EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2006)
Thomas Stahovich and Mario Costa Sousa (Editors)

Parts, Image, and Sketch based 3D Modeling Method

Jun Murakawa, Ilmi Yoon, Tracie Hong[1] and Edward Lank[2]

[1]Computer Science Department, San Francisco State University, 1600 Holloway Ave., San Francisco, CA, 94132

[2]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada, N2L 3G1
junmura@gmail.com, yoon@cs.sfsu.edu, traciely@sfsu.edu, lank@cs.uwaterloo.ca

Abstract

Despite their many benefits, challenges exist in the creation of 3D models, particularly for individual not currently skilled with 3D modeling software. To address this, we explore the creation of 3D modeling software for non-domain experts that uses a hierarchical parts database of generic 3D models, and deforms models into specific related target objects using image guided 3D model morphing. A human-in-the-loop sketching interface supports image registration and constrains our geometrical transformation to support real time morphing of generic models into accurate representations of new objects for which users wish a 3D model. Applying the application to the study of insects in biology, we find that the application supports the creation of realistic 3D models, and that the application is of value to educators and researchers in entomology.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Line and Curve Generation, H5.m [Information interfaces and presentation] (e.g., HCI): Miscellaneous.

1. Introduction

3D models allow an enhanced representation of the structure of an object in various fields, including engineering (CAD), design (architecture), and art (sculpture). One factor inhibiting more extensive use of 3D models is the relatively complex and time-consuming process associated with generating these models. Typical 3D modeling software requires significant expertise. As well, the time-consuming, pedantic nature of 3D model construction makes these software applications ill-suited to exploratory processes by end users. The time investment required to create 3D models argues, instead, for their use at later, rather than earlier stages of object construction or analysis.

In our work, we explore the design of 3D modeling tools to support researchers and students in the biological sciences. In particular, our goal was to design 3D model creation tools that would be useful for biological researchers or biological students during the early stage of structure object examination. Our first target domain in biology is entomologists, and we explore the creation of 3D models of various insects for education and early analysis of physical characteristics. Current 3D modeling applications require a significant time investment to create 3D models, and users with a new insect specimen have few simple options for creating a 3D model of an insect they are studying.

Recent research in 3D modeling software seeks to address the shortcomings the modeling software in two ways. The first is the design of systems that allow simplified creation of 3D objects, based on techniques such as suggestive geometries and other informal drawing methods that are augmented with computational intelligence. Two challenges exist with this method of 3D model creation. First, systems that support the creation of 3D models of objects do allow users to create simple 3D models, but complex objects are more difficult to model, requiring extensive, painstaking drawing of physical structure. Consider using pen-based sketches and suggestive geometries to create a 3D model of an ant. A user would need to extensively sketch all parts of the ant, adjusting geometries as needed. While this is possible for simple mechanical systems, the complex surface deformations of living organisms make this process prohibitively time consuming, even with enhancements such as image-guided sketching [TBS04].

The second method for end-user 3D model specification allows users to construct new 3D objects by combining or altering pre-existing 3D models of real world objects. Altering existing 3D models shows more promise for the creation of 3D models in domains such as biology. Current techniques in the research literature do this by pasting together parts to create 3D models of chairs, for example [FKP04], or by editing existing meshes into new shapes using techniques like sketching to intelligently manipulate control vertices [NSA05].

This paper describes a 3D modeling tool based on semi-automated, image-guided 3D model deformation. Taking a complex 3D model that is similar to a target object a user wishes to model, we describe a process where images, superimposed on the model, serve as a guide for model deformation. Users use a pen to select parts of the 3D model based from a parts database of 3D models of objects in the target domain, select a portion of an image of an object of which they wish a 3D model using lasso based selection, and our geometrical transformation algorithms alter the meshes on the 3D model to correspond to the shape of the target image. Using our model deformation strategy, we give biologists the ability to easily create near perfect 3D models of target organisms. Biologists have identified this tool as useful for education and in early stages of biological research.

This paper is organized as follows. In Section 2, we describe related work. Section 3 presents our modeling method. Section 4 displays modeling results. Finally, we conclude by describing current contributions and future work.

