Geometric inpainting of 3D structures

Pratyush Sahay, A.N. Rajagopalan
Indian Institute of Technology Madras
Chennai, India
pratyush.a.sahay@gmail.com, raju@ee.iitm.ac.in

Abstract

In this paper, we address the problem of inpainting in 3D digital models with large holes. The missing region inference problem is solved with a dictionary learning-based method that harnesses a geometric prior derived from a single self-similar structure and online depth databases. The underlying surface is recovered by adaptively propagating local 3D surface smoothness from around the boundary of the hole by appropriately harvesting the cue provided by the geometric prior. We showcase the relevance of our method in the archaeological domain which warrants 'filling-in' missing information in damaged heritage sites. The performance of our method is demonstrated on holes with different complexities and sizes on synthetic as well as real examples.

1. Introduction

With the introduction of real-time techniques for 3D model generation using low-cost sensors such as the Microsoft Kinect [10], creation of 3D world models has now been brought into the realm of public domain. It is also not uncommon to find laser scanners being employed for generating accurate point clouds. We are also witnessing growth of online services that allow for web-based 3D model generation from user provided images [23] or 3D point clouds [20]. In this work, our interest is in geometric inpainting i.e. filling-in of point clouds with missing data. The relevance of this problem stems from the need for inpainting 3D models of damaged structures in archaeological sites. With the advent of efforts such as the Digital Michelangelo project [14], there has been large-scale interest in heritage digitization in the vision, graphics, virtual reality and related research communities. The Google Art project [7] aims to provide a capability to perform a virtual walk-through, enabling internet-based access of rich 'common' heritage from across the world. While it is important to digitize a heritage site 'as is', building and showcasing 3D models of damaged archaeological structures can be visually unpleasant due to presence of large missing regions. The David restoration project [3] has successfully demonstrated the use of 3D models in the framework of cultural restoration. Such a restoration effort allows for the creation of visually pleasing 3D models with a two-fold purpose: i) heritage preservation, and ii) enabling a visually pleasing 'virtual tour' experience. Moreover, such a completed model can also serve as reference for any future attempt at physical restoration.

This paper deals with the scenario involving naturally existing large damaged regions in cultural artifacts. We do not consider missing information due to inaccurate scanning as it is a different research problem in its own right. Furthermore, several works already exist in the literature that deal exclusively with this scenario. The input data to the framework that we propose is a 3D mesh, generation of which constitutes a pre-processing step [6]. Because we deal with large missing regions, it is not possible to fill-in the missing data from neighboring regions of the damaged object itself. However, in the presence of an example that is self-similar to the damaged object, we propose a gradient map and dictionary learning-based method to harness the geometric prior. To assume the presence of a self-similar example (we need only one!) is not unreasonable for archaeological sites. The missing geometry is inferred by exploiting the constraint that the underlying 'missing' geometry in the hole shows locally smooth variations.

The problem that we address in this paper lends itself naturally to multi-sensor data fusion. The 3D point cloud could have been derived by processing a sequence of images (as in structure from motion (SFM)) or directly from range sensors (such as a laser scanner). Similar argument holds for online depth databases that we exploit for data fusion. The proposed dictionary learning (DL)-based geometric inpainting method works on point clouds irrespective of their regularity due to its local data access format. Since the camera poses are known (as we use an SFM-based 3D model generation pipeline), it is straight-forward to obtain the corresponding projection of the 3D model, and hence its depth map in each camera. The point cloud is projected in each of the cameras and only the region of the hole that best projects into a camera is filled up in that view. The best visibility of a 3D point in a camera is as defined in [27]. Thus, the workflow involves obtaining the corresponding depth map of a boundary patch on the surface of the 3D model, performing inpainting in the depth domain, and reprojecting the inpainted depth patch back onto the 3D surface. This resampling ensures that the algorithm can handle both regular and irregular point clouds, and at the same time helps to obtain a smoothly varying gradient map.

The main contribution of this work is a new methodology for geometric inpainting based on fusing information from a *single* self-similar example and a dictionary learnt from depth maps available online. Since our method works by propagating information from the boundary, we also perform accumulation error analysis and evaluate our scheme on many examples. Our work yields a framework that can be a valuable aid to heritage restoration and visualization applications by providing the ability to perform geometric reconstruction of large missing regions in the rendered 3D mesh models.

