Context-based Coherent Surface Completion

Gur Harary and Ayellet Tal
Technion
and
Eitan Grinspun
Columbia University

We introduce an algorithm to synthesize missing geometry for a given triangle mesh that has "holes." Similarly to previous work, the algorithm is context-based in that it fills the hole by synthesizing geometry that is similar to the remainder of the input mesh. Our algorithm goes further to impose a coherence objective. A synthesis is coherent if every local neighborhood of the filled hole is similar to some local neighborhood of the input mesh. This requirement avoids the undesired features, which can happen in context-based completion. We demonstrate the algorithm's ability to fill holes that were difficult or impossible to fill in a compelling manner by earlier approaches.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Curve, surface, solid, and object representations*

General Terms: Algorithms

Additional Key Words and Phrases: Surface completion, hole filling

ACM Reference Format:

Harary, G., Tal, A., and Grinspun, E. YYYY. Context-based Coherent Surface Completion. ACM Trans. Graph. VV, N, Article XXX (Month YYYY), 11 pages.
DOI: http://dx.doi.org/10.1145/XXXXXXX.YYYYYYY

1. INTRODUCTION

Physical artifacts often contain *holes* of missing geometry. Such holes are not the result of scanning, but rather come from the source objects themselves, which might be broken or incomplete. It is desirable to fill the holes in a manner that produces a compelling rendition of the complete geometry. We focus on the completion of holes with disc-like topology, which is a longstanding problem, and remains open especially for large holes. Such holes are particularly common in archaeology, where the artifacts are often found broken and missing large pieces.

Completing missing data is an inherently ill-posed problem. What makes for a good completion in one setting might not be in another. Examining the problem broadly, Ju [2009] considers geometric and topological correctness; but even such "correct" completions might not be compelling for some applications (see Fig. 1). For sufficiently small holes, a completion that is *as smooth as possible* can be effective [Verdera et al. 2003; Liepa 2003], but this approach does not generalize to larger holes covering prominent geometric features.

Researchers have considered filling larger holes by copying patches from other areas in the same object [Sharf et al. 2004; Bendels et al. 2005] or from a corpus of similar objects [Kraevoy and Sheffer 2005; Pauly et al. 2005; Gal et al. 2007]. We are inspired to build on such "context-based" approaches. They allow us to harness the knowledge that has been gained about example-based image synthesis, in which images or portions of images are synthesized from given exemplars [Kwatra et al. 2005]. In our setting, the input geometry is considered to be the exemplar, and the hole is considered to be the synthesis target.

A limitation of existing context-based approaches is that they may synthesize geometric features that are very different from those found on the original geometry. We consider a new objective that avoids this pitfall: *coherence* of the filled hole to the exemplar. A completion is *coherent* if every local neighborhood of the completion is similar to some local neighborhood of the exemplar. As seen in Fig. 1, the consideration of this objective results in compelling completion.

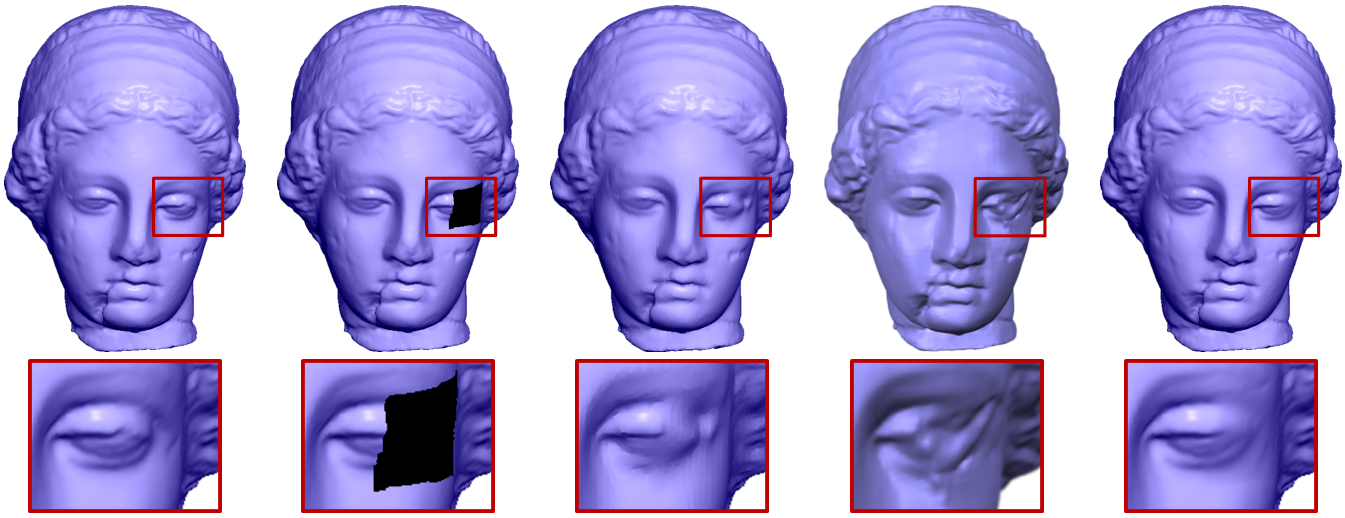
Contributions: We present a coherent surface completion method with state-of-the-art results for large holes. Our method is fully automatic and does not require parameter tuning. We present an evaluation of the approach on a number of challenging cases. Moreover, we demonstrate that our technique can be utilized for holes that are not necessarily disk-like, such as ragged holes from scanned data. The coherence of our completion is achieved through three key ideas. First, we formulate geometric hole completion as a global error minimization process (§3). Second, we propose a neighborhood similarity metric to establish exemplar-target correspondences, using both global and local information (§4-5). Third, we suggest a surface modification technique that minimizes the global error, while maintaining a proper triangulation of the surface (§6).

2. RELATED WORK

Ju [2009] offers a comprehensive survey of hole filling. Techniques may be classified as volume- or surface-based:

This research was supported in part by the Israel Science Foundation (ISF) 1420/12, the Argentinian Research Fund, Microsoft, the Jacobs-Qualcomm Foundation, the Ollendorff Minerva Center for Vision and Image Sciences, and the Viterbi Foundation. Authors' addresses: G. Harary (corresponding author), Technion; email: gur@tx.technion.ac.il; A. Tal, Technion; E. Grinspun, Columbia University.

DOI: http://dx.doi.org/10.1145/XXXXXXX.YYYYYYY



(a) original object (b) hole (c) detail-preserving (d) context-based (e) our completion

Fig. 1. **Completing a hole on Hygeia.** (e) Our hole completion better captures the structure of the eye compared to (c) the detail-preserving completion of [Davis et al. 2002] and (d) the context-based completion of [Sharf et al. 2004]. Our algorithm synthesizes the geometry by coherently drawing from the remainder of the model as an exemplar.

Volume-based methods convert the input mesh into a signed distance function over a volumetric grid. The function is then propagated using diffusion [Davis et al. 2002; Verdera et al. 2003; Guo et al. 2006], patching while enforcing minimum area [Ju 2004], or morphological operations [Bischoff et al. 2005]. The completed mesh is then extracted from the zero-level set of the distance function. The main benefit of this approach is its robustness in resolving geometric errors, i.e., the produced mesh is a manifold without self-intersections. The major drawback is the loss of geometric detail, even in areas far from the hole.

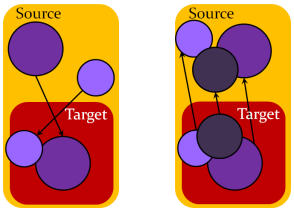
Surface-based methods operate directly on the given mesh, typically using one of three approaches: (i) triangulating the polygon delimiting the hole, then remeshing and smoothing to improve quality [Liepa 2003; Clarenz et al. 2004; Pernot et al. 2006; Zhao et al. 2007]; (ii) employing scattered data fitting techniques, such as Moving Least Squares [Wang and Oliveira 2007] or Radial Basis Functions [Branch et al. 2006; Chen and Cheng 2008] to smoothly fill the hole; (iii) context-based synthesis using the input mesh [Sharf et al. 2004; Bendels et al. 2005] or a corpus of exemplar meshes [Kraevoy and Sheffer 2005; Pauly et al. 2005; Gal et al. 2007]. Context-based synthesis may be followed by texture synthesis [Breckon and Fisher 2005; Nguyen et al. 2005; Xiao et al. 2007]. Additional information, such as global symmetry [Pauly et al. 2008; Xu et al. 2009] or an image [Xu et al. 2006; Park et al. 2006] can also assist the completion process. The benefits of context-based approaches are that they can recover lost geometric features, to the extent that the exemplars serve as good priors. In addition, regions far from the hole are not affected. The drawbacks of these approaches are the lack of robustness in the case of noisy data and the difficulty in preventing geometric intersections. Our method belongs to the family of context-based approaches. Differently from previous methods, ours also enforces coherence.

