2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission

Higher Order CRF for Surface Reconstruction from Multi-View Data Sets

Ran Song Yonghuai Liu
Department of Computer Science
Aberystwyth University
Aberystwyth, UK
Email: {res, yyl}@aber.ac.uk

Ralph R. Martin Paul L.Rosin
Department of Computer Science and Informatics
Cardiff University
Cardiff, UK
Email: {Ralph.Martin, Paul.Rosin}@cs.cardiff.ac.uk

***Abstract*—We propose a novel method based on higher order Conditional Random Field (CRF) for reconstructing surface models from multi-view data sets. This method is automatic and robust to inevitable scanning noise and registration errors involved in the stages of data acquisition and registration. By incorporating the information within the input data sets into the energy function more sufficiently than existing methods, it more effectively captures spatial relations between 3D points, making the reconstructed surface both topologically and geometrically consistent with the data sources. We employ the state-of-the-art belief propagation algorithm to infer this higher order CRF while utilizing the sparseness of the CRF labeling to reduce the computational complexity. Experiments show that the proposed approach provides improved surface reconstruction.**

***Keywords*-Conditional Random Field; Surface Reconstruction; Integration; Multi-View Data Sets;**

I. INTRODUCTION

Two types of method are currently popular for 3D surface reconstruction from multi-view data sets. One is multi-view stereo where the input usually comprises 2D photographs. The other acquires range data (2.5D or 3D point clouds) via laser scanning [2]. Both methods depend on initially accurately registering the input views into a common coordinate system. In the stereo method, alignment information is usually obtained by calibration techniques, which lead to accurate registration, but are not automatic. Many automatic registration techniques have been developed for laser range data [14], [15]. Ultimately, however, registration errors are inevitable. Such errors have a significant impact on surface reconstruction from multi-view data sets, especially when registration errors are accumulated.

Constructing a consistent surface (again as a point cloud) from such multi-view data sets is difficult: approaches often depend on the tricky problems of accurately finding the transformations linking the data sets, and optimally integrating them. Here, we specifically address *integration* of the data, meaning (i) effective reduction of redundant information in areas where the data sets overlap, while (ii) sufficiently preserving data describing surface details. Integration is a difficult task due to scanning noise and outliers, registration errors, and unreliable data measurements.

This paper focuses on robust (with respect to poor registration, in particular) and automatic integration of the most general multi-view data—multiple 3D unstructured point clouds. Most input data used for 3D surface reconstruction can be converted to point clouds (but not always *vice versa*), allowing the proposed method to address a wide range of applications.

II. PREVIOUS WORK

Existing methods for surface reconstruction from multi-view data sets can be analysed in terms of the different integration methods employed (rather than other steps, e.g. triangulation). Doing so leads to four categories: volumetric, mesh-based, clustering-based and Bayesian approaches.

Volumetric methods [5], [20] integrate data by merging them in each voxel using a data fusion algorithm. Most multi-view stereo-based reconstruction techniques (e.g. those in [7]) employ this approach to produce a complete 3D model after depth estimation via stereo. These methods require highly accurate alignment information (estimated via manually-assisted camera calibration, or given as known input in e.g. the Middlebury database [7]), and generally do not permit the multi-view data to be unstructured point clouds. Thus, in many cases the volumetric method works poorly or is inapplicable [28], [29].

Mesh-based methods [19], [22] 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 computationally expensive as triangles outnumber the sample 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 fail for highly curved regions where no unique projection plane exists. Furthermore, such strategies usually cannot deal with 3D unstructured point clouds as they rely on the 2D lattice structure of the input data.

Clustering-based methods [28], [29] employ classical clustering methods to minimise an objective function based on Euclidean distances. They are generally superior to previous methods, being more robust to scanning noise and registration errors. However, Euclidean distances are

978-0-7695-4369-7/11 $26.00 © 2011 IEEE
DOI 10.1109/3DIMPVT.2011.27

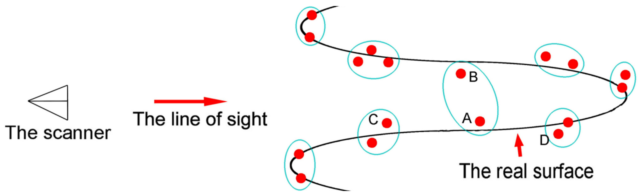

Figure 1. Local topology has a significant effect on the point clustering in non-flat areas

used to allocate points to the closest cluster centroid, and furthermore do not consider local surface topology and neighbourhood consistency. This leads to severe errors in highly curved areas. For instance, in Fig. 1, point A is closest to B so they would be clustered together, whereas A should be clustered with C or D to preserve correct surface topology and geometry.

Bayesian methods [4], [9], [8], [17], particularly those based on Markov random field (MRF), are very promising [17]. However, existing methods have two weaknesses: (i) energy functions based only upon pairwise MRF are not adequate to capture 3D information, and (ii) for those employing a high-order prior to better capture 3D long-range information, they do not infer the MRFs using state-of-the-art techniques such as message passing because such inferences are usually intractable as the computational complexity is exponential to the number of the order.

