

Texture Transfer Using Geometry Correlation

Tom Mertens
CSAIL – MIT[†]

Jan Kautz
University College
London

Jiawen Chen
CSAIL – MIT

Philippe Bekaert
Hasselt University
EDM[‡] – tUL[§]

Frédo Durand
CSAIL – MIT



Figure 1: The appearance of two texture-mapped models is transferred to a target model (the Bunny). We analyze the geometric features of the source and their correlation with texture. The source texture is transferred to the target mesh based on the correlation.

Abstract

Texture variation on real-world objects often correlates with underlying geometric characteristics and creates a visually rich appearance. We present a technique to transfer such geometry-dependent texture variation from an example textured model to new geometry in a visually consistent way. It captures the correlation between a set of geometric features, such as curvature, and the observed diffuse texture. We perform dimensionality reduction on the overcomplete feature set which yields a compact guidance field that is used to drive a spatially varying texture synthesis model. In addition, we introduce a method to enrich the guidance field when the target geometry strongly differs from the example. Our method transfers elaborate texture variation that follows geometric features, which gives 3D models a compelling photorealistic appearance.

1. Introduction

The visual richness of textures has long been recognized as critical to photorealism. Recent advances have enabled the synthesis of elaborate textures from example. However, for a large class of objects, texture can be quite simple locally, but may exhibit rich *variation* over the surface. For example, this is the case for weathered objects where dirt or corrosion tends to manifest itself differently over the surface, but it can also be caused by other aspects of an artifact’s fabrication and history. In addition, texture variation is often *correlated* with local geometric characteristics; dust tends to

accumulate in areas of low accessibility [Mil94], and corrosion often starts at exposed areas [DH96]. Instead of modeling the underlying processes that produce a particular texture variation, we empirically model the variation with statistical correlation. Our goal is not to simulate the physics of weathering processes but to reproduce the rich visual appearance of a textured object.

We take a *source* 3D mesh and its texture as an example. Given a *target* mesh, we synthesize a new texture that reproduces the variation and geometry correlation from the source (Fig. 1). Rather than seeking physical accuracy, we aim at reproducing a *visually consistent* transfer of the texture. This problem is related to the “texture-by-numbers” issue, as coined by Hertzmann et al. [HJO*01], in which a guidance field of scalar or vector values drives a texture synthesis algorithm. While much work has been dedicated to constrained synthesis [HJO*01, EF01] and to the ability to vary textures spatially [ZZV*03, LLH04, MZD05], little attention has been paid to the creation of the guidance field it-

[†] Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA

[‡] Expertise Centre for Digital Media – Wetenschapspark 2, Diepenbeek, Belgium

[§] transnationale Universiteit Limburg – School of Information Technology

self. In a nutshell, in the texture-by-numbers issue, previous work has addressed texture while we focus on the numbers.

Faithfully transferring the richness of real texture variation involves a number of challenges. First, we need to compile a list of geometric properties or *features* that are relevant to texture variation. Second, the number of possible features is too large and needs to be reduced to prevent the curse of dimensionality at the texture synthesis step. In addition, irrelevant elements in the guidance field could lead to spurious variations in the transferred texture. We introduce a dimensionality-reduction step that extracts the geometric features that are actually *correlated* with texture variation.

Because our approach leverages the geometry of the input and example 3D meshes to create rich texture variation, it can be challenging to transfer to a target mesh bearing little similarity with the source. To address this issue, we introduce a *feature matching* technique that transfers statistical characteristics of the source guidance field onto the target guidance field. While this step does not strictly respect the input geometry, it greatly increases the realism of the synthesized texture and also addresses missing data problems when the target contains features that are not present in the source guidance field.

The guidance field drives a constrained texture synthesis algorithm such as existing non-parametric “texture-by-numbers” approaches [HJO*01, EF01]. We also introduce a parametric variant based on the Heeger and Bergen model [HB95]. It lacks the ability to represent structured textures, but is superior in producing detailed variations in case of strong geometry correlation. Moreover, it is computationally less expensive and does not require parameter tweaking.

Our contributions can be summarized as follows:

- We capture the correlation between texture and geometry based on a set of surface descriptors. Using dimensionality reduction, we construct a parsimonious guidance field for constrained texture synthesis.
- We introduce a technique which matches the guidance field on the target object to the source guidance field.
- We extend Heeger and Bergen’s model [HB95] to synthesize texture over meshes by following a guidance field, as an alternative to existing non-parametric approaches.

1.1. Related Work

Our work builds on a large body of literature on texture and material appearance. In this section we focus on work most related to our goal.

Texture Synthesis over meshes. Synthesis of stationary textures on surfaces has been demonstrated by many authors [PFH00, Tur01, WL01, YHBZ01, SCA02, TZL*02, WGM05]. Gorla et al. [GIS03] synthesize a stationary surface texture using non-parametric sampling but adjust orientation based on curvature. We also take into account surface

metrics for the purpose of reproducing variation among different types of texture, but automatically deduce which geometric features are needed.

Spatially Varying Texture Synthesis has been demonstrated in 2D [Wei01, MZD05, ZFCG05] and on 3D surfaces [ZZV*03]. The specification of variation is left to the user, while we wish to explain the variation according to the characteristics of the underlying surface. Texture-by-numbers [Ash01, HJO*01, EF01] synthesizes texture according to a user-defined guidance field using non-parametric synthesis. We focus on creating and manipulating a guidance field specifically for geometry-correlated texture. We also introduce a novel synthesis approach based on a parametric model, which respects the correlation better.

User-Driven Decoration. Zhou et al. [ZWT*05] present a texture mapping approach with a focus on handling semantics. They carefully align the texture with the underlying geometry, but rely on a sparse set of user-defined correspondences to do so. We aim to automatically resolve a dense correspondence between texture and geometry. Zelinka et al. [ZG04] describe local surface characteristics using spatial neighborhoods that sample 3D offsets or curvature [GGGZ05]. They use this measure to guide geometry-dependent colorization based on sparse user input. Our technique yields automatic geometry-dependent texturing from an example. In addition, our guidance field is much more compact.

Weathering. The geometry-correlated textures we aim to reproduce are often caused by weathering processes like dirt accumulation and corrosion [DH96, DPH96, DEL*99, CXW*05]. Measures such as accessibility and exposure have been applied for mimicking a tarnished appearance [Mil94]. Wong et al. [WNH97] used accessibility, curvature and exposure to drive procedural texturing of weathered surfaces. Dorsey and Hanrahan [DH96] employed accessibility to guide generation of metallic patinas. In concurrent work, Lu et al. [LGR*05] and Giorghiades et al. [GLX*05] have sought to reproduce weathered appearance based on acquired examples. They propose suitable guidance fields for specific phenomena: drying [LGR*05], paint crackling and patina formation [GLX*05]. We do not make assumptions about the underlying phenomena or fabrication processes, but instead infer a suitable selection from an overcomplete set of geometric features for a given example. In addition, we can deal with the case of having vastly different source and target geometry using our feature matching technique.