Coherence is a defining aspect of our approach, and here we describe its application in other contexts. The importance of coherence for image synthesis was established by Efros et al. [1999]. This concept was recently used for image synthesis [Kwatra et al. 2005], video completion [Wexler et al. 2007], video summarization [Simakov et al. 2008], geometric texture synthesis [Zhou et al. 2006], fluids texturing [Kwatra et al. 2007], and solid texture synthesis [Kopf et al. 2007].

In the context of 3D completion, Pauly et al. [2005] propose a method that retrieves suitable context models from a database and consistently blends the warped models to obtain the final consolidated 3D shape. The entire hole is expected to match a single region of the deformed mesh. Moreover, it is assumed that there is some valid information also inside the hole (i.e., "floating" points), which is a reasonable assumption for some kinds of scanned data. We do not make these assumptions and therefore are able to address cases where the hole is devoid of any data, as well as cases where information is available on the interior. Moreover, we handle cases when no single region exists that can be copied to complete the hole.

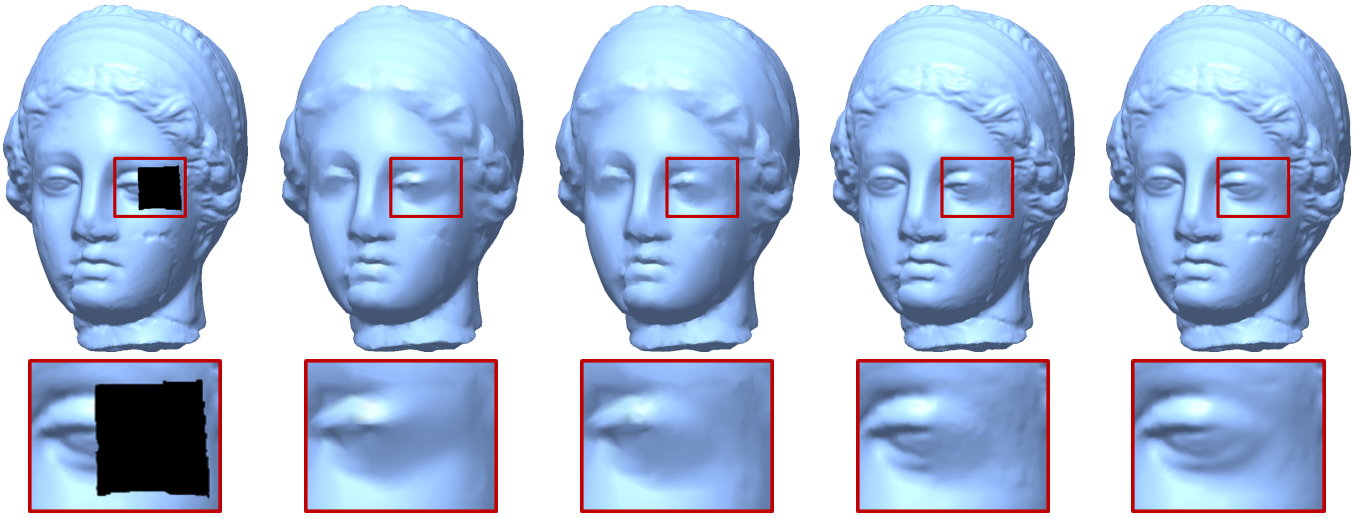
3. PROBLEM STATEMENT & ALGORITHM OUTLINE

Let S denote the *source*, the input surface, and T denote the target, the unknown surface that completes the hole. We define a patch around vertex $\mathbf{v}$ to be the connected neighborhood that falls within a ball of radius R centered at $\mathbf{v}$. The completion is context-based if T is completed using the geometry of S. We say that T is coherent with S if every patch of T is contained in S (Fig. 2).



(a) context-based (b) coherent

Fig. 2. **Context-based coherent completion.** (a) The completion is context-based if it uses the geometry of the input surface S. (b) The target T is *coherent* with the source S if every patch of T is contained in S.



(a) model with hole (b) coarse: smooth (c) coherent completion (d) fine: projected (e) result

Fig. 3. **Algorithm outline.** Given an object with a hole (a), it is simplified and the hole is smoothly completed on the simplified model (b). This smooth completion is modified to get a coherent context-based completion (c). Next, the smooth completion of the fine object is projected onto the modified simplified object (d) and modified to get the final result (e).

Let T_i and S_j be patches in T and S respectively, and $\mathcal{D}(T_i, S_j)$ be a scalar-valued *dissimilarity metric* between the two patches. We measure the *coherence error* between T and S as

$$\varepsilon(T,S) = \frac{1}{N_T} \sum_{T_i \subset T} \min_{S_j \subset S} \mathcal{D}(T_i, S_j)\,, \tag{1}$$

where N_T is the number of patches in T.

Our goal is to complete a given hole by minimizing the coherence error. Algorithm 1 performs this task (Fig. 3). It begins with some initial completion. We show that several initial completions are possible and that our algorithm is robust to the selected initialization (§8). Next, for each target patch several candidate source patches are found using a multi-scale signature (§4). From there, the target surface is iteratively refined to minimize the coherence error. The refinement procedure iterates over two steps: First, for each target patch the most similar source patch among the candidates is found (§5). Second, the locations of the target vertices are adjusted to reduce the coherence error, while avoiding degeneration of the mesh (§6). The iterative refinement is accelerated by a coarse-to-fine approach, where larger neighborhoods are first used to reduce low frequency error, and finer neighborhoods then reduce high frequency error (§7). Our implementation employs two scales.

Algorithm 1 Our context-based coherent completion algorithm

```
Initial completion of the hole
For varied object resolutions (from coarse to fine)
  For each target patch
    Find candidate similar source patches (§4)
  While the current error ε(S,T) decreases:
    For each target patch
      Find the most similar patch among the candidates (§5)
    Update the target according to the patches found in L7 (§6)
  Project the target of the fine object to the coarse object (§7)
```

4. CANDIDATE SELECTION OF SIMILAR PATCHES

Our goal is to replace each initial target patch with a better one. Accuracy in choosing the "right" patch is necessary for the completion to be compelling. Most works that address similarity of patches (as opposed to points) describe them using local information. Heider et al. [2011] present a survey on local shape descriptors. Examples of such information are the distances and the directions from corners and centers of embedding volumetric cells [Sharf et al. 2004], the extremum and the average of the curvatures [Park et al. 2006], and geodesic fans [Zelinka and Garland 2004; Bendels et al. 2005]. However, in our case, the initial completion is not necessarily accurate and therefore, the local information cannot be trusted. Instead, we wish to use a multi-scale information.

Moreover, in addition to seeking for a good descriptor, we do not simply select the patch whose descriptor is the most similar. Instead, we view the similar patches as "hints" regarding the locations where the most similar patch may be found. Candidate selection is performed once for each object resolution, by calculating a descriptor for each patch and choosing the patches with the similar descriptors as candidates.

Patch descriptor: Our descriptor extends the *Heat Kernel Signature* (HKS) [Sun et al. 2009; Vaxman et al. 2010] from points to patches. The HKS is an intrinsic point signature that describes the diffusion of heat across the surface over time. The heat kernel $h_t(x,y)$ records the heat transferred from point x to point y at time t, given a unit heat source at x at time 0. It is restricted to the temporal domain by considering only $h_t(x,x)$. Hence, the heat value over different times provide an effective multi-scale signature for shape matching.

To represent patches, we propose to consider statistical information on the HKS values in the patch. Our descriptor consists of two parts. The first is the average of the HKS values of the patch's vertices. Averaging is performed for each time t of the HKS, resulting in a vector of averages (HKS_μ). The second part is the variance of

these signatures, resulting in a vector of variances (HKS_{σ^2}). Given a patch P_i, its descriptor is

$$\text{HKS}(P_i) = \{\text{HKS}_{\mu}(P_i)_{[0,1]}, \text{HKS}_{\sigma^2}(P_i)_{[0,1]}\},$$

where the notation $[0,1]$ indicates that each part is normalized separately to the range $[0,1]$, by setting the L_∞-norm to 1.

