www.elsevier.com/locate/pr

Scan Integration as a Labeling Problem

Ran Song, Yonghuai Liu, Ralph R. Martin, Paul L. Rosin

PII: S0031-3203(14)00060-0
DOI: http://dx.doi.org/10.1016/j.patcog.2014.02.008
Reference: PR5036

To appear in: *Pattern Recognition*

Received date: 4 September 2012
Revised date: 5 December 2012
Accepted date: 17 February 2014

Scan Integration as a Labeling Problem

Ran Song[a,*], Yonghuai Liu[a], Ralph R. Martin[b], Paul L. Rosin[b]

[a]Department of Computer Science, Aberystwyth University, UK
[b]School of Computer Science and Informatics, Cardiff University, UK

Abstract

Integration is a crucial step in the reconstruction of complete 3D surface model from multiple scans. Ever-present registration errors and scanning noise make integration a nontrivial problem. In this paper, we propose a novel method for multi-view scan integration where we solve it as a labeling problem. Unlike previous methods, which have been based on various *merging* schemes, our labeling-based method is essentially a *selection* strategy. The overall surface model is composed of surface patches from selected input scans. We formulate the labeling via a higher-order Markov Random Field (MRF) which assigns a label representing an index of some input scan to every point in a base surface. Using a higher-order MRF allows us to more effectively capture spatial relations between 3D points. We employ belief propagation to infer this labeling and experimentally demonstrate that this integration approach provides significantly improved integration via both qualitative and quantitative comparisons.

Keywords: Integration, Multi-View Scans, MRF Labeling, Surface Details

[*] Corresponding Author. *Email*: res@aber.ac.uk
Department of Computer Science, Aberystwyth University, SY23 3DB, UK
Phone: +44(0)1970628762; Fax: +44(0)1970628536

1. Introduction

Reconstructing the 3D surface of an object from multi-view datasets is a classical problem in computer vision and graphics. It usually consists of two major steps: *registration* and *integration*. Registration estimates transformation parameters that align multi-view datasets in a single global coordinate system and often establishes the geometric and/or optical correspondences between overlapping datasets as a byproduct. Integration uses these registered multi-view datasets to produce a *single* dataset, which on one hand should preserve surface details, and on the other, reduce the redundant information within these overlapping datasets. For example, if the input datasets are a set of scans, we need to first align them in the same coordinate system through pairwise and global registrations and then integrate them to output a single 3D point cloud. Usually, the final step will be to triangulate this point cloud to output a watertight rendered surface. Here, albeit a nontrivial step for surface reconstruction, triangulation is out of the scope of this paper.

3D surface reconstruction from multiple datasets can be divided into *passive* and *active* methods, according to the means of data capture [1]. Passive methods typically use 2D images as input, with the advantage of low cost. However, because the 3D information used for reconstructing the 3D surface model is indirectly acquired (often through depth estimation using image-based multi-view stereo methods), it typically cannot achieve the accuracy of active 3D laser scanning. Active laser scanners have been expensive. Recently, cheap hardware has also become available which can provide depth data based on active lighting, such as the Microsoft Kinect, albeit with lower accuracy than laser scanning. Generally, the cheaper the scanner, the less accurate and more noisy the captured scans. Whichever input device is used, there remains a need for a reconstruction method which works well in the presence of data errors. To achieve this, naturally, we require both registration and integration to be robust.

Because of scanning errors, and because registration is a difficult global optimisation problem, registration errors must remain even when using state-of-the-art 3D registration algorithms [2, 3, 4, 5, 6, 7, 8, 9]. For some challenging input data, such as the range scans in the Minolta Database [10], the reported average registration error is almost as high as half of the scanning resolution of the input range scans [5, 7]. Registration errors and scanning noise make integration a difficult problem, if we are to avoid loss of details or artifacts in the integrated surface. In this paper, we focus on the robust integration of a collection of scans acquired by active laser scanners with unknown registration errors and scanning noise.

1.1. Related work

Existing methods for scan integration can be classified into four main categories: volumetric, mesh-based, clustering-based and Bayesian approaches.

Volumetric methods [11, 12, 13, 14] integrate data by voxelising them and then merging them in each voxel using data fusion algorithms. These methods require highly accurate alignment. In practice, volumetric methods work poorly or even fail due to their high sensitivity to registration errors, a problem demonstrated both theoretically and experimentally in [15, 16].

Mesh-based methods [17, 18, 19] detect overlapping regions between triangular meshes. Then, the most accurate triangles in the overlapping regions are kept, and all remaining triangles are reconnected. This is computation-

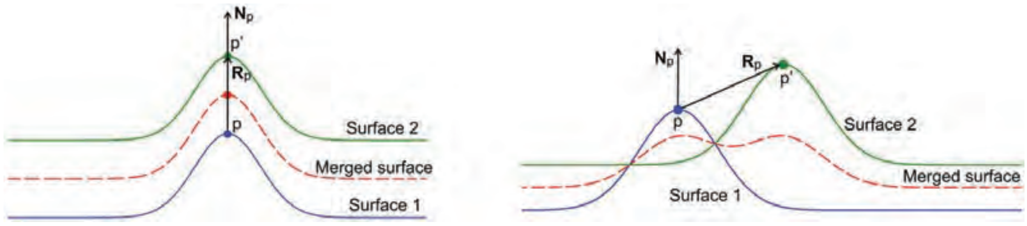

Figure 1: Registration merely seeks to minimise registration errors, but the effects on integration depend not only on the magnitude of registration error, but also its direction. Left: a registration error $\mathbf{R}_p$ in the direction of the surface normal $\mathbf{N}_p$. Right: a registration error in a different direction. A surface formed by averaging the input surfaces (a simple merging approach to integration) gives good results in the first case but not the second.

ally expensive as triangles outnumber the points and are more geometrically complex. Some mesh-based methods thus just use a 2D triangulation for efficiency, but the projection from 3D to 2D leads to ambiguities if it is not injective. Such methods can fail for highly curved regions where no suitable single projection plane exists. Mesh-based methods also lack a mechanism to cope with registration errors.

Clustering-based methods [15, 16] first initialises a set of cluster centroids. Then different clustering methods are employed to optimally find the corresponding points of these centroids and integrate them to produce new cluster centroids. This scheme is iteratively performed till convergence and finally the output points are triangulated to produce a watertight surface. Clustering-based methods are generally superior to previous methods, being more robust to scanning noise and registration errors. While Euclidean distances are used to allocate points to the closest cluster centroid, local surface geometry and neighbourhood consistency are not considered. This leads to errors in highly curved areas. Note carefully that, from the viewpoint of integration, registration errors have both a magnitude and a direction. See

Fig. 1, which shows that integration is harder to do correctly if the registration error vector and local surface normal have different directions: in the right hand example, the merged surface has incorrect local shape, as well as being over-smoothed.

Bayesian methods [20, 21, 22, 23, 24] formulate the integration problem as probabilistic models and employ local or global optimisation methods to minimise different energy functions related to integration error or some other predefined statistics. However, existing methods typically suffer from three limitations: (i) the input data must be parameterised (e.g. depth maps or 2D sweep patterns), (ii) many rely on the assumption that the distribution of the noise is known (e.g. i.i.d. Gaussian) when designing the energy function, and (iii) energy functions based only upon pairwise random fields do not adequately capture 3D information.

1.2. The proposed work

Due to inevitable registration errors and scanning noise, correct point correspondences are difficult, if not impossible, to establish. For a merging-based integration method, unfortunately, a tiny mismatch caused by registration errors can provoke strong artifacts in the integrated surface [25], in particular, in the areas of high curvature as shown in Fig. 1.

Instead of establishing correct correspondence, we select the most representative points from different input scans for the integration. From the integration point of view, we believe that the data from a single input scan is usually more accurate and reliable, than those from the integrated scans, since the integration process will unavoidably introduce more or less errors. Note that this is not in contradiction with super-resolution from multiple scans. The main target of super-resolution is to reduce the errors generated within the scanning process (i.e., scanning noise or systematic error) and typically the registration error is regarded as trivial. For instance, in [26], a model is scanned from similar but randomly perturbed viewpoints. Consequently, the transformations between scans are very small and the registration is not challenging. In [27], the transformation is fully controlled, which also eases the problem of registration since the transformation parameters are entirely or partly known. For super-resolution methods, as mentioned in [27], high-quality registration is important for preserving sharp edges. However, in integration, although scanning noise is inevitable, the major concern is large registration errors. As illustrated by the right figure in Fig. 1, sharp edges cannot be easily preserved when the registration error is not trivial. In this case, the local geometry is more reliably represented by a single scan than by the merged one. In addition, for scanning control purposes and some other applications, it is always quite rewarding to be able to see exactly what has been scanned [25]. In [25], a single scan is used as the ground truth for visual comparison.

Given a set of registered scans, there are usually several raw points representing the same ground truth position on the object surface due to the fact of overlaps. The proposed integration method is designed to select the best raw point and put it in the final integration to represent this position. In other words, we try to solve such a labeling problem: which candidate raw point should be used to label this position?

The labeling is formulated using a higher-order Markov Random Field (MRF). We first develop an iterative scheme to produce a graph composed

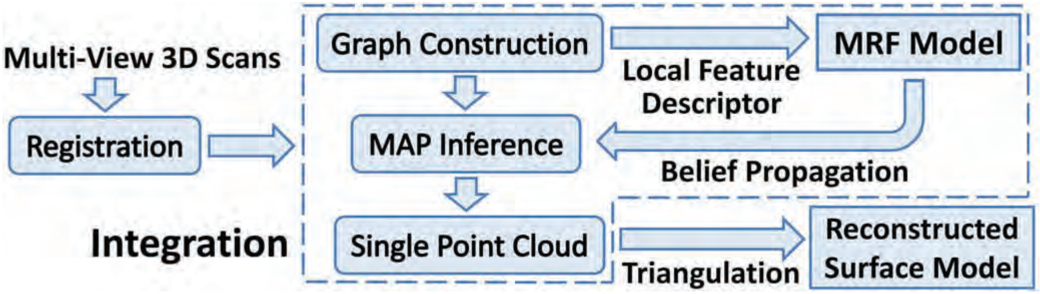

Figure 2: The workflow of the proposed method

of a set of base positions as well as the edges connecting them. This graph enables the following MRF labeling process.

