

Analysis of Appearance Space Attributes for Texture Synthesis and Morphing

Felix Manke

Graphics Group, Department of Computer Science
University of Auckland,
Auckland, New Zealand,
Email: fman020@aucklanduni.ac.nz

Burkhard C. Wünsche

Graphics Group, Department of Computer Science
University of Auckland,
Auckland, New Zealand,
Email: burkhard@cs.auckland.ac.nz

Abstract—Texture synthesis and morphing are important techniques for efficiently creating realistic and visually attractive textures. A popular class of synthesis algorithm are pixel-based techniques, which search in a given 2D exemplar for a pixel with a similar neighbourhood to the pixel currently being generated. The methods have the advantage that they are fast, they can be easily generalised to higher dimensions, and synthesised textures can be represented as references to the exemplar which allows definition of additional channels, such as displacement maps, at no additional cost.

The quality of pixel-based techniques depends on the metric used to compare pixel neighbourhoods. Lefebvre and Hoppe introduced the term appearance space for measures describing pixel neighbourhood similarity. In this paper we introduce new appearance space attributes and evaluate them and existing attributes for texture synthesis and morphing. Our results show that our proposed gradient estimate significantly improves synthesis and morphing quality with little additional cost.

I. INTRODUCTION & BACKGROUND

Texture synthesis and morphing are essential in computer graphics for creating visually rich textures in a controlled manner. Various concepts have been explored in the literature, including procedural techniques [1], [2], parametric exemplar-based methods [3], [4], and patch-based methods [5]–[7]. One of the most popular approaches are pixel-based methods [8]–[15]. This class of algorithms generates a new texture from a given exemplar texture by searching in it for a pixel with a similar neighbourhood to the pixel being generated. The search can be sped up by using k-coherence search [16] and a multi-resolution approach where synthesis is performed from the coarsest to the finest resolution, first establishing low frequencies and then defining fine details [12].

The texture synthesis can greatly benefit from using an *appearance space*, where pixels encode texture characteristics in addition to colour [17]. The high-dimensional appearance vectors are projected into a low-dimensional space defined by the first n principal components obtained from a principal component analysis (PCA). The appearance space is conceptually identical to the “feature vectors” proposed by Hertzmann et al. [18], although the integrated attributes differ. Hertzmann et al. use luminance information as well as the responses to steerable filters, but neither include neighbourhood information nor the signed feature distance. In previous

research we extended the technique to morphing by weighting neighbourhood matching results for multiple exemplars [15].

Synthesis and morphing quality depend on the chosen appearance space attributes. In this paper we introduce new appearance space attributes and evaluate them and existing attributes. Section II introduces the appearance space attributes. Section III investigates their effect on texture synthesis and morphing. The results of this analysis are presented in section IV. We conclude this paper and make suggestions for future research in section V.

II. DEFINITION OF APPEARANCE SPACE ATTRIBUTES

Texture synthesis is improved by introducing new attributes in addition to colour for testing neighbourhood similarity [17]. Some of the attributes presented below are based on the intensity image I of an three-channelled exemplar E :

$$I(P) = \frac{C_r + C_g + C_b}{3}, \quad (C_r \ C_g \ C_b)^T = E(P), \quad (1)$$

with $P \in I$ indexing a pixel. Textures can exhibit anisotropically shaped features, such as elongated blobs. When synthesizing or morphing textures these shape characteristics should be maintained. We capture them using contour extraction and contour sampling. Given a binary feature mask M that describes the dominant features of a corresponding exemplar texture, we extract features by tracking the contours of disjoint components. Our method is based on the *crack code encoding* of object boundaries [19]. In order to further optimise the detected contours, we remove “jaggies” that occur due to the four-neighbourhood. Figure 1 shows an example of the contour extraction for an actual binary feature mask. Note that the binary mask is mirrored, as it is also done during the synthesis in order to ensure well-defined neighbourhoods outside the exemplar.

