

# MRF Labeling for Multi-View Range Image Integration

Ran Song<sup>1</sup>, Yonghuai Liu<sup>1</sup>, Ralph R. Martin<sup>2</sup>, and Paul L. Rosin<sup>2</sup>

<sup>1</sup> Department of Computer Science, Aberystwyth University, UK  
`{res,yyl}@aber.ac.uk`

<sup>2</sup> School of Computer Science & Informatics, Cardiff University, UK  
`{Ralph.Martin,Paul.Rosin}@cs.cardiff.ac.uk`

**Abstract.** Multi-view range image integration focuses on producing a single reasonable 3D point cloud from multiple 2.5D range images for the reconstruction of a watertight manifold surface. However, registration errors and scanning noise usually lead to a poor integration and, as a result, the reconstructed surface cannot have topology and geometry consistent with the data source. This paper proposes a novel method cast in the framework of Markov random fields (MRF) to address the problem. We define a probabilistic description of a MRF labeling based on all input range images and then employ loopy belief propagation to solve this MRF, leading to a globally optimised integration with accurate local details. Experiments show the advantages and superiority of our MRF-based approach over existing methods.

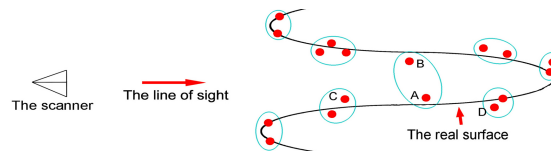
## 1 Introduction

3D surface model reconstruction from multi-view 2.5D range image has a wide range of applications in many fields such as reverse engineering, CAD, medical imagery and film industry, etc. Its goal is to estimate a manifold surface that approximates an unknown object surface using multi-view range images, each of which essentially represents a sample of points in 3D Euclidean space, combined with knowledge about the scanning resolution and the measurement confidence. These samples of points are usually described in local, system centred, coordinate systems and cannot offer a full coverage of the object surface. Therefore, to build up a complete 3D surface model, we usually need to register a set of overlapped range images into a common coordinate frame and then integrate them to fuse the redundant data contained in overlapping regions while retain enough data sufficiently representing the correct surface details. However, to achieve both is challenging due to its ad hoc nature. On the one hand, registered range images are actually 3D unstructured point clouds downgraded from original 2.5D images. On the other hand, scanning noise such as unwanted outliers and data loss typically caused by self-occlusion, large registration errors and connectivity relationship loss among sampled points in acquired data often lead to a poor reconstruction. As a result, the reconstructed surface may include holes, false connections, thick and non-smooth or over-smooth patches, and artefacts.

Hence, a good integration should be robust to registration errors and scanning noise introduced in the stages of registration and data acquisition. Once multiple registered range images have been fused into a single reasonable point cloud, many techniques [1–4] can be employed to reconstruct a watertight surface.

## 2 Related work

Existing integration methods can be classified into four categories: volumetric method, mesh-based method, point-based method and clustering-based method. The volumetric method [5–8] first divides the space around objects into voxels and then fuse the data in each voxel. But the comparative studies [9, 10] show they are time-consuming, memory-hungry and not robust to registration errors and scanning noise. The mesh-based method [11–14] first employs a step discontinuity constrained triangulation and then detects the overlapping regions between the triangular meshes derived from successive range images. Finally, it reserves the most accurate triangles in the overlapping regions and reconnects all remaining triangles subject to a certain objective function such as maximising the product of the interior angles of triangles. However, since the number of triangles is usually much larger than that of the sampled points, the mesh-based methods are computationally expensive. Thus some of the mesh-based methods just employ a 2D triangulation in the image plane to estimate the local surface connectivity in the first step as computation in a 2D sub-space is more efficient. But projection from 3D to 2D may lead to ambiguities if the projection is not injective. The mesh-based methods are thus highly likely to fail in non-flat regions where no unique projection plane exists. This speeded-up strategy cannot deal with 3D unstructured point clouds either. The point-based method [15, 16] produces a set of new points with optimised locations. Due to the neglect of local surface topology, its integration result is often over-smooth and cannot retain enough surface details. The latest clustering-based method [9, 10] employs classical clustering methods to minimise the objective dissimilarity functions. It surpasses previous methods since it is more robust to scanning noise and registration errors. Nonetheless, the clustering, which measures Euclidean distances to find closest centroids, does not consider local surface details, leading to some severe errors in non-flat areas. For instance, in Fig. 1, although point *A* is closer to point *B* and thus the clustering-based method will wrongly group them together, we would rather group *A* with *C* or *D* to maintain the surface topology.



