

Saliency-Guided Integration of Multiple Scans

Ran Song, Yonghuai Liu,
Department of Computer Science
Aberystwyth University, UK
{res,yy1}@aber.ac.uk

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

Abstract

We present a novel method to integrate multiple 3D scans captured from different viewpoints. Saliency information is used to guide the integration process. The multi-scale saliency of a point is specifically designed to reflect its sensitivity to registration errors. Then scans are partitioned into salient and non-salient regions through an Markov Random Field (MRF) framework where neighbourhood consistency is incorporated to increase the robustness against potential scanning errors. We then develop different schemes to discriminatively integrate points in the two regions. For the points in salient regions which are more sensitive to registration errors, we employ the Iterative Closest Point algorithm to compensate the local registration error and find the correspondences for the integration. For the points in non-salient regions which are less sensitive to registration errors, we integrate them via an efficient and effective point-shifting scheme. A comparative study shows that the proposed method delivers improved surface integration.

1. Introduction

Due to the development of laser scanning techniques, 3D reconstruction of a surface model from multiple scans has gained considerable attention. Such a reconstruction usually comprises three main steps: (1) scanning object surface from various viewpoints, (2) registering the scans into a common coordinate system, and (3) integrating the scans to produce a single composite model. While much attention has been paid to the second problem, rather less has been given to the third step. Although recent high precision laser scanners can scan surfaces of objects with a high accuracy, scanning errors caused by sensing noise, outliers and occlusions are still inevitable. Generally, the cheaper the scanner, the less accurate and more noisy the captured data. Furthermore, further errors are introduced by mis-registration: registration errors remain even when using state-of-the-art automatic 3D registration methods [12, 13, 20]. Integration should be robust in the presence of these errors.

1.1. Related work

Existing integration methods can be divided into four main groups: volumetric, mesh-based, point-based and segmentation-based approaches.

Volumetric methods such as [6, 8, 19] integrate data by voxelising them and then merging them in each voxel using data fusion algorithms. These methods require highly accurate registration (often estimated via manually-assisted camera calibration, or simply assumed to be given as known input. In practice, volumetric methods often work poorly or even fail in the presence of typical registration errors, a problem demonstrated both theoretically and experimentally in [25].

Mesh-based methods such as [18, 22, 24] 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 mesh vertices and are more geometrically complex. Some mesh-based methods thus just use a 2D triangulation for efficiency, but 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 is also not robust to registration errors [26].

Point-based methods like [16, 25] operate on points only. The detection of correspondences is performed through point repositioning where potential corresponding points are usually moved closer to each other. Then the detected corresponding points in overlapping areas are merged directly or via clustering. Point-based methods are relatively efficient because all processes are based on only points, of which there are significantly fewer than triangles. However, the integrated surface tends to be rough due to scanning noise and registration errors.

Segmentation-based methods such as [7, 26] partition or decompose the input data into different categories and employ different integration strategies to process the data in different categories. The idea is based on the fact that different types of data have different properties and thus one integration scheme may not be applicable to all data. In

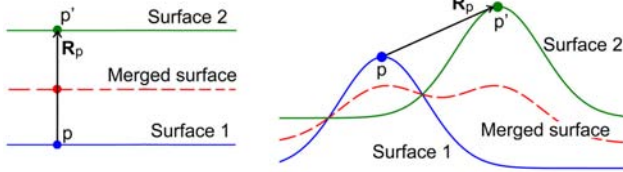


Figure 1. During integration, the same magnitude of registration error can have different effects in different regions. Left: a registration error \mathbf{R}_p in a non-salient region. Right: a registration error with the same magnitude in a salient region. A surface formed by averaging the input surfaces (a trivial approach to integration) gives good results in the non-salient region but not the salient region.

[26], principal component analysis (PCA) is employed to segment a base surface into non-featured and featured areas and then use fuzzy- c means and k -means clustering approaches to integrate the points in the two areas separately. The problem of this method is that the PCA-based segmentation essentially relies on thresholding and is thus not reliable in the presence of scanning noise and registration errors. As a result, some featured points are wrongly classified as non-feature points, often leading to an oversmoothed surface after the integration. In [7], input scans are decomposed into high and low frequency components and only the low frequencies are fused while the high frequency contents are kept intact. However, this method requires highly accurate registration.