We calculate the HKS, as well as our patch descriptor, once. This can be done since it scarcely changes throughout the iterative process of our algorithm (§5-6). For instance, the maximum percentage of change in the marked yellow vertices in Fig. 4 is 0.33% in vertex 1, 0.02% in vertex 2, and 0.0079% in vertex 3.

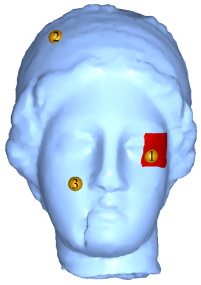


Fig. 4. **HKS can be precomputed once,** as it hardly changes throughout the iterative process of our algorithm.

The question is which time domain should be used. The larger the time domain, the wider the range that the two patches must match. Since this range may vary for different objects, we consider three time domains, which we empirically found to be beneficial for normalized objects. The first is the entire time domain, the second is the lower $3/4$, and the third is the lower half. Thus, for each patch we have three descriptors, each compared only to descriptors having the same time domain.

In our implementation, the entire time domain is calculated as suggested by Sun et al. [2009]. That is, for each model, we compute 300 eigenvalues (λ_i) and eigenvectors for the Laplace-Beltrami operator. Then, we uniformly sample 100 points in the time interval $[4ln(10)/\lambda_{300}, 4ln(10)/\lambda_2]$.

Dissimilarity measure: Given a target patch $T_i \subset T$, for each of its three descriptors we search for the k most similar source patches $S_j \subset S$, $j = 1, ..., k$ (we use $k = 0.1\%$ of total source vertices). We define the dissimilarity measure between T_i and S_j as

$$||\text{HKS}_{\mu}(T_i) - \text{HKS}_{\mu}(S_j)||^2 + ||\text{HKS}_{\sigma^2}(T_i) - \text{HKS}_{\sigma^2}(S_j)||^2. \quad (2)$$

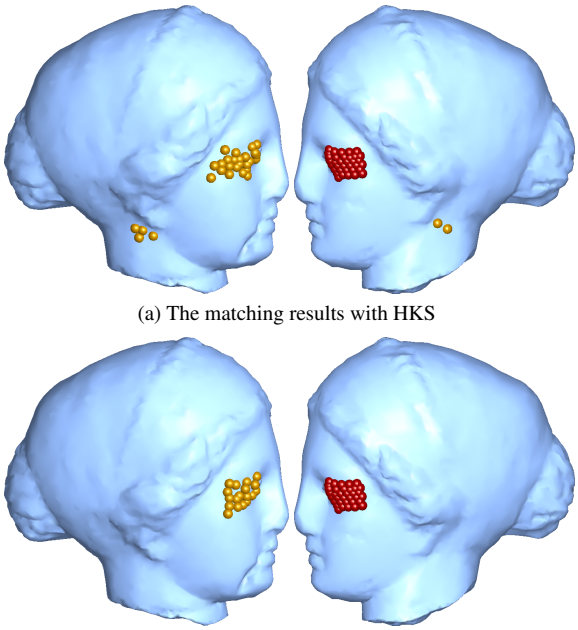
Evaluation: Fig. 5 shows centers of target patches in red and their most similar source centers in yellow. It can be seen that our patch descriptor significantly reduces the number of erroneous correspondences compared to the original HKS [Sun et al. 2009]. In these illustrations the entire time domain was used for both descriptors.

5. PATCH SIMILARITY

Our goal is to replace the initial completion with one that is both context-based and coherent. To do so, we would like to minimize the coherence error (1). Using the above procedure (§4), each target patch T_i is matched to a limited set of source patches $\mathcal{S}(T_i) \subset S$. Thus, (1) can be modified to be

$$\varepsilon(T, S) = \frac{1}{N_T} \sum_{T_i \subset T} \min_{S_j \subset \mathcal{S}(T_i)} \mathcal{D}(T_i, S_j)\,. \quad (3)$$

We minimize (3) by iterating over two steps. First, as our algorithm is context-based, we find for each target patch its most similar source patch among the candidates. Second, we modify the entire target to make it coherent. This section focuses on patch similarity and the next on the modification.



(a) The matching results with HKS

(b) The matching results with our patch descriptor

Fig. 5. **Our descriptor outperforms HKS.** The centers of the target patches are marked in red and the centers of the corresponding source patches are marked in yellow. The target is smooth due to our initialization and does not include the detailed geometry (e.g. valleys) of the eye. Therefore, the HKS errs when looking for matches. Our patch descriptor, on the other hand, finds the expected matches.

While (2) gave us patches that are similar globally, we now choose among them a patch that is similar locally. Given the set of candidate matching patches for T_i, $\mathcal{S}(T_i) \subset S$, we would like to choose the most similar patch, by using a different similarity measure from the one used in (2). This is done by a rigid alignment combining translation, rotation, and reflection. To find the rigid transformation, we use the HKS, already calculated, to find for each vertex in T_i its matching vertex in $S_j \subset \mathcal{S}(T_i)$. Using these matching pairs we calculate the rigid transformation between T_i and S_j by solving a least-squares minimization problem. Let $\mathbf{v}_k^{T_i}$ be a vertex of T_i, $\mathbf{v}_k^{S_j}$ be its matching vertex in S_j, n_i be the number of target vertices in T_i, and $\mathbf{t}$ be the translation vector from the center vertex of S_j to the center vertex of T_i. We seek an orthogonal matrix R (expressing rotation and reflection) that minimizes

$$\mathcal{D}(T_i, S_j) = \sum_{k=1}^{n_i} ||(\mathrm{R}\,\mathbf{v}_k^{S_j} + \mathbf{t}) - \mathbf{v}_k^{T_i}||^2. \quad (4)$$

We solve for R using the *shape matching* approach of Müller et al. [2005]. This process can be viewed as an *ICP*-like method [Rusinkiewicz and Levoy 2001]. However, unlike ICP, the matching vertices are found using HKS. This allows us to match a smooth target patch to a source patch with details. At the end, for each target patch, the minimizer of $\mathcal{D}(T_i, S_j)$ among all source patches is selected.

To illustrate the importance of combining global (§4) and local (§5) similarities, consider Fig. 6-7. Fig. 6 illustrates that examining several similar patches for each target patch (§4) and choosing the best patch among them (§5) is better than using only the single most similar patch. In Fig. 7, consider the depicted hole. Using only

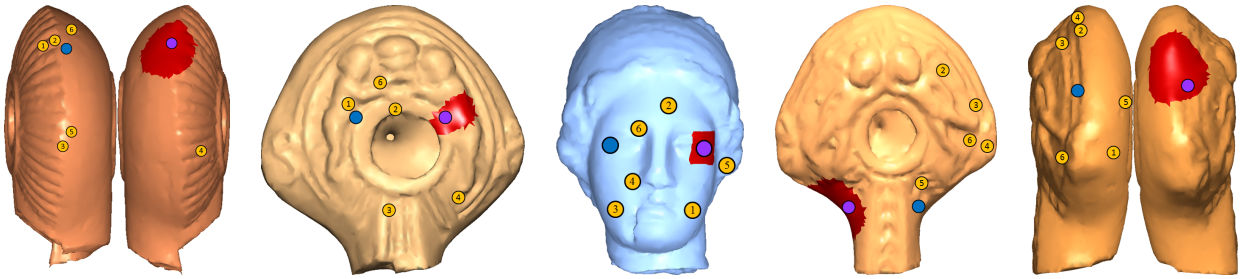
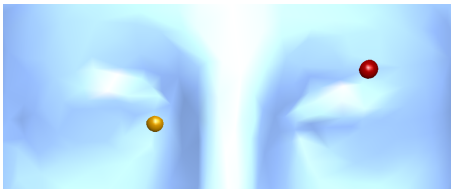


Fig. 7. **Our patch similarity outperforms state-of-the-art results.** Given the violet patch in the smooth initialized target, the yellow patches are found using the (1) maximal curvature of the center vertex, (2) mean of the maximal curvature, (3) shape index [Koenderink and van Doorn 1992], (4) spin image [Johnson and Hebert 1999], (5) HKS [Sun et al. 2009] at small t's of the center vertex, and (6) mean of the HKS values at small t's. Since our descriptor considers global information and our patch similarity considers local information, it allows us to find the blue patch, which is the patch we seek.