2. Method Overview

Our method takes a (source) triangular mesh with a texture map as input which constitutes the training pair. Given a target mesh, we then generate a texture map exhibiting similar texture variation and correlation. The meshes are acquired using a commercial scanner, while texture is obtained from calibrated photographs (see Section 6.1).

We compile an overcomplete set of geometric features such as curvature and visibility, which are potentially relevant to texture variation and compute them for every location on the mesh.

We then seek to characterize which features are actually correlated with texture. For this, we use Canonical Correlation Analysis [Hot36] to obtain a subspace in the feature set that maximizes correlation with the texture color. The reduced features constitute the basic guidance map for constrained texture synthesis.

In some cases, the target mesh might be more detailed than the source mesh, such that the transferred texture looks too granular. Another related problem occurs when not enough data are available in the source model to map to the target, e.g., due to a partial scan. We introduce a feature matching step that transfers multi-scale characteristics from the source to the target features to tackle these issues.

The guidance field is then used to drive constrained texture synthesis over the surface. Using Hertzmann et al.’s terminology [HJO*01], the correlated features of the source and target are the “unfiltered” A and B images, while the source and target textures correspond to the “filtered” versions A' and B' , respectively. We present techniques based on non-parametric and parametric texture synthesis, with a tradeoff between the precise handling of structure and speed.

3. Geometric Correlation

Before introducing our set of geometric features and our dimensionality reduction approach, we describe the infrastructure we use to handle data on meshes.

3.1. Data Representation

We use well-established techniques to represent data over meshes. We opted for uniformly distributed set of surface points, as used by Turk [Tur01] and Wei et al. [WL01]. In our examples, 512000 points were used. Alternatively, a texture atlas [LPRM02] can be employed, but special care needs to be taken to deal with borders and potential parameterization distortions. We follow Turk [Tur01] to build our representation. The mesh is sampled uniformly at multiple resolutions using point repulsion [Tur91]. Each point carries a vector \mathbf{x} of geometric features — as discussed in the next section — and a texel \mathbf{y} (RGB triplet). We will use superscripts s and t to denote the source and target features (texels), respectively. The feature (texel) signal can be evaluated at any surface location using scattered data interpolation [She68].

A number of technical steps such as the feature matching introduced in Section 4 or the texture synthesis in Section 5 require a multi-scale representation of the data. We construct a Gaussian pyramid over the mesh by successive smoothing and down-sampling [Tur01].



Figure 2: Two geometric features: solid angle curvature and directional occlusion (based on spherical harmonics). Other features in our set are height and surface orientation (not shown here). Left to right: solid angle curvature with a small radius captures local variations; a big radius captures global characteristics such as exposure; ambient occlusion (constant spherical harmonics band) contributes as an exposure indicator; directional occlusion (linear band) can measure influence along the direction of gravity.

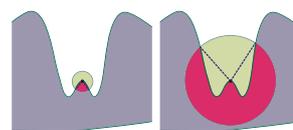


Figure 3: 2D analogy of the multi-scale solid angle curvature. The solid angle curvature approximates the fraction of a sphere enclosed by the surface as the solid angle subtended by the corresponding sphere-surface intersection curve. The sphere radius controls scale. In this example here, at the smaller scale, the surface is convex (left); at a large scale, it is concave (right).

3.2. Geometric Features

We compiled a set of geometric characteristics that are likely to be relevant for various geometry-dependent effects. Each of these feature vectors will be computed at each vertex of the source and target mesh, and interpolated at each texel. See Figure 2 for examples. We use the following features:

Normalized Height. We observed that texture and color variation often depend on height, that is, the distance to the supporting plane. Height-dependent variation can be attributed to weathering, for instance. We measure the location of each surface point along the vertical axis (normalized to $[0, 1]$ with respect to the object’s bounding box).

Surface Normal. The local orientation of the surface can be relevant for certain weathering-related effects (e.g., influence from the direction of the sky). This orientation is well represented by the surface normal.

Multi-Scale Solid Angle Curvature. We adapt the surface descriptor introduced by Connolly [Con86], which turns out to be useful for measuring curvature and exposure, e.g., to capture a tarnished appearance. It approximates the fraction of the sphere that lies inside the object. Given a sphere centered at the point of interest, the solid angle curvature is equal to the solid angle subtended by the intersection of the surface with this sphere (Fig. 3). We compute the intersection analytically using the spherical excess formula. The

radius acts as a scale parameter, akin to the size of a convolution kernel. We empirically chose four different radii which are scaled relatively to the diagonal of the object’s bounding box. Solid angle curvature is closely related to accessibility [Mil94], but has the advantage that it provides information for both concave *and* convex parts of the surface. Also, the solid angle curvature can be evaluated easily at multiple scales, unlike discrete mean curvature [MDSB02] and accessibility.

Directional Occlusion. Occlusion is a useful indicator of exposure [WNH97]. We introduce a novel measure which can be seen as a directional extension of ambient occlusion [ZIK98]. We capture the large-scale directional variation of visibility by projecting the binary visibility function at each location onto the spherical harmonic (SH) basis [SKS02] using ray casting. The resulting coefficients are used as features. We found that the constant and linear band are sufficient. Note that the constant SH band actually encodes ambient occlusion. Orientation is defined with respect to object space, in order to capture directional dependencies which could coincide with directions such as gravity.

The total dimensionality of our feature set is 12. We found these features to be sufficient for our purposes. One could easily extend this set with other measures, such as accessibility [Mil94], mean curvature [MDSB02, GGGZ05, GLX*05], and so on, but they are already closely related to the above features and thus provide little additional information. See Figure 4 for a demonstration of the usefulness of these features.

3.3. Correlation Analysis

Given the geometric features and a texture for a source model, we need to reduce the dimensionality of the former (12D) to make synthesis tractable. Our problem is different from the typical usage of a technique such as PCA, because we are interested in the correlation between two kinds of vectors (features and texture description), while PCA only characterizes the extent of a dataset in one space. Consider the simple case where color is a function of only curvature (Figure 5). PCA preserves an irrelevant feature like height because it exhibits significant variation in the dataset, although it is uncorrelated to texture color.