We begin by discussing related works in section 2. This is followed by a description of the proposed method in section 3. In section 4, we discuss experimental results and finally conclude with section 5.

2. Related Works

Suppose a 3D model $\mathcal{S} \in \mathbb{R}^3$ of a real world object is given with large missing or damaged regions $\mathcal{H}$. The goal is to faithfully estimate the underlying surface geometry in $\mathcal{H}$. A perceptually intuitive procedure woule be to provide for a natural progression of the surface topology existing in the neighbourhood of the boundary of $\mathcal{H}$, while maintaining local surface curvature and smoothness in different regions in the process [11]. Works such as [5] have attempted mesh completion for the case of small missing regions by using smoothness priors with respect to the local neighbourhood. They considered iterative extensions of the neighbourhood geometry into the hole using volumetric diffusion. However, such an approach fails to correctly inpaint large holes which tend to have a surface complexity unique to that object class. The scenario of filling holes in 3D surfaces arising from sensor imperfections, low surface reflectivity or self-occlusions has been discussed in [15], [29]. They advocate filling-up of small or medium sized surface deficiencies using local surface topology. Hole filling of smooth surfaces using tensor voting (TV) framework is discussed in [9]. TV-based inpainting of holes in depth maps using local geometry alone is addressed in [12]. Surface completion of 3D models for the restrictive case of repeating 'relief structures' can be found in [2]. 3D scan completion from examples is addressed in [21]. However, the requirements of a well-annotated and pre-segmented database, and manual marking of landmarks is quite cumbersome.

We address this difficult problem by making use of geometric prior harnessed from a *single* self-similar example in conjunction with depth maps available online by providing class-specific prior information about the complex surface variations within the hole-region. This can provide an effective prior for the surface complexity unique to that object class. Our approach is different from the tensor-voting (TV) based approach of [24] which requires *several* (typically > 5) self-similar examples. The non-availability of a large number of examples for such scenarios precludes the usage of big database methods such as principal components analysis (PCA). In the following section, we propose a hole-filling algorithm that judiciously utilizes a suitably derived geometric prior. We give details on the use of the proposed inpainting methodology for challenging scenarios such as damaged real-world structures.

3. Hole-filling with single self-similar example

We begin with a 3D model (point cloud) of the target structure to be inpainted. This could have been obtained using images within an SFM framework or could have been obtained from a range scanner. For real world structures 'in the wild', the availability of structures similar to the target object will typically be quite limited. We assume the availability of just *one* self-similar structure. This structure needs to be converted to its 3D model. Without the imposition of any constraints on the orientation of the acquisition device (optical or laser), large pose variations can exist between 3D point clouds of the damaged structure $\mathcal{S}$ and its self-similar example $\{\mathcal{M}\}$. In order to exploit geometric prior for the purpose of missing surface inference, registration of the example in $\{\mathcal{M}\}$ by means of a transform T, $T : \mathcal{M} \rightarrow \mathbb{R}^3$, is needed, wherein

$$T^* = \arg\min_T \|T(\mathcal{M}) - \mathcal{S}\|_2 \tag{1}$$

Since the acquisition environment is not a controlled setup, the 3D models may exhibit a large number of outliers too. However, the fact that $\{\mathcal{M}\}$ is self-similar to $\mathcal{S}$ works to the advantage of this registration step by providing a large ratio of similar undamaged regions compared to the damaged region and the outliers. Thus, a robust point cloud registration technique that efficiently handles outliers such as CPD (Coherent Point Drift) [18] is preferred to obtain the best possible transform T for the self-similar example. Further, we ensure that only the undamaged portions of the 3D model participate in the registration process.

Post registration, the aligned example $\{\mathcal{M}'\}$ can provide estimates for the missing region of $\mathcal{S}$ from the corresponding region. However, due to the unconstrained nature of the image capturing process, the scale of $\mathcal{S}$ and $\mathcal{M}$ can still vary significantly. A global point cloud registration may not resolve local scale changes, which in turn can lead to boundary artifacts in the inpainted result. Our goal is to propagate local region smoothness into the hole, while staying faithful to the prior provided by the single self-similar example.

