

# Plausible 3D Colour Surface Completion Using Non-parametric Techniques

Toby P. Breckon and Robert B. Fisher

Institute for Perception, Action & Behaviour,  
School of Informatics, University of Edinburgh, UK  
toby.breckon@ed.ac.uk, rbf@inf.ed.ac.uk

**Abstract.** We consider the combined completion of 3D surface relief and colour for the hidden and missing portions of objects captured with  $2\frac{1}{2}D$  (or 3D) capture techniques. Through an extension of non-parametric texture synthesis to facilitate the completion of localised 3D surface structure (relief) over an underlying geometric surface completion we achieve realistic, *plausible completion*<sup>1</sup> and extension of  $2\frac{1}{2}D$  partially visible surfaces. Additionally we show how this technique can be extended to the completion of increasingly available colour  $2\frac{1}{2}D$  / 3D range data.

## 1 Introduction

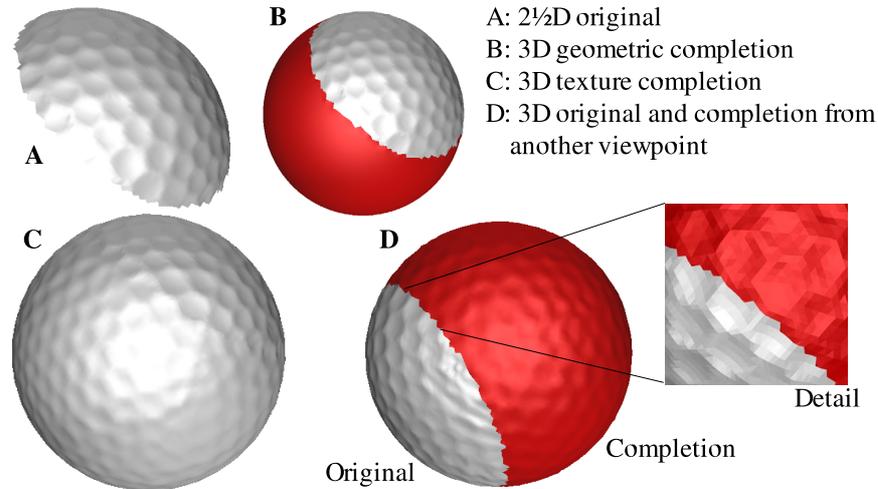
3D data acquisition techniques in computer vision, such as range scanning and stereo photography, suffer from the common problem of their  $2\frac{1}{2}D$  sensing limitation—meaning the back-facing or occluded portions of surfaces within a scene cannot be realised from a uni-directional capture. As a result capturing a complete object in 3D can involve the time-consuming process of multi-directional capture and subsequent data fusion and registration [1, 2]. Often, despite multi-directional capture some small regions of the object remain uncaptured, giving rise to the need for hole-filling techniques to produce a completed 3D model [3].

To date, work on this problem has been primarily limited to the completion of smooth surface continuation in small missing surface patches [4, 3, 5, 6, 7, 8] or the completion of geometrically conforming shapes through the use of shape fitting and parameterisation [9, 10, 11, 12, 13]. Such techniques, although valid, limit truly plausible surface completion (i.e. cases where the original surface portion and the completion are indistinguishable) to a smooth or geometrically conforming subset of all real-world occurring surfaces.

Contrastingly, here we consider the localised completion of *real* 3D surfaces in terms of both its 3D surface structure (relief) and colour detail. Our approach has two parts: firstly we assume the underlying surface can itself be completed

---

<sup>1</sup> By plausible, we mean passable to the viewer as if it were original and indistinguishable from any existing original part.



**Fig. 1.** Completion of a 2½D golfball

using one of these prior smooth completion techniques (1); over which the localised relief and colour completion is then achieved through the propagation of knowledge from the original to the unknown surface portion (2).

As an example, we complete both the geometric sphere and surface dimples of a 2½D golfball as shown in Fig. 1. Here we see the successful plausible completion of the surface relief pattern (Fig. 1(C,D)) over a geometric completion (Fig. 1(B)) of the original 2½D capture (Fig. 1(A)).

Concurrent work [14] has also considered a similar approach based on propagating 3D surface patches from visible to unknown surface portions. However, as shown in [14], 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- $\{\text{point—vertex—range sample}\}$  based approach proposed here does not suffer this limitation. Instead it lends itself well to the propagation of both tile-able surface textures (see Figs. 1 and 10) and the completion/extension of more stochastic surface textures (see Figs. 11 and 12) derived from the original—without any apparent ‘tiling’ or similar repetitive artifacts. In addition our combination of both localised 3D structure and colour completion extends beyond that of [14].

Considerable recent interest in the texturing of 3D objects with 2D colour textures is also of note (e.g. [15, 16, 17, 18, 19])—in some cases to give *illusion* of a real 3D surface displacement texture [18, 19] or merely to enhance the 3D appearance of a surface in a given rendering [15, 16, 17]. Although related in some aspects, many of these approaches fall short of the problem we pose here, concentrating instead on the consistent mapping of often fairly arbitrary, synthetic textures onto relatively smooth 3D surface. The main advantage of such smooth surfaces is the ease of which consistent localised reference frames can be derived

automatically [17, 18] or specified from user input [15, 16, 19] to facilitate the orientation of a 2D texture over the 3D surface.

Related work has also considered aspects of geometric texture transfer - the semi-automated transfer of surface relief from one geometric object to another [20, 21, 22, 23, 24]. Although addressing a similar problem a number of issues, in addition to the techniques utilised, differentiate it from the approach we consider here: the use of smooth surfaces [20, 21, 22], synthetic relief textures [21, 23], the overall lack of relief/colour integration and in some cases a degree of user interaction [20, 22].

