Navigating in a Shape Space of Registered Models

Randall C. Smith, *Member, IEEE*, Richard Pawlicki, István Kókai, Jörg Finger and Thomas Vetter, *Member, IEEE*

Abstract— New product development involves people with different backgrounds. Designers, engineers, and consumers all have different design criteria, and these criteria interact. Early concepts evolve in this kind of collaborative context, and there is a need for dynamic visualization of the interaction between design *shape* and other shape-related design criteria. In this paper, a Morphable Model is defined from simplified representations of suitably chosen *real cars*, providing a continuous shape space to navigate, manipulate and visualize. Physical properties and consumer-provided scores for the real cars (such as 'weight' and 'sportiness') are estimated for new designs across the shape space. This coupling allows one to manipulate the shape directly while reviewing the impact on estimated criteria, or conversely, to manipulate the criterial values of the current design to produce a new shape with more desirable attributes.

Index Terms—Morphable model, shape space, barycentric coordinates, design space.

1 Introduction

This paper describes techniques for navigating in a shape-space of *registered* models created from examples. Models are defined as vectors of geometric features with which to draw the model, and other associated model attributes. A set of registered models are all described by the same features in the same order, and only differ by the feature values. Our examples are cars, but the navigation techniques described are more widely applicable. They enable dynamic visual exploration of the shape-space, guided by estimates of physical properties for new shapes, and predictions for appearance-based semantic labels (e.g. 'sporty').

In any new product development effort, there is a planning phase in which coarse goals and criteria must first be rationalized and balanced. We do not prejudge the relative importance of any criteria—aesthetic, engineering, or market-based—and therefore emphasize exploration and visualization rather than optimization. We believe many 'requirements' are negotiable, and trade-offs will be made if alternatives are visually evident. In the following we develop integrated techniques and associated GUIs to support these activities which we think can greatly aid in design rationalization:

- Exploration of the interplay of criteria with each other and resulting design shapes.
- Navigation toward suitable regions of shape space.
- Direct adjustment of design geometry.
- Restriction to design subspaces by constraints.

The ability to create representative concepts simply and rapidly at a very early stage—in meetings, or with customers—has significant potential. It should be noted, however, that the described research is not a fielded application, nor are the GUIs described currently under usability testing.

1.1 Related Work

This effort builds on the original work on Morphable Models [1][2], and the performance-based methods for tailoring human face models during synthesis [3]. Barycentric coordinates [4], principal components [5], and eigensystems [6] are standard tools in this literature. We augment these with additional mathematical tools for constraining and guiding navigation through the infinite variations of the design space. Specifically, by integrating linear constraints and linear regression into this mix, we make it easier to support engineering and consumer-based criteria *as functions of shape*.

These techniques depend upon registered models in contrast with more general shape similarity measures [7][8] based on aggregated properties used in shape retrieval.

The "bottom-up" design-from-examples approach also contrasts with, but may complement, "top-down" design systems using shape grammars [9]. The latter embeds knowledge about appropriate feature combinations in generative rules; the former derives it from statistics of well-chosen examples. Both must decide which features are relevant, and a hybrid solution may ultimately be most powerful.

The change from 'faces' to 'cars' may seem like a small step, but isn't. Registration algorithms for faces [3], and human bodies [10] do not adapt well to 3D car models—whose surfaces are critically defined by a few major flow curves. Prior registration work with car surfaces [11] produced results not smooth enough for our purposes, and we hypothesize the lack of this constraint is one reason. Similarly, point-to-point correspondence editors [12] do not support these specialized needs. Most likely, car surface meshes need to be derived from these critical flow curves [13] prior to registration, or the curves themselves used to define the Morphable Model with the surfaces generated from them secondarily [14]. We have not solved the *automatic* registration problem either, but utilize flow curves in our representation and hand-fitting of models to templates in order to produce usable results for testing (see Section 2).

1.2 Organization of the Paper

Section 2 provides background on the representations used (both 2D and 3D examples). Section 3 reviews and integrates the basic mathematical steps for visually exploring the design space guided by different kinds of criteria. In Section 3.1 barycentric coordinates are used to browse through the variety of shapes, using GUIs based on the simplex. The GUI also acts as a color-coded parameter map of the shape region indicating directions to increase, decrease, or maintain the value of a selected scalar criterion. Section 3.2 follows with a recursive weighted least squares method that enables *direct manipulation* of shape features coupled with a parameter map showing how deviations from the current feature value affect any other selected criterion. Section 3.3 transforms the linear constraint system to principal components of the shape-space, making computations more tractable in the lower dimensional space. The

- *Randall C Smith and Richard Pawlicki are with GM R&D, E-Mail: randall.c.smith@gm.com.*
- *Thomas Vetter, István Kókai ,and Jörg Finger are with University of Basel, E-Mail: Thomas.Vetter@unibas.ch.*

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 2 November 2007.

relation of barycentric coordinates to PC space is established to integrate these efforts.

In Section 3.4 linear regression is added as another way to relate attributes to shape variables. Physical attributes of vehicles and consumer feedback on shapes may not be obviously related to the shape geometry, but may indeed show a relation through linear regression on the basis vehicles, or more generally on the shape principal components. Any established relationship can then be used as a linear constraint or color-coded in a parameter map to guide navigation. In addition, regression on shape PCs enables prediction of attributes of new shapes not derivable from the basis set of exemplar shapes, and opens the possibility for explanation of these attributes in terms of shape features.

In Section 4 a simple scenario ties these techniques (and GUIs) together. In Section 5 a number of open questions are discussed and Section 6 summarizes and concludes.

2 REPRESENTATION

In order to explore uses of morphable car models, we have created a succession of 2D and 3D hand-registered databases fit to standardized templates (Figure 1). The figures throughout the paper will use one or another representation, but the techniques are applicable to any of them. The old curve template (top) is used for models in Figures 7 and 8. The newer templates contain separate meshes for the body, roof, and wheel openings among others, since these sub-surfaces can move relative to one another by large amounts across a variety of cars—which would distort a single mesh. The meshes represent "base" surfaces—theoretical shapes defined primarily by a few curves. The "fillet" curves for the roof and body 3D templates are shown in yellow. They are instrumental by themselves in defining the size, proportions, and shape of the roof and body segments, in the same sense that a sketch artist defines a concept with a few strokes. They are responsible for controlling the major highlights in the vehicle appearance, which can be seen for 4 of the fitted 3D vehicles in Figure 11. The rest of the template mesh is designed around these curves with higher mesh density in areas that vary substantially from car to car.

Given the fully-detailed target vehicle meshes, the target's characteristic curves must be identified and matched to the template curves. The theoretical target "base" surfaces must be artfully *built*, passing smoothly through extraneous geometric detail, with the template vertices distributed to capture the shape, above, below, or on the target surfaces.

