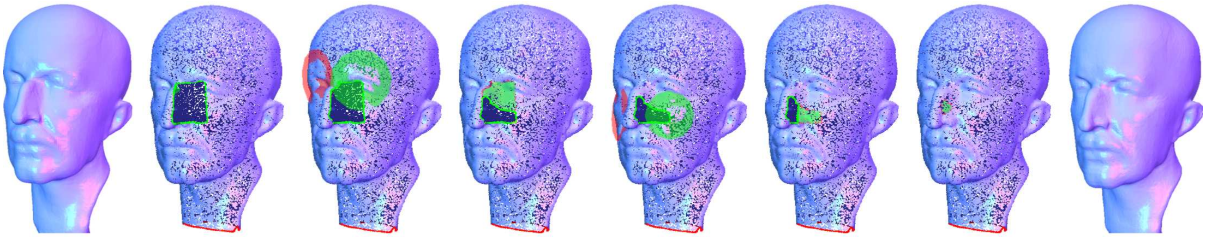
The 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage
VAST (2005)
M. Mudge, N. Ryan, R. Scopigno (Editors)

Detail-Preserving 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.

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. 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. In some applications, 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 *plausibly*, 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. 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. The two key challenges for such a 3D fragment-based approach are

- to identify symmetry, similarity and coherence relationships in the scene or on the object
- to exploit these relationships to inpaint surface regions.

Before we describe in detail how these challenges can be tackled, we'll shortly review the relevant literature in the following section, covering the inspiring works on 2D image processing, 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 a multi-scale approach.

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

Image completion 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 colours 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 minimising Willmore energy functionals, leading to smooth surface patches with guaranteed continuity properties.

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

Table 1: *Notation and Symbols*

Smooth completion is also the result of very recently published hole filling algorithms, where templates – constructed from some known basis mesh [KS05] or from a partly manual selection from a shape data base [PMG*05] – have been exploited.

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 ingredient 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

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

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 plausibly, 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. By these means, we are able to evaluate each point's confidence and adjust its influence on the algorithm appropriately.

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 (top of the page). 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 L point sets $\mathcal{P}^0, \ldots, \mathcal{P}^L$, where $\mathcal{P}^0$ is the original point set and $\mathcal{P}^1, \ldots, \mathcal{P}^L$ 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 (see fig. 1). 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]. 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 l.

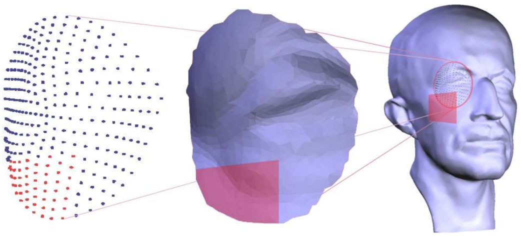


Figure 1: *Illustration of a local point set neighbourhood (triangulated for display purposes, centre) and its regularly resampled counterpart (left). Hole regions in the original surface (red square) lead to invalid descriptor components (coloured in red).*

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

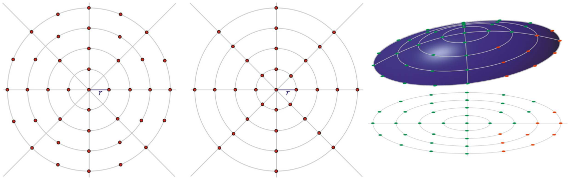


Figure 2: *Descriptor layout. Left: The number of sample points per ring grows linear with respect to its length, i.e. the sampling rate for each ring is identical (Four samples per $2\pi r$ in the depicted case). Centre: Descriptor as suggested in [ZG04]; here, the number of sample points per ring is constant, such that the sampling rate decreases linearly. Right: 3D illustration of the local resampling.*

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 (fig. 2, middle).

