Eurographics Symposium on Geometry Processing (2004)
R. Scopigno, D. Zorin (Editors)

Similarity-Based Surface Modelling Using Geodesic Fans

Steve Zelinka and Michael Garland

University of Illinois at Urbana–Champaign, USA

Abstract

We present several powerful new techniques for similarity-based modelling of surfaces using geodesic fans, a new framework for local surface comparison. Similarity-based surface modelling provides intelligent surface manipulation by simultaneously applying a modification to all similar areas of the surface. We demonstrate similarity-based painting, deformation, and filtering of surfaces, and show how to vary our similarity measure to encompass geometry, textures, or other arbitrary signals. Geodesic fans are neighbourhoods uniformly sampled in the geodesic polar coordinates of a point on a surface. We show how geodesic fans offer fast approximate alignment and comparison of surface neighbourhoods using simple spoke reordering. As geodesic fans offer a a structurally equivalent definition of neighbourhoods everywhere on a surface, they are amenable to standard acceleration techniques and are well-suited to extending image domain methods for modelling by example to surfaces.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations I.4.7 [Image Processing and Computer Vision]: Size and shape

1. Introduction

Recent years have seen an explosion of research in the image domain based on the idea of modelling by example. These techniques, grounded in the application of fast neighbourhood sampling and comparison methods, allow for intuitive, powerful image modelling with very little artistic skill required of the user. Most prominently, several high quality texture synthesis algorithms [EL99, WL00, Ash01] have been developed using neighbourhood comparison as the basic operation, allowing the creation of unlimited amounts of texture imagery from an example image. Applications for this range from scene modelling for production to foreground element removal in photographs. Image analogies [HJO*01] allow the reproduction of a sufficiently local image transformation, given by a pair of before- and after-transformation images, on new images using local joint neighbourhood comparisons between the pairs of sample and target images. Efficient and powerful image editing has been demonstrated [BD02] by applying a modification to all similar regions of an image simultaneously, where similarity is determined by neighbourhood comparison.

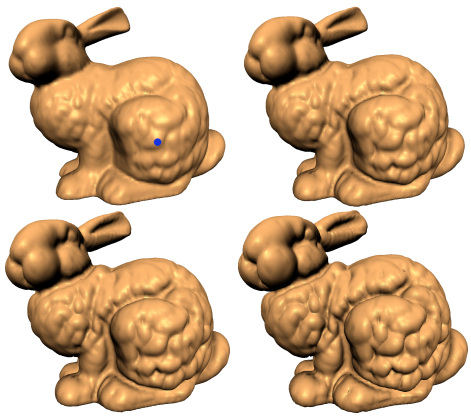

Figure 1: *Pulling a single vertex (blue) simultaneously deforms similar vertices by progressively greater amounts, each according to their similarity to the edited vertex.*

In this paper, we develop a framework for local neighbourhood definition, sampling, and comparison for surfaces, which allows us to leverage this existing research in the image domain to create novel surface manipulation tools. We focus on a number of tools which generalize self-similarity based image editing [BD02] to surfaces. *Similarity-based surface painting* is a direct generalization of their system, distributing colours over a manifold surface rather than

an image. With *similarity-based deformation*, on the other hand, we reproduce a vertex displacement over similar regions of the surface, allowing efficient and intuitive editing of the surface's geometry. Finally, *similarity-based filtering* directs automatic filters such as smoothing operators to modify only similar regions of the mesh. As in the image domain, surface modelling tools based on local neighbourhood comparison and modelling by example are more intelligent and intuitive than standard editing tools, and are especially useful for non-artists, allowing the character of a mesh to be changed convincingly with little user effort or artistic skill.

We also demonstrate a family of similarity measures. The key abstraction to our framework is to define similarity in terms of a signal over the surface. Typically, this signal measures geometry or colours, but it can be any arbitrary function defined on the surface. Thus, we can modify the geometry of an object based on similarities in its texture, or paint an object based on geometric similarity. We may also limit edits to *correlations* of signals, such as regions where both the geometry and the colours match. A user is able to pick the components of the similarity measure which make the most sense for the task at hand, allowing for powerful, widely-applicable and intuitive surface editing.

Another key contribution of this paper is our *geodesic fan-based* framework for local surface comparison. Geodesic fans are neighbourhoods where the sample positions are uniformly spaced in the geodesic polar coordinates of the center, thus resembling a fan of geodesics on the surface. The use of geodesics naturally avoids distortion of the neighbourhood, while uniformly sampling in this space offers very fast simultaneous alignment and comparison of local portions of surfaces, as we explain in Section 3.1. Their efficiency and capability to generate oriented correspondences across surfaces are critical components of our applications, and we believe they will prove useful in many other contexts as well.

2. Related Work

Our work is inspired by the self-similarity based image editing system of Brooks and Dodgson [BD02]. As pixels of a texture are painted, their system automatically reproduces the painting over all similar areas of the texture. The opacity of the paint is varied according to the similarity of the affected regions with the region the user painted, causing changes to blend naturally into the existing texture. Thus, self-similarity based image editing provides a very powerful, intuitive tool for quickly editing textures and photographs. The similarity measure they use is based on neighbourhood similarity, as is typically used in the texture synthesis community [EL99, WL00, Ash01, TZL*02]. Such measures have recently been used in many other related applications as well, such as image analogies [HJO*01], example-based super-resolution [FJP01, LSZ01], and colourizing black and white photographs from a colour example [WAM02].

Little of this work with neighbourhood comparison has been extended to surfaces. Recent techniques for texture synthesis over surfaces [Tur01, WL01, SCA02, MK03, ZG03] map surface regions into standard image-domain neighbourhoods, effectively performing matching in texture space, and assume that a pre-specified orientation field providing a canonical mapping from world space to texture space. Curve analogies [HOCS02] use comparison of neighbourhoods on curves to reproduce a curve transformation on a new curve. In order to achieve rotation invariant comparison, they use an explicit alignment step to bring neighbourhoods into correspondence before comparing them. This approach does not generalize well to surfaces, as their alignment procedure relies on curves' 1D nature, and more general point set alignment techniques are computationally very expensive.