2. Related Work

The creation of 3D models of objects is a time-consuming process, particularly for individuals unfamiliar with 3D modeling tools, or users attempting to re-create 3D models from complex perspective images. Authors have noted this shortcoming, and the opportunity that exists to make 3D modeling accessible to end users [IMT1999, TBS04, NSA05, ZHH96].

To design 3D modeling tools for end-users unskilled with complex 3D modeling software, researchers have explored two alternative approaches. The first uses a 2D sketch of 3D objects coupled with some form of computational intelligence to infer 3D geometries. The second is the manipulation of pre-existing 3D models, for example the combination of parts of models into a new composite model or the guided deformation of pre-existing models.

2.1. Sketch-Based 3D Modeling

Sketch-based 3D modeling can be further decomposed into three distinct approaches to the recognition of 3D structure from the sparse information conveyed by a 2D sketch. These approaches are sketch-based geometry generation, gesture-based 3D modeling, and predictive and suggestive interfaces.

The best-known sketch-based geometry generation system is Teddy [IMT99], which examines the contours created by 2D lines and constructs plausible 3D polygonal surfaces from the 2D line drawing. Users begin by drawing the simple contour of the object on the screen and the system constructs a candidate 3D shape for the 2D contour. The shape can be rotated and modified through the addition of additional strokes which define new 3D shapes, or by drawing a series of gestures to transform existing shapes, for example cut gestures, smoothing gestures, extrusion gestures, or painting and erasing gestures. Other researchers have extended this work [IH03, CSS05], but the basic nature of 3D model creation remains unchanged. Using knowledge of likely candidate shapes, a sketch defines a new 3D model. While this system does allow users to create simple 3D models, creating complex models is a much more time-consuming process, and may be impossible using these approximate 3D model generation systems.

Gesture-based 3D modeling, such as Zeleznik's *Sketch* system [ZHH96] are geared more toward near-expert users who want more intuitive 3D modeling tools. A pen-based system is used to sketch contours, and gestures augment the input by allowing users to manipulate and more fully refine 3D models. The basic gesture-based system allows users to create, scale, and orient objects in 3D environments using gestalt commands, and enhanced systems exist to support various domain experts such as engineers. While the system does allow effective 3D model creation, it places a significant burden on users to master a series of multistroke gestures to create precise 3D models.

Finally, predictive and suggestive systems suggest relevant geometries to the user using either a database of prior geometries. Tang et al. [TBS04] describe a 3D modeling that supports drawing directly in 3D perspective by displaying a 3D rectangle defining "faces" of a 3D model. The rectangle on which users draw can be rotated, and images for tracing can be placed on each face. Users can then trace images or draw freehand on a face, and the lines drawn are rendered in three dimensions. Geometrical information "suggests" good line placement, and switching views permits a user to adjust the three-dimensional position of the lines.

While the 3D sketching interface does permit line-drawn 3D objects to be created, two points merit note. First, for all but the simplest of outlines, the drawing and repositioning process is very time consuming. 3D sketching is inappropriate for complex 3D models, such as of biological specimens where many smooth contours shaded with appropriate texture maps must be generated. Second, the system requires a good awareness, on the part of the user, of the 3D model being constructed. As such, sketch-based 3D modeling is inappropriate for non-3D modeling experts.

2.2. 3D Model Transformation

The second approach to end-user creation of 3D models transforms pre-existing 3D models or combines pre-existing 3D models into new 3D models of objects. Two recent approaches to this are Nealen et al.'s detail-preserving mesh editing [NSA05] and Funkhouser et al.'s technique of creating 3D models by combining parts of pre-existing 3D models.

Mesh editing allows users to sketch on meshes to suggest edits to the mesh [NSA05]. Using discrete models of 3D meshes, contours can be moved while preserving much of the existing geometric surface detail. The user first selects a silhouette segment to be transformed, and draws a new candidate shape for the silhouette. Vertex normals on the silhouette segment are linearly interpolated based on the new candidate shape's position. The silhouette vertices are then mapped to new locations based on a transformation calculated from the overall movement defined by the new candidate shapes location.