1.2. The proposed work

Most existing integration methods are not robust in the presence of registration errors which may move real corresponding points away from each other and outliers closer to each other. Usually, registration methods merely seek to minimise such registration errors. However, Fig. 1 illustrates the fact that the same magnitude of registration errors have significantly different effects on the integration in salient and non-salient regions. In other words, during integration, points in salient regions are more sensitive to registration errors than ones in non-salient regions. Therefore, it is natural to consider partitioning scans into salient and non-salient regions and then using a robust strategy to integrate points in salient regions while using a less robust but more efficient strategy to integrate points in non-salient regions. Furthermore, a simple thresholding-based segmentation is not robust because saliency or ‘featureness’ [26] values are usually not reliable in the presence of scanning noise and registration errors. We thus employ an Markov Random Field (MRF) modeling neighbourhood consistency for a robust segmentation. Here, the neighbourhood consistency is based on the fact that if the neighbours of a point are salient/non-salient, this point is likely to be so as well.

By combining the aforementioned ideas, we propose

a novel method for the robust integration of multiple 3D scans. Firstly, saliency detection is performed by estimating the multi-scale representation of each input scan. Secondly, the detected saliency information is incorporated into an MRF framework and the Belief Propagation (BP) algorithm is employed to solve this MRF, partitioning scans into salient and non-salient regions. Thirdly, for the points in salient regions, we employ the Iterative Closest Point (ICP) algorithm to adjust their positions and then integrate them; for the points in non-salient regions, we integrate them via a point shifting scheme. The final output of the proposed integration method is a single point cloud. To render the point cloud as a watertight surface, a triangulation algorithm is necessary although this non-trivial technique is out of the scope of this paper.

2. Saliency detection

We first perform saliency detection for each scan. It involves two stages: 3D scale space construction and multi-scale saliency estimation.

2.1. 3D DoG scale space

SIFT [14] employs the Difference-of-Gaussians (DoG) operator to construct a 2D scale space. In this paper, we extend this method to 3D to construct a 3D scale space.

We apply a bank of S Gaussian filters on a scan M to produce a multi-scale representation D_s for M . $G(p, \sigma)$ is a Gaussian kernel with standard deviation σ centred at the point $p \in M$. Each Gaussian kernel is applied over a spherical region centred at p with a radius r . All points within this region are viewed as the neighbours of p and involved in the convolution. In this work, we set $r = 2.355 * F * \sigma$ in line with the principle of *full width at half maximum* where F is a normalisation parameter related to the scanning resolution R (average inter-point distance) of the scan M and we choose $F = 2R$. This neighbourhood region can be viewed as a good approximation of a geodesic region of radius r . We propose a new algorithm for the Gaussian filtering adaptive to the number of detected neighbours of each point:

- For a point p , find all of its k_p neighbours (including itself) within a distance equal to r from all of the points in the scan M .
- Sort the k_p neighbours in the descending order of the distance to p to produce a k_p dimensional vector v_p . So the first element in the vector is p itself and the last one is the point furthest from p .
- For the neighbourhood of each p , construct a discrete Gaussian kernel with standard deviation σ sampled as a k_p -dimensional vector.
- Sort the elements in this Gaussian kernel in descending order, yielding an adaptive Gaussian kernel G_p . Thus

in the following convolution, nearer neighbours have more weights.

- Do convolution using v_p and G_p .
- Repeat the steps listed above for all points on M .

After the Gaussian filtering, the 3D DoG scale space is constructed by computing the difference of a pair of layers at scale s :

$$D_s(p) = G_p(p, \sigma_s) - G_p(p, \eta\sigma_s), \quad s = 1, 2, \dots, S \quad (1)$$

where G_p denotes the Gaussian applied to the point p . η is set as 1.6, which makes the DoG a good approximation of the Laplacian of Gaussian (LoG). By ‘approximation’, we mean $DoG(x)/LoG(x) \approx \text{constant}$ or the DoG is approximately equal to the scale-normalised LoG which can achieve true scale invariance.