It is well-understood that dictionary learning (DL) provides a robust local representation for a given signal class [1]. Since several depth databases are available online, it stands to reason that one can attempt a DL-based approach for hole-filling. Surface gradients are known to be resilient to the effects of relative scale differences while at the same time they can exhaustively capture higher-order curvatures that may be present on the surface. We propose to solve the problem of performing 3D geometry inpainting (while simultaneously avoiding boundary artifacts) within the framework of DL by harvesting gradients from the self-similar example $\mathcal{M}'$ to guide the choice of the sparse representation learnt from online depth databases to infer the missing region that best represents the geometric prior from $\mathcal{M}'$. The actual process of inpainting proceeds from *outside-in*. In order to preserve local surface curvature variations, a hole-filling strategy from the current hole boundary $\{h_k\}$ to the center of the hole is followed, wherein the oriented surface points on $\mathcal{S}$ in the neighbourhood of the hole boundary infer the missing surface points along the hole boundary $\{h_k\}$. These newly inferred surface points $\mathcal{S}(h_k)$ along with the original values aid in inferring missing surface points along a new hole boundary, and this process is repeated till the hole gets completely filled up.

Incidentally, a related problem in the texture domain is addressed in Poisson image editing [22]. This technique has been shown to successfully blend a given texture patch onto a possibly completely different background image. We also make use of the known gradient field from the registered self-similar example $\mathcal{M}'$, from regions corresponding to the damaged region $\mathcal{H}$ of the broken structure. However, in contrast to [22], we search for a set of sparse representations from an overcomplete dictionary D that best represents the known regions of the current patch at the hole boundary ($\mathcal{P}$). Of these, the sparse representation that has the most similar gradient to $\mathcal{M}'$ at the current boundary point h_k in the unknown region is used to provide an estimate for the missing region.

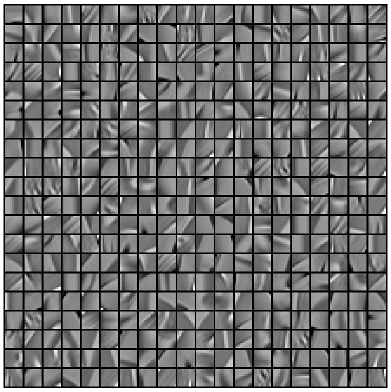


Figure 1. (a) Example of a $k = 400$ atom dictionary.

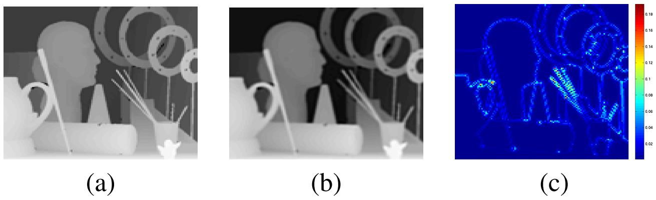


Figure 2. Error in reconstruction of depth images using the atoms of a depth dictionary. (a) Input image, (b) reconstructed image, and (c) error in reconstruction. The legend values vary between 0 and 0.2, with blue, green and red roughly corresponding to 0, 0.1 and 0.2, respectively.

3.1. Depth dictionary generation

With the ready availability of range scanners, several depth databases have been made available online in the last few years [13], [26] and [19]. We proceed to build a dictionary to facilitate sparse representation. Following k-SVD based DL [1], [16], we extract overlapping local patches $\{J_i\}$ of size $p \times p$ from a large number of randomly selected depth maps from online depth databases, and arrange them into a matrix Y such that $Y = [\text{vec}(J_1)\ \text{vec}(J_2)\ \cdots\ \text{vec}(J_v)]$, where $\text{vec}(J_i)$ represents an operation that lexicographically orders $J_i,\ i = 1, 2, ...v,\ Y \in \mathbb{R}^{m \times v},\ m << v$ and $m = p^2$. After normalization of the columns of Y, a k-atom dictionary ($k > m$) is learned. This problem is formulated as

$$(\alpha^*, D^*) = \arg\min_{\alpha, D} \parallel Y - D\alpha \parallel_F^2 + \gamma \parallel \alpha \parallel_1 \tag{2}$$

$$\text{s.t.}\ \parallel \mathbf{d}_i \parallel \leq 1,\ i = 1, 2, ...k, k < v$$

where $\{\mathbf{d}_i\}$ are the columns of the overcomplete dictionary D, $D \in \mathbb{R}^{m \times k}$ and α is the matrix encoding the sparsity for the dictionary-based representation. An example depth dictionary is shown in Fig. 1.

