# A Mesh-based Approach to Incremental Range Image Integration

Daniek Joubert and Willie Brink
Applied Mathematics
Department of Mathematical Sciences
University of Stellenbosch, South Africa
Email: $\{$daniekj, wbrink$\}$@sun.ac.za

***Abstract*—We consider the problem of integrating a sequence of range images, taken of a static scene from different viewpoints, in order to form a consistent surface reconstruction that preserves the geometric structure of the scene. We argue that an incremental approach is desirable, particularly for real-time mapping and navigation applications. Our proposed solution starts by triangulating 3D data from individual range images into a mesh structure, which is adaptively smoothed and decimated. A feature-based image registration procedure is performed to align consecutive meshes, and these meshes are segmented into regions which are matched. Overlapping parts of matched regions are retriangulated and a consensus surface is built using confidence (or certainty) measures. The algorithm is applied to a dataset captured by a Kinect and from the results we conclude that the proposed algorithm has significant potential.**

## I. Introduction

Advancements in speed, accuracy and cost-effectiveness of range scanning technologies have been vast over the past decade or so, which has lead to an increased interest in 3D measurement and reconstruction from range images. Many applications exist, for example in mobile robot navigation, non-evasive surgical procedures, reverse engineering and augmented reality.

A range image is a regularly-spaced lattice of depth values measured along the line of sight from the sensor to 3D object or scene points. Note that depth is therefore typically measured relative to the sensor.

Types of sensors vary from passive stereo vision, active ones such as structured light and time-of-flight, to more invasive probing. In our research we use the Microsoft Kinect, a structured light sensor which projects an infrared pattern onto objects in view and infers depth from pattern distortion. It is also equipped with a colour camera which enables the mapping of colours onto a reconstruction. For calibration of a depth and colour camera pair, including mapping range data to 3D points, the reader is referred to [1].

The problem we are concerned with in this paper is that of range image integration. We suppose that the sensor travels through a scene and returns a number of range images measured relative to the sensor. We wish to align reconstructions from these range images and create consistent surfaces described by a small amount of data, that preserve the underlying global structure of the scene. This may eventually be useful in 3D mapping for autonomous robot navigation.

In developing our solution we believe that the following issues should be considered and incorporated.

*1) Removal of redundant information:* For the sake of computational efficiency 3D information that will not contribute to the global structure (e.g. many points on a flat plane) should be removed — a process we term "decimation".

*2) Representation of range uncertainty:* Not all 3D points obtained from a range sensor may be sampled with the same confidence, and our algorithm should take this into account.

*3) Utilization of all the data:* We wish to include all points from all range images in the integrated surface, with the exception of those deleted during decimation, i.e. we do not necessarily want to discard less confident information.

*4) Incremental:* An incremental approach, desirable for real-time mapping, means that the algorithm must be able to incorporate a new range image into an integrated surface built from previous images. We note that such an integrated surface may not be of the same form as the input data (grid regularity, for example, may be lost). A computational advantage of such an approach is that only the new (previously unseen) and overlapping parts of the surface need to be updated.

*5) Robust against noise and outliers:* Range scanners typically introduce noise, particularly for points far away from the scanner. The design of our algorithm should reflect this.

Our proposed algorithm can be summarized as follows. Images are registered for estimating sensor motion from the previous to the current time step (section III). An initial mesh surface of the new range data is constructed, adaptive smoothing is applied to combat noise and outliers, and decimation is performed by eliminating vertices with negligible curvature (section IV). We then apply region segmentation and matching between our current (integrated) surface and the new mesh, retriangulate corresponding regions, and construct a consensus surface by taking measurement confidences into account (section V). Before we discuss these steps in more detail, we first provide a brief overview of related work from the literature.

## II. Related work

Previous research on range image integration may be divided roughly into two categories: volume-based methods and mesh-based methods.