A reasonable balance between reliability of saliency detection and computational cost (especially for the following MRF labeling) is achieved, in our experience, by using four scales of filtering with $\sigma_s \in \{0.6, 1.2, 1.8, 2.4\}$. Note that an optimal σ_{best} can be calculated using the method proposed in [2]. Other papers using different 3D Gaussian filters suggest that σ should be chosen according to the size of the object. For example, in [11], values used are $\sigma_s = \{2\epsilon, 3\epsilon, 4\epsilon, 5\epsilon, 6\epsilon\}$ where ϵ is 0.3% of the length of the diagonal of the bounding box of the model. Using such parameter settings, σ could be rather small if the object is very small in size. This may not be problematic for the original paper as its aim is saliency detection but it will result in difficulty during MRF labeling later. The reason is that a steep Gaussian puts too much weight on the centre point, leading to similar saliency values at different scales. The one-point cost measured by saliency difference in Eq. (4) can thus be ambiguous and consequently the labeling result will not represent a meaningful segmentation. Here, we fix the values of σ_s for easy implementation as our tests show that in most cases, an optimal value σ_{best} does not change the segmentation result too much although it has a significant effect on the resultant saliency map.

2.2. Multi-scale saliency estimation

$D_s(p)$ is a 3D vector representing the displacement of the point p from its original position in M after the filtering. Note that the effect of a registration error at a point is essentially an unexpected displacement of the point (to move it away from its corresponding point). This is the reason that we define the saliency in a scale-dependent manner on its Gaussian-weighted 3D position rather than mean curvature as in [11]. The displacement more directly indicates the sensitivity to registration error which is the key to our integration algorithm. Such a displacement is related to local surface geometry (i.e., the positions of the neighbours). We estimate a Gaussian-weighted value in a neighbourhood to

measure such a sensitivity to displacement, which is based on the fact that if one point is poorly registered, its neighbours are likely to be so as well. To reduce it in a scalar quantity, we project the vector $D_s(p)$ onto the normal $n(p)$ at the point p to obtain the scale map M_s :

$$M_s(p) = \|n(p) \cdot D_s(p)\|, \quad s = 1, 2, 3, 4 \quad (2)$$

This works because: (i) such a projection indeed reduces the displacement in a scalar quantity as $M_s(p) \leq \|D_s(p)\|$, (ii) the displacement of a point in the direction of normal will not change (or only slightly change) the size of the mesh, reducing the shrinking effect which rises typically when we do Gaussian filtering to 3D meshes [17], and (iii) the surface topology is retained as the displacement is along the normal.

Then, we employ the method proposed in [10] to normalise the scale maps. Firstly, the values in each map are normalised to a fixed range in order to eliminate modality-dependent amplitude differences. Secondly, the location of the map’s global maximum A is detected. Thirdly, the average \bar{a} of all other local maxima is computed and the map is globally multiplied by $(A - \bar{a})^2$.

The normalisation is designed to globally promote maps in which a small number of strong peaks (corresponding to salient locations) are present, while globally suppressing maps which contain numerous comparable peaks. To further enhance the difference between salient and non-salient locations, we apply an arc-tangent operation to each scale map to produce the final saliency map \hat{M}_s . These maps indicate the degree to which points in different locations have different sensitivity to potential registration errors in terms of the effects on the final integration results.

3. Saliency-guided segmentation

Once the multi-scale saliency map $\hat{M}_s, s \in \{1, 2, 3, 4\}$ is obtained, previous methods either (i) directly save all the information (all scales, all neighbourhoods, all saliency values, etc) attached to one point which can be used to produce a descriptor (often a vector of high dimension), or (ii) output a single-saliency map by simply computing the sum or the average of all multi-scale saliency maps to simplify the information. The first type of methods are good at producing a distinctive descriptor for each point, but such approaches result in high computational cost. For speed, some researchers thus reduce the dimension of the descriptor vector in some way, by eliminating unimportant information. This typically leads to a trade-off between quality of results and speed. The second type of methods [5, 11] are fast but do not make good use of the information embedded in the multi-scale saliency maps. In particular, the saliency of each point is no longer particularly distinctive. Furthermore, range scans normally included spikes, holes, or even cluttered background. Such scanning noise may

lead to unreliable saliency detection. Thus, to more robustly judge whether a point is salient or not, we should not only consider its detected saliency, but also investigate its neighbours as neighbouring points are likely to have consistent properties. In the proposed method, the neighbourhood consistency is incorporated through an MRF framework.