Then, intuitively, we should use a point set as the label set which corresponds to all possible states (candidate raw points) of each base position. However, doing so will make the inference of the labeling intractable (see Section 3). Instead, in our method, the label set is a set of scan indices. For each base position, we find its closest point from the input scan with a certain index. Such a closest point acts as a candidate raw point in the labeling. Using scan indices as labels also facilitates the design of the compatibility term of the energy function. If the base positions in a neighbourhood are all labeled with the same scan index, they will be represented by a set of raw points from the same input scan in the output point cloud. Local surface details within this neighbourhood can thus be well preserved.

We employ belief propagation (BP) to infer the labeling which assign a label (scan index) to each base position. In this step, we also utilise the sparseness of the label set to reduce the computational complexity.

Fig. 2 illustrates the workflow of our integration method, which we explain in the rest of this paper. In Section 2, we first produce a graph by point

s	the set of base positions
i	the index of a base position
L	the set of labels
x_i	a label corresponding to the base position i
$C_i(x_i)$	the closest point in the input scan with index x_i to i
m	the number of input scans (the number of labels)
n	the number of base positions
$\mathcal{N}(i)$	the neighbourhood of i
P	a point set
$\mathbf{P}$	a 3D point vector
$\mathbf{N}$	a normal vector
F	the truncation parameter
T	the iteration number of belief propagation
m_{ji}	the message that i receives from j in a two-point clique
m_{lkji}	the message that i receives from j, k and l in a four-point clique

Table 1: List of some frequently-used notations

shifting and triangulation, and then build a higher-order MRF on this graph. Section 3 discusses the BP algorithm. We give and evaluate experimental results in Section 4, followed by conclusion and discussion in Section 5.

2. Labeling using a higher-order MRF

To pose scan integration as a labeling problem, let $s = \{1, \ldots, n\}$ be used to index base positions. We define a label assignment $x = \{x_1, \ldots, x_n\}$ to all positions. We also denote a label set $L = \{1, \ldots, m\}$, where each label corresponds to the index of a particular input scan (we assume that the input scans have already been registered at this stage). Thus $x_i \in L$, $i \in s$. The basic idea is to employ an MRF to find an optimal label assignment as illustrated in Fig. 3. Table 1 lists the notations frequently used in this paper.

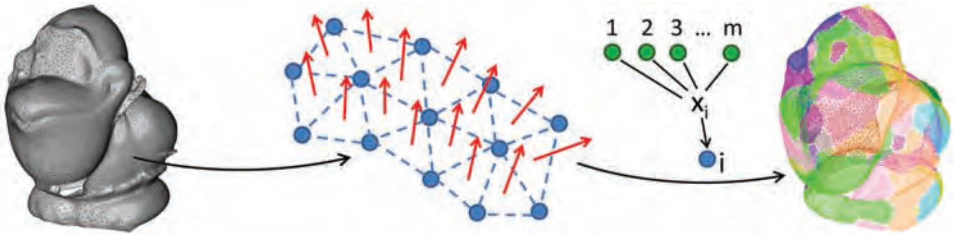

Figure 3: Label assignment. Left: A mesh (a base surface composed of base positions and edges connecting them) used as the graph for MRF labeling. Middle: A two-point neighbourhood relation (dashed) is defined between positions (blue dots) in the graph, and normals (red arrows) of the triangular facets are computed for calculating the cost of a four-point clique. Right: The output integration is composed of points from different input scans where we show them in different colours. In the labeling, each point i can take one of m different labels (given m input scans).

Our labeling model includes three terms operating on one-point, two-point and four-point cliques respectively. The first term $E_i(x_i)$ measures how much a point is in agreement with being assigned a particular label. The second term is modeled through a symmetric neighbourhood relation $\mathcal{N}$ and a compatibility cost function $E_{ij}(x_i, x_j)$. This term measures how compatible the label assignments of any two neighbouring points are. The third, higher-order term, has cost $E_c(x_c)$ where a clique $c \in \mathcal{C}$ contains four neighbouring points. It is used to capture spatial relations between 3D points in a larger, asymmetric, neighbourhood. The cost of other types of cliques is set to zero. The reason that we do not consider larger cliques in our computational model is that the complexity of MRF is very sensitive to the clique size (see Section 3.3 for details). The posterior energy $E(x)$ of such an MRF is thus given by

$$E(x) = \sum_{i \in s} E_i(x_i) + \lambda_1 \sum_{(i,j) \in \mathcal{N}} E_{ij}(x_i, x_j) + \lambda_2 \sum_{c \in \mathcal{C}} E_c(x_c) \qquad (1)$$

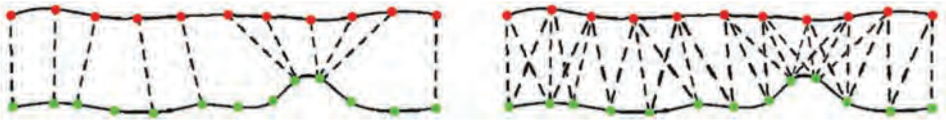

Figure 4: The routines to find corresponding points for base positions where the red points denote the base positions and the green points denote the raw points of the input scan. Left: Just finding the closest points might result in some holes or artifacts in the integration as some green points are not selected. Right: If we find the 3 closest points for each base position, the method will be more robust and probably avoid holes and artifacts.

where λ_1 and λ_2 are weighting parameters.

If a group of base positions $p = \{i, j, ...\}$ in a neighbourhood are assigned the same label x', we use the following routine to find their corresponding points from the input scan with index x'. For each point in p, we find its k closest points in the input scan with index x'. Then we put all these closest points together to produce a point set. And we delete the duplicate points from this set. The remaining points are the corresponding points of the points in p. We usually set $k = 3$. This routine is a robust strategy which avoids holes or artifacts in the integration as illustrated in Fig. 4. In the integrated surface, the positions in p are replaced with their corresponding points. Therefore, the final output of the integration is a single point cloud completely composed of some selected points from the input scans.

The following subsections discuss how to define the graph which offers two-point and four-point neigbhourhood relations, as well as the three terms of the energy function in detail.

2.1. Graph construction for labeling

In this work, we do not require that input scans must be parameterised since we intend to generalise our method to input datasets with various modalities. [15] proposes a two-view point shifting method to produce a collection of points for initialising the k-means clustering. We extend this method to multiple views and produce a collection of points representing a base surface used as a graph for labeling.

1) Overlapping area detection Given a set of consecutive scans $P_1, P_2, \ldots, P_m$, we employ the pairwise registration method from [5] to map P_1 into the coordinate system of P_2. To integrate the transformed point cloud P_1' and the reference point cloud P_2, the overlapping and non-overlapping areas of each have to be detected. To do so, a point in one point cloud is deemed to belong to the overlapping area if its distance to the nearest point in the other point cloud (its corresponding point) is within a threshold; otherwise it belongs to the non-overlapping area. The threshold is set to $3R$, where R is the scanning resolution of the input scans. Such a threshold is generally large enough not to miss any real correspondence between the overlapping scans to be integrated when their registration is reasonably accurate [15].

2) Iterative merging After detecting the overlap, we set S_1 and S_2 to the points in the non-overlapping areas belonging to P_1' and P_2 respectively, and initialise the point set P in the graph as:

$$P = S_{\mathrm{non-overlap}} = S_1 + S_2. \tag{2}$$

Next, we employ a point shifting strategy to compute a point set S_{overlap}. To bring the corresponding points closer to each other, each point $\mathbf{P}$ in both overlapping areas is shifted along its normal $\mathbf{N}$ towards its corresponding point $\mathbf{P}^*$ by half of its distance to $\mathbf{P}^*$:

$$\mathbf{P} \to \mathbf{P} + 0.5\mathbf{d} \cdot \mathbf{N}, \quad \mathbf{d} = \Delta\mathbf{P} \cdot \mathbf{N}, \quad \Delta\mathbf{P} = \mathbf{P}^* - \mathbf{P} \tag{3}$$

A sphere with radius $r = 1.5R$ is defined, centered at each such shifted point of the reference point cloud P_2. If other points fall into this sphere, then their original unshifted points are retrieved. The average position of these unshifted points is then computed and returned to form the point set S_{overlap}. Then the point set P is updated as:

$$P = S_{\text{non-overlap}} + S_{\text{overlap}} \tag{4}$$

This strategy (i) compensates for pairwise registration errors as corresponding points are closer to each other and (ii) leaves the surface topology unaffected, as the shift is along the normal.

We now consider the third input point cloud P_3. We map P into the coordinate system of P_3 and detect the overlap between P' transformed from P and the current reference point cloud P_3. We then update P based on Eq. (4). In this update, $S_{\text{non-overlap}}$ contains the points from P' and P_3 in non-overlapping areas and S_{overlap} is produced by the point shifting strategy. We iteratively apply this updating scheme to all input scans.

3) Triangulation Finally, we triangulate P to construct the graph using the power crust method [28]. Fig. 5 shows an example of the graph construction for two registered scans. The constructed graph, the base surface is oversmoothed and the more scans, the smoother the base surface (see Fig. 3). But the following formulation of the MRF energy function in Eq. (1)

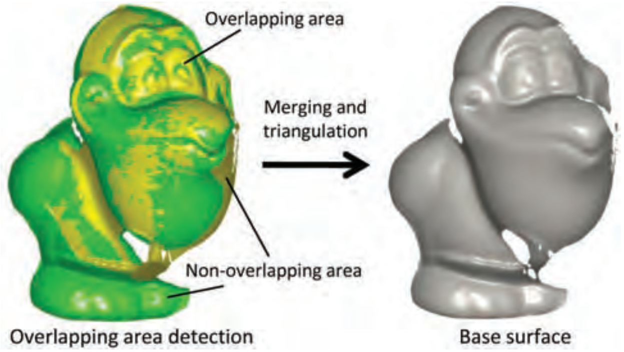

Figure 5: An illustration of graph construction for two registered scans

is based on it. The vertices of the neighbouring triangles of a point i in the graph are denoted as $\mathcal{N}(i)$. This approach has advantages over defining neighbourhoods using the k-nearest neighbours method: it facilitates the computation of normals and does not need an estimate for k.

