COMPUTER GRAPHICS TECHNICAL REPORTS

CG-2005/01

Fragment-based Surface Inpainting

Gerhard H. Bendels
Universität Bonn. bendels@cs.uni-bonn.de

Ruwen Schnabel
Universität Bonn. schnabel@cs.uni-bonn.de

Reinhard Klein
Universität Bonn. rk@cs.uni-bonn.de

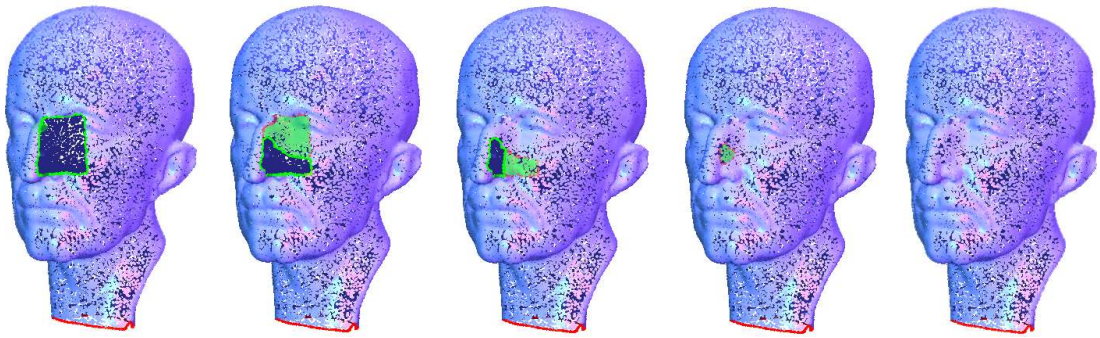
Institut für Informatik II
Universität Bonn
D-53117 Bonn, Germany

ISSN 1610-8892

Fragment-based Surface Inpainting

G.H. Bendels, R. Schnabel, R. Klein

Institut für Computergraphik, Universität Bonn, Germany



Max Planck model with missing data. The hole is inpainted by copying appropriate fragments of other parts of the object to the hole region.

Abstract

Inpainting is a well-known technique in the context of image and art restoration, where paint losses are filled up to the level of the surrounding paint and then coloured to match. Analogue tasks can be found in 3D geometry processing, as digital representations of real-world objects often contain holes, due to hindrances during data acquisition or as a consequence of interactive modelling operations. In this paper we present a novel approach to automatically fill-in holes in structured surfaces where smooth hole filling is not sufficient. Previous approaches inspired by texture synthesis algorithms require specific spatial structures to identify holes and possible candidate fragments to be copied to defective regions. Consequently, the results depend heavily on the choice and location of these auxiliary structures, such that for instance symmetries are not reconstructed faithfully. In contrast, our approach is based on local neighbourhoods and therefore insensitive with respect to similarity transformations. We use so-called guidance surfaces to guide and prioritise the atomic filling operations, such that even non-trivial and larger holes can be filled consistently. The guidance surfaces are automatically computed and iteratively updated during the filling process, but can also incorporate any additional information about the surface, if available. While custom-tailored for point-sets, our approach is suitable as-is for polygonal representations as well.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Comp. Geometry and Object Modeling]:

1. Introduction

Creating digital 3D copies of real-life objects is becoming a standard procedure for various application fields - ranging from cultural heritage applications and medicine to automotive, artistic, and entertainment industries, just as traditional photography has been for the past decades. Despite all technological progress, models resulting even from the most careful acquisition process are generally incomplete, due to occlusion, the object's material properties or spatial constraints during recording (among others), i.e. they contain undersampled regions and/or holes. On the other hand, holes are also deliberately introduced into an object, as removing damaged, undesired or unnecessary parts of an object is an important tool in interactive modelling.

In order to derive complete and visually appealing models, these holes have to be filled *appropriately*, i.e. the basic geometry has to be smoothly patched and the (unknown, yet assumed) detail geometry has to be restored or extrapolated, taking into account the context of the object. That this ill-posed task has hope of being solved at all is based

on the observation, that real-life objects often exhibit a high degree of coherence in the sense that for missing parts one can find similar regions on the object. This observation has been exploited extensively in the field of 2D texture synthesis and disocclusion, and also in 3D surface completion. The problem with previous approaches, though, is that they require specific spatial structures to identify holes and possible candidate fragments to be copied to defective regions. Consequently, the results depend heavily on the choice and location of these auxiliary structures, such that for example symmetries are not reconstructed faithfully. In contrast, we propose in this paper a surface inpainting method that analyses the neighbourhood of a hole, finds best matching local neighbourhood patches represented in local frames (the 3D analogue to what is called a *fragment* in image processing), and fills the hole with copies of these. By finding best matches hierarchically on several scales, the hole is filled in conformance with the context with respect to all considered scales.

Key contributions of this paper are

- A robust hole detection scheme, fully automatic, yet customisable to the application's demands.
- A novel automatic hole filling algorithm based on local neighbourhoods and thereby independent of any specific spatial subdivision
- A multi-level local neighbourhood descriptor to reflect different frequency bands

In the following section, we'll shortly review the relevant literature, covering the inspiring works on 2D image processing and texture synthesis, but also previous approaches to automatic hole filling for boundary representations in 3D. Then we describe our algorithm in a one-level, non-hierarchical way in section 4, and extend the algorithm in section 5 to exploit Guidance Surfaces in an multi-scale approach. The robust detection of holes in the surface – a task that is non-trivial if dealing with point-set-surfaces – is subject of section 6.

2. Previous Work

The work most relevant for our paper can be subdivided into two basic categories, 2D image completion and 3D surface completion.

2.1. Image Completion

In image processing, synthesising images or parts thereof, comes in two flavours: In texture synthesis, from a sample image a new (generally larger) image is to be created that appears realistic and preserves visual properties of the sample image. This problem has been approached by explicitly modelling the distribution of the textural features which humans perceive as a specific type of texture [Per85, Tur01], by histogram matching [HB95, Bon97, PS00], or Markov random fields [ZWM98]. Despite the appealing mathematical formulation, these parametric approaches have been outperformed by non-parametric models that synthesize textures exemplar-based by transferring pixels [EL99] or patches [WL00, NA03] with appropriate neighbourhoods to the new image.