In volume-based approaches the space in view of the sensor is discretized into voxels from which an isosurface can be extracted, normally using marching cubes [2] or by matching signed distance fields [3]. Many variations of these approaches exist [4], [5], [6]. However these methods can be sensitive to sampling noise, in many cases cannot provide exact surface topology, and thin surfaces can present problems.

A mesh-based reconstruction makes full use of the underlying topological and geometric structure as it provides neighbourhood, curvature and surface orientation information. It is also more amenable to data reduction, an important consideration for navigational purposes. By creating a triangulated surface we introduce 2-manifoldness which is another desirable property of any surface reconstruction method.

A mesh-based approach usually starts by building some initial mesh from the 3D range data [7]. Overlapping regions are detected, for example by using a proximity relation [8] or by projecting the mesh back to 2D and intersecting circumscribed circles of triangles [9]. The manner in which overlapping regions are handled varies. Less confident triangles can be discarded [9], [10] which may introduce holes that need to be filled. Our algorithm does not have this problem. Another option is to remove redundant triangles on the boundaries of the two meshes to be integrated, until they just meet, and then zipper them together [11]. However it may result in many small sharp-angled triangles, another problem our algorithm avoids. Finally the holes mentioned above need to be filled by seaming the patches remaining after surface removal and the non-overlapping parts together. This is normally done by identifying edges on the boundary and selecting candidate vertices with which to form new triangles, based on largest angles [10]. Careful considerations have to taken to ensure that the resulting mesh does not intersect itself. Many mesh-based algorithms terminate at this point, but valuable information has been discarded by keeping only the most confident triangles. There are methods that apply a post-processing step by projecting vertices along their surface normals to obtain some consensus geometry [11], or by smoothing the surface normals [8].

Some research has also been done to obtain the integrated surface using probability considerations [12].

## III. Image registration

We proceed with details of our incremental mesh-based algorithm for range image integration. The first range image in a sequence is converted to a mesh surface, as explained in section IV. It then serves as a base surface, and every range image following in the sequence is integrated with it in an incremental fashion.

Let us now consider a new range image to be integrated with the current surface obtained from integrating all the previous ones. We assume that the range sensor has been calibrated (see e.g. [1]) so that pixels in the range image can be converted to 3D points, and that colour information is also available.

As mentioned in the introduction, the first step is to register the new range image with the previous one, thereby also aligning it with the total reconstructed surface (since the previous one is already aligned and integrated with the total surface). This initial registration step is vital as it essentially controls the accuracy and overall success of the entire integration procedure.

We use colour information returned by the sensor to estimate sensor motion from the previous to the current time step. In brief, matches between salient image features are computed by a method such as SIFT [13] and used to infer the position and orientation of one camera relative to the other. More detail can be found in, for example, [14]. Note that the scale ambiguity problem is easily solved by using available 3D information.

The procedure gives a $3 \times 4$ camera matrix $\mathbf{P}$ associated with the new range image. It can be decomposed as

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R} \,\middle|\, \mathbf{t}\right], \tag{1}$$

where $\mathbf{K}$ is an upper-triangular matrix containing the internal parameters of the camera. The rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ give the necessary information to transform each 3D point $\mathbf{X}_{\text{old}}$ obtained from the range image, which currently is measured relative to the sensor, to new coordinates

$$\mathbf{X}_{\text{new}} = \mathbf{R}^T\left(\mathbf{X}_{\text{old}} - \mathbf{t}\right) \tag{2}$$

measured in the fixed "world" coordinate system.

## IV. Triangle mesh construction

Next it is explained how we build a mesh surface from a collection of 3D points given by the range sensor. An initial triangle mesh is constructed by exploiting the grid structure in range data, noise is removed by an adaptive smoothing technique, and the mesh is then decimated in an effort to remove redundant vertices that do not contribute to the geometric structure of the surface. The resulting mesh will be integrated into the current total surface.

