Evolutionary Morphing

David F. Wiley* Nina Amenta* Dan A. Alcantara* Deboshmita Ghosh* Yong J. Kil*
Eric Delson‡ Will Harcourt-Smith‡ F. James Rohlf§ Katherine St. John‡ Bernd Hamann*
* Institute for Data Analysis and Visualization and Department of Computer Science,
University of California, Davis
‡American Museum of Natural History and Lehman College of the City University of New York
§Department of Ecology and Evolution, State University of New York, Stony Brook

Abstract

We introduce a technique to visualize the gradual evolutionary change of the shapes of living things as a morph between known three-dimensional shapes. Given geometric computer models of anatomical shapes for some collection of specimens - here the skulls of the some of the extant members of a family of monkeys - an evolutionary tree for the group implies a hypothesis about the way in which the shape changed through time. We use a statistical model which expresses the value of some continuous variable at an internal point in the tree as a weighted average of the values at the leaves. The framework of *geometric morphometrics* can then be used to define a shape-space, based on the correspondences of landmark points on the surfaces, within which these weighted averages can be realized as actual surfaces.

Our software provides tools for performing and visualizing such an analysis in three dimensions. Beginning with laser range scans of crania, we use our landmark editor to interactively place landmark points on the surface. We use these to compute a "tree-morph" that smoothly interpolates the shapes across the tree. Each intermediate shape in the morph is a linear combination of all of the input surfaces. We create a surface model for an intermediate shape by warping all the input meshes towards the correct shape and then blending them together. To do the blending, we compute a weighted average of their associated trivariate distance functions and then extract a surface from the resulting function. We implement this idea using the squared distance function, rather than the usual signed distance function, in a novel way.

CR Categories: I.4.10 [Image Processing and Computer Vision]: Image Representation—Morphological; I.3.8 [Computer Graphics]: Applications

Keywords: morphometrics, morphing, surface blending, merging, warping, distance fields, extremal surface

*e-mail: wiley@cs.ucdavis.edu, amenta@cs.ucdavis.edu, eedlc@cunyvm.cuny.edu,willhs@amnh.org,rohlf@life.bio.sunysb.edu

1 Introduction

Darwin's theory of evolution was originally applied using morphology - discrete qualitative features such as number of toes, and also quantitative shape differences, such as elongation of a limb - to place species within the tree of life. While genomic sequence data is now the basis of most phylogenies (evolutionary trees), morphology continues to be an essential part of evolutionary biology. One important reason is that the morphology of fossils, rather than comparisons between genomic data, provides our only direct evidence for extinct species.

For instance, the shape of the skull is very important in the study of human evolution, and that of our primate relations. Quantitative differences, such as shape of the brow ridge, are essential in defining the criteria for comparison between skulls. The idea that quantitative shape differences can be analyzed in terms of the transformations required to "morph" one shape into another goes back at least to d'Arcy Thompson's classic 1917 book *On Growth and Form* [24], from which Figure 1 is taken.

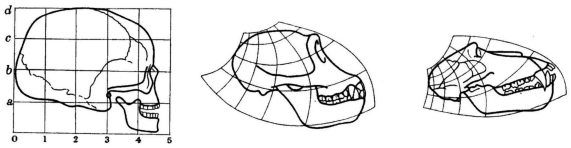

Figure 1: Spatial relationship between human, chimpanzee, and baboon skull, as envisioned by d'Arcy Thompson in 1917. The overall shape is recognizably similar, and the transformation between them describes the shape difference. In modern morphometric studies, the statistical analysis of the transformation is based on a matrix of selected landmark points from the surfaces, while warps of the embedding space, which are illustrated here, are more often used for visualization of the results.

The statistical analysis of geometric shape transformations is the program of *geometric morphometrics*. In addition to evolutionary biology, morphometric techniques are used widely in developmental biology, medical image analysis, and other areas. Morphometrics defines "shape spaces" based on sets of homologous sets of landmark points on the input objects. The spaces in which statistical analysis is generally done are linear spaces spanning the input shapes, so that the interpolating shapes in the space, such as those we use here, are linear combinations of the input shapes.

Using morphometrics, we have implemented a three-dimensional "tree-morph" visualizing the evolutionary changes implied by a given evolutionary tree. Surface meshes captured by a laser range scanner from cranial specimens for extant (living) species appear at the leaves. The interior nodes and interior points of the edges of the evolutionary tree correspond to hypothetical ancestor species. We realize these by computing synthetic surface meshes for the shapes at the internal nodes and at a dense set of points sampled along the tree edges. We then visualize the morph by displaying the precomputed meshes interactively, by sliding a cursor along tree edges, or as an animation.

This specific project was chosen to drive the development of computer tools for three-dimensional morphometric analysis and visualization. Our aim is to use visualization and interactive tools to support the kind of morphometric analysis that is currently done in paleontology and other biological disciplines, producing images which reflect the users' existing theories about shape transformation and which are therefore perceived as credible and relevant to the science. We do not propose in this initial project to change the current analytic techniques. But we do expect that as better tools make it possible to handle large amounts of three-dimensional data a greater emphasis on automation will be inevitable, and that the software framework we have developed here will make it easier to innovate.

We faced a number of design decisions and challenges in the project. Morphometric techniques are based on user-defined sets of landmark points, which are assumed to be exactly homologous. Recent work in graphics has tended to instead to emphasize the automatic determination of homology. This would be appropriate to the extent that it reliably corresponds to biological homology, but this is a terrific research question rather than an accepted technique to be applied. Our *landmark editor* supports existing practice by facilitating the placement of large sets of landmarks, using automatic placement only very conservatively. This has had immediate and obvious impact.

