Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Computer Vision and Image Understanding 92 (2003) 56–77

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

A volumetric approach for interactive 3D modeling

Dragan Tubic, Patrick Hébert, and Denis Laurendeau*

Computer Vision and Systems Laboratory, Laval University, Ste-Foy, Canada G1K 7P4

Received 15 April 2002; accepted 25 July 2003

Abstract

Range image registration and surface reconstruction have been traditionally considered as two independent processes where the latter relies on the results of the former. This paper presents a new approach to surface recovery from range images where the two processes are unified and performed in a common volumetric representation. While the reconstructed surface is described in its implicit form as a signed distance field within a volume, registration information for matching partial surfaces is encoded in the same volume as the gradient of the distance field. This allows coupling of both reconstruction and registration and leads to an algorithm whose complexity is linear with respect to the number of images and the number of measured 3D points. The close integration and performance gain improve interactivity in the process of modeling from range image acquisition to surface reconstruction. The distances computed in the direction of filtered normals improve robustness while preserving the sharp details of the initial range images. It is shown that the integrated algorithm is tolerant to initial registration errors as well as to measurement errors. The paper describes the representation and formalizes the approach. Experimental results demonstrate performance advantages and tolerance to aforementioned types of errors.

* Corresponding author.
E-mail addresses: tdragan@gel.ulaval.ca (D. Tubic), hebert@gel.ulaval.ca (P. Hébert), denis.laurendeau@gel.ulaval.ca (D. Laurendeau).

1077-3142/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2003.07.001

1. Introduction

Registration and surface reconstruction (integration of multiple range images) are two main steps in 3D modeling from multiple range images. Even though these two problems have received considerable attention during the past decade, real-time interactive modeling still remains out of reach. This partially results from the fact that registration and integration are performed separately but, more importantly, it is mainly a consequence of the complexity of existing algorithms, especially registration algorithms. Avoiding these limitations would be beneficial to numerous applications:

- *Interactive acquisition*. Providing a partially reconstructed model to the user during data acquisition greatly facilitates the selection of the next best view and assures that the set of acquired images is sufficient for model building. This is particularly important for real-time range sensors [10,24,25] where a large quantity of range images have to be registered and integrated. Usually the registration and integration are performed off-line which might take considerable time [10].
- *Filtering*. Real-time registration and integration of redundant range data can be used to improve the quality of the reconstructed model by reducing the variance of the noise while keeping the sharp details of the surface intact. Such filtering can be accomplished by averaging multiple observations of the same surface, a process equivalent to frame averaging in image processing. Nevertheless, this requires very well aligned images.
- *Self referencing*. Assuming that the change of viewpoint is small between subsequent range images in a sequence, registration can be used to reference the sensor with respect to its environment, thus providing a complementary or an alternative approach for positioning using external referencing systems. This application is also important in mobile robotics since the algorithm provides an up-to-date volumetric model of the scene as well as the rigid transformation between views.

Registration, the process of aligning surfaces, can be considered as two different problems: global registration where no a *priori* information on relative positions and orientations of images is available, and pose refinement where it is assumed that the images are initially close to their exact position. In this paper, registration is concerned with the latter problem. The initial position of images is usually obtained by using a sensor positioning device, but can also be obtained by manually aligning the input surfaces.

This paper takes a step towards real-time interactive modeling systems by providing a method for registration as well as for integration of range images. The integration and registration steps are merged and both the reconstructed surface model and its corresponding matching information can be built incrementally. This is made possible using a single volumetric structure which encodes the signed distance field for integration and its gradient for matching. By computing the gradient field for matching, the combinatorial complexity of pairwise matching is avoided. Actually, the algorithm becomes of linear complexity with respect to both the number of images and the measured points. Moreover, the volumetric structure is advantageously exploited to make the algorithm scalable and thus a good candidate for highly parallel implementation. Besides incremental modeling, simultaneous registration of multiple

range images can be processed within the same structure. In this case, the complete set of images is used to compute the two fields during the process.

Computing the distance between a point and a surface as the distance in the direction of filtered normals is another important aspect that is put forward in this paper since this affects both registration and integration. Choosing the closest point based on the Euclidean distance makes the result sensitive to range image noise; although it is possible to smooth (filter) the range images, sharp details are lost. It is instead proposed to only filter estimated normals while leaving the 3D points intact. This preserves details of the surface while making the algorithm more robust with respect to noise.

The paper is organized as follows: a short overview of registration strategies and problems related to their complexity as well as an overview of volumetric methods for integration are given in Section 2. While Section 3 introduces the notation, Section 4 presents the proposed solution and describes the formalism of the new approach for registration and integration of range images. Experimental results obtained with the proposed algorithm are presented in Section 5. Finally, directions of further research and concluding remarks are given in Section 6.

2. Related work

There are three main strategies for the registration of multiple range images: (i) registration of two surfaces at a time [27], usually referred to as pairwise registration, (ii) simultaneous registration of all images [13,19,21,29], and (iii) sequential registration of images to previously registered and merged images [3]. For a more detailed review of existing registration algorithms as well as a comparison of different algorithms, the reader is referred to [6,7,23]. Pairwise registration generally causes an accumulation of the registration error: when a pair of images are not perfectly registered, the registration error propagates to the next pair. On the other hand, simultaneous registration does not suffer from accumulation of registration errors but registering a single image requires matching to all other images. As a consequence, algorithm complexity grows exponentially with the number of images. The third solution is a compromise between the two others. In this case the propagation of the registration error is reduced and the number of pairwise matchings is equal to the number of images.

