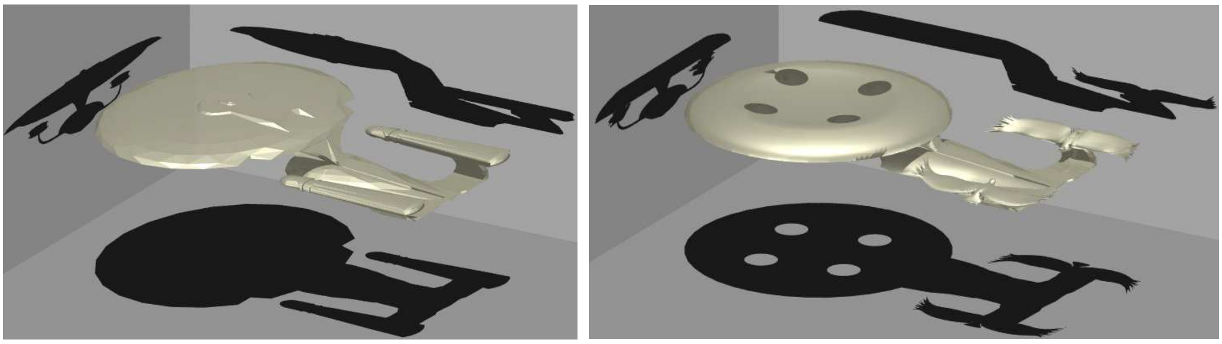
Then, each point in our point sample collection is translated toward the center of the mass and a principal component analysis (PCA) [11] is calculated to compute the appropriate rotation. The newly substituted object $\overline{G}$ is then oriented so that the two eigenvectors of its largest and smallest eigenvalues match those of G. The size of $\overline{G}$ is scaled to the size of G and its center of mass is translated to coincide with that of G. These operations completely register the position and orientation of the geometry.

The new part is placed, by default, in the same position, scale, and orientation as the original part. However, the artist may wish to modify the position or orientation of the new part, so we allow him to transform the geometry as desired.

3. Examples

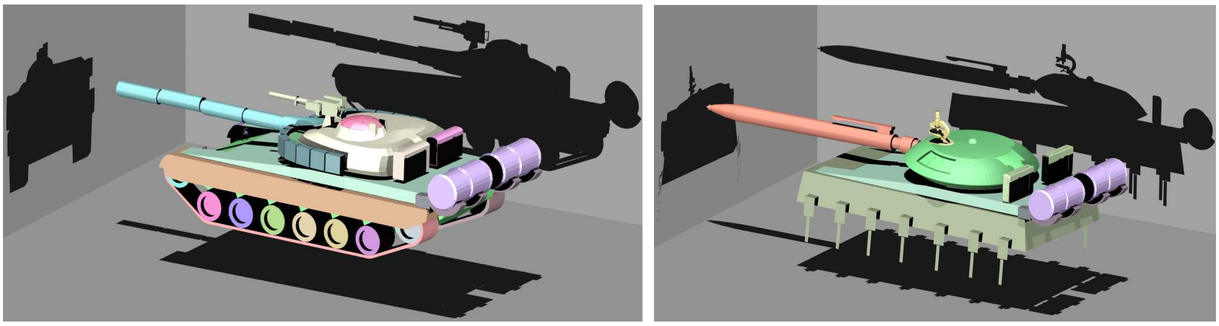
In this section, we present several examples of models that were created with the aid of our implementation. Figure 1 presents a simple model of the starship Enterprise. In Figure 1 (a), the original model is presented whereas in Figure 1 (b) the revised SWAPART version is shown. The main saucer section is replaced with a button, and the engines with birds. In Figure 2 (a), the original tank model is presented and in Figure 2 (b) the revised SWAPART version is shown. The tank's cannon, body gun and turret have been replaced by more peaceful objects.