**Fig. 1.** Local topology has a significant effect on the point clustering in non-flat areas

To overcome the drawbacks of the existing integration methods, we propose a novel MRF-based method. We first construct a network by a point shifting strategy in section 3. In section 4, the MRF labeling is then configured on this network where each node is formally labeled with an image number. In section 5, we employ loopy belief propagation (LBP) to solve the MRF and find the optimal label assignment. Each node will just select its closest point from the image with its assigned image number. These selected points, directly from the original data, are used for surface reconstruction. The integration achieves a global optimisation as the MRF labeling considers all image numbers for each node. Also, according to the experimental results shown in section 6, it effectively preserves local surface details since in the output point cloud, the points representing a local patch of the reconstructed surface are usually from the same input image due to the neighbourhood consistency of MRF. More importantly, our method can also cope with the more general input—multiple 3D unstructured point clouds.

### 3 MRF network construction for integration

MRF describes a system based on a network or a graphical model and allows many features of the system of interest to be captured by simply adding appropriate terms representing contextual dependencies into it. Once the network has been constructed, contextual dependencies can be cast in the manipulations within a neighbourhood. In a MRF network, the nodes are usually pixels. For example, in [17–20], the network is the 2D image lattice and 4 neighbouring pixels are selected to form a neighbourhood for each pixel. But it is difficult to define a network for multiple registered range images since the concept ‘pixel’ does not exist. The simplest method is to produce a point set by k-means clustering and the neighbours of a point can be found by Nearest Neighbours (NNs) algorithm. Since both k-means clustering and NNs are based only on Euclidean distance, in such a network neither the delivery of surface topological information nor the compensation for pairwise registration errors can be achieved.

In this paper, we propose a novel scheme to produce a MRF network  $I_{net}$ . Given a set of consecutive range images  $I_1, I_2, \dots, I_m$ , we first employ the pairwise registration method proposed in [21] to obtain a transform  $H_{12}$  mapping  $I_1$  into the coordinate system of  $I_2$ . The transformed image  $I'_1$  and the reference image  $I_2$  supply not only the redundant surface information (overlapping area) but also the new surface information (non-overlapping area) from their own viewpoints for the fused surface. To integrate  $I'_1$  and  $I_2$ , the overlapping area and non-overlapping area have to be accurately and efficiently detected. In the new scheme, we declare that a point from one viewpoint belongs to the overlapping area if its distance to its corresponding point from the other viewpoint is within a threshold, otherwise it belongs to the non-overlapping area. The corresponding point is found by a closest point search with k-D tree speedup. The threshold is set as  $3R$  where  $R$  is the scanning resolution of the input range images.

After the overlapping area detection, we initialise  $I_{net}$  as:

$$I_{net} = S_{non-overlap}, \quad S_{non-overlap} = S_1 + S_2 \quad (1)$$

where  $S_1$  from  $I'_1$  and  $S_2$  from  $I_2$  are the points in the non-overlapping area. Then, each point in the overlapping area is shifted by half of its distance to its corresponding point along its normal  $N$  towards its corresponding point.

$$P' = P + 0.5d \cdot N, \quad d = \vec{P} \cdot N \quad \text{and} \quad \vec{P} = P_{cor} - P \quad (2)$$

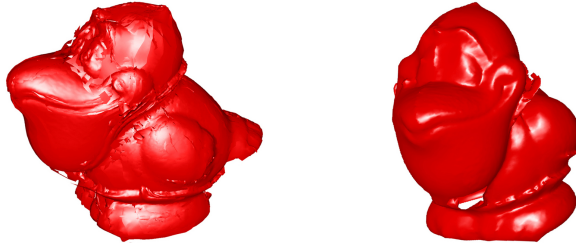
where  $P_{cor}$  is the corresponding point of  $P$ . After such a shift, the corresponding points are closer to each other. For each point shifted from a point in the overlapping region of the reference image  $I_2$ , a sphere with a radius  $r = M \times R$  ( $M$  is a parameter controlling the density of the output point cloud) is defined. If some points fall into this sphere, then their original points, without shifting, are retrieved. Their averages  $S_{overlap}$  are then computed and  $I_{net}$  is updated as:

$$I_{net} = S_{non-overlap} + S_{overlap} \quad (3)$$

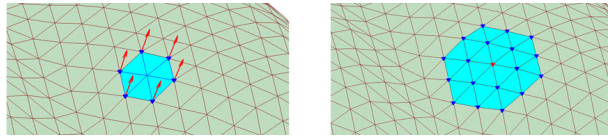
So, this point shifting (1) compensates the pairwise registration errors, (2) does not change the size of overlapping area as the operation of points is along the normal, and (3) still keeps the surface topology as the shift is along the normal.

Now, we consider the third input image  $I_3$ . We map the current  $I_{net}$  into the coordinate system of  $I_3$  through the transformation  $H_{23}$ . Similarly, we then do the overlapping area detection between  $I'_{net}$  transformed from  $I_{net}$  and the current reference image  $I_3$ . We still update  $I_{net}$  by Eq. (3). In this update,  $S_{non-overlap}$  contains the points from  $I'_{net}$  and  $I_3$  in non-overlapping regions and  $S_{overlap}$  are produced by the aforementioned point shifting strategy. We keep running this updating scheme over all input range images. The final  $I_{net}$  will be consistent with the coordinate system of the last input image  $I_m$ . Although it is not usual, if some points in  $I_{net}$  only appear in  $S_{non-overlap}$ , we delete them from  $I_{net}$ . Fig. 2 shows 18 range images of a bird mapped into a single global coordinate system and the MRF network  $I_{net}$  produced by our new scheme. In this paper, for clarity, we use surfaces instead of point clouds when we show range data. The triangulation method employed here for surface reconstruction is the ball-pivoting algorithm [1]. In Fig. 2, it can be seen that the registered images contain a massive amount of noise such as thick patches, false connections and fragments caused by poor scanning and registration errors. The point shifting scheme compensates the errors a bit whereas the resultant point set  $I_{net}$  is still noisy, leading to an over-smoothed surface without enough local details. But it offers us a roughly correct manifold for estimating the ground truth surface.