There are some surface processing techniques that make use of geodesic polar coordinates [Goe70], as we do. Welch and Witkin [WW94] used geodesic polar maps in the 1-ring of a vertex (a local conformal map) in fitting a local approximation of the surface. Mesh smoothing was demonstrated [MKY98] by casting and sampling positions for a standard smoothing kernel from geodesics, in both grid and geodesic fan patterns. Biermann *et al.* [BMBZ02] cut and paste mesh geometry using a set of geodesic spokes to previsualize for the user the extent of the pasted result on the target, as well as to help establish a common parameterization of the cut region on both the source and target meshes.

Systems capable of producing results somewhat similar to ours include Cellular texturing [FLCB95] and a more recent feature-based variant [LDG01]. These methods generate geometry or texture over a surface procedurally using interacting cell programs, but require some technical expertise on the part of the user to produce an appropriate cell program. Variational modelling systems [WW94, TSK97] compute minimal energy surfaces based on semantic multiresolution constraints ranging from interpolating sets of points to volume or surface area specification. The multiresolution signal processing framework of Guskov *et al.* [GSS99] is perhaps most similar to our own work in terms of the results produced, but is based on direct editing of frequency bands of the mesh. While this provides for powerful interaction, its global nature can cause unintended side-effects throughout the surface. Our system is in some respects complimentary to theirs, as similarity-based filtering may be used to intelligently avoid these side-effects where not desired.

Given that a goal of our research is to develop a framework for measuring and matching neighbourhoods on surfaces, work from the object matching community is also relevant. Much of this work focuses on matching entire objects and developing compact object descriptions [HSKK01, FMK*03, HK03], in the context of content-based retrieval (see [FMK*03] for a good survey), rather than matching small portions of surfaces. Spin Images [JH99] are the most notable local comparison method

here, and are widely used to establish feature correspondences on surfaces for object recognition and matching. Computed by spinning a sheet of buckets about an oriented surface point and aggregating other points of the surface into the buckets, it is not clear how they may be extended to sample other values, such as colours or displacements, which are unrelated to the spin image parameterization. Further, while point correspondences are straightforward to establish, it is more difficult to establish oriented point correspondences using spin images, as this would require three point correspondences to be established in close proximity; it seems computationally difficult to establish additional correspondences given one already established correspondence. Shape contexts [BMP02] have been used for matching shapes in images, and are quite similar to both spin images, being histogram-based, and geodesic fans, using bins in the polar domain. Work with shape contexts, however, has not considered orienting the histogram bins other than naturally according to the image orientation or externally according to some pre-specified frame (such as given by silhouette edges). This work was also recently extended to matching full 3D objects [KPNK03] using a canonical orientation step based on the principal axes of the surface.

3. Local Surface Comparison

We begin by defining the problem of local surface comparison. We are given a manifold surface M over which some signal $\omega : M \rightarrow R^n$ is defined. We define the local difference of two points $\mathbf{p}$ and $\mathbf{q}$ to be the integral of differences in ω over corresponding regions in the neighbourhoods of $\mathbf{p}$ and $\mathbf{q}$. We approximate this local difference by sampling ω at a set of corresponding points $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$, and summing their differences under the L_2 norm:

$$D(\mathbf{p},\mathbf{q}) = \sum_i ||\omega(\mathbf{p}_i) - \omega(\mathbf{q}_i)|| \qquad (1)$$

Typically, ω measures the surface geometry, colours, or a weighted combination of of such signals.

The main difficulty in evaluating D on surfaces is in establishing corresponding points $\{\mathbf{p}_i\}$ and $\{\mathbf{q}_i\}$. We use a local parameterization at each neighbourhood center and specify offsets in the parameter domain. If the parameterizations are aligned, then evaluating the same parameter domain offset in both neighbourhoods produces corresponding points. We first define our choice of local parameterization, then show how they may be efficiently aligned.

3.1. Geodesic Fans

We use the *geodesic polar parameterization* of a point to establish sample positions on surfaces, sampling uniformly in the associated parameter domain. If one arbitrary geodesic passing through a point $\mathbf{p}$ is designated a *polar base*, every other geodesic passing through $\mathbf{p}$ may be parameterized by its angle θ with respect to the polar base in the conformal

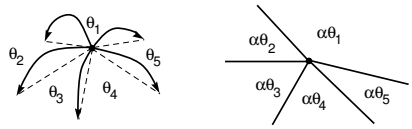

Figure 2: *The conformal plane of a point on a surface. The geodesics through a point have angles θ_i between them on the manifold. A conformal map scales these angles by $\alpha = 2\pi/\sum\theta_i$, so they sum to 2π, while preserving arclengths.*

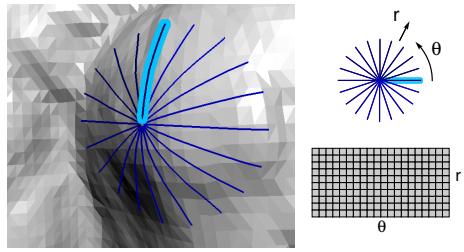

Figure 3: *A geodesic fan on a surface with the polar base highlighted. The parameter domain is shown in polar form and with samples arranged as a grid in parameter space.*

plane at $\mathbf{p}$ (the conformal plane of a point maps all geodesic curves through the point into the plane such that they cover an angular range of $[0,2\pi]$, as shown in Figure 2). Any point $\mathbf{q} \in M$ may be parameterized with respect to $\mathbf{p}$ as (θ, r), where θ identifies a geodesic through both $\mathbf{p}$ and $\mathbf{q}$ and r the arclength on this geodesic of $\mathbf{q}$. Note that these coordinates may not be unique for a given point, but any particular set of coordinates identifies a unique point on the surface. The pair (θ, r) are called the *geodesic polar coordinates* of $\mathbf{q}$ with respect to $\mathbf{p}$. Thus, as shown in Figure 3, if we position samples uniformly in these coordinates, the angle axis gives geodesic spokes equally-spaced about the neighbourhood center, and the arclength axis spaces samples equally along these spokes (note that samples will generally not fall on vertices). We call this arrangement a *geodesic fan*.