Computational complexity, sensitivity to noise and initial registration errors are key issues in registration. From the standpoint of computational complexity, the main problem is the matching step. The simplest approach to achieve matching between two images is to select control points in one image and project them onto the triangulated surface obtained from the second image along the direction of the sensor [2,13]. Since it is only determined by the sensor direction, the selection of corresponding points is fast and not sensitive to image noise. Nevertheless, it is sensitive to initial registration error and sensor viewpoint. Another commonly used approach is to choose the closest point as the corresponding point [1]. Although this approach provides the direction of the matched point based on surface shape, the matching is sensitive to image noise [6]. Actually matching errors occur whenever the distance

between two surfaces is large because many points on one surface are attracted by noisy points (see [23], Fig. 8 or Fig. 2 in this paper). As for the computational complexity, a brute force matching algorithm requires $O(N^2)$ operations, where N is the number of points in each image. Using more sophisticated approaches and data structures such as k–d trees, reduces complexity to $O(N \log N)$ [8,25].

Following registration, the images are *integrated* into a single surface. Recently, a number of volumetric approaches for integration have been proposed [5,11,17,20,22,28]. These algorithms use an implicit representation of the surface in the form of a signed distance field calculated on a discrete lattice of points. The images are merged by averaging fields for individual images. The reconstructed surface is recovered by extracting the zero-set of the resulting distance field, usually using the Marching Cubes algorithm [15]. A strong point of the volumetric approach is its ability to incrementally build a model by simply accumulating the distance fields for individual images.

None of the volumetric algorithms, except [16–18], provide the registration of range images. Masuda [16] initially proposed to merge both integration and registration in a common structure by matching and aligning signed distance fields. Nevertheless, registration does not fully exploit the potential of the volumetric structure to reduce complexity of closest point search and the extensive use of k–d trees makes this approach computationally expensive [17]. Moreover, the computation of the distance field is based on the distance to the closest point and is thus sensitive to noise. To limit this behavior, Masuda restricts the search for the closest points to those points within the distance equal to a voxel diagonal. However, this severely limits the maximum acceptable initial registration errors. We solve this problem by computing the distance in the direction of filtered normals and we reduce the computational complexity by matching and aligning surfaces with the implicit representation of the reconstructed model.

3. Notation

Throughout the paper, vectors are denoted using boldface letters while scalars are denoted in italic. A range image, I, is considered as a set of measured 3D points, $\mathbf{p}_i$,

$$I = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\},$$

defined on a rectangular grid, where N is the number of points in the image. It is assumed that the surface of an object is initially approximated by a triangulated range image. This triangulation, S, is represented as a set of triangles T_i,

$$S = \{T_1, \ldots, T_M\},$$

where triangles T_i are defined using three vertices $\mathbf{p}_{i,1}$, $\mathbf{p}_{i,2}$, $\mathbf{p}_{i,3} \in I$ and M is the number of triangles. The triangulation of a range image is easy to obtain since the rectangular grid implicitly encodes the connectivity between points.

An *iso-surface* of some surface S, denoted as $S(\delta)$, is a surface such that for all $\mathbf{p} \in S(\delta)$, $g(\mathbf{p}, S) = \delta$, where g is some distance function. Similarly an *iso-triangle*

$T(\delta)$ for a triangle T is a triangle such that for all $\mathbf{p} \in T(\delta)$, $g(\mathbf{p}, T) = \delta$ holds for a given distance function g. Triangle $T = T(0)$ is referred to as a *generator-triangle* while the surface $S = S(0)$ is referred to as a *generator-surface*.

4. Incremental registration and integration

4.1. Specific objectives and requirements

As our primary goal we set a method for registration and integration of multiple range images that can be used *during* acquisition, possibly in real-time, thus providing a partially reconstructed model after each image is acquired. Furthermore, the method should allow simultaneous registration since this strategy avoids accumulation of registration error. To achieve this goal, the method has to satisfy a number of requirements:

- The method must be *incremental* in order to allow updating of the model without further processing of previously acquired images. Moreover, computational complexity of registration and integration of an image should not depend on the number of previously integrated images. Otherwise, the method will slow down as new images are added to the model.
- The method has to be of *linear complexity* with respect to both the number of images and the number of triangles or points in acquired images. Linear complexity offers a performance advantage but also allows *parallelization* of the algorithm. This means that the computational tasks can be distributed over a large number of processors, thus drastically reducing the execution time.
- The method should be *robust* with respect to noise. In particular this requirement concerns registration: the noise should not affect significantly neither the quality of registration nor the execution time (convergence).
- Since range images must overlap in order to allow registration, they contain redundant data. The method should be able to exploit this redundancy to improve the quality of the reconstructed model, i.e. to reduce the noise. Furthermore, this filtering should not smooth sharp details of the surface.

4.2. An integrated volumetric approach

Volumetric integration algorithms [5,11,17] are incremental, thus satisfying the first requirement: a signed distance field is built for each image on a discrete lattice of points within some envelope (regular, rectangular volumetric grid). The reconstructed model is represented as the average of individual fields. From such a representation the surface model can be extracted using the Marching Cubes algorithm [15].