2.2. One-point clique cost

We measure the one-point clique cost, or the data term of the MRF energy function, in order to select the point which best represents the surface locally:

$$E_i(x_i) = \sum_{y_i \in L \setminus x_i} \min(D_i(y_i, x_i), F) \tag{5}$$

where F is a truncation parameter and $L \setminus x_i$ denotes the members of the label set L other than the label x_i. And,

$$D_i(y_i, x_i) = \|C_i(y_i) - C_i(x_i)\| \tag{6}$$

is a distance function where $C_i(l)$ denotes the closest point in the input scan with index l to the base position i. This converts the one-point clique cost of different labels at i into a measurement of the distances between i's closest

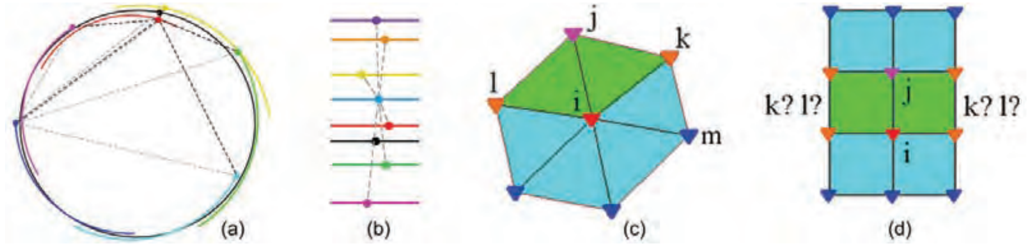

Figure 6: (a) & (b) illustrate the one-point clique cost where different colours represent different scans. (a) The truncation parameter can distinguish the scans covering the area around i (the black point) from the ones not covering it. (b) The sum of the lengths of the gray dash lines is the one-point clique cost of the blue scan for the base position i (the black point). (c) A four-point clique (i, j, k, l) uniquely determined by the point pair (i, j) in our graph. These four points determine the change in normals. (d) Two points cannot not uniquely determine a four-point clique in a lattice graph as the positions of the other two clique members k and l are not unique.

points in different input scans. F eliminates the effects of input scans which do not cover the area around i. The idea is illustrated in Fig. 6(a) and (b).

In Fig. 6(a), the black circle represents the base surface. 6 arcs in different colours represent 6 input scans. Due to registration errors, adjacent scans are not superimposed. After finding the 6 closest points of the black point (the base position i) from the 6 scans separately, it can be seen that only the red and the yellow ones cover the area around i. From Eq. (5), the cost of labeling i as red is $D_i(\text{red}, \text{yellow}) + 4F$ (the sum of the truncated lengths of the black dash lines). The cost of labeling it as blue is $5F$ (note the gray dashed lines). Hence, by setting F significantly larger than $D_i(\text{red}, \text{yellow})$, we can distinguish the scans covering the area around i from the ones not covering it. Generally, we have more than 6 input scans and several scans may cover the area around i. For these scans, the one-point clique cost function tends to select the one with the lowest summed distance to all the others as illustrated in Fig. 6(b) where for the point i (the black point), labeling it as blue produces the lowest cost.

2.3. Two-point clique cost

The two-point clique cost considers pairs of neighbouring points $\{(i, j) \in \mathcal{N}\}$. This two-point term is used to strengthen consistency of labeling of neighbouring points, meaning that they are more likely to be represented in the output by points from the same input scan. This increases the likelihood that the integrated surface is made up of surface patches each taken from a single input scan. The aim is to *select* pieces of 'good input' as the output. This is different from most existing techniques based on volumetric merging, mesh fusion or point clustering, where the output is formed as a weighted *merging* of the inputs. The two-point clique cost function is defined as below to measure the label compatibility of two neighbouring points:

$$E_{ij}(x_i, x_j) = \begin{cases} 1 & x_i \neq x_j \\ 0 & \text{otherwise} \end{cases} . \tag{7}$$

2.4. Higher-order clique cost

The two-point clique cost in Eq. (7) often directly achieves good results in 2D vision applications. However, for 3D problems, richer spatial information is needed to represent surface details accurately. Pairwise MRFs cannot capture such rich statistics, and typically lead to an oversmooth integrated surface (see Fig. 12). Hence, we need a higher-order prior term which better describes local surface geometry to enhance features such as edges. We also need to make sure that the higher-order MRF is tractable as the number of points involved in surface integration is usually quite large.

We take into account how normals vary across the surface: At least three points are needed for a normal estimate, so four points are needed to determine changes in normals. While usually a larger number of neighbouring points are required for accurate normal estimate, we only consider three points here for efficiency. Too large clique will make the MRF intractable (see Section 3.3 for details). In many cases, minimising the higher-order term based on the rough normal estimate is enough to rule out some large errors (i.e., artefacts). As the MRF is based on an undirected graph, we simply have $\{k, l\} = \{l, k\} = \mathcal{N}(i) \bigcap \mathcal{N}(j)$ where the adjacent points i, j, k, l compose a four-point clique as shown in Fig. 6(c). The sum of the higher order term in Eq. (1) can be computed as

$$\sum_{c \in \mathcal{C}} E_c(x_c) = \sum_i \sum_{j \in \mathcal{N}(i)} E_{ijkl}(x_i, x_j, x_k, x_l) \tag{8}$$

Please note that if the graph is a general regular lattice, $\mathcal{N}(i) \bigcap \mathcal{N}(j)$ will be a set of four points rather than a unique point pair (Fig. 6(d)). In that case, we have to sum over i, j, k and l (rather than just i and j in Eq. (8)) for computing the four-point clique energy. The proposed MRF graph thus elegantly reduces the computational cost of the algorithm. And,

$$E_{ijkl}(x_i, x_j, x_k, x_l) = \|\mathbf{N} - \mathbf{N}'\|_2 = \sqrt{(\mathbf{N} - \mathbf{N}')^T \cdot (\mathbf{N} - \mathbf{N}')} \tag{9}$$

where $\mathbf{N}$ and $\mathbf{N}'$ are adjacent normals, given by

$$\mathbf{N} = \frac{(\mathbf{P}_j - \mathbf{P}_i) \times (\mathbf{P}_k - \mathbf{P}_i)}{\|(\mathbf{P}_j - \mathbf{P}_i) \times (\mathbf{P}_k - \mathbf{P}_i)\|}, \quad \mathbf{N}' = \frac{(\mathbf{P}_j - \mathbf{P}_i) \times (\mathbf{P}_l - \mathbf{P}_i)}{\|(\mathbf{P}_j - \mathbf{P}_i) \times (\mathbf{P}_l - \mathbf{P}_i)\|} \tag{10}$$

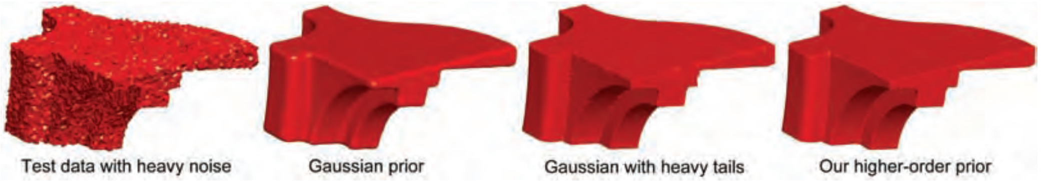

Figure 7: A comparison of different priors

and $\mathbf{P}_i = C_i(x_i), \mathbf{P}_j = C_j(x_j), \mathbf{P}_k = C_k(x_k), \mathbf{P}_l = C_l(x_l)$.

Feature retention arises because grows sublinearly in its arguments. So it tends not to distribute curvature over multiple vertices, as does a quadratic cost function. Fig. 7 illustrates a comparison for different prior terms using a noisy test model. It can be seen that the Gaussian prior, widely used in Bayesian surface reconstruction methods, oversmooths the edges and fails to retain fine features. The Gaussian with heavy tails tends to better retain the features but the reconstructed edges are often geometrically flawed. The higher-order prior achieves the best reconstruction result.

2.5. Robust strategy

We also use a threshold β in order to make the algorithm more robust to noisy points and clutters. It is essentially a voting process. A base position must receive votes from enough input scans to demonstrate that it does not correspond to a noisy point or is not in an area belonging to a clutter.

When we compute the one-point clique cost $E_i(x_i)$, we do the following. If $E_i(x_i) < \beta$, we go through the MRF labeling process, but otherwise, we terminate the calculation of $E_i(x_i)$ and delete the point i from the list s. We set $\beta = (m - q) \times F$ where m is the number of input scans (as well as the number of labels) and F is the truncation parameter defined in Section 2.2. Generally, we set $q = 2$ which is effective in reducing random noisy points

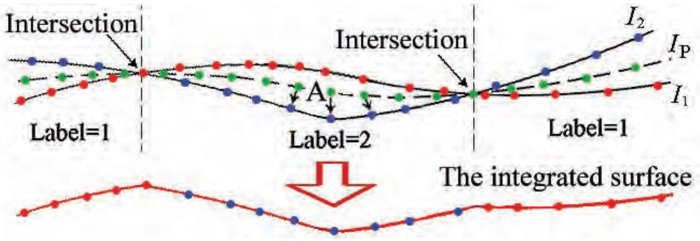

Figure 8: A two-view integration based on the labeling where I_1 and I_2 denote the two input scans with index 1 and 2 respectively. I_P represents the base surface.

and clutters; but it can be changed by the user. If the integrated surface has undesirable noise, q should be increased; if it has holes, q should be reduced.

2.6. Illustration and summary of the labeling

Fig. 8 illustrates the labeling for a two-view integration. I_1 and I_2 are already registered but not superimposed in the overlapping area due to registration errors. Point A (a base position) is closer to a red point in I_1 (Label=1) but is still represented by a blue point in I_2 (Label=2) in the integrated surface as the two-point compatibility term encourages neighbourhood consistency. A clustering method would tend to lead to an integrated surface which zigzags frequently between red and blue points, which is inconsistent with the real geometry here (both I_1' and I_2 are smooth surfaces). As can be seen, in a relatively smooth surface area, label changes usually occur where the input scans intersect (or are well-registered). This is because, here, the one-point clique cost is small according to Eqs. (5) and (6), allowing the MRF routine to tolerate larger two-point clique cost caused by the label change. But in feature-rich areas of high curvature, scans may frequently intersect each other due to registration errors. In such cases, an MRF labeling based only on one-point and two-point terms is usually unable to deliver the correct surface details, destroying the geometry of features, and leading to an oversmooth surface. In such areas, normals usually vary greatly, the higher-order term $\|\mathbf{N} - \mathbf{N}'\|_2$ in Eq.(9) will be large, providing feature retention. On the other hand, in a flatter area, this term is small and will have little effect.

3. Energy minimisation via Belief Propagation

Several methods exist for minimising posterior energy. The comparative study in [29] recommends two approaches, graph-cuts (GC) [30] and message passing, e.g. belief propagation (BP) [31], as efficient and powerful. Since our energy function is neither metric nor semi-metric, GC is not applicable.