3.2. Local Correspondence

Now that we have a local parameterization of the surface, the remaining question is how to determine the direction of the polar base in each neighbourhood that provides corresponding parameterizations. This is equivalent to arbitrarily choosing the polar base at one point, and finding an optimal corresponding direction at the other point. Note that when two neighbourhoods are optimally aligned, their difference according to (1) is minimal. Further, we may discretely rotate a geodesic fan simply by cyclically permuting its spokes. For example, let us number the spokes of a particular geodesic fan $g_0(\mathbf{p}) = (s_0, s_1, ..., s_n)$, where $s_i = \{\omega_{i0}, ..., \omega_{im}\}$ is the set of all samples from spoke i. If we cyclically permute this fan's spokes by 1, we get $g_1(\mathbf{p}) = (s_1, s_2, ..., s_n, s_0)$. Note that this geodesic fan is precisely the geodesic fan that would be

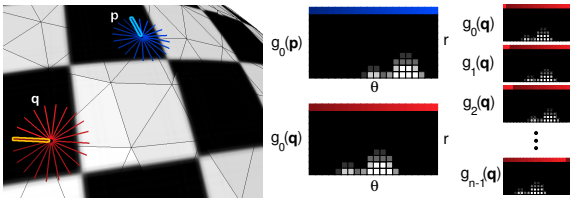

Figure 4: *Left: A colour signal on a surface and two unaligned geodesic fans with highlighted polar bases. Middle: Sampled signal from each fan; the difference would be overestimated since the fans do not correspond. Right: Cyclic permutations of the spokes of the fan at* ***q*** *(each spoke maps to a column in the sample grid). The permutation with minimal difference with the fan at* ***p*** *approximately aligns them.*

sampled if the polar base of $g_0(\mathbf{p})$ were rotated in the conformal plane by the spacing angle ($\frac{2\pi}{n}$). Thus, as shown in Figure 4, we may produce n cyclic permutations $\{g_i(\mathbf{p})\}$ of one geodesic fan. This allows us to approximate (1), which assumes corresponding points in each neighbourhood, as:

$$D(\mathbf{p},\mathbf{q}) = \min_i D(g_0(\mathbf{p}), g_i(\mathbf{q})) \tag{2}$$

where the polar bases at $\mathbf{p}$ and $\mathbf{q}$ are chosen arbitrarily. Thus, we approximate the optimal alignment by comparing one fixed fan against all discrete rotations of the other; the rotation with minimal difference aligns them.

Note that this alignment procedure is with respect to the signal ω. It will not necessarily approximate the optimal alignment of the *geometry* of the neighbourhoods. This is by design, as, for example, the optimal geometric alignment is likely to be quite different than the optimal alignment of texture values. A user may choose the signal (or combination thereof) most appropriate to the task at hand. We now describe these functions in more detail.

3.3. Signal Measurement

We have defined ω as a function over a surface which represents a signal of interest. An example of such a signal would be a texture map. Here, a signal sample at some point of the surface would be a 3-vector containing the colour of the texture at that point. With triangle meshes, we typically have signal values defined at the vertices which are linearly interpolated over triangles (map based signals have their map coordinates interpolated, rather than the signal itself).

Since a surface is embedded in 3D space, we naturally want to measure and compare the geometry of the surface for many applications. To this end we define a function ω_{geom} which represents the geometry of the surface; its value at any point on the surface is simply that point's position in space. In order for this function to be useful in our applications, it must be rotation-invariant, so we transform this position into a local frame on the surface. Note that in order to ensure a particular sample's value is constant over discrete rotations of the neighbourhood, this frame must be relative to the *current* spoke of the neighbourhood. Thus,

$$\omega_{geom}(\mathbf{p}_i) = \begin{bmatrix} \mathbf{t}_s \\ \mathbf{b}_s \\ \mathbf{n} \end{bmatrix} (\mathbf{p}_i - \mathbf{p}) \tag{3}$$

where $\mathbf{p}_i$ is the sample position, $\mathbf{p}$ is the neighbourhood center, $\mathbf{t}_s$ is the initial direction of the current spoke (a unit tangent vector), $\mathbf{n}$ is the surface normal at $\mathbf{p}$, and $\mathbf{b}_s = \mathbf{t}_s \times \mathbf{n}$.

3.4. Signal Correlation

It is often the case that features of various signals defined on a surface are correlated. For example, features of the texture on a face often correspond to features of the face's geometry. In order to capture this kind of feature correlation among signals, we can simply concatenate the vectors produced by each signal (this is equivalent to simply adding the neighbourhood difference produced by each, but is more amenable to acceleration). In order to accommodate weighting of each signal contributing to the similarity measurement, we typically do a global normalization of each signal sample based on their maximum possible values (note that elements of ω_{geom} are bounded by the neighbourhood extent). Thus, for a d-dimensional signal ω, a particular sample of that signal ω_i is a d-vector, each element of which is normalized to lie in the range $[0, 1/\sqrt{d}]$. The user can then easily choose relative weights for the contribution of each signal to the similarity metric, as any given signal, regardless of its dimension, will have a maximum sample difference of 1 under the L_2 norm.

3.5. Acceleration and Indexing

An important feature of geodesic fans for future applications is that they are amenable to accelerated search using standard indexing techniques. By arranging the geodesic fan samples into a long vector, searches over collections of geodesic fans may be accelerated with *kd*-trees or vector quantization. Instead of each position producing a single neighbourhood vector, however, it produces one for each possible discrete rotation (i.e., one per spoke). While this increases the size of the data structure by a constant factor, most such indexing structures produce at least logarithmic speedups, and thus remain useful. We can also use principal components analysis (PCA) to reduce neighbourhood vector dimension, further accelerating many techniques. We have successfully applied vector quantization, PCA, and *kd*-trees to geodesic fan search, achieving results similar to those made by the texture synthesis community [WL00, HJO*01].