Image completion, on the other hand, aims at filling-in holes in an image that are generated erasing defective, damaged or undesired parts of an image, by extending information available in the remaining image. Here, in addition to the overall visual properties of the image, the larger and highly irregular structures of the image have to be preserved. With this requirement in mind, Ballester et al. [BBC*01] fill images by explicitly propagating lines of equal brightness (so-called isophotes) by solving variational problems, whereas Jia et al. [JT03] segment the image and propagate segment borders into the hole region, before filling colors in a pixel-based approach. Using isophote-propagation to guide what is otherwise a pure texture synthesis approach, Criminisi et al. [CPT03] presented an approach that is similar to ours in the sense that, in order to propagate larger scale structures to hole regions, we also prioritise our hole filling steps according to the detection of feature lines on the surface – lines that can be considered as the 3D analogue to isophotes in images. Our approach also benefits from work presented by Drori et al. [DCOY03], who assign iteratively updated confidence values to each pixel in the image and exploit these confidence values for guiding the filling steps.

2.2. Surface Completion

As 3D data-acquisition generally leads to incomplete surfaces, the need to fill holes in 3D surfaces is traditionally part of surface reconstruction algorithms (see [CL96] as an example), but has also achieved recent research attention in its own right [DMGL02, VCBS03, Lie03, CDD*04].
Lifting the 2D-surface into a 3D volumetric representation, Davis et al. [DMGL02] extend a signed distance function that is initially only defined close to the known surface to the complete space using volumetric diffusion, thereby completing the surface even for non-trivial hole boundaries. Clarenz et al. [CDD*04] cover surface holes minimizing Willmore energy functionals, leading to surface patches with guaranteed continuity properties.
In some applications, however, smooth filling of holes is not sufficient; this is particularly the case in cultural heritage applications, where virtual museums require visual appealing reconstructions of cultural heritage objects.In such applications, so-called surface inpainting algorithms are needed that do not only reconstruct the basic geometry of the defective object, but also their fine scale geometric features. Although the problem of completing 2D images appears conceptually almost identical to completing 2D surfaces in 3D, transferring successful techniques from image completion to 3D is far from trivial. The lack of a regular grid deprives us from the universal parameter domain that is so extensively exploited in 2D image processing. As a consequence, already the construction of multi-scale hierarchies representing different frequency bands – apparently a key ingredi-

$h = 0, \ldots, H$	Hierarchy level (H for coarsest)
$\mathcal{P}^h = \{\mathbf{p}_i^h\}$	Point sets (hierarchy level h)
$\mathcal{B}^h = \{\mathbf{b}_i^h\} \subset \mathcal{P}^h$	Sets of border points
$\mathcal{C}^h = \{\mathbf{c}_i^h\} \subset \mathcal{P}^h \setminus \mathcal{B}^h$	Candidate sets
$\alpha : \mathcal{P}^h \mapsto [0,1]$	Confidence value
$\mathcal{N}(\mathbf{p}_i^h) \subset \mathcal{P}^h$	Local neighbourhood of p
$\mathcal{G}^h$	Guidance surface
n, n^h	Number of points in $\mathcal{P}^h$
N	Number of points in the descriptor
$\chi : \mathcal{P}^h \rightarrow \mathbb{R}^{2N}$	Descriptor

Table 1: *Notation and Symbols*

ent to many image completion approaches – proves to be challenging, as the vertices' positions at the same time encode both, signal *and* domain of the function to be analysed [GSS99, Tau95, PG01]. To our best knowledge, there are yet only few publications that address the problem of detail preserving during hole filling [SK02, SACO04]. Adapting technologies from exemplar-based image synthesis methods and similar in concept to our approach, Sharf et al. [SACO04] fill holes by copying existing surface patches from the object to the hole region. The fundamental problem of this algorithm is that it is completely octree-based: Holes in the surface are detected by checking for near-empty octree cells, different scales in their hierarchical approach are represented through octree levels, descriptors are based on a regular sampling of a distance field, and, most importantly, patches to be copied can be generated from other octree cells only. As a consequence, even perfectly symmetrical objects can only be reconstructed if the symmetry axis of a symmetrical feature happens to coincide with an octree axis (or one of the considered, discrete rotations thereof). Furthermore, due to the resulting non-invariance with respect to rotation, translation and scaling, very careful parameter tuning is required to successfully reconstruct real-world examples.

3. Overview and Terminology

Given a point set $\mathcal{P} \subset \mathbb{R}^3$ representing a manifold surface, the goal of our algorithm is to fill any existing holes suitably, i.e. taking into account the object's local and global context. This goal is achieved by first identifying *Boundary Points*, i.e. points that are close to regions in the point set with insufficient sampling, and then by copying appropriate local neighbourhood patches (so-called *fragments*) from a candidate set to the hole region. This way the hole is iteratively closed. As newly inserted points influence later filling steps, we assign to every point in the point set a confidence value, which is equal to 1 for all points in the original point set and is in the interval $[0,1)$ for inserted points.

In accordance with the terminology in texture synthesis approaches, we call the regions close to insufficient sampling *Target Fragments* and regions from where points to be inserted are drawn are called *Source* or *Candidate Fragments*. With the notation given in table 1, the basic workflow of our algorithm can best be seen in pseudo-code:

Fill (Point Set Hierarchy $\mathcal{P}^H, \ldots, \mathcal{P}^0$)
compute initial guidance surface $\mathcal{G}^H$
for all $h = H, \ldots, 0$ **do**
  $\mathcal{B}^h \leftarrow$ find boundary points in $\mathcal{P}^h$
  $\mathcal{C}^h \leftarrow$ find candidate points in $\mathcal{P}^h$
  compute descriptors $\chi(\mathcal{C}^h)$ and $\chi(\mathcal{B}^h)$ using $\mathcal{G}^h$
  $\mathcal{Q} \leftarrow$ prioritise $\mathcal{B}^h$
  while $\mathcal{Q}$ not empty **do**
    $b \leftarrow \text{top}(\mathcal{Q})$
    find best matching candidate $c \in \mathcal{C}^h$
    copy $\mathcal{N}(c)$ to $\mathcal{N}(b)$
    update $\mathcal{B}^h$ and $\mathcal{Q}$
  end while
  $\mathcal{G}^{h-1} \leftarrow \text{MLS}(\mathcal{P}^h)$
end for

The overall approach is hierarchical, i.e. it reconstructs the surface in the hole region on coarse scales first and exploits the result to derive the guidance function for the next levels. Hence, the first step in our algorithm is to compute a point set hierarchy, consisting of H point sets $\mathcal{P}^0, \ldots, \mathcal{P}^H$, where $\mathcal{P}^0$ is the original point set and $\mathcal{P}^1, \ldots, \mathcal{P}^H$ are smoothed and (optionally) subsampled copies thereof. For clarity of presentation, though, we describe a non-hierarchical, 1-level-formulation of our approach first, before we motivate and present the hierarchical formulation in section 5.