(a) considering only the most similar patch according to (2)

(b) considering several similar patches (§4) and finding the best match among them using (4)

Fig. 6. **The benefit of using two distinct matching procedures.** The center of a target patch is marked in red and the center of its most similar source patch in yellow. Evidently, the matching is improved when considering several similar patches and using two distinct matching procedures.

local information for the smooth target patch *(violet dot)*, we may find many similar patches at various locations *(yellow dots)*. Using instead a global-local metric, we find a single corresponding patch *(blue dot)*. This is so, as many patches are similar to the initial smooth target patch, but only a few match its global surroundings.

6. COHERENCE-ERROR MINIMIZATION

The aim is to move the target vertices, so as to decrease the coherence error of (3). This is done by minimizing the dissimilarity between the target and the source patches $\mathcal{D}(T_i, S_j)$. Let us first explore the pitfalls we encountered with a preliminary naïve approach.

Naïve approach: Let $\mathbf{v} \in T_i$ be a vertex in the target region, $T^{\mathbf{v}}$ be the group of M target patches that contain $\mathbf{v}$, and $S^{\mathbf{v}}$ be the group of their corresponding most similar source patches (as per §5). For each patch $S_m \subset S^{\mathbf{v}}$, we identify an *anchor point* $\mathbf{u}_m$ that best fits $\mathbf{v}$ when S_m is aligned to $T_m \subset T^{\mathbf{v}}$. (i.e., the closest point on the aligned source patch). Note that $\mathbf{u}_m$ is some point lying on the surface S_m, but not necessarily a vertex point. Let $\tilde{\mathbf{u}}_1, ..., \tilde{\mathbf{u}}_M$ be the positions of the anchor points after applying the rigid transformation of (4) that aligns the source to target patches.

The contribution of $\mathbf{v}$ to the coherence error is derived from its contribution to (4) for all $T_m \subset T^{\mathbf{v}}$:

$$\varepsilon_{\mathbf{v}} = \frac{1}{N_T} \sum_{m=1}^{M} (\mathbf{v} - \tilde{\mathbf{u}}_m)^2.$$

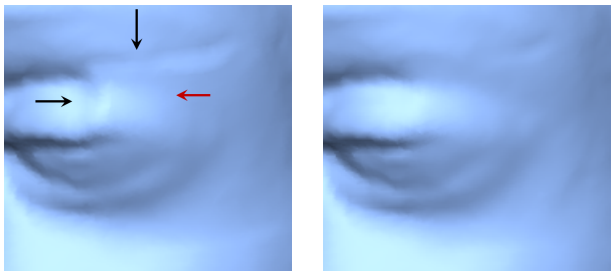
Minimizing this quadratic in $\mathbf{v}$ yields the locally ideal position

$$\mathbf{v}_d = \frac{1}{M} \sum_{m=1}^{M} \tilde{\mathbf{u}}_m. \tag{5}$$

Pitfalls with the naïve solution: While $\mathbf{v}_d$ indeed reduces the coherence error contribution of $\mathbf{v}$ more than any other position, we found that it produces a solution that is overly smooth: $\mathbf{v}_d$ is the average of all $\tilde{\mathbf{u}}_m$, and in effect we have attracted the target vertex toward an averaged (smoothed) version of the source data. This effect is especially noticeable on the surface's features, such as ridges and valleys.

Moreover, this solution does not take into account the contribution to the coherence error of source vertices that lie in the target patches, i.e., those in the neighborhood of the target's boundary. Therefore, the solution might not fit its surroundings, which will cause irregularities along the boundary of the target. These two effects are demonstrated in Fig. 8.

A third problem that arises is that if each target vertex moves independently, geometric intersections might occur.



(a) naïve solution (b) our solution

Fig. 8. **Problems of the naïve solution.** The eye is smoother than expected (red arrow) and contains irregularities on the boundary (black arrows).

Solution to the smoothness problem: Our initial completion is smooth and therefore lacks features. To get a compelling completion, we would like the target to have feature lines that are consistent with those of its surroundings. To do so, we utilize a vector field that conforms with the features of the object. This field is used to rate the fit between the target patch and its corresponding source patch, which will be employed as weights in (5).

The normalized vector field we use is the *feature-line field* proposed by Leifman et al. [2012], which is strongly affected by the feature lines, in particular ridges and valleys. Briefly, this field is set to be perpendicular to the ridges and valleys, pointing outward at the valleys and inward at the ridges (Fig. 9). To compute the field, first the faces through which the ridges and valleys pass are found [Ohtake et al. 2004]. Next, all the vertices that belong to these faces are marked. If such a vertex shares a face with exactly two marked vertices, it is assigned a line field direction. The field is interpolated smoothly to the rest of the object, by utilizing the Laplacian as the smoothness measure. This means that the field near the ridges and the valleys of the source continues smoothly into the target, as if there were features at the target.

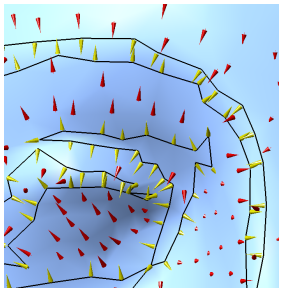


Fig. 9. **The feature line field near the eye of Hygeia.** The field is perpendicular to the ridges and valleys (yellow) and interpolated smoothly to the rest of the object (red).

It should be noted that though the use of the Laplacian operator is widespread in the literature [Fisher et al. 2007; Xu et al. 2009; Kolomenkin et al. 2011] and gives us good results, other operators are also possible, e.g. a biharmonic equation, or an anisotropic method.

Once the field is computed, we use it to set a weight indicating the desirable impact of $\tilde{\mathbf{u}}_m$ on $\mathbf{v}_d$ from (5). Let $\mathbf{f}_k^{T_m}$ be the field's vector at a vertex of T_m and $\mathbf{f}_k^{S_m}$ be the field at its closest vertex in the transformed source patch S_m. The weight w_{mk} of the dissimilarity between the vectors is set such that a ridge or valley vertex gets a high value, since we have more confidence in the direction of the field in these vertices. In practice, the weight of this dissimilarity is set to the number of vertices in T_m (n_m), whereas the weights of the other vertices are set to 1. The dissimilarity between the two patches is defined as

$$\mathfrak{d}_m = diss(T_m, S_m) = \frac{1}{\bar{w}_m} \sum_{k=1}^{n_m} w_{mk} ||\mathrm{R}\, \mathbf{f}_k^{S_m} - \mathbf{f}_k^{T_m}||^2,$$

where $\bar{w}_m = \sum_{k=1}^{n_m} w_{mk}$.

This dissimilarity is used as a weight for the aligned anchor points. We replace the averaging in (5) with the weighted average

$$\mathbf{v}_d = \frac{1}{\bar{\mathfrak{d}}} \sum_{m=1}^{M} \mathfrak{d}_m^{-5} \tilde{\mathbf{u}}_m, \qquad \text{where} \qquad \bar{\mathfrak{d}} = \sum_{m=1}^{M} \mathfrak{d}_m. \tag{6}$$

Handling the irregularities of the boundary: We utilize the contribution to the coherence error of the source vertices in the neighborhood of the target's boundary to reduce the irregularities. For these vertices, we know both the desirable position, calculated in (6) and the correct position, which is the input position (as source vertices do not move). The difference between these positions defines an offset vector.