In the MRF, the label set comprises the scale indices $\{s\} = \{1, 2, 3, 4\}$. For a point p , each label has a corresponding scale saliency $\{\hat{M}_s(p)\}$ for the scale s . In line with the standard MRF formulation where the points/sites are written as subscripts and the labels assigned to the points are variables to be determined, in the rest of this paper, we rewrite the saliency $\{\hat{M}_s(p)\}$ as $\{\hat{M}_p(s)\}$. The MRF energy function can be expressed as:

$$E(s) = \sum_{p \in M} E_p(s_p) + \beta \sum_p \sum_{q \in \mathcal{N}_p(s)} E_{pq}(s_p, s_q) \quad (3)$$

where β is a weighting parameter and the $\mathcal{N}_p(s)$ denotes the neighbourhood of p at scale s as used in Gaussian filtering, but now without p itself).

3.1. Observation term

In Eq. (3), the observation term $E_p(s_p)$ is a one-point cost function associated to the state (label) that we are most likely to observe at point p . It is defined as the difference between the saliency corresponding to a certain scale and the largest saliency at p :

$$E_p(s_p) = \left| \hat{M}_p(s_p) - \max_s \hat{M}_p(s) \right|, \quad s = 1, 2, 3, 4 \quad (4)$$

It can be seen that the label $\hat{s}_p = \arg \max_s (\hat{M}_p(s))$ always produces the lowest one-point labeling cost at p in such a model.

3.2. Compatibility term

The compatibility term captures the label compatibility between two neighbouring points in a pairwise MRF model. It can be regularised by the general and scene-specific knowledge. For instance, the smoothness prior, essentially an intensity consistency strategy applied to a neighbourhood, is widely used in 2D applications such as image segmentation, restoration and depth estimation, etc. In this work, we carry out a scale consistency strategy which encourages two neighbouring points to take the same label since neighbouring points are likely to have the same properties. The compatibility term is the widely used Potts model [3]:

$$E_{pq}(s_p, s_q) = \begin{cases} 1, & s_p \neq s_q, \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

3.3. Inference via belief propagation

We solve this MRF in accordance with the *maximum a posteriori probability* (MAP) criterion. The MAP criterion requires the minimisation of the MRF posterior energy expressed in Eq. (3) derived from the negative log-likelihood of the posterior probability. Therefore, our aim is to find a label assignment $s^* = \{s_p^* | p \in M\}$ which minimises the energy in Eq. (3). Several methods have been developed to solve such a problem [23]. We employ the *belief propagation* (BP) method here, which works by passing messages in an MRF network. It is briefly summarized as follows:

1. For all point pairs (p, q) where $q \in \mathcal{N}(p)$, initialising message m_{pq}^0 to zero, where m_{pq}^t is a vector of dimension given by the number of possible labels S ($S = 4$ in this work) and denotes the message that point p sends to a neighbouring point q at iteration t .
2. For $t = 1, 2, \dots, T$, updating the messages as

$$m_{pq}^t(s_q) = \min_{s_p} \left(E_p(s_p) + \beta E_{pq}(s_p, s_q) + \sum_{h \in \mathcal{N}(p) \setminus q} m_{hp}^{t-1}(s_p) \right). \quad (6)$$

$\mathcal{N}(p) \setminus q$ denotes the neighbours of p other than q .

3. After T iterations, computing a belief vector,

$$b_q(s_q) = E_q(s_q) + \sum_{p \in \mathcal{N}(q)} m_{pq}^T(s_q) \quad (7)$$

and then determining labels at each point as:

$$s_q^* = \arg \min_{s_q} (b_q(s_q)) \quad (8)$$

As can be seen from Eq. (6), the computational complexity for the calculation of the message vector is $O(S^2)$ as we need to minimise over s_p for each choice of s_q . By a simple analysis, we can reduce it to $O(S)$, as shown below:

1. Rewriting Eq. (6) as,

$$m_{pq}^t(s_q) = \min_{s_p} \left(\beta E_{pq}(s_p, s_q) + f(s_p) \right) \quad (9)$$

$$\text{where } f(s_p) = E_p(s_p) + \sum_{h \in \mathcal{N}(p) \setminus q} m_{hp}^{t-1}(s_p)$$