In contrast, we consider the completion and extension of existing non-smooth surfaces containing both highly regular 3D textures (e.g. Figs. 1 and 10) and more stochastic natural textures (e.g. Figs. 9 and 11) in terms of both *real* 3D surface displacement texture (relief) and colour. Our overall aim being the *plausible completion* and extension of a given real-world surface sample so that seamless *good continuation* exists between the original and the completion.

In this paper we combine our prior work in completion of 3D surface relief [25] with an additional aspect of colour completion. Firstly we outline and present our background work in this area [25] together with some results and then detail its recent extension to dual surface structure / colour completion.

## 2 2D Non-parametric Texture Synthesis

Non-parametric sampling, lying at the heart of our approach, 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 [26]. Unlike other approaches in the texture synthesis arena (e.g. [27, 28]) 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” [26] 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). It then uses the central pixel’s value as the new value for the target (see Fig. 2).

Matching is based upon using the normalised sum of squared difference metric (SSD) between two pixel neighbourhoods (i.e. the textured pixels surrounding the target and those surrounding each sample pixel). A 2D Gaussian kernel is used across the neighbourhood to assign weights reflecting pixel influence in inverse proportion to distance from the target.

The neighbourhoods are defined as  $W \times W$  square windows around each pixel where  $W$ , window size, is a parameter perceptually linked to the scale of the largest regular feature present in the texture [26]. From the set of all sample neighbourhoods, the top  $\eta\%$  of matches are selected as those with the lowest

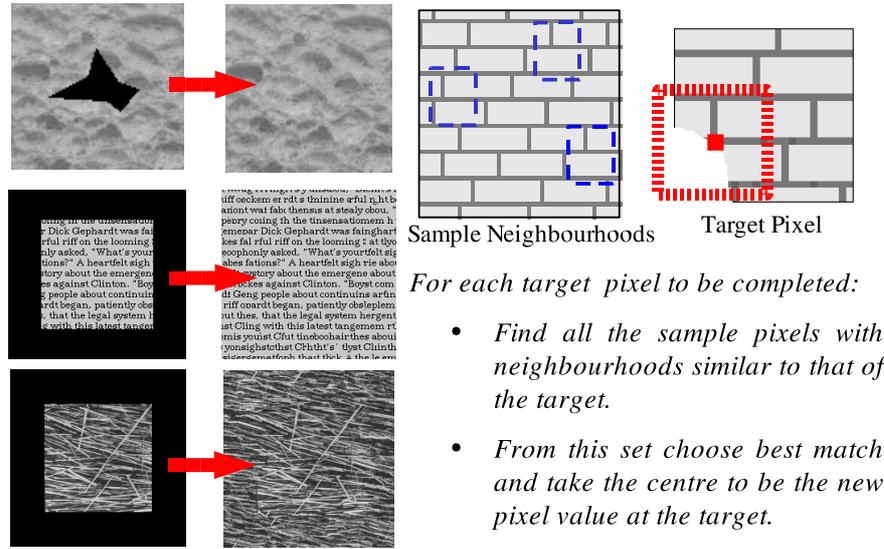


Fig. 2. 2D non-parametric texture synthesis

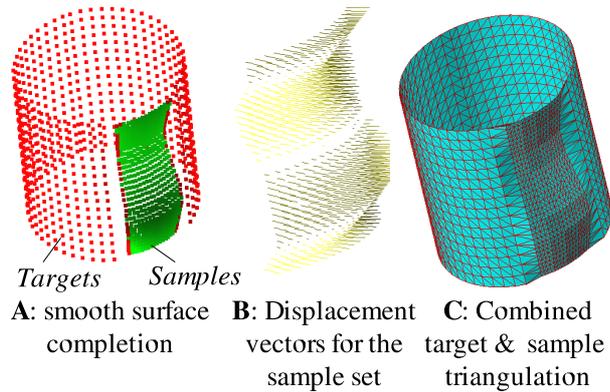
SSD values from which one is then randomly selected to provide the value at the target. As an additional constraint the randomly selected match is only used to fill the target provided it has a normalised SSD value less than a specified error threshold,  $\epsilon$ , related to the acceptable level of noise in the synthesised texture—a factor directly related to that present in the original sample. Here in our 3D approach, as in the original 2D work, we set  $\eta = 10$ .

### 3 3D Non-parametric Completion

Here we review our initial technique for the completion of 3D surface relief from [25]. The extension to combined relief/colour completion is detailed further on in this paper.

The basic aspects of non-parametric sampling 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 geometric surface derived from prior fitting.

The pre-processing stage estimates the underlying geometric surface model for the original scene portion [29, 30] from which a set of displacement vectors,  $\vec{D}(i)$ , and a corrected surface normal,  $n_i$ , for each point  $i$  can be derived (see Fig. 5). Additionally we derive a completed ‘smooth’ portion of the invisible surface based on parametric shape completion [9, 11, 10] (e.g. Figure 1(B)).



**Fig. 3.** ‘Smooth’ surface completion and displacement vectors

The main input to our non-parametric completion process is a geometrically complete version of the 3D 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 forming the completed ‘smooth’ portion, labelled untextured, are the target points,  $t \in \text{targets}$ , as shown in Fig. 3(A). Each point also has an associated surface normal,  $n$ , and each sample point an associated displacement vector,  $\vec{D}(s)$ , as shown in Figs. 3(B) and Fig. 5. 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 [31, 32] of both target and sample points (see Fig. 3(C)). Henceforth we now 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 [26]. 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. 4).  $W$  forms the window size parameter synonymous to that of the earlier 2D approach. The algorithm now proceeds by finding the best sample region matching the textured portion of a target vertex’s neighbourhood, as follows.