We employ max-product (equivalent to min-sum) BP to find a solution. For pairwise MRF, its computational complexity is $O(nm^2T)$, where n is the number of points in the graph, m is the number of labels and T is the number of iterations. If we used a point set as the label set, the number of labels would be too large and the BP would be intractable. Instead, our labeling uses scan indices as labels making it feasible to employ BP. BP operates by passing messages between points. Because the two-point belief is independent of the four-point belief, each iteration uses two types of message updates.

3.1. Two-point message passing

The first type of message $m_{ji}(x_i)$ is sent from a point j to its neighbour i in a two-point clique:

$$m_{ji}(x_i) = \min_{x_j} (\lambda_1 E_{ji}(x_j, x_i) + g(x_j)) \tag{11}$$

where $g(x_j) = E_j(x_j) + \sum_{h \in \mathcal{N}(j) \setminus i} m_{hj}(x_j)$. Then we consider two cases:
(1) If $x_j = x_i$, $E_{ji}(x_j, x_i) = 0$, so $m_{ji}(x_i) = g(x_i)$.

(2) otherwise, $E_{ji}(x_j, x_i) = 1$, so $m_{ji}(x_i) = \min_{x_j} g(x_j) + \lambda_1$.
These can be combined to give:

$$m_{ji}(x_i) = \min\,(g(x_i), \min_{x_j} g(x_j) + \lambda_1). \tag{12}$$

Eq.(12) shows that the minimisation over x_j need to be performed only once, independent of the value of x_i. In other words, Eq.(11) needs two nested loops to determine the messages while Eq.(12) just needs two independent loops, reducing the computational cost from $O(m^2)$ to $O(m)$.

3.2. Four-point message passing

Note that our anisotropic MRF graph with four-point cliques is different from an isotropic MRF graph with 2×2 cliques, which can easily be converted to a factor graph [32]. This is because the four points in one clique play unequal roles in message passing. As shown in Fig. 6(c), i and j are directly connected while there is no edge linking k and l. Thus, i, j, k, l compose a four-point clique for i and j but not for k and l.

The four-point message sent to a point i is written as $m_{lkji}(x_i)$ with $\{l, k, j\} \in N_t(i)$, where $N_t(i)$ denotes the point trifold set in which each point trifold forms a four-point clique with i as shown in Fig. 6(c). Let $b_i(x_i)$ denote the one-point belief and $b_{lkji}(x_i, x_j, x_k, x_l)$ denote the four-point belief. Then energy associated with the four-point cliques (we write it as E_{lkji} to distinguish it from the four-point clique cost E_{ijkl}[1]) can be defined as

[1] Please note that most of the subscripts used in this paper are not just placeholders but do have an order. For example, m_{ji} denotes the message sent from j to i and a four-point clique cost E_{ijkl} implies that i and j are directly linked by an edge but k and l are not. Here, such a notation will not cause any confusion or conflict. If E_{ijkl} is valid for representing a four-point clique cost, E_{lkji} must be invalid for that because there is no

$$E_{lkji} = E_i(x_i) + E_j(x_j) + E_k(x_k) + E_l(x_l) + \lambda_2 E_{ijkl}(x_i, x_j, x_k, x_l), \quad (13)$$

and the Bethe free energy [33] is

$$\begin{aligned} E &= \sum_{ijkl} \sum_{x_i x_j x_k x_l} e^{-b_{lkji}(x_i,x_j,x_k,x_l)} (E_{lkji}(x_i, x_j, x_k, x_l) \\ &\quad -b_{ljki}(x_i, x_j, x_k, x_l)) - \sum (q_i - 1) \sum_{x_i} e^{-b_i(x_i)} (E_i(x_i) - b_i(x_i)) \end{aligned}$$

where q_i is the number of points neighbouring i. Therefore the Lagrangian multipliers that enforce the normalisation constraints are:

$$r_{lkji} : \sum_{x_i x_j x_k x_l} e^{-b_{lkji}(x_i,x_j,x_k,x_l)} - 1 = 0, \quad r_i : \sum_{x_i} e^{b_i(x_i)} - 1 = 0.$$

The multiplier that enforces the max-marginalisation constraints is

$$\lambda_{lkji}(x_i) : e^{-b_i(x_i)} = \max_{x_l x_k x_j} e^{-b_{lkji}(x_i,x_j,x_k,x_l)}.$$

The Lagrangian L is the summation of the E and the multiplier terms. To maximise L, we set

$$\begin{aligned} &\frac{\partial L}{\partial e^{-b_{lkji}(x_i,x_j,x_l,x_k)}} = 0, \quad \text{and hence} \\ &-b_{lkji}(x_i, x_j, x_l, x_k) = E_{lkji}(x_i, x_j, x_k, x_l) + 1 + \lambda_{lkji}(x_i) + \lambda_{ilkj}(x_j) \\ &\qquad\qquad + \lambda_{jilk}(x_k) + \lambda_{kjil}(x_l) + r_{lkji}, \\ &\frac{\partial L}{\partial e^{-b_i(x_i)}} = 0, \quad \text{and hence} \\ &-b_i(x_i) = -E_i(x_i) + \frac{1}{q_i - 1} \sum_{(j,k,l) \in N_t(i)} \lambda_{lkji}(x_i) + r'_i, . \end{aligned}$$

where r'_i is the rearranged constant. By change of variable, defining

edge linking l and k (see Fig. 6(c)).

$$\lambda_{lkji}(x_i) = -\sum_{(h,g,f)\in N_t(i)\backslash(l,k,j)} m_{hgfi}(x_i), \tag{14}$$

we obtain the following equations

$$b_i(x_i) = E_i(x_i) + \sum_{(l,k,j)\in N_t(i)} m_{lkji}(x_i); \tag{15}$$

$$\begin{aligned} b_{lkji}(x_i, x_j, x_l, x_k) \;=\; & E_i(x_i) + E_j(x_j) + E_k(x_k) + E_l(x_l) + \lambda_2 E_{ijkl}(x_i, x_j, x_k, x_l) \\ & + \sum_{(h,g,f)\in N_t(i)\backslash(l,k,j)} m_{hgfi}(x_i) + \sum_{(h,g,f)\in N_t(j)\backslash(i,l,k)} m_{hgfj}(x_j) \\ & + \sum_{(h,g,f)\in N_t(k)\backslash(j,i,l)} m_{hgfk}(x_k) + \sum_{(h,g,f)\in N_t(l)\backslash(k,j,i)} m_{hgfl}(x_l). \end{aligned} \tag{16}$$

We now consider the following three constraints:

(i) the min-sum constraint arising from the max-marginalisation (max-product) constraint: $b_i(x_i) = \min_{x_l x_k x_j} b_{lkji}(x_i, x_j, x_k, x_l)$,

(ii) that the terms $V(y_i|x_i)$, $\sum_{(l,k,j)\in N_t(i)} m_{lkji}(x_i)$ and $\sum_{(h,g,f)\in N_t(i)\backslash(l,k,j)} m_{hgfi}(x_i)$ are not related to the values of x_j, x_k and x_l, and

(iii) $N_t(k)\backslash(j,i,l) = N_t(k)$ and $N_t(l)\backslash(k,j,i) = N_t(l)$ as the point trifold $(j,i,l) \notin N_t(k)$ (for comparison, $(j,i,m) \in N_t(k)$ in Fig. 6(c)) and $(k,j,i) \notin N_t(l)$. This leads to

$$\begin{aligned} m_{lkji}(x_i) = \min_{x_l x_k x_j} \Big(& E_j(x_j) + E_k(x_k) + E_l(x_l) + \lambda_2 E_{ijkl}(x_i, x_j, x_k, x_l) \\ & + \sum_{(h,g,f)\in N_t(j)\backslash(i,l,k)} m_{hgfj}(x_j) + \sum_{(h,g,f)\in N_t(k)} m_{hgfk}(x_k) + \sum_{(h,g,f)\in N_t(l)} m_{hgfl}(x_l) \Big). \end{aligned} \tag{17}$$

3.3. Label assignment and complexity analysis

All entries in the messages are initialised to zero. We update the two kinds of message in each iteration, and after T iterations, a belief vector is computed for each point in the graph:

$$B_i(x_i) = E_i(x_i) + \sum_{j \in \mathcal{N}(i)} m_{ji}(x_i) + \sum_{(l,j,k) \in N_t(i)} m_{lkji}(x_i) \tag{18}$$

The label x_i^* that minimises $B_i(x_i)$ individually at each point is selected. In most of our experiments, the BP converged within 10 iterations.

Computing m_{lkji} is extremely costly: $O(m^4)$, as the message vector has m elements which are computed by minimising Eq. (17) over 3 variables each of which has m possible states (labels). Most existing efficient higher-order MRF optimisation methods are just applicable to specific families of energy functions or specific (small) cliques such as the P^n model [34], quadratic functions [35] truncated functions [36] and 2×2 cliques [32], etc. None of these methods work for the complicated structure of our energy function.

Instead, we utilise sparseness of the higher-order function to reduce the costs. In [32], the authors discretized the label set for three of the four member pixels into h bins and only considered those h^3 different combinations, decreasing the complexity for one message update to $O(m \cdot h^3)$. Here, we record the three labels that minimise the two-point message m_{ji} at each point i. When we calculate the four-point message m_{lkji} in the same iteration, we just consider these three labels as possible states for each member point l, k, j. As we place more importance on two-point neighbourhood consistency, the four-point cliques becomes 'sparse' (i.e. many labelings of the four-point cliques are unlikely) and the cost for each four-point message update is reduced to $O(27m)$, and the cost for the whole BP is $O(n(m + 27m)T)$=$O(28nmT)$. Clearly, such an algorithm is still intractable if both n and m are large. As

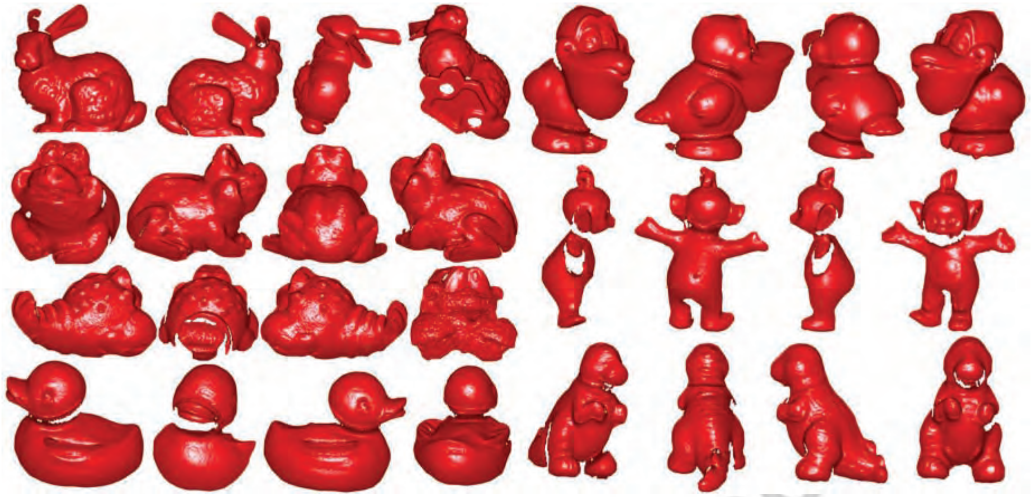

Figure 9: Real scans used in our tests. We show here 4 scans selected from each dataset as representatives of each dataset.

noted, n must be large enough to sufficiently represent surface details, so we use a point-to-scan labeling framework to bypass this problem. We label each point in the graph with a scan index and make use of the uniqueness of closest point in one scan (scan-to-point) to finally determine the point-to-point label assignment (our final aim is to select one point from all input points to represent each base point). In this way, we significantly reduce the size of label set which largely determines the computational cost of the algorithm as explained above. In essence, our point-to-scan-to-point labeling is a coarse-to-fine scheme which greatly speeds up the algorithm.