We consider these error vectors as an initial *offset field*. This vector field is then interpolated smoothly into the target by utilizing the Laplacian, similarly to the feature field. Assuming the offset vector on a target vertex $\mathbf{v}$ is $\delta\mathbf{v}$, the sought-after position for $\mathbf{v}$ becomes

$$\mathbf{v}_d = \frac{1}{\bar{\mathfrak{d}}} \sum_{m=1}^{M} \mathfrak{d}_m^{-5} \tilde{\mathbf{u}}_m + \delta\mathbf{v}. \tag{7}$$

Solution to geometric intersections: To avoid self-intersections, we perform the update in (7) gradually, in several steps. Let $\tilde{\mathbf{v}}_{d_q}$, $q = 1, ..., Q$, be the position of $\mathbf{v}$ after the qth update step, where $\tilde{\mathbf{v}}_0 = \mathbf{v}$. Our update rule (where $Q = 10$ in our implementation) is

$$\tilde{\mathbf{v}}_{d_q} = \tilde{\mathbf{v}}_{d_{q-1}} + \frac{1}{Q}\Big(\underbrace{\frac{1}{\bar{\mathfrak{d}}} \sum_{m=1}^{M} \mathfrak{d}_m^{-5} \tilde{\mathbf{u}}_m + \delta\mathbf{v}}_{\mathbf{v}_d} - \tilde{\mathbf{v}}_{d_{q-1}}\Big).$$

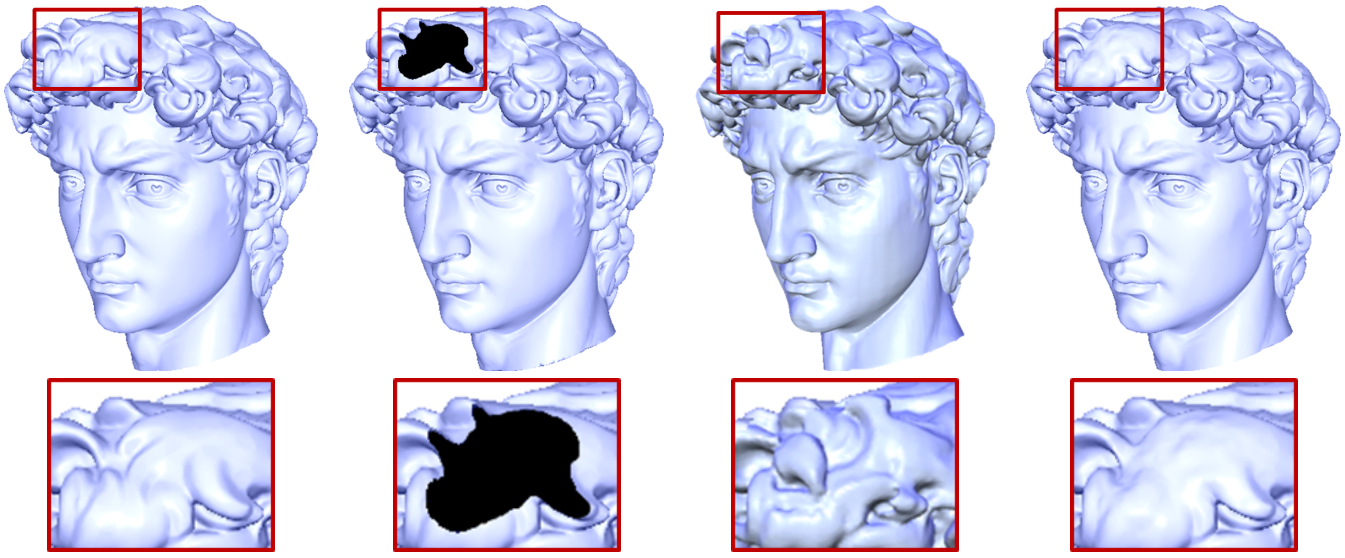
Each such update step is followed by a *mesh repair* step, which removes long and skinny triangles and very large or very small triangles. Long or small triangles might cause self-intersections when adjacent vertices move. Large triangles prevent our algorithm from representing small geometric details. Mesh repair performs edge flips, edge collapses, triangle collapses, or edge splits (in the case of large triangles). In our implementation, an edge is flipped if the sum of its two counter angles is larger than π; an edge is split if it is longer than 1.5 times the averaged edge length; a triangle is collapsed if its area is smaller than 0.01 of the average triangle area; and an edge is collapsed if its adjacent angle is smaller than 10°.

7. BI-LEVEL COMPLETION

In a patch-based algorithm, such as ours, a fundamental consideration is the size of the patches. Large patches are essential for capturing the low frequencies of the object, whereas small patches are important for capturing the high frequencies. Moreover, since all the computations in §5–6 depend on the size of the patch, larger patches result in a slower computation. In order to avoid long running times and still be able to get completions that correctly capture the low frequencies of the object, we use a bi-level algorithm (line 2 in Algorithm 1).

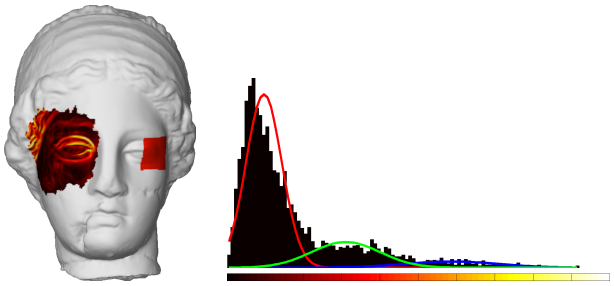
In particular, we first simplify the object to $\frac{1}{50}$ of its size [Garland and Heckbert 1997]. The target on the simplified object is then modified, as explained in §5-6, using patches whose size is 1% of the object's surface area. Next, the results for the coarse object are mapped to the fine object to get an initialization. Finally, the fine object is refined by reapplying the algorithm (§5-6) to smaller patches. The patch size is determined according to the size of the features on the input object, as explained below.

We would like the radius of a patch to be slightly larger than that of a feature, e.g., a valley and its immediate surroundings. This allows us to capture these features. To find this size, we roughly classify the object's vertices into three categories: background, prominent features, and sharp features or noise. To model this classification and to find the three sizes of features, we use a *Gaussian mixture model* (GMM). We first calculate the maximal curvature of all the vertices at the source patches that were found to match any target patch. Then, we approximate the histogram of the curvatures with a GMM with three Gaussians $f = \sum_{i=1}^{3} \alpha_i G(\mu_i, \sigma_i)$, where μ_i and σ_i are the mean and the standard deviation of the ith component and, α_i is its weight, where $\sum_{i=1}^{3} \alpha_i = 1$ (see



(a) original (b) hole (c) [Sharf et al. 2004] (d) our completion

Fig. 11. **Completing a hole on David's hair.** Both our completion and Sharf et al. [2004]'s look natural, however, our completion is more similar to the ground truth due to our coherence requirement.



(a) maximal curvature (b) histogram of maximal curvature

Fig. 10. **Patch size calculation.** (a) The absolute value of the maximal curvature is calculated on the relevant source patches. (b) Three Gaussians are fit to the histogram of these values. The green Gaussian represents the features of the eye.

Fig. 10). The parameters α_i, μ_i, and σ_i are automatically estimated using the Expectation Minimization (EM) method [Dempster et al. 1977]. The Gaussian with the smallest mean represents the background, the one with the medium mean represents the prominent features of the object, and the one with the highest mean represents the sharp features and noise. We set the radius of a patch to be slightly larger than that of a prominent feature by setting it to $R = \left(\frac{1}{|e_{avg}|\cdot\mu_2} + 1\right)\cdot|e_{avg}|$. To understand this formula recall that $\frac{1}{\mu_2}$ is the radius of curvature of the prominent feature, so $\frac{1}{|e_{avg}|\cdot\mu_2}$ indicates the number of times the average edge ($|e_{avg}|$) appears in the feature.

8. RESULTS

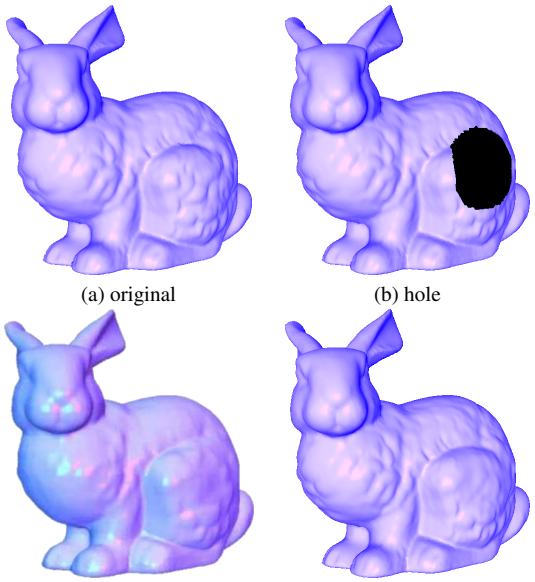
Qualitative evaluation: Figs. 1,11-16 show several examples where appealing completions of large holes are produced by our algorithm. In particular, Fig. 1 is mentioned by Sharf et al. [2004] as a limitation of their context-based algorithm. We show the results side-by-side to demonstrate that our algorithm indeed handles this challenging case of a broken eye. Fig. 1 also compares our result to that of Davis et al. [2002]. We note that this detail-preserving method manages to capture the overall structure of the missing region, but not the fine details. In contrast, our method reconstructs the details, such as the structure of the eye.