This is why we employ Canonical Correlation Analysis (CCA) [Hot36, MKB00, Bor98] to compute a parsimonious feature subset for a given example (Fig. 5). CCA finds an affine low-rank transformation such that the source feature vectors \mathbf{x}^s and the texel RGB vectors \mathbf{y}^s have maximal correlation. In contrast, standard dimensionality reduction using PCA transforms \mathbf{x}^s into a space that maximizes variance (see Figure 6 for a comparison between CCA and PCA). Under the assumption that \mathbf{x}^s and \mathbf{y}^s are zero-mean, CCA returns two matrices \mathbf{W}_x and \mathbf{W}_y and a diagonal matrix of respective correlations $\sqrt{\Lambda}$. These are related as follows:

$$\sqrt{\Lambda} \mathbf{W}_x^\top \mathbf{x}^s \approx \mathbf{W}_y^\top \mathbf{y}^s, \quad (1)$$

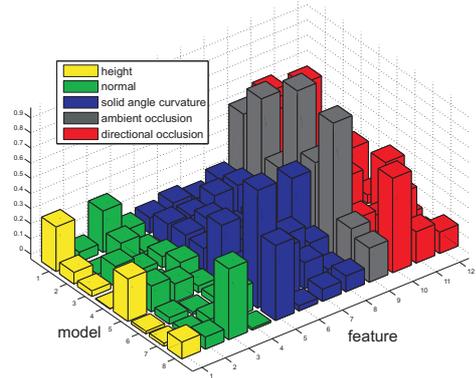


Figure 4: Importance of our feature set. We take 8 textured models, apply our correlation analysis using CCA and observe the principal direction (i.e., the subspace with the highest correlation value). The magnitude of the components of this direction vector are plotted for each model (each vector is normalized for clarity). In general, we see that every feature contributes significantly in at least a few of the examples. Solid angle curvature appears to have importance across various scales. Ambient occlusion has a large influence in many cases, which can be attributed to the presence of residual illumination artifacts in the input texture (these did not lead to visible artifacts however).

The matrices \mathbf{W}_x and \mathbf{W}_y transform the features and texture respectively into a correlated latent space. We are only interested in transforming the features, so matrix \mathbf{W}_y is not used. The correlated source feature vectors \mathbf{x}^{s*} are computed with $\mathbf{x}^{s*} = \mathbf{W}_x^\top \mathbf{x}^s$, where \mathbf{W}_x^\top retains only d components. The resulting feature dimensionality d is always equal to the lowest dimensionality among the two datasets [Bor98]. Thus for our case, d is always equal to three (cf. RGB), regardless of the size of the feature vector. The dimensionality reducing matrix-vector multiplication boils down to three weighted summations over the feature components. These weights in matrix \mathbf{W}_x^\top basically tell us which features are relevant (see Figure 4). Finally, the correlated target features are obtained by re-applying the same transform to the target features: $\mathbf{x}^{t*} = \mathbf{W}_x^\top \mathbf{x}^t$, where \mathbf{x}^t are the target features. The result of our analysis is the set of correlated features \mathbf{x}^{s*} and \mathbf{x}^{t*} , which form the source and target guidance fields respectively. See Figure 8 for a plot of the guidance fields.

Other techniques exist that find a relation between two datasets, in particular the ones that maximize covariance instead of correlation (e.g. Partial Least Squares [Bor98]). Covariance however is heavily dependent on the scale of the two spaces \mathbf{x}^s and \mathbf{y}^s , an issue that is ill-defined in our case. In contrast, CCA is insensitive to the respective scales of the two spaces and therefore is more robust in our context.

Currently we are only correlating geometry with color (RGB triplets). More elaborate texture descriptors could be applied, however, the approximation with RGB triplets al-

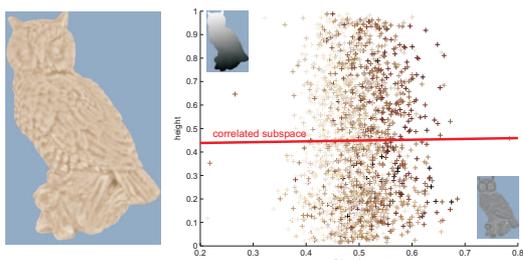


Figure 5: Illustration of CCA. We compute 2 features on the owl: solid angle curvature (horizontal axis) and height (vertical axis). The scatter plot displays the points in feature space, along with their color (enhanced contrast for clarity). We apply CCA between features and texel brightness, and the resulting linear subspace (red axis) captures the strong correlation between brightness and solid angle curvature. In contrast, PCA would find the opposite direction (vertical) because it does not take correlation with brightness into account and only considers the variation of the features.

ready performs sufficiently well. More advanced texture descriptors are likely to add redundant complexity to our approach. We experienced that as long as the average texture colors for the different parts (e.g. flat, exposed, concave, etc.) of the object are discernible, our technique is able to establish a sufficient amount of correlation, even for very intricate and structured textures (e.g. as seen in Figure 1).

4. Feature Matching

We now address the case where the geometry of source and target model differ significantly, leading to vastly different feature distributions in the source and target. This raises an important issue: we will need to synthesize texture for features that were not observed in the source, making the transfer problem ill-posed. We address this problem by forcing the statistics of the target features to match that of the source. In addition, the overall texture distribution will be different if the geometric feature distribution is different. Since our goal is visual faithfulness rather than physical accuracy, we choose to match the features of the new model to comply

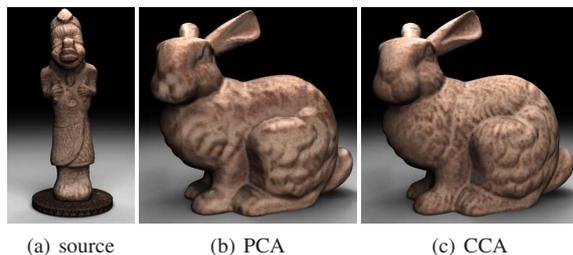


Figure 6: Comparing transfer results for PCA and CCA. The texture generated using PCA is not correlated with the geometry, whereas the texture using CCA emphasizes the fine detail on the bunny.

with the source. For brevity, we will refer to the correlated features obtained after CCA simply as “features”.

4.1. Marginals

Matching the features is similar to matching color distributions [RAGS01, HB95]. This is usually done using 1D histogram matching [GW02] for each axis, often in a decorrelated space [RAGS01, HB95]. In order to better handle complex distribution, we chose the n -dimensional histogram matching technique recently introduced by Pitie et al. [PKD05]. Their technique operates by repeatedly rotating the data, and performing standard histogram transfers along the axes, and projecting the data back to the original space.

Altering the features through matching corresponds to changing the geometric description of the target model. However, we still remain as faithful to the original geometry as we can, since histogram matching monotonically modifies existing features and will not introduce spurious novel elements such as new bumps or creases.

4.2. Multiscale Feature Content

We have found that the feature distribution must be matched at different scales. For instance, when the source model is smooth and the target contains mainly small details, it is useful to reduce the features at smaller scales, while coarser features should be enhanced. We therefore decompose the features into a Laplacian pyramid, and match the pyramid coefficients in addition to the actual feature values, in the same spirit as Heeger and Bergen [HB95]. The Laplacian pyramid is computed from the Gaussian pyramid of features, by subtracting each up-sampled coarser level from the current level, while leaving the coarsest one unchanged [AB81]. Figure 7 shows the influence of feature matching on a transfer between very different geometries.