For similarity-based modelling, indexing is not applicable. We still considerably accelerate geodesic fan comparisons with vector quantization on the *spokes* of the geodesic fans. We consider the set of samples on each spoke as a long vector, and apply tree-structured vector quantization (TSVQ) [GG92]. This effectively reduces the dimension of

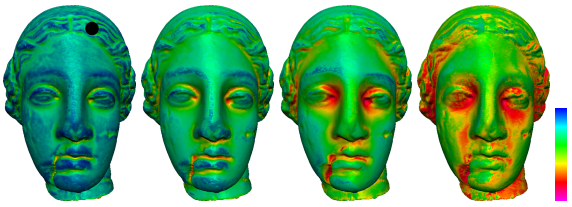

Figure 5: *Similarity maps for the indicated position (black) in the left image, with increasing neighbourhood extents. Similarity increases from red to blue, as in the ramp (right). From left to right, neighbourhood extents as a fraction of the bounding box diagonal are 0.01, 0.02, 0.06 and 0.125.*

each spoke to 1, as all pairwise differences of quantized spokes may be precomputed and stored in a lookup table. Fan-to-fan comparison time speedup is linear in the dimensional reduction; for example, geodesic fans whose spokes have 10 samples of a 3D signal are compared 30 times faster. For similarity-based modelling, which requires storing a geodesic fan for each vertex of a mesh, there is a similarly large reduction in memory usage as well. However, TSVQ imposes a small cost for constructing the quantization codebook, as well as for actually quantizing the spokes.

3.6. Implementation Issues

In practice, we restrict ourselves to triangulated manifold surfaces. Geodesic fans have three parameters: the maximum arclength of a spoke, the number of spokes, and the number of samples per spoke. Note that the number of spokes provides a tradeoff between the accuracy of our alignment approximation and the computation time. The maximum arclength determines the extent of the neighbourhood on the surface, and the number of samples per spoke influences the sample density. These parameters are typically selected in the same manner as the comparable parameters to image domain neighbourhoods: by visual inspection, such that the smallest features of interest are adequately sampled (i.e., such that one neighbourhood containing a smallest feature, and one identical but for the smallest feature, could be discriminated). For illustration, we show similarity maps over the Venus model in Figure 5 with increasing neighbourhood extents. Note that as more of the surface is taken into account, greater discrimination of neighbourhoods occurs (there is also relatively more noise due to lower sample densities, as the other parameters are held constant).

We can use arbitrarily-shaped neighbourhoods for matching using geodesic fans. For example, a user could paint on the surface to identify a region they wish to match against. Given such an arbitrary shape, we compute its centroid and bounding circle in the polar domain, setting the geodesic fan extent to contain this bounding circle. At comparison time, after a fan has been discretely rotated, we mask off any samples which are outside of the desired arbitrary shape and omit them from comparison. In practice, however, we have not found this to be very useful. The ability to arbitrarily orient the desired shape on a surface can be unintuitive, leading to unexpected matching results. It is likely that constraints need to be imposed on which discrete rotations of the geodesic fan should be considered in this case. Note that it is trivial, in general, to apply some set of constraints to which rotations of a fan are allowable, if so desired; for example, a vector field over the surface can supply a fixed orientation to each fan, or indicate axes of symmetry to be considered.

For mapping a sample position from geodesic polar coordinates (θ, r) to world coordinates, we first arbitrarily choose a world-space polar base, apply the rotation θ in the conformal plane, and cast a geodesic in the resulting direction, placing the sample at arclength r. There are numerous formulations for computing geodesics on surfaces [PS98, KS98, BMBZ02]. We use the formulation of Biermann *et al.*, as it is well-suited to these operations, correctly handles saddle points, and is quite efficient. Note that the quality of geodesic tracing and the geometry signal depends on the quality of the vertex normals, so very noisy surfaces may require smoothed normals (all examples shown use face area-weighted normals). If a geodesic hits a mesh boundary, the rest of its samples are marked unavailable and omitted from further matching. For our applications, this works well, though more sophisticated boundary handling could be required in other cases.

4. Similarity-Based Surface Modelling

Similarity-based surface modelling generalizes the image-based technique of Brooks and Dodgson [BD02] to surfaces. As a portion of the surface is modified, the system simultaneously reproduces the modification at all similar areas of the surface. As we are working with triangle meshes, we restrict editing to occur at the vertices of the surface.

In the next section, we describe our method for geometry editing in detail. We then describe simple extensions to allow similarity-based painting on surfaces, and similarity-driven filtering of surfaces. Finally, we discuss some benefits and drawbacks of our system.

4.1. Similarity-Based Deformation

For purposes of description, we restrict editing to occur one vertex at a time, although it should be evident that more sophisticated manipulation is possible using the same general technique.

We sample a geodesic fan at each vertex of the surface as pre-processing. The user then selects a vertex $\mathbf{p}$ to be edited, and we compute $D(\mathbf{p},\mathbf{q})$ for each vertex $\mathbf{q} \in M$, $\mathbf{q} \neq \mathbf{p}$. The user then selects a difference threshold α; vertices whose geodesic fan difference with the geodesic fan at $\mathbf{p}$ is greater than α will not be affected by the similarity-based edit. Finally, the user moves the vertex to a new position $\mathbf{p}_{new}$, and

the system automatically transports a similar displacement to each affected vertex $\mathbf{q}$ with $D(\mathbf{p},\mathbf{q}) < \alpha$.