Dealing with point sets, the computation of geodesics is an ill-defined and expensive operation. Nevertheless, we adopt the geodesic fan approach to our setting by deriving a locally regular resampling of the point cloud that can then be used to come up with a straight-forward descriptor. To this end, we choose the sampling pattern depicted in fig. 2 (left) (as it does not emphasise the regions close to the centre) and interpret these N sample positions as polar coordinates in the parameter plane described above, scaled to fit into the parameter domain of the fragment. We define as the *fragment shape descriptor* the vector of height values of this local regular resampling over the parameter plane:

$$\chi(\mathbf{p}) = \left(h_1^{\mathbf{p}},\dots,h_N^{\mathbf{p}}\right)^t \in \mathbb{R}^N.$$

Please note that although on coarser scales where fragments are of larger size (see below) a considerable number of points may have to be resampled, this can be performed efficiently using natural neighbour interpolation techniques that are performed directly on the GPU [KEHKL*99]. Alternatively, interpolated height values for the new sample points can also be computed quite efficiently by constructing a 2D Delaunay triangulation and using the barycentric coordinates of the sample positions for interpolation.

An obvious choice for a distance metric on the space of descriptors would be the weighted *Mean Squared Error*

$$\delta(\chi(\mathbf{p}),\chi(\mathbf{q})) = \frac{1}{N}\sum_i w_i||h_i^{\mathbf{p}} - h_i^{\mathbf{q}}||^2, \qquad (1)$$

with some appropriate weights w_i. However, as the sampling pattern is uniquely defined only up to the direction of the x-axis (see fig. 2), we allow for a set of transformations φ corresponding to discrete rotations and mirroring to be applied to the descriptor before evaluation of eqn. 1 and perform linear interpolation along the rings where necessary [ZG04].

Obviously, for some points in the data set, the parameter domain will stretch into regions containing points with confidence value < 1 (points that have been inpainted in some previous step) or no points at all. The basic idea is that descriptor values corresponding to these regions should contribute less to the selection of appropriate target-candidate pairs. This is achieved by setting the weights in eqn. 1 to $w_i = \alpha_i(\mathbf{p}) \cdot \alpha_i(\mathbf{q})$, where the $\alpha_i(\mathbf{p})$ are the interpolated confidence values of the points in $\mathcal{N}(\mathbf{p})$. For descriptor values corresponding to empty regions in $\mathcal{N}(\mathbf{p})$ we assign a confidence value of zero.

4.2. 1-Level Inpainting

Before we can start reconstructing the missing surface in hole regions, we have to detect the hole boundary first. While trivial in the case of triangle meshes, this boundary detection is a difficult problem when dealing with point sets. To solve this, we use the approach very recently presented in [Sch05], that robustly identifies loops of boundary points. The basic concept is now to find for every boundary point $\mathbf{b} \in \mathcal{B}$ an appropriate candidate $\mathbf{c} \in \mathcal{P}$ and to copy its neighbourhood to the invalid (empty) 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.

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

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 $\mathcal{N}(\mathbf{b})$.

In order to reduce the time required for searching a best matching candidate, we apply the tree structured vector quantisation method (TSVQ, [WL00]) to the candidate set. By means of the TSVQ the search for a best matching candidate is significantly accelerated and renders the filling procedure interactive even for large candidate sets.

In addition to the minimising transformation from sec. 4.1, we also perform one ICP-step for the best matching candidate, 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 $\mathcal{P}$, receiving the aggregated confidence from eqn. 2 of the target descriptor used to compute the match. Afterwards the sets $\mathcal{B}$ and $\mathcal{C}$ are updated.

4.3. Structure-Driven Inpainting

So far, we have described an algorithm that performs filling in random order. Analogue to what was noted for images by Criminisi et al. [CPT03], this often has the adverse effect that flat surface regions are undueley propagated into the hole region. Our algorithm tackles this by assigning priority values to all possible targets and performing a best-first filling algorithm. The priority values are computed to favour those targets which are on the continuation of strong features and are surrounded by a high confidence neighbourhood.

Unfortunately, feature detection on point sets in itself is a non-trivial task, let alone in the presence of holes. We therefore use a simple heuristic 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. Please note that this criterion is well-adapted to our hierarchical setting described in sec. 5, as here the fragment size corresponds to the amount of detail contained in the fragment and therefore σ faithfully encodes the probability of the presence of a feature on the resp. scale.

The confidence of the neighbourhood is measured by means of the aggregated confidence value

$$\alpha(\chi_{\mathbf{p}}) = \frac{1}{N}\sum_i \alpha(h_i^{\mathbf{p}}) \qquad (2)$$

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 favour the most discriminative.