4. Non-Hierarchical Formulation

Suppose we are given a point set $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \subset \mathbb{R}^3$. Following the notion from 2D-image synthesis, we define for every point $\mathbf{p} \in \mathcal{P}$ in conjunction with a local frame F_p and a radius $\rho \in \mathbb{R}$ a corresponding *surface fragment* $\mathcal{N}_\rho(\mathbf{p}) \subset \mathcal{P}$ as

$$\mathcal{N}_\rho(\mathbf{p}) = \{\, \mathbf{p}_i \in \mathcal{P} \mid d(\mathbf{p}, \mathbf{q}(\mathbf{p}_i)) \leq \rho \,\},$$

where $\mathbf{q}(\mathbf{p}_i)$ is the projection of $\mathbf{p}_i$ into the plane defined by F_p. In order to establish the defining local frame, we take the best fitting plane to the k-nearest neighbours of $\mathbf{p}$, as suggested in [HDD*92], and use it as parameter plane for the fragment and its normal as surface normal in $\mathbf{p}$. Given this frame, the points in the fragment can efficiently be collected traversing a proximity graph (as suggested in [KZ04]), starting at $\mathbf{p}$ and proceeding in a *best first manner*, always adding the point with smallest $d(\mathbf{p}, \mathbf{q}(\mathbf{p}_i))$ to the fragment, until no more points inside ρ are encountered [WK04]. Please note that we use the terms *fragment* and *local neighbourhood* synonymously throughout this paper, and that we suppress the index ρ in unambiguous cases, as we do with the index h.

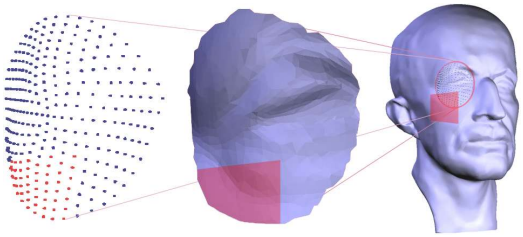


Figure 1: *Illustration of a local point set neighbourhood (triangulated for display purposes, centre) and its regularly resampled counterpart (left). Hole regions in the original surface (red square) lead to invalid descriptor components (colored in red).*

4.1. Neighbourhood Descriptors

Unlike 2D image fragments, 3D surface fragments constitute an irregular and unstructured sampling of a surface. As a consequence, there is no canonic distance measure to quantify the alikeness of two fragments. Therefore, a neighbourhood descriptor (together with a corresponding distance function) has to be defined. In a recent approach, Zelinka et al. [ZG04] have shown so-called *Geodesic Fans* to faithfully identify regions on a surface that are geometrically similar. Their descriptor is a vector of N discrete samples of one or several signals defined on the surface. The samples are taken at N fixed sample positions according to some sampling pattern given in geodesic polar coordinates.

Dealing with point sets, the computation of geodesics is an

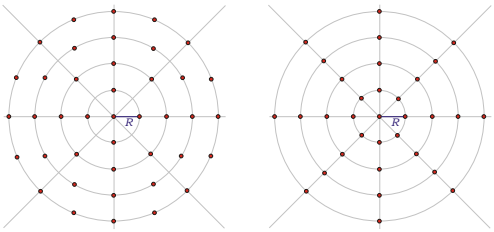


Figure 2: *Descriptor layout. Left: The number of sample points per ring grows linear with respect to its length, i.e. the sampling rate for each ring is identical (Four samples per $2\pi \cdot R$ in the depicted case, where R is the innermost ring's radius). Right: Descriptor as suggested in [ZG04]; here, the number of sample points per ring is constant, such that the sampling rate decreases linearly.*

ill-defined and expensive operation. Therefore, we interpret these N sample positions as polar coordinates in the parameter plane described above, scaled to fit into the parameter domain of the fragment. Also, we use two signals for our descriptor, the height of the fragment over the parameter plane, as well as its confidence at that position. Thus our descriptor for $\mathbf{p}$ becomes

$$\chi(\mathbf{p}) = \left(\xi_1^{\mathbf{p}}, \ldots, \xi_N^{\mathbf{p}}, \alpha_1^{\mathbf{p}}, \ldots, \alpha_N^{\mathbf{p}}\right)^t \in \mathbb{R}^{2N}$$

We chose the sampling pattern depicted in fig. 2 (left) as it does not emphasize the regions close to the centre.

To compute the descriptor $\chi(\mathbf{p})$ we construct a 2D Delaunay triangulation of the projection of $\mathcal{N}(\mathbf{p})$ into the parameter plane yielding an efficient height field representation with both height and confidence values being interpolated linearly inside the triangles.

Obviously, for points close to the boundary of a hole, the parameter domain will stretch into regions containing no points. In order to reliably detect all sample positions of the descriptor that fall into this empty region, we add as constraints to the triangulation the projected edges of the boundary loop (see sec. 6). Then all sample points outside the convex hull of the fragment or within a triangle with all vertices boundary receive a confidence value of zero. We call these descriptor components *invalid*.

Finally, as in [ZG04], the distance between two descriptors is given by a minimisation over a set of transformations φ corresponding to discrete rotations and mirroring (We perform linear interpolation on the rings if necessary due to the rotations):

$$\delta(\chi(\mathbf{p}), \chi(\mathbf{q})) = \min_{\varphi} \tilde{\delta}(\chi(\mathbf{p}), \chi(\mathbf{q}_\varphi)), \qquad (1)$$

with

$$\tilde{\delta}(\chi(\mathbf{p}), \chi(\mathbf{q})) = \frac{1}{\sum_i \alpha_i^{\mathbf{p}} \alpha_i^{\mathbf{q}}} \sum_{i=1}^{N} \alpha_i^{\mathbf{p}} \alpha_i^{\mathbf{q}} (\xi_i^{\mathbf{p}} - \xi_i^{\mathbf{q}})^2$$

4.2. 1-Level Inpainting

The basic idea is now to find for every $\mathbf{b} \in \mathcal{B}$ an appropriate candidate $\mathbf{c} \in \mathcal{P}$ and to copy its neighbourhood to the invalid parts of $\mathcal{N}(\mathbf{b})$. To guarantee that invalid regions in $\mathcal{N}(\mathbf{b})$ can indeed be filled with the corresponding regions in $\mathcal{N}(\mathbf{c})$, the candidate set $\mathcal{C}$ is built by collecting all points $\mathbf{p} \in \mathcal{P}$, whose descriptors do not contain any invalid components, i.e.

$$\mathcal{C} = \left\{ \mathbf{c} \in \mathcal{P} \mid \xi_i^{\mathbf{c}} \text{ valid } \forall i \right\}.$$

With a suitable candidate set and a discriminative descriptor at hand, inpainting simply consists of finding the best matching candidate $\mathbf{c}_b$ for any boundary point $\mathbf{b}$ and co-aligning $\mathbf{c}_b$'s local frame with the frame of $\mathcal{N}(\mathbf{b})$.