We also need to define a neighbourhood for each node in  $I_{net}$ , consistent with the MRF theory. Instead of performing a NNs algorithm to search the neighbours for a point  $i$  in such a network, we find the neighbouring triangles of  $i$  in the triangular mesh. The vertices of these triangles, excluding  $i$ , are defined as the neighbours of  $i$ . The collection of all neighbours of  $i$  is defined as the neighbourhood of  $i$ , written as  $\mathcal{N}(i)$ . In this work, to strengthen the neighbourhood consistency, we define a larger neighbourhood  $\mathcal{N}(i)$  consisting of neighbours and neighbours' neighbours as shown in Fig. 3. Doing so is allowed since the condition  $i \in \mathcal{N}(j) \Leftrightarrow j \in \mathcal{N}(i)$  is still satisfied. The superiority of this definition over the NNs method is that it reflects the surface topology and is parameter-insensitive.



**Fig. 2.** Left: 18 range images are registered (Please note that a patch from a certain range image is visible if and only if it is the outermost surface); Right: The surface representing point set  $I_{net}$  produced by point shifting is usually noisy and over-smooth.



**Fig. 3.** Left: A neighbourhood and the normalised normal vectors attached to the neighbours. Right: the expanded neighbourhood used in this work. Please note that the number of neighbours of each point in such a network can vary.

## 4 MAP-MRF labeling for integration

We denote  $s = \{1, \dots, n\}$  representing the points in  $I_{net}$  and define the label assignment  $x = \{x_1, \dots, x_n\}$  to all points as a realisation of a family of random variables defined on  $s$ . We also define the feature space that describes the points and understand it as a random observation field with its realisation  $y = \{y_1, \dots, y_n\}$ . Finally, we denote a label set  $L = \{1, \dots, m\}$ , where each label corresponds to an image number. Thus, we have  $\{x_i, y_j\} \in L, \{i, j\} \in s$ .

An optimal labeling should satisfy the *maximum a posteriori probability* (MAP) criterion [22] which requires the maximisation of the posterior probability  $P(x|y)$ . Considering the Bayes rule, this maximisation can be written as:

$$x^* = \arg \max_x (P(x|y)) = \arg \max_x (P(x)p(y|x)) \quad (4)$$

In a MRF, the prior probability of a certain value at any given point depends only upon neighbouring points. This Markov property leads to the Hammersley-Clifford theorem [22]: the prior probability  $P(x)$  satisfies a Gibbs distribution with respect to a neighbourhood  $\mathcal{N}$ , expressed as  $P(x) = Z^{-1} \times e^{-(U(x)/Q)}$ , where  $Z$  is a normalising constant and  $U(x)$  is the *priori energy*.  $Q$ , a global control parameter, is set to 1 in most computer vision and graphics problems [22].

Let  $p(y|x)$  be expressed in the exponential form  $p(y|x) = Z_y^{-1} \times e^{-U(y|x)}$ , where  $U(y|x)$  is the *likelihood energy*. Then  $P(x|y)$  is a Gibbs distribution

$$P(x|y) = Z_U^{-1} \times e^{-(U(x)+U(y|x))} \quad (5)$$

The MAP-MRF problem is then converted to a minimisation problem for the posterior energy  $U(x|y)$ :  $x^* = \arg \min_x (U(x|y)) = \arg \min_x (U(x) + U(y|x))$ .

Under the MRF assumption that the observations are mutually independent,  $U(y|x)$  can be computed by the sum of the *likelihood potentials*  $V(y_i|x_i)$  at all points. In this work, we estimate  $V(y_i|x_i)$  by a truncation function:

$$U(y|x) = \sum_{i \in s} V(y_i|x_i) = \sum_{i \in s} \sum_{y_i \in L \setminus x_i} \min(D_i(y_i, x_i), F) \quad (6)$$

where  $F$  is a constant and  $L \setminus x_i$  denotes the labels in  $L$  other than  $x_i$ , and

$$D_i(y_i, x_i) = \|C_i(y_i) - C_i(x_i)\| \quad (7)$$

is a distance function where  $C_i(l), l \in L$  denotes  $i$ 's closest point in the  $l$ th input image. In this way, we convert the estimation for the likelihood potential of different labels at point  $i$  (or the single label cost) into the measurement of the distances between  $i$ 's closest points from the input images with different labels. The truncation parameter  $F$  plays an important role here. It eliminates the effect from the input images which do not cover the area around  $i$ .

