

# Non-parametric 3D Surface Completion

Toby P. Breckon and Robert B. Fisher  
Institute for Perception, Action & Behaviour  
School of Informatics  
University of Edinburgh, UK  
toby.breckon@ed.ac.uk

## Abstract

We consider the completion of the hidden or missing portions of 3D objects after the visible portions have been acquired with  $2\frac{1}{2}D$  (or 3D) range capture. Our approach uses a combination of global surface fitting, together with an extension, from 2D to 3D, of non-parametric texture synthesis in order to complete localised surface texture relief and structure. Through this combination and adaptation of existing completion techniques we are able to achieve realistic, plausible completion of  $2\frac{1}{2}D$  range captures.

## Introduction

Common 3D acquisition techniques in computer vision, such as range scanner and stereo capture, are realistically only  $2\frac{1}{2}D$  in nature - such that the backs and occluded portions of objects cannot be realised from a single viewpoint. As a result capturing a complete object in 3D can involve the time-consuming process of multi-view capture and subsequent fusion and registration [1, 17]. Often despite multi-view capture some small regions of the object are still missing post-registration thus requiring hole-filling techniques to produce a completed 3D model [4].

To date the majority of prior work within this area has considered smooth surface continuation in small missing surface patches [2, 4, 15, 23, 21, 12] or the completion of geometrically conforming shapes through the use of shape fitting and parameterisation [5, 3, 10].

This prior work solely concerns itself with the completion of the underlying surface shape and not any texture or features present on the surface. By contrast, here we consider how the localised 3D surface texture and features (*relief*) of a surface can be completed, through the propagation of knowledge from the visible

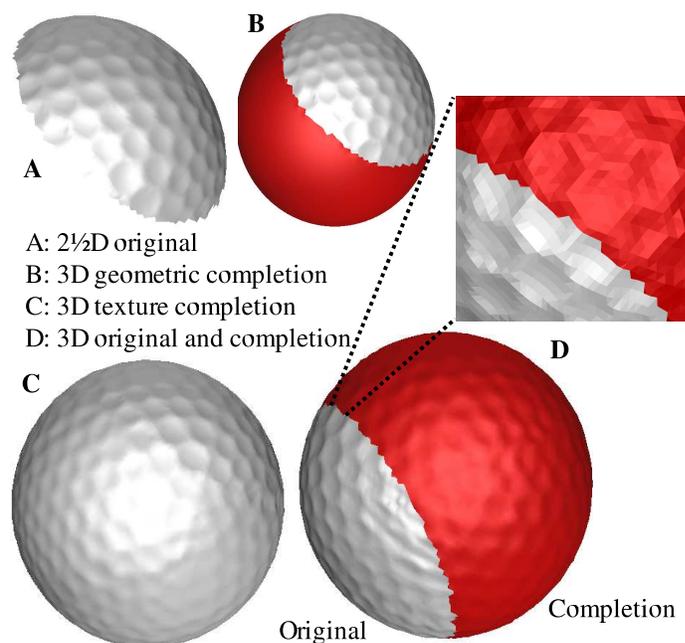


Figure 1. Completion of a  $2\frac{1}{2}D$  golfball

to the unknown surface portion, given that the underlying shape itself can be completed by one of these earlier techniques. For example, we complete both the geometric sphere and surface dimples of a  $2\frac{1}{2}D$  golfball as shown in Fig. 1. Here we see the successful completion of the surface pattern (C/D) over a geometric completion (B) of the original  $2\frac{1}{2}D$  capture (A).

Our approach has 2 parts : firstly to complete the underlying surface shape using simple geometrical techniques akin to [20, 5, 3] and secondly to propagate the 3D surface texture from visible portion to the geometric completion using an adaptation of the 2D texture synthesis technique of non-parametric sampling [8]. The goal is the *plausible completion* of the surface based on the propagation of knowledge from the visible to invisible surface portions - this process itself governed by

the geometric constraint of the earlier shape completion. Although the completions achieved will not be a precise reconstruction of the invisible portion, which is unobserved and hence unknown, they will at least be *visually acceptable* to be viewer and *plausible* as an original.

Concurrent work [19] has considered a similar approach to that proposed here based on propagating 3D surface patches from visible to unknown surface portions. However, as shown in [19], this patched based approach relies on the existence of suitable propagatable patches in the original surface portion. Although computationally more expensive, the fine-detailed per-{point|vertex|range sample} based approach proposed here does not suffer this limitation and lends itself well to the propagation of both tile-able surface textures (see Fig. 7 & 9) and completion/extension of more stochastic surface textures (see Fig. 8 & 10) *derived* from the original without any apparent “tiling” or similar repetitive artifacts.