Here, we also apply the minimising transformation φ according to eqn. 1. In addition to that, we also perform one ICP-step (taking into account the descriptor samples only and using fixed correspondences) in order to compensate for little deviations that might result from the discreteness of our set of considered rotations.

Finally, all points from $\mathcal{N}(\mathbf{c})$ corresponding to invalid regions in $\chi(\mathbf{b})$ are inserted into $\mathcal{P}$ and the sets $\mathcal{B}$ and $\mathcal{C}$ are updated.

4.3. Ordering of the Filling Operations

Local neighbourhoods of boundary points cannot be guaranteed to cover also a part of "the other side" of the hole (in fact, in most cases they won't). In cases when the descriptor gives no hint to what the basic geometry of the missing surface is like, this might lead to inappropriate candidate selection, as illustrated in figure 3. This is especially the case for small neighbourhood sizes, where the valid region captures only very local properties of the surface, and reflects the fact that the local neighbourhood size must correspond to the *scale* of the feature to be reconstructed. (We will discuss this into more detail in section 5). To address this problem, we prioritise the atomic filling operations, much in the spirit of the fragment-based image completion approach by Criminisi et al. [CPT03]. The main idea is to select the most discriminative target fragments for first filling. To measure the expressiveness of a fragment, we compute the standard deviation σ of the descriptor values along the sampling rings depicted in fig. 2. By means of this criterion, regions of high curvature are preferred over flat regions.

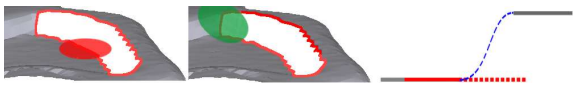


Figure 3: *Filling operations in planar regions (left) might lead to undesired extension of the planar structure.*

In addition to this, we want to favor target points where only little geometry is missing over those points that lie comparably isolated in large undersampled regions. This is measured by means of an aggregated confidence value $\alpha(\chi_{\mathbf{p}}) = \frac{1}{N}\sum_i \alpha(\xi_i^{\mathbf{p}})$ that is computed for every target descriptor.

We experimented with several combinations of the two criteria α and σ to prioritise the filling operations. According to our experiments, a threshold approach performed best: Among those target descriptors that have the highest confidence value (quantised into ten bins), we choose the one with highest σ to be filled first. This way, of those target fragments with a high confidence we favor the most discriminative.

The heuristic approach presented here helps to resolve problems as illustrated in figure 3. In this case our algorithm correctly chooses the target fragment indicated as a green disc in the centre image to be processed first. It is not, however, sufficient to faithfully fill larger holes with smaller neighbourhood sizes, that might be required to reconstruct also fine detail properties of the object.

4.4. Fragment Blending

The distance function δ defined in eqn. 1 minimises the alignment error of fragments to be pasted into hole regions in the least squares sense, and therefore does not always prevent the appearance of local discontinuities at the boundary. To ameliorate this, we perform an additional blending step, based on local diffusion of the descriptor values. First, all confidence values α_i in the target descriptor are set to $\lceil \alpha_i \rceil$. We then perform $r/2$ diffusion steps affecting only those entries with an initial confidence value of zero, where r is the number of rings in the descriptor. For each step $t \rightarrow t+1$, we set

$$\xi_i^{t+1} = \frac{\sum_{j\in N_i} \alpha_j^t \xi_j^t}{\sum_{j\in N_i} \alpha_j^t} \quad \text{and} \quad \alpha_i^{t+1} = \frac{\sum_{j\in N_i} \alpha_j^t}{|N_i|}.$$

Here N_i denotes the index set of the four neighbouring descriptor samples of ξ_i. The blended height $\tilde{h}$ of an incoming point $\mathbf{q}$ in the target frame is then given by

$$\tilde{h} = (1-\alpha_q)h + \alpha_q \xi_q,$$

where α_q and ξ_q are the entries of the diffused descriptor at the corresponding position, estimated by bilinear interpolation. This way, we allow inserted points farther away from confident target points to deviate from these target points' height values.

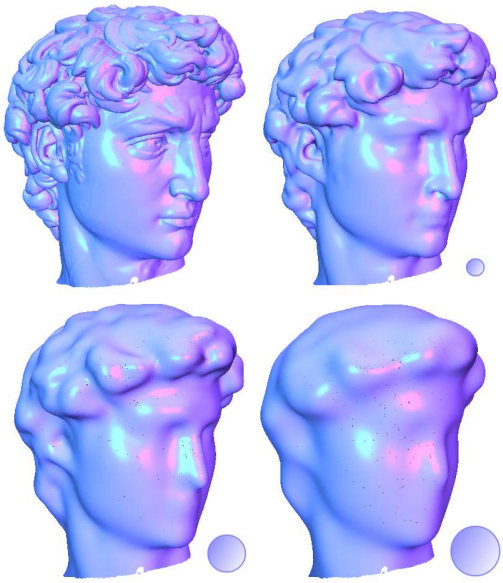


Figure 4: *David's head, subsampled to 300000 points, original (top left), iteratively smoothed once with $k = 100$ (top right), after 5 (bottom left), and 8 iterations (bottom right). The discs indicate the corresponding neighbourhood size.*

5. Hierarchical Formulation

The essence of exemplar-based completion is to exploit coherence and similarity between the region of interest and appropriate candidate regions of the considered object. Geometric properties of the hole region, though, might be represented in different scales, and in many cases, similarity relations present in different scales correspond to very different

regions on the object. It is therefore important to allow candidates to stem from the optimal object region *per scale*, such that for instance the bunny's missing left knee (see fig. 9) is reconstructed on coarse levels by copying the bunny's right knee, whereas the fur structure, exhibiting different similarity relations is reconstructed from various different locations on the bunny's back.

Only in the presence of real symmetry, where similarity on all considered scales happens to relate to the same candidate region, the one-level approach described in the previous section is sufficient. For instance, the missing left eye and nose region of the Max-Planck-model (as shown in fig. 8) can be reconstructed using transferred and mirrored copies of parts of the opposite side of the face. This type of similarity relation ranging over all considered scales – rarely encountered when dealing with images – is relevant for large classes of 3D objects. Nevertheless, in order to handle cases as described above, we propose a hierarchical, multi-level approach, whose first step is to create a point set hierarchy $\mathcal{P}^1,\ldots,\mathcal{P}^H$ with according scales $\rho_1,\ldots,\rho_H$, each point set representing the (defective) object and its geometrical properties up to the according scale.