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 displacement vector,  $\vec{D}(s)$ , from the chosen sample vertex,  $s$ , to  $t$ . The current target,  $t$ , is now labelled as textured and the 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,  $W$ , associated with  $t$  for future

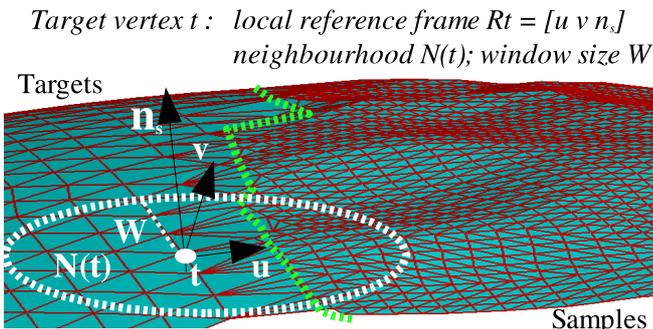


Fig. 4. 3D vertex neighbourhoods

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 [26].

The remaining key element in this algorithm outline is the matching of textured target neighbourhoods (as shown in Fig. 4) 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 or 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  $s$ ,  $\vec{D}(s)$ , as  $s' = s - \vec{D}(s)$ . Overall we have a resulting,  $t \rightarrow s'$ , homogeneous co-ordinate transformation as follows:

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

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,

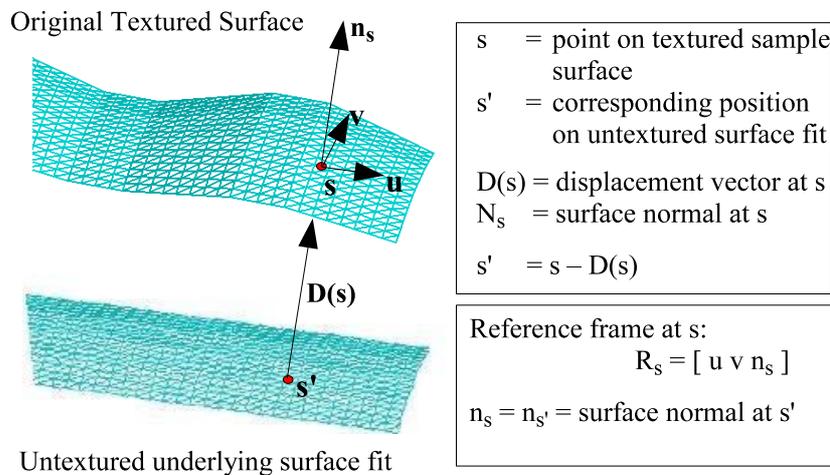


Fig. 5. Sample vertex geometry example

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,  $\vec{u} \ \vec{v}$ , to the surface normal,  $n = (x, y, z)$ :

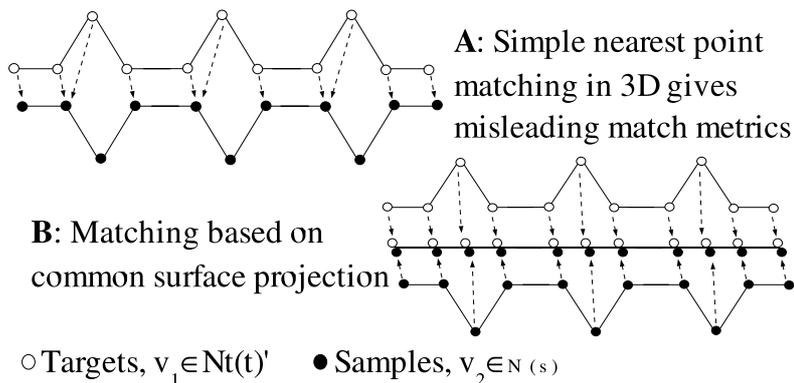
$$\begin{aligned} &\text{if } x = \min(|x|, |y|, |z|), \\ &\text{choose } u = (0, -z, y), \\ &v = n \times u, \end{aligned}$$

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  $0, \pi/2, \pi, 3\pi/2$ ).

To aid understanding, an overview of the surface geometry described here is shown in Figs. 4 and 5.

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. 6(A). 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



**Fig. 6.** Point matching via surface projection

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,  $v_1 \rightarrow v_2$ ,  $v_1 \in Nt(t)$ ,  $v_2 \in Nt(s)$ , based on their relative projected positions on the common surface model, embodied in the displacement vector associated with every vertex,  $v'_i = v_i - \vec{D}(v_i)$ . This effectively matches vertices based solely on their relative spatial surface position rather than relative textured-related depth as shown in Fig. 6(B). 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,  $\Delta_j \in \text{triangles}(v_2)$ :

$$SSD_{\text{shape}} = \sum_{v_1}^{Nt(t)} d_{v_1} \min_{\Delta_j \in \text{triangles}(v_2)} (\text{dist}(v_1, \Delta_j)^2)$$

Additionally, as in [26], 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 for the outlined technique is given in Appendix A.

## 4 Sampling Theory of 3D Surfaces

As identified in [25], one aspect highly relevant to this work is the adaptation of common sampling theory to 3D capture [33].

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 [34] it would appear to have received little attention in 3D vision [33]. The specific sampling question that concerns us here is: given an existing surface capture, 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  $1/\delta$  where  $\delta$  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,  $f_{max}$ , of the signal,  $f_{Ny} = 1/\delta = 2f_{max}$ .