For our implementation, we set $p = 8, v = 10000$ and $k = 1024$. In order to establish the representative nature of the learned dictionary, we empirically show the effectiveness of dictionary-based representation in reconstructing some standard range images. The Middlebury stereo dataset [25] provides several depth maps and stereo pairs which have been widely used in the computer vision community. The sparse representation of some of these images is first found in the learned dictionary D, followed by reconstruction of the respective images from this overcomplete dictionary D. This is analogous to KSVD-based DL and image reconstruction using the learnt dictionary [28], but in the depth domain. The original image, the reconstructed image and the error in reconstruction are shown in Fig. 2 for a representative example. The low values of the

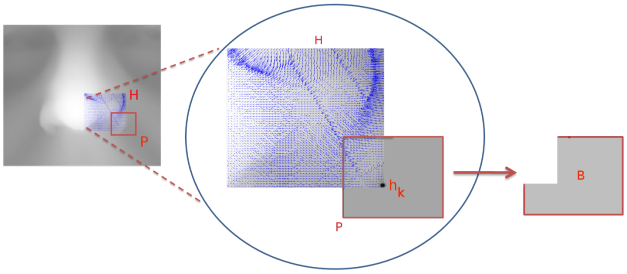


Figure 3. Gradients as cue for inpainting. The region filled with gradients (blue in colour) is a hole. The gradients from the self-similar example are superimposed over the hole. The region bounded in the red box labeled $\mathcal{P}$ is the current patch, while h_k is the current boundary pixel within $\mathcal{P}$. B is the mask corresponding to known regions of $\mathcal{P}$.

reconstruction error (as indicated by the legend in Fig. 2(c)) reveal that widely varying depth images admit a sparse yet effective representation in the overcomplete dictionary D. It is to be noted that none of these depth patches were used to learn the dictionary.

3.2. Data Fusion using Gradient

Consider the reconstruction of a patch $\mathcal{P}$ (see Fig. 3) such that it contains a large known region and an unknown body. The example shows the depth map of the region around the nose of a human face. Square region ($\mathcal{H}$) is marked as damaged. Suppose $\alpha_{\mathcal{P}}$ is the column of α corresponding to this patch from the learnt dictionary. Our objective is to find the set of sparse representations $\{\alpha_{\mathcal{P}}(\gamma)\}$ that best explains the known regions in patch $\mathcal{P}$. This can be done by varying γ over a set of values and obtaining the corresponding basis representations. Minimizing the error in representating the large known region of $\mathcal{P}$ will ensure local surface smoothness in the region around h_k. Though there may exist local scale changes in terms of the absolute depth values between the damaged 3D model $\mathcal{S}$ and its self-similar example $\mathcal{M}$, a gradient domain representation of the depths is largely unaffected by the relative scale of the 3D models. Also, gradients exhaustively capture local surface variations, thereby serving as a good cue to guide the inpainting process.

Generating the gradients in $\mathcal{H}$ by fusing gradient information from the corresponding region of the registered self-similar example $\mathcal{M}'$, we propose to use the sparse representation $\alpha_{\mathcal{P}}(\gamma)$ that has the most similar gradient at the point corresponding to h_k as the best representation for $\mathcal{P}$. This in turn provides an estimate for the missing surface h_k. This problem can be formulated in terms of the dot-product of the gradients as

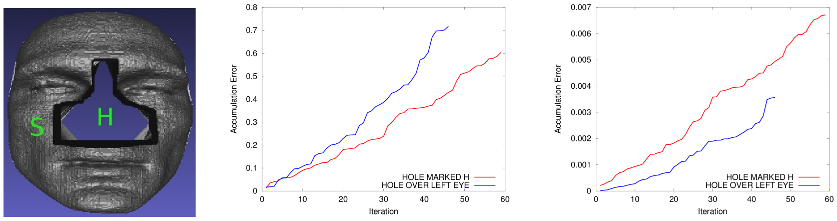
$$\begin{aligned}\alpha^*_{\mathcal{P}}(\gamma) = \arg\min_{\alpha_{\mathcal{P}}(\gamma)} \big(\| (D\alpha_{\mathcal{P}}(\gamma) - \mathcal{P}) \otimes B \|_2^2 + \\ \mu\left(1 - (\nabla_n(D\alpha_{\mathcal{P}}(\gamma)) |^T_{h_k} \nabla_n(\mathcal{P}) |_{h_k})\right)\big)\end{aligned} \quad (3)$$