### A. Constructing an initial triangle mesh

We aim to create a neighbourhood structure over the 3D points obtained from a range image, that defines a consistent triangle mesh surface. Vertices close to one another should be connected, and depth discontinuities should be preserved. We achieve the latter by imposing an upper limit on edge lengths.

The rectangular grid structure of the 3D points, due to the fact that they originate from pixel locations in a range image, means that 4-connectivity between vertices is easily identified. We consider four neighbouring vertices at a time, as depicted in Fig. 1. If both diagonals are longer than the maximum allowed edge length, no triangles are formed. If not, the shorter of the two diagonals is chosen and those two vertices are

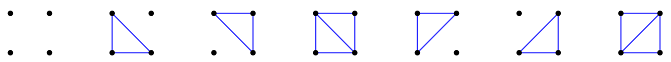

Fig. 1. The seven possible (topological) structures arising from triangulating four neighbouring vertices.

connected. The two triangles that can be formed with this diagonal are evaluated and if all edges of a triangle satisfy the edge constraint, we form that triangle. This procedure will result in one of the seven possibilities shown in Fig. 1 for each such set of four vertices.

### *B. Adaptive smoothing*

Range sensors typically add a fair amount of noise and, if deemed necessary, the initial mesh can be smoothed at this stage. Once again the grid structure in the 3D points can be exploited: the 3D coordinates can simply be convolved with some averaging (say Gaussian) filter in much the same way as we would perform image filtering in the spatial domain.

We decide not to run a smoothing mask blindly over the 3D data in this way, as it may result in undesired behaviour near depth discontinuities. Instead we adapt the filter mask at every vertex to include only the positions to which the specific vertex is connected. Fig. 2 shows a mesh before and after smoothing, and indicates that our technique has the ability to reduce noise while preserving underlying structure.

### *C. Mesh decimation*

Depending on the scene to be reconstructed, many of the triangles in the mesh may be redundant. A planar or piecewise planar surface can be described accurately by a relatively small number of triangles. So, if we can identify vertices approximately coplanar with their neighbours, we may remove those without losing geometric structure.

A commonly used discrete analogy to the continuous notion of Gaussian curvature (or deviation from planarity) is quite simply $2\pi$ minus the sum of all the angles formed by triangles around a particular vertex [15]. The closer that this value is to zero, the closer the vertex is to being coplanar with its immediate neighbours.

We define a cost to every vertex $i$ in the mesh as

$$c_i = \begin{cases} \left|2\pi - \sum \alpha_j\right|, & \text{if } i \text{ is an interior vertex,} \\ \left|\pi - \sum \alpha_j\right|, & \text{if } i \text{ is a boundary vertex.} \end{cases} \tag{3}$$

Here $\alpha_j$ is the angle at vertex $i$ of triangle $j$ connected to that vertex, and a vertex is said to be an interior vertex if its number of connected neighbours equals the number of connected triangles. Vertices with minimum cost are deleted iteratively until a pre-specified maximum cost is reached or until a desired number of vertices have been deleted.

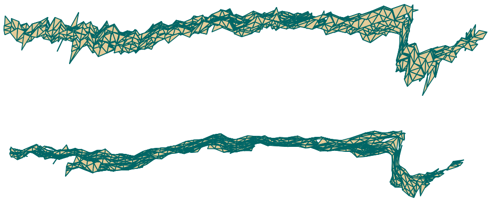

Fig. 2. The effect of our smoothing technique: the original is shown on top, and the smoothed mesh on the bottom. These are side views of the mesh shown on the left of Fig. 4.

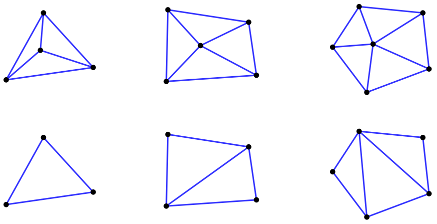

Fig. 3. Removal of internal vertices of degree 3, 4 or 5, and retriangulation. The top row shows the affected part of the mesh before removal, and the bottom row after removal and retriangulation.