In order to obtain information about feature shapes for any pixel in the exemplar, the surrounding feature has to be found. However, features tend to be sparse, so that most pixels lie outside of encoded features. For this reason, we sample the surrounding contour boundaries by radially casting rays. The closest intersection point of each ray with any contour is stored in a set of sample points. In this way, pixels do not have to lie within features for obtaining information about

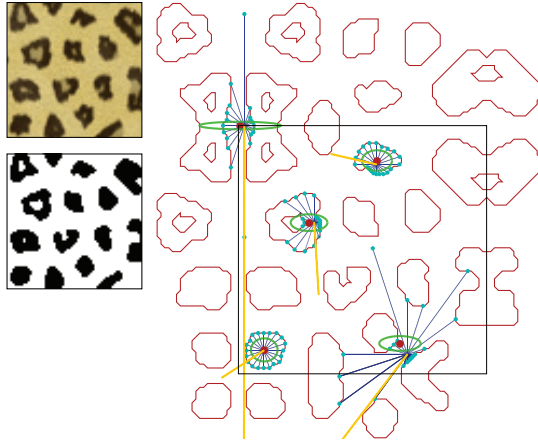


Fig. 1. The contours of a binary feature mask of an actual texture. Left: The exemplar and its binary feature mask. Right: The extracted contours, the result of contour sampling for five selected pixels, and the illustration of derived appearance space attributes (see section II). 20 rays are used to radially sample the region surrounding each pixel. The sample points are shown in cyan. The lines in yellow indicate the direction of the first principal axis (that is, the direction of maximal variance). The green ellipses represent the elongation of the point sets. The red dots depict the centroids of the sample points. Note that the binary mask is mirrored. The area in the black box corresponds to the original binary mask.

the surrounding area. In addition, holes within features can be taken into account. Figure 1 shows some examples of the sampling. Note that the maximal distance of a hit point is restricted to twice the average contour diameter in order to avoid extremely high variance due to scattering of sample points.

We define the following appearance space attributes. The corresponding textures \mathbf{A} contain only one single appearance space attribute and are used in the next section for analysing the attribute's effect on texture synthesis and morphing.

RGB Colour: The RGB colour triples of the input exemplar, so that:

$$\mathbf{A}(P) = \mathbf{E}(P).$$

Signed Feature Distance: The distance from pixel P to the closest pixel $Q \in \mathbf{M}$ for which $\mathbf{M}(P) \neq \mathbf{M}(Q)$. We define the following distance equation:

$$\begin{aligned} \mathbf{A}(P) &= s \cdot \left(\max_{U \in \mathbf{M}} (\delta(U)) - \delta(P) \right), \\ \delta(X) &= \frac{1}{|Q_X - X|^k}, \\ s &= \begin{cases} 1 & \text{if } \mathbf{M}(P) = 1 \\ -1 & \text{if } \mathbf{M}(P) = 0. \end{cases} \end{aligned} \quad (2)$$

The vector $(Q_X - X)$ points from X to the closest pixel Q_X with a different value in \mathbf{M} . The exponent k serves as control parameter that stretches or compresses the distance curve. With $k > 1$, the curve converges faster to the maximum.

Gradient Estimates: The partial derivatives of \mathbf{I} , $(\frac{\partial \mathbf{I}}{\partial x}, \frac{\partial \mathbf{I}}{\partial y})$, approximated by using the convolution kernels of a standard

Sobel operator. Given the two kernels, $S_x = (1 \ 2 \ 1)^T \cdot (1 \ 0 \ -1)$ and $S_y = (1 \ 0 \ -1)^T \cdot (1 \ 2 \ 1)$, the attribute image is defined as:

$$\mathbf{A}(P) = ((S_x * \mathbf{I})(P), (S_y * \mathbf{I})(P)), \quad (3)$$

where the operator $*$ denotes a convolution.