The prior energy  $U(x)$  can be expressed as the sum of *clique energies*:

$$U(x) = \sum_i V_1(x_i) + \sum_i \sum_{j \in \mathcal{N}(i)} V_2(x_i, x_j) + \sum_i \sum_{j \in \mathcal{N}(i)} \sum_{k \in \mathcal{N}(i)} V_3(x_i, x_j, x_k) + \dots \quad (8)$$

We only consider the unary clique energy and the binary one. Here, single-point clique  $V_1(x_i)$  are set to 0 as we have no preference which label should be better. We hope that in the output point cloud, neighbouring points are from the same input image so that the local surface details can be well preserved. Therefore, we utilize the neighbourhood consistency of MRF to 'discourage' the routine from assigning different labels to two neighbouring points:

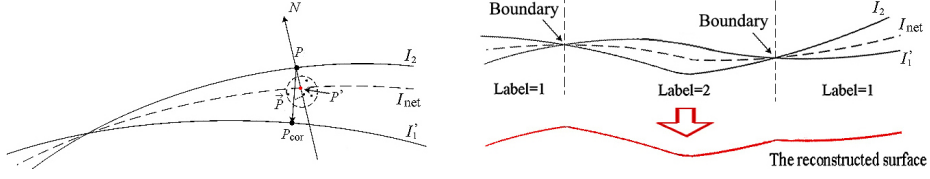
$$U(x) = \sum_i \sum_{j \in \mathcal{N}(i)} V_2(x_i, x_j) = \sum_i \sum_{j \in \mathcal{N}(i)} \lambda A_{ij}(x_i, x_j) \quad (9)$$

where  $\lambda$  is a weighting parameter and here we use the Potts model [23] to measure the clique energy (or the pairwise label cost):

$$A_{ij}(x_i, x_j) = \begin{cases} 1, & x_i \neq x_j, \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Once we find the optimal  $x^*$  which assigns a label  $x_i$  to point  $i$  by minimising the posterior energy,  $i$  will be replaced with its closest point from the input image with the number  $x_i$  ( $x_i$ th image). Please note that in this step we do not need to run a closest point search again as this point  $C_i(x_i)$  has already been searched for (see Eq. (7)). The final output of the integration is thus a single point cloud composed of points directly from different registered input range images.

The labeling is actually a segmentation for the point set  $s$ . Due to Eq. (9) and (10), neighbouring points in  $s$  tend to be labeled with the same image



**Fig. 4.** Left: The idea of point shifting to generate  $I_{net}$ . Right: The illustration of a 2-image integration based on the MAP-MRF labeling

number. The surface reconstructed from the output point cloud will consist of patches directly from different registered input images. Therefore, one advantage of the new integration method is the preservation for surface details, which avoids the error illustrated in Fig. 1. Another significant advantage is its high robustness to possible moving objects. The image patches containing moving objects are less probably selected. On the one hand, the single label cost for labeling the points in the area corresponding to a moving object with the image containing the moving object must be high in terms of Eq. (6). On the other hand, the points on the boundaries among patches always have high pairwise label costs according to Eq. (9) and (10). As a result of minimising the posterior energy, the boundaries usually bypass the areas where the moving objects appear.

Here we also add a robustness parameter  $\beta$  in order to make the algorithm more robust to noise. When we compute the single label cost  $V(y_i|x_i)$ , we always carry out the robustness judgment as below:

- if  $V(y_i|x_i) < \beta$ , Go through the routine;
- if  $V(y_i|x_i) > \beta$ , Terminate the calculation of  $V(y_i|x_i)$  and delete the point  $i$  from the list of  $s$ . And,

$$\beta = (m - q) \times F \quad (11)$$

where  $m$  is the number of input images ( $m = \max(L)$ ) and  $q$  is the number of input images that contain the roughly same noise if known. Generally, we set  $q$  as 2 so that random noise can be effectively reduced. We can also adjust  $q$  experimentally. If the reconstructed surface still has some undesirable noise,  $q$  should be increased. If there are some holes in the surface, we use a smaller  $q$ .

In summary, Fig. 4 illustrates the idea of point shifting which is the key step to construct the MRF network in section 3 and a 2-image integration based on the MAP-MRF labeling described in details in section 4.

## 5 Energy minimisation using Loopy Belief Propagation

There are several methods which can minimise the posterior energy  $U(x|y)$ . The comparative study in [24] shows that two methods, Graph-Cuts(GC) [23, 25] and LBP [26, 19], have proven to be efficient and powerful. GC is usually regarded fast and has some desirable theoretical guarantee on the optimality of the solution

it can find. And since we use the ‘metric’ Potts model to measure the difference between labels, the energy function can be minimised by GC indeed. However, in the future research, we plan to extend our technique by exploring different types of functions for the measurement of pairwise label cost such as the linear model and quadratic model [19]. Thereby, LBP becomes a natural choice for us.