Fig. 12. **Our completion integrates the information from several regions** (in magenta) to complete the curl (in dark blue).

Fig. 11 provides another comparison, completing a curl of David's hair. Both completions look pretty natural, nevertheless, our completion is closer to the original curl. This example is difficult since the missing curl does not appear elsewhere in the object. The strength of our method is in its ability to integrate information from several areas, as illustrated in Fig. 12.

Fig. 13 compares our result to that of Bendels et al.[2005]'s. It can be seen that our result is not as smooth and therefore, better captures both the structure of the bunny's leg and its 3D texture.

Figs. 14-16 demonstrate the applicability of our algorithm to archaeological artifacts. Archaeology has recently attracted a lot of attention in computer graphics, and graphics research has much to



(a) original (b) hole

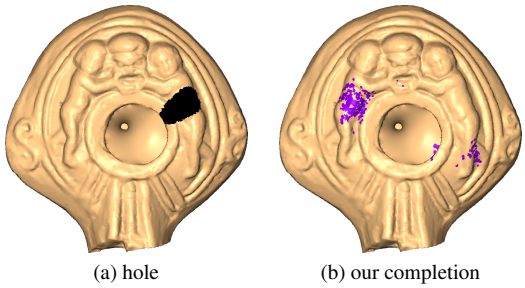
(c) [Bendels et al. 2005] (d) our completion

Fig. 13. **Our completion manages to reconstruct the structure of the leg,** as well as its 3D texture.

offer. In this domain, the objects are commonly found broken and archaeologists are interested in visualizing the shape of an object before it was broken. Since the models in this domain are noisy due to erosion and other processes the objects underwent after remaining underground for thousands of years, smooth completions are inadequate. Also, large pieces of the objects cannot be copied to complete the holes, since no identical or symmetric parts exist. Nevertheless, our algorithm manages to complete the holes reasonably well.

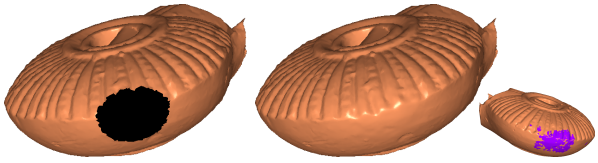
Fig. 17 presents several results, where different sizes of holes are considered. Our method manages to capture the structure of the column even when the holes are large, however, as expected, the quality of the completion slowly degrades as the hole grows.

Quantitative evaluation: To evaluate our results not only qualitatively, but also quantitatively, we calculate the MSE and the Hausdorff distances between the original (unbroken) object and its com-



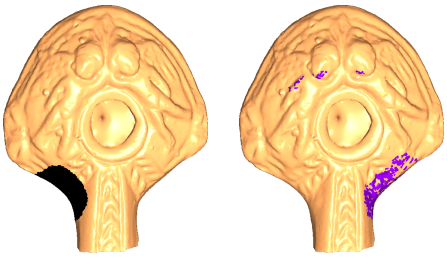
(a) hole (b) our completion

Fig. 14. **Completing a manually broken Hellenistic oil lamp.** Notice that the inner circle is restored even though its width is smaller in its counterpart. The cupid's belly is also restored in a compelling manner, even though the two cupids differ. Our completion uses the regions marked in purple.



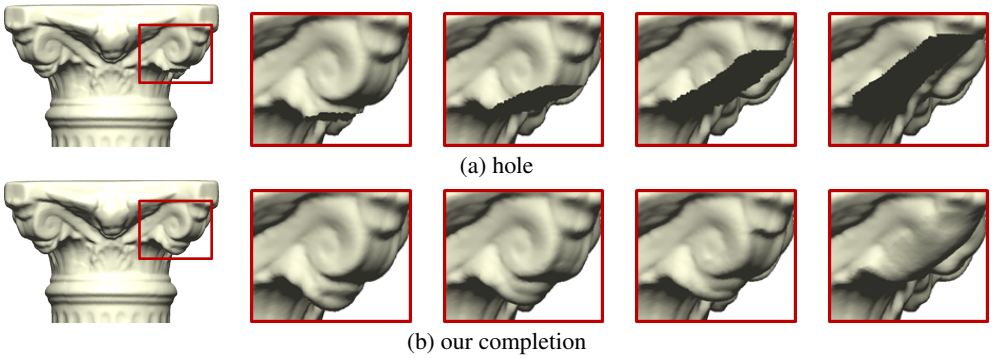
(a) hole (b) our completion

Fig. 15. **Our algorithm manages to capture the structure of the relief,** although the width and the shape of each strip is different. The structure of the strips is reconstructed only where it should, whereas the rest of the hole is reconstructed smoothly. Our completion uses the regions marked in purple.



(a) hole (b) our completion

Fig. 16. **Our result successfully captures the 3D texture of the lamp.**



(a) hole

(b) our completion

Fig. 17. **Our completion results look compelling even as the size of the hole grows.** This is demonstrated on a column model. The rightmost example shows a limitation of our algorithm: When the neighborhood of the hole does not contain prominent features, the algorithm does not have enough information to produce these features.

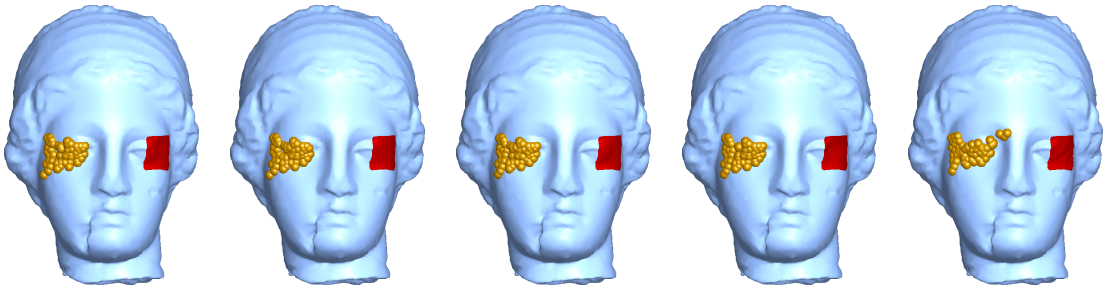
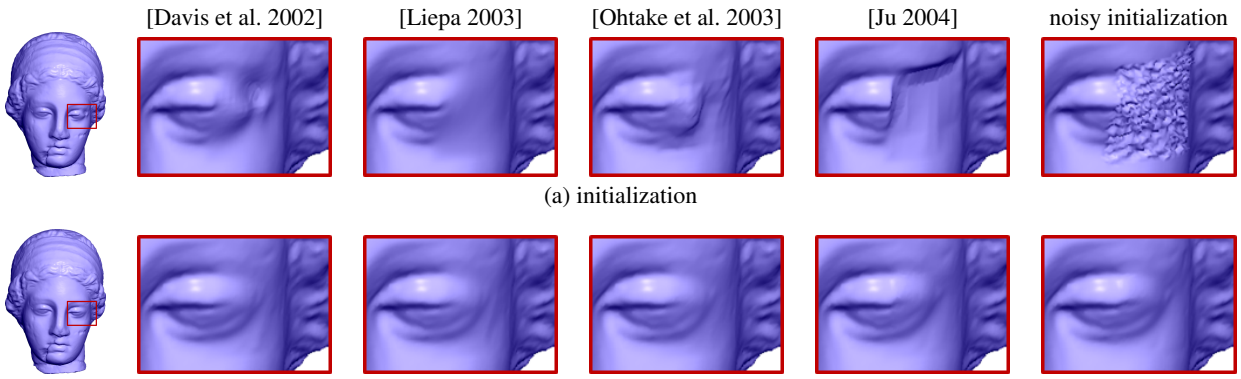


Fig. 18. **Our descriptor is robust w.r.t. the initial completion.** The centers of the source patches, which correspond to the target patches (target in red), are marked in yellow. Their location hardly changes when different initializations are used. Since the methods of [Davis et al. 2002; Ohtake et al. 2003; Ju 2004] change the entire object, for fair comparison we projected our initial target to the result of these algorithms.