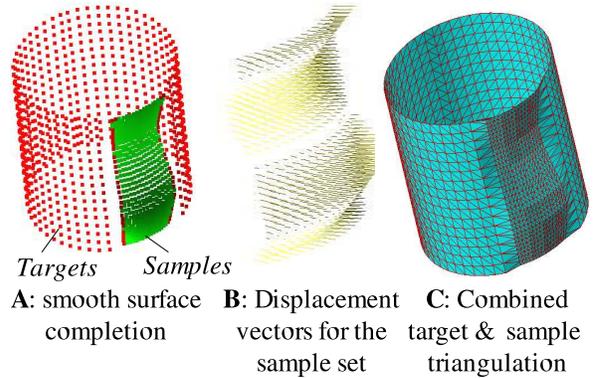
## Non-parametric sampling

Non-parametric sampling was proposed as a method for texture synthesis in 2D images based on using a statistical non-parametric model and an assumption of spatial locality [8]. Unlike other approaches in the texture synthesis arena (e.g. [24, 13]) which attempt to explicitly model the texture prior to synthesis this approach samples directly from the texture sample itself - a kind of implicit modelling akin to the robotics paradigm “the world is its own best model”. As a result it “is very powerful at capturing statistical processes for which a good model hasn’t been found” [8] and thus highly suited to our work in 3D.

In 2D operation non-parametric sampling is very simple - it successively grows a texture outwards from an initial seed area, one pixel at a time, based on finding the pixel neighbourhood in the sample image that best matches that of the current target pixel (i.e. the one being synthesised) and uses the central pixel’s value as the new value for the target.

Matching is based upon using the normalised sum of squared difference metric (SSD) the between two pixel neighbourhoods (i.e. the textured pixels surrounding the target and those surrounding each sample pixel). In addition, to give more influence to pixels closer to the target, each pixel difference contributing to the metric is weighted by a 2D Gaussian kernel across the neighbourhood thus reflecting its influence in inverse proportion to its distance from the neighbourhood centre (i.e. the target).

The neighbourhoods are defined as  $W \times W$  square



**Figure 2. Completion process inputs**

windows around each pixel where  $W$ , window size, is a free parameter perceptually linked to the scale of the largest regular feature present in the texture. In determining the set of similar neighbourhoods for a given target pixel, the normalised SSD between the target neighbourhood and all possible samples are computed. From this set the top  $n\%$  of matches are selected as those with the lowest SSD values from which in turn one is randomly selected to provide the value at the target. Here, as in the original texture synthesis work, we set  $n = 10$ . As an additional constraint the randomly selected match is only used to fill the target provided it has a normalised SSD value less than a specified error threshold,  $e$ , related to the acceptable level of noise in the synthesised texture - a factor directly related to that present in the original sample.

## 3D non-parametric completion

We now adapt the 2D technique to 3D synthesis across a geometric surface. The basic aspects of the approach map well from 2D to 3D : the 2D image becomes a 3D surface, the individual pixel becomes a point on that surface, a pixel neighbourhood becomes the set of nearest neighbours to a surface point and the actual pixel values being synthesised become displacement vectors mapping discrete points on a textured surface to the infinite geometric surface derived from prior fitting.

The pre-processing stage estimates the underlying geometric surface model for the visible scene portion [11, 9] from which a set of displacement vectors,  $D(i)$ , and a corrected surface normal,  $n_i$ , for each point  $i$  can be derived (see Fig. 4). Additionally we derive a completed “smooth” portion of the invisible surface based on parametric shape completion [3] (e.g. Figure 1B).

The main input to our non-parametric completion process is a geometrically completed version of the 3D

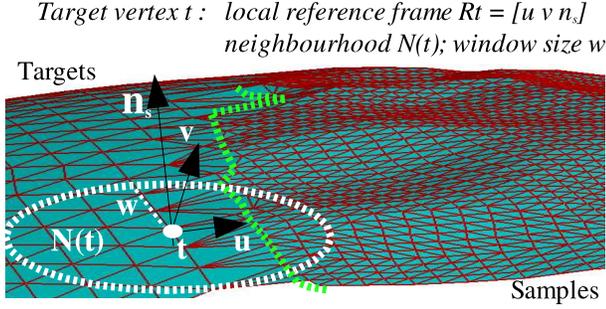


Figure 3. 3D vertex neighbourhoods

surface represented as a discrete set of labelled points,  $P$ . The originals, labelled as textured, are the sample points,  $s \in \text{samples}$ , whilst those of those forming the completed “smooth” portion, labelled untextured, are the target points,  $t \in \text{targets}$ , as shown in Fig. 2A. Each point also has an associated surface normal,  $n$ , and each sample point an associated displacement vector,  $D(s)$ , as shown in Fig. 2B and Fig. 4. For convenience and to aid the construction and spatial use of point neighbourhoods on the surface this input is represented as a combined homoeomorphic surface triangulation [6, 7] of both target and sample points (see Fig. 2C). Hence, from now onwards we consider our points,  $i \in P$ , as vertices,  $i \in \text{triangulation}(P)$ .