To overcome these weaknesses, we propose a method based on Conditional Random Field (CRF). Section 3 gives a background about CRF in 3D applications which has mainly been used to cope with 2D problems. The following three sections describe the proposed method in detail. In Section 4, we first produce a graph by point shifting and triangulation. And then, in Section 5, we give the details on how a higher-order CRF model is configured on this graph. Section 6 discusses the belief propagation (BP) algorithm used to infer this CRF, subject to the *maximum a posteriori* (MAP) constraint. Experimental results are given in Section 7, and conclusions in Section 8.

III. CRF in fully 3D

Unlike techniques [3], [26], [27] which essentially operate on 2D projections, or 2.5D slices, and at the same time assume lattice-based input data, our method does not require such assumption and is applicable to the most general multi-view data sets—multiple 3D unstructured point clouds. For instance, in [27], a set of 2D range images with lattice structure is used as input, where the image domain is strictly a rectangle in $\mathbb{R}^2$. However, due to large registration errors and other noise, this assumption is often not satisfied in practice. Fig. 2 shows an example. In Fig. 2, an original range image (shown as green points and used as the reference image here) without any transformation *does* have a regular 2D image lattice structure in its image plane. But another range image

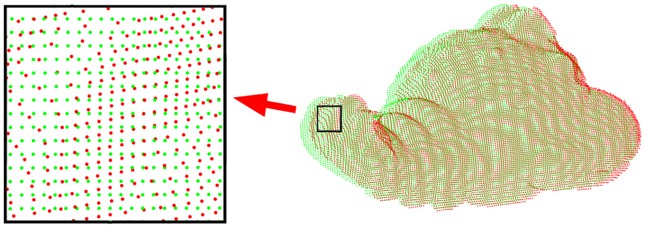

Figure 2. Projection on the image plane of an original range image (the green points) and a registered range image (the red points).

(shown in red) registered to the same coordinate system has lost such lattice structure due to the registration. For multi-view range images, this issue is more severe due to error accumulation as we map all input range images into the space of the reference image.

When used in fully 3D, as opposed to 2.5D slices or a lattice-based 3D voxel space, pairwise CRF/MRF typically cannot model geometrical features of interest, as it only captures point-pair constraints. For example, an important geometrical feature for most 3D data—the normal vector of a mesh facet—is determined by three neighbouring points. Thus, we use a higher order CRF model to represent complicated spatial interactions.

Our approach also differs in that no assumption is made about the distribution of noise. Methods employing probabilistic models for surface reconstruction [17] often assume that the noise is independent and identically-distributed Gaussian. However, in practice, registration errors are the major source of noise, and these are not Gaussian but have a distribution with a long tail [15].

Fig. 3 illustrates the workflow of the proposed CRF-based surface reconstruction scheme from multi-view 3D unstructured point clouds. Please note that the main contribution of this paper is the novel integration method composed of three steps: graph construction, CRF modeling and MAP inference using BP.

IV. CRF graph construction

In a CRF/MRF graph, the nodes are usually pixels or voxels. For example, in [3], [6], [26], [24], the graphs are 2D or 3D lattices. Here, we do not have the benefit of a lattice. We also note that simply finding the k-nearest neighbours of each point does not suffice for finding neighbours, as these are based only on Euclidean distance, which fails to take into account surface topology, and the presence of registration errors.

We use a four-step scheme to construct an CRF graph G from multiple unstructured point clouds.

1) Overlapping area detection Given a set of consecutive point clouds $P_1, P_2, \ldots, P_m$, we employ the pairwise registration method from [14] to obtain a transform H_{12} mapping P_1 into the coordinate system of P_2. To integrate the transformed point cloud P_1' and the reference point cloud

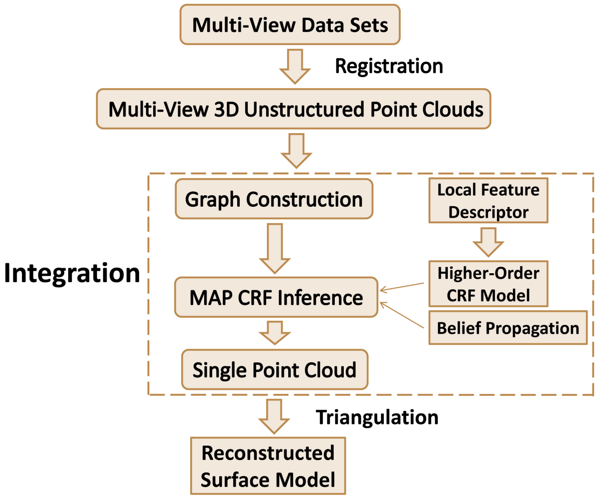

Figure 3. The workflow of the proposed method

P_2, the overlapping and non-overlapping areas of each have to be accurately and efficiently 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. A k-D tree is used for speed of search. The threshold is set to $3R$, where R is the scanning resolution of the input data.