A second issue was producing the intermediate surfaces which linearly interpolate the input shapes, as defined in morphometric analyses. The multiple-alignment and interpolation procedures we use to handle the landmark point sets, while standard in morphometrics, differ from those common in computer graphics. They are designed to minimize the error induced by forcing the shapes into a linear space. We then needed to develop a procedure to produce linearly interpolated surfaces as well as linearly interpolated sets of landmark points. This is a different problem from standard morphing or blending. Besides having several inputs instead of just two (not all methods generalize in an intuitive way), we wanted to preserve properties of linear interpolation. In particular, we wanted the operation to be commutative, meaning that, for instance, the surface one third of the way from A to B ($2/3A + 1/3B$) should be the same as the surface two-thirds of the way from B to A ($1/3B + 2/3A$); this is not true for many existing two-input morphing methods. This motivated our choice of a morphing method based on linear interpolation of implicit functions.

A third challenge was handling the somewhat messy scanned input data, which does not even approximate a closed manifold surface; there are large holes and only one side of the solid is captured in most areas, while in others both sides are very close together or even self-intersecting. In the implict function interpolations, we use the squared distance function to the surface rather than the more usual signed distance function, from which we extract an *extremal surface* (defined in Section 2.5). This approach does not require closed or oriented surfaces as input nor do we require any preprocessing of the laser range scan surfaces.

1.1 Geometric Morphometrics

Geometric morphometrics is a branch of biostatistics dealing with the analysis of shape [4, 27, 1]. Scientists need to be able to define and analyze statistically significant variables expressing biological shape. This task is difficult because the choice of what to measure and analyze affects the results. Rather than measuring specific distances, angles, and so on, the approach used in geometric mophometrics is to choose a discrete set of K homologous (or "corresponding") *landmark points* $\mathbf{L}^i = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_K\}, 1 \leq i \leq N$ on all N input object surfaces. The representation of the shape by its set of landmark points subsumes measurements of specific distances between landmarks, angles produced by three landmarks, etc. A dense enough landmark point set provides an adequately sampled representation of the shape. Statistical analysis of landmark point sets provides a method for making assertions such as "This fossil cranium resembles a macaque rather than a baboon," more precise and quantifiable.

The sum-squared distance between two sets of landmark points is

$$\mathscr{D}(\mathbf{L}^i, \mathbf{L}^j) = \sqrt{\sum_{n=1}^{K} \left(\mathbf{L}^i_n - \mathbf{L}^j_n\right)^2} \tag{1}$$

Procrustes distance between them is the minimum of $\mathscr{D}(\mathbf{L}^i, \mathbf{L}^j)$ over all rotations, scales, and translations of $\mathbf{L}^j$ (with the scale of $\mathbf{L}^i$ normalized). Procrustes distance imposes a geometry on the space of landmark configurations, forming a non-linear *shape space*. Unfortunately, the pairwise alignments do not produce a multiple mutual alignment of all the landmark sets; if we align $\mathbf{L}^a$ to $\mathbf{L}^b$ and $\mathbf{L}^c$ to $\mathbf{L}^b$, we find that $\mathbf{L}^a$ is not aligned to $\mathbf{L}^c$. The common practice is to compute an averaged consensus landmark point set and align all of the landmark point sets to this consensus configuration. Using a process called *General Procrustes Alignment* (GPA), a consensus landmark set is chosen to minimize the total squared difference between the aligned input landmark sets and the consensus configuration [9]. Statistical analysis can then be performed on the data matrix formed by the aligned landmark sets. This is treated as a linear space, in which standard statistical techniques can be applied directly. This linear space is an approximation to the non-linear shape space defined by the Procrustes distances, and the choice of a good consensus configuration is important to reduce the error caused by forcing the data into a linear space.

After GPA we can linearly interpolate our set of input configurations, producing intermediate configurations which also lie in the approximating linear space. All of our intermediate landmark sets in the tree-morph belong to such an interpolating linear space, with the weights for the linear combinations determined as described in Section 2.2.

It is common practice in morphometrics to visualize an averaged configuration by warping one of the input surfaces so that its landmarks coincide with the averaged configuration using a *thin-plate spline* (TPS) warp. The TPS is favored because the resulting surface interpolates the landmarks (which are the "real" data being visualized), because it is the smoothest such warp, minimizing the bending energy, which is related to the curvature, and because it is fairly straightforward to compute.

1.2 Application to Primate Evolution

We use this well-accepted morphometric framework to define the biologically "correct" interpolation of the skull shapes for five species of Old World monkeys. The Old World monkeys evolved

in the same time and place as early humans, making them a particularly interesting group to study. There are many extinct species of Old World monkeys, known from fossils, so that there is a lot of interesting data related to their evolutionary history. Yet this history, defined as the exact shape of the evolutionary tree, continues to be a matter of controversy.

We have used a laser range scanner to capture three-dimensional shapes of the crania of many species of Old World monkeys, both extant and fossil, as part of a larger effort at the American Museum of Natural History to develop a database of three-dimensional primate morphology. Here, we use five of these surface models to develop and apply a method for visualizing morphometric estimates of skull-shape variation that is relevant to the evolution of the group. We selected five extant species sampling both subfamilies of Old World monkeys, and use a best-estimate for the evolutionary tree of the five species, derived from DNA sequence data, which is only available for the extant species. We then visualize what the sequence-based tree implies about the morphology of ancient monkeys by interpolating the cranial shapes across the fixed tree. Figure 2 shows the tree.