The reconstruction algorithm adapts to 3D by considering vertex neighbourhoods on the 3D surface in place of the pixel neighbourhoods of [8]. Each vertex neighbourhood,  $N(i)$ , is the set of vertices lying within a radius of  $W$  edge connections from the vertex being reconstructed (see Fig. 3).  $W$  forms the window size parameter synonymous to that of the earlier 2D approach. The algorithm now proceeds, as follows, by finding the best sample region matching the textured portion of a target vertex’s neighbourhood.

Firstly, the set of target vertices currently lying on the textured/untextured surface boundary are identified as the current target list,  $L$ . The first target vertex,  $t \in L$ , is then matched, using neighbourhood based matching, against every available vertex  $s \in \text{samples}$ . A match is then randomly chosen from the best 10% of this set, based upon matching score. Provided the matching score for this choice is below the specified acceptable error threshold parameter,  $e$ , this choice is accepted and the current target vertex,  $t$ , is textured by mapping the disparity vector,  $D(s)$ , from the chosen sample vertex,  $s$ , to  $t$ . The current target,  $t$ , is now labelled as textured and then algorithm proceeds to the next vertex in  $L$ . If the match is not accepted (or no match was possible) the vertex is simply skipped and returned to the pool of target vertices for future synthesis - in this specific case the window size,  $W$ , as-

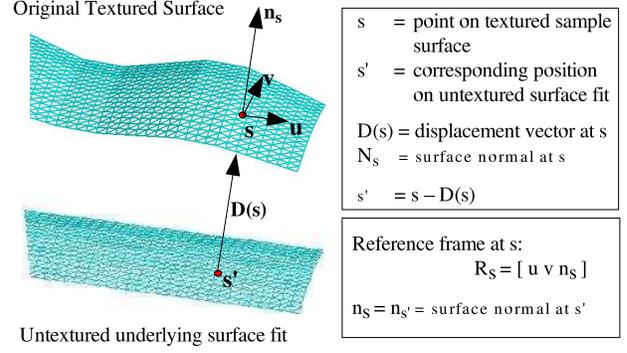


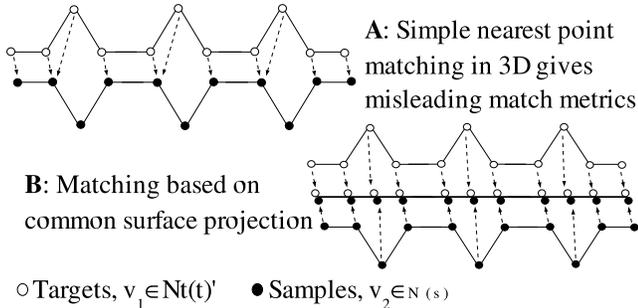
Figure 4. Sample vertex geometry example

sociated with  $t$  for future matching is reduced in size,  $W_t = W - 1$ , to facilitate matching on a scale of reduced constraint, global  $\rightarrow$  local, where required.

Once  $L$  is exhausted, the next set of boundary targets are identified, based on the updated vertex labelling, and the process is continued until all  $t \in \text{targets}$  are labelled as textured. To ensure target vertices are processed in the order of most to least constrained  $L$  is sorted by decreasing number of textured neighbours prior to processing. Additionally, synthesis progress is monitored over each target list constructed - should no match choices be accepted over an entire list, the acceptable error threshold  $e$  is raised slightly (10%) to relax the acceptable error constraint for synthesis as per [8].

The remaining key element in this algorithm outline is the matching of textured target neighbourhoods (as shown in Fig. 3) to vertices in the sample region. This is performed using an adaptation of the SSD metric based on the projection of neighbourhood vertices onto the surface at each sample point. In order to compute the match between target vertex  $t$ , with textured neighbourhood vertices  $Nt(t)$ , and a sample vertex  $s$  with textured neighbourhood  $Nt(s)$ ,  $Nt(t)$  is first transformed rigidly into the co-ordinate system of  $s$ . This is based on knowing the local reference frames at  $s$  and  $t$ , denoted  $R_s$  and  $R_t$  respectfully, which combined with the positional translations given by  $t$  and  $s$  facilitate the transformation of  $Nt(t)$  relative to  $s$  as  $Nt(t)'$ . However, as  $t$  is itself untextured whilst  $s$  is textured, the natural misalignment (owing to the presence/absence of texture) has to be avoided by transforming to the corresponding untextured position of  $s$  on the underlying surface -  $s'$ , calculated using the displacement vector at  $s$ ,  $\overrightarrow{D(s)}$ , as  $s' = s - \overrightarrow{D(s)}$ . Overall we have a resulting,  $t \rightarrow s'$ , transformation as follows:

$$Nt(t)' = \begin{bmatrix} [R_s] & s' \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} [R_t] & t \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} Nt(t)$$