2) Two-view integration 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 set P of nodes in G as:

$$P = S_{\text{non-overlap}} = S_1 + S_2 \tag{1}$$

Next, 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 $\mathbf{P}^*$:

$$\mathbf{P} \rightarrow \mathbf{P} + 0.5\mathbf{d} \cdot \mathbf{N}, \quad \mathbf{d} = \Delta\mathbf{P} \cdot \mathbf{N}, \quad \Delta\mathbf{P} = \mathbf{P}^* - \mathbf{P} \tag{2}$$

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:

$$P = S_{\text{non-overlap}} + S_{\text{overlap}} \tag{3}$$

This strategy (i) compensates for pairwise registration errors as corresponding points are closer to each other, (ii) does not alter the tangential spread of the overlap, as points are moved along their normals, and (iii) leaves the surface topology unaffected, as again, the shift is along the normal.

3) Multi-view integration We now consider the third input point cloud P_3. We map the current P into the coordinate system of P_3 using the transform H_{23}. The overlap between P' transformed from P and the current reference point cloud P_3 is then detected. We now update P based on Eq. (3). 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 two-view integration strategy. We iteratively apply this updating scheme to all input point clouds.

4) Triangulation Finally, we triangulate P to construct the graph G. G is a mesh and we use the notation $\mathcal{V}$ to denotes the vertices in G for clarity albeit $\mathcal{V} = P$. We define two types of neighbourhood systems. The two-point neighbourhood $\mathcal{E}$ is the set of edges connecting vertices. The three-point neighbourhood $\mathcal{F}$ is the set of vertex trifolds and each of them is composed of the three vertices in the same triangular facet. This approach has advantages over defining neighbourhoods using the k-nearest neighbours method: it reflects the surface topology and does not need an estimate for k.

V. Higher order CRF modeling

CRF is a probabilistic framework where no effort is wasted on modeling the observations and arbitrary attributes of the observation data may be captured by the model, without the modeler having to worry about how these attributes are related. We define a label assignment $\mathbf{x} = \{x_i, \forall i \in \mathcal{V}\}$ to all vertices as a realisation of a family of random variables defined on G. We also define a random observation field $\mathbf{y} = \{y_i, \forall i \in \mathcal{V}\}$. We define the label set $L = \{1, \dots, m\}$ as the set of serial numbers of input point clouds. Thus, we have $\forall x_i \in L$ and $\forall y_i \in L$. A CRF is defined as below:

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{Z} \exp\Big\{ -\sum_{c \in C} (\lambda_c \cdot \psi_c(x_c|y_c)) \Big\}, \tag{4}$$

where Z is a normalising constant. The factors ψ_c are potential functions of the random variables x_c within a clique $c \in C$. It can be seen that a CRF is actually an MRF globally conditioned on the observed data. Mathematically, it is equal to an MRF where all prior terms are conditioned on the observed data. Therefore, Eq. (4) is still in the form of a Gibbs distribution due to the Markovian property of the CRF. The weighting parameter λ_c is usually estimated through learning. An efficient learning can be fairly sophisticated (e.g. [1]). In this work, we just tune λ_c empirically. On one hand, unlike many 2D vision tasks, in our 3D surface reconstruction problem, we do not have a large and proper dataset for training. On the other hand, because our CRF only has three terms, it is not difficult to find two proper weighting parameters by adjusting them in tests, leading to an acceptable result. Furthermore, this allows us to make a fair comparison with other aforementioned existing methods on performance and computational time.

The Gibbs energy of the proposed CRF is expressed as

$$\begin{aligned} E(\mathbf{x}) &= -\log p(\mathbf{x}|\mathbf{y}) - \log Z = \sum_{c\in C} \lambda_c \cdot \psi_c(x_c|y_c) \\ &= \sum_{i\in\mathcal{V}} \psi_i(x_i|y_i) + \sum_{(i,j)\in\mathcal{E}} \lambda_1 \cdot \psi_{ij}(x_i, x_j|y_i, y_j) \\ &\quad + \sum_{(i,j,k)\in\mathcal{F}} \lambda_2 \cdot \psi_{ijk}(x_i, x_j, x_k|y_i, y_j, y_k). \end{aligned} \tag{5}$$

Our aim is to find the *maximum a posteriori* (MAP) label assignment $\mathbf{x}^*$

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in L} p(\mathbf{x}|\mathbf{y}) = \arg\min_{\mathbf{x}\in L} E(\mathbf{x}). \tag{6}$$

A. One-point potential

The one-point potential $\psi_i(x_i|y_i)$ is computed by

$$\psi_i(x_i|y_i) = \sum_{y_i\in L\backslash x_i} \min(D_i(x_i, y_i), A) \tag{7}$$

where A is a truncation parameter, and

$$D_i(x_i, y_i) = \|C_i(x_i) - C_i(y_i)\| \tag{8}$$

is a distance function; $C_i(l), l \in L$ denotes i's closest matching point in the lth input point cloud. In this way, we convert the estimation for the likelihood potential of different labels at point i into the measurement of the distances between i's closest points in input point clouds with different labels. A eliminates the effects of input point clouds which do not cover the area around i. It can be seen that the real observed data for the node i in the graph is actually the set of its closest points in different point clouds.