4. Experiments

In our experiments, we use range scans downloaded from three databases (the Minolta database [10], the Stanford 3D Scanning Repository and the Farman database [37]). In Fig. 9, we show some original range scans from

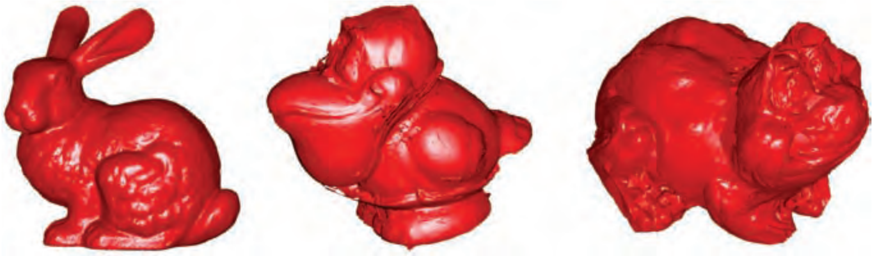

Figure 10: L: Joint meshes of 10 Bunny scans without integration; M: Joint meshes of 17 Bird scans without integration; R: Joint meshes of 17 Frog scans without integration

the Minolta database and the Stanford 3D Scanning Repository for an intuitive visual understanding of the objects. The Standford Bunny dataset (362230 points, 10 scans) is captured at high scanning resolution and highly accurate alignment parameters are given. The other 6 sets of Minolta scans are captured at much lower resolutions with more noise and the alignment parameters are not given. We thus employed the algorithms proposed in [5] and [38] to perform pairwise and global registration for these scans. The registered multi-view scans were then used as the input data for the experiments of the integration methods.

Fig. 10 visualises the different scales of registration errors within different datasets by showing joint meshes of all (registered) input scans without integration. It can be seen that the Bunny scans contain much smaller registration and scanning errors than other scans. The registration errors and some other quantitative statistics of these scans are shown in Table. 2. Note that existing integration methods usually require that the registration error is an order of magnitude less than the measurement error. However, in practice, this assumption is often not satisfied. As demonstrated in Table. 2, the average registration error is about 1/3 to 1/2 of the scanning resolution. In

Dataset	Scans	Points	RE(mm)	SDRE(mm)	MRE/RE	SR(mm)
Bird	17	156094	0.29	0.12	6.46	0.66
Frog	17	174097	0.30	0.15	9.73	0.63
Lobster	18	178653	0.30	0.18	10.62	0.68
Teletubby	17	90848	0.25	0.12	8.00	0.61
Duck	18	252509	0.28	0.19	13.71	0.60
Dinosaur	11	87736	0.54	0.61	34.61	1.25

Table 2: RE: average registration error (RE) over reciprocal correspondences; SDRE: average standard deviation of registration errors; MRE/RE: average of the ratios between the maximum registration error and the mean registration error); SR: average scanning resolution.

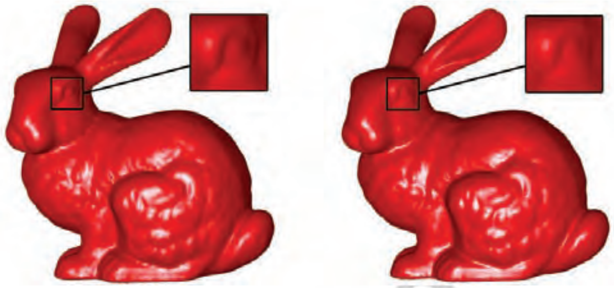

Figure 11: Left: Integration of the Bunny scans using the proposed method; Right: Integration using the SFK method [16]

Table. 2, the ratio between the maximum registration error and the average registration error is very meaningful for integration. It denotes possible visually-poor local reconstruction. In addition, in our experiments, the parameter settings are closely related to the scanning resolution of the input scans to better balance the weights of the three terms of the MRF energy function. For the Bird, the Frog, the Lobster, the Teletubby and the Duck datasets, the truncation parameter F was set to 4 and the weighting parameters λ_1 and λ_2 were set to 5 and 1 respectively. For the Dinosaur dataset, the parameter settings are $F = 6$, $\lambda_1 = 10$ and $\lambda_2 = 2$ as its scanning resolution is significantly different from other datasets.

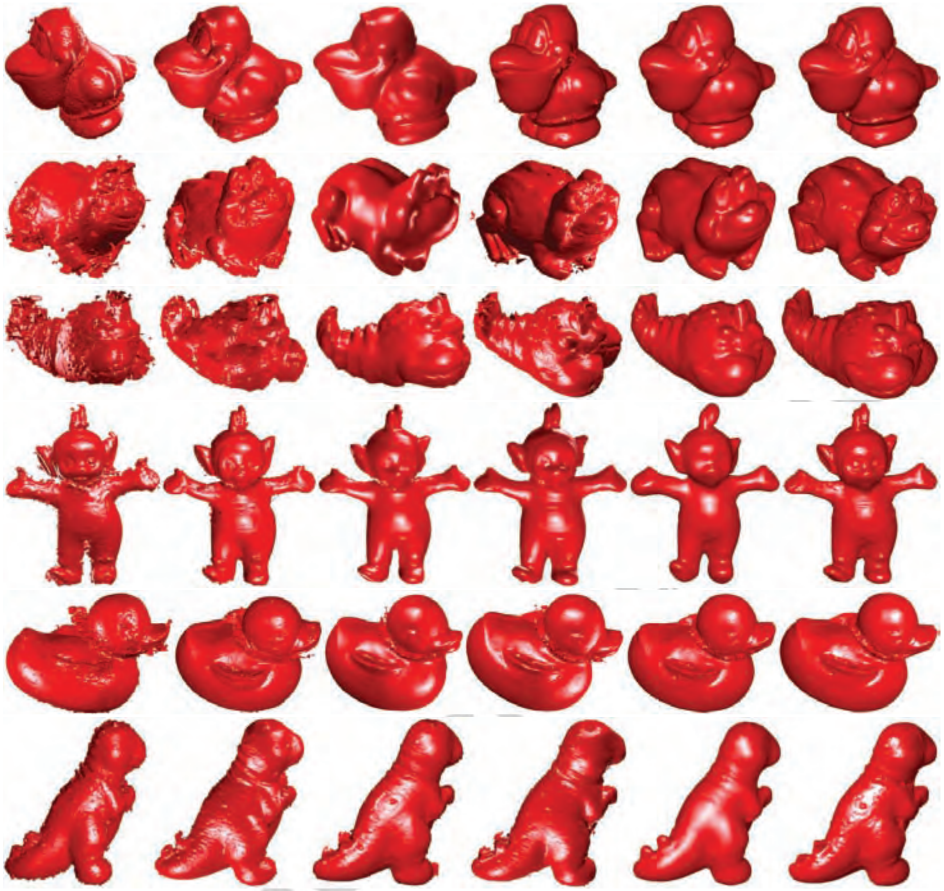

Figure 12: Rows: Integration results of 17 Bird scans, 17 Frog scans, 18 Lobster scans, 17 Teletubby scans, 18 Duck scans and 11 Dinosaur scans. From left to right: volumetric method[12], mesh-based method[18], SFK[16], k-means[15], pairwise MRF[22], our method

Most integration methods can produce a good surface model for the well-registered Bunny scans. For example, Fig. 11 compares our integration method with the segmentation-based method [16] (we call it SFK for short as it first performs a segmentation and then employs fuzzy-c means and k-means clustering to integrate points). Although both deliver good results, our method still performs slightly better, especially on preserving local surface details (see the eye of the bunny).

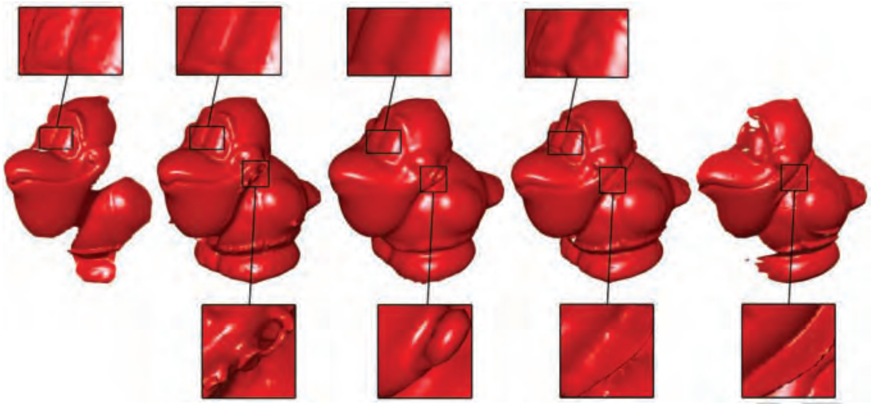

Figure 13: Integration results of the bird's neck and eyes. From left to right: one input scan used as the local ground truth for the surface area of the neck, integration using k-means clustering [15], integration using pairwise MRF [22], integration using our method, another input scan used as the ground truth for the surface area of the eyes

The integration of the other 6 datasets from the Minolta database are more challenging and they are widely used for comparing methods [15, 16]. We hence test different integration methods on these datasets for a direct comparison. As shown in Fig. 12, our method produces clear and geometrically realistic eyes, mouth and wings for the bird, eyes, fingers and pocket (on the chest) for the teletubby, and toes, eyes, mouth for the frog and tail for the dinosaur, etc. In general, the volumetric method fails to produce a clean surface model (also demonstrated by [15]). The mesh-based method and the k-means clustering produce improved surface models but also sometimes generate fragments (see the bird's neck in Fig. 13, the frog's toes in Fig. 14, the teletubby's ears, the lobster's eyes and the duck's neck and mouth). The pairwise MRF and the SFK suffer from oversmoothing although they usually produce a clean surface.