It is worth noting that our algorithm can of course trivially be modified to a semi-automatic approach, where a few appropriate candidates are suggested to the user, who then selects the one to be pasted into the target region.

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 figs. 6/7) 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 on the title page of this paper) 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 $\mathcal{P}^1,\ldots,\mathcal{P}^L$ with according scales $\rho_1,\ldots,\rho_L$, each point set representing the (defective) object and its geometrical properties up to the according scale.

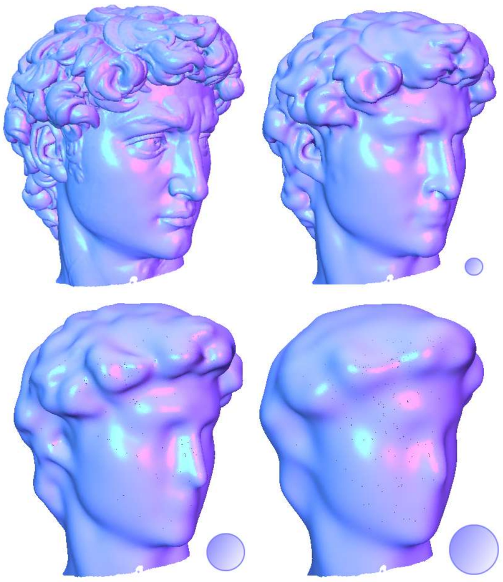


Figure 3: *David's head, subsampled to 300000 points, original (top left), iteratively smoothed once with $k = 100$ (top right), after 5 (bottom left), and 8 iterations (bottom right). The discs indicate the corresponding neighbourhood size.*

5.1. 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 $\mathcal{P}^{l+1}$ from $\mathcal{P}^l$ we compute new point positions $\mathbf{p}^{l+1} = \frac{1}{\mu}\sum_{j=1}^{k}\mu_j\mathbf{p}_j^l$, as the weighted mean of all k nearest neighbours $\mathbf{p}_j^l$ of $\mathbf{p}^l$, 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 $\mathcal{P}^l$ with a kernel width of

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

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 to the respective hierarchy level, with all higher level detail information smoothed out (see fig.3).

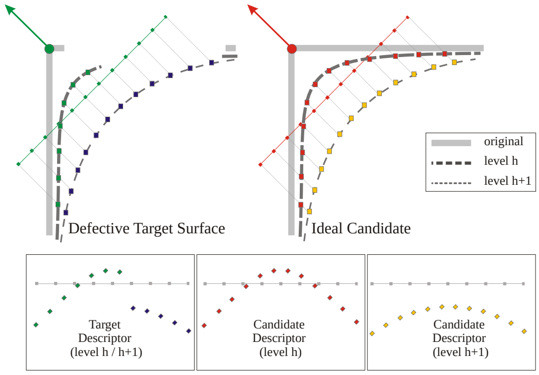

Defective Target Surface
Ideal Candidate
original
level h
level h+1
Target Descriptor (level h / h+1)
Candidate Descriptor (level h)
Candidate Descriptor (level h+1)

Figure 4: *Defective target surface and an ideal candidate (bold), together with two levels from the scale-space representation (dashed, level l+1 filled, level l incomplete). Updating target descriptor values invalid on level l using the guidance surface from level l+1 leads to a descriptor (bottom left) that is not well comparable with either of the candidate descriptors (bottom centre / right).*

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 — this causes no harm to our method, as all comparisons are evaluated on each level separately and consequently any potential shrinkage is cancelled out. 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 roughly uniformly sampled point sets.

To address this drawback, 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 our simple approach proved to be sufficient.

5.2. Multi-Level Inpainting

Based on the point set hierarchy $\mathcal{P}^0, \ldots, \mathcal{P}^L$, we formulate the inpainting as a bottom-up process, filling the hole in the coarsest scale representation $\mathcal{P}^L$ first and then consecutively on the finer levels up to the finest level $\mathcal{P}^0$. In each step (aside from the first step, where $\mathcal{P}^L$ 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* $\mathcal{G}^l$ be any implicit representation of the (completed) point set $\mathcal{P}^{l+1}$. In our approach, we use the zero set of the MLS-approximation of $\mathcal{P}^{l+1}$, 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 $\mathcal{G}^l$ (see fig. 4, 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 $\mathcal{G}^l$ 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 l nor the coarser level l+1. This fact is illustrated in fig. 4.