where B is a mask corresponding to the known region of $\mathcal{P}$ (see Fig. 3), the notation $\otimes$ refers to pixel-wise multiplication operation, ∇_n calculates the normalized gradient of a function, and $(.)\ |_{h_k}$ indicates the value of function evaluated at the boundary h_k. The first term in Eqn. 3 attempts to reconstruct the known region in the patch $\mathcal{P}$ as close as possible to the original values, while the second term tries to minimize the angle between the normalized gradient vectors. This has the effect of generating a surface that is as similar in orientation as possible to the corresponding surface region in the self-similar example. For solving equation Eqn. 3, we use an iterative greedy algorithm: for the current patch under consideration, we obtain a set of sparse representations by varying λ. This is followed by iterating over this set and choosing the atom which minimizes Eqn. 3 as the best match for the current patch. The blue-arrows over the region in Fig. 3 indicate the harvested gradients from self-similar example $\mathcal{M}$. The zoomed-in region indicates the positioning of the patch ($\mathcal{P}$) at the current hole boundary h_k. It is to be noted that though gradients encode only the difference of depth values, taking an overlapping patch around the boundary and propagating the information into the hole-region ensures a geometrically inpainted result which is both smooth with respect to the boundary of the hole-region, and effectively follows the geometric prior from the self-similar example $\mathcal{M}$. Our methodology is described in Algorithm 1.

Algorithm 1 Fill-in large complex missing region $\mathcal{H}$ using a single self-similar example.

```
Require: (a) Set of overlapping patches $\{J_i\}$ from several
    range maps. (b) Self-similar example $\mathcal{M}$.
Ensure: $\mathcal{M}$ is registered with $\mathcal{S}$
$\mathcal{H} \leftarrow$ missing regions of $\mathcal{S}$
Obtain the overcomplete dictionary $D$ using Eqn. 2.
while $\mathcal{H} \neq \phi$ do
  $\{h_k\} \leftarrow$ boundary($\mathcal{H}$)
  for $k = 1$ to No. of elements in $\{h_k\}$ do
    Obtain $\mathcal{P}$ containing $h_k$ as shown in Fig. 3.
    Get the optimum $\alpha^*_{\mathcal{P}}(\gamma)$ using the cost function in
    Eqn. 3.
  end for
  Re-estimate the missing region $\mathcal{H}$.
end while
```

We wish to point out that such a sparse representation has been used in image inpainting [17] too. While they discuss an optimization technique to obtain the sparse representation 'online', the proposed work uses traditional DL and combines the atoms and gradients obtained from a self-similar example into a single formulation to achieve the intended objective.



(a) (b) (c)

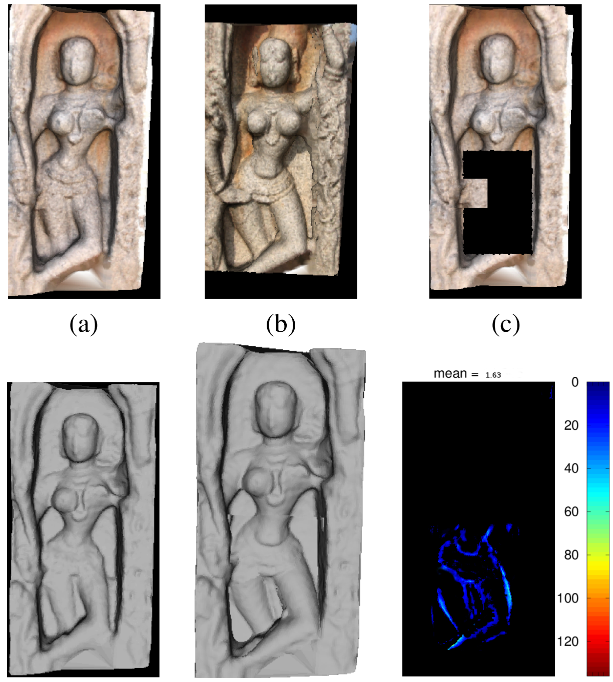
Figure 4. (a) Example of a damaged structure. $\mathcal{H}$ represents the damaged region while $\mathcal{S}$ represents the known geometry in the 3D model. (b,c) Accumulation error plots for our method and TV-based method, respectively. The hole regions considered correspond to $\mathcal{H}$ (shown in blue) and the other plot (shown in red) corresponds to a region chosen over the left-eye of the same face model.