B. Two-point potential

The two-point potential $\psi_{ij}(x_i, x_j|y_i, y_j)$ for two neighbouring points i and j takes the form of the Potts model [6]:

$$\psi_{ij}(x_i, x_j|y_i, y_j) = \begin{cases} 0 & x_i = x_j, \\ g_{ij}(x_i, x_j|y_i, y_j) & \text{otherwise}, \end{cases} \tag{9}$$

where $g_{ij}(x_i, x_j|y_i, y_j)$ is estimated by

$$g_{ij}(x_i, x_j|y_i, y_j) = \frac{D_i(x_i, x_j) + D_j(x_i, x_j)}{\psi_i(x_i|y_i) + \psi_j(x_j|y_j)}. \tag{10}$$

We thus hope that the neighbouring points i and j can be labeled consistently, which means in the output point cloud, they are replaced with two neighbouring points from the same point cloud. In this way, the local surface details can be well preserved. If they are labeled with two different point cloud serial numbers x_i and x_j respectively, we hope that the two closest points (in the two point clouds) of i (or j) can be close to each other, subject to the distances between the closest points (in different point clouds) of i (or j). It also tends to maintain correct local surface details.

The one-point potential ψ_i and the two-point potential ψ_{ij} constitute a typical pairwise CRF model. This model often achieves good results in low-level 2D vision applications. However, for 3D problems, richer spatial information is needed to more accurately represent local surface details such as local surface geometry and topology. Pairwise CRF/MRF cannot capture such rich statistics, and typically lead to an oversmooth reconstructed surface (see Fig. 4).

C. Higher order potential

Thus we propose a higher order CRF by introducing a three-point potential. This idea is inspired by the nature that a very important property of a 3D triangular mesh, directly related to local surface geometry and topology–the normal of each facet, is determined by three vertices jointly. That the normals within the output point cloud are correctly consistent with some reliable input data is vital to produce geometrically realistic surface for the integration.

The normal of a triangular face defined by the three adjacent points i, j and k is given by

$$\mathbf{N} = \frac{(\mathbf{P}_j - \mathbf{P}_i) \times (\mathbf{P}_k - \mathbf{P}_i)}{\|(\mathbf{P}_j - \mathbf{P}_i) \times (\mathbf{P}_k - \mathbf{P}_i)\|}, \tag{11}$$

where $\mathbf{P}_i$ denotes the 3D coordinates of point i.

The three-point potential for three neighbouring points i, j and k is defined as

$$\psi_{ijk}(x_i,x_j,x_k|y_i,y_j,y_k) = \left\|\mathbf{N}(x_i,x_j,x_k) - \overline{\mathbf{N}}(y_i,y_j,y_k)\right\| \tag{12}$$

where

$$\mathbf{N}(x_i,x_j,x_k) = \frac{(C_j(x_j)-C_i(x_i))\times(C_k(x_k)-C_i(x_i))}{\|(C_j(x_j)-C_i(x_i))\times(C_k(x_k)-C_i(x_i))\|} \tag{13}$$

and $\overline{\mathbf{N}}(y_i, y_j, y_k)$ is the mean of a set of observed normals $\{\mathbf{N}(y_i, y_j, y_k), \{y_i, y_j, y_k\} \in L\}$ in which each $\mathbf{N}(y_i, y_j, y_k)$ satisfies $\mathbf{N}(y_i, y_j, y_k) \neq \mathbf{0}$. Here $\mathbf{0}$ is the zero vector and $\mathbf{N}(y_i, y_j, y_k)$ is calculated by

$$\mathbf{N}(y_i, y_j, y_k) = \begin{cases} \frac{(C_j(y)-C_i(y))\times(C_k(y)-C_i(y))}{\|(C_j(y)-C_i(y))\times(C_k(y)-C_i(y))\|} & \text{if } *, \\ \mathbf{0} & \text{otherwise}, \end{cases} \tag{14}$$

where $*$ denotes the condition

$$\begin{aligned} &\|\mathbf{P}_i - C_i(y)\| < 3R, \quad \text{and} \quad \|\mathbf{P}_j - C_j(y)\| < 3R, \quad \text{and} \\ &\|\mathbf{P}_k - C_k(y)\| < 3R, \quad \text{and} \quad y = y_i = y_j = y_k,\ y \in L. \end{aligned}$$

By constraining the distance between a vertex in the graph and its closest point in some input point cloud, the condition $*$ guarantees that only those input point clouds which cover the area around the points i, j and k have contributions to the calculation of the mean. We also require that $y = y_i = y_j = y_k$ because the normal defined by three points from different point clouds is a meaningless observation and essentially unreliable due to registration errors. Thus the three-point potential encourage the labeling to be consistent with valid and reliable observed normals.

VI. Energy minimisation via Belief Propagation

Several methods exist for minimising posterior energy. The comparative study in [23] recommends two approaches, graph-cuts (GC) [11] and message passing, e.g. belief propagation (BP) [6], as efficient and powerful. Since our energy function is neither metric nor semi-metric, GC is not applicable.

We employ BP to find a MAP solution. BP operates by passing messages between points in the graph G. Because the two-point belief is independent of the three-point belief, each iteration uses two types of message updates.