A good compromise between computational complexity and quality of registration is to register an image to a model reconstructed from previously registered and merged images. However, this does not solve the complexity problem itself: whenever an image is merged to the model, the matching complexity grows with

the number of points and triangles. Since the model is generally not a graph surface, the points cannot be projected onto the surface of the model and one must *search* for the corresponding point. This leads to a complexity of $O(N^2)$ or $O(N \log N)$ using k–d trees [9]. The proposed solution to this problem and the main idea of incremental registration is to build the matching information incrementally in the same way as the distance field is built for integration. This consists in pre-computing closest points in the neighborhood of the reconstructed surface such that the matching information can be obtained as the value from the closest lattice point (voxel centre). To do so, one will note that the direction of the closest point on the surface is actually given as the direction of the gradient of the distance field d (where differentiable), and that the following relation holds

$$\nabla \sum_i d_i = \sum_i \nabla d_i. \tag{1}$$

The gradient of the integration (summed) field, and thus the direction towards the closest point of the reconstructed model can be computed incrementally in the same way as the distance field itself. This solves the performance problem related to the number of images since each image is used only once for matching. It is possible to compute the gradient directly from the distance field but the result is inaccurate since the distance field is calculated only on a discrete lattice of points. Therefore, the gradient is rather computed *explicitly* on the same lattice points for which the distance field is calculated.

Since the direction of matching corresponds to the gradient of the distance field, the choice of distance plays a major role and not all distances are equally suited for registration. Computing the distance in the direction of the sensor results in gradient values, and therefore in matching directions, that are determined by the direction of the sensor instead of depending on the shape of the surface. On the other hand, using the distance relative to the closest points is sensitive to noise and might result in invalid matchings (see Fig. 2). Our solution to this problem is to compute the distance field in the direction of filtered and interpolated normals rather than towards the closest point on the surface. The rationale is that the normals can be filtered efficiently without filtering range data and, by doing so, the influence of noise can be reduced to a very small area which makes it practically insignificant. The distance in the direction of interpolated normals has been introduced in [26].

To make the complexity of the field computation proportional to the number of triangles we note that the distance to the surface is always calculated relative to a single triangle on the surface, and that there is a connected region in 3D space where, for each point, that triangle contains the closest point. In the case of the classical Euclidean distance, this region corresponds to a Voronoi cell for the triangle. Partitioning a surface envelope in these regions allows independent computation of the field for each triangle. An example of the envelope partition is shown in Fig. 3. This reduces the computational complexity to be proportional to the number of triangles. Furthermore, the computation of the field using this strategy is highly parallelizable and can be distributed among any number of processors (smaller or equal to the number of triangles).

4.3. Computing the distance field in the direction of filtered normals

The first step in computing the distance field is obtaining normals at all points of a triangulated surface. The normal $\mathbf{n}$ for a triangle whose vertices are $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ is obtained as

$$\mathbf{n} = \frac{(\mathbf{p}_1 - \mathbf{p}_2) \times (\mathbf{p}_1 - \mathbf{p}_3)}{\left\|(\mathbf{p}_1 - \mathbf{p}_2) \times (\mathbf{p}_1 - \mathbf{p}_3)\right\|}. \tag{2}$$

The sign of the normals should be consistent with the sensor direction. Thus for all triangle normals $\mathbf{n}_i$, the scalar product $\langle \mathbf{n}_i, \mathbf{s} \rangle$ with the sensor direction, $\mathbf{s}$, should have the same sign. The normal at each vertex of a triangulated image is first computed as the average of normals of all triangles containing the vertex. Second, this normal is filtered by averaging using normals at vertices of adjacent triangles. This procedure yields normals at vertices. In order to match points not only to vertices but to the triangulated surfaces, the normal at interior points within triangle boundaries is computed as a linear interpolation of the normals at vertices. Thus for any point $\mathbf{p}$ on a triangle, the normal $\mathbf{n}(\mathbf{p})$ is obtained as

$$\mathbf{n}(\mathbf{p}) = b_1\mathbf{n}_1 + b_2\mathbf{n}_2 + b_3\mathbf{n}_3, \tag{3}$$

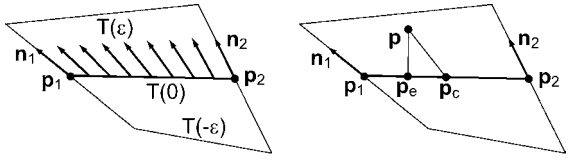
where b_1, b_2, and b_3 are barycentric coordinates of $\mathbf{p}$ such that $b_1 + b_2 + b_3 = 1$. Vectors $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ represent normals at vertices $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ of the triangle.

Using Eq. (3) the closest point $\mathbf{p}_c$ to point $\mathbf{p}$ is defined as

$$\mathbf{p} = \mathbf{p}_c + d\mathbf{n}_c, \tag{4}$$

where $\mathbf{n}_c$ is the normal at $\mathbf{p}_c$ and d is the distance between $\mathbf{p}$ and $\mathbf{p}_c$ along $\mathbf{n}_c$. In this equation d is the distance between the point and the surface. The computation of the signed distance field is based on this definition. Interpolated normals, closest point $\mathbf{p}_c$ along the normal $\mathbf{n}_c$, and the closest point $\mathbf{p}_e$ found using Euclidean distance are illustrated in Fig. 1. The effect of using the distance in the direction of filtered normals for matching is illustrated in Fig. 2.

The distance field is calculated only within some envelope of the surface. According to our definition of the distance, this envelope is bounded by two iso-surfaces that are obtained by displacing each vertex of the original triangulated image in the direction of the normals for some constant value ϵ referred to as the envelope size. An example of an envelope is shown in Fig. 3.