4. Experimental Results

In this section, we demonstrate the effectiveness of our method on synthetic as well as real data examples. Complex and large holes that present significantly challenging situations for 3D inpainting are considered. We also provide comparisons with the recently proposed tensor-voting based method of [24] which uses multiple self-similar examples. We are grateful to the authors of [24] for sharing data as well as their code for the examples discussed in this section. We show that despite using just a single self-similar example, our results are quite comparable to (and sometimes even better than) that of [24].

In the first example, we consider the 3D model of a human face (Fig. 4(a)) with significant loss of information. We use the Texas 3D Face Recognition Database (Texas 3DFRD) [8] to obtain self-similar examples from which the underlying geometry to be filled-in can be inferred. While our method uses a single face example, the method of [24] is run with 4 face samples. We show in Fig. 4(b) the average accumulation error with iterations for our method, while Fig. 4(c) gives the error plot for [24]. The hole-filling proceeds from the hole boundary to the centre of the hole region. For both the plots, the line-plot in blue is for the hole region marked $\mathcal{H}$ in Fig. 4(a), while that in red is for hole-filling performed over a synthetically created hole over the left-eye region of the same face. From the plots, for the TV-based method, the accumulation error is of the order of 10^{-3}, while for the DL-based method it is of the order of 10^{-1}. While the TV-based method results in a finer reconstruction (as it uses multiple examples), the DL-based method too yields accumulation error that is low enough for practical applications.

Next consider the synthetic example of a statue of a human in a dancing pose in Fig. 5. A large hole region is created as shown in Fig. 5(c), and hole-filling is performed using a single self-similar example (Fig. 5(b)). The reconstructed 3D model (Fig. 5(e)) is visually quite close to the original 3D model (Fig. 5(d)). In addition, the low values of the reconstruction error (as indicated by the mean and legend in Fig. 5(f)) reflect effective hole-filling using our method.



(a) (b) (c)

(d) (e) (f)

Figure 5. 3D geometry inpainting of a synthetically created hole-region using a single self-similar example. (a,d) The original structure, (b) self-similar example used, (c) the synthetically created damaged structure, (e) reconstructed 3D structure, and (f) reconstruction error.

Archaeological sites provide for numerous examples of structures showing structural deformities or broken regions due to exposure to several natural and man-made forces of degradation over centuries and thus provide good examples over which the performance of an algorithm can be tested. We consider here a few interesting examples of 3D models of real world structures.

Consider the mythical “yali” structure (Fig. 6(a)), in which a large region of the 3D model is missing in comparison to its self-similar example (Fig. 6(b)). It is to be noted that the missing region was not bounded on all sides