**Figure 5. Point matching via surface projection**

In order to estimate this spatial transformation the reference frames  $R_s$  and  $R_t$  are required. Given each vertex normal this can be generally derived using either localised curvature or more global fitting based techniques. Both, however, have disadvantages - notably their intolerance to noise and additionally the underlying ambiguity of surface orientation on many common geometric surfaces. Here, localised reference frames are derived deterministically based on finding mutually perpendicular vectors,  $u, v$ , to the surface normal,  $n = (x, y, z)$ :

$$\begin{aligned}
 \text{if } x = \min(\{|x|, |y|, |z|\}) \\
 \text{choose } u = (0, -z, y) \\
 v = n \times u
 \end{aligned}$$

And by similar construct when  $y$  or  $z$  is the smallest.

Although far from perfect, this ensures at least localised consistency whilst the problems of global inconsistency are solved by simply augmenting the algorithm to match the target neighbourhood to every sample region at  $R$  different rotational orientations around the normal axis - additional parameter  $R$  specifies the divisions of  $2\pi$  giving a set of rotations (e.g.  $R = 4$  gives 4 orientations at  $0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$ ).

To aid understanding, an illustrative overview of the surface geometry described here is shown in Fig. 3 and Fig. 4.

The task now is to compute the SSD as a vertex matching problem between this transformed neighbourhood,  $Nt(t)'$ , and the textured surface vertices at  $s$ . Although this seems to be a simple 3D point matching problem the presence of sampled surface texture means that simple Euclidean space “nearest point” matching using the raw textured vertices can produce artificial matches in common scenarios as shown in Fig. 5A. Although such problems could be overcome by enforcing a

scheme of one-to-one minimal distance cross-matching between the sets, this relies on the assumption that the densities of both point sets are equal - this is both difficult to assert uniformly and, as we shall discuss later, their inequality becomes a salient issue.

Here we ensure consistent vertex matching, independent of relative density, by matching vertices,  $v_1 \rightarrow v_2$   $v_1 \in Nt(t)'$   $v_2 \in N(s)$ , based on their relative projected positions on the common surface model, embodied in the displacement vector associated with every vertex,  $v'_i = v_i - \overrightarrow{D}(v_i)$ . This effectively matches vertices based solely on their relative spatial surface position rather than relative textured-related depth as shown in Fig. 5B. From these pairings in surface projected space,  $v'_1 \rightarrow v'_2$ , the SSD is calculated based on the original vertex positions,  $v_1 \rightarrow v_2$ .

It should also be noted that here we are *not* performing a neighbourhood,  $Nt(t)'$ , to closed neighbourhood,  $Nt(s)$ , match. Although our notation,  $Nt(s)$ , conceptually represents the surface vertices in the local region of  $s$ ,  $Nt(t)'$  actually is matched against the unrestricted set of textured vertices,  $N(s) = (\{i \in P \mid \text{label}(i) = \text{textured}\})$ , with a viable match only being considered when all matching partners,  $v_2$ , of  $v_1 \in Nt(t)'$  are themselves also textured (i.e.  $v_2$  has assigned label “textured”). When a viable match is found the SSD is calculated based on the distance of each target vertex,  $v_1 \in Nt(t)'$ , directly to the complete triangulated surface (not just the closest vertex) - i.e. the minimum squared distance to any surface triangle,  $\Delta_j$ , that has  $v_2$  as a vertex,  $\Delta_j \in \text{triangles}(v_2)$ :

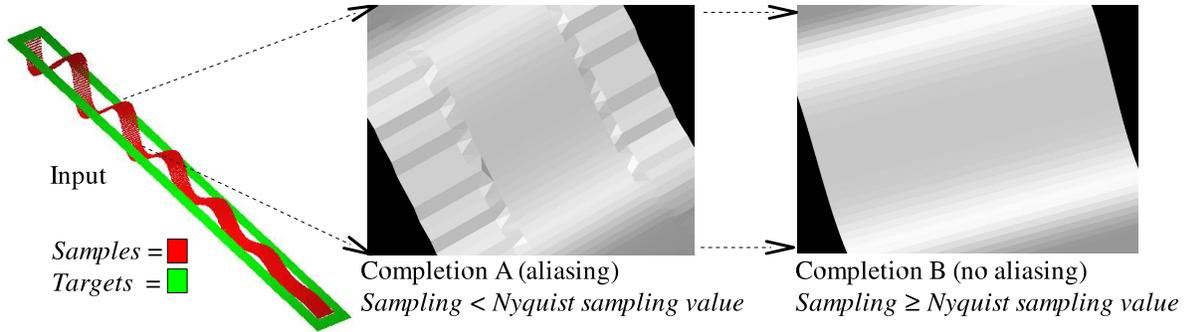
$$SSD = \sum_{v_1}^{Nt(t)'} d_{v_1} \min_{\Delta_j \in \text{triangles}(v_2)} (\text{dist}(v_1, \Delta_j)^2)$$

Additionally, as in [8], a weight  $d_{v_i}$ , based on a 2D Gaussian kernel is used to weight the SSD vertex matches,  $v_1 \rightarrow v_2$ , relative to the distance  $t \rightarrow v_1$   $v_1 \in N(t)$  (i.e. spatial proximity to  $t$ ).