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

$$m_{ji}(x_i) = \min_{x_j} \Big(\psi_i + \lambda_1 \psi_{ij} + \sum_{(h,j)\in\mathcal{E}\setminus(i,j)} m_{hj}(x_j) \Big) \quad (15)$$

Please note that in Eq. (15) and the following mathematical reasoning, we use a set of simplified notations:

$$\psi_i = \psi_i(x_i|y_j), \quad b_i = b_i(x_i), \quad \psi_{ij} = \psi_{ij}(x_i, x_j|y_i, y_j)$$
$$\psi_{ijk} = \psi_{ijk}(x_i, x_j, x_k|y_i, y_j, y_k)$$
$$b_{kji} = b_{kji}(x_i, x_j, x_k|y_i, y_j, y_k)$$
$$E_{kji} = E_{kji}(x_i, x_j, x_k|y_i, y_j, y_k)$$

The other type of message sent to i is written as $m_{kji}(x_i)$ with $\{j, k\} \in \mathcal{F}_p(i)$, where $\mathcal{F}_p(i)$ denotes the point pair set in which each point pair (j, k) forms a three-point clique with i and $(i, j, k) \in \mathcal{F}$. Let b_i denote the one-point belief and b_{kji} denote the three-point belief. Then energies associated with the three-point cliques can be define as

$$E_{kji} = \psi_i + \psi_j + \psi_k + \lambda_2 \psi_{ijk}, \quad (16)$$

and the Gibbs energy [25] is

$$E = \sum_{ijk} \sum_{x_i x_j x_k} e^{-b_{kji}} (E_{kji} - b_{jki}) - \sum (q_i - 1) \sum_{x_i} e^{-b_i} (\psi_i - b_i)$$

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

$$r_{kji}\colon \sum_{x_i x_j x_k} e^{-b_{kji}} - 1 = 0, \quad r_i\colon \sum_{x_i} e^{b_i} - 1 = 0.$$

The multiplier that enforces the max-marginalisation constraints is

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

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

$$\frac{\partial L}{\partial e^{-b_{kji}}} = 0, \quad \text{and hence}$$
$$-b_{kji} = E_{kji} + 1 + \lambda_{kji}(x_i) + \lambda_{ikj}(x_j) + \lambda_{jik}(x_k) + r_{kji},$$

$$\frac{\partial L}{\partial e^{-b_i}} = 0, \quad \text{and hence}$$
$$-b_i = -\psi_i + \frac{1}{q_i - 1} \sum_{(j,k)\in\mathcal{F}_p(i)} \lambda_{kji}(x_i) + r'_i,$$

where r'_i is the rearranged constant.

By change of variable, defining

$$\lambda_{kji}(x_i) = - \sum_{(h,g)\in\mathcal{F}_p(i)\setminus(k,j)} m_{hgi}(x_i), \quad (17)$$

we obtain the following two equations

$$b_i = \psi_i + \sum_{(k,j)\in\mathcal{F}_p(i)} m_{kji}(x_i),$$

$$b_{kji} = \psi_i + \psi_j + \psi_k + \lambda_2 \psi_{ijk} + \sum_{(h,g)\in\mathcal{F}_p(i)\setminus(k,j)} m_{hgi}(x_i)$$
$$+ \sum_{(h,g)\in\mathcal{F}_p(j)\setminus(i,k)} m_{hgj}(x_j) + \sum_{(h,g)\in\mathcal{F}_p(k)\setminus(j,i)} m_{hgk}(x_k).$$

Due to the min-sum constraint (arising from the MAP constraint): $b_i = \min_{x_k x_j} b_{kji}$, we have

$$m_{kji}(x_i) = \min_{x_k x_j} \Big(\psi_j + \psi_k + \lambda_2 \psi_{ijk} + \sum_{(h,g)\in\mathcal{F}_p(j)\setminus(i,k)} m_{hgj}(x_j)$$
$$+ \sum_{(h,g)\in\mathcal{F}_p(k)\setminus(i,j)} m_{hgk}(x_k) \Big). \quad (18)$$

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:

$$B_i(x_i) = \psi_i + \sum_{(i,j)\in\mathcal{E}} m_{ji}(x_i) + \sum_{(j,k)\in\mathcal{F}_p(i)} m_{kji}(x_i). \quad (19)$$

The label x_i^* that minimises $B_i(x_i)$ individually at each point is selected.

Computing m_{kji} is extremely costly: $O(m^3)$, as the message vector has m elements which are computed by minimising Eq. (18) over 2 variables each of which has m possible states. Similarly, the cost for computing m_{ji} is $O(m^2)$. If the graph G has n vertices and the we run T iterations, the total cost will be $O(nm^2(m + 1)T)$. Clearly, such an algorithm is intractable if both n and m are large. Most existing efficient higher-order CRF/MRF optimisation methods are just applicable to specific families of energy functions or specific (small) cliques such as the P^n model [10], quadratic functions [18] truncated functions [12] and 2×2 cliques [13], etc. None of these methods work for the complicated structure of our energy function.