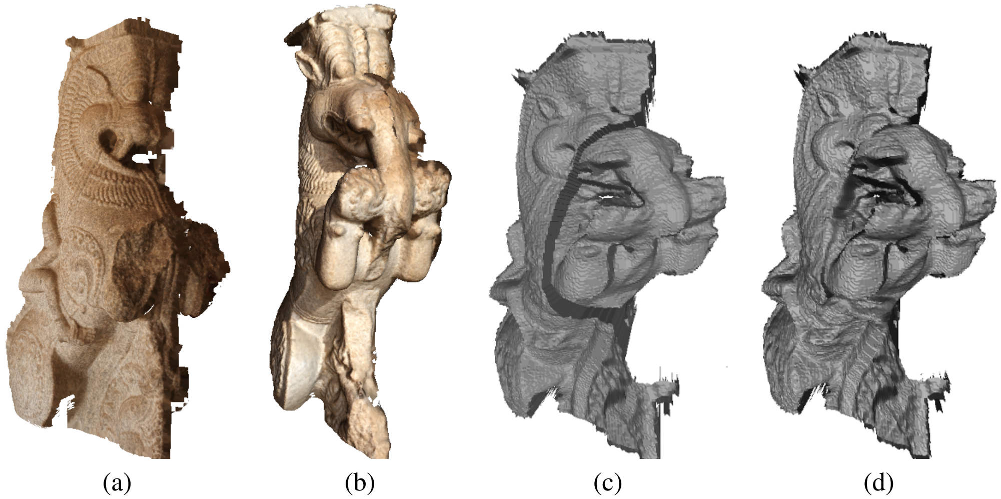


Figure 6. 3D geometry inpainting using a single self-similar example. (a) The damaged structure. (b) Self-similar example used. (c) Result using [24]. (d) Reconstruction using our method. Note that the result in (d) is devoid of boundary artifacts.

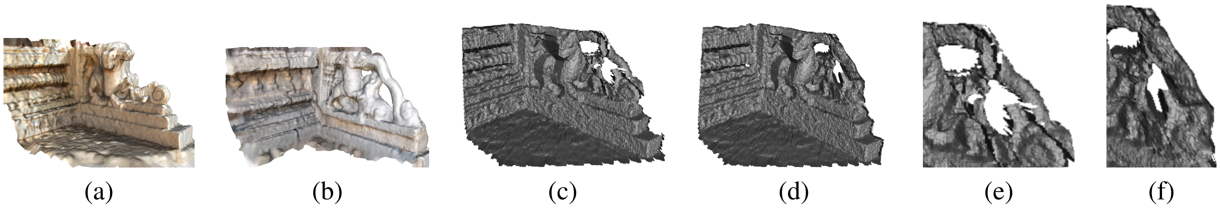


Figure 7. Another example for 3D geometry inpainting using a single self-similar example. (a) The damaged structure. (b) Self-similar example. (c) Result using [24]. (d) Reconstruction using the proposed method. Zoomed-in regions corresponding to (c) and (d) are shown in (e) and (f), respectively.

by known values. Hence, boundary interpolation methods such as Poisson image editing [22] will fail in this scenario. The result using the proposed method is shown in Fig. 6(d). The result using the method of [24] (Fig. 6(c)) suffers from boundary artifacts due to local scale mismatch. Since the DL method is based on gradients and local smoothness, it results in better surface inference.

The next real example considered is that of a damaged staircase of a temple structure, where the entire side-arm is missing (Fig. 7(a)). Since only one self-similar example (Fig. 7(b)) is available in this case, the TV method of [24] results in an inpainting with visible discontinuities at the boundary (Fig. 7(c)) with the corresponding zoomed-in reconstructed region shown in Fig. 7(e)). The DL-based method, in comparison, yields better inpainting results (Fig. 7(d)) with the corresponding zoomed-in reconstructed region displayed in Fig. 7(f)).

For the examples shown in Fig. 8, the results obtained using the proposed method follow the geometric prior quite faithfully and yield good inpainting results in the hole-region. In Fig. 8(d) the result of mesh completion using the hole-filling option found in Meshlab [4] is provided for comparison. Meshlab is an open-source tool that includes several 'state-of-the-art' mesh processing algorithms. The hole-filling algorithm provides a customizable interface using which the best hole-filling suitable to a given mesh can be obtained. It tries to connect the vertices at the boundary of the hole-region using non-self-intersecting patches. Note that such a vanilla fit often vastly deviates from the true underlying surface of the object class. Also, the hole should be bounded by known geometry all around in order to use this method.

5. Conclusions

We proposed a dictionary learning and surface gradients-based method for inpainting of 3D holes when only a single self-similar example (captured in an uncontrolled environment) is available as geometric prior. The results given in this paper indicate the extent of faithful reconstruction possible for even very challenging scenarios using the proposed framework, albeit with occasional minor local artifacts.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Sig. Proc.*, 54(11):4311–4322, Nov. 2006. 3