The difficulties inherent in dealing with segmentation of the target into base shapes, identifying and ignoring extraneous detail, and generalizing the target shapes to an ideal partially defined by some existing curves is beyond the ability of current automatic registration algorithms.

2.1 Shape-Space Definition

A shape-space is defined by n+1 exemplars—models chosen for some particular qualities. Each is represented by a d-dimensional vector $\mathbf{x}$ — the defining geometric features for the vehicles—which together with some fixed topological information used in drawing, form the template. Typically, $d >> n+1$. An exemplar is named (e.g. $\mathbf{x}_0$), identifying it as a particular instance of $\mathbf{x}$ with particular feature values. A montage of thirty 2D exemplars is shown in Figure 11. A subscript ($\mathbf{x}_i$), or no subscript, will represent any design; a superscript plus ($+$) indicates an updated design.

The exemplars are grouped as columns in a matrix. The deviation matrix additionally has the average ($\bar{\mathbf{x}}$) of the column vectors subtracted.

$$\begin{aligned} \mathbf{D}_{d,n+1} &\equiv \left[\mathbf{x}_0, \quad \mathbf{x}_1, \quad \ldots \quad , \mathbf{x}_N\right] \\ \tilde{\mathbf{D}}_{d,n+1} &\equiv \left[\mathbf{x}_0 - \bar{\mathbf{x}}, \quad \mathbf{x}_1 - \bar{\mathbf{x}}, \quad \ldots \quad , \mathbf{x}_N - \bar{\mathbf{x}}\right] \end{aligned} \tag{1}$$

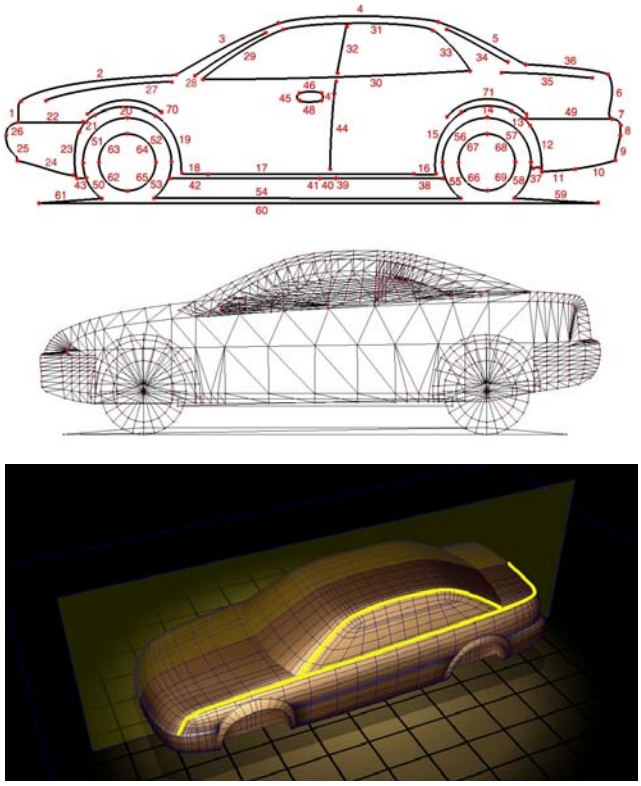

Fig. 1. Versions of 2D and 3D templates.

Designs ($\mathbf{x}$) are *affine combinations* of the exemplars, with the affine constraint on the components of the blending vector $\mathbf{t}$:

$$\mathbf{x} = \mathbf{D}\mathbf{t} = \bar{\mathbf{x}} + \tilde{\mathbf{D}}\mathbf{t} \qquad \left|\sum t_{[i]} = 1. \right. \tag{2}$$

3 NAVIGATION

The following sections develop techniques for navigation in the design space while simultaneously viewing and manipulating the design shape and criteria of interest.

For navigation, it is conceptually convenient to view any created design to be the starting point for the next evaluation in a sequence by adding combinations of the (original) shape deviations to it:

$$\mathbf{x}^+ = \mathbf{x}_{i+1} = \mathbf{x}_i + \tilde{\mathbf{D}}\mathbf{t}_i \qquad \left|\mathbf{x}_0 = \bar{\mathbf{x}}. \right. \tag{3}$$

All designs, including $\bar{\mathbf{x}}$, lie in the same hyperplane. The deviation vectors span the hyperplane, so any *linear combination* of them will stay in the subspace. The additional restriction on $\mathbf{t}$ in (2) is only necessary for equality with the left-hand side of the equation. The vector $\mathbf{t}$ in (3) is unrestricted, but can be adjusted for different purposes without affecting the result. Since the sum of deviations from the mean is zero,

$$\mathbf{0} = \tilde{\mathbf{D}} \cdot \mathbf{1} = \tilde{\mathbf{D}} \cdot (\alpha \mathbf{1}) \tag{4}$$

any vector with the same value in each component can be added to $\mathbf{t}$ and is mapped to $\mathbf{0}$ by (3). Two such adjustments are useful. The vector components can be shifted by the minimum component value to a new minimum value of zero—useful for setting sliders that have minimum zero values:

$$\mathbf{t}_{sliders} = \mathbf{t} - t_{\min}\mathbf{1}. \tag{5}$$

The other adjustment sets $\mathbf{t}$ so its components sum to one again:

$$\mathbf{t}' = \mathbf{t} - \frac{\sum \mathbf{t}_{[i]} - 1}{n+1}\mathbf{1}. \tag{6}$$

Equation (3) can be rewritten, if desired, as a single step adjustment to the mean, or an affine combination of the exemplars as before:

$$\mathbf{x}^{+} = \bar{\mathbf{x}} + \tilde{\mathbf{D}}\mathbf{t}' = \mathbf{D}\mathbf{t}'. \tag{7}$$

3.1 Barycentric Coordinates

In this section, the vector $\mathbf{t}$ in (2) will be manipulated through a user interface, enabling exploration of the design space—calculation of a multilinear estimate of any vehicle's properties and visualization of the resultant interpolated shape.

The most direct implementation to specify $\mathbf{t}$ uses n+1 sliders ranging from zero to one—with one slider for each exemplar. To save space, these were arranged in a circle, as shown on the left in Figure 4. This type of interface is inconvenient, but is a way to *fully* specify any vehicle in the space.

If n+1 points are affinely independent they form the vertices of an n-simplex—the convex hull of the points—embedded in a hyperplane in $\Re^n$ or greater dimension [15]. The 2-simplex is a triangle, embedded in the plane containing its three vertices. Any other point in the hyperplane can be referred to the simplex as an affine combination of its vertices. In (2), $\mathbf{t}$ is the barycentric coordinate of the point $\mathbf{x}$ with respect to the n-simplex defined by the points (columns) of $\mathbf{D}$. If any components of $\mathbf{t}$ are negative or greater than one the point is outside the simplex but still in the containing hyperplane.