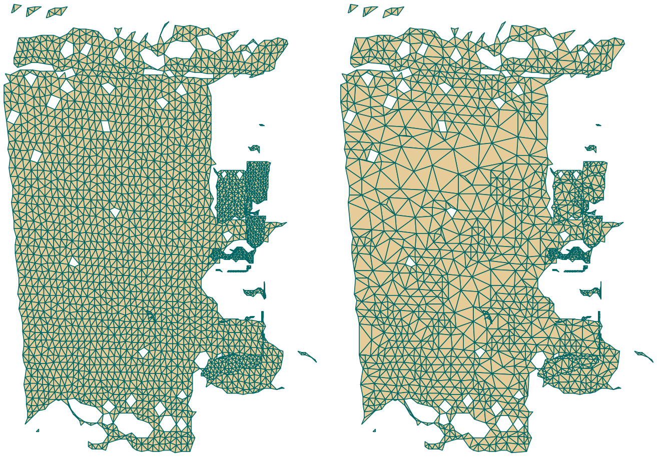

Fig. 4. An example of a triangle mesh before (left) and after (right) our decimation procedure.

After the deletion of a vertex the neighbouring vertices need to be retriangulated to avoid holes being introduced in the mesh. For simplicity and computational efficiency we will consider deleting boundary and interior vertices only of degree 3, 4 or 5 (which in fact, from a graph-topological point of view, is not that constrictive [16]).

Fig. 3 demonstrates our way of retriangulating holes caused by the removal of interior vertices. In the case of a degree-3 vertex the three remaining neighbours are simply connected by a new triangle. The removal of a degree-4 vertex requires one new edge and we pick the shorter of the two possible diagonals. For a degree-5 vertex we first choose the shortest edge between non-adjacent pairs of vertices. This edge divides the hole into a triangle and a loop of four vertices, and both are triangulated as described.

Retriangulation after the removal of a boundary vertex of degree 3, 4 or 5 is performed similarly, except that an extra edge is first added between the two neighbouring boundary vertices. This produces a loop of 3, 4 or 5 vertices which can be triangulated as above.

After a vertex removal and subsequent retriangulation we update the curvatures of the remaining neighbours and move on to select a next candidate for deletion. An example of a mesh before and after decimation is shown in Fig. 4.

## V. Mesh integration

The methods of smoothing and decimating a triangle mesh can be applied to the 3D data from a single range image. Next we discuss the integration of two such meshes that are also already aligned by the procedure outlined in section III.

In short our integration algorithm consists of segmenting each mesh into connected regions and then matching overlapping regions across the meshes. Delaunay triangulation is performed on the vertices of paired regions to build a new integrated mesh. Finally, in an effort to suppress slight misalignment errors, a consensus surface is obtained by taking point confidences into consideration.

### *A. Region segmentation and matching*

The segmentation of a mesh into regions is fairly simple. We start with any particular vertex and all the vertices connected to it. The connected neighbours of those vertices are added to the set. This procedure is repeated recursively until no more vertices are added, and the set thus obtained is classified as a region. A vertex not yet assigned to any region is then selected, and a new region is built from it. The whole process continues until every vertex is assigned to a region.

Regions from two different meshes can now be matched. Many such matching algorithms exist, see e.g. [17], but we simply consider two regions a match if any point of the region from one mesh is contained in the smallest 3D bounding box of the region from the other mesh. By iteratively enlarging this bounding box, multiple regions from the first mesh may be grouped together with multiple regions from the second.

The triangulations of unmatched regions in the two meshes remain unaffected. The matched regions, however, need to be retriangulated in order to integrate them.

### *B. Delaunay triangulation*

Two matched regions, either partially or wholly overlapping, give two separate representations of the same surface. We aim to combine them, in some sensible way, into one triangle mesh.