To transport this displacement, we transform the movement into the local neighbourhood frame given by the surface normal $\mathbf{n}_p$, polar base $\mathbf{s}_p$, and their mutual cross product $\mathbf{b}_p$. Then, at each affected vertex $\mathbf{q}$ we produce the similarly edited vertex $\mathbf{q}_{new}$ by transforming the displacement back to world space using the local matched neighbourhood frame; this frame (at $\mathbf{q}$) is given by the surface normal $\mathbf{n}_q$, the spoke $\mathbf{m}_q$ of the geodesic fan at $\mathbf{q}$ which corresponds to $\mathbf{s}_p$, and their mutual cross product $\mathbf{b}_q$ (note that this frame is different than the frame used in sampling ω_{geom}). The length of the displacement is also scaled by the similarity between $\mathbf{q}$ and $\mathbf{p}$. Following Brooks and Dodgson, we use a simple linear scaling according to the distance between $D(\mathbf{p},\mathbf{q})$ and α. Thus, we the new position of $\mathbf{q}$ is computed as:

$$\mathbf{q}_{new} = \mathbf{q} + \frac{\alpha - D(\mathbf{p},\mathbf{q})}{\alpha} [\mathbf{m}_q \mathbf{b}_q \mathbf{n}_q] \begin{bmatrix} \mathbf{s}_p \\ \mathbf{b}_p \\ \mathbf{n}_p \end{bmatrix} (\mathbf{p}_{new} - \mathbf{p}) \quad (4)$$

Note that while we have formulated this such that arbitrary displacements may be transported, neighbouring vertices on the surface will often have incoherent matched polar bases. Thus, a displacement with a significant tangential component may be transported in different directions at neighbouring vertices. More work remains to determine whether this incoherence is inherent to the problem or an artifact of our matching procedure; in any case, we expect some spatial relaxation of matched orientations can alleviate this problem. All deformation results presented here use displacements in the normal direction only. On a related note, we have sometimes found it helpful to use spatial (Laplacian) smoothing of the similarity *values* over the surface, as in some cases they can be undesirably noisy.

4.2. Similarity-based Surface Painting

Similarity-based painting may be achieved simply by transporting colour vectors rather than displacements. However, since colour vectors are unrelated to the local surface frame, we need not transform them through the local matching neighbourhood frames, and may just transport them directly. Thus, if ω_c is the colour signal, then (4) simplifies in the case of similarity-based surface painting to:

$$\omega_c(\mathbf{q}) \leftarrow \omega_c(\mathbf{q}) + \frac{\alpha - D(\mathbf{p},\mathbf{q})}{\alpha} \omega_c(\mathbf{p}) \quad (5)$$

where $\omega_c(\mathbf{p})$ is the new painted colour at $\mathbf{p}$.

4.3. Similarity-based Filtering

We may also drive filtering operations according to similarity. Here, we have some filter which may be evaluated at positions on the surface to produce displacements, colour changes, or other signal changes. For example, a geometric smoothing filter produces a displacement of a vertex based on its neighbours' positions. With similarity-based filtering, we scale the changes produced by the filter according to the similarity of the unfiltered position with some base vertex. This allows very intelligent and intuitive localization of filtering operations. Note that this is effectively quite similar to similarity-based deformation or painting, except that the change at each vertex is produced computationally by the filter, rather than being interactively chosen by the user.

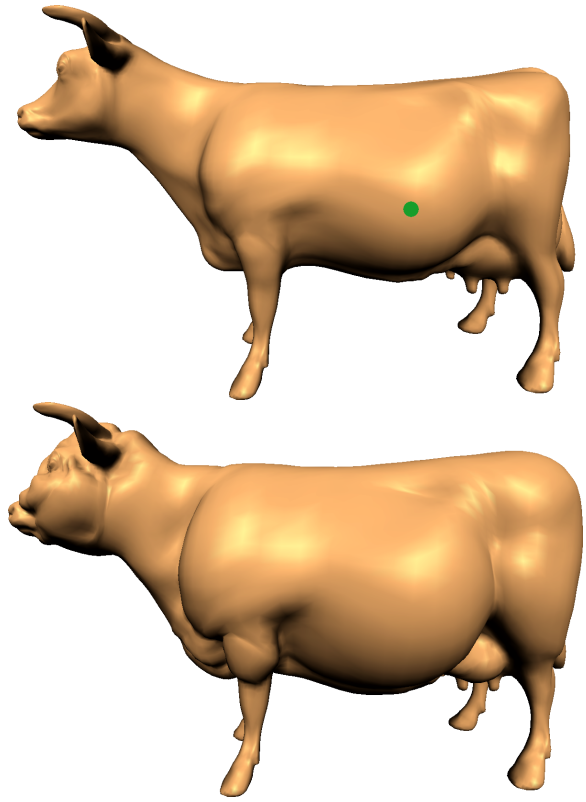

Figure 6: *A single similarity-based deformation to the cow (above, at green vertex) produces a much plumper version. As expected, the legs, horns and ears are left unchanged.*

The examples presented in this paper use a smoothing filter for simplicity, though other more complex filters may be used. Following Mokhtarian *et al.* [MKY98], we compute the new position of each vertex as the weighted average of the sample positions in its geodesic fan, where weighting is a Gaussian function of the geodesic distance from the vertex.

5. Results and Discussion

We demonstrate a range of similarity-based deformations in Figures 1, 6, and 7. The deformations in each of these figures used ω_{geom} for similarity measurement. With the bunny, we emphasize the fur with increasing displacements, naturally and intuitively modifying all of the fur simultaneously, while