Pseudocode of the non-parametric 3D completion algorithm is available at: <http://homepages.inf.ed.ac.uk/~s9808935/research/NP3D/alg.pdf>.

## Sampling in 3D

One aspect highly relevant to this work is the adaptation of common sampling theory to 3D capture. Although the concepts of under-sampling, aliasing and the Nyquist frequency for a given real world signal are common to general signal processing in lower dimensions [18] it would appear to have received little attention in 3D vision. The specific sampling question that concerns us here is: given an existing surface capture



**Figure 6. Aliasing in 3D completions**

what is the required target vertex density to achieve synthesis without suffering aliasing effects? This is synonymous to obtaining the Nyquist frequency for the capture itself.

Based upon the Nyquist sampling theorem, that a signal must be sampled at twice the frequency of its highest frequency component, it can thus be derived that the upper limit on the Nyquist frequency,  $f_{Ny}$ , of a given signal capture is  $\frac{1}{d}$  where  $d$  represents the signal sampling density. This represents the minimum frequency at which the capture must be sampled in order to allow perfect reconstruction and is equal to twice the highest frequency component,  $v$ , of the signal,  $f_{Ny} = \frac{1}{d} = 2v$ .

Transferring this principle back into the context of 3D triangulated surfaces, where the vertices are the sample points and the depth value of the signal, we have to consider that the sampling frequency across the whole surface may be non-uniform due to variation in the original capture process. Hence only a lower limit on the sampling density required to successfully represent the maximum detail or highest frequency components can be considered based on the maximum surface sample density. This translates as the minimum distance between any two signal samples or conversely the minimum edge length,  $min(e)$ , present in a Delaunay based triangulation (e.g. [6, 7]). This gives an upper limit on the Nyquist frequency,  $f_{Ny} = \frac{1}{min(e)}$ , and an upper spectral component limit,  $v = \frac{1}{2min(e)}$ , for the surface capture.

Surface extension must thus use a vertex sampling density,  $d$ , of at least  $min(e)$  to avoid the effects of aliasing and ensure restoration of the surface ( $d \geq min(e)$ ). This is illustrated in Fig. 6 where for a synthetic surface case we see that using a sampling density for the target vertices set below that associated with the Nyquist frequency (Fig. 6:A) causes aliasing, whilst using the minimum edge length removes the aliasing artifacts, (Fig. 6:B).

Our final issue in 3D sampling arises from remem-

bering that here we are sampling and reconstructing from a finite digitised representation of a signal, a set of vertices representing surface sample points, rather than the infinite analogue signal commonly considered. Although the infinite surface is arguably represented by the surface lying through these points, embodied here in a triangulation, the nature of the non-parametric sampling technique requires finite to finite domain reconstruction, represented here by the sets of sample and target vertices. This introduces an issue relating to vertex alignment between the two regions. If there exists a significant phase shift between the target vertex set and the samples this results in a scenario where the suitable displacement value for a given target vertex, given its spatial position on the surface, is not adequately represented in the sample set - it in fact lies at some other point on the infinite surface. Due to the nature of this technique and limitations in the ability to identify/correct phase shifts in this domain we solve this problem by oversampling the original surface capture - creating the intermediate samples as required. It should now be clear that having an approach that is independent of a common point density for the sample and target portions is highly desirable. Practically, oversampling is achieved by subdividing the surface using an adaptation to surface tessellation such that each triangle is replaced by 4 co-planar triangles. For  $v$  original vertices, by reference to Euler's formula, this results in  $v'$  vertices where  $v' \geq 2v$  but with no increase in the surface detail, and hence no increase in the Nyquist related surface properties.

Overall, from our 3D sampling discussion, we now have a practical means of determining a suitable surface reconstruction, the minimum triangulation edge length, and an oversampling solution for phase alignment problems.