Steerable Filter Responses: The responses of \mathbf{I} to oriented first derivatives of a Gaussian distribution G . Using steerable filter responses as appearance space attribute is inspired by the attributes Hertzmann et al. [18] suggest for their “feature vectors” (although the authors use a full steerable pyramid of oriented third derivatives). The oriented first derivative of the Gaussian function is given by Freeman and Adelson [20]:

$$G'_\theta = \cos(\theta) \cdot \frac{\partial G}{\partial x} + \sin(\theta) \cdot \frac{\partial G}{\partial y},$$

where θ is the rotation angle in radians. Given G'_θ , the filter response of \mathbf{I} can be computed using the convolution $G'_\theta * \mathbf{I}$. For n rotation angles, the attribute image is given by:

$$\mathbf{A}(P) = ((G'_{\theta_0} * \mathbf{I})(P), (G'_{\theta_1} * \mathbf{I})(P), \dots, (G'_{\theta_{n-1}} * \mathbf{I})(P)),$$

$$\theta_i = \frac{i \cdot \pi}{n}, \quad i \in [0, n).$$

Neighbourhood Variance: The variance σ^2 of vectorised neighbourhoods in \mathbf{I} . For a set of neighbourhood offsets \mathcal{N} , the attribute image is given by:

$$\mathbf{A}(P) = \sigma^2(\mathcal{N}, P) = \frac{1}{|\mathcal{N}|} \cdot \sum_{\Delta \in \mathcal{N}} (\mathbf{I}(P + \Delta) - \mu(\mathcal{N}, P))^2,$$

where $|\mathcal{N}|$ is the number of elements in the neighbourhood, and $\mu(\mathcal{N}, P)$ gives the mean of the whole neighbourhood around P .

Feature Shape — First Principal Axis: The first eigenvector \vec{e}_1 of the feature shape sample points around P , scaled by the first eigenvalue λ_1 (\vec{e}_1 and λ_1 are obtained by performing a principal component analysis (PCA) on the feature shape sample points). The two-channelled attribute image is defined as:

$$\mathbf{A}(P) = \lambda_1(P) \cdot \vec{e}_1(P).$$

The first principal axis defines an axis of maximal variance through the set of sample points, and therefore characterises the orientation of the sample points. Figure 1 illustrates the first principal axis for selected pixels by yellow lines.

Feature Shape — Elongation: The ratio of the first and second eigenvalue, λ_1/λ_2 , as a measure of elongation and shape of the sampled surrounding. Thus, the attribute image is given by:

$$\mathbf{A}(P) = \frac{\lambda_1(P)}{\lambda_2(P)}.$$

In figure 1, elongation is illustrated by green ellipses.

Feature Shape — Centroid Offset: The x - and y -component of the displacement vector from the pixel P to the centroid \check{c} of the feature shape sample points, such that:

$$A(P) = \check{c} - P, \quad \check{c} = \frac{1}{|S(P)|} \cdot \sum_{Q \in S(P)} Q,$$

where $S(P)$ depicts the set of feature shape sample points around P . The displacement of the centroid depends on how close a pixel is to a feature boundary. The red dots in figure 1 depict the centroids \check{c} of the set of sample points.

Of the defined attributes, Lefebvre and Hoppe [17] proposed the first two. However, our signed feature distance defined in equation 2 is more flexible than theirs as it allows adjusting the distance curve using the control parameter k . The use of steerable filter responses is inspired by the attributes Hertzmann et al. propose for their “feature vectors” [18], although the authors use different steerable filters.

III. ANALYSIS OF APPEARANCE SPACE ATTRIBUTES

This section presents an evaluation of the effectiveness and suitability of the appearance space attributes described above. The results of the analysis allow the reader to maximise texture synthesis and morphing performance with respect to both speed and visual quality.

The definition of some of the proposed attributes includes parameters that adjust the behaviour of the function (see section II). We used the following settings: For the signed feature distance we set $k = 2$. For the steerable filter responses, we used a Gaussian filter kernel of size 7×7 , and four steerable filters at angles $\theta \in \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$. For the neighbourhood variance, we used a full 7×7 neighbourhood. For the attributes based on feature shape information (first principal axis, elongation, and centroid offset), we used 50 rays for radially sampling the feature contours.