(a) (b)

Fig. 1. 2D slice through a fundamental prism associated with a generator-triangle $T(0)$ and bounded by two iso-triangles $T(\epsilon)$ and $T(-\epsilon)$. (a) Interpolated normals (b) Closest point $\mathbf{p}_e$ on the surface to a point $\mathbf{p}$ in standard Euclidean sense and the closest point $\mathbf{p}_c$ in the direction of interpolated normals.

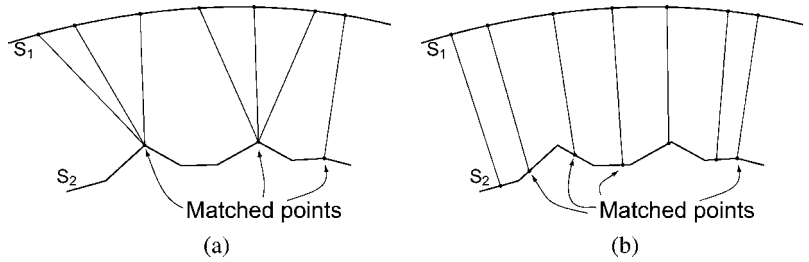


Fig. 2. Effect of matching in the direction of filtered normals. (a) Matching to the closest point. (b) Matching in the direction of filtered and interpolated normals. The matched points are spread more evenly over the second surface.

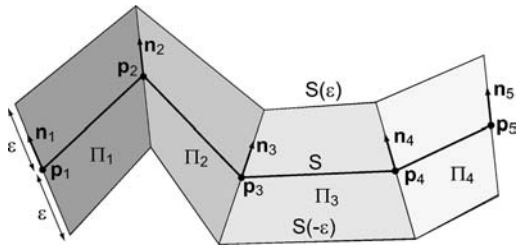


Fig. 3. 2D slice through a volumetric envelope for surface S. Iso-surfaces $S(\epsilon)$ and $S(-\epsilon)$ bounding the envelope are obtained by displacing each vertex $\mathbf{p}_i$ of a surface S for a constant value ϵ in the direction of normals $\mathbf{n}_i$. Fundamental prisms Π_i are depicted as regions in different shades of gray.

As explained in Section 4.2, the envelope is partitioned into regions associated with generator-triangles such that for all points in a region, the generator-triangle contains the closest point (see Fig. 3). From the definition of distance in Eq. (4), these regions coincide with fundamental prisms introduced in [4]. A fundamental prism associated with a generator-triangle T is a region bounded by two iso-triangles $T(\epsilon)$ and $T(-\epsilon)$ as well as by three edge-surfaces. The edge-surfaces are bilinear patches defined using two vertices of a generator-triangle $T(0)$ and their associated normals. An example of a fundamental prism is shown in Fig. 4.

The three bilinear patches are parameterized as

$$\begin{aligned}\mathbf{b}_1(u,d) &= (1-u)(\mathbf{p}_1 + d\mathbf{n}_1) + u(\mathbf{p}_2 + d\mathbf{n}_2),\\ \mathbf{b}_2(u,d) &= (1-u)(\mathbf{p}_2 + d\mathbf{n}_2) + u(\mathbf{p}_3 + d\mathbf{n}_3),\\ \mathbf{b}_3(u,d) &= (1-u)(\mathbf{p}_3 + d\mathbf{n}_3) + u(\mathbf{p}_1 + d\mathbf{n}_1),\end{aligned} \tag{5}$$

where $u \in [0,1]$ and $d \in [-\epsilon,\epsilon]$, $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are vertices of the generator-triangle $T(0)$ and $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ are normals at the vertices. As defined in Eq. (4), the distance for any point located inside a fundamental prism with respect to the associated generator-triangle is computed as the solution of the following system:

$$[\mathbf{p}_1 + d\mathbf{n}_1 - \mathbf{p}, \mathbf{p}_2 + d\mathbf{n}_2 - \mathbf{p}, \mathbf{p}_3 + d\mathbf{n}_3 - \mathbf{p}] = 0. \tag{6}$$

In Eq. (6), square brackets denote the scalar triple product. This equation yields up to three real roots. We choose the root d for which the point $\mathbf{p}$ is contained within

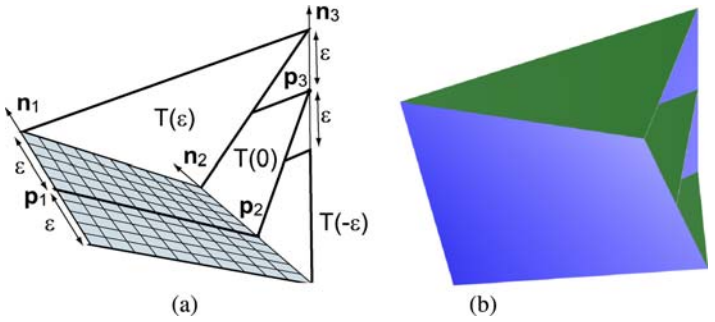


Fig. 4. Example of a fundamental prism. (a) Fundamental prism formed by displacing vertices $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ of triangle $T(0)$ in the direction of normals $\mathbf{n}_1$, $\mathbf{n}_2$ and $\mathbf{n}_3$ for a constant distance ϵ. Triangles $T(\epsilon)$, $T(-\epsilon)$ as well as three bilinear patches bound the prism. One of the three bilinear patches is depicted as a shaded region. (b) Shaded fundamental prism.