Barycentric coordinates are 'coordinate frame free'; given a particular point $\mathbf{p}$, each barycentric coordinate for it is calculated by a function per vertex of the simplex [4]. There is a mapping for any point $\mathbf{p}$ of one n-simplex to its corresponding point in any other n-simplex through the simplex vertices (columns of $\mathbf{V}$ and $\mathbf{D}$) and $\mathbf{p}$'s barycentric coordinate:

$$\mathbf{p} = \mathbf{V}\mathbf{t}_{\mathbf{p}} \mapsto \mathbf{x} = \mathbf{D}\mathbf{t}_{\mathbf{p}} \tag{8}$$

The important property of barycentric coordinates is their well-known ability to interpolate vectors on the simplex vertices smoothly over the interior—commonly used to interpolate colors on triangle vertices across interior pixels or to interpolate physical properties across finite elements. The interpolated scalar property at a point is computed using the barycentric coordinate of that point, and a vector $\mathbf{a}$ of attribute values (one value per simplex vertex)

$$a_{\mathbf{p}}^{+} = a + \left(\mathbf{a} - \bar{a} * \mathbf{1}\right) \cdot \mathbf{t}_{\mathbf{p}}. \tag{9}$$

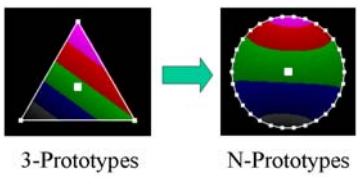

Fig. 2. "Mousing around" in a parameter value map.

Deviations from the attribute mean are blended and added to the attribute value a for the current design. It follows from (8) that if the same attributes are associated with the vertices of both simplexes, the barycentric coordinate of a point in one can be used to estimate the attribute at the corresponding point in the other. With that rationale, the rest of the section will describe the use of simplex-related GUIs for *partially* specifying $\mathbf{t}$.

Generalized barycentric coordinates [4][16] extend barycentric coordinate calculations from the 2-simplex to planar n-sided convex polygons. We use it in a 2D GUI for blending three (or n) vectors (Figure 2). The barycentric coordinate for the cursor point inside the figure is calculated [4] and blends the deviation shape vectors assigned to the vertices by (3). The result is added to the current design shape and the vehicle is then drawn. A cursor point on a vertex regenerates the associated exemplar, or on an edge interpolates two exemplars, assuming the deviations are being added to the mean design. A cursor point in the middle—the barycenter for a standard simplex—averages the deviations (which equals $\mathbf{0}$). Otherwise an interior point blends all deviations—those associated with nearby vertices have the most weight. Any other scalar property of the models (e.g., 'wheelbase') can be interpolated across the interior of the polygon, and its value color-coded (through five 20% ranges, from minimum to maximum values in the Figure).

While a GUI could be made using a 3-simplex (tetrahedron) and a 3D cursor, this approach cannot be directly applied in higher dimensions. However, other than the triangle, the polygon is *not* a simplex, and only a small fraction of the values of $\mathbf{t}$ available in the n-simplex can be generated. For example, it is not possible for widely separated vertices to have strong affects on the blended result; so the ordering of properties on the vertices is important. In Figure 3, exemplars were sorted on wheelbase, and associated with the n-gon vertices in a zigzag pattern across the n-gon, from bottom to top, to create low and high clusters on opposite poles. Deviations were blended for long (top) and short (bottom) wheelbase and added to the average car shape (middle). Note that the average car "character" is generally maintained. Whichever solution is being added to the deviations will appear when the cursor is in the middle of the parameter map—in that case all the deviations are being weighted equally and sum to zero.

If the attributes distributed over the basis vehicles have significant outliers, linear interpolation may not be sufficient. Appropriateness must be determined in any case.

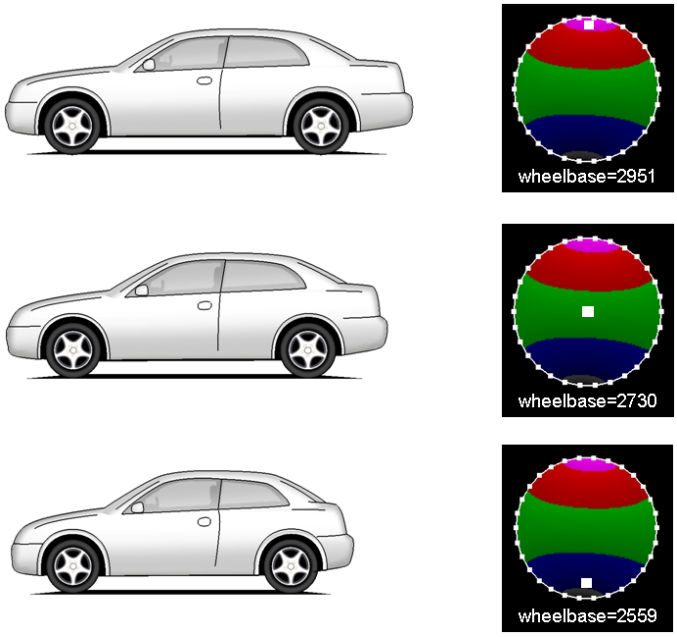

Fig. 3. Three wheelbase variations on the average 2D car (middle).

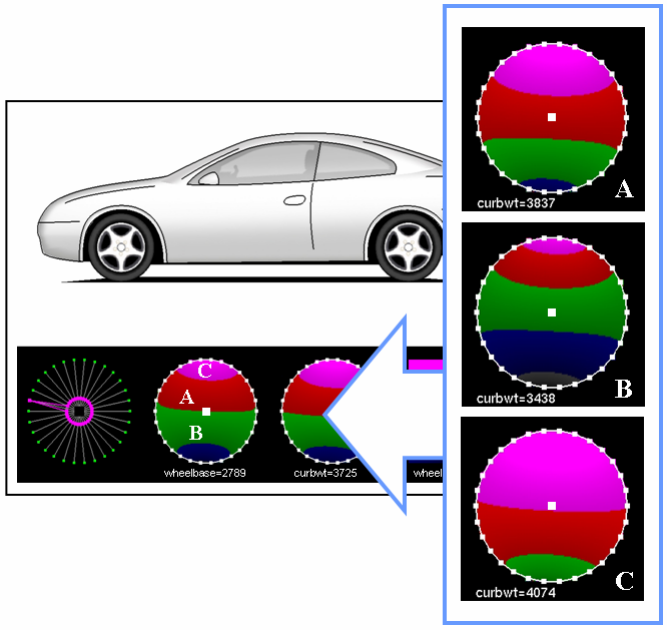

Fig. 4. Changing the wheelbase updates other parameter maps.