In Figure 3, a chess set is presented with its original set as the white pieces and SWARART pieces from the Web opposing them. The geometry obtained with SWAPART can be processed by further SWAPART operations. In Figure 3 (a), the original model's head of the lady warrior that was fetched from the Web and that served as the SWAPART queen in Figure 3 is processed further. In Figures 3 (b)-(e), several revised SWAPART versions are presented. The sword in the original model is replaced by new swords, a drill-bit and a pencil, in the order of the quality of the similarity as detected via the geometry search server query.



(a) (b)

Figure 1. (a) The Enterprise. (b) The main section replaced with a button and the engines swapped to birds.



(a) (b)

Figure 2. "...and they shall beat their tanks into chips, and their guns into pens and microscopes...". (a) A tank. (b) Cannon and body replaced with peaceful alternatives.

To gain some idea of the duration of the modeling process, a SWAPART session of a single item such as the sword in Figure 3 several minutes. The creation of moderately complex examples such as Figures 1 or 2 took ten to fifteen minutes of user interaction time. The session that built the chess board in Figure 3, lasted about half an hour. All presented SWAPART results were created using a database of around 4000 models, which were decomposed to over 20000 sub-models.

4. Conclusions and Future Work

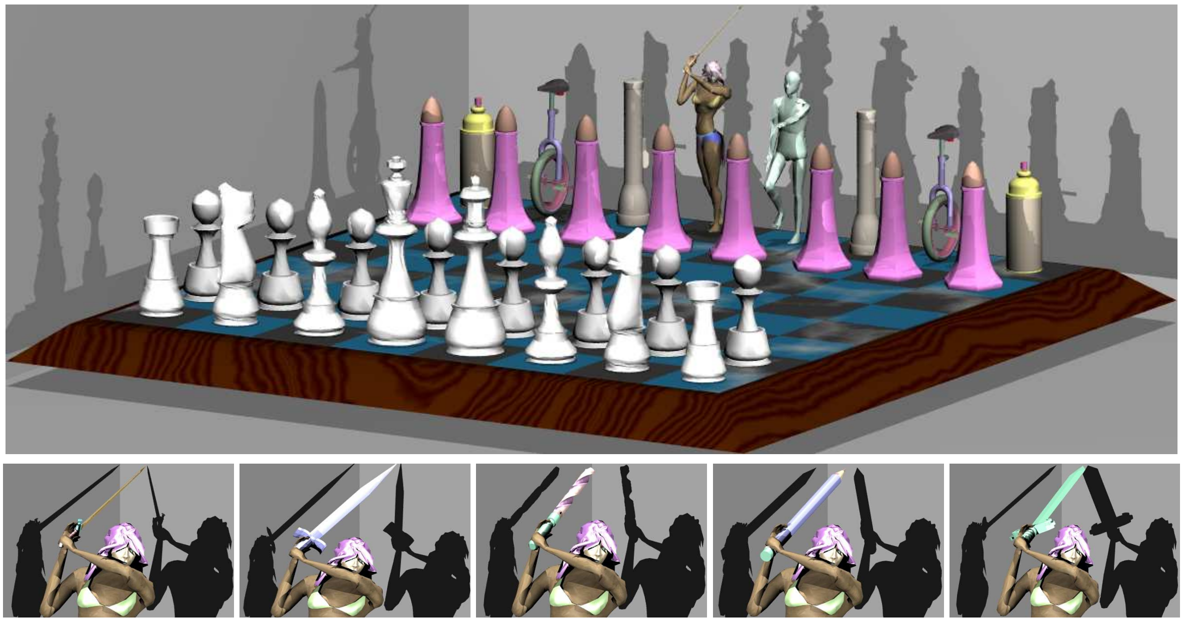
In this work we have presented SWAPART, an artistic modeling scheme for environments with rich geometric content such as the Web. The proposed architecture is based on geometry search servers on the Web, which contain search information regarding many geometric objects.

The current system simply swaps the original part with a new one. A clear and more appealing alternative is to continuously deform the original part into the new and similar part, possibly taking advantage of one of several mesh fusion and cutting and pasting algorithms such as [1, 12].