2. Considering two cases: $s_p = s_q$ and $s_p \neq s_q$

If $s_p = s_q$, $\beta E_{pq}(s_p, s_q) = 0$, thus $m_{pq}^t(s_p) = f(s_q)$

If $s_p \neq s_q$, $\beta E_{pq}(s_p, s_q) = \beta$, $m_{pq}^t(s_q) = \min_{s_p} f(s_p) + \beta$

3. Synthesizing the two cases to give:

$$m_{pq}^t(s_q) = \min \left(f(s_q), \min_{s_p} f(s_p) + \beta \right) \quad (10)$$

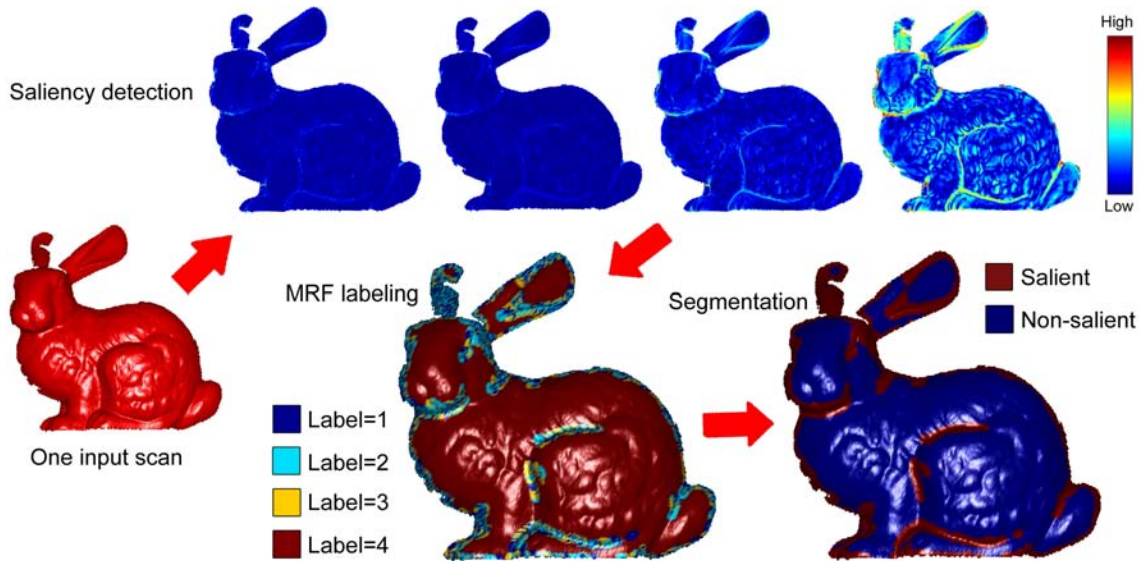


Figure 2. Saliency detection, MRF labeling and segmentation of one scan of the Stanford Bunny.

Using Eq. (10), the minimisation over s_p need be performed only once, independent of the value of s_q . In other words, Eq. (6) needs two nested FOR loops to compute the messages but Eq. (10) just needs two independent FOR loops. Therefore, the computational complexity is reduced from $O(S^2)$ to $O(S)$.

We then convert the label assignment to a segmentation. Typically, most points are assigned the same label as shown in Fig. 2, and these points comprise the non-salient regions. All other points comprise the salient regions.

4. Point shifting and local ICP

In this section, we give the details of the integration scheme guided by the saliency-determined segmentation. Given two registered 3D scans (3D point clouds), P_1 and P_2 , we use two schemes to integrate points in salient and non-salient regions respectively.

Integration in non-salient regions. For the points in non-salient regions, we present an efficient point shifting method modified from the centroid initialisation algorithm in [25] to integrate them. First, the overlapping and non-overlapping areas of P_1 and P_2 are efficiently detected. 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 to speed up the search. The threshold is set to $3R$, where R is the scanning resolution of the input scans. This threshold is generally large enough not to miss any real correspondence between the overlapping scans to be integrated when their registration is reasonably accurate [25].