5.1. Creating the Point Set Hierarchy

We approximate the scale-space representation of the input model by iteratively applying Laplacian smoothing, deriving coarser and coarser scales, corresponding to ever larger kernel widths. Specifically, to derive $\mathcal{P}^{h+1}$ from $\mathcal{P}^h$ we compute new point positions

$$\mathbf{p}^{h+1} = \frac{1}{\mu}\sum_{j=1}^{k}\mu_j\mathbf{p}_j^h,$$

as the weighted mean of all k-nearest-neighbours $\mathbf{p}_j^h$ of $\mathbf{p}^h$, where $\mu = \sum_j \mu_j$. (Actually, we perform the smoothing in direction of the surface normal only, as we wish to smooth the surface itself, rather than the distribution of sample points *in* the surface). This corresponds to smoothing $\mathcal{P}^h$ with a kernel width of

$$\rho_{\mathbf{p}} = \max_{j=1,\ldots,k} d(\mathbf{p},\mathbf{p}_j^h).$$

The average distance to the k^{th}-nearest neighbour

$$\rho = \frac{1}{n}\sum \rho_{\mathbf{p}}$$

is called the *k-Ring Radius* and describes a natural size of the local neighbourhood patches, as it contains all the detail information up the respective hierarchy level, with all higher level detail information smoothed out (see fig.4).
We are aware that our smoothing scheme has two main drawbacks: On the one hand, it is a well known fact that Gaussian filtering causes shrinkage and ultimately converges to a single point. On the other hand, as the points contributing to the new, filtered point positions are a fixed number of nearest neighbours, the sampling density influences the smoothing. Strictly speaking, claiming that a certain "scale" is represented in a smoothed point set, therefore holds only for uniformly sampled point sets.
To address these drawbacks, more sophisticated filtering methods could be applied, in the spirit of [KH98, CDR00] or [Tau95]. However, in our setting the approximated scales are used to guide the filling process only and therefore we found our simple approach to be sufficient.

5.2. Multi-Level Inpainting

Based on the point set hierarchy $\mathcal{P}^0,\ldots,\mathcal{P}^H$, we formulate the inpainting as a bottom-up process, filling the hole in the coarsest scale representation $\mathcal{P}^H$ first and then consecutively on the finer levels up to the finest level $\mathcal{P}^0$. In each step (aside from the first step: $\mathcal{P}^H$ is completed using the non-hierarchical formulation of our algorithm as described in sec. 4), we use the previous, next coarser level point set to construct a guidance surface that can be used in the target descriptors for the filling step on the current level. This way we can encode hints to the larger scale geometry into the descriptor components that have been invalid till now and hence neglected.

Let the *Guidance Surface* $\mathcal{G}^h$ be any implicit representation of the (completed) point set $\mathcal{P}^{h+1}$. In our approach, we use the zero set of the MLS-approximation of $\mathcal{P}^{h+1}$, but any other locally evaluable representation could also be applied. A straight-forward approach (that would also resemble comparable approaches in 2D image processing) would then be to assign height values to invalid target descriptor components by sampling $\mathcal{G}^h$ (see fig. 5, bottom left). This straight-forward approach, however, would have the adverse effect that even ideal candidates would not be considered a perfect match. The reason for this is that inserting samples from $\mathcal{G}^h$ to the current level's descriptor inherently causes two scales to be mingled. The resulting *hybrid* descriptor – incorporating two scales at the same time – is in fact comparable to descriptors on neither the current level h nor the coarser level h+1. This fact is illustrated in fig. 5.

5.3. 2-Layer Descriptor

As a consequence, we define 2-layer descriptors as illustrated in fig. 6:

- The first layer χ_h is constructed as described in sec. 4.1, capturing the available local geometry from $\mathcal{P}^h$ only, and assigning zero confidence to the invalid descriptor components.
- For the second layer χ_{h+1}, we use the same parameter plane and the same sampling pattern, but height values are derived from the zero level set of the MLS-approximation of $\mathcal{P}^{h+1}$.

The distance function for the two-layer descriptor is then simply a weighted sum of the distance functions per level:

$$\delta(\chi(\mathbf{p}),\chi(\mathbf{q})) = \delta(\chi_h(\mathbf{p}),\chi_h(\mathbf{q})) + \tau\,\delta(\chi_{h+1}(\mathbf{p}),\chi_{h+1}(\mathbf{q})).$$

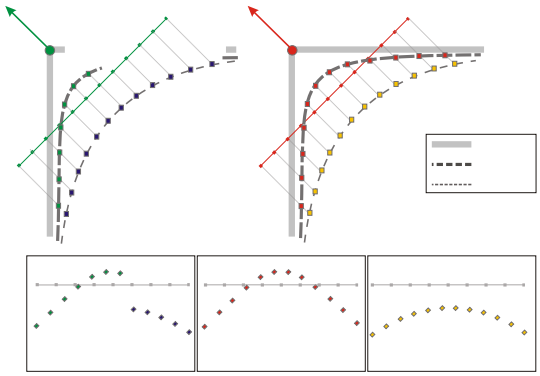


Figure 5: *Defective target surface and an ideal candidate (bold), together with two levels from the scale space representation (dashed, level h+1 filled, level h incomplete). Updating target descriptor values invalid on level h using the guidance surface from level h+1 leads to a descriptor (bottom left) that is not well comparable with either of the candidate descriptors (bottom centre / right).*

While the parameter τ is arbitrary in principle, a value of 0.5 has proven to produce good results in our experiments. In cases where multiple hierarchy levels are reconstructed, it is advisable to increase τ for finer levels, as they can be expected to be already a reliable reconstruction.

6. Hole Detection

Point set surfaces are by nature unstructured and do not contain any adjacency or connectivity information. As a consequence, defining and detecting holes – a trivial task in the case of surfaces given as triangle mesh – is an ill-defined problem in the case of point sets. Nevertheless, the robust detection of connected hole boundaries is compulsory for our algorithm.

The basic layout of our hole detection scheme is as follows: For each point $\mathbf{p} \in \mathcal{P}$ we compute a boundary probability $\pi(\mathbf{p})$, reflecting the probability that $\mathbf{p}$ is located on or near a hole in the surface sampling. Thereafter, we exploit that the boundary points we are looking for have proximate neighbours that are also boundary, and construct closed loops circumscribing the hole in a shortest cost path manner.