	μ	σ	σ_{norm}
<i>RGB</i> colour	2.167E-4	1.665E-5	0.0725
Signed feature distance	0.0518	0.0212	0.1861
Gradient estimates	0.0163	1.159E-4	6.643E-3
Steerable filter responses	0.0871	5.557E-4	6.068E-3
Neighbourhood variance	0.024	2.25E-4	7.965E-3
First principal axis	167.3237	122.4592	0.2093
Elongation	167.3403	122.3929	0.2097
Centroid offset	167.2188	122.4365	0.2092

TABLE I

AVERAGE COMPUTATION TIMES FOR DIFFERENT APPEARANCE SPACE ATTRIBUTES, BASED ON 32 EXEMPLARS OF SIZE 128×128 . THE TABLE SHOWS THE MEAN μ AND THE STANDARD DEVIATION σ IN SECONDS, AS WELL AS THE STANDARD DEVIATION σ_{norm} OF THE NORMALISED TIMINGS (THAT IS, THE MEASURED TIMINGS DIVIDED BY THE LARGEST MEASURED TIMING PER ATTRIBUTE).

In order to test the appearance space attributes we compute the appearance vectors of the exemplars with each attribute alone. The results were generated with the algorithm we presented in [15] and are shown in figures 2–5. For comparison we included a reference run using the a combination of *RGB* colour, signed feature distance, and gradient estimates (since

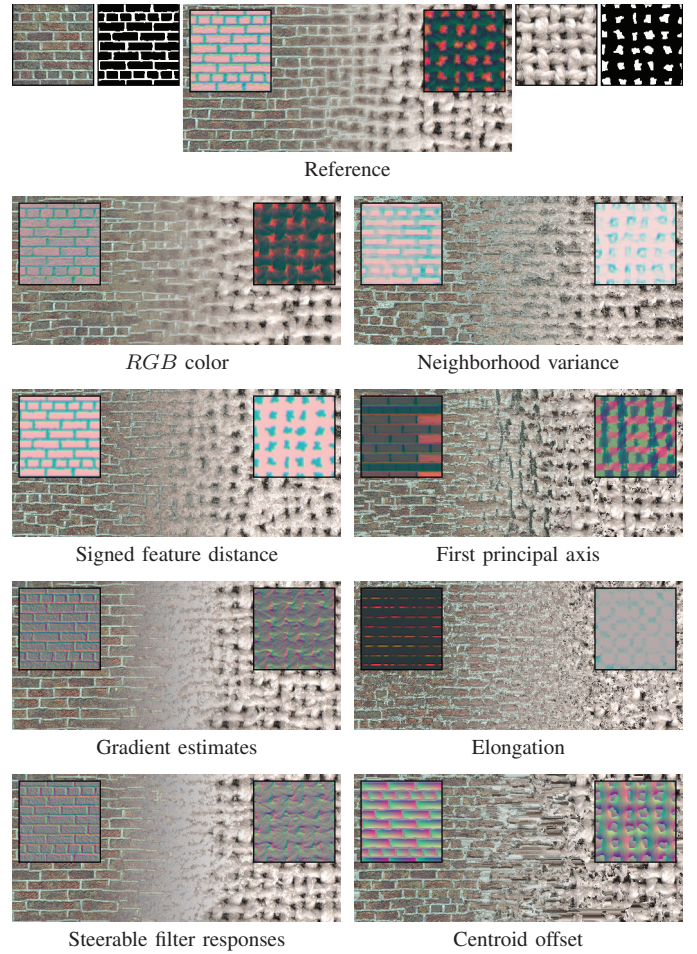


Fig. 2. Morphing results for the single appearance space attributes. The top row shows the exemplars, feature masks, and the reference result. The insets show the first three dimensions of the projected appearance space mapped into *RGB* colour space.

this proved subsequently to be the best combination). Note that the employed exemplars range from near-regular over irregularly structured to near-stochastic textures. Table I shows the average duration and standard deviation for computing the attributes.

With respect to the characteristics of the input exemplars, we can observe that the performance of the synthesis and morphing algorithm increases as the input exemplars become more structured or regular. The algorithm relies on distinct features that yield well distinguishable local neighbourhoods, which are not present in the stochastic exemplars selected for the test.

The examples show that the *RGB* colour alone is not capable of preserving features within the exemplars. Features “melt” into each other, which causes blurring and smoothing (particularly noticeable in figure 4). In fact, our results with *RGB* colour show similarities to the outputs of Wei’s early morphing algorithm, which only compares colours [21].