In the modeling by example approach, a parts database is mined for 3D models of interest. Portions of these meshes can be segmented from the models and selectively added to other models. For example, using rough shapes drawn using a mouse to indicate position, arms from one 3D model can be glued onto a 3D model of an armless statue or different styles of chair arms can be combined into a new chair design.

2.3. Shortcomings of Existing Work

While many of the components to support non-expert creation of 3D models have been proposed, no one solution fully addresses the needs of the non-expert in the rapid creation of complex 3D models of real world objects. Sketching systems require the depiction of every line, and 3D Model transformation either requires painstaking alteration of 3D models or the need for an extensive parts database that contains all possible pre-existing 3D models that you might want to use.

In domains such as biology, researchers and students may examine new, un-modeled species containing structures that, while similar to pre-existing models of other species within a broader family of organisms, have unique characteristics not currently represented. Our goal is to provide domain experts with access to informal 3D modeling tools that allow the creation of near-perfect complex 3D representations rapidly.

3. Proposed Modeling Method

To provide interfaces for domain experts with little 3D modeling experience, we propose a modeling tool that supports interactions on 2D images to guide 3D model deformation. Users start with image(s) of their subject and choose a generic 3D model from the list of available ones for their particular domain of study.

The proposed modeling process is as below (Figure 1). A user first provides images of the subject that is going to be modeled and selects a generic 3D model from the list of models available. Ideally, the selected model is "similar" to the subject; for example, a 3D model of one ant species would be used to model a target ant species. Second, given the generic 3D model and images of the subject to be modeled by the user, the system displays the selected 3D model overlaid on the target images. The user then selects a part to be morphed. The hierarchical structure of the generic 3D model aids with this selection, as shown in figure 2, by providing a natural segmentation of the 3D meshes in the generic 3D model. Next, the user draws a stylus lasso that selects the corresponding part of the target subject on top of the image of the target subject. Finally, the system morphs the generic 3D model based on the silhouette of the part extracted from the selected component of the image. The user can repeat this process for each part until significant parts of the 3D model are deformed to the desired shapes in the subject of interest to the domain expert.

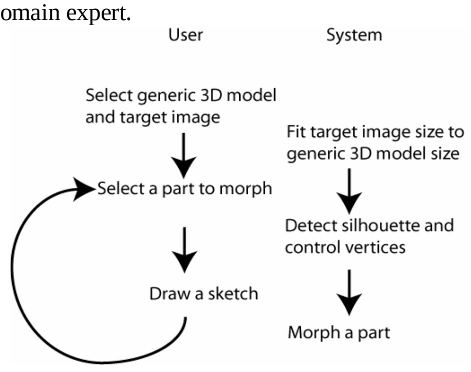

User
System
Select generic 3D model and target image
Fit target image size to generic 3D model size
Select a part to morph
Detect silhouette and control vertices
Draw a sketch
Morph a part

Figure 1: Overall Modeling Process

When the user selects the part from the 3D model, the system detects the silhouette of the 3D model component and identifies control vertices from the silhouette as source points for morphing. When the user later defines the silhouette of the corresponding part from the image using a pen-based selection gesture, the system computes sequences of points as a destination of the morphing. The details of each step in our modeling process are described in the following subsections.

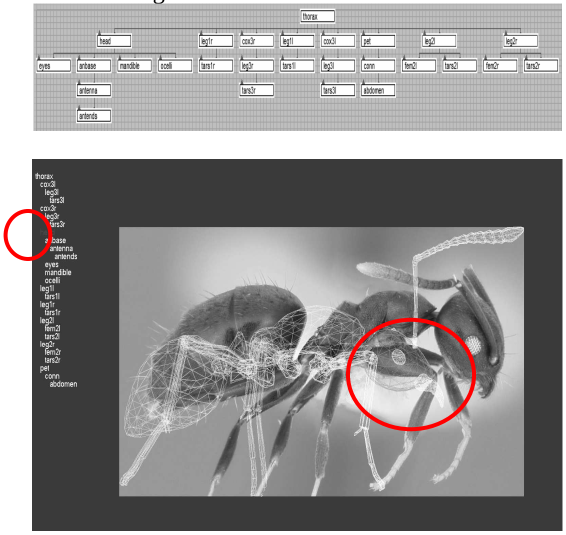


Figure 2: Tree Display of Hierarchical Structure of Parts on the left side (and the top) and the selected part (circled) is highlighted as red.