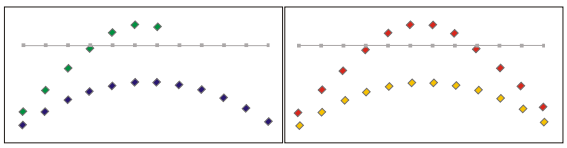


Figure 6: *2-Layer descriptor for a target and an ideal candidate region.*

6.1. Boundary Probabilities

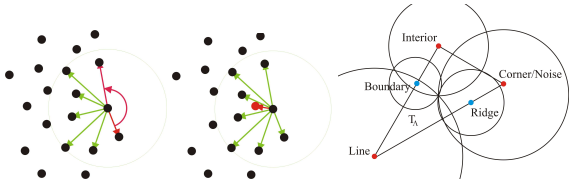


Figure 7: *Left:The angle criterion is based on the maximum angular gap. Centre: The halfdisc criterion computes the difference vector to the average of its neighbours, that points in direction of the interior surface. Right: The triangle formed by all Λ values and the characteristic points for certain shapes.*

We use three criteria to identify points lying on a surface boundary. All compute a boundary probability measure based on the local neighbourhood $\mathcal{N}(\mathbf{p})$ for each sample point $\mathbf{p}$.

The angle criterion

The angle criterion, that is described in [GWM01], [LP02] and [MD04], projects all neighbouring points to the tangent plane and sorts them according to their angle around the centre sample. The largest gap g between two consecutive neighbors is computed and the boundary probability is given as

$$\pi_{angle}(p) = \min\left(\frac{g - \frac{2\pi}{k}}{\pi - \frac{2\pi}{k}}, 1\right).$$

To make the angle criterion less susceptible to small inaccuracies in the normal direction, we modify the standard angle criterium and ignore all points in $\mathcal{N}(\mathbf{p})$ whose difference vector with $\mathbf{p}$ is almost in normal direction, i.e. with $\angle(\mathbf{n}, \mathbf{p}_i - \mathbf{p}) < 10$. This is especially beneficial in the presence of noise and in case of point clouds constructed from multiple range images, as here small errors in normal direction often cannot be avoided.

The halfdisc criterion

Requiring our point set to be a representation of a manifold surface (with boundary), the neighbourhood of points on the boundary is homeomorphic to a halfdisc. As a consequence, the average of the neighbours will deviate from the centre point in direction of the interior surface. In fact, we expect it to lie in the centre of mass of a halfdisc in the plane, see figure 7. Thus we compute the average of the neighborhood and project it onto the tangent plane. The boundary probability can then be deduced from the distance between the centre point and the projected average μ in the following manner:

$$\pi_{halfdisc}(\mathbf{p}) = \min\left(\frac{\|p - \mu\|}{\frac{4}{3\pi}\rho}, 1\right).$$

(Note that the centre of a halfdisc is located at a distance of $\frac{4}{3\pi}\rho$ to the centre of the full disc.)

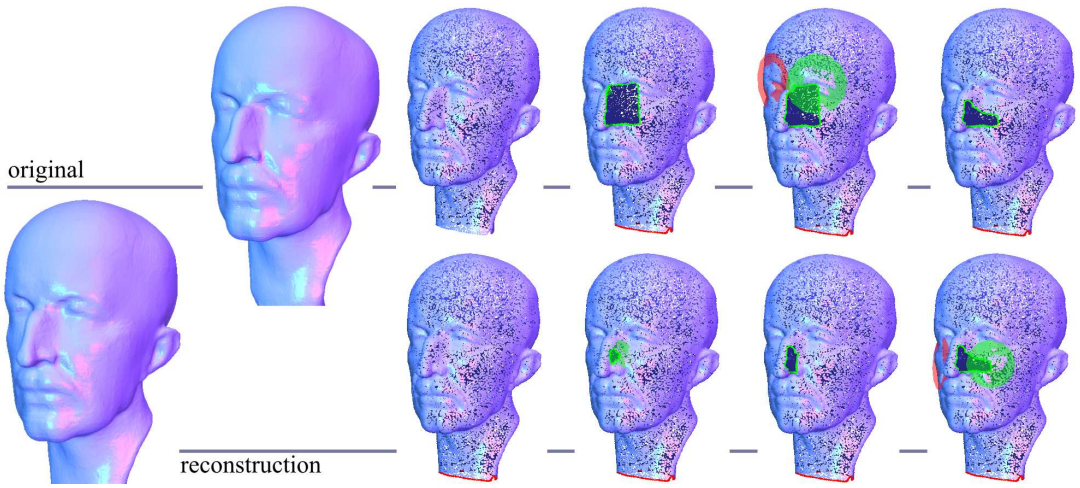


Figure 8: *Max Planck data set. The artificially introduced hole is filled successively by copying best matching candidates (red discs) to target regions (green discs).*

The shape criterion

As noted in [GWM01], the shape of the correlation ellipsoid of $\mathcal{N}(\mathbf{p})$ approximates the general form of the neighbouring points. The shape of the ellipsoid in turn is encoded in the eigenvalues $\lambda_0 \geq \lambda_1 \geq \lambda_2$ of the weighted covariance matrix of the points in $\mathcal{N}(\mathbf{p})$. We collect the relative magnitudes of the eigenvalues into a decision vector

$$\Lambda_p = \frac{1}{\lambda_0+\lambda_1+\lambda_2}\,(\lambda_0,\lambda_1,\lambda_2),$$

by which we can categorise four basic different situations (*Boundary*, *Interior*, *Corner/Noise*, and *Line*) according to the following table:

$\Lambda \approx (\frac{2}{3},\frac{1}{3},0)$	Boundary
$\Lambda \approx (\frac{1}{2},\frac{1}{2},0)$	Interior
$\Lambda \approx (\frac{1}{3},\frac{1}{3},\frac{1}{3})$	Corner/Noise
$\Lambda \approx (1,0,0)$	Line

The basic idea is now to identify boundary points $\mathbf{b}$ by the location of their decision vector Λ_b in the triangle spanned by the last three values of Λ in the table above. (see figure 7, right). This way we derive a shape probability indicated by the decision vector as $\widetilde{\pi}_\Phi(p) = \|\Lambda_p - \Lambda_\Phi\|$, (where Φ is any one of the four categories) and define the boundary probability according to the ellipsoidal criterion as

$$\pi_{shape}(p) = \frac{\widetilde{\pi}_{boundary}(\mathbf{p})}{\sum_\Phi \widetilde{\pi}_\Phi(\mathbf{p})}.$$

6.2. Closed boundary loops

Combining the three probabilities above, the total boundary probability is given as

$$\pi(\mathbf{p}) = \frac{1}{3}\left(\pi_{angle}(\mathbf{p}) + \pi_{halfdisc}(\mathbf{p}) + \pi_{shape}(\mathbf{p})\right).$$

Based on this probability, we construct closed boundary loops, that circumscribe the holes to be detected in the given point set. This is achieved by assigning to every edge $(\mathbf{p},\mathbf{q})$ in the proximity graph a cost value $\frac{1}{2d(\mathbf{p},\mathbf{q})}(\pi(\mathbf{p})+\pi(\mathbf{q}))$ and constructing a minimal spanning tree. Boundary loops are then derived by adding to this tree edges that are at the same time both minimal and closing loops that are longer than a certain threshold.