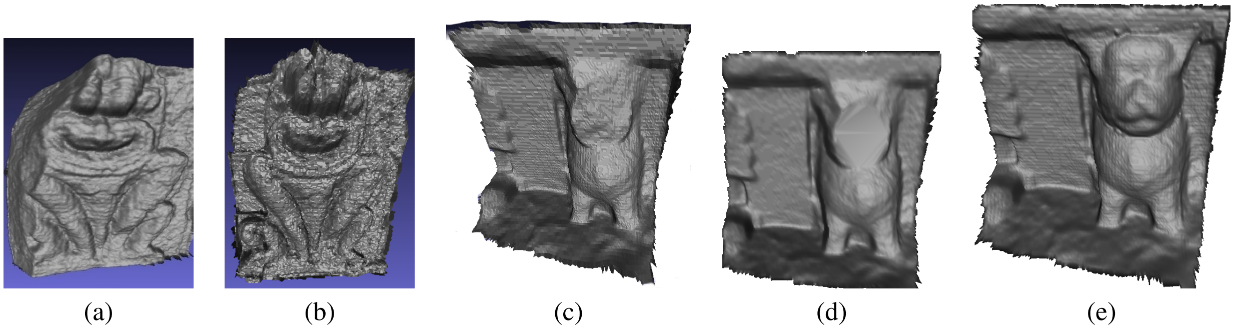


Figure 8. More real results using the proposed method. (a,c) Original damaged structure. (d) Result obtained using Meshlab. (b,e) Inpainted results using our method.

[2] T. P. Breckon and R. B. Fisher. Three-dimensional surface relief completion via nonparametric techniques. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(12):2249–2255, Dec. 2008. 2

[3] M. Callieri, P. Cignoni, F. Ganovelli, G. Impoco, C. Montani, P. Pingi, F. Ponchio, and R. Scopigno. Visualization and 3d data processing in the david restoration. *IEEE Computer Graphics and Applications*, 24:16–21, 2004. 1

[4] P. Cignoni, M. Corsini, and G. Ranzuglia. Meshlab: an open-source 3d mesh processing system. *ERCIM News*, (73):45–46, April 2008. 6

[5] J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. *3D Data Processing Visualization and Transmission, International Symposium on*, 0:428–438, 2002. 2

[6] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309 – 1331, 2009. 1

[7] Google-Art-Project. http://www.googleartproject.com/, 2011. 1

[8] S. Gupta, K. R. Castleman, M. K. Markey, and A. C. Bovik. Texas 3d face recognition database. In *Proc. SSIAI*, pages 97–100, 2010. 5

[9] J. Jia and C.-K. Tang. Inference of segmented color and texture description by tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):771–786, June 2004. 2

[10] Kinect. http://www.xbox.com/en-IN/Kinect/, 2010. 1

[11] M. Kulkarni and A. N. Rajagopalan. Depth inpainting by tensor voting. *J. Opt. Soc. Am. A*, 30(6):1155–1165, Jun 2013. 2

[12] M. Kulkarni, A. N. Rajagopalan, and G. Rigoll. Depth inpainting with tensor voting using local geometry. In *Proc. VISAPP*, 2012. 2

[13] K. Lai, L. Bo, X. Ren, and D. Fox. Rgb-d object recognition: Features, algorithms, and a large scale benchmark. In *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*, pages 167–192. Springer, 2013. 3

[14] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3d scanning of large statues. In *ACM SIGGRAPH 2000*, pages 131–144, 2000. 1

[15] P. Liepa. Filling holes in meshes. In *Proc. 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '03, pages 200–205, 2003. 2

[16] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. ICML '09, pages 689–696. ACM, 2009. 3

[17] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, Mar. 2010. 4

[18] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32:2262–2275, 2010. 2

[19] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 3

[20] p3d. http://p3d.in, 2013. 1

[21] M. Pauly, N. J. Mitra, J. Giesen, M. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Proc. Third Eurographics symposium on Geometry processing*, SGP '05, 2005. 2

[22] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, July 2003. 3, 6

[23] Photosynth. http://photosynth.net, 2010. 1

[24] P. Sahay and A. N. Rajagopalan. Harnessing self-similarity for reconstruction of large missing regions in 3d models. In *ICPR*, pages 101–104, 2012. 2, 5, 6

[25] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, Apr. 2002. 3

[26] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2012. 3

[27] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., 2010. 2

[28] I. Tosic, B. A. Olshausen, and B. J. Culpepper. Learning sparse representations of depth. *Arxiv preprint arXiv:1011.6656*, 2010. 3

[29] J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro. Inpainting surface holes. *In Proc. ICIP*, 2:II–903–6, 2003. 2