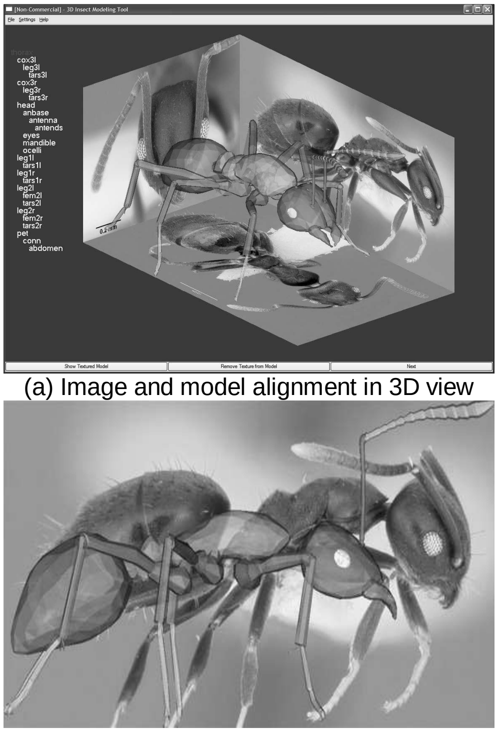
3.1. Parts Database

The idea of parts database is inspired by research in modeling by example [FKP04]. In our research, the generic parts database which categorizes 3D geometry by model structure was designed. Using this database, the system allows a user to search, browse and select body parts of interest. This parts database was inspired by biology education, where species taxonomy is based on the structure of physical parts of the body of a target species to be classified. Users with knowledge of how physical structure defines species classification can search and assemble body parts to create a generic model that can be easily morphed to the desired shape depicted by photos of a subject of interest.

3.2 Generic 3D Model and Target Image

Generic 3D models are used as the starting point for the modeling process. A hierarchy structure is defined for a category of insects. As one example, in ants the thorax is the root of a hierarchy, and a head, legs, and abdomen as the children of the Thorax part. The hierarchy structure is not mandatory for the modeling process but aids the process by allowing the user to select body parts efficiently.

In this first step, the user also provides target image(s). The target image can be a single view of a subject or multiple views of the same subject, preferably from various perspectives if such images are available. For example, the antweb [ANTWEB] provides a large number of images of different species of ants. If multiple images are available, the user can select images for the top, front, side or another random non-orthogonal view of the ant subject to be modeled. Then, the system displays the generic 3D model on top of the target images. Internally, this 3D rendering is done by the OpenGL 3D library. Images are also rendered as texture applied on large quads.



(a) Image and model alignment in 3D view

(b) Modeling view in 2D

Figure 3: Modeling View: Generic Ant Model rendered on Top of Target Image – User selects thorax part (red – half transparent part) to be morphed.

The simplest images to use for a user are orthogonal views, specifically the front, top, rear, and side views at perpendicular angles to the dominant axes of a species being modeled. Using a random, non-orthogonal photo does require some additional user-directed adjustment to accurately align the image with the generic 3D model.

The application interface that supports this step allows the user to browse target images and 3D models in multiple views. Basic navigation tools (zoom in/out, rotation, and 3D centering) are implemented for convenience. The application interface for model and image selection and alignment is shown in Figure 3a and 3b.

3.3 Part Selection

Live organisms consist of several body parts and parts are interconnected in varying orientations. The variation in orientation creates poses. Automatic recognition of every body part based on diverse poses from photos is challenging. Instead, our system asks the user to choose a body part from the 3D model and use a pen sketch to outline the corresponding part in images of the subject being modeled. This human-in-the-loop image registration task simplifies the morphing algorithm, specifically to support real time morphing of our 3D models. Due to the use of generic parts, the user need only morph significant body parts, leaving insignificant parts unaltered from the generic model. To help the user select a part of the 3D model to be modified, the list of part names is displayed on the left side of the modeling screen in a tree view that depicts the hierarchical structure of the parts, shown at figure 2.