7. Results

We applied our fragment-based inpainting algorithm to various data sets of point sampled geometry. The objects depicted in the images of this paper exhibit holes in structured surface regions and are in addition to this comparably large in size. Reconstructing the surface for these holes using traditional smooth hole filling algorithms would have lead to disturbing visual artifacts. In fig. 8, the basic workflow of our algorithm can be seen. For target fragments (illustrated as green disc) an optimal candidate fragment (red disc) is identified. The points corresponding to invalid target regions are pasted into the point set after the according transformations (translation, rotation, optional mirroring), which are deduced from the descriptor comparisons, are applied. In near symmetric cases like faces, the non-hierarchical formulation of our algorithm already gives satisfying results, given that the required scale to cover the hole can be represented without scale-space segmentation.

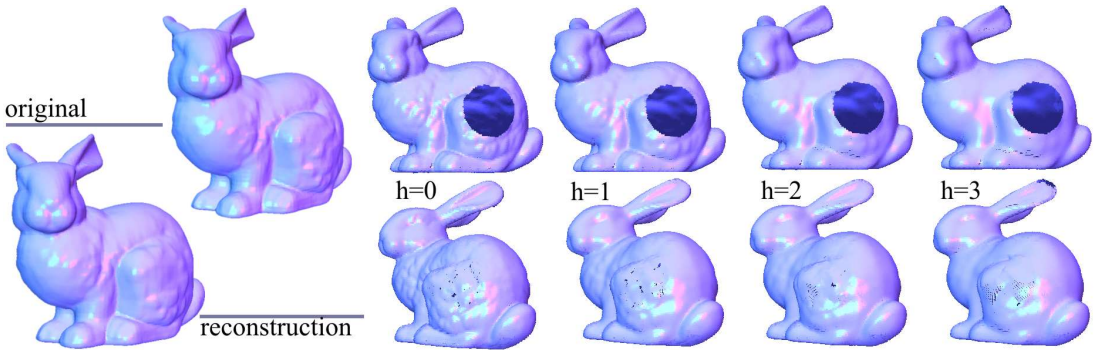


Figure 9: *Hierarchical reconstruction of the Stanford Bunny. First, a point set hierarchy (h = 0 to 3) of the defective bunny is constructed (top). Starting with level 3, each level is filled per-se, where in level h, the level $h+1$ serves as Guidance Surface (bottom).*

Fig. 9 demonstrates the use of our hierarchical formulation for the exploitation of similarities that are spread over several scales. Using our approach we were able to reconstruct both the knee as a symmetrical large scale feature and the fur structure that itself does not exhibit an analogue symmetry, but is also well presented as a coherent feature on the surface on finer scales. During the coarse level inpainting steps, corresponding target-candidate descriptor pairs were identified. In this example, prioritising the target fragments for filling according to their discriminativity was particularly useful. This way, the target regions close to the bunny's knee were selected for filling first. During the finer scale filling operations, the fur structure was transferred to the hole region from various (other) regions on the bunny's back.

In order to assess the influence of the automatically computed guidance surface and the candidate set on the inpainting results, we reconstructed the bunny data set with the help of the perfect guidance and the perfect candidate set, namely the complete point set itself. The combination of both, perfect guidance and perfect candidate set, resulted in the perfect reconstruction of the bunny (fig. 10). As opposed to that, fig. 9 shows the hierarchical reconstruction of the bunny without knowledge of the complete bunny.

We are aware that the method we use for generating a scale space representation of the object has a number of insufficiencies: It is non-uniform for irregularly sampled point clouds, causes shrinkage, and the filtering behaviour in regions around the holes differs from filtering the corresponding complete regions. This leads to non-identical scale space representations for what would have been identical regions if not for the hole. Consequently, the smoothing in boundary regions is of negative effect in cases, where, even on coarsest scale, the hole region is of considerable curvature. Excessively smoothing causes the flanks of the hole to be straightened, such that a satisfying filling of this hole already at coarsest level fails. As stated before, in these cases other sources of information could be considered for deriving guidance functions for the next finer levels.

8. Conclusions & Future Work

Inspired from exemplar-based techniques in 2D image processing, we have introduced in this paper a novel method for the filling of holes in structured point set surfaces. In order to be able to recognise and exploit similarity and coherence properties in the object, we derived target and candidate fragments given in local frames, thereby making our algorithm insensitive to similarity transformations as rotation, translation and scaling. As the construction of descriptors depending on the local frames is particularly apt for regions that are well parameterisable over the tangent plane, we are currently looking into incorporating 3D shape descriptors in the spirit of [BMP02] into our approach. Here, the handling of confidence attributes and the adaption of a 2-layer shape descriptor is a challenge.

As a consequence of our hierarchical approach based on a scale space representation of the object, our algorithm is able to robustly identify and exploit similarity relations between the region of interest and possibly various other locations on the surface, depending on the respective scale.

In our current implementation, we did not yet apply any acceleration strategy, that were proposed especially in the corresponding image completion and texture synthesis literature. In particular, the incorporation of clustered analysis methods applied on the set of candidates into our method would be an easy yet interesting direction of future work.