Figure 4 shows a typical arrangement with the design window, sliders, and multiple parameters maps. One maximized slider is an indication that the current design is an exemplar (**X22** in Figure 11), and not something else. The inset shows how the parameter map for 'curb weight' changes as the cursor is moved within the 'wheelbase' n-gon from A to B to C while holding down a mouse button. After the first change is made (A) the estimated curb weight for the new design is shown, and the color map indicates it is in the 60%-80% of maximum (red) range. While creating a smaller wheelbase (B), all other parameter maps continuously change to center on the evolving current design—and estimated curb weight has decreased. It increases again as the wheelbase is increased (C), so the two variables are positively correlated. The user could have kept the wheelbase nearly constant instead by moving the cursor along a color contour while observing the other parameters, and the changing design shape. The whole subspace of vehicles with a constant wheelbase, however, is not reachable with this one GUI alone, which can access only a portion of the n-simplex and its interior. The shapes reached will be influenced strongly by the current design. To get to other portions of the space, steps involving other GUIs need to be performed.

3.2 Recursive Weighted Least-Squares

We also need to manipulate the geometric features of the current design directly. The design can be constrained by a linear relation $\mathbf{H}$ (a constant matrix) on the design (for example, geometric point offsets):

$$\mathbf{z} = \mathbf{H}\mathbf{x}\,. \tag{10}$$

Different $\mathbf{H}$ functions can be pre-determined and stored. For any one of them, the current design value for $\mathbf{Z}$ can simply be calculated and displayed. If we provide a target (constraint) $\mathbf{Z}$ value that is different, we want the overall design to adjust, but we also allow a 'tolerance' for the target. How much the design bends to the constraint, and how much the constraint accommodates the current design, are handled by a weighted average described below.

This form is usually seen in a probabilistic filter context [17]. The initial design point is the mean. The covariance matrices $\mathbf{C}$ and $\mathbf{R}$ act as (inverse) weight matrices for the current design and the constraint respectively—the larger the variance, the smaller the weight. $\mathbf{C}$ is initially approximated by the sample covariance of the exemplar data and $\mathbf{R}$ is user-adjustable. Any common scale factor in $\mathbf{C}$ and $\mathbf{R}$ cancels out of (11) and can be divided out of the final result (12). So unless a probability is needed, (13) will be used for convenience. It should be clear that in a sequence $\mathbf{H}$, $\mathbf{Z}$, and $\mathbf{R}$ are potentially different each step, and that the updated $\mathbf{x}^+$ becomes $\mathbf{x}$ in the next step. To enable navigating the design space freely without limiting the space to smaller and smaller subspaces, equation (11) will be utilized without updating $\mathbf{C}$ each time.

$$\mathbf{x}^+ = \mathbf{x} + \mathbf{C}\mathbf{H}^t[\mathbf{H}\mathbf{C}\mathbf{H}^t + \mathbf{R}]^{-1}(\mathbf{z} - \mathbf{H}\mathbf{x}) \tag{11}$$

$$\mathbf{C}^+ = \mathbf{C} - \mathbf{C}\mathbf{H}^t[\mathbf{H}\mathbf{C}\mathbf{H}^t + \mathbf{R}]^{-1}\mathbf{H}\mathbf{C} \tag{12}$$

$$\mathbf{C}_{initial} \approx \mathbf{S}_x \propto \left[\tilde{\mathbf{D}}\tilde{\mathbf{D}}^t\right]_{d,d} \tag{13}$$

In Figure 5 (right side) $\mathbf{H}$ is a 3xd matrix of zeros with a single '1' in each row; it *selects* a particular 3D point from the design. Dragging that point with the 3D cursor to a new position ($\mathbf{Z}$) creates a position difference used in (11) to adjust the entire model, finding the solution closest to the starting design that satisfies the constraint within a tolerance. In a probabilistic context, it is the minimum variance solution for Gaussian variables [18].

It is considerably easier to use a 2D cursor and manipulate a 2D section—e.g., the mid-plane—extracted from the 3D model. In Figure 5 (left side), dragging a point on the section updates the section dynamically (as a morphable sub-model) in its window, and the 3D morphable model updates simultaneously as well.

A range of deviated positions of any particular 2D point in the current design can be encoded in a grid, and the minimum distance design solutions (11) calculated with the same starting design value for each grid point deviation. Any *other* scalar property or linear function of the design variables can then be calculated per grid point solution, and the grid point assigned a color based on its value. In Figure 6, the red curve in the top section is being manipulated by one of its endpoints, and two instances of the design and the outline difference is shown. Associated with the two designs are two parameter maps color-coded through five 20% ranges, from minimum to maximum values for the parameter across the exemplar set. On top of that is a scatter plot of the deviations from the mean (center grid point) of the designated "drag" point for the 30 exemplars. The left-most grid shows a cursor square at the initial "drag" point value for the design; in the center image the cursor has moved to the left, either from dragging the 2D point in the model window, or moving the cursor on the parameter map. In either case, it is clear how to move the point to increase, decrease, or maintain the current parameter value while viewing the resulting and transitional shapes.

A map for the deviation of any two parameters from the current design values can be similar constructed by combining the two scalar linear functions.

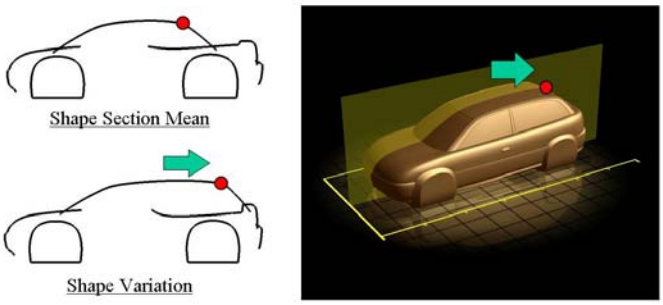

Fig. 5. Dragging a selected 2D or 3D point to reshape the 3D car.