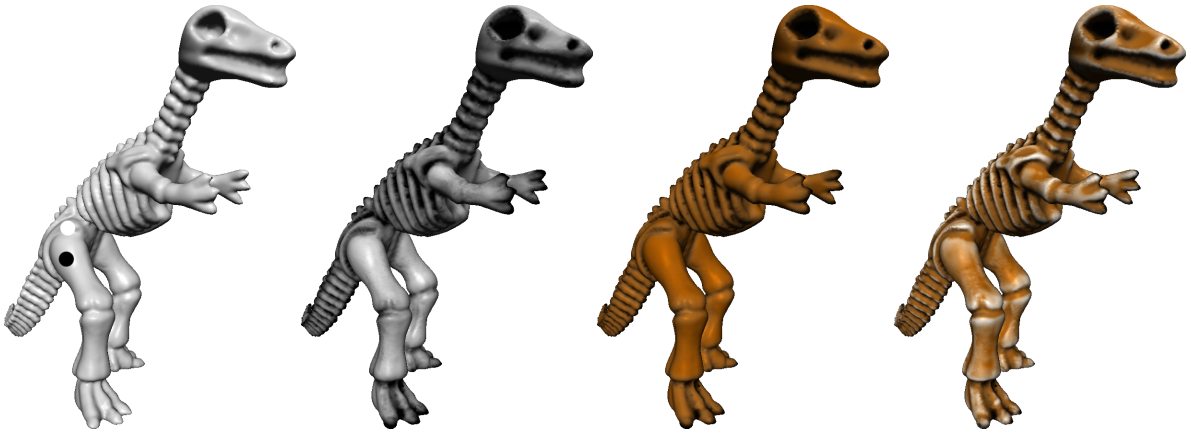

Figure 9: *Inverting the similarity-based opacities used for painting distributes black paint to all areas dissimilar to the smooth position on the leg of the dinosaur. Brown is distributed elsewhere by inverting opacities again, while white highlights are produced by one more edit. Since opacity varies with similarity, the resulting colours blend well over the surface.*

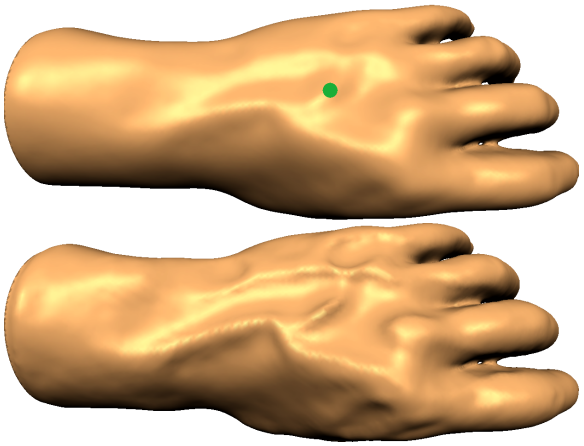

Figure 7: *Our geometry measurement is sensitive enough to identify and emphasize only the veins of this very smooth hand model with one similarity-based deformation.*

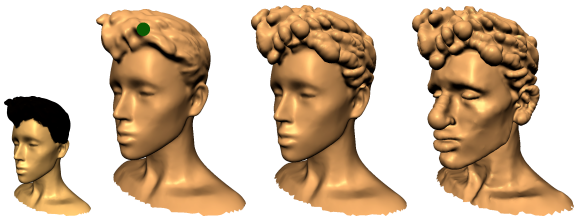

Figure 8: *Correlating colour (left) with geometry (middle left) in the similarity measure emphasizes the mannequin's hair while preventing changes to its face (middle right), as would occur using only geometric similarity (right).*

leaving the smoother tail, feet, and head mostly unchanged. In contrast, one edit on a smooth region of the cow increases its apparent bulk significantly without affecting any of its detailed geometry, such as the legs, horns, ears, or udders. We should note, however, that the geometry of the tail becomes enveloped by the cow's backside, as can be seen in the video, as our system does not attempt to prevent self-intersections when deforming the mesh. The hand example shows the sensitivity of our geometric measurement; we are able to pick out the veins of the hand model for emphasis, even though the model is very smooth to begin with.

The control affording by signal correlation within our framework is demonstrated in Figure 8. The middle-right result, a deformation using correlated geometric and colour signals, emphasizes features of the mannequin's hair, while leaving the face untouched. The very same deformation applied using only geometric similarity is shown at right. Since portions of the face such as the nose and lips match the modified lock of hair geometrically, they undergo deformation as well. While the result remains plausible, the character of the face has been changed significantly. Our framework lets the user decide the correct result.

We may also use colours alone to drive the similarity-based deformation, as in Figure 10. Here, the depression created in a groove of the bark texture is automatically applied to all such grooves on the model, effectively embossing the model with the selected texture feature. The effect, while subtle in still images, is quite convincing in animation, especially compared to the original smooth model as we demonstrate in the video. Note that somewhat similar results have been obtained by interpreting a texture synthesized over a surface as a displacement map [YHBZ01]; however, our technique allows the displacement to be based on arbitrary texture features, and not simply those which have grey levels easily distinguishable from the rest of the texture.

Figure 10: *The user depresses a single vertex in the groove of the bark texture. With similarity based on texture features rather than geometry, the texture pattern is stamped into the surface, producing a much more convincing textured mesh.*

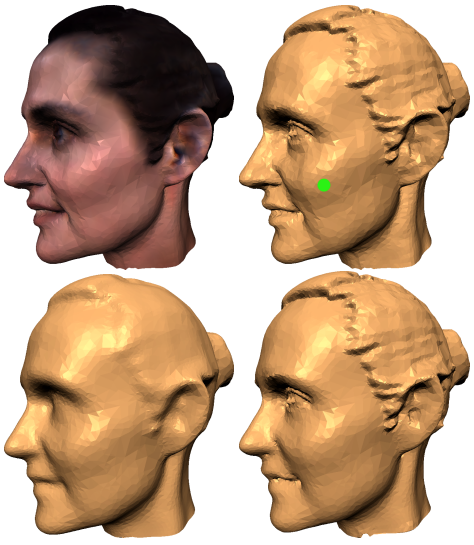

Figure 11: *Smoothing is limited to the model's face with similarity-based filtering. Using colours for similarity and comparing against the highlighted position on the cheek prevents the filter from modifying the geometry of the hair.*