Max-product and sum-product are the two variants of LBP. The algorithm proposed in [19] can efficiently reduce their computational complexity from  $O(nk^2T)$  to  $O(nkT)$  and  $O(nk \log kT)$  respectively, where  $n$  is the number of points or pixels in the image,  $k$  is the number of possible labels and  $T$  is the number of iterations. Obviously, max-product is suitable here for the minimisation. It takes  $O(k)$  time to compute each message and there are  $O(n)$  messages to be computed in each iteration. In our work, each input range image has about  $10^4$  points and typically a set of range images have 18 images. To reflect the true surface details of the object, the number of points in the output point cloud must be large enough. Thus if we use a point set as the label set, the number of labels would be tens of thousands. In that case LBP would be extremely time-consuming and too memory-hungry to be tractable. A MATLAB implementation would immediately go out of memory because of the huge  $n \times k$  matrix needed to save the messages passing through the network. But the final output of integration must be a single point cloud. So, our MRF labeling using image numbers as labels makes it feasible to employ LBP for the minimisation.

The LBP works by passing messages in a MRF network, briefed as follows:

1. For all point pairs  $(i, j) \in \mathcal{N}$ , initialising message  $m_{ij}^0$  to zero, where  $m_{ij}^t$  is a vector of dimension given by the number of possible labels  $k$  and denotes the message that point  $i$  sends to a neighbouring point  $j$  at iteration  $t$ .
2. For  $t = 1, 2, \dots, T$ , updating the messages as

$$m_{ij}^t(x_j) = \min_{x_i} \left( \lambda A_{ij}(x_i, x_j) + \sum_{y_i \in L \setminus x_i} \min(D_i(y_i, x_i), F) + \sum_{h \in \mathcal{N}(i) \setminus j} m_{hi}^{t-1}(x_i) \right) \quad (12)$$

3. After  $T$  iterations, computing a belief vector for each point,

$$b_j(x_j) = \sum_{y_j \in L \setminus x_j} \min(D_j(y_j, x_j), F) + \sum_{i \in \mathcal{N}(j)} m_{ij}^T(x_j) \quad (13)$$

and then determining labels as:

$$x_j^* = \arg \min_{x_j} (b_j(x_j)) \quad (14)$$

According to Eq. (12), the computational complexity for the calculation of the message vector is  $O(k^2)$  because we need to minimise over  $x_i$  for each choice of  $x_j$ . By a simple analysis, we can reduce it to  $O(k)$ . The detail is given as below:

1. Rewriting Eq. (12) as,

$$m_{ij}^t(x_j) = \min_{x_i} \left( \lambda A_{ij}(x_i, x_j) + g(x_i) \right) \quad (15)$$



$$\text{where } g(x_i) = \sum_{y_i \in L \setminus x_i} \min(D_i(y_i, x_i), F) + \sum_{h \in \mathcal{N}(i) \setminus j} m_{hi}^{t-1}(x_i)$$

2. Considering two cases:  $x_i = x_j$  and  $x_i \neq x_j$ .

(1) If  $x_i = x_j$ ,  $\lambda A_{ij}(x_i, x_j) = 0$ , thus  $m_{ij}^t(x_j) = g(x_j)$ .

(2) If  $x_i \neq x_j$ ,  $\lambda A_{ij}(x_i, x_j) = \lambda$ , thus  $m_{ij}^t(x_j) = \min_{x_i} g(x_i) + \lambda$

3. Synthesizing the two cases:

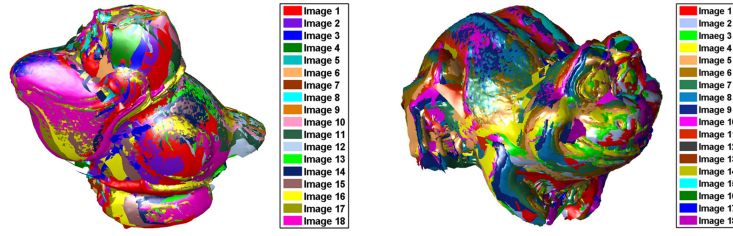
$$m_{ij}^t(x_j) = \min \left( g(x_j), \min_{x_i} g(x_i) + \lambda \right) \quad (16)$$

According to Eq. (16), the minimisation over  $x_i$  can be performed only once, independent of the value of  $x_j$ . In other words, Eq. (12) needs two nested FOR loops to compute the messages but Eq. (16) just needs two independent FOR loops. Therefore, the computational complexity is reduced from  $O(k^2)$  to  $O(k)$  and the memory demand in a typical MATLAB implementation is reduced from a  $k \times k$  matrix to a  $1 \times k$  vector.