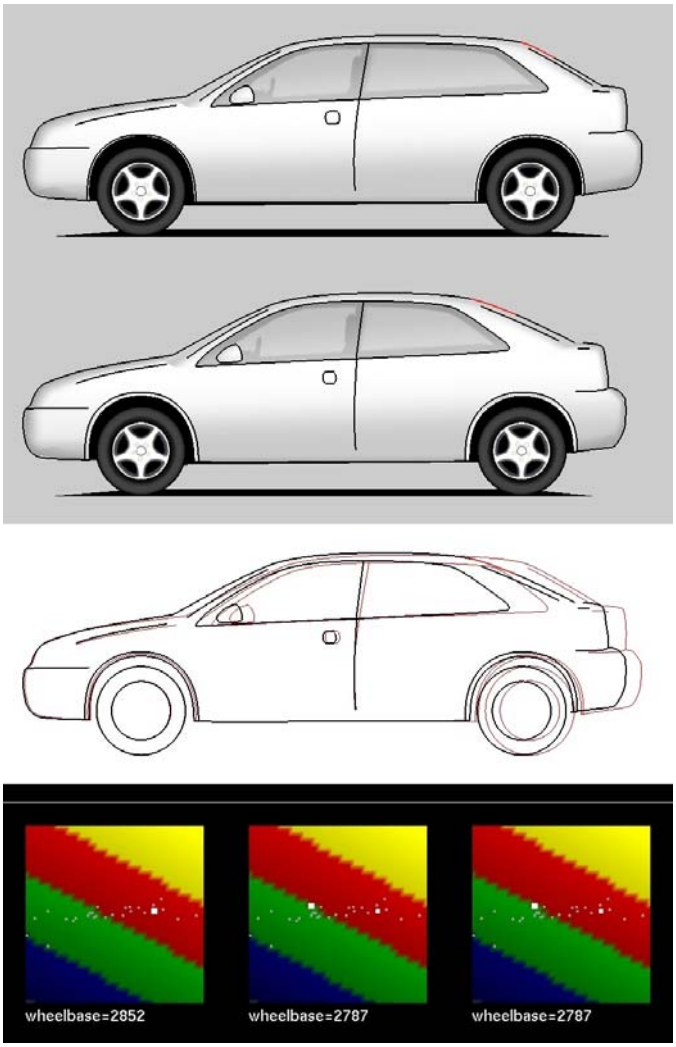

Fig. 6. Dragging a 2D point guided by a parameter value map.

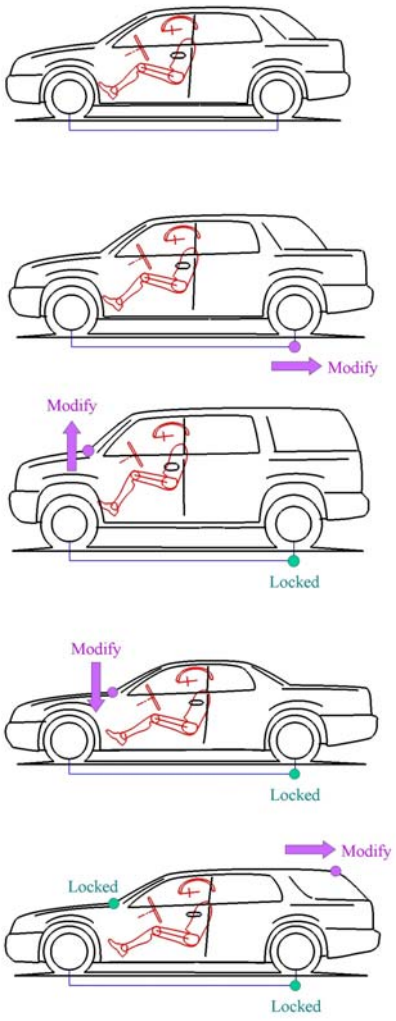

Fig. 7. Locking—using linear constraints to limit the design space.

Applying Equation 11 without applying Equation 12 allows the user to jump around in the whole design space freely; but previous constraints are not enforced. That's plausible while browsing, to understand the "landscape" and shape interactions with various criteria shown in parameter maps. Equation 12 is applied to commit to the constraint. In Figure 7 a vehicle wheelbase is extended and locked at a particular value. The rest of the vehicle reshapes when a second point is subsequently moved (up, and then down) but the wheelbase is fixed (within a specified tolerance). The second point can be locked down, and so on. Significantly, the updated covariance matrix 'remembers' these lost degrees of freedom and implements locking in new vehicle shape calculations (12).

3.3 Principal Components

$\mathbf{C}$ (Equation 12) can be large, even though it is sparse. Transforming the problem into its principal components avoids creation and use of $\mathbf{C}$ altogether, reducing the dimensionality of the problem space and enhancing real-time performance. PCs are a set of uncorrelated parameters that may be used to generalize the idea of shape beyond the basis set, and allow predictions and explanations of properties for other vehicles. In this section the recursive weighted least squares formulae are transformed into the PC space, allowing direct manipulation of meaningful design features economically.

Singular Value Decomposition is applied in (14) to rewrite the exemplar deviation matrix into a product of matrices: $\mathbf{U}$ is column orthogonal; $\mathbf{V}$ is square and orthogonal; and $\mathbf{\Sigma}$ is a diagonal matrix.

$$\tilde{\mathbf{D}}_{d,n+1} = \mathbf{U}_{d,n+1}\mathbf{\Sigma}_{n+1,n+1}\mathbf{V}^t_{n+1,n+1}$$
$$\mathbf{U}^t\mathbf{U} = \mathbf{I}, \mathbf{V}^t\mathbf{V} = \mathbf{I} \qquad (14)$$

Since d>n+1, and $\tilde{\mathbf{D}}$ is composed of mean-deviation column vectors, the rank of $\tilde{\mathbf{D}}$ is at most n. At least one Singular Value will appear as a zero diagonal element in $\mathbf{\Sigma}$. Using (14), the scaled covariance matrix in (13) becomes

$$\left[\tilde{\mathbf{D}}\tilde{\mathbf{D}}^t\right]_{d,d} = \mathbf{U}\mathbf{\Sigma}V^t\mathbf{V}\mathbf{\Sigma}^t\mathbf{U}^t = \mathbf{U}\mathbf{\Lambda}U^t. \qquad (15)$$

The columns of $\mathbf{U}$ are the eigenvectors of (13) and the diagonal elements of $\mathbf{\Lambda}$ are the corresponding eigenvalues. In Principal Component Analysis, the diagonal elements of $\mathbf{\Lambda}$ (and $\mathbf{\Sigma}$) are sorted in descending order—placing any zeros in the last diagonal elements. The columns of $\mathbf{U}$ and $\mathbf{V}$ are sorted correspondingly.