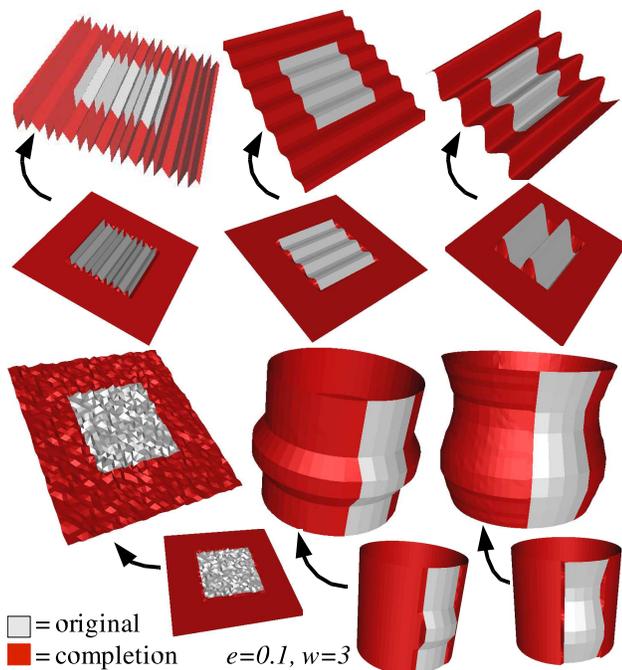


Figure 7. Completion of synthetic examples

## Results

Here we present a number of 3D surface completions using our approach. Firstly, in Fig. 7 we see the successful completion of synthetic wave and noise patterns over planar surfaces and the completion of localised surface shape on cylindrical surfaces. Surface completions based on using real object portions, scanned with our 3D Scanner's Reversa laser scanner, are presented in Figures 1, 8, 9, 10. These show the successful completion of a range of surface types from the propagation of golfball dimples across the completed sphere (Fig. 1), natural tree bark texture realistically completed over an extended cylinder (Fig. 8) and structured surface completion of a scale model of the Pisa tower (Fig. 9). The extension of natural surface texture from a small surface sample over a wider region is shown in Figure 10. Additionally we show the suitability of this technique to realistic surface hole-filling (Fig. 11) akin to the untextured approach of [2, 4, 15, 21, 12].

These results are based on using Euclidean [9] or least squares fitting [11] for initial geometric surface completion, oversampling the original portion once and Cocone surface triangulation [6, 7]. Mersenne twister [16] provided the random source and k-d search trees provided fast point location queries. All completions are based on using only the set of original textured points as the sample vertices. (The variation called "boot-strapped" completion, whereby the usable sam-

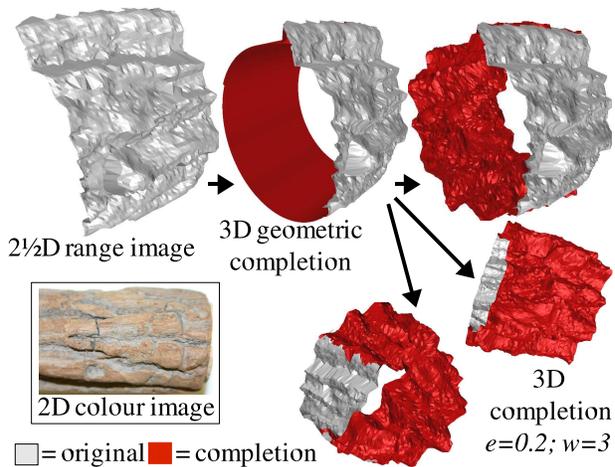


Figure 8. Completion of natural textures - tree bark

Object	Original	Completion	% diff.
Fig. 7 bottom right	0.247123	0.252846	2.32%
Fig. 9 bottom right	0.807048	0.828891	2.71%
Fig. 8	1.18208	1.24769	5.55%
Fig. 10 left	1.22093	1.30366	6.77%
Fig. 10 right	0.417207	0.476877	14.30%
Fig. 11	0.659935	0.549649	16.71%

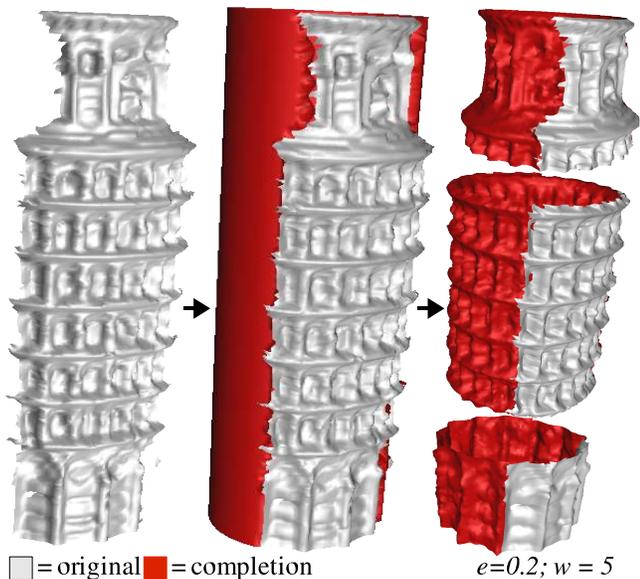
Table 1. Mean Integral below surface texture.

ple regions grow as the textured surface area grows, is not considered.)

Overall the results produce realistically structured and textured surface completions representing *plausible completion*. Erroneous completions were, however, encountered in some cases due to the effects of accumulated error and illustrate the reliance on good parameter choice (see Fig. 12). Future work will aim to address this issue.