Visualizations of the intermediates (the hypothetical species at any interior point of the tree) are interesting in at least two ways. Scientists want to answer questions like, "Are the skull shapes predicted by this model biologically plausible?" and "Where would this known fossil fit into the tree we hypothesize from genomic data?" The visualization of the subset of the skull shape-space defined by the tree helps to answer both kinds of questions.

1.3 Technical Overview

Our goal is to produce a three-dimensional tree-morph visualizing the evolutionary hypothesis presented by a specific evolutionary tree using as input surface models captured by a laser range scanner and the transformations specified by the morphometric model. The output of our procedure is a set of polygonal surface models, each one representing an intermediate shape corresponding to an interior point of the tree. Each of these intermediate models is a weighted average of the input models; they differ only in the choice of the weights. We developed a weighted average blending procedure, applicable to rigid objects. It is not intended to handle inputs in which the conformation varies as well as the shape (such as an arm bending at the elbow). Here is a brief description of each step; more details are provided in the following sections.

1. **Landmark specification:** The user interactively places landmark points at biologically meaningful locations providing homologous points on each of the input specimens.

2. **Alignment and target computation:** For each set of weights, we produce a weighted average target configuration for the landmark points, and aligne the input landmark sets to the target configuration, using GPA.

3. **Warp:** We compute a TPS warp taking the landmark points on each input surface to the target configuration and use it to warp the entire input surface.

4. **Blend:** We compute a distance function for each warped surface and take their weighted average. Extracting an extremal surface from this function produces the output surface.

The early steps follow the conventional course of a geometric morphometrics analysis, which is the "gold standard" for the scientists in terms of how they model shape change. We relay on user-defined landmark points, which we accept as truly homologous. We focus on making it easy for the user to define many landmarks, providing more data for the landmark-based statistical model. We use the GPA alignment procedure followed in morphometric analysis. Our warping step uses the TPS warp, which is well-accepted for the reasons discussed above. The final blending step meets two objectives. First, it carries through the principle of constructing the intermediate shapes of the morph as a linear combination of the input shapes, by representing the shapes as trivariate functions for which linear combinations are unambiguously well-defined. Second, our choice of trivariate function works well for the raw captured data meshes, which are not manifold and have holes, and which contain very thin shell-like regions and occasional self-intersections which are difficult to morph.

1.4 Related Work

Existing geometric morphometric software has mainly focused on the alignment and multivariate statistical analysis of specimens, with less emphasis on either the landmark placement user interface or on visualization. *Morpheus* [22], *morphologika* [18], and the *TPS* suite of programs [19] are the packages most widely-used by morphologists.

Placing landmark points on 3D specimens for morphometric analysis is generally done using 3D contact digitizers on the actual objects where the collected points stored in a spreadsheet. This is extremely tedious, so that landmark point sets consisting of tens of points are typical. For virtual 3D images of specimens, such as laser range surfaces or CT scans (ideal when the actual specimen cannot be obtained firsthand), there are generic software packages can be used, but these programs are not specialized for landmark placement, so the process remains quite cumbersome.

The interesting visualization problem of morphing primate skull shapes across an evolutionary tree was first approached by Delson et al. [8] using the three-dimensional analog of the practice from morphometrics mentioned above, in which the transformation from one shape to another determined by the landmark points is visualized by warping one of the input surface models. This approach has the drawback that the visualization of an intermediate produced by warping one input surface is not the same as the visualization produced by warping another instead. Our work improves on this approach in that we produce a single surface for each intermediate that represents the desired proportions of the input shapes. Also, since we can generate many more landmarks, we achieve a better representation of the shape and its variation in the resulting model.

When the shapes are captured by computed tomography (CT) rather than laser range scans, the trivariate density functions for the different specimens can be blended, after warping to align significant features. This idea has been applied to visualizing the evolution of toads by Hodges et al. [10]. The problem of merging similar surfaces is replaced, in this case, with the problem of isosurfacing as the function is averaged across time, which is also non-trivial.

Subsol et al. [23] produced an interesting morph between a CT scan of a modern human skull and that of a fossil humaniod. Their goal was to compare the two shapes using a deformation, and to demonstrate some possible applications of three-dimensional geometry processing in paleontology. They establish correspondences using automatically selected crest lines; the crest lines are noisy, the correspondences between them are approximate, and they fail to cover many areas of the skull such as the brain case. So Subsol et al. used them with a non-interpolating warp including a regularization component. This is a demonstration of the potential of possible automatic techniques, while we concentrate on the facilitation and