In Figure 8, we show a transfer from a scanned bagel to the Bunny using parametric synthesis. It is a particularly challenging case because the source data is incomplete, in the sense that certain feature values are not present. In particular, directional occlusion is not sampled in the downward direction because only the top half of the bagel was acquired. Using our multi-scale feature matching approach, we are able to fill in the missing data, and generate a convincing and consistent transfer.

5. Texture Transfer

Given the texture on the source model and the guidance field on the target mesh, we use constrained synthesis to generate a texture with appropriate variation. We adapt two standard constrained synthesis algorithms that trade off between the ability to handle structured textures, speed, and robustness to parameters.

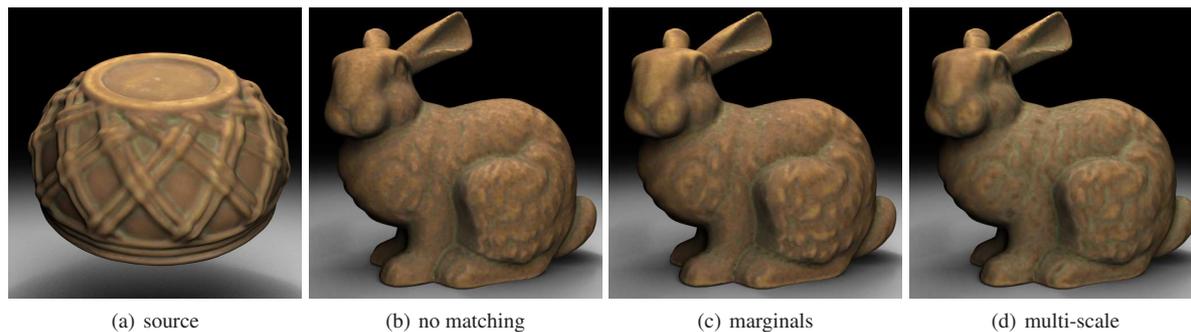


Figure 7: Feature matching enhances the guidance field of the target for more consistent transfers when source and target are different. The bowl only has thick round features, in contradistinction to the bunny’s sharp bumps and creases. Image (b) shows the result without feature matching. We observe that the green texture almost did not carry over to the bunny. Also, the bunny texture seems too have a too finely grained appearance. Matching the marginals (c) improves the result slightly. Only if we match the features at multiple scales (d), the transfer becomes visually consistent. Note how the head of the bunny has a smoother appearance, as seen on the exposed parts of the bowl.

5.1. Non-parametric Synthesis

Our guidance field can be readily applied to the previously introduced non-parametric constrained synthesis techniques [HJO*01, EF01]. We implemented the Image Analogies framework [HJO*01] on top of Turk’s non-parametric texture synthesis method for surfaces [Tur01]. For each unsynthesized texel on the target, a local surface neighborhood containing the already synthesized part and the underlying correlated features is compared against all neighborhoods on the source model. The texel with the closest match according to the L_2 norm, is copied onto the target. Synthesis proceeds coarse-to-fine in a Gaussian pyramid [WL00]. As described by Hertzmann et al., we improve texture consistency by combining the best match strategy [EL99, WL00] with Ashikhmin’s coherence heuristic [Ash01]. Several parameters control the synthesis: the relative weights of texels and features, the coherence bias (to decide whether to pick a coherent texel or a best match-texel) and the neighborhood size.

5.2. Parametric Synthesis

Although non-parametric synthesis is capable of reproducing compelling textures, it is challenging to use for all cases, in particular when the texture shows intricate variations due to strong correlation with geometry. This introduces spatial patterns on the target which were not observed on the example object, thereby making it challenging to satisfy the constraints in the neighborhood search from the limited number of possible matches. Reducing the constraints by downsizing neighborhoods improves correlation, but at the risk of destroying texture consistency. In addition, non-parametric synthesis has many parameters to tweak, which are strongly dependent on the input.

We propose a weaker texture model when texture is strongly correlated and unstructured. We extend the Heeger

and Bergen model (H&B) [HB95] to follow the guidance field. The basic H&B algorithm decomposes texture into a band-pass multi-resolution representation like the steerable [SFAH92] or Laplacian [AB81] pyramid. Texture is described by the global intensity histogram and the histograms at each pyramid level. Synthesis is performed by simultaneously transferring pyramid and intensity histograms of the source to target image.

5.2.1. Constrained Heeger and Bergen

Heeger and Bergen’s algorithm relies on global histograms. In order to extend it to spatially-varying texture, we need to handle such information locally. While we could compute a different histogram for each pixel, including only neighbors with similar features, the cost would be prohibitive. This is why we introduce a segmentation of the object through k -means clustering [Llo82] on the values in the guidance map. Similar values in the guidance map will be partitioned into the same segment. Because the feature values are correlated, each segment will contain the same type of texture, thereby providing a neighborhood over which texture histograms can be measured locally. For each of these segments, we compute a set of histograms for the (decorrelated [HB95]) intensity and Laplacian pyramid levels. Each segment is identified by its average correlated feature F^s , and its local texture is described by the histogram set H^s . This collection of pairs $F^s \rightarrow H^s$ defines a mapping from features to texture. See Figure 9(a) for an example of this segmentation.

The same clustering algorithm is performed on the target guidance field. For each target segment, we compute the average feature F^t . We use the source mapping $F^s \rightarrow H^s$ to infer the histogram set H^t for each F^t using interpolation. The “nearest” source clusters with respect to F^t are looked up using the L_2 norm. We interpolate the histogram set H^t from the k -nearest H^s ’s using the reciprocal square distance [She68]. We employ the inverse cumulative distri-