(a) initialization

(b) our completion for each of the initialization methods

Fig. 19. **Our algorithm is robust w.r.t. the initial completion.** Even when the initialization is very noisy our algorithm handles it well.

Table I. **Our completion outperforms others in terms of distances to the original object.**

Figure	Vollfill	Polymender	Our result
1	3.1	530.6	**0.05**
	16.4	562.4	16.9
11	8.84	8.81	**0.04**
	302.4	301.5	**16.1**
13	8.8	244.3	**8.2**
	67.3	309.3	**25.0**
14	38.3	5.0	**0.08**
	362.2	59.3	**16.9**
15	37.6	7.6	**0.26**
	547.7	105.3	**20.6**
16	28.2	4.8	**0.16**
	315.5	23.6	**20.3**
17 (top)	5.2	2.3	**0.007**
	134.7	40.7	**7.6**

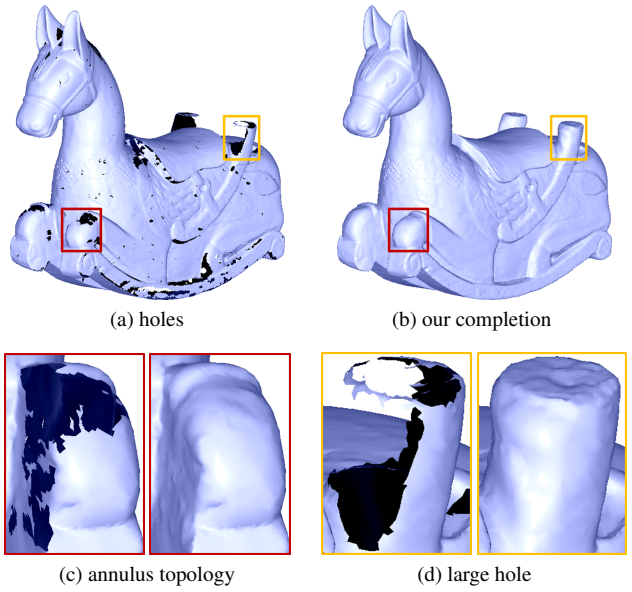
This table compares the mean (top) and Hausdorff (bottom) distances for the objects shown in this paper. Our results are compared to Vollfill [Davis et al. 2002] and to Polymender [Ju 2004]. The darker text indicates the best result. All the numbers are multiplied by 10^{-3}.

pletion. Table I compares our results to those of the two methods whose codes are available [Davis et al. 2002; Ju 2004]. Our results are consistently better, except for one case (Hausdorff on Igea). But even in this case, visual inspection (Fig. 1) indicates that our completion is more natural.

Robustness to initialization: In all the results shown in this paper we initialized our completion using the advancing front method of Zhao et al. [2007]. We replaced the smoothing step in this algorithm with the least-squares smoothing operator proposed by Sorkine and Cohen-Or [2004], since it only requires a linear solver. Fig. 18-19 show that our descriptor as well as our final result are robust w.r.t. the initial completion. Thus, other algorithms can also be used for initialization.

Hole topology: Our algorithm was designed to complete large holes having a disk-like topology, which despite years of research, is still challenging. However, our algorithm can also handle other types of topologies. Figs. 20-22 demonstrate the ability of our algorithm to handle objects with both many small holes and holes with annulus topology. The only modifications needed to our algorithm is in the initialization phase. In the front-advancing technique that we use [Zhao et al. 2007], we allow fronts to merge.

Running times: The algorithm was implemented in C/C++ and ran on a 2.67Ghz Intel i7-processor laptop with 4Gb of memory. The running time, which depends on the size of the object, the size of the patch, and the size of the hole, is of several minutes. For instance, on the Hellenistic oil lamp from Fig. 14, which consists of 63000 vertices, where the radius of a fine patch is automatically set by the algorithm to $R = 4 \cdot |e_{avg}|$, and the hole is filled with 682 vertices, the algorithm runs for 4 minutes. On Hygeia (Fig. 1), which consists of 134000 vertices, where the radius of a fine patch is set by the algorithm to $R = 7 \cdot |e_{avg}|$, and the hole is filled with



(a) holes (b) our completion (c) annulus topology (d) large hole

Fig. 20. **Completing many holes with various topologies.** Holes of different sizes and topologies are completed on a scanned model of a horse. The second row zooms-in into a completion of an annulus hole on one of the horse's legs (left) and on a large hole on the saddle (right).

1346 vertices, the algorithm runs for 15 minutes. Most of the time is spent on the patch similarity step (§5) of the fine object.

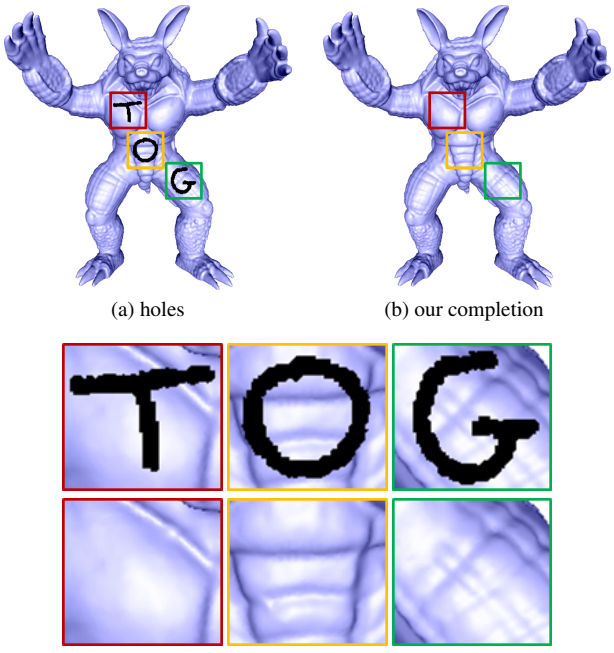
Limitations: Our algorithm has a few limitations. First, when the hole is large and there are not enough features on its surrounding area, the completion might be smoother than we wish. This is so, since a large hole might result in an initial completion that is too far from the desired shape of the missing region and thus, has a descriptor that does not characterize the original object well. Moreover, having no features on the surroundings of the hole might cause our patch similarity to prefer the smooth patches among the erroneous selected candidates. This is demonstrated at the rightmost example at Fig. 17. Second, as with most earlier algorithms, ours does not guarantee the avoidance of self-intersection. However, in practice, our completions are intersection-free in all the cases that we checked.

9. CONCLUSION

This paper presents a novel algorithm for surface completion. We propose a new objective—coherence. We formulate the coherence requirement as a global error minimization process and propose a novel modification technique that minimizes this error. It utilizes our new patch similarity metric, which takes into account both local and global shape attributes.

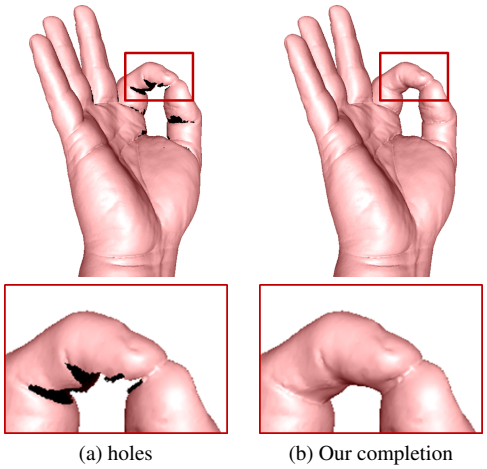
The quality of our completion is demonstrated on challenging cases. These include both general objects and archaeological objects. The latter is especially interesting, not only because of its importance, but also because archaeological objects are usually eroded and noisy and hence challenging. Furthermore, these object are handmade and thus, though they look symmetric at a first glance, in fact they are inherently asymmetric and no sub-surface exists that can be used as is to complete the hole.

In the future, we would like to explore different methods for allowing patch similarity with scaling differences.



(a) holes (b) our completion (c) zoom-in

Fig. 21. **Completing narrow holes with various topologies across different surface texture.** Our algorithm manages to nicely complete the 3D texture.



(a) holes (b) Our completion

Fig. 22. **Completing many holes of different sizes.** The zoomed-in region is completed in a reasonable manner.