triangle $T(d)$ whose vertices are $\mathbf{p}_1 + d\mathbf{n}_1, \mathbf{p}_2 + d\mathbf{n}_2, \mathbf{p}_3 + d\mathbf{n}_3$. It can be shown that only one root satisfies this condition. Barycentric coordinates b_1, b_2 and b_3 of the point $\mathbf{p}$ in triangle $T(d)$ correspond to the barycentric coordinates of the closest point in the generator-triangle. Therefore, once the distance is known, the closest point is obtained as

$$\mathbf{p}_c = b_1\mathbf{p}_1 + b_2\mathbf{p}_2 + b_3\mathbf{p}_3. \tag{7}$$

The gradient direction of the distance map is nothing but the normalized direction towards the closest point:

$$\nabla d(\mathbf{p}) = \frac{\mathbf{p} - \mathbf{p}_c}{\|\mathbf{p} - \mathbf{p}_c\|}. \tag{8}$$

The two fields associated with a surface are the distance computed using Eq. (6) and its gradient Eq. (8). These values are evaluated at each voxel center inside the envelope of the surface. In practice, both fields (distance and gradient) are calculated independently for each fundamental prism, and, to do so efficiently, they are computed only at voxels inside the bounding box of the prism. As suggested in [11] in order to improve efficiency, the fields should not be computed at all points in the bounding box since only a few of these points might actually be inside the prism (see Fig. 5). To find the set of points located inside the prism, it is first checked whether a row of voxels in the bounding box intersects the prism. If this is the case, there are two intersection points and the field is then computed only at lattice points between these two boundary points. By doing so, the time required for computing the field is significantly reduced.

4.4. Determining the size of the volumetric envelope

In order to create a connected distance field, the minimum size of the envelope ϵ must at least be equal to the length of the voxel diagonal [11]. On the other hand, the maximum size of the envelope is a trade-off between maximum acceptable registra-

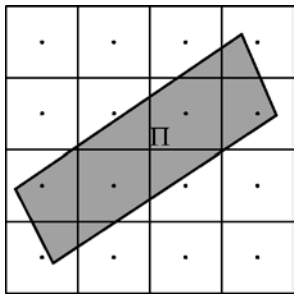


Fig. 5. A fundamental prism and its bounding box. Only a fraction of voxels are located inside the prism. The fundamental prism is shaded in gray.

tion error and the requirement that the envelope has no self-intersection. Therefore, since the size and the shape of the model limit the size of the envelope, they also limit maximal initial registration error. However, for pose refinement, it is reasonable to expect that the sensor produces smaller registration errors than the size of the model. Moreover it is worth noting that, in order to be successfully registered, a surface does not have to be completely contained within an envelope: if the points initially located in the envelope suffice to improve the pose of a surface, a larger number of points should enter into the field, at each iteration.

As explained earlier, the purpose of dividing the envelope into non-overlapping prisms is to allow an independent computation of the field for each triangle. However if the normals are insufficiently filtered, some prisms might degenerate (''twist'') within the envelope, as illustrated in Fig. 6. Consequently, the prisms in the neighborhood of the degenerate prism overlap and (see shaded region in Fig. 6b) the distance is not uniquely defined in the regions of overlap. There are three ways to prevent prisms from degenerating. The first one is to reduce the size of the envelope (which imposes another constraint on the maximum size of the envelope), so that ''twists'' occur outside the envelope. The second way is to filter normals sufficiently thus moving ''twists'' outside the envelope as in Fig. 6c and, finally, to truncate only degenerate prisms and their neighbors before the ''twist'' occurs (see Fig. 6d). Since range images are graph surfaces, one should note that it is always possible to choose (filter) normals so that an envelope of arbitrary size can be constructed. For instance, taking all normals to be parallel to the sensor's direction would satisfy this condition, although it is a rather extreme case. Results presented in this paper were obtained by filtering normals sufficiently to ensure that there is no degenerate prisms.

In order to truncate the degenerate prisms or to reduce the size of the envelope to avoid self-intersections, the maximum allowable distance $\epsilon_{\max}$ has to be calculated for each prism such that the prism bounded by $T(\epsilon_{\max})$ and $T(-\epsilon_{\max})$ does not contain twists. This maximum distance can be obtained by computing the intersection point $\mathbf{q}_i$ of each line $\mathbf{l}_i(u) = \mathbf{p}_i + v\mathbf{n}_i,\ i = 1,2,3$ with the facing edge-surface defined in Eq. (5). If they exist, the three intersection points satisfy the following equations:

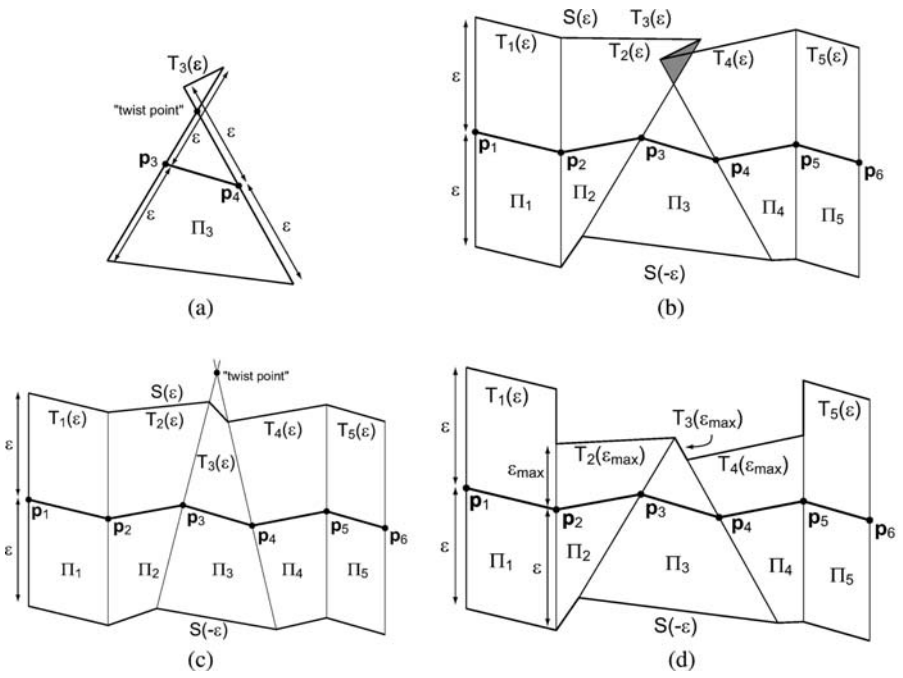


Fig. 6. Example of degenerate prisms within an envelope. (a) Degenerate prism (b) Effect of a degenerate prism on its neighbors. Region of overlap of prisms Π_2, Π_3 and Π_4 is shaded in gray. (c) Same envelope with better filtered normals. The ''twist'' occurs outside the envelope. (d) Overlap of prisms avoided by truncating degenerate prism and its neighbors.

$$\begin{aligned}
\mathbf{b}_1(u_1, d_1) &= \mathbf{l}_3(v_3) = \mathbf{q}_3, \\
\mathbf{b}_2(u_2, d_2) &= \mathbf{l}_1(v_1) = \mathbf{q}_1, \\
\mathbf{b}_3(u_3, d_3) &= \mathbf{l}_2(v_2) = \mathbf{q}_2.
\end{aligned} \tag{9}$$

Finally, a line intersects the patch within the envelope if

$$0 \leqslant u_i \leqslant 1 \quad \text{and} \quad -\epsilon \leqslant d_i \leqslant \epsilon. \tag{10}$$

The intersection points can be easily calculated by representing the lines as intersections of two planes and substituting these two equations into Eq. (5). The maximum positive distance is given as the minimum positive value of parameters d_i and v_i satisfying Eq. (10), while the maximum negative distance is given as their maximal negative value. A degenerate cell and the intersection points described above are illustrated, in 3D, in Fig. 7.

4.5. Incremental update of the fields

The sum of fields for multiple images contains both the implicit representation of the surface as a distance field and its associated matching information in the form of

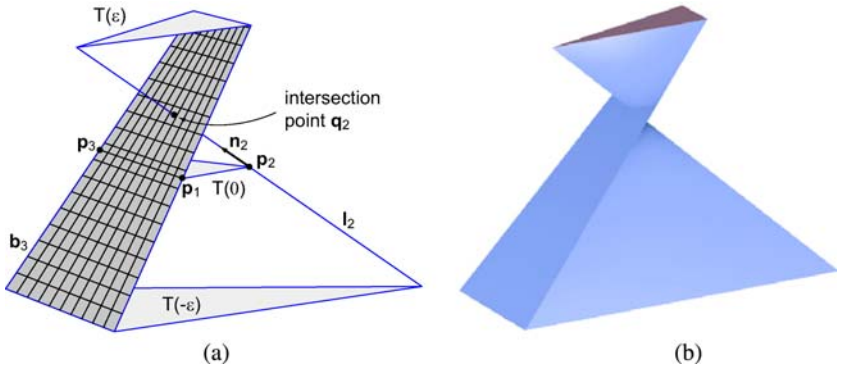


Fig. 7. An example of a degenerate prism in 3D. (a) Degenerate prism. (b) Shaded degenerate prism.

a vector field corresponding to the gradient of the distance field. We refer to these fields as *integration fields*. As mentioned above, the integration fields are obtained by averaging fields for N individual images:

$$\mathbf{f}_{\text{int}}(\mathbf{p}) = \left[\sum_i^N \mathbf{f}_i(\mathbf{p})\omega_i(\mathbf{p})\right] \Big/ \left[\sum_i^N \omega_i(\mathbf{p})\right], \tag{11}$$

$$d_{\text{int}}(\mathbf{p}) = \left[\sum_i^N d_i(\mathbf{p})\omega_i(\mathbf{p})\right] \Big/ \left[\sum_i^N \omega_i(\mathbf{p})\right], \tag{12}$$

where $\mathbf{f}$ denotes the vector (gradient) field, d denotes the signed distance field, and where ω represents the confidence level for the measured points, usually expressed as the cosine of the angle between the direction of the sensor and surface normal. To preserve the continuity of the fields, the weights ω_i should be interpolated in the same way as the normals using the barycentric coordinates, i.e.

$$w(\mathbf{p}) = w_1 b_1 + w_2 b_2 + w_3 b_3. \tag{13}$$

An example of the distance and vector fields is shown in Fig. 8. It should be noted that the distance field d and the gradient field $\mathbf{f}$ can also be considered as a single vector field $d\mathbf{f}$.