References

[BBC*01] BALLESTER C., BERTALMIO M., CASELLES V., SAPIRO G., VERDERA J.: Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing 10*, 8 (August 2001), 1200–1211. 2

[BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 4 (2002), 509–522. 9

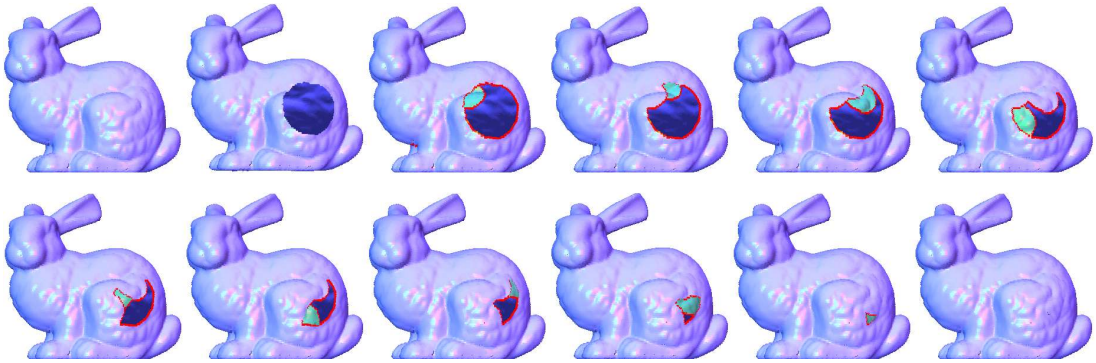


Figure 10: *From top left to bottom right: Original Bunny data set, data set with artificially introduced hole, iterative repair steps, resulting reconstruction. In this example, the candidate set and the guidance function have been built from the original bunny data set.*

[Bon97] BONET J. S. D.: Multiresolution sampling procedure for analysis and synthesis of texture images. In *Computer Graphics* (1997), ACM SIGGRAPH, pp. 361–368. 2

[CDD*04] CLARENZ U., DIEWALD U., DZIUK G., RUMPF M., RUSU R.: A finite element method for surface restoration with smooth boundary conditions. *Computer Aided Geometric Design 21*, 5 (2004), 427–445. 2

[CDR00] CLARENZ U., DIEWALD U., RUMPF M.: Anisotropic geometric diffusion in surface processing. In *Proceedings of the conference on Visualization '00* (2000), IEEE Computer Society Press, pp. 397–405. 6

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM Press, pp. 303–312. 2

[CPT03] CRIMINISI A., PÉREZ P., TOYAMA K.: Object removal by exemplar-based inpainting. In *Conference on Computer Vision and Pattern Recognition (CVPR 2003)* (Madison, WI, USA, 16-22 June 2003), IEEE Computer Society, pp. 721–728. 2, 5

[DCOY03] DRORI I., COHEN-OR D., YESHURUN H.: Fragment-based image completion. *ACM Trans. Graph. 22*, 3 (2003), 303–312. 2

[DMGL02] DAVIS J., MARSCHNER S. R., GARR M., LEVOY M.: Filling holes in complex surfaces using volumetric diffusion. In *Proceedings of the 1st Int. Symp. on 3D Data Processing Visualization and Transmission* (Padova, Italy, june 29–21 2002), Cortelazzo G. M., Guerra C., (Eds.), IEEE Computer Society, pp. 428–438. 2

[EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2* (1999), IEEE Computer Society, p. 1033. 2

[GSS99] GUSKOV I., SWELDENS W., SCHRÖDER P.: Multiresolution signal processing for meshes. *Computer Graphics Proceedings (SIGGRAPH 99)* (1999), 325–334. 3

[GWM01] GUMHOLD S., WANG X., MCLEOD R.: Feature extraction from point clouds. In *Proceedings of 10th International Meshing Roundtable* (Sandia National Laboratories, Newport Beach, CA, October 2001), pp. 293–305. 7, 8

[HB95] HEEGER D. J., BERGEN J. R.: Pyramid-based texture analysis/synthesis. In *SIGGRAPH* (1995), pp. 229–238. 2

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (1992), ACM Press, pp. 71–78. 3

[JT03] JIA J., TANG C.-K.: Image repairing: Robust image synthesis by adaptive nd tensor voting. In *Conference on Computer Vision and Pattern Recognition (CVPR 2003)* (Madison, WI, USA, 16-22 June 2003), IEEE Computer Society, pp. 643–650. 2

[KH98] KARBACHER S., HÄUSLER G.: New approach for the modeling and smoothing of scattered 3d data. In *Proceedings of the Conference on 3D Image Capture and Applications* (San Jose, CA, USA, January 1998), Ellson R. N., Nurre J. H., (Eds.), vol. 3313 of *SPIE Proceedings*, SPIE, pp. 168–177. 6

[KZ04] KLEIN J., ZACHMANN G.: Proximity graphs for defining surfaces over point clouds. In *Symposium on Point-Based Graphics* (June 2004). 3

[Lie03] LIEPA P.: Filling holes in meshes. In *SGP'03: Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2003), Eurographics Association, pp. 200–205. 2

[LP02] LINSEN L., PRAUTZSCH H.: Fan clouds - an alternative to meshes. In *Proceedings Dagstuhl Seminar 02151 on Theoretical Foundations of Computer Vision - Geometry, Morphology and Computational Imaging* (2002), Alano T., Klette R.,, Ronse C., (Eds.), Springer-Verlag Berlin Heidelberg, p. [10]. 7

[MD04] MOENNING C., DODGSON N.: Intrinsic point cloud simplification. In *Proc. 14th GraphiCon* (Moscow, September 2004), vol. 14. 7

[NA03] NEALEN A., ALEXA M.: Hybrid texture synthesis. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 97–105. 2

[Per85] PERLIN K.: An image synthesizer. In *Proc. of the 12th annual conference on Computer graphics and interactive techniques* (July 1985), vol. 19(3), pp. 287–296. 2

[PG01] PAULY M., GROSS M.: Spectral processing of point-sampled geometry. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, pp. 379–386. 3

[PS00] PORTILLA J., SIMONCELLI E. P.: A parametric texture model based on joint statistics of complex wavelet coefficients. *Intern. Journal of Computer Vision 40*, 1 (2000), 49–70. 2

[SACO04] SHARF A., ALEXA M., COHEN-OR D.: Context-based surface completion. *ACM Trans. Graph. 23*, 3 (2004), 878–887. 3

[SK02] SAVCHENKO V., KOJEKINE N.: An approach to blend surfaces. *Advances in Modeling, Animation and Rendering* (2002), 139–150. 3

[Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), ACM Press, pp. 351–358. 3, 6

[Tur01] TURK G.: Texture synthesis on surfaces. In *SIGGRAPH 2001* (2001), pp. 347–354. 2

[VCBS03] VERDERA J., CASELLES V., BERTALMIO M., SAPIRO G.: Inpainting surface holes. In *IEEE International Conference on Image Processing (ICIP 2003)* (Barcelona, Spain, September 2003). 2

[WK04] WU J., KOBBELT L.: Optimized sub-sampling of point sets for surface splatting. vol. 23, pp. 643–652. 3

[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* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 479–488. 2

[ZG04] ZELINKA S., GARLAND M.: Similarity-based surface modelling using geodesic fans. In *SGP'04: Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing* (July 2004), Eurographics Association. 4

[ZWM98] ZHU S. C., WU Y., MUMFORD D.: Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *Int. J. Comput. Vision 27*, 2 (1998), 107–126. 2