After detecting the overlap, we compute the normals for

the points in overlapping areas and set S_1 and S_2 to denote the points in the non-overlapping areas belonging to P_1 and P_2 respectively. Next, we compute a point set S_{overlap} which represents the integrated points for the points from both overlapping areas. To bring the corresponding points closer to each other, each point \mathbf{P} in the overlapping areas is shifted along its normal \mathbf{N} towards its corresponding point \mathbf{P}^* by half of its distance to \mathbf{P}^* :

$$\mathbf{P} \rightarrow \mathbf{P} + 0.5\mathbf{d} \cdot \mathbf{N}, \quad \mathbf{d} = \Delta\mathbf{P} \cdot \mathbf{N}, \quad \Delta\mathbf{P} = \mathbf{P}^* - \mathbf{P} \quad (11)$$

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. The set of all such positions forms the point set S_{overlap} . Then the integration result in non-salient region is $P_{\text{non-salient}} = S_{\text{overlap}} \cup S_1 \cup S_2$.

This strategy (i) compensates for 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.

Integration in salient regions. For the points in salient regions, the aforementioned point shifting scheme is not suitable (again, see Fig. 1) – it usually leads to an over-smoothed surface. We thus use the *iterative closest point* (ICP) algorithm to reposition these points to reduce the errors caused by inaccurate registration. Although ICP is a classical method to register entire scans, it has also been used for local registration [4]. In our method, the ICP is merely applied to the points in salient regions. Doing so has four advantages. First, registering the whole dataset is

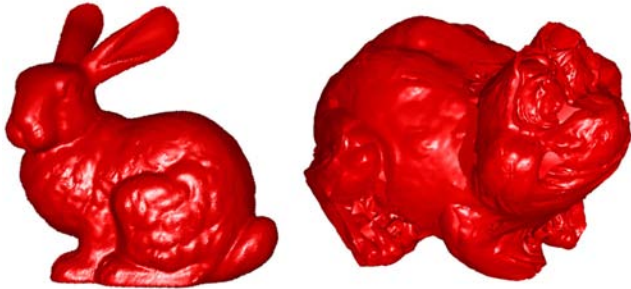


Figure 3. Left: Joint meshes of 10 Bunny scans without integration; Right: Joint meshes of 17 Frog scans without integration

more difficult as it usually includes noise such as outliers or even clutters. An ICP applied to local salient regions is less likely to suffer from such noise. Second, initial transformation between neighbouring scans is usually good enough for refinement. Thus the local ICP here is more likely to produce reliable result. Thirdly, the local ICP offers a more accurate registration for the salient points which are visually more important than the non-salient points. It is essentially a desired error distribution which usually leads to a visually better integration. Fourthly, a local ICP is more efficient than a global ICP as fewer points are involved.

In detail, for the points in salient regions, we first detect the overlapping area using the same scheme as for integration of non-salient regions. Then, we employ the ICP algorithm to align the points from P_1 in the overlapping area with the points from P_2 in the overlapping area. Note that in this local registration, we do not only reposition some points from P_1 , but also, as a byproduct, find the correspondences between some repositioned points from P_1 and some points from P_2 in the overlapping areas. We then simply integrate each pair of corresponding points by averaging to obtain the integrated point set S_{overlap} . The points S_1 and S_2 in non-overlapping areas remain unchanged. Similarly, the integration result in non-salient region is $P_{\text{salient}} = S_{\text{overlap}} \cup S_1 \cup S_2$.

Finally, we obtain the integrated point cloud $P_{\text{integrated}} = P_{\text{non-salient}} \cup P_{\text{salient}}$ for the input point clouds P_1 and P_2 . Then we can do the integration for $P_{\text{integrated}}$ and the next input 3D scan P_3 . After all input scans have been integrated through this procedure, a single integrated point set is obtained. We then employ the power crust method [1] to triangulate the final integrated points to make a mesh.

5. Experimental results

In the experiments, we test 6 sets of multi-view 3D scans. The Bird (156094 points, 17 scans), the Frog (174097 points, 17 scans), the Lobster (176906 points, 18 scans), the Teletubby (90848 points, 17 scans) and the Duck (220560 points, 18 scans) datasets were obtained from the Minolta Database [9] and the Bunny dataset (362230 points, 10