We employ Delaunay triangulation [18] to connect an unorganized set of points, which will be optimal in the sense that the circumscribed circle of any triangle contains no other vertices. This property maximizes the minimum angle of all triangles, implying that there will be few triangles with small interior angles.

Although Delaunay triangulation can be extended to higher dimensions [18] we decide to project the vertices of the two mesh regions in question to a 2D plane in order to triangulate them. Since one of the meshes (the new one to be integrated into the total surface) is built from a range image, it naturally projects to the 2D image plane of the current sensor. The vertices of both mesh regions are projected onto this image plane with the camera matrix obtained from the registration stage. Fig. 5(a) illustrates with an example.

Those projected vertices that fall within the boundary of the image become eligible for retriangulation (note that we are thus guaranteed that all the vertices of the new mesh will be eligible) as well as immediate neighbours in the old mesh which fall outside the boundary (the circled vertices in Fig. 5). Those triangles that connect the remaining vertices in the old mesh are kept in tact, as in Fig. 5(b).

The vertices eligible for retriangulation are now connected by Delaunay triangulation. Fig. 5(c) shows the result for that example. This triangulation process may form new edges between "circled" vertices. Two such cases are shown in Fig. 5(c), one in red and one in green. It is clear that the red edge is unwanted because it intersects the old mesh. We circumvent this problem by requiring that no new triangles may be formed between 3 circled vertices. We do allow triangles that connect 0, 1 or 2 circled vertices. Fig. 5(d) shows a final triangulation.

We transfer this 2D triangulation to the original 3D coordinates of the two meshes, thus obtaining the desired triangulation over them.

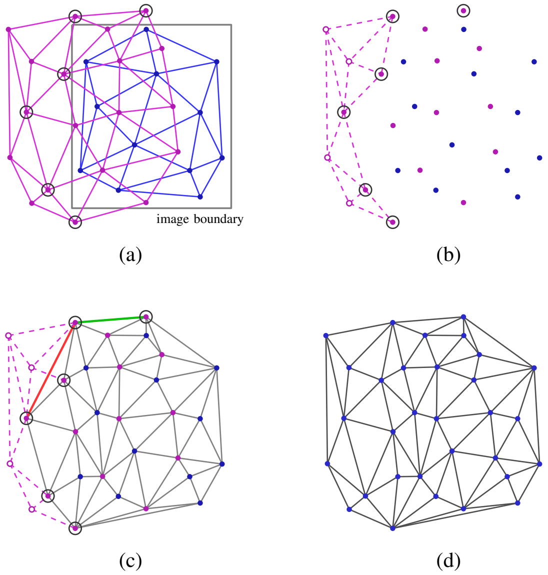

Fig. 5. In (a) the old (shown in purple) and new (blue) meshes are projected to the new sensor image plane. Vertices inside the image boundary, as well as their immediate neighbours outside (the circled vertices), will be subjected to retriangulation. The part of the old mesh not affected by those vertices are kept in tact as shown in (b). Delaunay triangulation is performed, the result of which is shown in (c). The red line is marked as an unwanted edge and removed. The final triangulation is shown in (d).

### *C. Consensus surface*

Errors in the registration process, or inaccurate depth information returned by the range scanner, may cause slight misalignment in the surfaces that we integrate. We may therefore be faced with a situation where the Delaunay retriangulation results in a surface that zig-zags between the two original meshes. Fig. 6 clarifies.