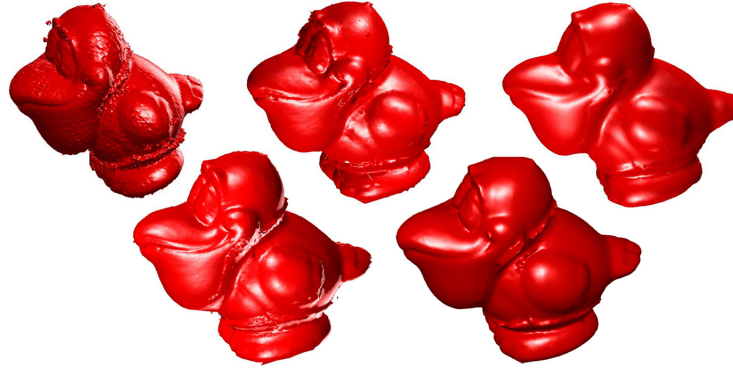
As for implementation, we use a multi-scale version to further speed up the LBP algorithm, as suggested in [19], which significantly reduces the total number of iterations needed to reach convergence or an acceptable solution. Briefly speaking, the message updating process is coarse-to-fine, *i.e.* we start LBP from the coarsest network, after a few iterations transfer the messages to a finer scale and continue LBP there, so on and so forth.

## 6 Experimental results and performance analysis

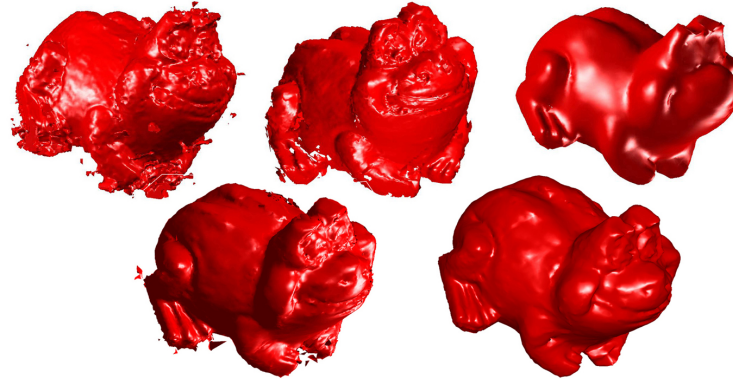
The input range images are all downloaded from the OSU Range Image Database (<http://sampl.ece.ohio-state.edu/data/3DDB/RID/index.htm>). On average, each ‘Bird’ image has 9022 points and each ‘Frog’ image has 9997 points. We employed the algorithm proposed in [21] for pairwise registration. Inevitably, the images are not accurately registered, since we found the average registration error (0.30mm for the ‘Bird’ images and 0.29mm for the ‘Frog’ images) is as high as 1/2 scanning resolution of the input data. Fig. 5 shows that the input range data are registered into the same coordinate system, rendering a noisy 3D surface model with many thick patches, false connections and fragments due to local and global registration errors. A large registration error causes corresponding points in the overlapping region to move away from each other and the overlapping area detection will thus be more difficult. The final integration is likely to be inaccurate accordingly. In our tests, the truncation parameter  $F$  is set to 4 and the weighting parameter  $\lambda$  is set to 10. Fig. 6 and 7 show the integration results produced by existing methods and our MRF-based integration, proving that our method is more robust to registration errors and thus performs best. Now we analyse some details of the new algorithm. Fig. 8 shows the labeling result. On the integrated surface, we use different colours to mark the points assigned with



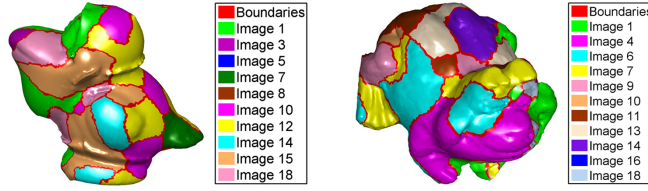
**Fig. 5.** Range images (coloured differently) are registered into the same coordinate system as the input of the integration. A patch from some range image is visible if and only if it is the outermost surface. Left: 18 ‘Bird’ images; Right: 18 ‘Frog’ images.



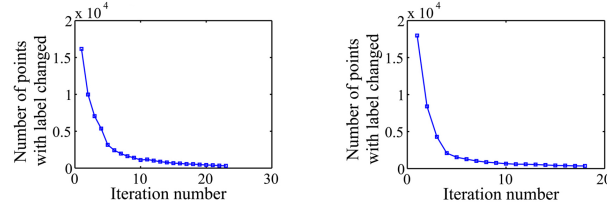
**Fig. 6.** Integration results of 18 ‘Bird’ images. Top left: volumetric method [6]. Top middle: mesh-based method [14]. Top right: fuzzy-c means clustering [10]. Bottom left: k-means clustering [9]. Bottom right: the new method.