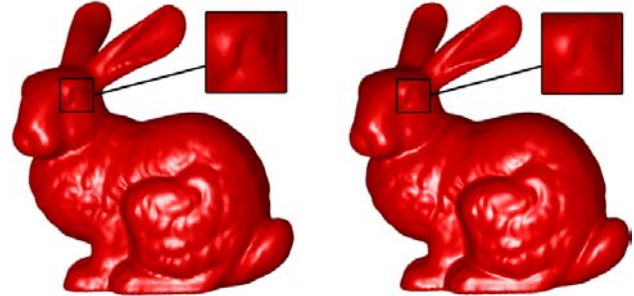


Figure 4. Left: Integration of the Bunny scans using the proposed method; Right: Integration using the SFK method [26]

scans) was obtained from the Stanford 3D Scanning Repository. The Bunny dataset was captured at high resolution and highly accurate alignment parameters are given. The more noisy Minolta scans are captured at much lower resolutions and the alignment parameters are not given. We thus employed the state-of-the-art algorithms proposed in [13] and [15] to perform automatic pairwise and global registration for these scans. The registered multi-view 3D scans were then used as the input data for our experiments. As shown Fig. 3, the Minolta scans contain larger registration and scanning errors. Note that existing integration methods usually require the registration error to be an order of magnitude lower than the measurement error. However, for the Minolta scans, this assumption is not satisfied. As demonstrated in Table 1, the average registration errors are about 1/3 to 1/2 of the scanning resolution. Fig. 3 visualises the different scales of registration errors within different datasets by showing joint meshes of all (registered) input scans without integration.

Most integration methods can produce a good surface model for the well-registered Bunny scans. For example, Fig. 4 compares our integration method with the segmentation-based method [26] (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 details (see the eye of the bunny).

The more challenging Minolta scans are widely used for comparing methods. As shown in Fig. 5 and 6, different methods performs significantly differently. In sharp contrast, our method produces clear eyes, mouth and wings for the bird, eyes, fingers and pocket (on the chest) for the teletubby, and toes, eyes, and mouth for the frog, etc. In general, the volumetric method fails to produce a clean surface model. The mesh-based method and the k -means clustering produce improved surface models but also sometimes generate fragments (see the frog’s toes, 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



Figure 5. Rows: Integration results of Bird and Teletubby scans. From left to right: one input scan (ground truth), volumetric method [8], mesh-based method [22], k -means clustering [25], pairwise MRF [16], SFK [26], our method



Figure 6. Columns: Integration results of Frog, Lobster and Duck scans. From top to bottom: one input scan (ground truth), volumetric method [8], mesh-based method [22], k -means clustering [25], pairwise MRF [16], SFK [26], our method

	RS (mm)	RE (mm)	SDRE (mm)	3DGC1	3DGC2	3DGC3
Bird	0.66	0.28	0.12	0.1173	0.3228	0.2189
Frog	0.63	0.30	0.15	0.1557	0.2276	0.2021
Teletubby	0.61	0.25	0.12	0.0619	0.2557	0.1238
Lobster	0.68	0.30	0.16	0.1836	0.3308	0.3190
Duck	0.73	0.33	0.12	0.1235	0.1723	0.1655

Table 1. RS, average of resolutions of all range scans; RE, average registration error over reciprocal correspondences [13]; SDRE, standard deviation of registration errors; 3DGC1, average 3D Gini coefficient of the surface model produced by our method; 3DGC2, average 3D Gini coefficient of the surface model produced by the pairwise MRF [16]; 3DGC3, average 3D Gini coefficient of the surface model produced by the k -means clustering [25].

usually produce a clean surface.

We have also made quantitative comparisons by computing 3D Gini coefficients [21]. The 3D Gini coefficient is a quantitative version of the comparison shown in Fig. 5 and 6 where we use a single scan as the ground truth and compare it with its corresponding region of the integrated surface model. The 3D Gini coefficient quantitatively measures the mesh similarity between one input scan (taken as ground truth) and its corresponding surface region of the integration by computing the cumulative distribution of the joint probability of two transformed curvatures. The smaller the 3D Gini coefficient, the more similar the input scan with its corresponding region of the integrated surface. Overall integration quality is given by averaging the 3D Gini coefficients between the integrated surface model and each input scan. The smaller the average 3D Gini coefficient, the better the integration.

Fig. 5 and 6 qualitatively demonstrate that our new method, pairwise MRF [16] and k -means clustering [25] seem most successful. Therefore, in the quantitative com-

parison, we compute the average 3D Gini coefficients of the surface models produced by the three integration methods. The results are shown in the last three columns of Table 1. For each Minolta dataset, our new method achieves the smallest average 3D Gini coefficient. In general, according to the average 3D Gini coefficient, our new method (corresponding to 3DGC1) achieves the best result, k -means clustering (corresponding to 3DGC3) is next best, and pairwise MRF (corresponding to 3DGC2) is worst.