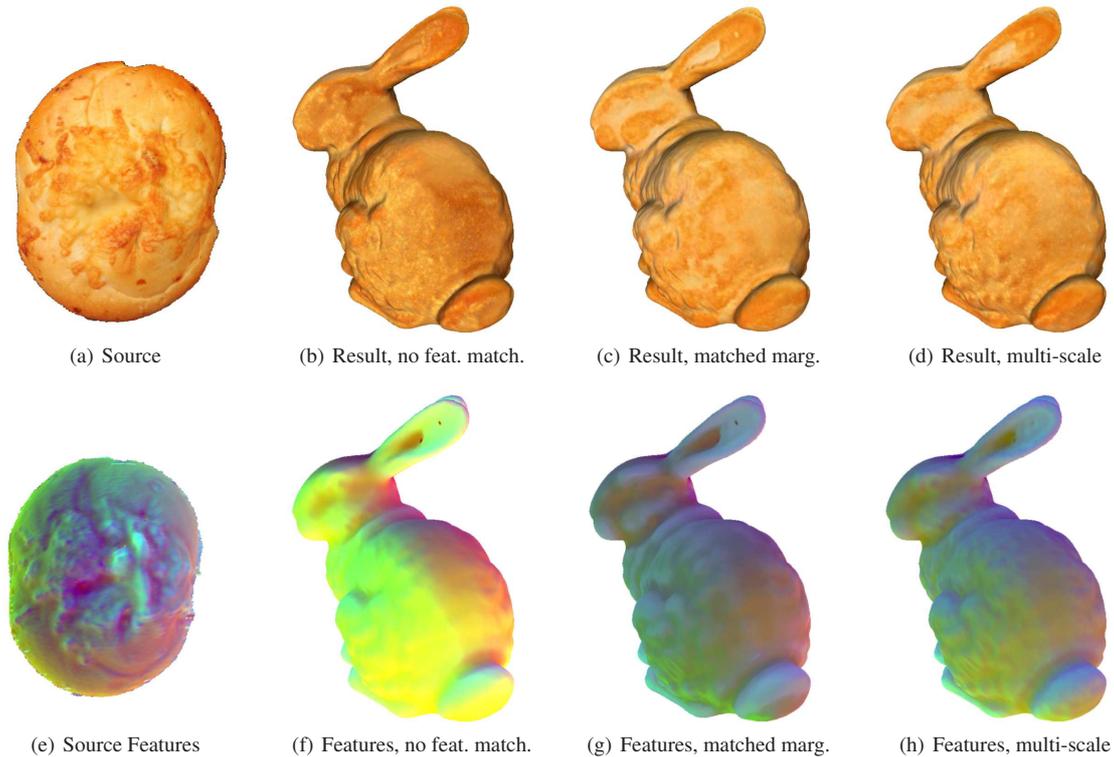


Figure 8: Application of feature matching. The top row shows texture; the bottom row shows the corresponding guidance fields, where the feature values have been shifted and scaled to $[0, 1]$ and displayed as RGB values. We scanned the top half of a bagel (a), and transferred its texture to the bunny model using parametric synthesis. Since the source model is incomplete, the transfer is ill-posed. Not all feature values in the target are available in the source: directional occlusion is only sampled for the upper hemisphere because the top half of the bagel is available. Consequently, the guidance field takes on spurious values (f), yielding a transferred texture that is not consistent with the source (b): the texture is too rough and there are artifacts on the head and back of the bunny. Matching marginal feature statistics partially solves the problem, but the transfer still has some artifacts (c), which are due to discontinuities in the guidance field (g). The multi-scale feature matching solves this, yielding a smooth and visually consistent transfer (d). Note that the lighting and the orientation of the bunny was selected to highlight differences between textures. The supplementary material shows renderings under more natural illumination.

bution functions instead of the actual histograms in order to consistently blend the distributions [MZD05]. Then, a new texture is synthesized for each segment based on the derived histogram set H^i [HB95]; see Figure 9(d) for a result.

As an alternative to interpolation, we could establish explicit correspondences between source and target clusters. However, interpolation has the advantage of being able to generate novel mixtures of texture.

5.2.2. Feathering

Since k -means provides a hard segmentation, we feather the result to avoid seams using a Gaussian with a standard deviation equal to that of the corresponding cluster’s distribution. At segment borders, the weighting functions are forced to sum to one on a per-texel basis, thereby assuring consistent blending. The feathering is used on both the source and target. Synthesis proceeds by applying the histogram transfers

to each segment, and re-combining all segments using a per-texel weighted summation. Figure 9(b) shows the effect of feathering.

5.2.3. Initialization

Although clustering already enforces much of the structure, additional measures should be taken to obtain a crisp result. The Heeger and Bergen model is too weak to synthesize structure such as crevices. As a result, naively initializing the target with random noise generates an unstructured and uncorrelated result. We therefore create a “good guess” of what the texture should look like. In particular, the initialization should contain most of the structure, which will be refined further by H&B. This is similar to the sharpness preservation for texture morphing, as proposed by Matusik et al. [MZD05]. In our case, H&B generates texture on top of the initialization instead of attenuating high frequencies.

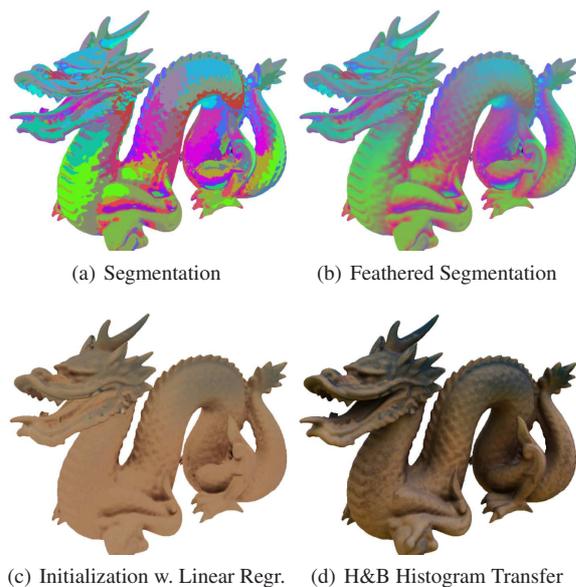


Figure 9: Steps involved in computing a transfer to the dragon model using constrained parametric synthesis. Segmentation on the feature values is used to define neighborhoods in which texture can be measured locally. In figure (a), we show such a segmentation where each cluster is displayed by unique color. We turn the hard segmentation into a soft one using feathering. In figure (b), cluster colors are displayed using the feathered weights. Before applying the histogram transfer, we initialize the result using simple linear regression per segment (c), yielding an good estimate geometry-correlated structure. Finally, Heeger and Bergen-style histogram transfer [HB95] adds textural detail (d).

We apply a simple linear regression model by fitting the point-wise mapping between correlated features \mathbf{X}^s and texels \mathbf{Y}^s for each source cluster. Linear regression finds a matrix \mathbf{A}^s and offset vector \mathbf{b}^s , such that $\mathbf{Y}^s = \mathbf{A}^s \mathbf{X}^s + \mathbf{b}^s$. The linear regression coefficients for the target — \mathbf{A}^t and \mathbf{b}^t — can also be inferred using the the k-nearest neighbor interpolation technique described earlier. The target is initialized by applying $\mathbf{Y}^t = \mathbf{A}^t \mathbf{X}^t + \mathbf{b}^t$. Finally, a fixed amount of white noise is added to the resulting initialization which acts as “seed” for generating texture details. See Figure 9(c) for an example of initialization.