With respect to the preservation and morphing of dominant features, the signed feature distance produces the best results among the tested attributes for all exemplars. While other

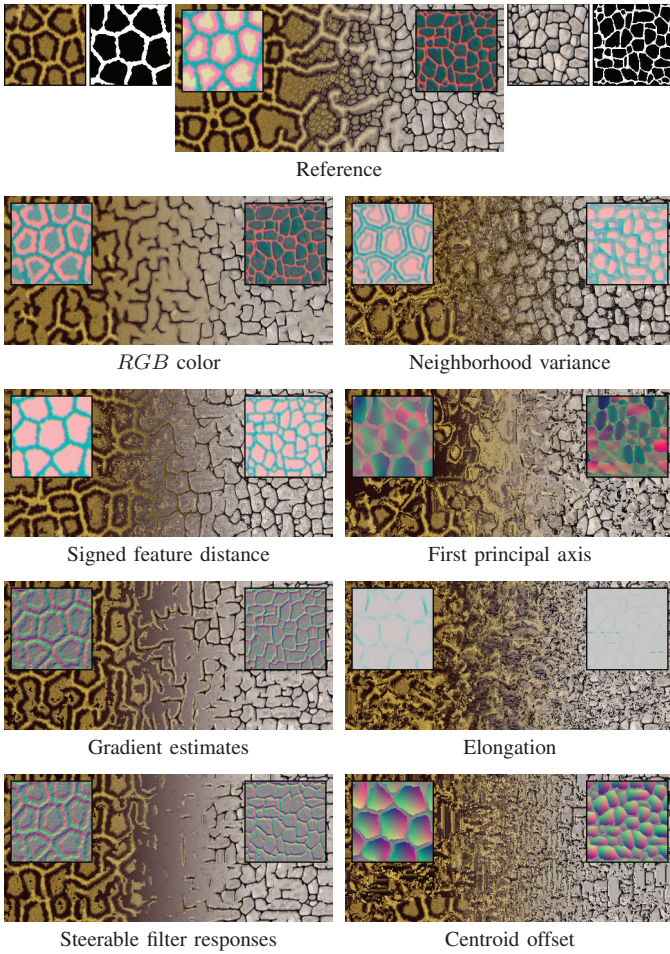


Fig. 3. Morphing results for the single appearance space attributes. The top row shows the exemplars, feature masks, and the reference result. The insets show the first three dimensions of the projected appearance space mapped into *RGB* colour space.

attributes tend to lose features in the transition region, the signed feature distance is capable of morphing the structures between different materials (for example, see figures 4 and 5). This is highly desirable, particularly for more complex texture morphing problems.

Despite its superior morphing capabilities, the signed feature distance tends to produce a high-frequency noise, noticeable as fine-grained speckles (see figures 2 and 5). In comparison, the results with *RGB* colour as single attribute look much smoother. The noise can be explained by small synthesis errors within features, for which the binary feature mask does not provide any neighbourhood information.

Gradient estimates and steerable filter responses perform similarly, since the two attributes are closely related. However, gradient estimates preserve features slightly better and can be computed more efficiently. As illustrated in table I, computing the four steerable filter responses takes more than five times as long as computing gradient estimates.

The neighbourhood variance performs best if the highest variance is clustered around the texture features (for example, see figure 3). In these cases, the projected appearance space