The use of 3D model deformation is not unique to our work, but has been recognized as an effective way to draw, select, and manipulate objects in a sketch guided two-dimensional interface. Pen-based input devices such as Tablet PCs or data tablets support sketching operations by leveraging familiarity with pen and paper to create precise outlines of relevant structure. Initially, the user sketch is a set of points on the screen. To use the sketch to deform the 3D geometry, the screen coordinates are converted into 3D world coordinates using un-projection.

3.4. Silhouette and Control Vertices

The silhouette is the boundary of the 3D object projected onto the 2D screen. For the morph algorithm to deform the 3D geometry using the user drawn 2D outline of a target shape, vertices on the silhouette are used as control vertices for the 3D geometry deformation. We don’t use a popular silhouette edge detection algorithm that requires connectivity information [BS03]. Instead of building the connectivity information, we detect silhouette vertices directly from the dot product of a normal vector and a camera vector. As visualized in figure 4, the red color shows small values of the dot product (near 0.0), and the green color shows large values of the dot product (close to 1.0). Those (red) vertices with smaller dot products are orthogonal to the camera view and are selected as control vertices for morphing.

3.5. Correlation between Control Vertices and Sketch Points

After the user sketch and control vertices are obtained, the center point of the shape defined by the control vertices and the center of the shape defined by the sketch points are aligned using global translation. The system then finds a corresponding sketch point for each control vertex, and this correspondence is used to compute movement vectors.

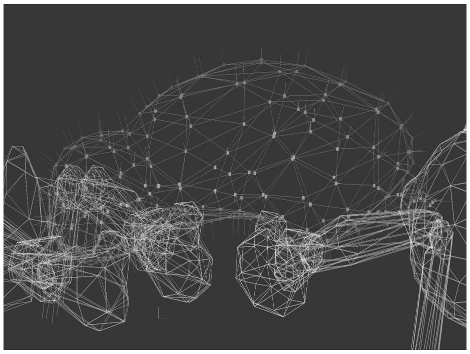


Figure 4: Dot Products of Normal Vectors and Camera Vector

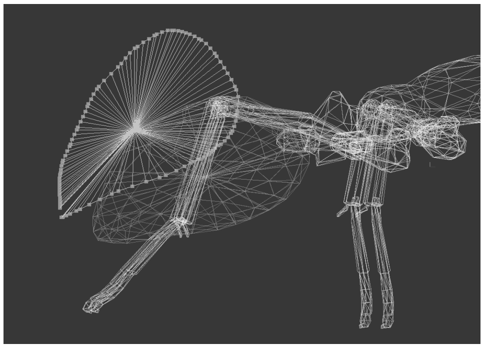


Figure 5: Control Vectors and Sketch Vectors

For each control vertex, the algorithm identifies the point in the user gesture on the image which has the closest angle to the control vertex. This is done by computing the dot product of two vectors: the control vector (a vector from the centre of the 3D shape to the control vertex) and the sketch vector (a vector from the centre of the sketched outline to the point on the gesture). The sketch vertex which has maximum dot product value with the specific control vertex is selected as the corresponding sketch point. Figure 5 visualizes control vectors and sketch vectors before global translation.

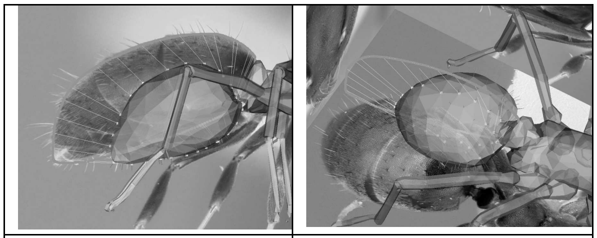
3.6. Movement Vectors

Once a corresponding sketch point for the control vertex is found, movement vectors are computed. A movement vector is a vector originating at the control vertex and directed to the corresponding sketch point. However, control vertices are not necessarily contained in the plane of the sketch points. Therefore, movement vectors have to be computed as if movements are parallel with the sketch plane. We compute the movement vector as the sum of local movement and global translation. The local movement is computed by multiplying the sketch vector by the ratio between the length of the sketch vector and control vector, and the global translation is a vector from the center of the 3D geometry to the center of the sketch.