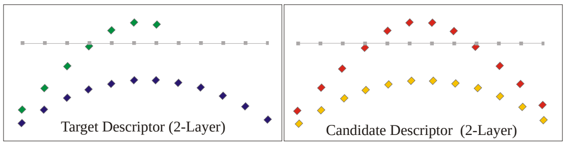


Figure 5: *The 2-Layer descriptor for the situation in fig. 4*

5.3. 2-Layer Descriptor

As a consequence, we define 2-layer descriptors as illustrated in fig. 5:

- The bottom layer χ_{l+1}, is constructed as described in sec. 4.1, with the exception that height values are derived from the zero level set of the MLS-approximation of $\mathcal{P}^{l+1}$.
- The top layer χ_l is constructed using the same parameter plane and the same sampling pattern, but capturing the available local geometry from $\mathcal{P}^l$ only, and assigning zero confidence to the invalid descriptor components.

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

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

While the parameter τ is arbitrary in principle, a value of 0.3 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.

As stated above, the coarsest level L is filled without guidance (formally setting $\tau = 0$), as there is no previous reconstruction available for evaluation as guidance surface. Given that the coarsest level's scale size corresponds well to the scale of the hole, this problem is sufficiently well-posed. However, for very large holes, considerable filtering might be necessary to this end. In this case a natural and trivial extension to our algorithm is to use any one of the available smooth hole filling schemes for the coarsest level and use the result as guidance surface $\mathcal{G}^L$.

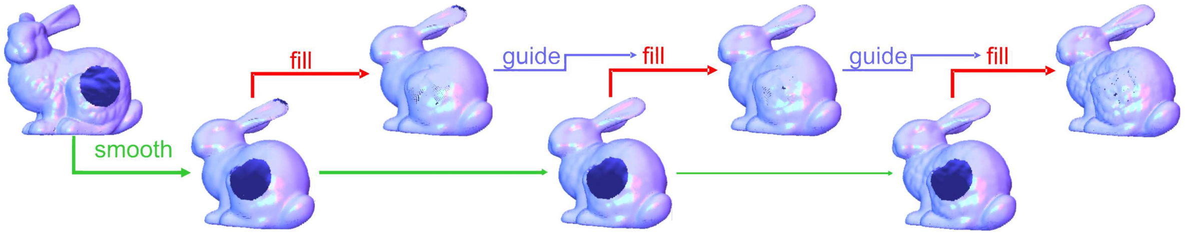


Figure 6: *Hierarchical reconstruction of the Stanford Bunny. First, a point set hierarchy (l = 0 to 3) of the defective bunny is constructed (bottom). Starting with $l = 3$, each level is filled per-se, where in level l, the level $l+1$ serves as Guidance Surface.*

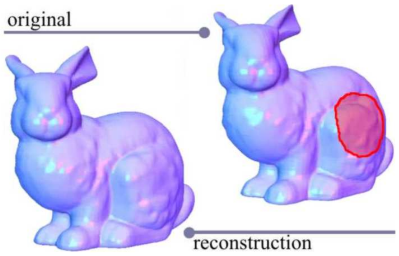


Figure 7: *Reconstruction result.*

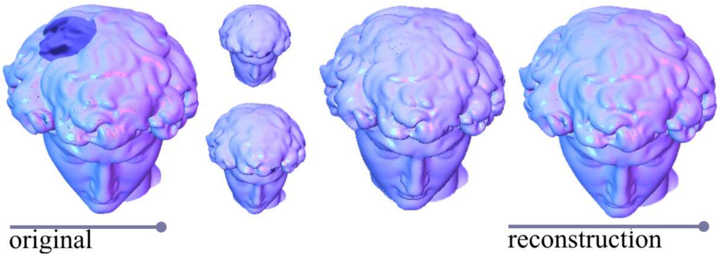


Figure 8: *Hierarchical reconstruction of the David Head Model.*

6. 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.

The figure on the title page illustrates the basic workflow of our algorithm. For target fragments (illustrated as green discs) an optimal candidate fragment (red discs) 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.

Figs. 6 and 7 demonstrate 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.