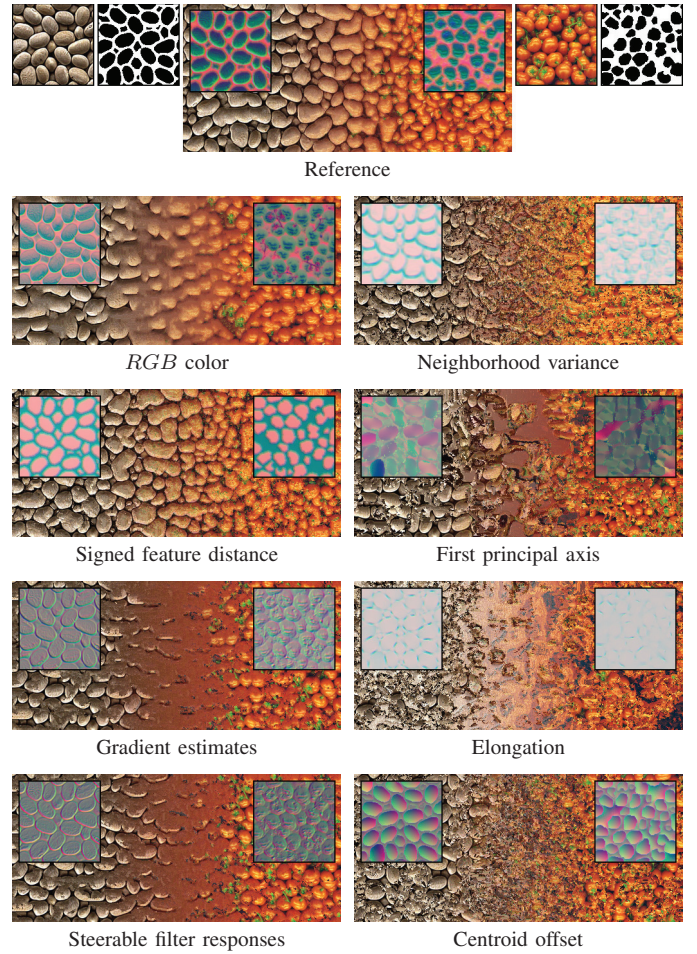


Fig. 4. Morphing results for the single appearance space attributes. The top row shows the exemplars, feature masks, and the reference result. The insets show the first three dimensions of the projected appearance space mapped into *RGB* colour space.

looks similar to the projected appearance space of the signed feature distance. However, the neighbourhood variance, by taking the intensity image as basis, is noisier than the signed feature distance, which is based on binary feature masks. If there is significant variance within features, the results for neighbourhood variance are not convincing. The example shown in figure 4 illustrates the difference between clustered high variance (left exemplar) and more uniformly distributed variance (right exemplar).

The morphing results of the attributes based on feature shape information (first principal axis, elongation, and centroid offset) are unexpectedly poor. Elongation fails in all cases. The other two attributes perform only marginally better (for example in figure 2, where near-regular textures are morphed). The inset images of the projected appearance space suggest that elongation provides very little information. For the first principal axis, the appearance space looks very unstructured and chaotic, which could explain the poor performance of this attribute.

The standard deviation of the measured timings (table I) for signed feature distance and the three attributes that depend

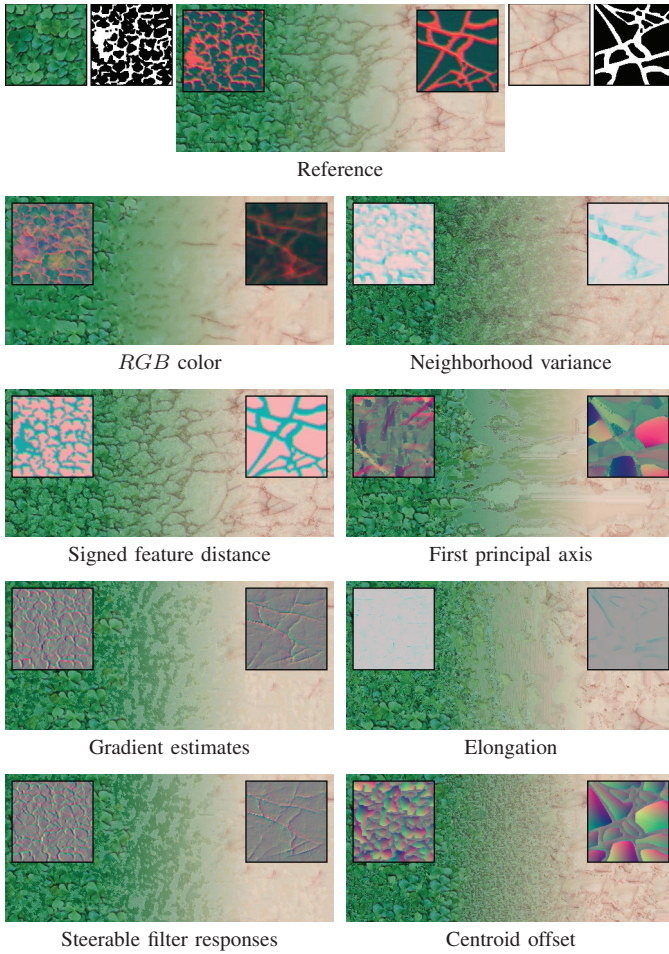


Fig. 5. Morphing results for the single appearance space attributes. The top row shows the exemplars, feature masks, and the reference result. The insets show the first three dimensions of the projected appearance space mapped into *RGB* colour space.