Transferring this principle back into the context of 3D triangulated surfaces, where the vertices are the sample points and the depth value is the signal, we have to consider that the sampling frequency across the whole surface may be non-uniform due to common-place 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. [31, 32]). This gives an upper limit on the Nyquist frequency,  $f_{Ny} = 1/\min(e)$ , and an upper spectral component limit,  $f_{max} = 1/2 \min(e)$ , for the surface capture.

Surface extension must thus use a vertex sampling density,  $\delta$ , of at least  $\min(e)$  to avoid the effects of aliasing and ensure restoration of the surface (i.e.  $\delta \geq \min(e)$ ). This is illustrated in Fig. 7 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 causes aliasing (Fig. 7(A)), whilst using the minimum edge length removes the aliasing artifacts, (Fig. 7(B)).

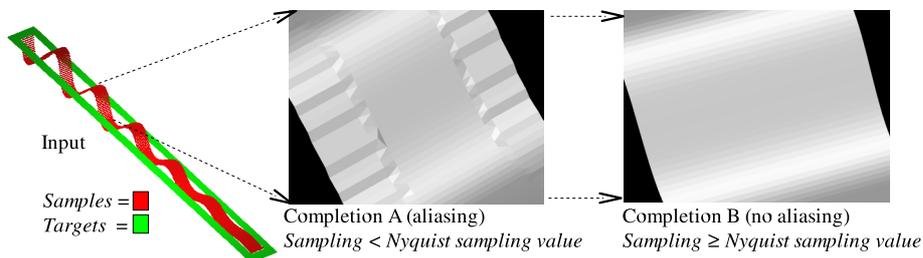


Fig. 7. Aliasing in 3D completions

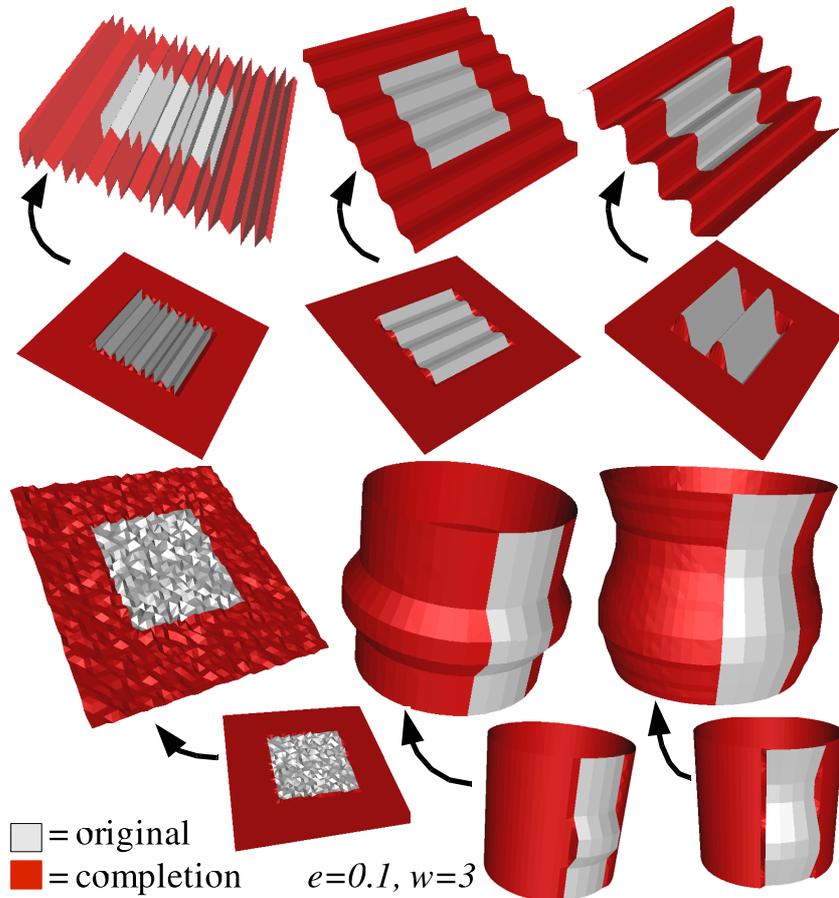
Our final issue in 3D sampling arises from remembering that here we are sampling and reconstructing from a finite digitised representation of a signal, a set of vertices representing surface sample points, rather than the infinite analogue signal commonly considered. Although the infinite surface is 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 and 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 coplanar triangles. For  $i$  original vertices, by reference to Euler’s formula, this results in  $i'$  vertices where  $i' \geq 2i$  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.

## 5 Surface Relief Results

Here we present a number of examples of isolated localised surface structure (relief) completion using our approach. Firstly, in Fig. 8 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 objects portions, scanned with our 3D Scanners’ Reversa laser scanner, are presented in Figures 1, 9, and 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. 9) and structured surface completion of a scale model of the Pisa tower (Fig. 10).

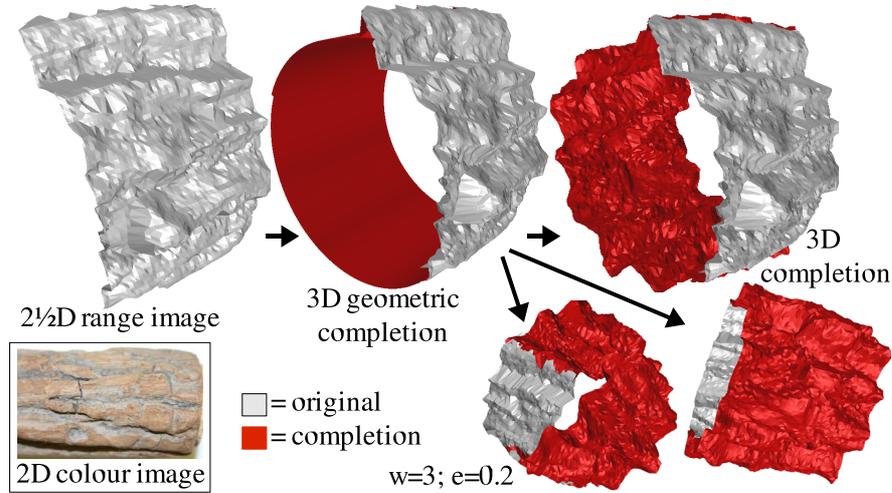
These results were produced using Euclidean [30] or least squares [29] fitting for initial geometric completion. The original sample surface was oversampled once, triangulation utilised the Cocone algorithm [31, 32] and Mersenne twister [35] provided the random source. All completions use only the original, oversampled set of textured points as sample vertices—the variation, called ‘boot-strapped’ completion, whereby the usable sample set grows as the textured surface area grows, is not considered here.