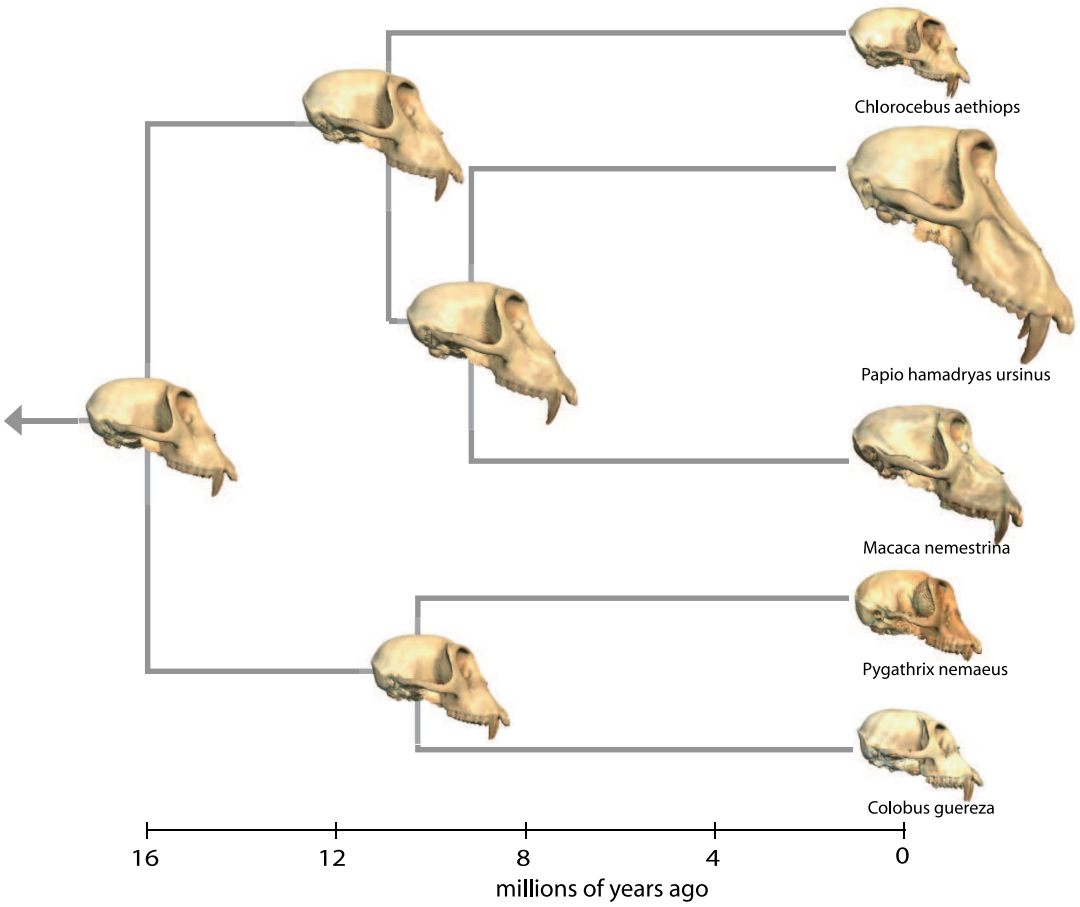

Figure 2: The input surface meshes, from laser range scans of the crania of existing monkey species, are shown on the right-hand side at the leaves of this tree. The surface meshes at the internal nodes, placed at the estimated dates at which the species are hypothesized to have diverged, represent the skull shapes of the hypothetical ancestors as computed using our system.

visualization of existing techniques which are actually used in statistical shape analysis.

We also draw on methods known in computer graphics and visualization. We were inspired by one particularly relevant project [2], in which a collection of full-body scans of humans was aligned to a closed synthetic "base mesh." The base mesh could then be warped to resemble any of the inputs, or a linear combinations of the inputs. This method produces a "space of human body shapes" useful in computer graphics, for instance for generating crowds of digital extras. In morphometrics, there is a strong emphasis on producing results that are derived from the data rather than introduced for computational convenience, so we wanted to avoid the synthetic base mesh; also, we had no appropriate mesh to use.

Instead, we use a warp-and-merge paradigm to produce the intermediate surface models. An early example of this paradigm is due to Lerios, Garfinkle and Levoy [12]. Their system included a user interface which allowed user to establish correspondences between curves and regions on the input models, similar to features of our landmark editor. The morphing method which formed their back end was more focused on efficiency and less on the accuracy of the intermediate shapes than some later approaches, including ours, and their method for color blending is somewhat similar to ours. Our morphing method is most closely related to that of Levin, Cohen-Or and Solomovici [7] (and see also [25]). Their method uses the TPS to warp the surfaces so that they resemble each other closely. This nicely coincides with the common practice in morphometrics, where the TPS is favored. The surfaces are then merged by converting each input surface into a signed distance function defined over a finite three-dimensional domain in the target space, taking a weighted average of the functions, and extracting the zero-set of the resulting function. This is appealing in our application since we can take a weighted average of several input functions in a straightforward way. Their method works well for closed manifold input surfaces, and reasonably well on our messy inputs except where there are self-intersecting surfaces or oppositely-oriented surfaces passing very close to each other. We developed an alternative method based on averaging the squared distance function, which produces a single-sheet output in such areas, with somewhat better results.

Other morphing paradigms, which are better in the traditional graphics context where a surface A is morphed into another surface B, are not appropriate in our application. For example, the attractive level-set method of Breen and Whitaker [6] produces a morph by moving every point of surface A with velocities controlled by the distance field of B. We do not see a straightforward way in which this can be modified to produce a linear combination of several input surfaces. Also, the intermediate shape one-third of the way from A and B in the level-set morph is different from the intermediate shape two-thirds of the way from B to A, so it cannot reasonably be interpreted as a linear interpolation.

2 System Details

2.1 Landmark Specification

An essential part of the project was developing the interactive landmark editor. A basic, but important, feature is that the homology between the landmarks on different input surfaces is shown explicitly; with conventional methods, the user had to imply the homology by carefully placing landmarks in a specific order. In the landmark editor, two surface meshes are shown at the same time in the main window. Figure 3 shows a screenshot. A dialog window shows the correspondences between pairs of landmark points; the windows are linked, so that when editing a correspondence in the dialog window the selected points on the surface meshes are highlighted.