$$\text{Movement Vector}: MV = \frac{|SV| - |CV|}{|SV|} SV + T$$

CV is a control vector
SV is the corresponding sketch vector
T is the global translation vector

Figure 6 shows an example of movement vectors. Figure 6a shows the movement vectors in side view, and figure 6b shows the same movement vectors in perspective view. The movement vectors are in 3D space, but point to the user drawn 2D sketch points. The movement direction is restricted to the sketch plane.



(a) Side view

(b) Perspective View

Figure 6: User Drawn Sketch, Control Vertices, and Movement Vectors.

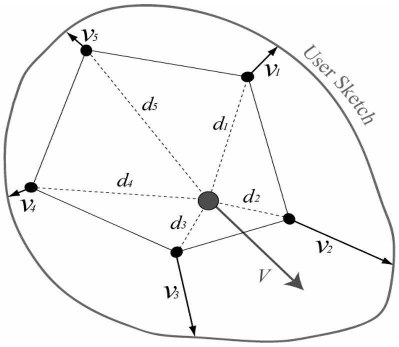


Figure 7: Vertex, Movement Vectors and Displacement Vector

After the movement vectors of all control vertices are computed, the movement is distributed throughout all vertices in the 3D mesh. The displacement vector of each vertex in the 3D mesh is computed as a weighted sum of movement vectors. Figure 7 shows geometrical relationships between movement vectors and the displacement vector. To morph an example geometry with five vertices to the user drawn sketch shown as the green curve, the displacement vector *V* for the red colored vertex in the 3D mesh is computed as the weighted sum of the movement vectors $v_{1,2,\ldots,5}$, according to their distances $d_{1,2,\ldots,5}$.

To examine the effects of weight functions on the morphing algorithm, both parabolic and linear weight functions were tested. The parabolic weight function uses the reciprocal of relative distance from a vertex to the control vertex as the weight, and the linear weight function uses the relative distance from the vertex to the control vertex as the weight. Mathematically, the displacement vector of a vertex in the mesh is defined by the following equations. Here $\boldsymbol{v}_\mathbf{i}$ is a movement vector for each control vertex *i*, ***V*** is displacement vector for each vertex and computed as a weighted sum of movement vectors:

$$V = \sum_{i=0}^{n} \frac{w_i}{W} v_i .$$

where the weight $\boldsymbol{w_i}$ for a control vertex *i* is defined using attenuation factor p:

$$w_i = \left(\frac{D - d_i}{d_i} \right)^p \qquad \text{- Parabolic weight function}$$

and the total weight ***W*** is sum of weights across all vertices:

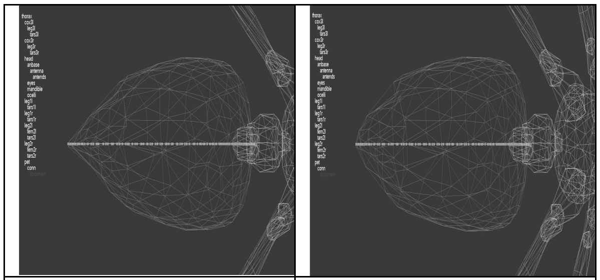
$$W = \sum_{i=0}^{n} w_i ,$$

where ***D*** is the longest distance in all control vertices:

$$D = \underset{i=0,1,\ldots,n}{Max}(d_i)$$

When applied on a 3D mesh, parabolic and linear weight functions result in different shapes. Figure 8 shows the different morphed shapes of an insect abdomen using parabolic and linear weight functions. The part is morphed in the side view and observed in the top view.

The parabolic weight function places higher weight on control vertices close to the vertex, and lower weight on more distant control vertices. In this way, it preserves the shape of the original mesh, especially in the area close to control vertices. On the other hand, the linear weight



(a) Linear

(b) Parabolic with P = 2 preserves original shape

Figure 8: Movement Distribution

function takes a larger number of control vertices into account because of its gentle fall off, thus resulting in lower geometrical noise.