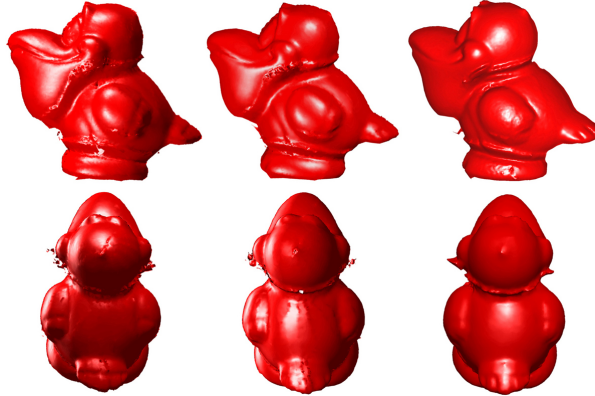
**Fig. 7.** Integration results of 18 ‘Frog’ images. Top left: volumetric method [6]. Top middle: mesh-based method [14]. Top right: fuzzy-c means clustering [10]. Bottom left: k-means clustering [9]. Bottom right: the new method.



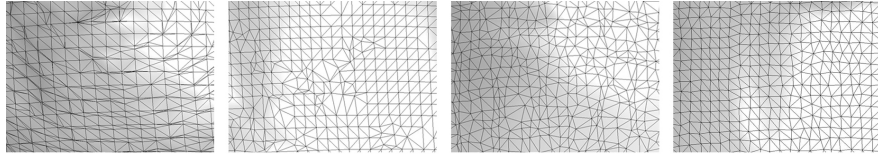
**Fig. 8.** Left: labeling result of ‘Bird’ images. Right: labeling result of ‘Frog’ images



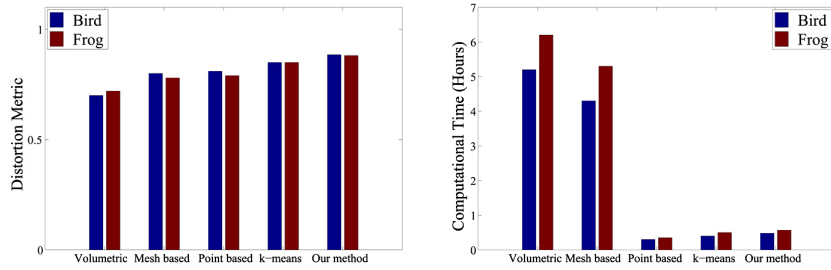
**Fig. 9.** Convergence performance of the new method using different input images. Left: The ‘Bird’ image set; Right: The ‘Frog’ image set



**Fig. 10.** Integration results of 18 ‘Bird’ images produced by different clustering algorithms. From left to right (including the top row and the bottom row): k-means clustering [9], fuzzy-c means clustering [10], the new algorithm.



**Fig. 11.** The triangular meshes of integrated surfaces using different methods. From left to right: volumetric [6], mesh-based method [14], point-based method [16], our method



**Fig. 12.** Different performance measures of integration algorithms. Left: distortion metric. Right: computational time

different labels, corresponding to different input range images. The boundary areas coloured with red actually represent the triangles where the three vertices are not assigned with the same label (Please note that the surface is essentially a triangular mesh). If a group of points are all assigned with the same label, the surface they define will be marked with the same colour. We find that not all input images have contribution to the final integrated surface. For the ‘Bird’ and the ‘Frog’ images, the reconstructed surfaces are composed of the points from 10 and 11 source images respectively. We plot in Fig. 9 how the LBP algorithm converges. To observe the convergence performance, in the LBP algorithm, we calculate Equ. (13) to determine the label assignment in each iteration of message passing. And then we count the number of points assigned with different labels from what they are assigned in last iteration. When the number is less than 2% of the total number of the initialisation points ( $I_{net}$ ), the iteration is terminated. Fig. 9 shows that the new method achieves a high computational efficiency in terms of iteration number required for convergence.

Due to the different objective functions, it is very difficult to define a uniform metric such as the integration error [9, 10] for a comparative study because to maintain local surface topology, we reject the idea that the smaller the average of the distances from the fused points to the closest points on the overlapping surfaces, the more stable and accurate the final fused surface. Thus we cannot use such a metric in this paper. Nonetheless, Fig. 10 highlights the visual difference of the integration results produced by classical clustering methods (according to Fig. 6 and 7, they are superior to other existing methods) and the new method. It can be seen that our algorithm performs better, particularly in the non-flat regions such as the neck and the tail of the bird.

Even so, for a fair comparison, we adopt some measurement parameters widely used but not relevant to the objective function: 1) The distribution of interior angles of triangles. The angle distribution shows the global optimal degree of triangles. The closer the interior angles are to  $60^\circ$ , the more similar the triangles are to equilateral ones; 2) Average distortion metric [27]: the distortion metric of a triangle is defined as its area divided by the sum of the squares of the lengths of its edges and then normalised by a factor  $2\sqrt{3}$ . The value of distortion metric is in  $[0, 1]$ . The higher the average distortion metric value, the higher the

quality of a surface; 3) The computational time; Fig. 11 and 12 show that the new method performs better in the sense of the distribution of interior angles of triangles, the distortion metric and the computational time. All experiments were done on a Pentium IV 2.40 GHz computer. Additionally, because there is no specific segmentation scheme involved, our method based on MRF labeling saves computational time compared with the techniques using some segmentation algorithms as a preprocessing before the integration [10].