As noted, n must be large enough to sufficiently represent surface details, so we use a point-to-cloud labeling framework to bypass this problem. We label each point with a point cloud serial number and make use of the uniqueness of closest point in one point cloud (cloud-to-point) to finally determine the point-to-point label assignment. In essence, our

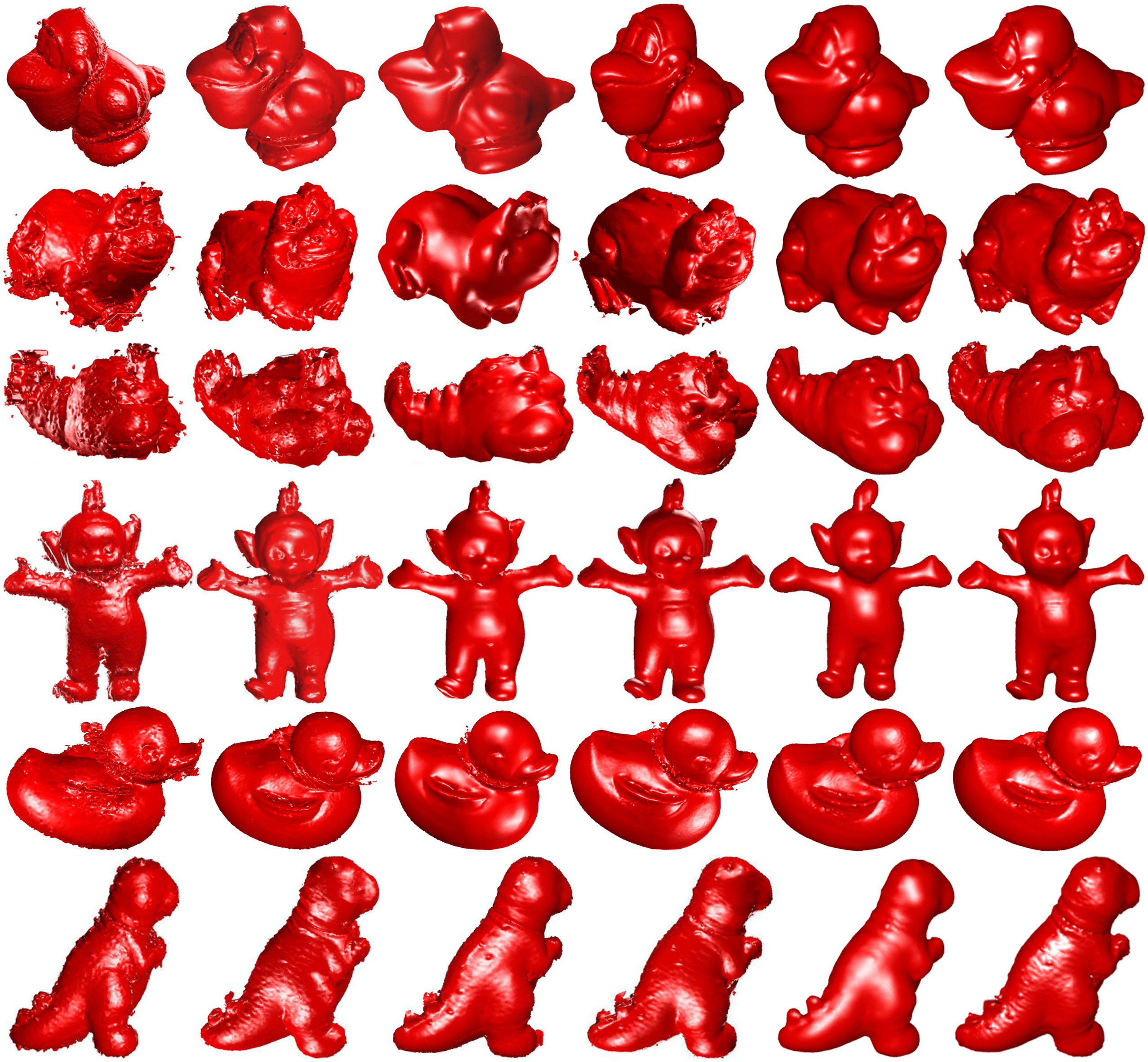

Figure 4. Rows: Reconstruction results of 18 'Bird' images, 18 'Frog' images, 18 'Lobster' images, 20 'Teletubby' images, 18 'Duck' images and 8 'Dinosaur' images. From left to right: volumetric method [5], mesh-based method [22], FCM clustering [29], k-means clustering [28], pairwise MRF [17], higher-order CRF

point-to-cloud-to-point labeling is a coarse-to-fine scheme which greatly speeds up the algorithm.

Also, we utilize the sparseness of the CRF to further reduce the costs. In [13], 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 $O(mh^3)$. Here, when we update m_{ji} and m_{kji}, we only consider the h labels corresponding to the point clouds covering the area around the vertex i, subject to the one-point potential defined by Eq. (7). The two-point and three-point cliques become 'sparse' (i.e. many labelings are unlikely) and the cost for the inference is $O(nh^2(h+1)T)$. For a data sets containing 18 point clouds ($m = 18$), h is usually equal to 6.