5.3. Discussion

Non-parametric synthesis achieves good results when the guidance field stays fairly smooth. One must be careful however in specifying the correct parameters (neighborhood size, relative weighting of features versus texture, and the Ashikhmin coherence parameter), which may require tedious trial-and-error, especially given that it takes over 2 hours per transfer (for 5×5 neighborhoods).

A particular difficult case for non-parametric sampling

is when texture is strongly correlated with the underlying geometry (feature values). Despite efforts to find the right parameters, non-parametric sampling will fail to reproduce texture in a visually consistent manner here. The reason is that consistency in texture and features must be traded off against each other, which is not always possible. In order to achieve a decent degree of texture consistency, one must employ a sufficiently large neighborhood size (5×5 in our experiments). Unfortunately, as neighborhood size grows, the search space will exponentially increase in volume, while the number of available example neighborhoods on the source remains constant. It will therefore be harder to find a match because not enough examples are available. In particular, when a certain pattern in the target guidance field does not occur on the source, the feature neighborhood cannot be matched, which makes it hard to conform with the target guidance field. Using very small neighborhoods (3×3) or increasing the guidance weights improves consistency with the features, but produces poor textures. Typically, the synthesis “gets stuck” in certain parts of the example texture (as reported previously [EL99]). Several researchers have shown that texture quality can be improved by enforcing coherency, i.e. by copying over patches instead of pixels (this can be controlled by the Ashikhmin coherence parameter). Unfortunately, this results in similar problems since copying large patches of texture is now encouraged, which again assumes that the source and target guidance fields are very similar to each other.

If the texture does not contain much structure, aside from structure defined by the guidance field, the parametric texture model will create good transfers. It is fast — transfers take about 15 minutes — and virtually parameter-free; only the number of clusters can be changed, which we set to 50 by default. The algorithm follows the guidance map very closely since it operates at a very high granularity, resulting in textures that are highly correlated with the underlying geometry. Furthermore, it generalizes well to unseen patterns in the guidance field. On the downside, structured textures cannot be handled with this texture model. The use of a Laplacian pyramid limits the textures to be isotropic. Using a steerable pyramid instead would allow anisotropy but the associated image filters are oriented, requiring a local frame at each point that is consistent with the input texture. However, the majority of the textures we observed are isotropic, and for the remaining cases, we use non-parametric synthesis. A detailed visual comparison between the two algorithms can be found in Figure 12 and in the supplementary material.

6. Results

6.1. Data Acquisition

Our input data consist of scanned texture-mapped objects. Most of the models in this paper were downloaded from Esteban and Schmitt’s collection [ES03].

In addition, we acquired several models ourselves using

a 3D laser scanner. Texture maps were constructed from one or more images taken by a CCD camera. We reduced illumination artifacts by using only indirect lighting during acquisition, and high pass filtering in log-space as post-processing [OCDD01]. The images were registered using Lensch et al.'s silhouette matching [LHS00]. Finally, we re-sampled and combined all images in a texture atlas using "Graphite" [Gra03].

The user is provided with a simple tool to mask out parts of the texture. Although this is optional, it sometimes helps in removing unwanted information which can possibly "contaminate" the output, like markings on statues and incomplete parts of the texture map. The masked texels are simply discarded from the training set.

6.2. Discussion of Results

For all parametric synthesis results in the paper, we used three Laplacian pyramid levels, three H&B iterations, and 50 clusters for segmentation. We used the following parameters for non-parametric synthesis in Figures 1 (right) and 11: neighborhood size of 5×5 and a coherence parameter of 1.0. We used the ANN library [AMN*98] to retrieve the nearest neighborhoods. Features and texture were given equal weights, and the number of levels in the Gaussian pyramid was 3. We ran our implementation on a Pentium 4 (2.8Mhz) with 1Gb of main memory. The time required to create the guidance field creation (CCA) is about 15 seconds. It takes roughly 15 minutes to synthesize a texture using the parametric version, and about 2 to 3 hours using non-parametric synthesis (for 5×5 neighborhoods). Multi-scale feature matching requires an extra 10 minutes.

Figure 1 shows a transfer of a strongly correlated texture and a slowly varying structured texture, which were generated with parametric (left) and non-parametric synthesis (right). Figure 11 shows another transfer of structured texture using the non-parametric version.

Figure 10 shows a transfer from a computer-generated texture. We applied Dorsey et al.'s simulation [DH96] to generate a patina texture on the bunny model. The patina is transferred to the dragon. The transfer resembles the "ground truth" patina, which was created using the same parameters as the source.

Figure 12 compares parametric to non-parametric synthesis for the case of strongly correlated texture. We tried non-parametric synthesis with different settings, but were unable to find a good set of parameters for which the transfer conforms with the input. The parametric version does not require parameter tweaking, and is able to reproduce the appearance faithfully.

7. Conclusions and Future Work

We have presented a method for the transfer of geometry-correlated texture variation from a texture-mapped source

mesh to a new mesh. We compile an overcomplete set of geometric features and reduce it to only the relevant ones by analyzing correlation. The reduced feature set is compact in order for it to be used as guidance field for constrained texture synthesis. Matching the statistics of the target guidance field to the source at multiple scales makes improves transfer quality when source and target geometry differ significantly.

We have used two constrained texture synthesis approaches with different characteristics. Our parametric model trades the ability of handling structure texture for stronger correlation (vice versa for the non-parametric model). Consequently, structured textures that exhibit a high degree of correlation may pose a problem.

While our method can reproduce texture variation due to weathering processes, it does not capture all weathering effects. In particular effects that do not sufficiently correlate to geometry are not handled, such as paint peeling [GLX*05] and water flows [DPH96]. The extension of our feature set to non-geometric parameters is an interesting avenue of future work.

The feature matching enhances existing features in the guidance field, but cannot create novel elements. For instance, when transferring from a detailed model to a coarser one, small-scale details might be missing which are needed to match the target to the source's frequency content.

We are planning to transfer the variation of material appearance as encoded by BRDFs and BTFs. We want to extend our work to the animation of weathering processes, both using manual specification and acquired time-lapse data — similar to the work by Georghiadis et al. [GLX*05].

Acknowledgements

We would like to thank the following people for their help: Jon Chu, Eugene Hsu, Hendrik Lensch, Frank Van Reeth and Tiffany Wang. Bruno Lévy was so kind to provide his mesh parameterization tool [Gra03]. Tom Mertens received a research fellowship from the Belgian American Educational Foundation. Part of this work was realized while Tom Mertens was a PhD student at the Expertise Centre for Digital Media (Hasselt University). Jan Kautz was supported in part by an Emmy-Noether fellowship from the German Research Foundation for his stay at MIT. Philippe Bekaert received financial support from the European Regional Development Fund and the Flemish Interdisciplinary Institute for Broadband Communication. This work was supported by a National Science Foundation CAREER award 0447561 "Transient Signal Processing for Realistic Imagery.", an NSF Grant No. 0429739 "Parametric Analysis and Transfer of Pictorial Style.", and a Microsoft Research New Faculty Fellowship held by Frédo Durand.