## 7 Conclusion and future work

Clustering-based methods proved superior to other existing methods for integrating multi-view range images. It has, however, been shown that classical clustering methods lead to significant misclassification in non-flat areas as the local surface topology are neglected. In this paper, we develop a MRF model to describe the integration problem and solve it by LBP, producing better results as local surface details are well preserved. Since the novel MRF-based method considers all registered input images for each point belonging to the network  $I_{net}$ , the integration is globally optimised and robust to noise and registration errors. The reconstructed surface is thus geometrically realistic. Also, it is applicable to many other data sources such as 3D unstructured point clouds.

However, a couple of works can still be done to improve the integration in the future. As we mentioned above, the Potts model used in this paper is merely the simplest one. So we plan to explore different types of MRF models. We also hope to reduce the accumulated registration errors. One pairwise registration can produce one equation subject to the transform matrix. If we have more locally registered pairs than the total number of images in the sequence, the system of equations will be overdetermined. Solving this linear system of equations in a least squares sense will produce a set of global registrations with minimal deviation from the set of calculated pairwise registrations. In the future work, we will develop a scheme that can make a good balance between the extra cost caused by more pairwise registrations and the reduction of error accumulation.

*Acknowledgments.* Ran Song is supported by HEFCW/WAG on the RIVIC project. This support is gratefully acknowledged.

## References

1. Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., Taubin, G.: The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. Visual. Comput. Graph.* **5** (1999) 349–359
2. Katulakos, K., Seitz, S.: A theory of shape by space carving. *Int. J. Comput. Vision* **38** (2000) 199–218
3. Dey, T., Goswami, S.: Tight cocone: a watertight surface reconstructor. *J. Comput. Inf. Sci. Eng.* **13** (2003) 302–307
4. Mederos, B., Velho, L., de Figueiredo, L.: Smooth surface reconstruction from noisy clouds. *J. Braz. Comput. Soc.* **9** (2004) 52–66

5. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proc. SIGGRAPH. (1996) 303–312
6. Dorai, C., Wang, G.: Registration and integration of multiple object views for 3d model construction. IEEE Trans. Pattern Anal. Mach. Intell **20** (1998) 83–89
7. Rusinkiewicz, S., Hall-Holt, O., Levoy, M.: Real-time 3d model acquisition. In: Proc. SIGGRAPH. (2002) 438–446
8. Sagawa, R., Nishino, K., Ikeuchi, K.: Adaptively merging large-scale range data with reflectance properties. IEEE Trans. Pattern Anal. Mach. Intell **27** (2005) 392–405
9. Zhou, H., Liu, Y.: Accurate integration of multi-view range images using k-means clustering. Pattern Recognition **41** (2008) 152–175
10. Zhou, H., Liu, Y., Li, L., Wei, B.: A clustering approach to free form surface reconstruction from multi-view range images. Image and Vision Computing **27** (2009) 725–747
11. Hilton, A.: On reliable surface reconstruction from multiple range images. Technical report (1995)
12. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: Proc. SIGGRAPH. (1994) 311–318
13. Rutishauser, M., Stricker, M., Trobina, M.: Merging range images of arbitrarily shaped objects. In: Proc. CVPR. (1994) 573–580
14. Sun, Y., Paik, J., Koschan, A., Abidi, M.: Surface modeling using multi-view range and color images. Int. J. Comput. Aided Eng. **10** (2003) 137–50
15. Li, X., Wee, W.: Range image fusion for object reconstruction and modelling. In: Proc. First Canadian Conference on Computer and Robot Vision. (2004) 306–314
16. Zhou, H., Liu, Y.: Incremental point-based integration of registered multiple range images. In: Proc. IECON. (2005) 468–473
17. Wang, X., Wang, H.: Markov random field modelled range image segmentation. Pattern Recognition Letters **25** (2004) 367–375
18. Suliga, M., Deklerck, R., Nyssen, E.: Markov random field-based clustering applied to the segmentation of masses in digital mammograms. Computerized Medical Imaging and Graphics **32** (2008) 502–512
19. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. International Journal of Computer Vision **70** (2006) 41–54
20. Diebel, J., Thrun, S.: An application of markov random fields to range sensing. In: Proc. NIPS. (2005)
21. Liu, Y.: Automatic 3d free form shape matching using the graduated assignment algorithm. Pattern Recognition **38** (2005) 1615–1631
22. Li, S.: Markov random field modelling in computer vision. 2nd ed. Springer-Verlag (1995)
23. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell **23** (2001) 1222–1239
24. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields with smoothness-based priors. Pattern Analysis and Machine Intelligence. IEEE Trans. Pattern Anal. Mach. Intell **30** (2008) 1068–1080
25. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE Trans. Pattern Anal. Mach. Intell **26** (2004) 147–159
26. Yedidia, J., Freeman, W., Weiss, Y.: Understanding belief propagation and its generalizations. Technical Report TR-2001-22, MERL (2002)
27. Lee, C., Lo, S.: A new scheme for the generation of a graded quadrilateral mesh. Computers and Structures **52** (1994) 847–857