**Fig. 8.** Completion of synthetic examples

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 (i.e. Figs. 8 and 10) but also show a statistical increase in difference where the texture is stochastic in nature (i.e. Figs. 9 and 12). In both cases the statistics identify a difference not apparent to visual inspection (see Figs. 9 and 12) and hence arguably within the bounds of visually plausible completion—our desired goal.

Overall the results produce realistically structured and textured surface completions representing *plausible completion*. However, two limitations experienced in the original 2D work [26] were similarly encountered. Firstly, erroneous completions were encountered in some cases due to the effects of accumulated error and illustrate the reliance on good parameter choice (see [25]). Additionally, despite extensive pre-computation and memoisation, this technique is computationally



**Fig. 9.** Completion of natural textures—tree bark

**Table 1.** Mean integral below surface texture

Object	Original	Completion	% diff.
Fig. 8 bottom right	0.247123	0.252846	2.32%
Fig. 10 bottom right	0.807048	0.828891	2.71%
Fig. 9	1.18208	1.24769	5.55%
Fig. 12	0.15616	0.166552	6.65%

tionally very expensive ( $O(stw)$  for  $s$  samples and  $t$  targets and window size  $w$ . Fig. 9 requires about 13 hours on a 2.6Ghz Pentium 4 with  $t = 7200$ ,  $s = 12852$ ).

## 6 Extension to Colour

In addition to completing the localised surface relief it is also possible to extend our approach to perform combined surface relief and colour completion on increasingly available colour 2½D or 3D range data.

The SSD equation utilised previously,  $SSD_{shape}$ , can be adjusted as follows to take account of both surface shape and colour,  $SSD_{colour}$ :

$$SSD = (\mu)SSD_{shape} + (1 - \mu)SSD_{colour}$$

where  $\mu$  defines the relative weighting of shape (i.e. relief) and colour in the overall matching value. This weight balances the relative importance of localised surface shape against colour in the completion problem. Commonly, as in many natural objects, both are closely interrelated and equal weighting may be suit-

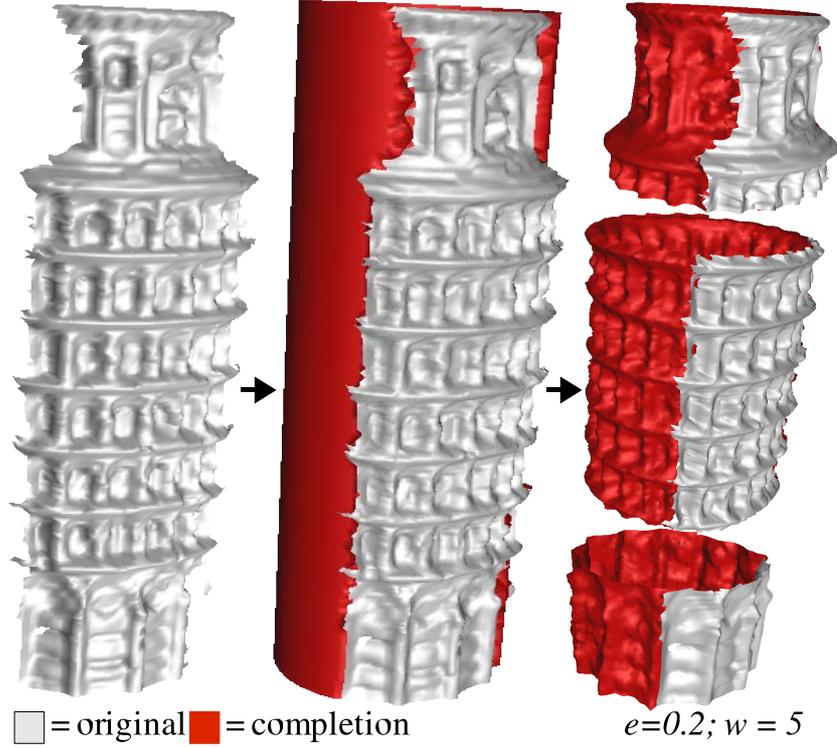


Fig. 10. Completion of tower of Pisa

able. More generally the correlation of colour to surface relief (or relative 3D position) is left as an aspect of psychological vision research [36].

The  $SSD_{colour}$  is calculated based on comparing the RGB colour,  $c \in \{r, g, b\}$ , of each vertex  $v_1 \in Nt(t')$  to that of the nearest sample vertex,  $v_2$ , on the surface at  $s$ :

$$SSD_{colour} = \sum_{v_1}^{Nt(t')} d_{v_1} \sum_c^{\{r,g,b\}} \frac{|c_{v_1} - c_{v_2}|}{3}.$$

As before,  $d_{v_1}$  represents the Gaussian weight associated with the target neighbourhood vertex,  $v_1$ , based on its spatial proximity to  $t$  in  $Nt(t)$ .