References

- [AB81] ADELSON E. H., BURT P. J.: Image data compression with the laplacian pyramid. In *Proceeding of the Conference on Pattern Recognition and Image Processing* (Los Angeles, CA, 1981), IEEE Computer Society Press, pp. 218–223.

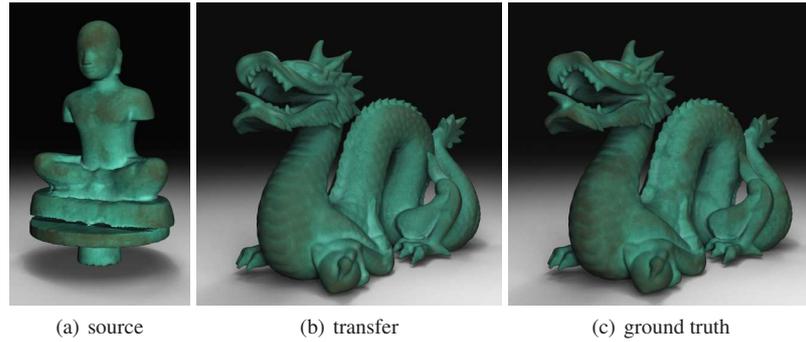


Figure 10: Transfer of a computer-generated texture. We computed a patina simulation on a buddha model (a) using Dorsey and Hanrahan’s method [DH96], and transferred it to the dragon (b) using parametric synthesis. The transfer resembles the ground truth texture (c) which was obtained by running the same simulation on the dragon model.



Figure 11: Transfer of a structured texture using non-parametric synthesis. Parameters: 5×5 neighborhood, coherence = 1 and feature weight = .8 (texture weight = .2).

[AMN*98] ARYA S., MOUNT D. M., NETANYAHU N. S., SILVERMAN R., WU A.: An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM* 45, 6 (1998), 891–923. Source code available from <http://www.cs.umd.edu/mount/ANN>.

[Ash01] ASHIKHMINE M.: Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics* (2001), ACM Press, pp. 217–226.

[Bor98] BORGA M.: *Learning Multidimensional Signal Processing*. PhD thesis, Linköping University, 1998.

[Con86] CONNOLLY M.: Measurement of protein surface shape

by solid angles. *J. Mol. Graph.* 4, 1 (1986), 3–6.

[CXW*05] CHEN Y., XIA L., WONG T.-T., TONG X., BAO H., GUO B., SHUM H.-Y.: Visual simulation of weathering by gamma-ton tracing. *ACM Trans. Graph.* 24, 3 (2005), 1127–1133.

[DEL*99] DORSEY J., EDELMAN A., LEGAKIS J., JENSEN H., PEDERSEN H. K.: Modeling and Rendering of Weathered Stone. In *Proceedings of SIGGRAPH 99* (Aug. 1999), Computer Graphics Proceedings, Annual Conference Series, pp. 225–234.

[DH96] DORSEY J., HANRAHAN P.: Modeling and Rendering of Metallic Patinas. In *Proceedings of SIGGRAPH 96* (Aug.

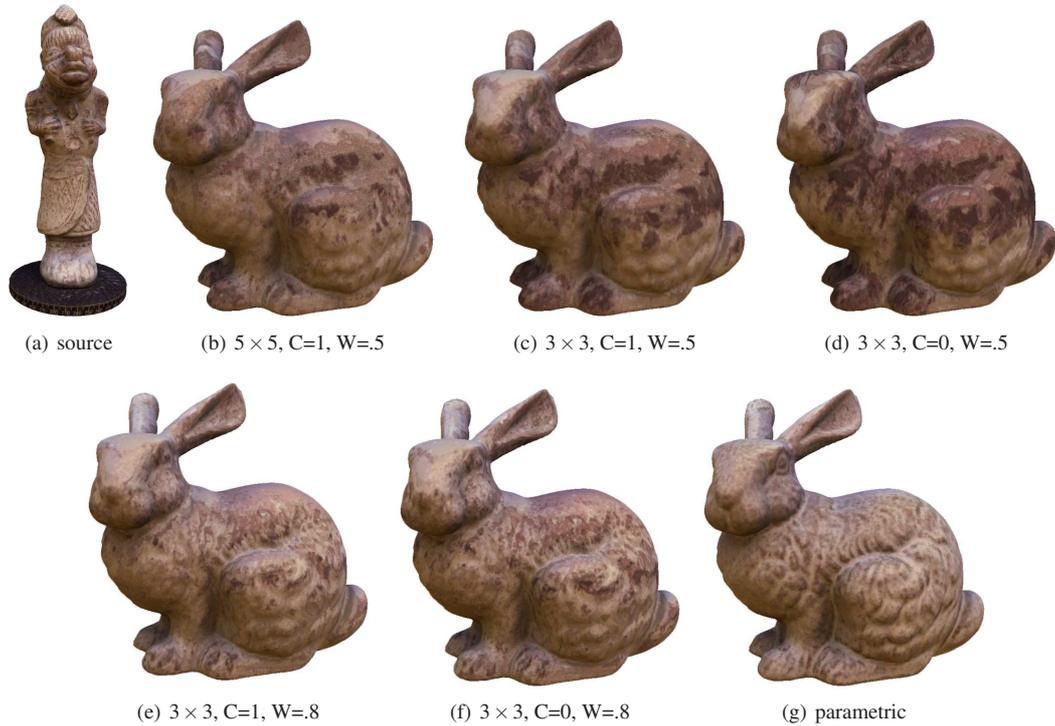


Figure 12: Tweaking parameters for the non-parametric transfer. We synthesized transfers for different neighborhood sizes (5×5 and 3×3), coherence values C , and feature weights W . The weight for texture equals $1-W$. Better results are obtained for higher feature weights. However, the texture still looks very random and uncorrelated with the underlying geometry (unlike the source model), while the parametric version is able to reproduce the fine correlations better.