Since both functions have advantages in certain types of geometry, both are left as options. The parabolic function is used as the default weight function because it shows better morphing results in the general case. For either the parabolic or linear weight function, the attenuation factor, p, also affects the result of the morph. The weight function shows a sharper attenuation curve for higher attenuation factors, p, and the differences in restuls are shown at figure 8. Various attenuation factor values were tested and p=2 appears to balance speedy convergence with low noise in our deformation of generic models of biological organisms.

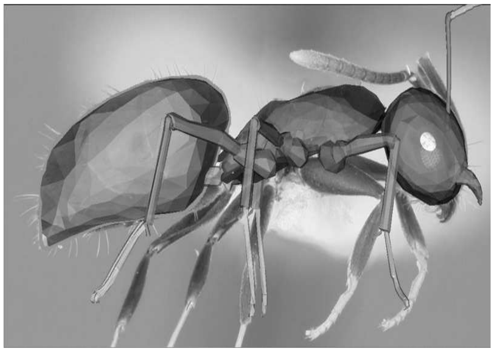
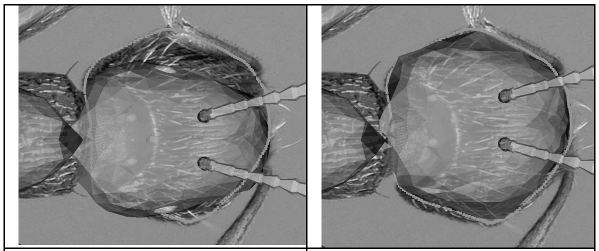


Figure 9: **Morphed Ant Model**

In figure 9, the silhouette of the 3D part is precisely matched to the bounds drawn by the user's sketch, while the topology and shape of the original geometry is preserved. The morph algorithm also supports multiple views. Figure 10 shows the user sketch and morph results applied in the side view first and then in the top view. Note the head part has already been morphed in the side view prior to the top view morph; the user can apply morphing in multiple views in any order they prefer, and continue working on significant parts until the desired shape is obtained. As well, as shown in these figures the topology of the 3D geometry is well preserved after multiple morphings.



(a) Sketch drawn in Top View

(b) Morphed in top view

Figure 10: Adding top view image for refining the shape

3.7. Texture Extraction

Texture is an important attribute of the 3D model for realistic appearance. We utilize the symmetry of the insect model and also utilize view-dependent texture blending to use textures from multiple view images. These textures are applied to the new 3D model to provide a more realistic depiction of the subject organism.

4. Results

4.1. Modeling Algorithms

To model insects, we used 13 insect models from the virtual insect web site [VRI]. The original VRML format was converted to 3DS format using 3D Studio Max [3DS], and loaded into the system. For the ant example model which is used throughout this research, a hierarchy structure is also configured by 3D Studio Max as follows.

In only a few morphing steps by a user, the shape of each body part in a 3D insect model is matched to the target images. Textures are extracted from multiple views and blended smoothly. A new ant model is created from the generic ant model as shown in figure 11.

4.2. Supporting Biological Research

The 3D modeling application has been well-received by entomologists. There are two particular ways that entomologists feel the modeling tool will aid them in their work.

First, entomologists view the application as a high valued aid to teaching. When teaching organism classification, an appreciation of physical structures, and how the physical structures of an organism feed into its taxonomy must be communicated to students. The 3D modeling tool allows students to perceive the structural similarities and differences clearly, and provides a simple morphing algorithm to convert a generic 3D model into a model of a specific insect under examination. While 2D photos can convey similar information, the realism of a 3D perspective view will, in the view of biological educators, more clearly depict for students the structural similarities and differences of various categories (phyla, classes, order, families, and genera) of species.

In research, too, it is often less interesting to examine 2D representations of structure, whereas 3D can more compellingly allow communication and discussion during early biological research into new species identification. While more information is not present, the depiction of the information spurs more interest and is more easily described by highlighting areas and structures on a 3D model.