Both  $SSD_{shape}$  and  $SSD_{colour}$  are normalised to the range  $[0, 1]$  utilising a known upper bound on each. For colour this is based on the size of the utilised colour map and for shape based on maximum difference between two displacement vectors in the set of samples taking into consideration displacement occurring on either side of the surface fit. This maximum difference is computed by:

$$\max_{s_i, s_j \in samples} \left\| \left( n_{s_i} \cdot \frac{\vec{D}(s_i)}{\|\vec{D}(s_i)\|} \right) \vec{D}(s_i) - \left( n_{s_j} \cdot \frac{\vec{D}(s_j)}{\|\vec{D}(s_j)\|} \right) \vec{D}(s_j) \right\|$$

where  $n_s$  is the surface normal at sample  $s$ .

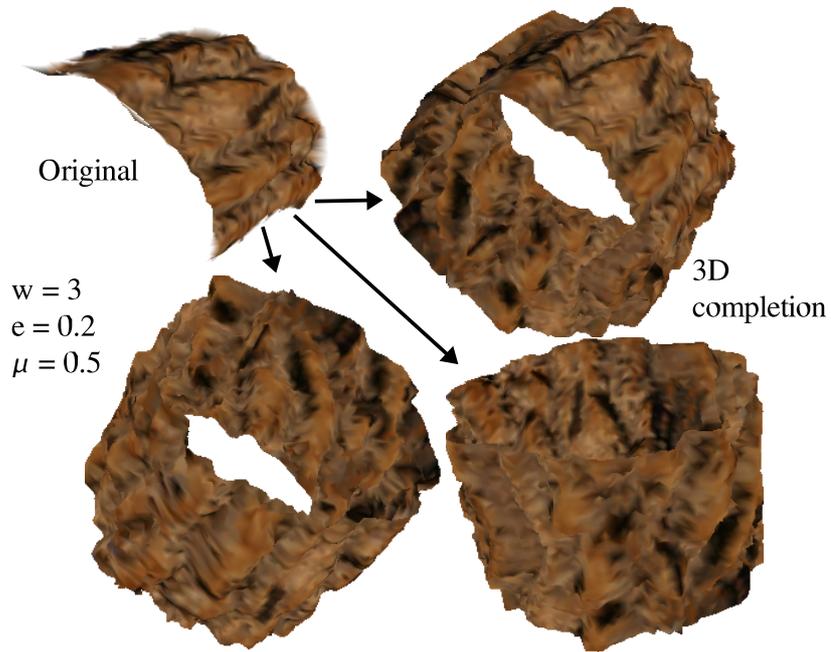


Fig. 11. Completion of tree bark surface relief and colour

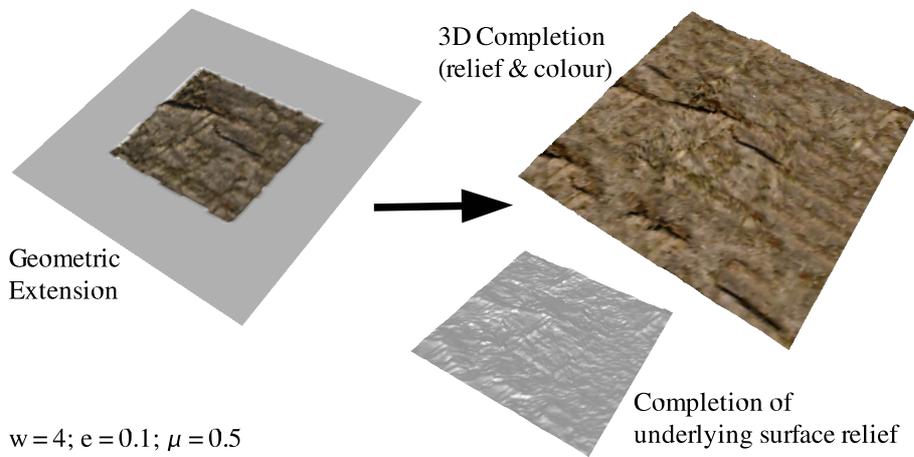
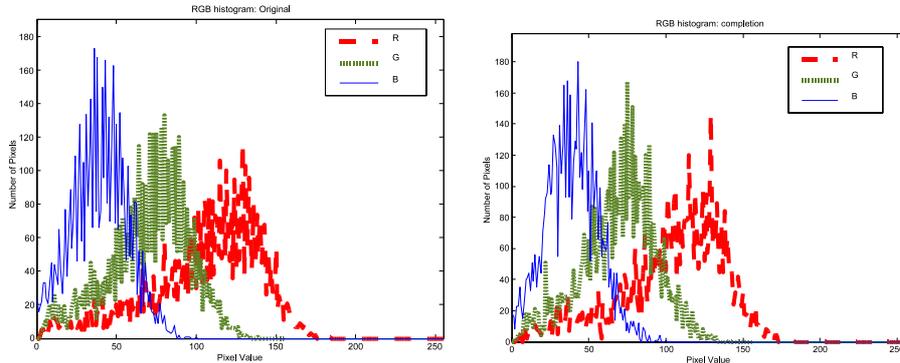


Fig. 12. Extension of tree bark relief and colour

Using this alternative SSD equation the algorithm operates as previously outlined with the additional step of colour propagation from sample to target when a suitable match is chosen (see Appendix A). Oversampling of the coloured



**Fig. 13.** RGB histogram of original (left) and completed (right) tree bark portion in Fig. 11

sample surface is performed by colouring each new sample, due to tessellation, using a Gaussian weighted sum of the  $k$  nearest neighbours.