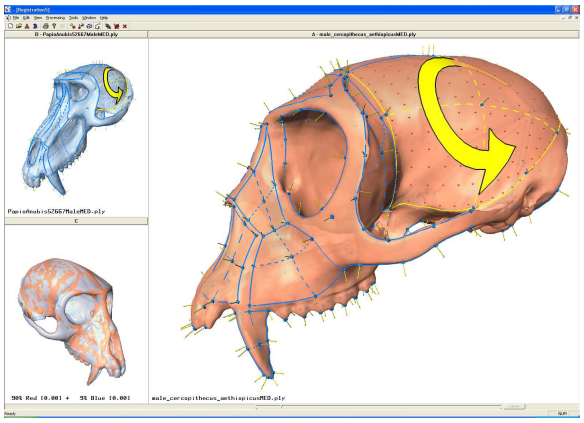

Figure 3: Screen capture of our landmark editor. Two input meshes are shown in the large pane and upper-left pane while the two warped models are overlayed in the lower-left pane. The yellow arrow indicates the selected patch orientation.

Points can of course be added, deleted and adjusted in any order. We show the surface normal as well as the point itself as the user adjusts the landmark, which helps to place it exactly, especially on high-curvature features.

Not all shape differences can be captured by single points. To capture curvature of an eye socket or the area of the brain case, we want to establish correspondences between curves and surface patches as well as points. Since morphometrics is based on the analysis of matrices of homologous points, in morphometrics this is done by distributing points, called *semi-landmarks*, across such features. We implement this by allowing the user to place the control points of Bézier curves or patches on the surface. The system automatically generates a user-controlled number of semi-landmark points across the curve or patch, in a fixed order derived from the order of the control points. The semi-landmarks are then projected onto the surface. The user establishes correspondences between the control points two curves or two patches, implicitly defining the correspondences between all pairs of semi-landmarks on the primative. The orientation of curves and the orientation and rotation of patches is shown with arrows, since it is easy to get these swapped (by placing the control points in different orders on the two surfaces), and the user can re-orient a patch or curve to correct a mismatch without having to move the points.

Bookstein [5] introduced a method for optimizing the positions of semi-landmarks on a curve or surface, minimizing the bending energy of the induced TPS warps. We have implemented this method, and it seems to have minimal effect on our semi-landmarks, which are generally well-spaced to begin with.

We have also implemented a method that transfers landmark primitives from one surface to another semi-automatically. The user places at least four landmarks to produce a crude warp, which is then used to transfer the rest of the landmarks; the user then has to adjust their positions, but just transferring the overall configuration greatly simplifies the experience and reduces errors. We can also export the landmark points, which allows existing morphometric packages to use them.

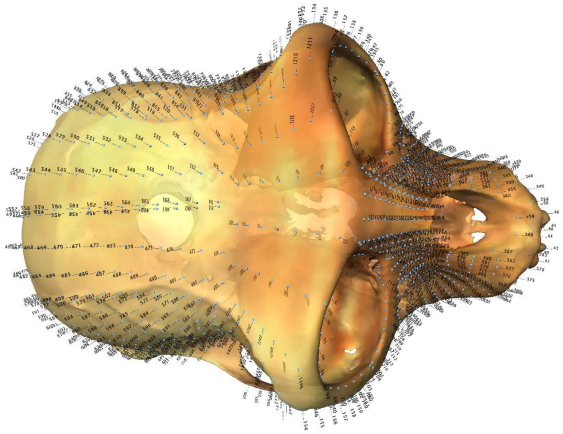

Figure 4: A full set of 853 landmarks placed on one of the scanned crania. These were created using 45 are single points, 32 curves and 9 surface patches.

Using this interface, it is easy to create large sets of correctly corresponding landmark and semi-landmark points (Figure 4). While placing these was indeed tedious (it took our novice users about three hours per skull), it would have been completely infeasible using previous methods

The landmark editor is currently being used by the primatologists on our team for a separate research project investigating congruence between joint surfaces in the primate skeleton, in which a grid of points are placed on the opposing joint surfaces. Landmarks have been collected on laser range scan surfaces of over 80 primate lower limb-bone specimens, and results are being analyzed. The software is greatly facilitating an otherwise lengthy and complex process.

2.2 Weights

Each internal point of the tree corresponds to a set of consensus landmark points, which is a weighted average of the landmark points at the leaves. The weights are determined using a principle known in evolutionary biology as *squared-change parsimony*: the integral of the squared change of a variable v (in this case, a single landmark point coordinate) over the tree is minimized, within the constraints imposed by the values of v at the leaves. This principle is sometimes used to estimate the structure of the evolutionary tree from the values at the leaves [26]. Here instead we have the much easier problem in which the structure of the tree, including the lengths of the branches, is fixed, and we just want to compute the values of the variables at internal points in the tree.

A generalized least-square method for this problem was introduced by Matrins and Hansen [14]. A covariance matrix for the values of v at the nodes is derived from the structure of the tree and used to weight a least-squares fit of the values at the internal nodes to the actual leaf data. The value v_0 of v at the root is assumed to change randomly as it evolves through the tree. The value of v at some point in the tree is assumed to follow a Gaussian distribution, with

mean v_0 and and variance proportional to the distance from the root (this is the *Brownian motion* model of evolution). The covariance for two points x and y in the tree is proportional to the distance from the root to their least-common ancestor (lca); we assume the values of v changed independently on the two separate paths from the lca to x and y.

This method is discussed in Rohlf [20]; his Equations 16 and 17 give the weights found by the least-squares fit as closed-form formulas. We use them to assign, for a given internal node in the tree, a set of weights $w^1,\ldots,w^N$ for each of the N input surfaces. Along each tree edge, we linearly interpolate the weights of the endpoints.