4.6. Image registration

Once the fields are computed, the registration is straightforward: the corresponding point $\mathbf{p}_c$ for a control point $\mathbf{p}$ is given as

$$\mathbf{p}_c = \mathbf{p} + \mathbf{f}_{\text{int}}(\mathbf{p}_{\mathbf{v}})[d_{\text{int}}(\mathbf{p}_v) + \langle \mathbf{f}_{\text{int}}(\mathbf{p}_v), (\mathbf{p} - \mathbf{p}_v)\rangle], \tag{14}$$

where $\mathbf{p}_v$ is the closest lattice point (voxel centre)

$$\mathbf{p}_v = \lfloor \mathbf{p} \rfloor = [\lfloor x/\Delta + 0.5 \rfloor, \lfloor y/\Delta + 0.5 \rfloor, \lfloor z/\Delta + 0.5 \rfloor]^T, \tag{15}$$

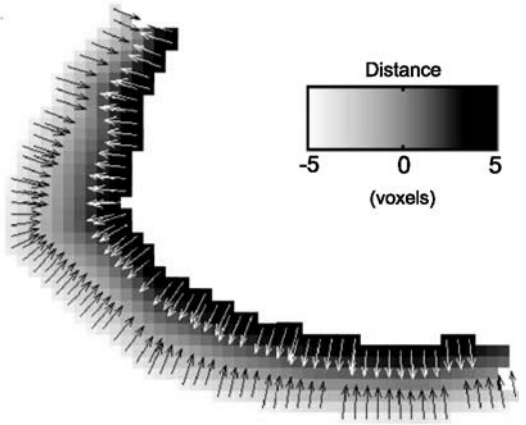


Fig. 8. 2D slice through the integration fields. The distance field d_{int} is depicted as different shades of gray, while the directions of gradient field $\mathbf{f}_{\text{int}}$ are depicted as arrows.

while Δ denotes voxel size (see Fig. 9). The factor $\mathbf{f}_{\text{int}}(\mathbf{p}_v)\langle\mathbf{f}_{\text{int}}(\mathbf{p}_v), (\mathbf{p}-\mathbf{p}_v)\rangle$ compensates for the error introduced by discretization of the field. Note that since this is a linear interpolation, the closest point will not in general be exactly on the surface. However, the produced error is sufficiently small as justified in the next section.

All measured points in an image are considered as control points. The optimal rigid transformation in the least-squares sense, which aligns control and corresponding points, can be found using quaternions [12] or geometric algebra [14].

There are two implementations of the registration algorithm. The first one is sequential since each image is registered to the integration field and then added to it. The second one creates the integration field using all images and then registers each image individually. Both algorithms are described in pseudo-code below.

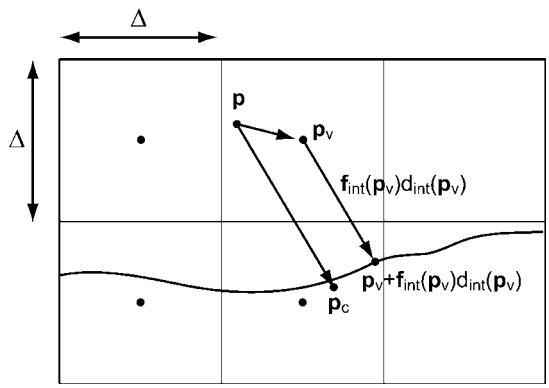


Fig. 9. Matching a point $\mathbf{p}$ using the closest voxel centre $\mathbf{p}_v$. The closest point $\mathbf{p}_c$ is obtained as in Eq. (14).

Algorithm 1: Incremental Registration

Calculate fields $\mathbf{f}_1$ and d_1 for the first image
$\mathbf{f}_{int} \leftarrow \mathbf{f}_1$ and $d_{int} \leftarrow d_1$
$i \leftarrow 2$
repeat
  repeat
    Find matching points for control points of image I_i using $\mathbf{f}_{int}$ and d_{int}
    Calculate and apply transformation on image I_i
  until *convergence*;
  Calculate fields $\mathbf{f}_i$ and d_i for registered image I_i
  Add fields $\mathbf{f}_i$ and d_i to $\mathbf{f}_{int}$ and d_{int}
  $i \leftarrow i + 1$
until *Until no images left*;

Algorithm 2: Simultaneous Registration

repeat
  Initialize integration fields $\mathbf{f}_{int}$ and d_{int} to zero
  for *i = 1:number of images* **do**
    Calculate fields $\mathbf{f}_i$ and d_i for image I_i
    Add fields $\mathbf{f}_i$ and d_i to $\mathbf{f}_{int}$ and d_{int}
  end
  repeat
    for *i = 1:number of images* **do**
      Find matching points for control points of image I_i using $\mathbf{f}_{int}$ and d_{int}
      Calculate and apply transformation on image I_i
    end
  until *convergence*;
until *convergence*;

While the first algorithm is more useful for registration and integration during image acquisition, the second algorithm can be applied once the acquisition is completed in order to obtain the final model. Even though simultaneous registration is expected to give better results since it does not accumulate registration errors, the first algorithm should always be used prior to simultaneous registration. To see this, consider the example in Fig. 10a where a single image is unregistered. During simultaneous registration, this error propagates and disturbs the registration of other images (Fig. 10b). For this reason simultaneous registration might take a considerable number of iterations before converging, which is computationally expensive since the

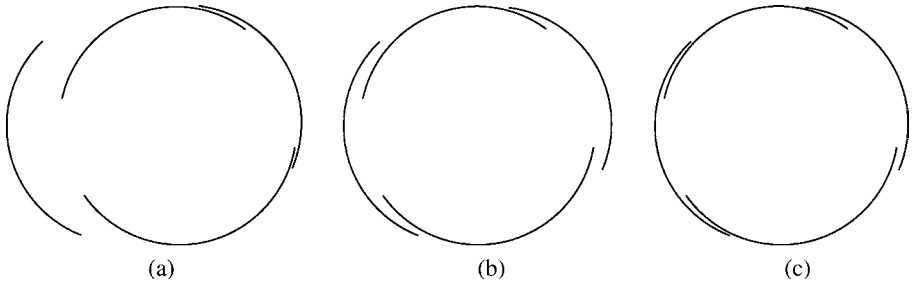


Fig. 10. Error propagation during simultaneous registration.

fields should be recalculated at each iteration. It is thus wiser to reduce the initial registration errors as much as possible using the first algorithm and then to apply the second one to remove residual registration errors.