- 1996), Computer Graphics Proceedings, Annual Conference Series, pp. 387–396.
- [DPH96] DORSEY J., PEDERSEN H. K., HANRAHAN P.: Flow and Changes in Appearance. In *Proceedings of SIGGRAPH 96* (Aug. 1996), Computer Graphics Proceedings, Annual Conference Series, pp. 411–420.
- [EF01] EFROS A., FREEMAN W.: Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, pp. 341–346.
- [EL99] EFROS A. A., LEUNG T. K.: Texture Synthesis by Non-parametric Sampling. In *IEEE International Conference on Computer Vision* (September 1999), pp. 1033–1038.
- [ES03] ESTEBAN C. H., SCHMITT F.: Image-based 3d models archive, 2003. <http://www.tsi.enst.fr/3dmodels>.
- [GGGZ05] GATZKE T., GRIMM C., GARLAND M., ZELINKA S.: Curvature maps for local shape comparison. In *Proceedings of Shape Modeling International 2005* (2005).
- [GIS03] GORLA G., INTERRANTE V., SAPIRO G.: Texture synthesis for 3d shape representation. *IEEE Transactions on Visualization and Computer Graphics* 9, 4 (2003), 512–524.
- [GLX*05] GEORGHIADES A. S., LU J., XU C., DORSEY J., RUSHMEIER H.: *Observing and Transferring Material Histories*. Tech. Rep. YALEU/DCS/TR-1329, Yale University, June 2005.
- [Gra03] GRAPHITE., 2003. <http://www.loria.fr/levy/Graphite/index.html>.
- [GW02] GONZALEZ R. C., WOODS R. E.: *Digital Image Processing*, second ed. Prentice Hall, 2002.
- [HB95] HEEGER D., BERGEN J.: Pyramid-based texture analysis/synthesis. In *Proceedings of SIGGRAPH 95* (Aug. 1995), Computer Graphics Proceedings, Annual Conference Series, pp. 229–238.
- [HJO*01] HERTZMANN A., JACOBS C., OLIVER N., CURLESS B., SALESIN D.: Image Analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, pp. 327–340.
- [Hot36] HOTELLING H.: Relations between two sets of variates. *Biometrika* 28 (1936), 321–377.
- [LGR*05] LU J., GEORGHIADES A., RUSHMEIER H., DORSEY J., XU C.: Synthesis of material drying history: Phenomenon modeling, transferring and rendering. In *Proceedings of Eurographics Workshop on Natural Phenomena* (2005).
- [LHS00] LENSCH H., HEIDRICH W., SEIDEL H.-P.: Automated texture registration and stitching for real world models. In *Proceedings of the 8th Pacific Conference on Computer Graphics and Applications* (2000), IEEE Computer Society, p. 317.
- [LLH04] LIU Y., LIN W.-C., HAYS J. H.: Near regular texture analysis and manipulation. *ACM Transactions on Graphics (SIGGRAPH 2004)* 23, 3 (August 2004), 368–376.

- [Llo82] LLOYD S.: Least squares quantization in PCM. *IEEE Trans. on Information Theory* 28, 2 (1982), 127–138.
- [LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), ACM Press, pp. 362–371.
- [MDSB02] MEYER M., DESBRUN M., SCHRÖDER P., BARR A.: Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Proceedings of VisMath'02* (2002).
- [Mil94] MILLER G.: Efficient algorithms for local and global accessibility shading. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (1994), ACM Press, pp. 319–326.
- [MKB00] MARDIA K., KENT J., BIBBY J.: *Multivariate Analysis*. Academic Press, 2000.
- [MZD05] MATUSIK W., ZWICKER M., DURAND F.: Texture design using a simplicial complex of morphable textures. *ACM Transactions on Graphics* 24, 3 (Aug. 2005), 787–794.
- [OCDD01] OH M., CHEN M., DORSEY J., DURAND F.: Image-based modeling and photo editing. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, pp. 433–442.
- [PFH00] PRAUN E., FINKELSTEIN A., HOPPE H.: Lapped textures. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 465–470.
- [PKD05] PITIE F., KOKARAM A. C., DAHYOT R.: N-dimensional probability density function transfer and its application to colour transfer. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 2* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 1434–1439.
- [RAGS01] REINHARD E., ASHIKHMIN M., GOOCH B., SHIRLEY P.: Color transfer between images. *IEEE Computer Graphics and Applications* 21, 5 (2001), 34–41.
- [SCA02] SOLER C., CANI M.-P., ANGELIDIS A.: Hierarchical pattern mapping. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), ACM Press, pp. 673–680.
- [SFAH92] SIMONCELLI E., FREEMAN W., ADELSON E., HEEGER D.: Shiftable multi-scale transforms. *IEEE transactions on information theory* 38, 2 (1992), 587–607.
- [She68] SHEPARD D.: A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference* (1968), ACM Press, pp. 517–524.
- [SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), ACM Press, pp. 527–536.
- [Tur91] TURK G.: Generating textures on arbitrary surfaces using reaction-diffusion. *Computer Graphics* 25, 4 (July 1991), 289–298. Proceedings of SIGGRAPH 91.
- [Tur01] TURK G.: Texture synthesis on surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, 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. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), ACM Press, pp. 665–672.
- [Wei01] WEI L.-Y.: *Texture Synthesis by Fixed Neighborhood Searching*. PhD thesis, Stanford University, 2001.
- [WGM05] WANG L., GU X., MÜLLER K., YAU S.-T.: Uniform texture synthesis and texture mapping using global parameterization. In *Proceedings of Pacific Graphics* (2005).
- [WL00] WEI L.-Y., LEVOY M.: Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 479–488.
- [WL01] WEI L.-Y., LEVOY M.: Texture synthesis over arbitrary manifold surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM Press, pp. 355–360.
- [WNH97] WONG T.-T., NG W.-Y., HENG P.-A.: A Geometry Dependent Texture Generation Framework for Simulating Surface Imperfections. In *Eurographics Rendering Workshop 1997* (June 1997), pp. 139–150.
- [YHBZ01] YING L., HERTZMANN A., BIEMANN H., ZORIN D.: Texture and shape synthesis on surfaces. In *Proceedings of 12th Eurographics Workshop on Rendering* (2001).
- [ZFCG05] ZALESNY A., FERRARI V., CAENEN G., GOOL L. V.: Composite texture synthesis. *International Journal of Computer Vision* 62, 1/2 (2005), 161–176.
- [ZG04] ZELINKA S., GARLAND M.: Similarity-based surface modelling using geodesic fans. In *Proceedings of the Second Eurographics Symposium on Geometry Processing* (2004), Eurographics Association, pp. 209–218.
- [ZIK98] ZHUKOV S., IONES A., KRONIN G.: An ambient light illumination model. In *Rendering Techniques '98 (Proceedings of the Eurographics Workshop on Rendering)* (1998), pp. 45–55.
- [ZWT*05] ZHOU K., WANG X., TONG Y., DESBRUN M., GUO B., SHUM H.-Y.: TextureMontage: Seamless texturing of arbitrary surfaces from multiple images. In *ACM SIGGRAPH 2005* (2005), pp. 1148–1155.
- [ZZV*03] ZHANG J., ZHOU K., VELHO L., GUO B., SHUM H.-Y.: Synthesis of progressively variant textures on arbitrary surfaces. *ACM Transactions on Graphics* 22, 3 (July 2003), 295–302.