Jolliffe [5] defines the (centered) principal components using the sorted eigenvectors as

$$\mathbf{u} \equiv \mathbf{U}^t\tilde{\mathbf{x}} \qquad (16)$$

which, using (14) gives

$$\mathbf{u} = \mathbf{U}^t \tilde{\mathbf{D}} \cdot \mathbf{t} = \mathbf{\Sigma} V^t \mathbf{t} \,. \tag{17}$$

Finally, using the definition

$$\mathbf{G} \equiv \mathbf{H}\mathbf{U} \tag{18}$$

we get the recursive forms of (11) and (12) in the lower-dimensional PC space, and can move back and forth to feature space where the variables have more obvious meanings for user manipulation.

$$\begin{aligned} \mathbf{u}^+ &= \mathbf{u} + \mathbf{\Lambda}\mathbf{G}^t[\mathbf{G}\mathbf{\Lambda}G^t + \mathbf{R}]^{-1}(\mathbf{z} - \mathbf{H}\mathbf{x}) \\ \mathbf{u}_{i=0} &= \mathbf{0} \end{aligned} \tag{19}$$

$$\mathbf{\Lambda}^+ = \mathbf{\Lambda} - \mathbf{\Lambda}\mathbf{G}^t\left[\mathbf{G}\mathbf{\Lambda}G^t + \mathbf{R}\right]^{-1}\mathbf{G}\mathbf{\Lambda} \tag{20}$$

Note that $\mathbf{\Lambda}^+$ and $\mathbf{\Sigma}^+$ are no longer diagonal.
Barycentric coordinates can now be tied to PC space by solving

$$\mathbf{t} = \mathbf{V}\mathbf{\Sigma}_0^{-1}\mathbf{U}^t(\mathbf{x}^+ - \bar{\mathbf{x}}) = \mathbf{V}\mathbf{\Sigma}_0^{-1}\mathbf{u}^+ \tag{21}$$

using (14) and (16). Note that locking (use of equation 20) restricts subsequent new designs $\mathbf{u}^+$ to a subspace, which restricts $\mathbf{t}$ in turn via (21) using the *un-updated* $\mathbf{\Sigma}$ from (14). Corresponding cursor points in the n-gon for these restricted $\mathbf{t}$ vectors are computed as before (see [4]). Alternatively, an unrestricted $\mathbf{t}$ vector can be computed from the cursor position in the n-gon [4], and used in (17) with updated $\mathbf{\Sigma}^+$ to generate new designs in the restricted subspace.

3.4 Linear Regression

In [3], a subjective score (like facial 'boniness') for each exemplar was used to calculate a 'boniness' shape deviation vector that could be scaled and added to another face model for effect:

$$\begin{aligned} \tilde{\mathbf{x}}_a &= \sum \tilde{\mathbf{x}}_i * a_i \\ \mathbf{x}^{(+)} &= \mathbf{x}^{(-)} + s * \tilde{\mathbf{x}}_a \end{aligned} \tag{22}$$

with a_i the elements of an attribute vector (the scores), and s the scale factor. The technique associates a semantic label with a "direction" in shape-space. The attribute scores can be garnered from experts, or drawn from the population. In Figure 8, we self-scored a number of vehicles (not the same set as Figure 11) for the word 'sporty'. Not surprisingly, cars attain characteristics of sports car—yet the transitions are interesting.

Alternatively, (23) uses scores to determine a linear function on the principal components of the shape-space. For scores on all the exemplars, $\mathbf{t} = \mathbf{e}_i$ trivially gives back the already-supplied attribute for the designated exemplar, or otherwise interpolates the values.

$$\begin{aligned} a &= \mathbf{a} \cdot \mathbf{t} \\ a &= \mathbf{a}(\mathbf{V}\mathbf{V}^t)\mathbf{t} = \mathbf{\beta} \cdot (\mathbf{V}^t\mathbf{t}) \end{aligned} \tag{23}$$

Substituting the parenthetical term—which is equal to the Identity matrix—and regrouping, transforms the identify function into a linear function with coefficients $\mathbf{\beta}$ on the standardized PCs. Since PCs are orthogonal, eliminating small PCs to produce a linear relation with only a few important parameters is simple: zero out their coefficients.

Figure 9 shows a linear fit of six of the largest PCs from a 2D sample set of 30 cars to their actual 'weight', with approximately 90% of the weight variance 'explained' by the linear model. The largest PCs are related to size and proportion, but it is still not obvious that a 2D outline should be a good first-order predictor for real vehicle mass. Nevertheless, the example is only illustrative of the potential—not intended to be regarded as a validated result.

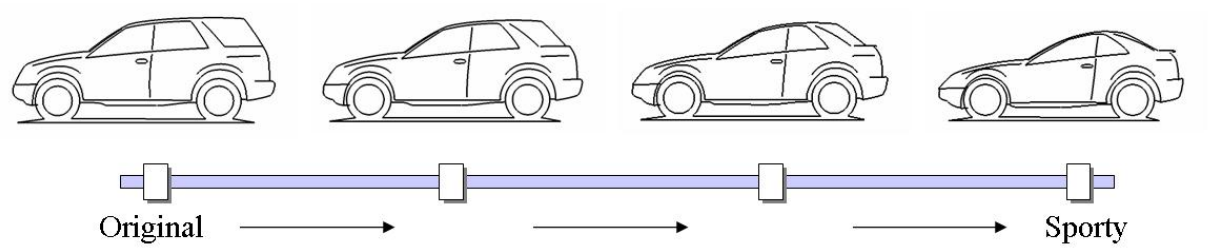

Fig.8. Adding increasing amounts of 'sporty' to an initial vehicle.

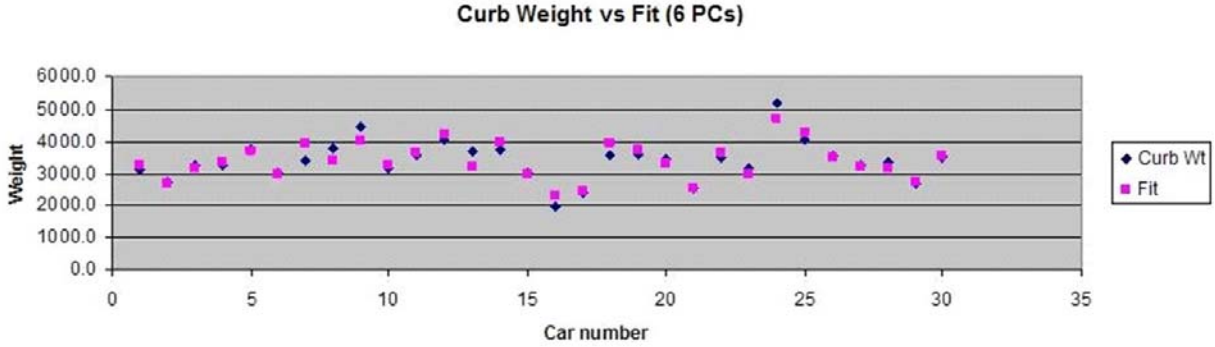

Fig. 9. Real and fit curb weight for thirty cars from six 2D shape PCs

4 Usage Scenario

An example of use is shown in Figure 10. An initial design has already been created (top frame), as indicated by the pattern in the sliders below it. The depicted vehicle was initially exemplar **X4** (from Figure 11) and has had its wheelbase greatly reduced using the n-gon labelled "wheelbase". The curb weight was also reduced as a result. The character of the original exemplar is still plainly evident.