The combined colour and surface shape completion of a section of tree bark, captured using a high resolution stereo capture rig, is shown in Fig. 11<sup>2</sup>. This shows a highly *plausible* completion of the bark surface texture and associated colour over the underlying geometric cylinder completion of the original bark sample (as per Fig. 9). In addition to this visual comparison, the colour histograms for the original and completed portions, separated by RGB colour channel (Fig. 13), show a strong correlation between the colour distribution achieved in the completed portion and that of the original. A further example of successful colour and surface shape completion is shown in Fig. 12 where we see the clear extension of existing features from the original (i.e. continuation major ridges present in bark sample) and the derivation of similar distinct features (i.e. new ridges formed in foreground of completion) in addition to the general plausibility of the completed relief and colour correlation.

## 7 Conclusions and Further Work

In this paper we have presented a novel method for 3D surface completion that, given the underlying surface geometry, facilitates the *plausible completion* and *good continuation* of surfaces without strict surface localised surface geometry both in terms of localised surface structure (relief) and colour.

This work extends, and indeed complements, earlier work on the smooth surface completion and hole-filling [4, 3, 5, 6, 7, 8] and on strict geometric completion [9, 10, 11, 12, 13] with both its completion abilities for localised surface structure and also the integration of concurrent structure and colour completion.

<sup>2</sup> Colour versions of Figures 11 and 12 are also available from the Web site <http://www.iplab.inf.ed.ac.uk/mvu/breckon/>

Our successful completion / extension of *real* surface relief, combined with colour, presents interesting future directions for work in creating illusory surface relief [18, 19] and enhancing 3D model rendering through realistic and consistent texturing with a 2D sample [15, 16, 17]. Similarly it offers an interesting parallel to work in the transfer of surface reliefs [20, 21, 22, 23, 24] both in terms of the technique proposed and its practical application.

In contrast to the specific completion work of [14], our 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 [3, 6, 5, 8, 7] and fitting [37] in subsequent work.

In terms of future work a number of possibilities remain—notably the extension to surfaces without strict underlying geometry and also work to address the issues of computational complexity and accumulated error identified previously [25]. It is hoped that the investigation of a multi-resolution variant on the proposed technique, together with the adaptation of other 2D texture synthesis techniques to this problem domain, will allow future progress in these areas.

In addition, interesting issues relating to the approximation of the Nyquist frequency of a 3D surface and in synthesising surfaces through infinite representation models still require investigation—an area of equal interest in 3D storage, transmission and compression as it is in synthesis.

**Acknowledgements.** This work was supported by EPSRC and QinetiQ PLC. Textures in Fig. 2 reproduced by kind permission of A. Efros [26].

## References

1. Besl, P.J., McKay, N.D.: A method for registration of 3D shapes. *IEEE Transactions Pattern Analysis and Machine Intelligence* **14** (1992) 239–256
2. Rodrigues, M., Fisher, R., Liu, Y.: Special issue on registration and fusion of range images. *Computer Vision and Image Understanding* **87** (2002) 1–7
3. Davis, J., Marschner, S., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. In: *Proc. First Int. Sym. on 3D Data Processing, Visualization and Transmission*. (2002) 428– 861
4. Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3D objects with radial basis functions. In: *Proc. 28th SIGGRAPH*, ACM Press (2001) 67–76
5. Wang, J., Oliveira, M.: A hole-filling strategy for reconstruction in smooth surfaces in range images. In: *16th Brazilian Symp. on Computer Graphics and Image Processing*, IEEE Computer Society (2003) 11–18
6. Liepa, P.: Filling holes in meshes. In: *Proc. of the Eurographics/ACM SIGGRAPH symp. on Geometry processing*, Eurographics Association (2003) 200–205
7. Ju, T.: Robust repair of polygonal models. *ACM Transactions on Graphics* **23** (2004) 888–895
8. Tekumalla, L., Cohen, E.: A hole-filling algorithm for triangular meshes. Technical Report UUCS-04-019, School of Computing, University of Utah (2004)