While the currently selected similarity method seems to be sufficient to get interesting results, other shape similarity methods (such as [10, 7]) may provide better results. Our method can easily incorporate any current or future method.

Acknowledgments. This research was supported in part by the Technion Vice-President for Research Fund - New York Metropolitan Research Fund, by the Israeli Ministry of Science Grant No. 01-01-01509, by the European FP6 NoE grant 506766 (AIM@SHAPE), by a donation from IBM's Shared University Research program to the Technion in 2003, and by the Caesarea Rothschild Institute at the University of Haifa.

The models required to create the examples in this paper were retrieved from 3DCafe (http://www.3dcafe.com), the Drexel repository (http://www.designrepository.org) and the Princeton Shape Benchmark (http://shape.cs.princeton.edu/benchmark/). We used the black box implementation of spherical harmonics signature provided by Michael Kazhdan [13].



(a) (b) (c) (d) (e)

Figure 3. Chessboard with black pieces replaced by similar objects found on the WEB. (a) Head of the lady warrior. (b)-(e): Sword replacements in order of similarity.

References

[1] H. Biermann, I. Martin, F. Bernardini, and D. Zorin. Cut-and-Paste editing of multiresolution surfaces. In *Proceedings of the 29th Conference on Computer Graphics and Interactive Techniques (SIGGRAPH-02)*, volume 21, 3, pages 312–321, 2002.

[2] E. Cohen, R. F. Riesenfeld, and G. Elber. *Geometric Modeling with Splines: An Introduction*. 2001.

[3] M. Delahunt. Artlex art dictionary. http://www.artlex.com/ArtLex/Aru.html.

[4] L. Eggli, C. Hsu, B. D. Brüderlin, and G. Elber. Inferring 3D models from freehand sketches and constraints. *Computer-aided Design*, 29(2):101–112, 1997.

[5] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin. Modeling by example. *ACM Transactions on Graphics (SIGGRAPH 2004)*, Aug. 2004.

[6] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3D models. *ACM Transactions on Graphics*, 22(1):83–105, January 2003.

[7] M. Heczko, D. A. Keim, D. Saupe, and D. V. Vranic. Verfahren zur Ähnlichkeitssuche auf 3D-objekten. *Datenbank-Spektrum*, 2002.

[8] E. Hobson. *The Theory of Spherical and Ellipsoidal Harmonics*. 1955.

[9] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3D freeform design. In *Proceedings of the Conference on Computer Graphics (Siggraph99)*, pages 409–416, 1999.

[10] C. Y. Ip, D. Lapadat, L. Sieger, and W. C. Regli. Using shape distribution to compare solid models. In *Proceedings of the seventh ACM Symposium on Modeling and Application (SM-02)*, pages 273–280, June 2002.

[11] I. T. Jolliffe. *Principal component analysis*. 1986.

[12] T. Kanai, H. Suzuki, J. Mitani, and F. Kimura. Interactive mesh fusion based on local 3D metamorphosis. In *Proceedings of Graphics Interface 99*, pages 148–156, 1999.

[13] M. Kazhdan. Tools for spherical harmonic representations. http://www.cs.princeton.edu/∼mkazhdan/ HarmonicSignatures/.

[14] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proceedings of the 2003 Eurographics Symposium on Geometry Processing (SPG-03)*, pages 156–165, june 2003.

[15] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *Proceedings of the International Conference on Shape Modeling and Applications (SMI-01)*, pages 154–166, May 2001.

[16] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, October 2002.

[17] M. Ranta, M. Inui, F. Kimura, and M. Mantyla. Cut and paste based modeling with boundary features. In *Proceedings of the 2nd Symposium on Solid Modeling and Applications (SM-93)*, pages 303–312, May 1993.

[18] H. Read. *A concise history of modern sculpture*. Thames & Hudson, 1964.

[19] J. Ron. *The early sculpture of Picasso 1901 - 1914*. Garland Pub, 1976.

[20] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. SKETCH: An interface for sketching 3D scenes. In *SIGGRAPH 96 Conference Proceedings*, pages 163–170, August 1996.