Also, the david head model from the Michelangelo project (fig. 8) could not have been filled using a 1-level approach, as the model itself does not contain appropriate candidates that correspond to the (unknown) hole regions' full spectrum of scales. By filling the hole for coarser regions, representing the basic geometry, first, and adding more and more detail with decreasing neighbourhood size at later stages, our algorithm was able to inpaint this hole in a visual believable way, taking into account the objects global and local context.

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 complete point set itself as guidance surface and candidate set. The combination of both, perfect guidance and perfect candidate set, resulted in the perfect reconstruction of the bunny (fig. 9). As opposed to that, figs. 6 and 7 show the hierarchical reconstruction of the incomplete bunny without any additional knowledge.

7. 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, each living in their specific scale with a

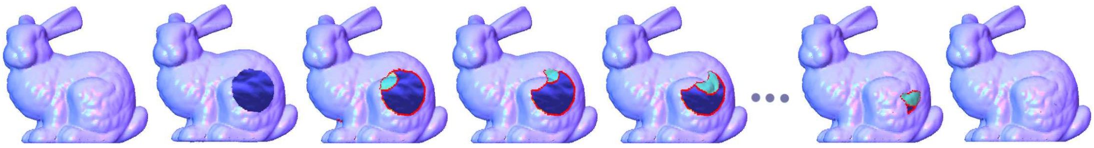


Figure 9: *From left to right: Original Bunny data set, data set with artificially introduced hole, iterative repair steps, resulting reconstruction. In this example, the candidate set and the guidance surface have been built from the original bunny data set.*

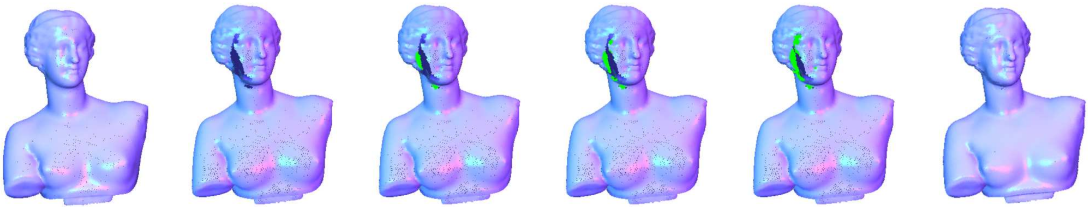


Figure 10: *Original Aphrodite data set (left), an incomplete scanning session and its iterative completion.*

naturally defined fragment size that is well correlated to the respective scale of the filling operations. In addition to that, the fragments are defined in local frames, thereby making our algorithm insensitive to similarity transformations as rotation, translation and scaling.

Being based on a scale-space representation of the object, our hierarchical 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.

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.

[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.

[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.

[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.

[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.

[DCOY03] Drori I., Cohen-Or D., Yeshurun H.: Fragment-based image completion. *ACM Trans. Graph. 22*, 3 (2003), 303–312.

[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.

[GSS99] Guskov I., Sweldens W., Schröder P.: Multiresolution signal processing for meshes. *Computer Graphics Proceedings (SIGGRAPH)* (1999), 325–334.

[HDD*92] Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W.: Surface reconstruction from unorganized points. In *Proc. of the 19th annual conf. on Computer graphics and interactive techniques* (1992), ACM Press, pp. 71–78.

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

[KEHKL*99] Kenneth E. Hoff I., Keyser J., Lin M., Manocha D., Culver T.: Fast computation of generalized voronoi diagrams using graphics hardware. In *SIGGRAPH '99: Proc. of the 26th annual conf. on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press, pp. 277–286.

[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.

[KS05] Kraevoy V., Sheffer A.: Template based mesh completion. In *Symposium on Geometry Processing* (2005), pp. 13–22.

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

[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.

[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.

[PMG*05] Pauly M., Mitra N. J., Giesen J., Gross M., Guibas L.: Example-based 3d scan completion. In *Symp. on Geometry Processing* (2005), pp. 23–32.

[SACO04] Sharf A., Alexa M., Cohen-Or D.: Context-based surface completion. *ACM Trans. Graph. 23*, 3 (2004), 878–887.

[Sch05] Schnabel R.: Detecting holes in surfaces. Central European Seminar on Computer Graphics (CESCG05) (TO APPEAR), May 2005.

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

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

[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).

[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.

[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.