on the feature shape information is much higher than for the other attributes (about 3–35 times as high). This fact can be explained by the dependence on exemplar features. For the signed feature distance, sparse binary feature masks result in longer radial searches for the nearest feature boundary. For the feature shape information, the contour sampling takes longer the more features are present. Because the other attributes do not depend on the characteristics of the input texture, their computation time shows little variation among different exemplars.

IV. RESULTS

Our analysis confirms that the *RGB* colour and signed feature distance proposed by Lefebvre and Hoppe [17] are very powerful appearance space attributes and essential for successful texture synthesis and morphing. The above results suggest that gradient estimates can improve the structure of synthesised and morphed textures.

We tested this hypothesis by comparing morphing results obtained using *RGB* colour and signed feature distance with results obtained by adding additionally our gradient estimate.



Fig. 6. Comparison of 2D texture morphing without (left) and with (right) partial derivatives as additional appearance space attribute. The insets show the two input exemplars and the first three dimensions of the projected appearance space mapped into *RGB* colour space.

When integrating the new attribute, the synthesis and morphing result improved in almost every case as illustrated in figure 6. Note how both the synthesis in regions with only one texture as well as the morphing in the transition region between the two textures improve when including partial derivatives as additional appearance space attribute. When comparing the visual (colour-encoded) representation of the projected appearance space of the two exemplars, more structure and emphasis of feature boundaries can be noticed if partial derivatives are included.

We also tested various combinations of the above three attributes with feature shape information, but no visually significant improvement could be observed.

The fact that all three attributes that depend on feature shape information fail to produce convincing results suggests that a more sophisticated approach is required. As detailed in section II the region surrounding a pixel is sampled by radially casting rays and intersecting them with extracted feature boundaries. The result is a set of sample points, from which the different appearance space attributes are derived. As illustrated in figure 1, the samples are sometimes very scattered and do not properly reflect the surrounding (for example, shown by the sample in the bottom-right corner). At the same time, sample points cluster at a feature boundary if a pixel is very close to that boundary. These clusters can alter the result of the PCA and influence the position of the centroid.

Ellipse fitting presents a potentially more robust alternative to sampling the surrounding features using rays. Various algorithms exist for fitting an analytical ellipse to a set of points (e.g., [22]). For a given pixel in the exemplar, the “visible” vertices of all surrounding feature boundaries could serve as a set of input data points for the fitting algorithm.

In order to avoid extremely elongated ellipses (particularly for pixels outside of encoded features, as illustrated in figure 1), the ratio between semi-major axis and semi-minor axis might have to be constrained. For deriving appearance space attributes, the orientation of the fitted ellipse as well as the ratio between semi-major axis and semi-minor axis could serve as basis.

A second alternative to the proposed three attributes could



Fig. 7. 3D texture morphing results with a target texture size of 128^3 voxels (top left) and 256^3 voxels.

be a metric based on shape matching (e.g., [23]). For example, the shape of the sampled contours could be matched against a representative set of pre-defined shapes. The best-matching shape (e.g. its index) together with the shape similarity could then serve as channels of the attribute image.

Since the synthesis of 3D textures from 2D exemplars is more complicated we tested the most promising combinations of appearance space attributes for 3D texture morphing. We found that a combination of *RGB* colour, signed feature distance and our gradient estimate was again superior. Examples of the texture morphing results are given in figure 7.

V. CONCLUSION & FUTURE WORK

The appearance space attributes *RGB* colour and signed feature distance, originally introduced by Lefebvre and Hoppe [17], are essential for obtaining good synthesis and morphing results. We found that integrating the partial derivatives of the intensity image (equation 1) of the input exemplar supports the texture morphing extremely well. The derivatives in *x*- and *y*-direction are approximated using the convolution masks of a standard Sobel operator (equation 3).