All experiments used a dual core, 2.4GHz, 3.25GB RAM PC. The integration of each dataset took 15–30 minutes, mainly depending on the number of the points in the datasets and the number of iterations that the BP algorithm used to reach convergence or an acceptable solution.

6. Conclusion

We present a saliency-guided method for the robust integration of multiple 3D scans. Its novelty is twofold. On the one hand, using the specifically defined saliency to guide a segmentation-based integration achieves the robustness to large registration errors. On the other hand, incorporating the saliency information into an MRF for the segmentation increases the robustness to potential scanning noise. While most existing methods uniformly treat data to be integrated, we first partition input scans into salient and non-salient regions and then integrate 3D points in different regions using different strategies. A comparative study using 6 models with altogether 97 scans from two well-known databases shows that the proposed method outperforms the selected state of the art ones. Future work will concentrate on the robust and efficient integration of all input scans simultaneously using global optimisation.

References

- [1] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *Proc. the Sixth ACM Symposium on Solid Modeling*, pages 249–260, 2001.
- [2] Y. Belyaev and H. Seidel. Mesh smoothing by adaptive and anisotropic gaussian filter applied to mesh normals. In *Vision, modeling, and visualization*, pages 203–210, 2002.
- [3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.
- [4] B. Brown and S. Rusinkiewicz. Global non-rigid alignment of 3-d scans. *ACM Transactions on Graphics*, 26(3):21–es, 2007.
- [5] U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Computer Graphics Forum (Eurographics 2008)*, 27(2):643–652, 2008.
- [6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH*, pages 303–312, 1996.
- [7] J. Digne, J. Morel, N. Audfray, and C. Lartigue. High fidelity scan merging. *Computer Graphics Forum*, 29(5):1643–1651, 2010.
- [8] C. Dorai and G. Wang. Registration and integration of multiple object views for 3d model construction. *PAMI*, 20(1):83–89, 1998.
- [9] P. Flynn and R. Campbell. A www-accessible 3d image and model database for computer vision research. *Empirical Evaluation Methods in Computer Vision*, pages 148–154, 1998.
- [10] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11):1254–1259, 1998.
- [11] C. Lee, A. Varshney, and D. Jacobs. Mesh saliency. In *ACM SIGGRAPH 2005 Papers*, pages 659–666. ACM, 2005.
- [12] Y. Liu. Replicator dynamics in the iterative process for accurate range image matching. *IJCV*, 83(1):30–56, 2009.
- [13] Y. Liu. Automatic range image registration in the markov chain. *PAMI*, 32(1):12–29, 2010.
- [14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [15] K. Nishino and K. Ikeuchi. Robust simultaneous registration of multiple range images. In *Proc. ACCV*, 2002.
- [16] R. Paulsen, J. Bærentzen, and R. Larsen. Markov random field surface reconstruction. *IEEE Trans. on Visualization and Computer Graphics*, 16(4):636–646, 2010.
- [17] M. Pauly, R. Keiser, and M. Gross. Multi-scale feature extraction on point-sampled surfaces. *Computer Graphics Forum*, 22(3):281–289, 2003.
- [18] M. Rutishauser, M. Stricker, and M. Trobina. Merging range images of arbitrarily shaped objects. In *Proc. CVPR*, 1994.
- [19] R. Sagawa, K. Nishino, and K. Ikeuchi. Adaptively merging large-scale range data with reflectance properties. *PAMI*, 27:392–405, 2005.
- [20] L. Silva, O. Bellon, and K. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *PAMI*, 27(5):762–776, 2005.
- [21] R. Song, Y. Liu, R. Martin, and P. Rosin. 3d gini coefficient: An evaluation methodology for multi-view surface reconstruction algorithms. Technical report, 2011.
- [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] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proc. SIGGRAPH*, pages 311–318, 1994.
- [25] H. Zhou and Y. Liu. Accurate integration of multi-view range images using k-means clustering. *Pattern Recognition*, 41(1):152–175, 2008.
- [26] 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.