2.3 Alignment and Target Configuration

Given N sets of homologous landmark points $\mathbf{L}^i, 1 \le i \le N$ and a set of weights $w^1,\ldots,w^N$, the next step is Generalized Procrustes Alignment (GPA) [9]. This is an iterative procedure that simultaneously determines the positions of the landmarks on the output surface and aligns the input landmark sets to this target configuration.

We begin by scaling each input set of landmark points so that the sum of the squared distances between all of the points and the center of gravity is one. Thus, if $\mathbf{g}^i = \left(\sum_{n=1}^{K} \mathbf{L}_n^i\right)/K$ is the center of gravity for point set i and the sum for each set is $d^i = \sum_{n=1}^{K} \|\mathbf{L}_n^i - \mathbf{g}^i\|^2$ then the scaling factor for each point set relative to the first is $s^i = d^i/d^1, 1 \le i \le N$. The scale variation can be re-introduced into the visualization at the end, as was done to make the images in Figure 2.

We pick one of the input configurations of landmark points, arbitrarily, as the first iteration of the target configuration. Then in each iteration, we align all of the input landmark sets to the proposed target configuration. We use Horn's algorithm [11], which gives an efficient and closed form solution to the problem of finding the rotation and translation of an input landmark set minimizing the sum of squared distances between each landmark point and the homologous point in the target configuration. After aligning all input landmark sets, we compute the new target configuration by taking the weighted average of all of the homologous copies of each landmark point. We terminate the iterative process when the target configuration is "stable." Fewer than ten iterations are typically needed.

2.4 Warp

The next step is to warp all surface models into the target configuration space using the thin-plate spline (TPS). The TPS warp is defined using an input set of landmarks and the target set of consensus landmark points, and it brings the input landmarks into coincidence with the target set. Taking the surface mesh through the same transformation warps the surface into the target space. Warping all N surfaces into this target space results in all surfaces being close to one another.

Conveniently, it is possible to get a closed-form solution for the TPS warp, since it can be expressed as a weighted sum of radial basis functions centered at the landmark points. The weights are determined by solving a linear system, which we do in a straightforward manner.

2.5 Merging

The final step is merging the N surfaces, which after the warp lie very close together. This merge interpolates the surface details.

We initially implemented the method of [7], computing a signed distance field around each surface, computing the weighted average of those distance fields, and extracting the zero set. This works well in most regions, but introduced holes in regions where both sides of a thin sheet-like region of the object are captured in the input scans. If these thin sheets are not perfectly aligned by the warp, holes appear in the output surface, even at arbitrarily high resolution (Figure 6). This occurs because even in the continuous case there are regions where the averaged function is positive over the whole neighborhood of the input surfaces and does not achieve zeros. These (and other artifacts near the boundaries) led us to consider an alternative approach, based on linearly interpolating the squared distance function rather than the signed distance function of the surface. Our new method is also somewhat less sensitive to discretization artifacts.

The squared distance function $d^i(x,y,z)$ associated with one input surface M^i is zero at points of M^i and positive elsewhere, and its gradient at points of M^i is the zero vector; M^i lies at the bottom of the two-dimensional "valley" in the three-dimensional function d^i. The weighted average of the d^i, $d(x,y,z) = \sum w^i d^i$, is similar to a squared distance function, in that there is still a distinct two-dimensional "valley" in the function. The points at the bottom of this valley have small but not zero values of d, and the gradient is exactly zero only at a set of discrete minimum points. We extract the surface tracing out the bottom of this valley. Notice that even at points where the averaged distance function fails to achieve a zero, the averaged squared distance function still has a valley.

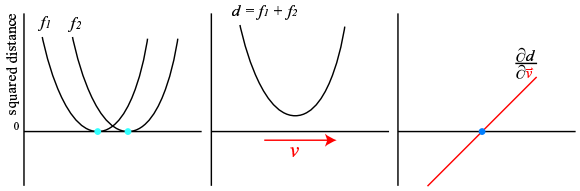

Figure 5: Averaging multiple squared distance functions produces a function that is similar to a squared distance function, but generally not zero anywhere. In one dimension, the squared distance function is a parabola in two-space, and the average of several is also a parabola. In higher dimensions the situation is similar, although complicated by the fact that the input surface normals do not match exactly. Taking a directional derivative in a direction $\mathbf{v}$, roughly perpendicular to the desired surface, produces a signed function, with its zeros defining the bottom of the parabolic "valley."

To extract a surface from a two-dimensional valley in a trivariate function we use an *extremal surface* construction. Extremal surfaces were introduced for extracting surfaces from noisy tensor fields [17], and they have been used recently for defining surfaces from point clouds [3]. We consider the directional derivative of d in a direction $\mathbf{v}$ roughly perpendicular to the valley. Once we have an appropriate vector field $\mathbf{v}$, the directional derivative

$$g = \frac{\partial d}{\partial \mathbf{v}} = \frac{\nabla d \cdot \mathbf{v}}{2} \tag{2}$$

is a signed function, whose zero set is taken as the desired output surface (Figure 5). Since the function g is locally nearly linear, extracting the output surface using marching cubes [13] works well and produces smooth surfaces free of "jaggy" artifacts, except again in regions where the input formed thin sheets (Figure 6).