This undesirable result can be remedied to some extent by the application of a special surface smoothing procedure. We introduce a confidence measure for every 3D point $i$, defined as

$$C_i = \left| \frac{1}{L\theta} \right|, \tag{4}$$

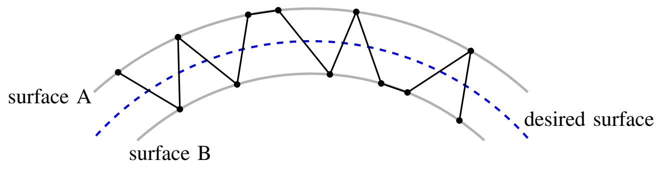

Fig. 6. A top-down view of two misaligned surfaces that were integrated. This misalignment may result in a triangulation that zig-zags between the two surfaces, and necessitates the building of some consensus surface.

where $L$ is the distance from the 3D point to the optical centre of the corresponding range sensor (the position that the sensor was in when that point was captured). The angle $\theta$ is given by

$$\theta = \arccos\left(\mathbf{n}_i \cdot \mathbf{r}_i\right), \tag{5}$$

where $\mathbf{n}_i$ is the average normal vector over the triangles surrounding vertex $i$, and $\mathbf{r}_i$ is the normalized measurement ray from the sensor's optical centre to the point. The confidence measure captures the fact that points close to the sensor, as well as surfaces close to a fronto-parallel orientation, are typically captured more accurately by range sensors.

We now apply weighted smoothing on all the vertices affected by the retriangulation in the integrated mesh. The position of each of these vertices is substituted by a weighted average of its position and the positions of all its immediate neighbours, where the weights are proportional to the confidence measures.

In doing so we obtain a consensus surface that takes into account the confidence we have in the data while, contrary to existing approaches mentioned in section II, avoiding the creation of holes that must later be filled.

## VI. Results

We now demonstrate the performance of our incremental range image integration algorithm by means of an example. It is important to stress that we do not test the accuracy of the proposed method here, as independently generated ground truth is unavailable. We rather show that the method has potential, warranting further development and testing.

A dataset of the interior of a static office environment was captured with a hand-held Kinect. Fig. 7 shows a number of colour images from this set. Range images were also captured and converted to 3D point clouds. SIFT features were found in the colour images and, as explained in section III, used to estimate sensor motion.

Meshes built from the first and second range images in the sequence are shown in Fig. 8(a) and (b), and the result from integrating these two meshes is shown in (c). We see that, particularly along the boundaries of the walls in the two separate meshes, many holes occur. These holes are filled by our integration algorithm, not by false edge extensions but because of the shift in the sampling of points brought about by the movement of the sensor. The desk, for example, is disconnected in the separate meshes but not in the integrated one. This extra filling of holes, due to the way in which

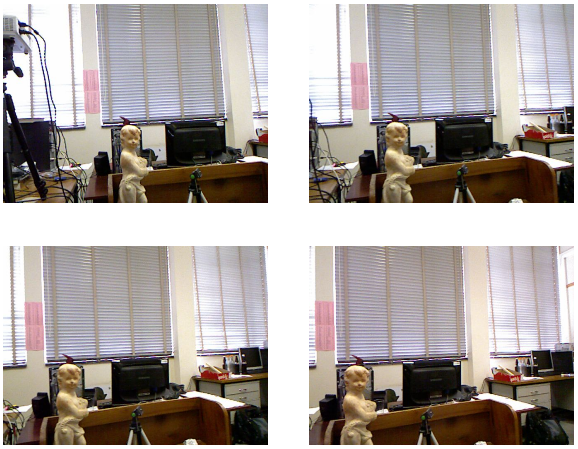

Fig. 7. A subset of the colour images of the dataset. Corresponding range images were also captured and used to evaluate our algorithm's performance.