The results obtained by using feature shape attributes were disappointing. We conclude that even though feature shape attributes are useful from a theoretical perspective the presented attributes are not suitable for the types of textures tested and better attributes must be found.

Nevertheless, taking the shape of dominant structures in the exemplars into account is promising, and the proposed ellipse fitting or template shape similarity might yield more suitable attributes. In addition, numerous alternative appearance space attributes could be derived from image processing techniques not mentioned in this paper, for example channels of other colour spaces, responses to other filters, image statistics, and so forth. We look forward to explore these ideas in the future.

REFERENCES

- [1] G. Turk, "Generating textures on arbitrary surfaces using reaction-diffusion," in *Proceedings of SIGGRAPH '91*. New York, NY, USA: ACM Press, 1991, pp. 289–298.
- [2] S. Worley, "A cellular texture basis function," in *Proceedings of SIGGRAPH '96*. New York, NY, USA: ACM Press, 1996, pp. 291–294.
- [3] J. S. De Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *Proceedings of SIGGRAPH '97*. New York, NY, USA: ACM Press, 1997, pp. 361–368.
- [4] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Texture mixing and texture movie synthesis using statistical learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 2, pp. 120–135, 2001.
- [5] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of SIGGRAPH '01*. New York, NY, USA: ACM Press, 2001, pp. 341–346.
- [6] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," *ACM Transactions on Graphics (Proceedings of SIGGRAPH '03)*, vol. 22, no. 3, pp. 277–286, 2003.
- [7] K. Takayama, M. Okabe, T. Ijiri, and T. Igarashi, "Lapped solid textures: filling a model with anisotropic textures," in *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*. New York, NY, USA: ACM, 2008, pp. 1–9.
- [8] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of ICCV '99*. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1033–1038.
- [9] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proceedings of SIGGRAPH '00*. New York, NY, USA: ACM Press, 2000, pp. 479–488.
- [10] L.-Y. Wei, "Texture synthesis from multiple sources," in *SIGGRAPH '03: ACM SIGGRAPH 2003 Sketches & Applications*. New York, NY, USA: ACM Press, 2003, pp. 1–1.
- [11] M. Ashikhmin, "Synthesizing natural textures," in *Proceedings of I3D '01*. New York, NY, USA: ACM Press, 2001, pp. 217–226.
- [12] S. Lefebvre and H. Hoppe, "Parallel controllable texture synthesis," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*. New York, NY, USA: ACM Press, 2005, pp. 777–786.
- [13] J. Kopf, C.-W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski, and T.-T. Wong, "Solid texture synthesis from 2d exemplars," *ACM Transactions on Graphics (Proceedings of SIGGRAPH '07)*, vol. 26, no. 3, pp. (2.1)–(2.9), 2007.
- [14] Y. Dong, S. Lefebvre, X. Tong, and G. Drettakis, "Lazy solid texture synthesis," in *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)*, 2008.
- [15] F. Manke and B. C. Wünsche, "Fast spatially controllable 2d/3d texture synthesis and morphing for multiple input textures," in *Proceedings of the 4th International Conference on Computer Graphics Theory and Applications (GRAPP 2009)*, Feb. 2009, pp. 5–12.
- [16] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo, and H.-Y. Shum, "Synthesis of bidirectional texture functions on arbitrary surfaces," in *Proceedings of SIGGRAPH '02*. New York, NY, USA: ACM Press, 2002, pp. 665–672.
- [17] S. Lefebvre and H. Hoppe, "Appearance-space texture synthesis," *ACM Transactions on Graphics (Proceedings of SIGGRAPH '06)*, vol. 25, no. 3, pp. 541–548, 2006.
- [18] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 2001, pp. 327–340.
- [19] G. Wilson, "Properties of contour codes," *IEEE Proceedings Vision, Image and Signal Processing*, vol. 144, no. 3, pp. 145–149, Jun 1997.
- [20] W. Freeman and E. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.
- [21] L.-Y. Wei, "Texture synthesis by fixed neighborhood searching," Ph.D. dissertation, Stanford University, 2002.
- [22] P. L. Rosin, "Ellipse fitting by accumulating five-point fits," *Pattern Recognition Letters*, vol. 14, no. 8, pp. 661–669, 1993.
- [23] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.