VII. Experimental results

We performed all experiments using multi-view registered range images. They are actually 3D unstructured point clouds due to the loss of the original lattice structure (see Subsection 3.1) and doing so also allows direct comparison with other methods using the same datasets. We used data from the well-known OSU/SAMPL Range Image Database.

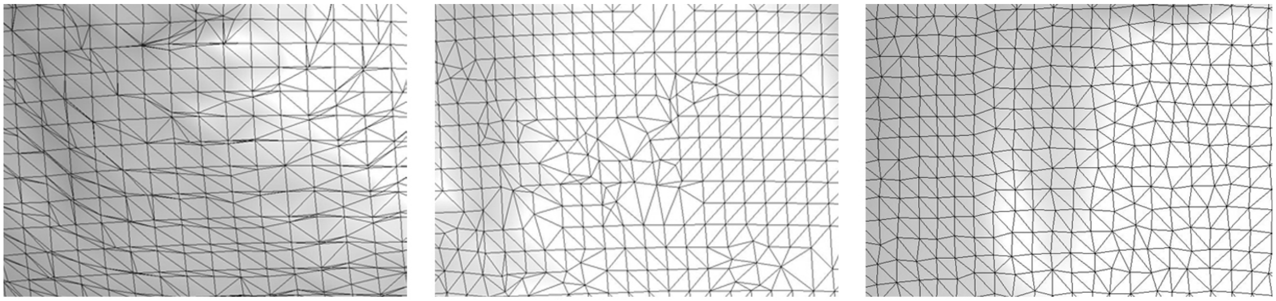

Figure 5. Meshes for integrated surfaces produced using, from left to right: volumetric method, mesh-based method, higher-order CRF.

The 'Bird', 'Frog', 'Lobster', 'Teletubby', 'Duck' and 'Dinosaur' image sets have 156210, 181866, 176806, 105932, 268398 and 62558 points respectively. We employed the algorithm in [14] to perform pairwise registration. Average registration error is about 0.3mm for the first five data sets—approximately half of the scanning resolution and that of the 'Dinosaur' sets is as high as 0.6mm.

In our tests, the truncation parameter A was set to 4 and the weighting parameters λ_1 and λ_2 were set to 3 and 1 respectively. Note that it is not possible to obtain real test data with ground truth, a common problem in assessing automatic multi-view integration problems as perfect registration is not available.

We have thus evaluated our method by comparison to competing methods. Fig. 4 show integration results produced by existing methods and our higher-order CRF method. They demonstrate that, of the methods compared, our method is more robust to registration errors and gives the most geometrically realistic surface models.

As ground truth data are not available, it is difficult to define applicable metrics for *accuracy* and *completeness* [21], to give a quantitative comparison. Note that either *accuracy* or *integration error* [28] (the average of squared distances between the output points and their closest points in the input range images) just measures the *global* accuracy of the reconstruction. Clearly, in terms of *local* accuracy, our method performs best because a local patch of the reconstructed surface is directly taken from a certain input point cloud. The integration error proposed in [28] is not applicable in this work as it always is 0 for our method obviously. Instead, we propose a new method to measure the integration error. For each point in the output point cloud, we compute the average of the distances between it and a set of points in the input point clouds whose distances to it are smaller than a threshold set as the scanning resolution R here. If no distance is smaller than R, we directly set the average as R. Then the integration error of an output point cloud is the average of these per-point averages. Also, we have adopted some widely used measurement parameters which do not require the ground truth. Firstly, we measured the distribution of interior angles of triangles [28], to help assess the quality of the final mesh—ideally, interior angles should be close to 60°. Secondly, we computed the average distortion metric [29] for each triangle, 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 the distortion metric lies in [0,1]. The higher the average distortion metric value, the higher the quality of a surface. Finally we measured the computation time. Figs. 5 and 6 show that our new method performs best in terms of the integration error, the distribution of interior angles, and distortion metric, but it is slower than k-means clustering and pairwise CRF methods. All experiments used a Pentium IV 2.4GHz computer. Note that we do not use a segmentation scheme, thus saving time compared to those methods which perform segmentation before integration [29].

VIII. Conclusion and future work

We have given a higher-order CRF model for surface reconstruction from multi-view data sets. The CRF is configured on a specific graph which can handle 3D unstructured point clouds. We infer the CRF via belief propagation, and produce geometrically realistic surface models with well preserved local details automatically and robustly.

However, a couple of works can still be done to improve the quality of integration in the future. We found that not all the normals are reliable due to scanning noise and registration errors. In particular, for the points from the registered image, their normals are more likely to be inaccurate. Therefore, we are looking for a more robust property attached to the local surface around the point of interest. A promising method is to develop a feature descriptor bounded at each point rather than only the feature points (e.g., the likes of 2.5D SIFT [16]).

Another approach focuses on reducing the accumulated registration errors. A 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.