On one hand, if noisy points are spatially not far from the real surface, wrong clustering will produce incorrect local surface geometry, resulting in

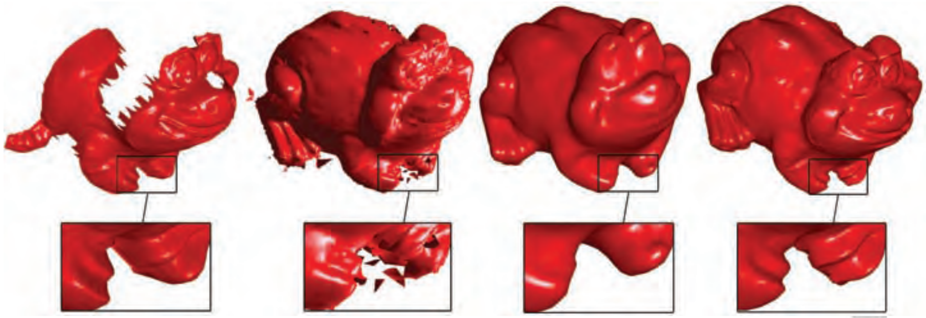

Figure 14: Integration results of the frog's feet. From left to right: one input scan used as the ground truth for the surface area of the feet, integration using k-means clustering [15], integration using pairwise MRF [22], integration using our method

the loss of detail. On the other hand, if enough noisy points are spatially far from the real surface, they will be grouped together and form fragments. Integration methods based on *merging*, where each point from the input data has a weighted contribution to the output, are generally not robust to such noise. In contrast, our method is based on a *selection* scheme where outliers have no contribution to the output, leading to an improved integration.

Compared to pairwise MRF-based methods, our higher-order MRF-based method has the advantage of preserving surface details particularly where the local surface geometry is complicated (e.g. for thin protrusions and deep concavities). A detailed comparison of two recent pairwise MRF-based methods with our higher-order MRF method, using the Dinosaur dataset, is shown in Fig. 16, which scrutinises the tail of the dinosaur. The first pairwise MRF-based method [39] fails to reconstruct a geometrically realistic surface while the second [22] suffers from oversmoothing. This is because pairwise MRF can only capture first-order features; its energy functions cannot model some complicated second-order features such as normal vectors and curvatures. A similar phenomenon is also illustrated by the bird's eyes in Fig. 13, the frog's

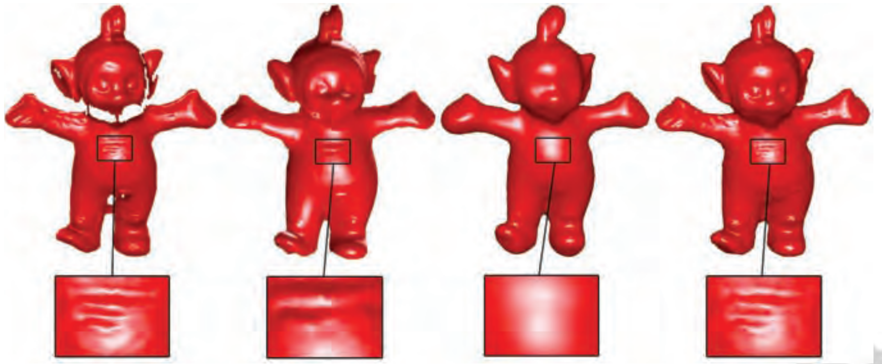

Figure 15: Integration results of the teletubby's chest. From left to right: one input scan used as the ground truth for the surface area of the chest, integration using k-means clustering [15], integration using pairwise MRF [22], integration using our method

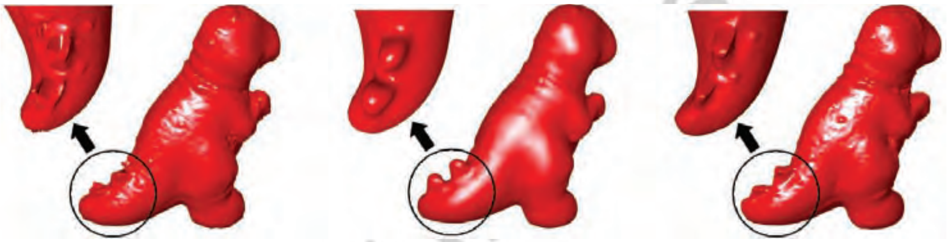

Figure 16: Integration results of the dinosaur tail. Left: The pairwise MRF-based method [39] produced holes; Middle: The pairwise MRF-based method [22] delivered an over-smooth surface; Right: Our method produced a geometrically realistic surface.

face, the lobster's back, the teletubby's face and chest in Fig. 15 and the dinosaur's back.

In general, Fig. 12 qualitatively demonstrate that the proposed method, the pairwise MRF [22], the k-means clustering [15] and the SFK[16] are most successful. Therefore, we conduct quantitative comparisons for these four methods. We quantitatively measures the integration error between each input scan (taken as partial ground truth) and its corresponding surface region in the integration via the highly cited MESH method [40], an efficient technique to estimate the distance between 3D surfaces. The correspond-

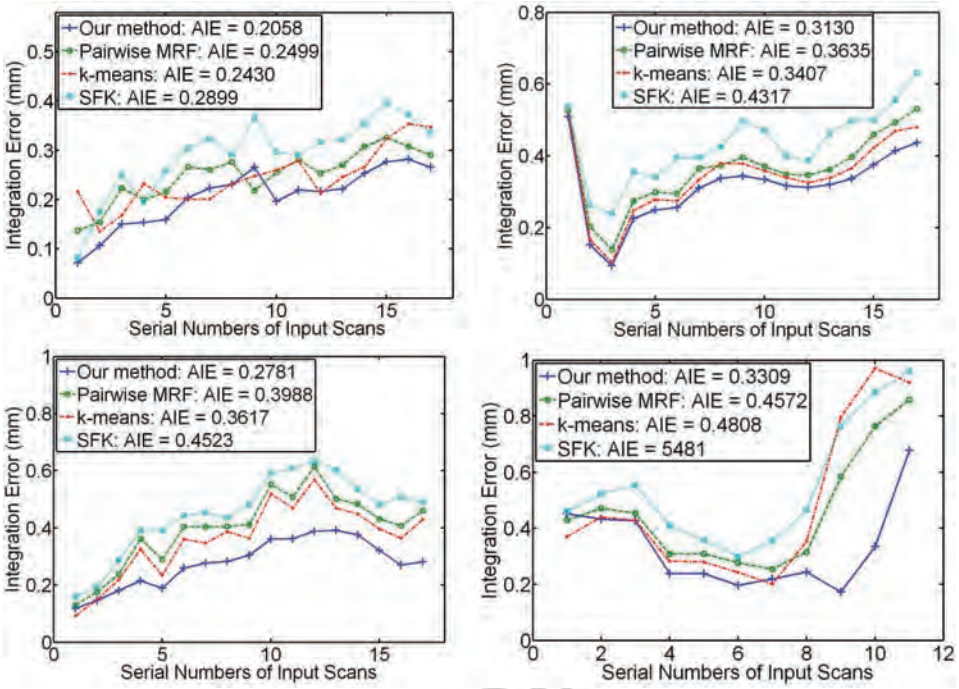

Figure 17: Integration error of different methods based on different datasets. Top left: Bird. Top right: Frog. Bottom left: Teletubby. Bottom right: Dinosaur. AIE: average integration error of all scans

ing surface region of one input scan can be found by searching the nearest points of each point in the registered input scan from the integration, which is similar to the overlapping area detection in Section 2.1. The smaller the integration error, the more similar the input scan with its corresponding region of the integrated surface. Overall integration quality is given by averaging integration errors. The smaller the average integration error, the better the integration. We also measure the root mean square error (RMSE) between each input scan (taken as partial ground truth) and its corresponding surface region in the integration and then compute the average. Consequently, the smaller the average RMSE, the better the overall integration.

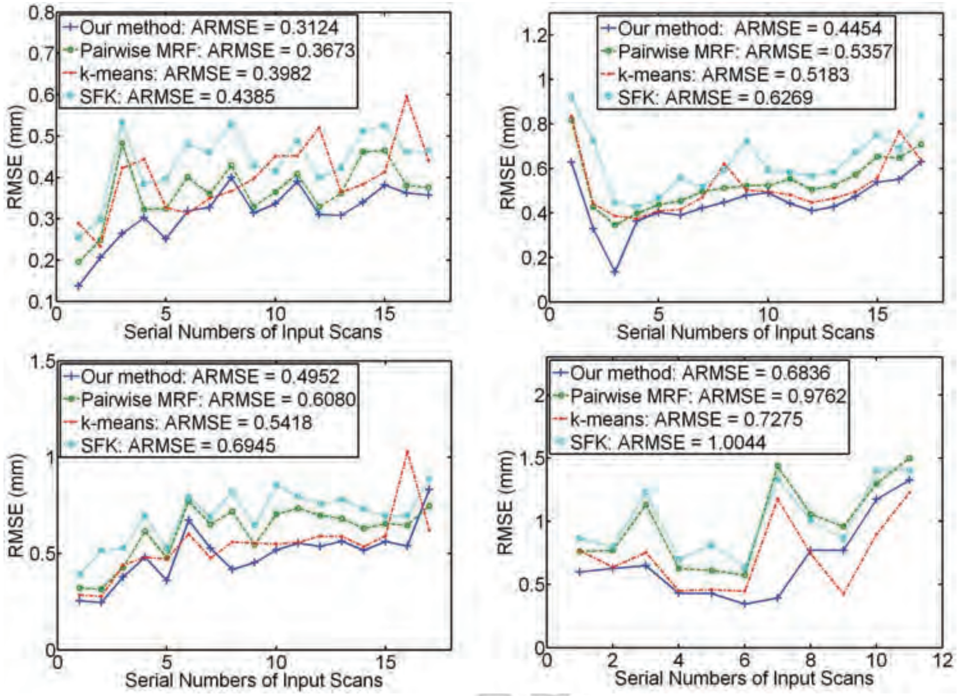

Figure 18: Root mean square error (RMSE) of different methods based on different datasets. Top left: Bird. Top right: Frog. Bottom left: Teletubby. Bottom right: Dinosaur. ARMSE: average root mean square error of all scans

The results of the comparisons through integration error and RMSE are shown in Fig. 17 and 18. In most cases, the errors increase with the number of integrated scans increasing due to the accumulation of registration errors. Some exceptions are possibly caused by large local scanning noise or a significant amount of outliers. With the proposed method, both the integration errors and the RMSEs are usually minima. According to the average integration error and the average RMSE, our method usually improves the overall quality of the integrations of different datasets by 15%–40% compared to the pairwise MRF [22], the k-means clustering [15] and the SFK [16].