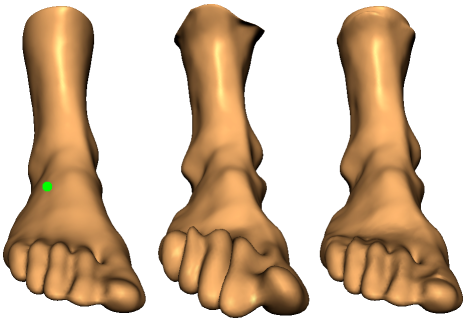

Figure 12: *A frequency enhancement filter produces global changes in a foot model. Similarity-based filtering naturally preserves the toes (middle right).*

An example of the power of similarity-based painting is given in Figure 9 (this example uses geometric similarity). We first distribute black paint to all the nooks and crannies of the model using *dissimilarity*. Note that while these features are geometrically fairly diverse, one thing they all have in common is that they are concave and not smooth. Thus, selecting a smooth convex region, such as on the leg, produces low similarity with these areas. Inverting the usual opacities generated by our similarity-based painting system (equation 5) distributes black paint according to the degree of geometric difference. The second edit uses the usual (non-inverted) opacities to distribute shades of brown to the rest of the model. Finally, we highlight strong edges of the model in white with one more painting operation (to the white-highlighted vertex in the original model). Thus, we produce the coloured dinosaur at right with only three similarity-based edits.

Finally, we demonstrate similarity-based filtering in Figure 11 (this figure is flat-shaded, to highlight the geometric differences). Normally, a uniform smoothing filter applied to the model destroys features all over the model (bottom left). More recent feature-sensitive smoothing methods based on bilateral filtering [JDD03, FDCO03] are sensitive to geometric features, but identify them only according to a geometric smoothness metric. Here, we exercise control over a general smoothing filter according to colour similarity. By selecting a point on the cheek, we are able to limit smoothing to occur only on the face of the model, leaving the geometry of the hair untouched. While the bilateral smoothing methods could use our similarity metric for the signal they measure

(instead of their geometric smoothness functions), our approach is more amenable to interaction; the general filter is only evaluated over the surface once, and the changes it produces are themselves filtered afterward according to similarity. Thus, the similarity parameters (threshold, base position) may be chosen interactively to produce the best result. Figure 12 shows a global frequency enhancement filter [GSS99] intuitively directed only whether the user desires using similarity-based filtering. Pairing multiresolution signal processing with similarity-based filtering yields a flexible, powerful system for surface modelling.

For the models we have shown, the bulk of computation time is in pre-computing geodesic fans and the related geodesic marching. Geodesic fans are constructed at the rate of 500-2500 fans per second on the models shown, which range from 5000-50000 vertices. Parameters used were: neighbourhood extents of 1-4% of the length of the model's bounding box diagonal, 10-25 spokes per geodesic fan, and 5-15 samples per spoke. After a change is made to the surface, neighbourhoods may require resampling if any sampled signal was changed. We can perform 2500-10000 comparisons per second between pairs of geodesic fans without acceleration.

Application of spoke quantization yields impressive results. Computing a TSVQ codebook of 500 to 2000 codewords takes under two minutes for all examples shown. Quantitatively, relative geodesic fan comparison error from quantization (measured on the bunny model) averaged 0.027% (maximum 0.4%). Over 300000 spokes can be quantized per second, and so quantization has negligible impact on neighbourhood construction time. Further, both memory usage and fan to fan comparison time are in practice linearly improved according to the effective dimension reduction. Generally, 25000-200000 comparisons per second may be made between geodesic fans with quantized spokes, resulting in similarity map computation time under a second for all examples shown. All timings reported here are on a standard PC with a 1.5Ghz processor and 768MB RAM.

6. Conclusions and Future Work

We have presented a new framework for local comparison of surfaces using geodesic fans. Our method simultaneously aligns and compares surface neighbourhoods, and is easily accelerated using standard techniques. This framework allows us to draw on the large body of image domain research into modelling by example to produce powerful new tools for surface manipulation. We have presented one such extension to self-similarity based image editing, which generalizes to surfaces in numerous ways, taking advantage of the possibility of defining several common signals over surfaces with which to measure similarity. We have further extended the control and applicability of this technique by decoupling the signal being measured from the signal being affected by the similarity-based edit. We expect many more such extensions of image domain research are possible using geodesic fans, including parameterization-free texture transfer (generalizing texture synthesis to the case where the sample is a surface rather than an image), and surface analogies (reproducing sufficiently local transformations on new surfaces).

We expect a multi-resolution generalization of geodesic fans will prove useful in improving match quality and efficiency. We are exploring the degree to which generic spoke quantization codebooks, useful over classes of models, may be made by including spoke data from numerous models. This would significantly improve model preprocessing time, but may yield unacceptable quantization artifacts. We have not yet properly evaluated the accuracy of our alignment procedure. It should be possible to produce an error bound in our approximation given a frequency decomposition of the signal within the neighbourhood. Relatedly, the geodesic fan-based approximation could serve as a useful initial solution to finding the optimal non-discrete alignment of two neighbourhoods. Finally, we are interested in a possible volumetric generalization (casting spokes in space). This would not only remove the manifold surface requirement (necessary to evaluate geodesics), but would also properly consider collections of manifolds. Our surface-based sampling, for example, cannot distinguish a point on a single cable from a point on a bundle of topologically disconnected cables.

Acknowledgements

We would like to thank the anonymous reviewers for all of their helpful comments. This research was supported under a grant from the NSF (CCR-0086084).

References

[Ash01] ASHIKHMIN M.: Synthesizing natural textures. In *Proceedings of 2001 ACM Symposium on Interactive 3D Graphics* (North Carolina, Mar. 2001), ACM SIGGRAPH, pp. 217–226.

[BD02] BROOKS S., DODGSON N. A.: Self-similarity based texture editing. *ACM Transactions on Graphics 21*, 3 (2002), 653–656.

[BMBZ02] BIERMANN H., MARTIN I., BERNARDINI F., ZORIN D.: Cut-and-paste editing of multiresolution surfaces. *ACM Transaction on Graphics 21*, 3 (2002), 312–321.

[BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 4 (2002), 509–522.

[EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision* (1999), IEEE Computer Society, pp. 1033–1038.

[FDCO03] FLEISHMAN S., DRORI I., COHEN-OR D.: Bilateral mesh denoising. *ACM Transactions on Graphics 22*, 3 (2003), 950–953.

[FJP01] FREEMAN W. T., JONES T. R., PASZTOR E. C.: *Example-based Super-resolution*. Tech. rep., Mitsubishi Electric Research Laboratories, July 2001. TR2001-30.

[FLCB95] FLEISCHER K. W., LAIDLAW D. H., CURRIN B. L., BARR A. H.: Cellular texture generation. In *Proceedings of SIGGRAPH 95* (1995), ACM SIGGRAPH, pp. 239–248.

[FMK*03] FUNKHOUSER T., MIN P., KAZHDAN M., CHEN J., HALDERMAN A., DOBKIN D., JACOBS D.: A search engine for 3d models. *ACM Transactions on Graphics 22*, 1 (Jan. 2003), 83–105.

[GG92] GERSHO A., GRAY R. M.: *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

[Goe70] GOETZ A.: *Introduction to Differential Geometry*. Addison-Wesley, 1970.

[GSS99] GUSKOV I., SWELDENS W., SCHRÖEDER P.: Multiresolution signal processing for meshes. In *Proceedings of SIGGRAPH 99* (1999), pp. 325–334.

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *Proceedings of SIGGRAPH 2001* (2001), ACM SIGGRAPH, pp. 327–340.

[HK03] HAMZA A. B., KRIM H.: Geodesic object representation and recognition. In *DGCI 2003* (2003), Nystroöm I., de Baja G. S.,, Svensson S., (Eds.), vol. 2886 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 378–387.

[HOCS02] HERTZMANN A., OLIVER N., CURLESS B., SEITZ S. M.: Curve analogies. In *Proceedings of the 13th Eurographics Workshop on Rendering* (Pisa, Italy, June 2002), pp. 233–245.

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of SIGGRAPH 2001* (2001), ACM SIGGRAPH, pp. 203–212.

[JDD03] JONES T. R., DURAND F., DESBRUN M.: Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics 22*, 3 (2003), 943–949.

[JH99] JOHNSON A. E., HEBERT M.: Using spin-images for efficient multiple model recognition in cluttered scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 5 (1999), 433–449.

[KPNK03] KÖRTGEN M., PARK G.-J., NOVOTNI M., KLEIN R.: 3d shape matching with 3d shape contexts. In *Proceedings of the 7th Central European Seminar on Computer Graphics* (2003).

[KS98] KIMMEL R., SETHIAN J.: Computed geodesic paths on manifolds. *Proceedings of the National Academy of Sciences 95*, 15 (July 1998), 8431–8435.

[LDG01] LEGAKIS J., DORSEY J., GORTLER S. J.: Feature-based cellular texturing for architectural models. In *Proceedings of SIGGRAPH 2001* (2001), ACM SIGGRAPH, pp. 309–316.

[LSZ01] LIU C., SHUM H.-Y., ZHANG C.-S.: A two-step approach to hallucinating faces: Global parametric model and local non-parametric model. In *IEEE Conference on Computer Vision and Pattern Recognition* (2001), IEEE, pp. 192–198.

[MK03] MAGDA S., KRIEGMAN D.: Fast texture synthesis on arbitrary meshes. In *Proceedings of the Eurographics Symposium on Rendering 2003* (2003), Eurographics Association, pp. 82–89.

[MKY98] MOKHTARIAN F., KHALILI N., YUEN P.: Multi-scale 3d free-from surface smoothing. In *Proceedings of the Ninth British Machine Vision Conference* (1998), British Machine Vision Association, pp. 730–739.

[PS98] POLTHIER K., SCHMIES M.: Straightest geodesics on polyhedral surfaces. *Mathematical Visualization* (1998), 135–150.

[SCA02] SOLER C., CANI M.-P., ANGELIDIS A.: Hierarchical pattern mapping. *ACM Transactions on Graphics 21*, 3 (2002), 673–680.

[TSK97] TAKAHASHI S., SHINAGAWA Y., KUNII T. L.: Curve and surface design using multiresolution constraints. In *Proceedings of Computer Graphics International '97* (1997), IEEE, pp. 121–130.

[Tur01] TURK G.: Texture synthesis on surfaces. In *Proceedings of SIGGRAPH 2001* (2001), ACM SIGGRAPH, pp. 347–354.

[TZL*02] TONG X., ZHANG J., LIU L., WANG X., GUO B., SHUM H.-Y.: Synthesis of bidirectional texture functions on arbitrary surfaces. *ACM Transactions on Graphics 21*, 3 (2002), 665–672.

[WAM02] WELSH T., ASHIKHMIN M., MUELLER K.: Transferring color to grayscale images. *ACM Transactions on Graphics 21*, 3 (2002), 277–280.

[WL00] WEI L.-Y., LEVOY M.: Fast texture synthesis using tree-structured vector quantization. In *Proceedings of SIGGRAPH 2000* (2000), ACM SIGGRAPH, pp. 479–488.

[WL01] WEI L.-Y., LEVOY M.: Texture synthesis over arbitrary manifold surfaces. In *Proceedings of SIGGRAPH 2001* (2001), ACM SIGGRAPH, pp. 355–360.

[WW94] WELCH W., WITKIN A.: Free-form shape design using triangulated surfaces. In *SIGGRAPH '94 Proc.* (1994), ACM, pp. 247–256.

[YHBZ01] YING L., HERTZMANN A., BIERMANN H., ZORIN D.: Texture and shape synthesis on surfaces. In *Proceedings of the Twelfth Eurographics Workshop on Rendering Techniques* (2001), Eurographics Assocation, pp. 301–312.

[ZG03] ZELINKA S., GARLAND M.: Interactive texture synthesis on surfaces using jump maps. In *Proceedings of the Eurographics Symposium on Rendering 2003* (2003), Eurographics Association, pp. 90–96.