To find a vector field $\mathbf{v}$ roughly perpendicular to the valley, we use the directions perpendicular to the input surfaces. Specifically, we average the unoriented gradient directions $\nabla d^i$of all of the input squared distance fields d^i; by *unoriented* we mean the directions of the lines supporting the vectors, so that the vectors $\mathbf{v}$ and $-\mathbf{v}$ are

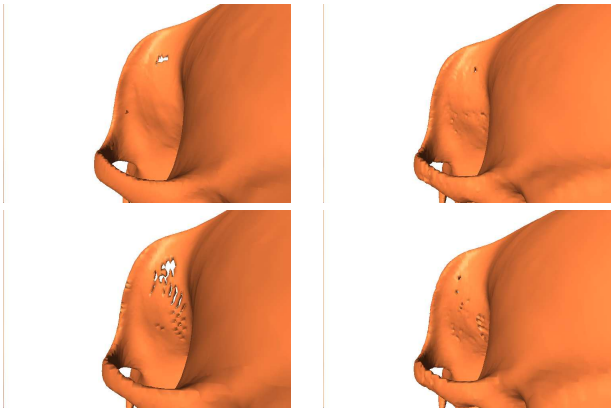

Figure 6: Comparison of signed distance field blending, on the left, and our novel unsigned blending procedure on the right. In the first row, we use a high resolution voxel grid of $300 \times 192 \times 147$. The signed distance field produces a merged surface with a hole behind the eye socket, where two oppositely-oriented surface are close to each other, even though none of the input surfaces had a hole. Our method of averaging the squared distance functions produces the correct topology, although there is still some discretization noise in the difficult region. At lower resolution (in the second row, $150 \times 96 \times 73$), our method is less sensitive to discretization artifacts.

considered to be the same. Even in the difficult regions in which the objects form thin sheets, the unoriented gradient directions are roughly consistent with each other (while the oriented gradient directions can point in opposite directions), so the averaged unoriented vector field is smooth near the surface.

We then propagate a consistent orientation through $\mathbf{v}$. Again, this has to be done carefully in the difficult regions in which the objects form thin sheets. We detect the difficult regions while extracting the distance function, as we describe in a moment. With this information, we can propagate the orientation of $\mathbf{v}$ first in the easier regions and then in the more difficult ones.

We compute the squared distance function d_i for each surface on a voxel grid using the closest point transform (CPT) code distributed by Mauch [16], which implements his robust computational-geometry-based method [15]. This code finds for each voxel $\mathbf{x}$ the nearest *foot-point* on the polyhedral input surface; a foot-point is a surface point $\mathbf{p}$ such that a sphere centered at $\mathbf{x}$ is tangent to the surface at $\mathbf{p}$. The nearest foot-point is also the closest surface point. We modified the code to find not only the closest foot-point but also the second-closest foot-point as well. When the object forms a thin sheet, or at a sharp corner, the second-closest foot-point is at nearly the same distance as the closest. We use the difference between the distance to the closest and the second-closest foot-points as a measure of the difficulty of orienting $\mathbf{v}$. For efficiency, we also modified the CPT code to only compute the squared distance field around each input surface mesh M^i to within a small distance α (we use $1/10$ the largest dimension of the model). We use the CPT code to find the trivariate squared distance function d^i for each M^i and also the exact gradient ∇d^i of the squared distance. Note that the gradient is the same as the unsigned distance function times two. A weighted average of all d^i functions produces a single scalar trivariate function d, and, because of the linearity of the derivative operator, averaging the ∇d^i functions produces its exact gradient ∇d.

Color is also averaged. Each triangle in the input mesh is assigned the average color of its three vertices (we could have interpolated the vertex colors across the triangle but the differences are negligible at the grid resolution we are using). Every voxel closest to this triangle inherits this triangle color. The colors at the voxels are then averaged along with the distance functions. After the surface is extracted, color is assigned to each surface vertex by interpolating the surrounding voxel colors.

The resulting surfaces still have some extraneous parts. One issue is that $g = 0$ at ridges of d as well as at valleys, that is, near the the medial axis of the desired surface as well as the surface itself. We handle this simply by taking the single largest connected component of the output surface; this removes the medial axis components and also some small noise artifacts. We also delete any part of the surface where $|g|$ is not nearly zero (zero crossings at transitions between large positive and large negative values of g). These occur near boundaries.

Finally, at voxels where $\mathbf{v}$ is hard to define because the averaged gradients at the vertices point in very different directions, we do not compute g directly. Instead, we compute g for the surrounding voxels and then interpolate the values, by averaging, to the empty voxels. Note that this hole-filling would not be possible without a consistent orientation for $\mathbf{v}$ and hence a consistent sign for g.

3 Results

Figure 2 illustrates our results. The main point is that the synthetic skulls, created by averaging the input meshes, are virtually indistinguishable from the original models. A video, including an animation of the tree-morph and some examples of interaction with the landmark editor, accompanies this paper.

The input surface meshes varied in size, from 797K to 433K triangles, except for the *Papio* model, for which only a 75K triangle mesh was available. Computing the trivariate distance function from the input mesh is the most expensive part of the computation, and this is roughly linear in the size of the input mesh. For the animation, we simplified all of the meshes down to about 75K triangles, since more detail would not be resolved at video resolution. Note that the trivariate distance function has to be computed for each frame, since the warp for each frame is different. We used the full resolution input meshes for the figures.

We did our processing on four Intel 3.2GHZ Hyperthreaded workstations, each with 2GB of memory. The distance field for the high-resolution meshes is computed in about 500 seconds per model on a voxel grid of size $300 \times 192 \times 147$. Other processing, including the GPA, the TPS warp, and the extraction of the extremal surfaces, required about an hour altogether and was minor compared to the time required to generate the distance functions.