We also perform the proposed integration method using the Farman

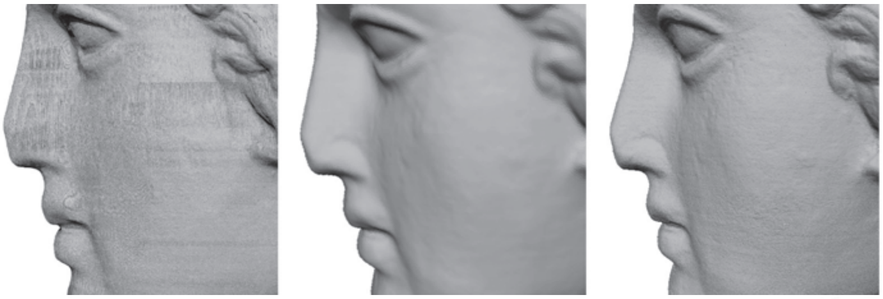

Figure 19: Integration results of the mask scans. Left: without integration; Middle: pairwise MRF-based integration [22]; Right: our integration

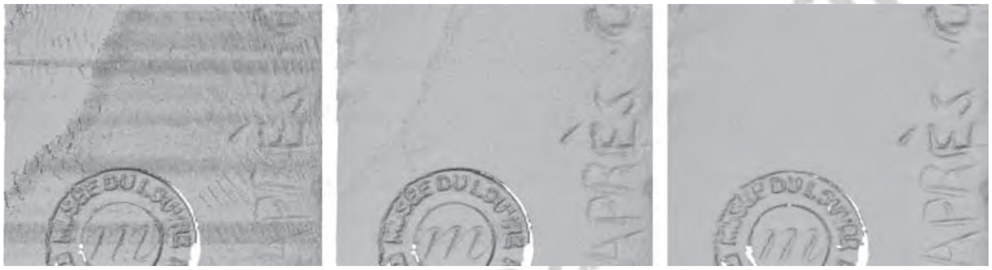

Figure 20: Integration results of the Nefertiti scans. Left: without integration; Middle: high fidelity scan merging [25]; Right: our integration

datasets. Note that the Farman scans are captured by accurately calibrated last generation laser scanners and thus the registration error is very small. As reported in [25], the average registration error for the Farman datasets is $0.081mm$ with standard deviation 0.012 (see Table. 2 for a comparison with the Minolta datasets). Figs. 19-22 present the integration results on these data. For comparison, we display the renderings produced by directly triangulating the registered raw scan points without any integration. Our observations of the results are listed as below.

In general, compared to the Minolta data, the Farman scans are less challenging due to accurate registration. Even so, directly triangulating the scan points without integration always creates some local artifacts such as the

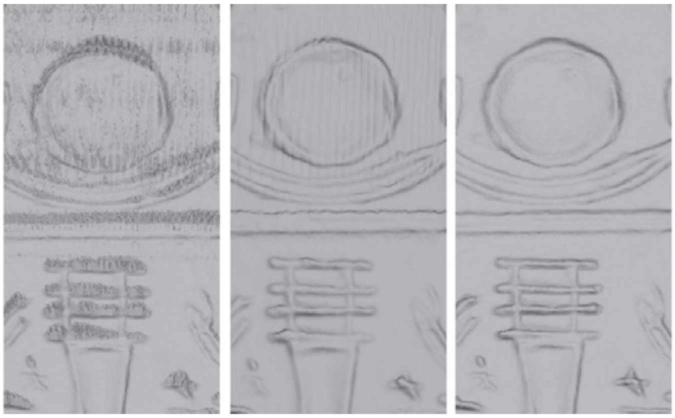

Figure 21: Integration of the pyramid scans. Left: without integration; Middle: Poisson reconstruction [41]; Right: our integration

aliasing visible as grid or tiling effects. The aliasing effect is usually caused by locally inaccurate registration where the overlapping surfaces are not superimposed. In contrast, our integration method removes these undesirable effects almost completely.

In Fig. 19, like its performance on the Minolta scans, the pairwise MRF-based method [22] introduces an important smoothing and loses surface details in comparison to our integration result. In Fig. 20, we observe that the merging method [25] significantly attenuates the aliasing compared with the results produced by directly triangulating the raw scan points without integration. However, the artifact seam is still visible. Such seams often appear at the scan boundaries due to the scan offsets. In contrast, our method produces a seamless integration.

In Figs. 21 and 22, we also compare our method with the widely-used Poisson reconstruction [41]. In Fig. 21, the aliasing effect is not fully removed on the surface produced by Poisson reconstruction. As mentioned above, registration error is a major challenge for integration. But as a level set reconstruction method, Poisson reconstruction does not have a particular

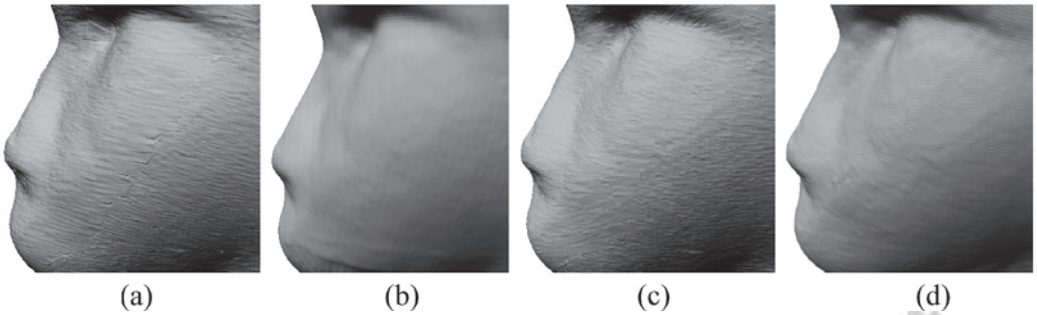

Figure 22: Integration of the Dame de Brassempouy scans. (a) Without integration (b) Poisson reconstruction (c) Integration using $\lambda_1 = 5$ and $\lambda_2 = 1$ (d) Integration using $\lambda_1 = 5$ and $\lambda_2 = 3$

mechanism to deal with such errors. It effectively smooths the surface in a global manner, which could therefore cause a loss of sharpness or a loss of local details as shown in Fig. 22 (b). And in most cases, the distribution of registration error is not globally even. As a result, we can see that in Fig. 21, even with the smoothing, the result of Poisson reconstruction still keeps many vertical artifact lines in the upper part while the smoothing is just appropriate for removing the artifacts in the lower part. Similarly, in Fig. 22 (b), even with the degradation caused by oversmoothing on the facial region of the Dame de Brassempouy, artifacts remain on the chin although the seam across the face which is obvious in Fig. 22 (a) has been removed. The reasons of the visual success of our method, as shown in Fig. 21 and Fig. 22 (c), are that (1) our labeling-based method is essentially a *selection* strategy: the overall surface model is composed of surface patches from selected input scans, which thus preserves local surface details; (2) as mentioned in Section 2.6, in a relatively smooth surface area, label changes usually occur where the scans are well registered (superimposed), which significantly attenuates artifacts such as the aliasing effect and the integration seams. These two

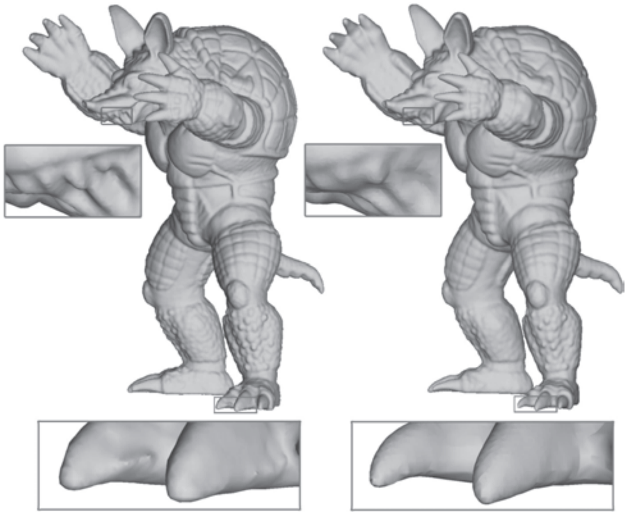

Figure 23: Integration of the Armadillo scans. Left: Integration using $\lambda_1 = 5$ and $\lambda_2 = 1$; Right: Integration using $\lambda_1 = 5$ and $\lambda_2 = 1.5$

reasons can be explained by the one-point and two-point terms of the MRF as summarised in Section 2.6. The effect of the higher-order term can be investigated via Figs. 22(d) and Fig. 23 where we also vary an important parameter λ_2.

It is worth noting the parameters involved in this algorithm. Apparently, there are 6 parameters $k, F, q, \beta, \lambda_1, \lambda_2$. But k and q can be fixed to 3 and 2 respectively in most cases; F is actually related to the average interpoint distance of the input scans and Section 2.2 gives a clue how to set it; β is determined by q, F and the number of input scans as mentioned in Section 2.5. Therefore only the two MRF weighting parameters λ_1 and λ_2 need to be selected empirically. In Figs. 22(d), it can be seen that a larger λ_2 leads to a smoother surface since the higher-order term which computes the difference of adjacent normals gains more weight. Although such smoothing can remove some artifacts (see the toes of the Armadillo in Fig. 23), it inevitably results in the loss of surface details or geometrical features (see the facial region in

Figs. 22(d) and the teeth of the Armadillo).

All experiments used a dual core, 2.4GHz, 3.25GB RAM PC. The integration of each Minolta dataset took 15–30 minutes, mainly depending on the number of the points in the datasets and the number of iterations that the BP used to reach convergence or an acceptable solution. Please refer to Fig.24 in [15] for the computational time of other methods tested using the same datasets. It can be seen that our method is comparable with the most efficient ones.