In the middle frame, the design has been further modified by dragging a point on the rear glass of the model (highlighted in red) in the "sporty" direction dictated by the "sporty" GUI. The cursor (large white dot) in that GUI indicates the movement of the dragged point from middle to left (in the top frame to the middle frame), making the wheelbase smaller still. The resulting design is shown in the bottom panel with the original **X4** outline in red.

5 Discussion

There are many questions that have not been addressed in this paper: linear vs. non-linear methods; the number of exemplars needed; whether to treat size and proportion information separately from shape; and how to handle "details"—those aspects of individual vehicles that are not amenable to capture through common feature vectors and summarization through statistics.

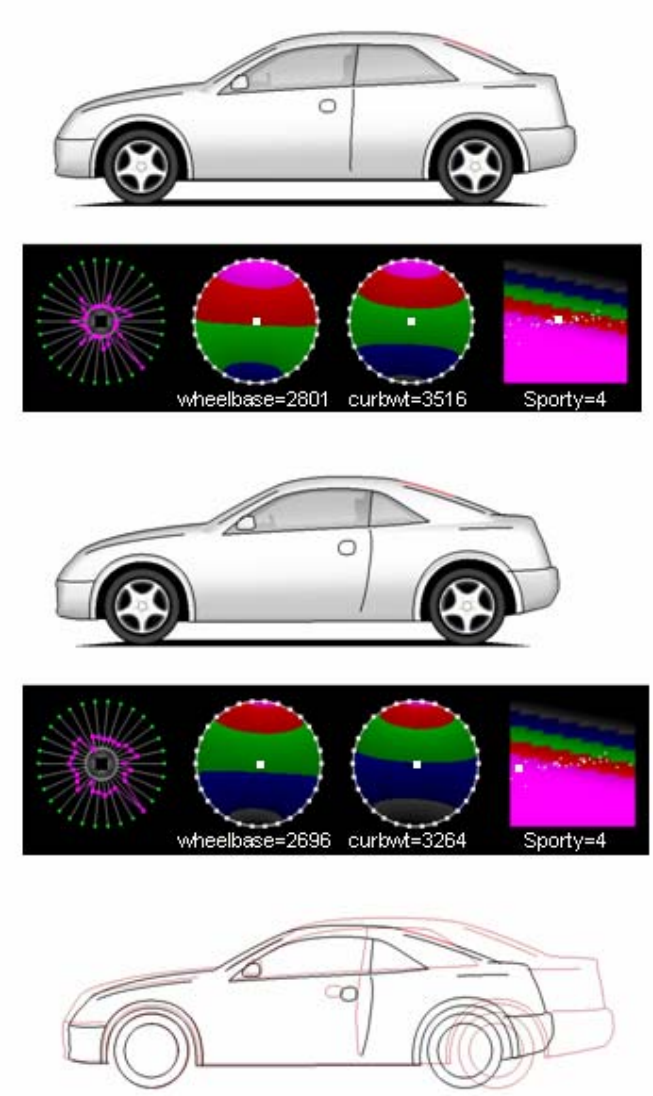

Fig. 10. Two step modification of **X4**.

We have been pleasantly surprised at how well linear models have worked—both in the 2D models sets, and in a small (8 model) set of 3D models. As mentioned in the introduction, initial template definition and registration of models are difficult problems. We have iterated over a number of templates, and artfully hand-registered models to these templates, giving us good results. One obvious place to test the validity of linear interpolation is shown in Figure 7. Each car was modelled *with* an occupant in design position (heel point, hip point), and the occupant was linearly interpolated with the vehicle. The occupant was *not* a kinematic model—its limbs were just drawn between the repositioned joints (points). One would expect that with too much shape variation the figure's limbs would stretch, rather than rotate into position, and indeed this happens. Over a large range (from truck to low-slung sports car) this change was only a small proportion of limb length, amounting to a few millimeters—perfectly valid for illustration and discussion.

Since the 2D template has been evolving, we have not been able to accumulate models without rework from stage to stage—limiting us at any particular stage to less than one hundred models at any time in the basis set. Mixing vehicles across segment—trucks, SUVs, sedans—can produce very interesting cross-over results, but is going to take many vehicles to span the large shape variations. For many practical purposes, it is more desirable to model within segment (Figure 11 has no SUVs or trucks in it). The more homogenous the model set, the more likely it is that linear regression will work well. Projecting a new vehicle into the shape-space, and regenerating it does *not* produce a near-replica of the shape; more models will be needed in the set, but we can't predict the number. On the other hand, smaller model sets like the ones with which we are experimenting have been used to predict properties of new vehicles to first order, and their interpolations do produce an interesting variety of pleasing shapes.

The exemplars have not been normalized to separate shape characteristics from the dominating influence of size and proportion—this will be a future investigation. The first few PCs describe most of the set variance, since they represent the large variations in size and configuration (cab forward or rear). With a sample of 30 2D cars, there are still 29 PCs (the maximum number) achieving numerical significance based on a common cut-off threshold—but their value after the first few is dubious. However, if no PCs are eliminated, they by definition must reconstruct the exemplars, leading to some interesting cases where a "small" PC matters. Even though an exemplar may be registered to the shape template, it can still contain an idiosyncratic sub-shape highly unique to it within the set. In one instance, a unique bumper was "identified" with a small PC, in the sense that the PC value changed dramatically in the one vehicle only. We were able to "graft" the bumper onto another car by substituting the unique PC value for this into the second vehicle's PC description.

To define a design template, or achieve model registration, we have to focus on common characteristics. Two-door and four-door vehicles are both represented in Figure 11 by hinting at door placement with a single door cut. Otherwise we would segregate these two types of vehicles—we don't want doors partially emerging, or popping in and out of the design as a whole during morphing.

However, some level of detail seems required to gain acceptance. It would be interesting to understand how effective (in terms of information content) simple line models are vs. more realistic models. We have added pseudo-3D shading as a texture to the 2D models, controlled by the underlying mesh so that it reshapes with the vehicle. Shading information is not contained in the morphable model, and additional care needs to be taken for decisions based on appearance.