As a means of quantitative evaluation, the mean integral of the volume between the geometric surface fit and the original and synthetic (completed) surface portions for a sample of results are shown in Table 1. These statistics support the visual similarity of the results but also show a statistical increase in difference where either the original sample is limited (i.e. Fig. 11) or the texture is stochastic in nature (i.e. Fig. 8 & 10). In both cases the statistics identify a difference not apparent to visual inspection (see Fig. 8, 11 & 10) and hence arguably within the bounds of visually plausible completion - our desired goal.

Additionally, despite extensive pre-computation and memoisation, this technique is computationally very expensive.  $(\mathcal{O}(stw))$  for  $s$  samples and  $t$  targets and



**Figure 9. Completion of tower of Pisa**

window size  $w$ . Fig. 8 requires  $\sim 13$  hours on a 2.6Ghz Pentium 4 with  $t = 7200$ ,  $s = 12852$ .)

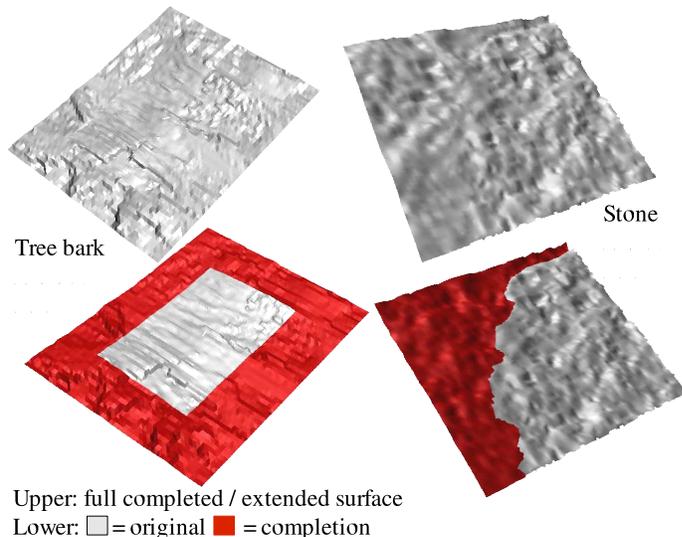
Improvements may be gained upon this computational bound by constraining the set of samples considered for matching to a given target,  $t \in \text{targets}$ , to a subset of those available from the original  $2\frac{1}{2}D$  surface,  $s \in S$ ;  $\{S\} \subset \text{samples}$ . In cases where reasonable regularity or texture repetition in the original  $2\frac{1}{2}D$  surface can be assumed a randomly chosen set of samples,  $S$ , may provide adequate sampling to facilitate plausible completion. However, if the set,  $S$ , is too small or this assumption invalid then aliasing and “tiling” artifacts may become apparent in the resulting completion. Such sample selection could be random for each given target  $t$  or utilise a precomputed match heuristic such as the shape signatures of [19] and remains an area for future work.

Alternatively, in terms of practical computation, the proposed technique lends itself well to a parallelism.

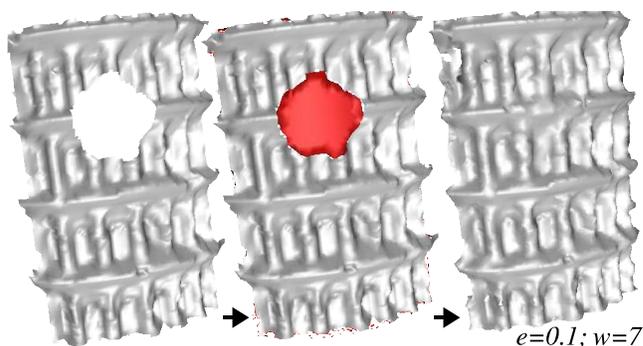
Both these limitations, in computation and error accumulation, echo those identified in the earlier 2D work [8].

## Conclusions and further work

We have presented a method for 3D surface completion that, given the underlying surface geometry, plausibly completes textured surfaces without strict localised surface geometry. This extends earlier work in this field based on surface hole filling [4, 15, 23, 21, 12] and strict geometric completion [20, 3, 5, 10] and also a related use of this technique in completing range data



**Figure 10. Extension of natural surface textures**



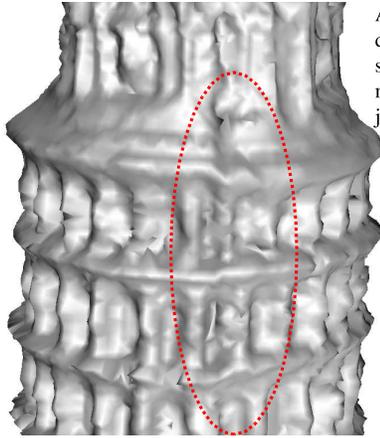
**Figure 11. 3D completion for hole filling**

based on explicit intensity knowledge of the unknown area [22].