5. Conclusion and discussion

Existing methods for integrating multiple scans can fuse non-corresponding points and blur the integrated surface or generate artifacts if the point correspondence cannot be correctly established. Unfortunately, this is just the case because inevitable registration errors and scanning noise make the establishment of correct correspondence difficult, if not impossible. In this paper, we solve the integration as a labeling problem.

Compared to the clustering method which only considers the Euclidean distances between points and the pairwise MRF-based method which only considers the two-point spatial interactions, the proposed method delivered more accurate integration and an integrated surface of higher quality due to a more sufficient utility of the 3D spatial information within the input datasets. Be aware that the proposed higher-order MRF model is configured on a specific graph which can also handle 3D unstructured point clouds. The MRF energy function consists of three terms. The one-point term is also the data term which is directly determined by the known input data. The

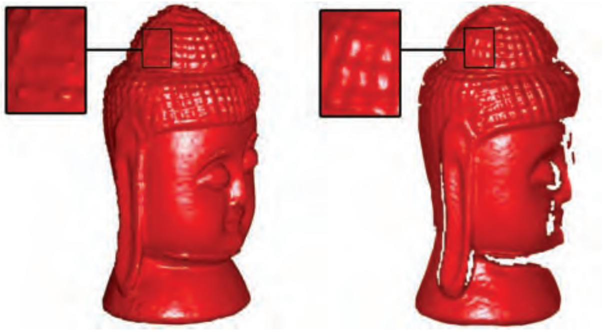

Figure 24: The limitation of our method. Although the proposed method is more robust to registration error than other methods, it fails to retain surface details when the registration error is excessively large. Left: the integration of 18 Buddha scans produced by our method. Right: one input scan used as the ground truth

compatibility term is modeled as a two-point clique cost. We formulate the four-point higher-order term as the difference between neighbouring normals to better capture local surface details. We solve the MRF via belief propagation which is modified to speed up the inference. And the experiments demonstrate that the proposed method produces geometrically realistic surface models with well preserved surface details robustly.

The experiments also demonstrate that our method has a significant degree of robustness to registration errors. This is because (i) we incorporate an error compensation scheme into the proposed graph construction scheme, and (ii) our MRF-based method is essentially a *selection* scheme rather than a merging. This scheme works because registration errors are not evenly distributed across points. Global registration [42, 38, 43, 44, 45] is often used with the aim of evenly distributing registration errors *between* scans, but *within* a single scan, different points suffer from different registration errors. In our final integration, a certain surface patch is derived directly from the corresponding surface area of some selected input scan. Boundaries, where we change from selecting one scan to another, usually occur in places where points are well registered, allowing the selected surfaces patches to be pieced together seamlessly to form a complete 3D surface model of the object.

However, we have to admit that the quality of the integration is still dependent on the registration. In Fig. 24, the average registration error for the Buddha scans is almost doubled compared with the datasets with similar resolutions listed in Table. 2. We consequently observe the failure of retaining local surface details for the Buddha's bumpy hair region in the integration although for the flat surface regions, our method still deliver a good integration. In our method, the difference between neighbouring normals is used as a feature descriptor to capture local surface geometry. Nonetheless, Fig. 24 indicates that normal information is not reliable when the registration is relatively poor. We thereby believe that further improvements may be gained by using different higher-order priors based on more reliable geometric feature descriptors such as the invariant feature descriptor proposed in [46].

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

References

[1] M. Lhuillier, L. Quan, A quasi-dense approach to surface reconstruction from uncalibrated images, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 418–433.

[2] P. Besl, N. McKay, A method for registration of 3-d shapes, IEEE Transactions on pattern analysis and machine intelligence (1992) 239–256.

[3] A. Johnson, Spin-Images: A Representation for 3-D Surface Matching, Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA., 1997.

[4] G. Mori, S. Belongie, J. Malik, Efficient shape matching using shape

contexts, Pattern Analysis and Machine Intelligence, IEEE Transactions on 27 (2005) 1832–1837.

[5] Y. Liu, Automatic range image registration in the markov chain, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 12–29.

[6] L. Silva, O. Bellon, K. Boyer, Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 762–776.

[7] Y. Liu, Automatic 3d free form shape matching using the graduated assignment algorithm, Pattern Recognition 38 (2005) 1615–1631.

[8] D. Aiger, N. Mitra, D. Cohen-Or, 4-points congruent sets for robust pairwise surface registration 27 (2008) 85.

[9] Y. Liu, Penalizing closest point sharing for automatic free form shape registration, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 1058–1064.

[10] P. Flynn, R. Campbell, A www-accessible database for 3d vision research, Proc. IEEE Workshop on Empirical Evaluation Methods in Computer Vision (1998) 148–154.

[11] B. Curless, M. Levoy, A volumetric method for building complex models from range images, in: Proc. SIGGRAPH 1996, pp. 303–312.

[12] C. Dorai, G. Wang, Registration and integration of multiple object views for 3d model construction, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 83–89.

[13] S. Rusinkiewicz, O. Hall-Holt, M. Levoy, Real-time 3d model acquisition, in: Proc. SIGGRAPH 2002, pp. 438–446.

[14] R. Sagawa, K. Nishino, K. Ikeuchi, Adaptively merging large-scale range data with reflectance properties, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 392–405.

[15] H. Zhou, Y. Liu, Accurate integration of multi-view range images using k-means clustering, Pattern Recognition 41 (2008) 152–175.

[16] H. Zhou, Y. Liu, L. Li, B. Wei, A clustering approach to free form surface reconstruction from multi-view range images, Image and Vision Computing 27 (2009) 725–747.

[17] M. Rutishauser, M. Stricker, M. Trobina, Merging range images of arbitrarily shaped objects, in: Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 1994, pp. 573–580.

[18] Y. Sun, J. Paik, A. Koschan, M. Abidi, Surface modeling using multi-view range and color images, Int. J. Comput. Aided Eng. 10 (2003) 37–50.

[19] G. Turk, M. Levoy, Zippered polygon meshes from range images, in: Proc. SIGGRAPH 1994, pp. 311–318.

[20] J. Diebel, S. Thrun, An application of markov random fields to range sensing, in: Proc. Advances in Neural Information Processing Systems (NIPS) 2006, pp. 291–298.

[21] J. Diebel, S. Thrun, M. Brunig, A bayesian method for probable surface reconstruction and decimation, ACM Transactions on Graphics 25 (2006) 39–59.

[22] R. Paulsen, J. Bærentzen, R. Larsen, Markov random field surface reconstruction, IEEE Transactions on Visualization and Computer Graphics 16 (2010) 636–646.

[23] C. Zach, T. Pock, H. Bischof, A globally optimal algorithm for robust tv-l1 range image integration, in: Proc. IEEE International Conference on Computer Vision (ICCV) 2007, pp. 1–8.

[24] Q. Huang, B. Adams, M. Wand, Bayesian surface reconstruction via iterative scan alignment to an optimized prototype, in: Proc. Eurographics Symposium on Geometry Processing 2007, pp. 213–223.

[25] J. Digne, J.-M. Morel, N. Audfray, C. Lartigue, High fidelity scan merging, Computer Graphics Forum 29 (2010) 1643–1651.

[26] Y. Kil, B. Mederos, N. Amenta, Laser scanner super-resolution (2006) 9–16.

[27] F. Abbasinejad, Y. Kil, A. Sharf, N. Amenta, Rotating scans for systematic error removal, in: Computer Graphics Forum, volume 28, Wiley Online Library, pp. 1319–1326.

[28] N. Amenta, S. Choi, R. Kolluri, The power crust, in: Proc. the Sixth ACM Symposium on Solid Modeling 2001, pp. 249–260.

[29] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, C. Rother, A comparative study of energy minimization methods for markov random fields with smoothness-based priors, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 1068–1080.

[30] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts?, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 147–159.

[31] P. Felzenszwalb, D. Huttenlocher, Efficient belief propagation for early vision, International Journal of Computer Vision 70 (2006) 41–54.

[32] X. Lan, S. Roth, D. Huttenlocher, M. Black, Efficient belief propagation with learned higher-order markov random fields, in: Proc. European Conference on Computer Vision 2006.

[33] J. Yedidia, W. Freeman, Y. Weiss, Understanding belief propagation and its generalizations, Exploring artificial intelligence in the new millennium 8 (2003) 236–239.

[34] P. Kohli, M. Kumar, P. Torr, $\mathcal{P}^3$ & Beyond: Solving energies with higher order cliques, in: Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2007, pp. 1–8.

[35] C. Rother, P. Kohli, W. Feng, J. Jia, Minimizing sparse higher order energy functions of discrete variables, in: Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2009, pp. 1–8.

[36] N. Komodakis, N. Paragios, Beyond pairwise energies: Efficient optimization for higher-order MRFs, in: Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2009, pp. 1–8.

[37] J. Digne, N. Audfray, C. Lartigue, C. Mehdi-Souzani, J.-M. Morel, Farman institute 3d point sets - high precision 3d data sets, Image Processing On Line 2011 (2011).

[38] K. Nishino, K. Ikeuchi, Robust simultaneous registration of multiple range images, in: Proc. Asian Conference on Computer Vision 2002, pp. 1–8.

[39] R. Song, Y. Liu, R. Martin, P. Rosin, MRF labeling for multi-view range image integration, in: Proc. Asian Conference on Computer Vision 2010, pp. 27–40.

[40] N. Aspert, D. Santa-Cruz, T. Ebrahimi, Mesh: Measuring errors between surfaces using the hausdorff distance, in: Proc. IEEE International Conference onMultimedia and Expo 2002, pp. 705–708.

[41] M. Kazhdan, M. Bolitho, H. Hoppe, Poisson surface reconstruction, in: Proceedings of the fourth Eurographics symposium on Geometry processing.

[42] R. Bergevin, M. Soucy, H. Gagnon, D. Laurendeau, Towards a general multi-view registration technique, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1996) 540–547.

[43] N. Gelfand, N. Mitra, L. Guibas, H. Pottmann, Robust global registration, in: Proc. Eurographics Symposium on Geometry Processing 2005, pp. 1–10.

[44] S. Krishnan, P. Lee, J. Moore, S. Venkatasubramanian, Global registration of multiple 3d point sets via optimization-on-a-manifold, in: Proc. Eurographics Symposium on Geometry Processing 2005, pp. 1–10.

[45] B. Brown, S. Rusinkiewicz, Global non-rigid alignment of 3-d scans, ACM Transactions on Graphics (Proc. SIGGRAPH) 26 (2007) 21–30.

[46] T. Lo, J. Siebert, Local feature extraction and matching on range images: 2.5d sift, Computer Vision and Image Understanding 113 (2009) 1235–1250.