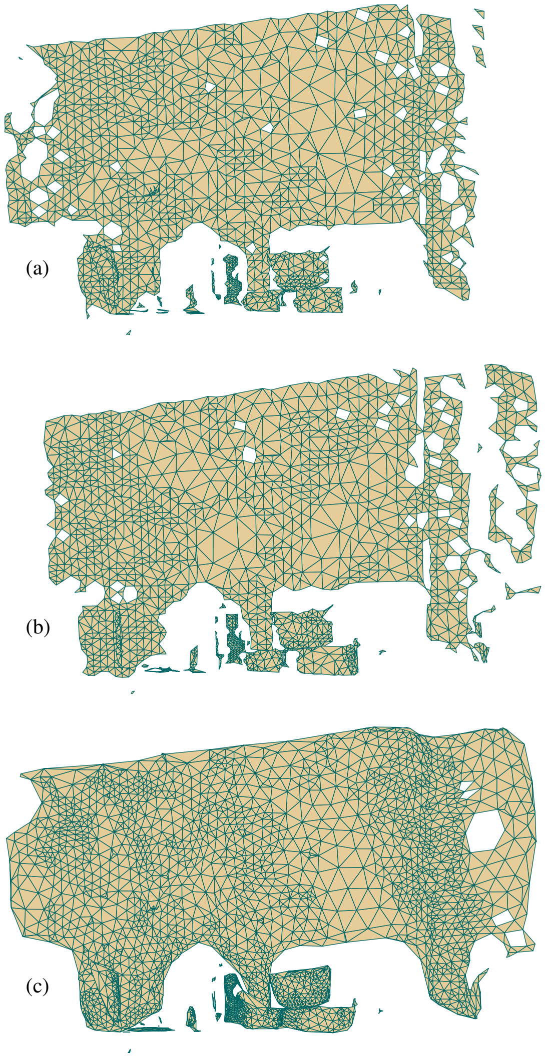

Fig. 8. The individual meshes in (a) and (b) were obtained from two range images captured by the Kinect. The result of integrating these two meshes is shown in (c).

the Delaunay triangulation is performed, is (arguably) an added advantage of our approach. Note also that no holes are introduced where the two separate meshes overlap, nor is the integrated mesh intersecting itself.

Structured light sensors typically struggle to measure strong edges accurately. This is visible, for example, around the edges of the computer screen in the meshes before integration. The integration step improves matters rather dramatically, as clear straight lines are now seen as opposed to jagged edges.

Fig. 9 shows the result of integrating 7 range images. Colours were obtained by projecting points to the available colour images.

It should be mentioned that error propagation from the incremental image registration procedure may prohibit the full functioning of our algorithm, particularly on datasets containing more images. The partial reconstructions obtained, however, appear to deliver accurate representations of the structure of observed scenes.

## VII. Conclusions and future work

We have presented a mesh-based algorithm for incrementally integrating range images. The algorithm satisfies the criteria listed in section I, which include the removal of redundant data and the incorporation of range data uncertainty (or rather, in the context of our algorithm, range data confidence measures).

Our experiments suggest that consistent surface reconstructions that preserve the underlying structure of the scene, using a relatively small number of triangles, are attainable. The algorithm also appears to handle noise well, and succeeds in combating small misalignment errors by building a consensus surface that incorporates point confidences in a final smoothing process.

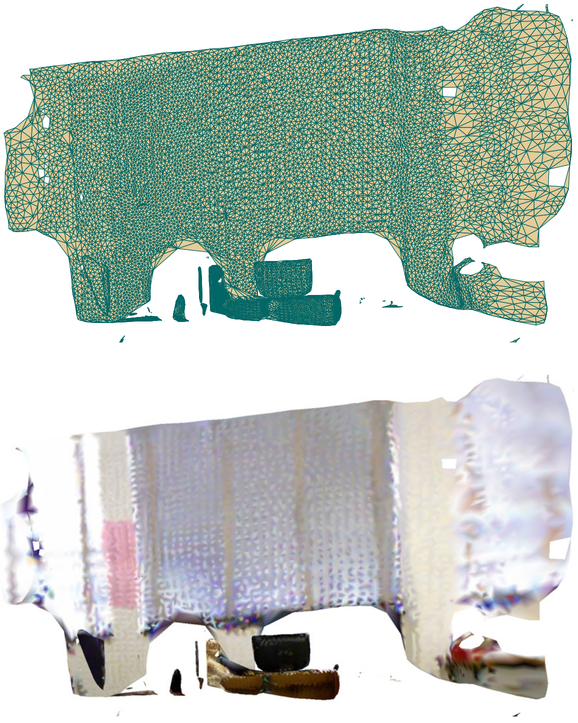

Fig. 9. This surface was obtained from incrementally integrating 7 range images. Colours were mapped by projecting vertices onto the colour images in the dataset.