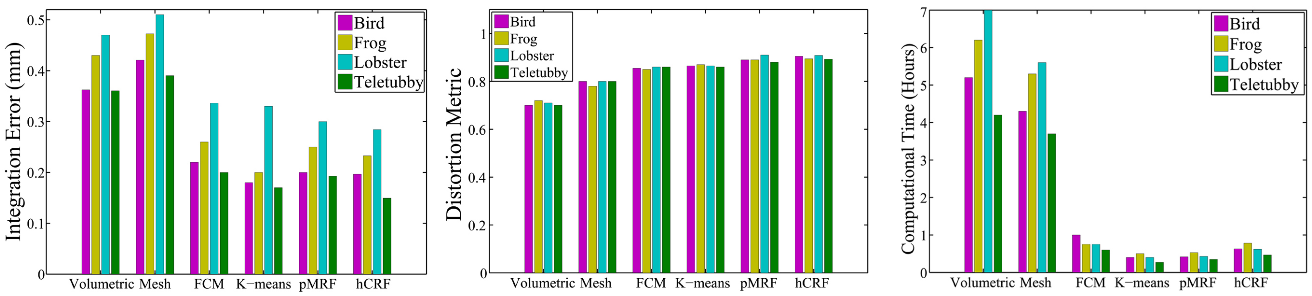

Figure 6. Performance measures for integration algorithms. From left to right: integration error, distortion metric, computational time

Thus we plan to develop a scheme that can make a good balance between the extra cost caused by more pairwise registrations and the reduction of error accumulation.

Also, future work will focus on improving the speed.

Acknowledgment

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

References

[1] K. Alahari, C. Russell, and P. Torr. Efficient piecewise learning for conditional random fields. In *Proc. CVPR*, 2010.

[2] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt. 3d shape scanning with a time-of-flight camera. In *Proc. CVPR*, 2010.

[3] J. Diebel and S. Thrun. An application of markov random fields to range sensing. In *Proc. NIPS*, 2006.

[4] J. Diebel, S. Thrun, and M. Brunig. A bayesian method for probable surface reconstruction and decimation. *ACM Transactions on Graphics*, 25(1):59, 2006.

[5] C. Dorai and G. Wang. Registration and integration of multiple object views for 3d model construction. *PAMI*, 20:83–89, 1998.

[6] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.

[7] http://vision.middlebury.edu/mview/eval/.

[8] Q. Huang, B. Adams, and M. Wand. Bayesian surface reconstruction via iterative scan alignment to an optimized prototype. In *Proc. Eurographics symposium on Geometry processing*, 2007.

[9] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Strasser. Bayesian point cloud reconstruction. *Computer Graphics Forum*, 25(3):379–388, 2006.

[10] P. Kohli, M. Kumar, and P. Torr. p^3 beyond: Solving energies with higher order cliques. In *Proc. CVPR*, 2007.

[11] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004.

[12] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *CVPR*, 2009.

[13] X. Lan, S. Roth, D. Huttenlocher, and M. Black. Efficient belief propagation with learned higher-order markov random fields. *Proc. ECCV*, pages 269–282, 2006.

[14] Y. Liu. Automatic 3d free form shape matching using the graduated assignment algorithm. *Pattern Recognition*, 38(10):1615–1631, 2005.

[15] Y. Liu. Automatic range image registration in the markov chain. *PAMI*, 32(1):12–29, 2010.

[16] T. Lo and J. Siebert. Local feature extraction and matching on range images: 2.5d sift. *Computer Vision and Image Understanding*, 113(12):1235–1250, 2009.

[17] R. Paulsen, J. Bærentzen, and R. Larsen. Markov random field surface reconstruction. *IEEE Trans. Visual. Comput. Graph.*, 16(4):636–646, 2010.

[18] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *Proc. CVPR*, 2009.

[19] M. Rutishauser, M. Stricker, and M. Trobina. Merging range images of arbitrarily shaped objects. In *Proc. CVPR*, 1994.

[20] R. Sagawa, K. Nishino, and K. Ikeuchi. Adaptively merging large-scale range data with reflectance properties. *PAMI*, 27:392–405, 2005.

[21] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*, 2006.

[22] Y. Sun, J. Paik, A. Koschan, and M. Abidi. Surface modeling using multi-view range and color images. *Int. J. Comput. Aided Eng.*, 10:137–50, 2003.

[23] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *PAMI*, 30(6):1068–1080, 2008.

[24] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *PAMI*, 31(12):2115–2128, 2009.

[25] J. Yedidia, W. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Techincal Report TR-2001-22, MERL, 2002.

[26] Z. Yin and R. Collins. Belief propagation in a 3d spatio-temporal mrf for moving object detection. In *Proc. CVPR*, pages 1–8, 2007.

[27] B. H. Zach Christopher, Pock Thomas. A globally optimal algorithm for robust tv-l1 range image integration. In *Proc. ICCV*, 2007.

[28] H. Zhou and Y. Liu. Accurate integration of multi-view range images using k-means clustering. *Pattern Recognition*, 41(1):152–175, 2008.

[29] H. Zhou, Y. Liu, L. Li, and B. Wei. A clustering approach to free form surface reconstruction from multi-view range images. *Image and Vision Computing*, 27(6):725–747, 2009.