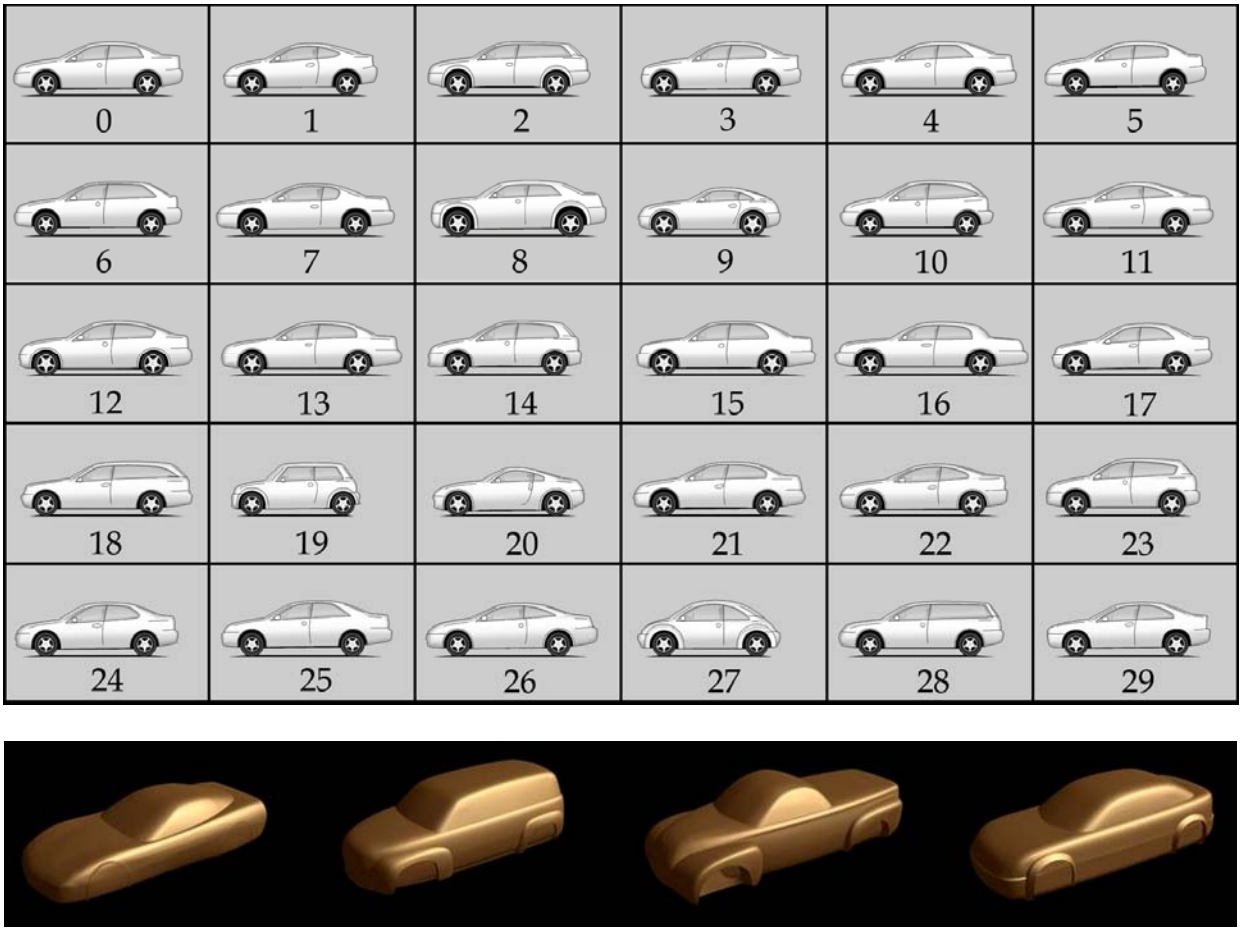

Fig. 11. Montage of thirty 2D vehicles and four of the eight 3D vehicles fit to the templates

6 Conclusion

Conceptual design involves many people with different backgrounds. Artists and designers, engineers, and consumers all have different criteria. The techniques developed in the previous sections can be used to explore the interplay of these criteria with shape, and suggest ways to move the design in a positive direction. Initial discussions with prospective users have generated many ideas in design, engineering, and marketing where an effective shape-space tool could be of use. To make our results more effective, we need to build substantially larger databases to be able to capture more subtle variations; test and validate the linear prediction models vs. higher order models; and look into the effects of removing size and proportion information and treating it separately from shape. Finally, a good automatic 3D model registration algorithm would make large 3D databases possible, and 3D applications viable.

References

[1] M. Jones, T. Poggio, "Hierarchical Morphable Models", CVPR, pp. 820-826, 1998.

[2] M. Jones, T. Poggio, "Multidimensional Morphable Models", Proceedings International Conference on Computer Vision, 1998.

[3] V. Blanz, T. Vetter, "A Morphable Model for Synthesis of 3D Faces", Proceedings SIGGRAPH, 1999.

[4] J. Warren, S. Schaefer, A. Hirani, M. Desbrun, "Barycentric Coordinates for Convex Sets", Technical Report, Rice University, 2004.

[5] I. T. Jolliffe, *Principal Component Analysis*, New York, Springer-Verlag, 1986.

[6] M. Turk, A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86, 1991.

[7] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewics, D. Dobkin, "Modeling by Example", Proceedings SIGGRAPH, 2004.

[8] M. Kazhdan, T. Funkhouser, S. Rusinkiewics, "Shape Matching and Anisotropy", Proceedings SIGGRAPH, 2004.

[9] S. Orsborn, J. Cagan, R. Pawlicki, R. Smith, "Pushing the Limits of Vehicle Design: Utilizing a Parametric Shape Grammar to Explore Cross-Over Vehicle Concepts", Proceedings DETC, 2006.

[10] B. Allen, B. Curless, Z. Popovic, "The Space of Human Body Shapes: Reconstruction and Parameterization from Range Scans", Proceedings SIGGRAPH, 2003.

[11] C. Shelton, "Morphable Surface Models", International Journal of Computer Vision, Vol. 38, No. 1, pp. 75-91, 2000.

[12] D. Wiley, N. Amenta, D. Alcantara, D. Ghosh, Y. Kil, E. Delson, W. Harcourt-Smith, F. Rohlf, K. St. John, B. Hamann, "Evolutionary Morphing", Proceedings IEEE Visualization, VIS 05, pp. 431-438.

[13] P. Alliez, D. Cohen-Steiner, O.Devillers, B. Levy, M. Desbrun, "Anisotropic Polygonal Remeshing",SIGGRAPH, 2003/ACM TOG.

[14] I. Kokai, J. Finger, R. Smith, R. Pawlicki, T. Vetter, "Example-Based Conceptual Styling Framework for Automobile Shapes", Proceedings, Fourth Eurographics Workshop on Sketch-Based Interfaces and Modeling, 2003. (Accepted for publication)

[15] H.S.M. Coxeter, *Regular Polytopes*, Third Edition, Dover Edition 1973.

[16] M. Meyers, H. Lee, A. Barr, M. Desbrun, "Generalized Barycentric Coordinates on Irregular Polygons", Journal of Graphics Tools, Nov. 2002.

[17] A. Gelb, (Ed.), *Applied Optimal Estimation*, M.I.T. Press, 1984.

[18] R. Smith, R. Pawlicki, "Conceptual Shape Design Using Optimal Estimation Techniques" GM R&D Collaboration Report, 2003.