Future work may include a more accurate image registration procedure where error propagation is somehow suppressed. Some possibilities here include moving away from pairwise registration and rather employ a Kalman filter-type estimator, to simultaneously integrate the range images and refine the motion parameters [19], or to apply a variant of iterative closest point (ICP) matching to refine the alignment.

Another opportunity for further investigation would be to model the uncertainty in vertex positions and the pose of the sensor more completely, which would also be beneficial for navigational and mapping purposes.

## References

[1] D. Herrera, J. Kannala, J. Heikkilä, *Accurate and practical calibration of a depth and color camera pair*, International Conference on Computer Analysis of Images and Patterns, 437–445, 2011.

[2] C. Rocchini, P. Cignoni, F. Ganovelli, C. Montani, P. Pingi, R. Scopigno, *The marching intersections algorithm for merging range images*, The Visual Computer, 20(2):149–164, 2004.

[3] T. Masuda, *Registration and integration of multiple range images by matching signed distance fields for object shape modeling*, Computer Vision and Image Understanding, 87(1):51–65, 2002.

[4] B. Curless, M. Levoy, *A volumetric method for building complex models from range images*, Proceedings of SIGGRAPH, 303–312, 1996.

[5] A. Hilton, A. Stoddart, J. Illingworth, T. Windeatt, *Reliable surface reconstruction from multiple range images*, European Conference on Computer Vision, 117–126, 1996.

[6] K. Pulli, L. Shapiro, *Surface reconstruction and display from range and color data*, Graphical Models, 62(3):165–201, 2000.

[7] T. Lu, D. Yun, *Optimizing triangular mesh generation from range images*, Three-Dimensional Image Capture and Applications, 161–171, 2000.

[8] H. Zhou, Y. Liu, L. Li, *Incremental mesh-based integration of registered range images: robust to registration error and scanning noise*, Asian Conference on Computer Vision, 958–968, 2006.

[9] Y. Sun, C. Dumont, M. Abidi, *Mesh-based integration of range and color images*, SPIE Conference on Sensor Fusion: Architectures, Algorithms and Applications, 110–117, 2000.

[10] R. Pito, *Mesh integration based on co-measurements*, IEEE International Conference on Image Processing, 397–400, 1996.

[11] G. Turk, M. Levoy, *Zippered polygon meshes from range images*, Proceedings of SIGGRAPH, 311–318, 1994.

[12] M. Rutishauser, M. Stricker, M. Trobina, *Merging range images of arbitrarily shaped objects*, IEEE Conference on Computer Vision and Pattern Recognition, 573–580, 1994.

[13] D. Lowe, *Object recognition from local scale-invariant features*, IEEE International Conference on Computer Vision, 1150–1157, 1999.

[14] W. Brink, D. Joubert, F. Singels, *Dense stereo correspondence for uncalibrated images in multiple view reconstruction*, Proceedings of the 21st Annual Symposium of PRASA, 39–44, 2010.

[15] N. Dyn, K. Hormann, S. Kim, D. Levin, *Optimizing 3D triangulations using discrete curvature analysis*, Mathematical Methods for Curves and Surfaces, 135–146, 2001.

[16] A. Shapiro, A. Tal, *Polyhedron realization for shape transformation*, The Visual Computer, 14:429–444, 1998.

[17] H. Shvaytser, *A surface matching algorithm for two perspective views*, IEEE Conference on Computer Vision and Pattern Recognition, 742–743, 1993.

[18] M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd edition, Springer-Verlag, 2008.

[19] R. Tyleček, R. Šára, *Depth map fusion with camera position refinement*, Computer Vision Winter Workshop, 59–66, 2009.