9. Stulp, F., Dell'Acqua, F., Fisher, R.: Reconstruction of surfaces behind occlusions in range images. In: Proc. 3rd Int. Conf. on 3D Digital Imaging and Modelling. (2001) 232–239
10. Dell'Acqua, F., Fisher, R.B.: Reconstruction of planar surfaces behind occlusions in range images. *IEEE Transactions Pattern Analysis and Machine Intelligence* **24** (2002) 569–575
11. Castellani, U., Livatino, S., Fisher, R.: Improving environment modelling by edge occlusion surface completion. In: Proc. First Int. Sym. on 3D Data Processing, Visualization and Transmission. (2002) 672–675
12. Fisher, R.: Solving architectural modelling problems using knowledge. In: Proc. 4th Int. Conf. on 3D Digital Imaging and Modelling. (2003) 343–351
13. Chalmoviansky, P., Juttler, B.: Filling holes in point clouds. In Wilson, M., Martin, R., eds.: *Mathematics of Surfaces X*. Number 2768 in LNCS, Springer-verlag (2003) 196–212
14. Sharf, A., Alexa, M., Cohen-Or, D.: Context-based surface completion. *ACM Transactions on Graphics* **23** (2004) 878–887
15. Praun, E., Finkelstein, A., Hoppe, H.: Lapped textures. In: Proc. ACM SIGGRAPH. (2000) 465–470
16. Turk, G.: Texture synthesis on surfaces. In: Proc. ACM SIGGRAPH. (2001) 347–354
17. Wei, L.Y., Levoy, M.: Texture synthesis over arbitrary manifold surfaces. In: SIGGRAPH: Proc. 28th annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM Press (2001) 355–360
18. Gorla, G., Interrante, V., Sapiro, G.: Texture synthesis for 3d shape representation. *IEEE Transactions Visualization and Computer Graphics* **9** (2003) 512–524
19. Zelinka, S., Garland, M.: Interactive texture synthesis on surfaces using jump maps. In: Proc. 14th Eurographics workshop on Rendering, Eurographics Association (2003) 90–96
20. Biermann, H., Martin, I., Bernardini, F., Zorin, D.: Cut-and-paste editing of multiresolution surfaces. *ACM Transactions on Graphics* **21** (2002) 312–321
21. Bhat, P., Ingram, S., Turk, G.: Geometric texture synthesis by example. In: Proc. Eurographics/ACM SIGGRAPH symp. on Geometry processing, New York, NY, USA, ACM Press (2004) 41–44
22. Sorkine, O., Lipman, Y., Cohen-Or, D., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: Proc. Eurographics/ACM SIGGRAPH symp. on Geometry processing, Eurographics Association (2004) 179–188
23. Zelinka, S., Garland, M.: Similarity-based surface modelling using geodesic fans. In: Proc. Eurographics/ACM SIGGRAPH symp. on Geometry processing, New York, NY, USA, ACM Press (2004) 204–213
24. Lai, Y.K., Hu, S.M., Gu, D., Martin, R.: Geometric texture synthesis and transfer via geometry images. *ACM Solid and Physical Modelling (in publication)* (2005)
25. Breckon, T., Fisher, R.: Non-parametric 3D surface completion. In: Proc. 5th Int. Conf. on 3D Digital Imaging and Modelling, IEEE Press (2005) (to appear)
26. Efros, A., Leung, T.: Texture synthesis by non-parametric sampling. In: *IEEE Int. Conf. on Computer Vision*. (1999) 1033–1038
27. Zhu, S., Wu, Y., Mumford, D.: Filters, random-fields and maximum-entropy (frame): Towards a unified theory for texture modelling. *Int. Journal of Computer Vision* **27** (1998) 107–126
28. Kokaram, A.: Parametric texture synthesis for filling holes in pictures. In: Proc. Int. Conf. on Image Processing. (2002) I: 325–328

29. Forbes, A.: Least-squares best-fit geometric elements. Technical Report 140/89, National Physical Laboratory, Teddington, UK (1989)
30. Faber, P., Fisher, R.: Euclidean fitting revisited. In: 4th International Workshop on Visual Form. Volume 2059 of LNCS., Springer-verlag (2001) 165
31. Dey, T.K., Giesen, J.: Detecting undersampling in surface reconstruction. In: Proc. 17th ann. symp. on Computational Geometry, ACM Press (2001) 257–263
32. Dey, T.K., Goswami, S.: Tight cocone: a water-tight surface reconstructor. In: Proc. 8th ACM sym. on Solid modelling and applications, ACM Press (2003) 127–134
33. Taubin, G.: Geometric signal processing on polygonal meshes. In: Proc. EURO-GRAPHICS: State of the Art Report (STAR), Interlaken, Switzerland (2000)
34. Shannon, C.: Communication in the presence of noise. Proc. Inst. of Radio Engineers **37** (1949) 10–21
35. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans. Modelling and Computer Simulation **8** (1998) 3–30
36. Wandell, B.: Foundations of Vision. Sinauer Associates (1995)
37. Krishnamurthy, V., Levoy, M.: Fitting smooth surfaces to dense polygon meshes. In: Proc. SIGGRAPH, ACM Press (1996) 313–324

## A Algorithm Pseudocode

This technique is now outlined in pseudocode, following in the style of the original 2D work [26], based on the definition of the following key data items:

- surface = triangulated surface with vertices labelled as textured/untextured.
- targets = list of untextured vertices (targets) of A.
- samples = list of textured vertices (samples) of A.
- $\vec{D}(s_i)$  = surface displacement vector of textured vertex  $S_i$ ,  $S_i \in samples$ .

*GrowSurface(surface, targets, samples)*

```

while targets is not empty
  progress = 0
  theTargets = vertices on textured/untextured boundary of surface.
  for each vertex v in theTargets do
    BestMatches = FindMatches(v, surface, samples)
    if BestMatches not empty
      BestMatch = random selection from (BestMatches)
      if (BestMatch.error < MaxErrThreshold) then
        Propagate D(BestMatch) to v
        progress = 1
        Remove v from targets
        Label v as textured
      end if
    end if
  end for
  if progress == 0

```

```

    MaxErrThreshold = MaxErrThreshold * 1.1;
  end if
end while

```

*FindMatches(v, surface, samples)*

```

neighbours = textured vertices in vertex neighbourhood of v
for every vertex v' of neighbours
  calculate Gaussian weight w(v') relative to distance(v', v)
end for
TotalWeight = sum(i in neighbours, w(i))
for every vertex s in samples do
  neighbours' = neighbours transformed relative to s
  ssd = 0
  validMatch = TRUE
  for each point i in neighbours' do
    closestVertex = closest vertex on surface to i
    if (closestVertex is in samples)
      tri = closest triangle of surface to i
      (v1, v2, v3) = vertices of triangle tri
      if (v1, v2, v3 are in samples)
        distance = min distance from i to tri
        ssd += w(i) * distance
      else
        validMatch = FALSE
      end if
    else
      validMatch = FALSE
    end if
  end for
  if validMatch do
    ssd = ssd / TotalWeight
    Add (s, ssd) to Matches
  end if
end for
BestMatches = all i in Matches with i.ssd <= min(i.ssd) * (1.1)
Return BestMatches

```

For the extension to combined relief and colour, SSD and propagation are redefined as stated earlier.