In contrast to the work of [19] this technique does not suffer the limitations of such a patch based approach, at the expense of computational cost, but does rely on knowledge of the underlying “smooth” surface completion - here derived from geometric fitting but possibly obtainable from prior techniques in smooth surface completion [4, 15, 23, 21, 12] and fitting [14] in future work.

A number of further possibilities remain with this work including the integration of intensity data, the extension to non-analytic base surfaces, sub-sampling to reduce computation and the adaptation of other 2D texture synthesis approaches to this problem domain. It is also hoped that future work in pursuing a multi-resolution variant to this technique will address the issues of accumulated error identified previously.

Additionally, interesting issues related to approx-



Accumulated error due to noise causes slight texture mismatches at surface joins.

**Figure 12. Accumulated error due to surface noise**

imating the Nyquist frequency of a 3D surface and in synthesising surfaces through infinite representation models still require investigation - this is of equal interest in 3D storage, transmission and compression as it is in synthesis.

[This work was supported by EPSRC and QinetiQ PLC]

## References

- [1] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.
- [2] J. C. Carr, R. K. Beaton, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proc. 28th SIGGRAPH*, pages 67–76. ACM Press, 2001.
- [3] U. Castellani, S. Livatino, and R. Fisher. Improving environment modelling by edge occlusion surface completion. In *Int. Symp. on 3D Data Proc. Vis. and Trans.*, pages 672–675, 2002.
- [4] J. Davis, S. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *Proc. First Int. Sym. on 3D Data Proc., Vis., Trans.*, pages 428– 861, 2002.
- [5] F. Dell’Acqua and R. B. Fisher. Reconstruction of planar surfaces behind occlusions in range images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):569–575, 2002.
- [6] T. K. Dey and J. Giesen. Detecting undersampling in surface reconstruction. In *Proc. of the 17th ann. symp. on Comp. geo.*, pages 257–263. ACM Press, 2001.
- [7] T. K. Dey and S. Goswami. Tight cocone: a watertight surface reconstructor. In *Proc. of the 8th ACM sym. on Solid modeling and applications*, pages 127–134. ACM Press, 2003.
- [8] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *IEEE Int. Conf. on Comp. Vis.*, pages 1033–1038, 1999.
- [9] P. Faber and R. Fisher. Euclidean fitting revisited. In *Workshop on Visual Form*, page 165 ff., 2001.
- [10] R. B. Fisher. Applying knowledge to reverse engineering problems. *Computer Aided Design*, 36(6):501–510, May 2004.
- [11] A. Forbes. Least-squares best-fit geometric elements. Technical Report 140/89, National Physical Laboratory, Teddington, UK, 1989.
- [12] T. Ju. Robust repair of polygonal models. *ACM Trans. Graph.*, 23(3):888–895, 2004.
- [13] A. Kokaram. Parametric texture synthesis for filling holes in pictures. In *Proc. Int. Conf. on Image Proc.*, pages I: 325–328, 2002.
- [14] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. In *Proc. SIGGRAPH*, pages 313–324. ACM Press, 1996.
- [15] P. Liepa. Filling holes in meshes. In *SGP ’03: Proc. of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 200–205. Eurographics Association, 2003.
- [16] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998.
- [17] M. Rodrigues, R. Fisher, and Y. Liu. Special issue on registration and fusion of range images. *Comput. Vis. Image Underst.*, 87(1-3):1–7, 2002.
- [18] C. Shannon. Communication in the presence of noise. *Proc. Inst. of Radio Engineers*, 37(1):10–21, 1949.
- [19] A. Sharf, M. Alexa, and D. Cohen-Or. Context-based surface completion. *ACM Trans. Graph.*, 23(3):878–887, 2004.
- [20] F. Stulp, F. Dell’Acqua, and R. Fisher. Reconstruction of surfaces behind occlusions in range images. In *Proc. 3rd Int. Conf. on 3-D Dig. Imag. and Modeling*, pages 232–239, 2001.
- [21] L. Tekumalla and E. Cohen. A hole-filling algorithm for triangular meshes. Technical Report UUCS-04-019, School of Computing, University of Utah, Dec. 2004.
- [22] L. Torres-Mendez and G. Dudek. Range synthesis for 3d environment modeling. In *Proceedings of the IEEE/RSJ Conf. on Intelligent Robots and Systems*, page 8, 2003.
- [23] J. Wang and M. Oliveira. A hole-filling strategy for reconstruction in smooth surfaces in range images. In *16th Brazilian Symp. on Comp. Graphics and Image Proc.* IEEE Computer Society, 2003.
- [24] S. Zhu, Y. Wu, and D. Mumford. Filters, random-fields and maximum-entropy (frame): Towards a unified theory for texture modeling. *Int. Journal of Comp. Vis.*, 27(2):107–126, 1998.