ACKNOWLEDGMENTS

The models are courtesy of the Digital Michelangelo Project at Stanford University (Head of David), the Stanford 3D Scanning Repository (Bunny, Armadillo), Cyberware (Hygeia), the MIT CSAIL database (Column), and INRIA by the AIM@SHAPE Shape Repository (Horse, Hand). We also thank Dr. A. Gilboa and the Zinman Institute of Archaeology at the University of Haifa for providing the archaeological models.

REFERENCES

G.H. Bendels, R. Schnabel, and R. Klein. 2005. Detail-preserving surface

inpainting. In *VAST*. 41–48.

S. Bischoff, D. Pavic, and L. Kobbelt. 2005. Automatic restoration of polygon models. *ACM Transactions on Graphics* 24, 4 (2005), 1332–1352.

J. Branch, F. Prieto, and P. Boulanger. 2006. A hole-filling algorithm for triangular meshes using local radial basis function. In *3DPVT*. 727–734.

T.P. Breckon and R.B. Fisher. 2005. Non-parametric 3D surface completion. In *3DIM*. 573–580.

C.Y. Chen and K.Y. Cheng. 2008. A sharpness-dependent filter for recovering sharp features in repaired 3D mesh models. *IEEE Transactions on Visualization and Computer Graphics* 14, 1 (2008), 200–212.

U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. 2004. A finite element method for surface restoration with smooth boundary conditions. *Computer Aided Geometric Design* 21, 5 (2004), 427–445.

J. Davis, S.R. Marschner, M. Garr, and M. Levoy. 2002. Filling holes in complex surfaces using volumetric diffusion. In *3DPVT*. 428–441.

A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B* 39, 1 (1977), 1–38.

A.A. Efros and T.K. Leung. 1999. Texture synthesis by non-parametric sampling. In *ICCV*, Vol. 2. 1033–1038.

M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe. 2007. Design of tangent vector fields. 26, 3 (2007), 56: 1–9.

R. Gal, A. Shamir, T. Hassner, M. Pauly, and D. Cohen-Or. 2007. Surface reconstruction using local shape priors. In *SGP*. 253–262.

M. Garland and P.S. Heckbert. 1997. Surface simplification using quadric error metrics. *SIGGRAPH* (1997), 209–216.

T.Q. Guo, J.J. Li, J.G. Weng, and Y.T. Zhuang. 2006. Filling holes in complex surfaces using oriented voxel diffusion. In *ICMLC*. 4370–4375.

P. Heider, A. Pierre-Pierre, R. Li, and C. Grimm. 2011. Local Shape Descriptors, a Survey and Evaluation. In *3DOR*. 49–56.

A.E. Johnson and M. Hebert. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 5 (1999), 433–449.

T. Ju. 2004. Robust repair of polygonal models. *ACM Transactions on Graphics* 23, 3 (2004), 888–895.

T. Ju. 2009. Fixing geometric errors on polygonal models: a survey. *J. of Computer Science and Technology* 24, 1 (2009), 19–29.

J.J. Koenderink and A.J. van Doorn. 1992. Surface shape and curvature scales. *Image and Vision Computing* 10, 8 (1992), 557–564.

M. Kolomenkin, I. Shimshoni, and A. Tal. 2011. Prominent Field for Shape Processing and Analysis of Archaeological Artifacts. *International Journal of Computer Vision* 94, 1 (2011), 89–100.

J. Kopf, C.W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski, and T.T. Wong. 2007. Solid texture synthesis from 2D exemplars. *ACM Transactions on Graphics* 26, 3 (2007), 2: 1–9.

V. Kraevoy and A. Sheffer. 2005. Template-based mesh completion. In *SGP*. 13: 1–10.

V. Kwatra, D. Adalsteinsson, T. Kim, N. Kwatra, M. Carlson, and M.C. Lin. 2007. Texturing fluids. *Visualization and Computer Graphics, IEEE Transactions on* 13, 5 (2007), 939–952.

V. Kwatra, I. Essa, A. Bobick, and N. Kwatra. 2005. Texture optimization for example-based synthesis. *ACM Transactions on Graphics* 24, 3 (2005), 795–802.

G. Leifman and A. Tal. 2012. Mesh Colorization. *Computer Graphics Forum* 31, 2 (2012).

P. Liepa. 2003. Filling holes in meshes. In *SGP*. 200–205.

M. Müller, B. Heidelberger, M. Teschner, and M. Gross. 2005. Meshless deformations based on shape matching. *ACM Transactions on Graphics* 24, 3 (2005), 471–478.

M.X. Nguyen, X. Yuan, and B. Chen. 2005. Geometry completion and detail generation by texture synthesis. *The Visual Computer* 21, 8–10 (2005), 669–678.

Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.P. Seidel. 2003. Multi-level partition of unity implicits. *ACM Transactions on Graphics* 22, 3 (2003), 463–470.

Y. Ohtake, A. Belyaev, and H.P. Seidel. 2004. Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics* 23, 3 (2004), 609–612.

S. Park, X. Guo, H. Shin, and H. Qin. 2006. Surface completion for shape and appearance. *The Visual Computer* 22, 3 (2006), 168–180.

M. Pauly, N.J. Mitra, J. Giesen, M. Gross, and L.J. Guibas. 2005. Example-based 3D scan completion. In *SGP*. 23: 1–10.

M. Pauly, N.J. Mitra, J. Wallner, H. Pottmann, and L.J. Guibas. 2008. Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics* 27, 3 (2008), 43: 1–12.

J.P. Pernot, G. Moraru, and P. Véron. 2006. Filling holes in meshes using a mechanical model to simulate the curvature variation minimization. *Computers & Graphics* 30, 6 (2006), 892–902.

S. Rusinkiewicz and M. Levoy. 2001. Efficient variants of the ICP algorithm. In *3DIM*. 145–152.

A. Sharf, M. Alexa, and D. Cohen-Or. 2004. Context-based surface completion. *ACM Transactions on Graphics* 23, 3 (2004), 878–887.

D. Simakov, Y. Caspi, E. Shechtman, and M. Irani. 2008. Summarizing visual data using bidirectional similarity. In *CVPR*. 1–8.

O. Sorkine and D. Cohen-Or. 2004. Least-squares meshes. In *SMI*. 191–199.

J. Sun, M. Ovsjanikov, and L. Guibas. 2009. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. *Computer Graphics Forum* 28, 5 (2009), 1383–1392.

A. Vaxman, M. Ben-Chen, and C. Gotsman. 2010. A multi-resolution approach to heat kernels on discrete surfaces. *ACM Transactions on Graphics* 29, 4 (2010), 121: 1–10.

J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro. 2003. Inpainting surface holes. In *ICIP*, Vol. 2. 903–906.

J. Wang and M.M. Oliveira. 2007. Filling holes on locally smooth surfaces reconstructed from point clouds. *Image and Vision Computing* 25, 1 (2007), 103–113.

Y. Wexler, E. Shechtman, and M. Irani. 2007. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 3 (2007), 463–476.

C. Xiao, W. Zheng, Y. Miao, Y. Zhao, and Q. Peng. 2007. A unified method for appearance and geometry completion of point set surfaces. *The Visual Computer* 23, 6 (2007), 433–443.

K. Xu, D. Cohen-Or, T. Ju, L. Liu, H. Zhang, S. Zhou, and Y. Xiong. 2009. Feature-aligned shape texturing. 28, 5 (2009), 108: 1–8.

K. Xu, H. Zhang, A. Tagliasacchi, L. Liu, G. Li, M. Meng, and Y. Xiong. 2009. Partial intrinsic reflectional symmetry of 3D shapes. *ACM Transactions on Graphics* 28, 5 (2009), 138: 1–10.

S. Xu, A. Georghiades, H. Rushmeier, J. Dorsey, and L. McMillan. 2006. Image guided geometry inference. In *3DPVT*. 310–317.

S. Zelinka and M. Garland. 2004. Similarity-based surface modelling using geodesic fans. In *SGP*. 204–213.

W. Zhao, S. Gao, and H. Lin. 2007. A robust hole-filling algorithm for triangular mesh. *The Visual Computer* 23, 12 (2007), 987–997.

K. Zhou, X. Huang, X. Wang, Y. Tong, M. Desbrun, B. Guo, and H.Y. Shum. 2006. Mesh quilting for geometric texture synthesis. *ACM Transactions on Graphics* 25, 3 (2006), 690–697.

Received November 2012; accepted May 2013