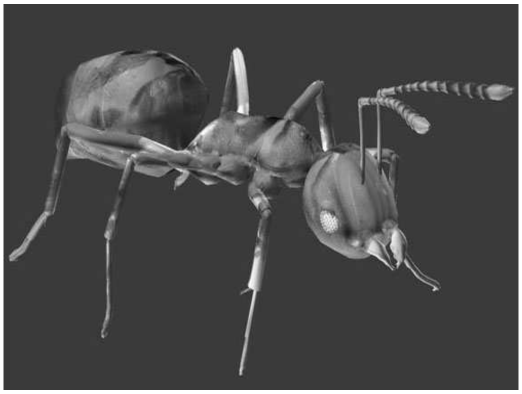


Figure 11: Final Ant Model

5. Conclusion

In this paper, we propose a new parts, image, and model morphing application to support 3D model generation by non-3D modeling experts. The goal is to make accessible 3D models to those individuals who can make use of models without requiring them to master the intricacies of 3D modeling software. Our application has proven effective for the generation of 3D models of biological organisms, specifically focused on insect modeling.

Biology is only one area of 3D modeling, but is undoubtedly the most suited to our 3D modeling application. While engineers and artists need access to 3D models, it is true that our system has disadvantages when applied to these fields. First, engineers are currently comfortable with CAD software, multiple images, and generated, accurate 3D representations of objects. Generating near approximate, one-off models of pre-existing objects is of less interest in this domain. Similarly sculptural artists, another demographic interested in 3D models, generally seek new, novel objects to visualize, depict and create, where our tool seeks to generalize from similar structures.

Despite its focus on biological research, our application has been successful in providing biological researchers and educators with an alternative 3D modeling platform for the rapid generation of accurate 3D models of organisms. We continue to examine modifications to our algorithms, and additional application domains for these techniques.

Acknowledgements

This research was funded by the National Science Foundation, grants DBI-0234980 and IIS-0448540, and by the Natural Science and Engineering Research Council of Canada.

References

[ANTWEB] Antweb: Ants of the world. http://antweb.org.

[BV99] Blanz V., Vetter T.: A Morphable Model for the Synthesis of 3D Faces. SIGGRAPH 99 Conference Proceedings.

[BS03] Stefan Brabec and Hans-Peter Seidel: Shadow Volumes on Programmable Graphics Hardware, Eurographics 2003 (Computer Graphics Forum)

[CSS05] Cherlin J. J., Samavati F., Sousa M. C., Jorge J. A.: Sketch-based modeling with few strokes. Spring Conference on Computer Graphics 05.

[FKP04] Funkhouser T., Kazhdan M., Shilane P., Min P., Kiefer W., Tal A., Rusinkiewicz S., Dobkin D.: Modeling by Example. SIGGRAPH 04, p 652-663.

[IH03] Igarashi T., Hughes J. F.: Smooth meshes for sketch-based freeform modeling. SIGGRAPH 03.

[IMT99] Igarashi T., Matsuoka S., Tanaka H.: Teddy: A Sketching Interface for 3D Freeform Design. SIGGRAPH 99.

[NSA05] Nealen A., Sorkine O., Alexa M., Cohen-Or D.: A sketch-based interface for detail-preserving mesh editing. SIGGRAPH 05.

[SYN05] Sebe I. O., You S., Neumann U.: Rapid Part-Based 3D Modeling. ACM Symposium on Virtual Reality Software and Technology (VRST) 05.

[VRI99] Sharov A.: Virtual Insect. http://www.ento.vt.edu/~sharov/3d/virtual.html.

[TBS04] Tsang S., Balakrishnan R., Singh K., Ranjan A.: A suggestive interface for image guided 3D sketching. SIGCHI 04.

[YSP05] Yang C., Sharon D., Panne M.: Sketch-based Modeling of Parameterized Objects. SIGGRAPH 05, Sketch Session.

[ZHH96] Zeleznik R. C., Herndon K. P., Hughes J. F.: SKETCH: An Interface for Sketching 3D Scenes. SIGGRAPH 96. p. 163-170.