4 Discussion and Future Research

This application raises a number of research questions that we are interested in pursuing. With respect to primate evolution, we plan to compare the average ancestral shapes predicted by the statistical model and illustrated in this visualization with the shapes of known fossils, both visually and statistically. Integration of fossil evidence with a trees such as ours, whose structure is inferred from genomic sequence data on existing species, has to be based on morphological features. Visualizations such as these help paleontologists develop intuition about morphological change and encourage them to accept or reevaluate statistical models.

Generating landmarks automatically in a way that users would find sufficiently accurate and biologically meaningful is an important area for future research. As more data becomes available, the need

for automation becomes more pressing. For instance, it would be very helpful to be able to attract landmark points onto significant geometric features, especially ridges. More ambitiously, it would be useful to be able to develop reliable surface correspondences using only a small number of landmarks, and hence transfer large sets of landmarks almost automatically. There has been some work on this problem in the graphics community [21] and extending these techniques to handle inputs that are not closed manifolds would be helpful.

Finally now that we have a method for producing intermediate surfaces, we can use it to construct a base mesh to use in a system such as that of [2], which might produce more attractive or more efficient warps.

ACKNOWLEDGEMENTS

We thank James Jones, a former U.C. Davis undergraduate, for writing the thin-plate spline code, and Tom Brunet, currently a graduate student at the University of Wisconsin, for the code for Horn's algorithm. We thank Prof. Steve Frost for comments on the manuscript, Lissa Tallman for beta-testing the landmark editor and John Liechty for sound recording. We thank Sean Mauch for making his CPT code available. We thank several anonymous referees for references and helpful suggestions.

REFERENCES

[1] D. C. Adams, F. J. Rohlf, and D. E. Slice. Geometric morphometrics: Ten years of progress following the 'revolution'. *Italian Journal of Zoology*, 71:5–16, 2004.

[2] B. Allen, B. Curless, and Z. Popovic. The space of human body shapes: reconstruction and parameterization from range scans. In *Proceedings of ACM SIGGRAPH*, pages 587–594, 2003.

[3] Nina Amenta and Yong Joo Kil. Defining point-set surfaces. In *Proceedings of ACM SIGGRAPH*, pages 264–270, 2004.

[4] F. L. Bookstein. *Morphometric tools for landmark data: Geometry and Biology*. Cambridge Univ. Press, New York, 1991.

[5] F.L. Bookstein. Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3):225–243, 1997.

[6] D. Breen and R. Whitaker. A level-set approach for the metamorphasis of solid models. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):173–192, 2001.

[7] Daniel Cohen-Or, David Levin, and Amira Solomovici. Three-dimensional distance field metamorphosis. *ACM Transactions on Graphics*, 17:116–141, 1998.

[8] E. Delson, D. Reddy, S. Frost, F.J. Rohlf, M. Friess, K. McNulty, K. Baab, T. Capellini, and S. E. Hagell. 3d visualization of inferred intermediates on a phylogenetic tree–applications in paleoanthropology (abstract). *Amer. J. Phys. Anthropol. Suppl.*, 36:86–87, 2003.

[9] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, pages 33–51, 1975.

[10] W.L. Hodges, R. Reyes, Jr. T. Garland, and T. Rowe. Visualizing horn evolution by morphing high-resolution x-ray ct images. In *Sketches*. SIGGRAPH 2003, 2003.

[11] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, 1987.

[12] A. Lerios, C.D. Garfinkle, and M. Levoy. Feature-based volume metamorphosis. In *Proceedings of ACM SIGGRAPH*, pages 449–456, 1995.

[13] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Computer Graphics (Proceedings of SIGGRAPH 87)*, volume 21, pages 163–169, July 1987.

[14] E.P. Martins and T.F. Hansen. Phylogenies and the comparative method. *American Naturalist*, 149:646–667, 1997.

[15] Sean Mauch. *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*. PhD thesis, California Institute of Technology, Pasadena, California, 2003.

[16] Sean Mauch. Closest point transform, 2004. http://www.acm.caltech.edu/ seanm/projects/cpt/cpt.html.

[17] G. Medioni, M. S. Lee, and C. K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier Science B.V., 2000.

[18] Paul O'Higgins and Nicholas Jones. Morphologika: Tools for shape analysis.

[19] F. James Rohlf. tpsrelw: analysis of relative warps, 2004.

[20] F.J. Rohlf. Comparative methods for the analysis of continuous variables: geometric interpretations. *Evolution*, 55(11):2143–2160, 2001.

[21] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. In *Proceedings of ACM SIGGRAPH*, pages 870–877, 2004.

[22] Dennis E. Slice. Morpheus: Software for morphometric research.

[23] G. Subsol, B. Mafart, A. Silvestre, and M.A. de Lumley. 3d image processing for the study of the evolution of the shape of the human skull: presentation of the tools and preliminary results. In B. Mafart, H. Delingette, and G. Subsol, editors, *Three-Dimensional Imaging in Paleoanthropology and Prehistoric Archaeology*, number 1049 in BAR International Series, pages 37–45. 2002.

[24] D'Arcy Wentworth Thompson. *On Growth and Form*. Dover Publications, 1992 edition.

[25] G. Turk and J. O'Brien. Shape transformation using variational implicit functions. In *Proceedings of ACM SIGGRAPH*, pages 335–342, 1999.

[26] J. J. Wiens, editor. *Phylogenetic analysis of morphological data*. Smithsonian Institution Press, 2000.

[27] Miriam Zelditch, Donald Swiderski, David H Sheets, and William Fink. *Geometric Morphometrics for Biologists*. Academic Press, 2004.