4.7. Surface reconstruction and visualization

The reconstructed surface model is simply obtained by extracting the zero-set of the distance field. As mentioned before, the Marching Cubes algorithm, or a variant, can be used for this purpose. Nevertheless, if the extracted surface is to be used only to display the model, it is advantageous to perform ray-tracing through the volume instead of rendering the surface extracted by the Marching Cubes algorithm. To do so, the zero-crossing of the distance fields are located along the rays, and the local surface orientation is estimated at the zero-crossing closest to the virtual camera. There are two ways to estimate the local orientation of the surface. The simplest one is to take the value of the gradient field at the closest voxel centre. This yields a smooth rendered image if the normals were filtered. The second approach is to estimate the local orientation in the same way as Marching Cubes: by locally triangulating the surface inside the box formed by the 8 voxels closest to the point of zero-crossing.

5. Results

In order to assess the performance of the algorithm it is very convenient to use synthetic range images since both registration and measurement errors can be perfectly controlled. More importantly, the position of the images following registration can be compared to their exact position. For this purpose, 12 perfectly aligned and noiseless range images from a CAD model of a Beethoven statue were generated. Images were then transformed as follows: each image was translated along each axis for a random value between 0 and 5 voxels and was rotated around each axis (while centered at the origin) for a random angle between 0 and 5 degrees. Rotation angles and translation vectors have uniform distribution. Noise added to measured points

followed a normal distribution. The assessment of the registration error is made by comparing the position of each point in the registered model *to its exact position*. The resolution of the synthetic images was 150×150 while the resolution of the 3D lattice was chosen $128 \times 128 \times 128$ in order to match approximately the resolution of images.

To provide experimental evidence supporting the claim that the filtering of normals makes the algorithm less sensitive to noise, the residual registration error was measured for varying levels of image noise while keeping the registration errors constant. The results shown in Fig. 11 indicate that the noise has a minor impact on the performance of the algorithm.

One might argue that the discretization of the fields should result in a less accurate registration. This is true but, since the model is reconstructed on a discrete lattice of points, registration errors smaller than voxel size are invisible. Therefore it is sufficient to reduce the registration errors below voxel size. Experimental results confirm that the proposed algorithm meets this requirement. Fig. 12 shows the average and

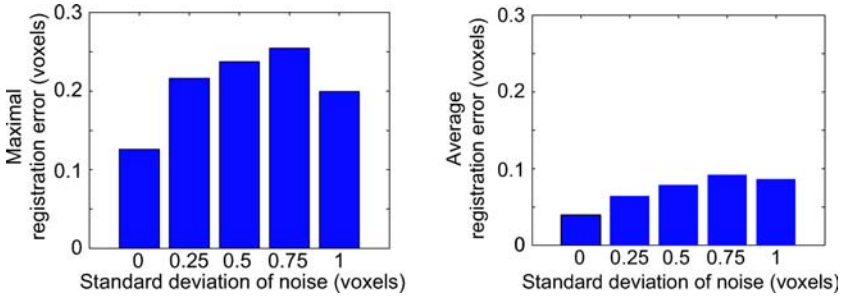


Fig. 11. Registration error vs. level of image noise.

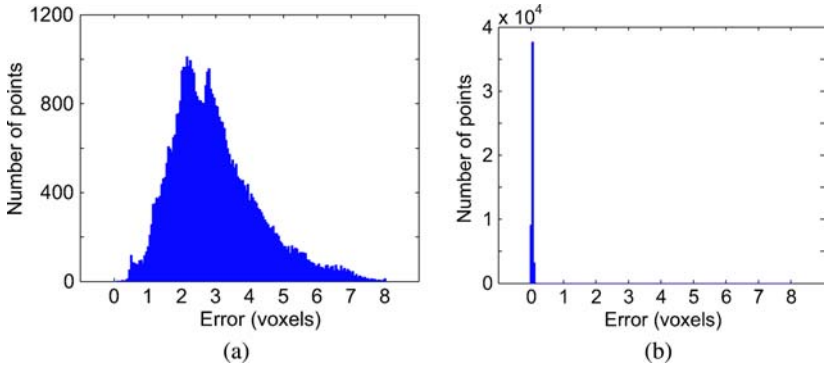


Fig. 12. Distribution of registration error for 12 noiseless images of a Beethoven statue. (a) Registration error before registration. (b) Residual registration error after registration.

the maximum error distribution for the Beethoven model before (a) and after (b) registration.

The influence of noise on the convergence speed of the algorithm is illustrated in Fig. 13 for both 12 noiseless images and 12 images corrupted by noise with a standard deviation of 1 voxel. The diagrams on the right side show the change of the norm of the rotation matrix while the plots on the left side show the change of the translation vector as a function of the number of iterations. These diagrams show that the convergence of the algorithm is practically unaffected by the noise.

Since the integration field is an average of all individual fields, model filtering is performed automatically. An example of the surface reconstructed by registering and integrating a single image with 1, 10, and 50 observations of the object from a different viewpoint is shown in Fig. 14. The second image covers the left-hand side of the first image. The images were generated as in previous experiments by randomly perturbing position and orientation of images and corrupting them with

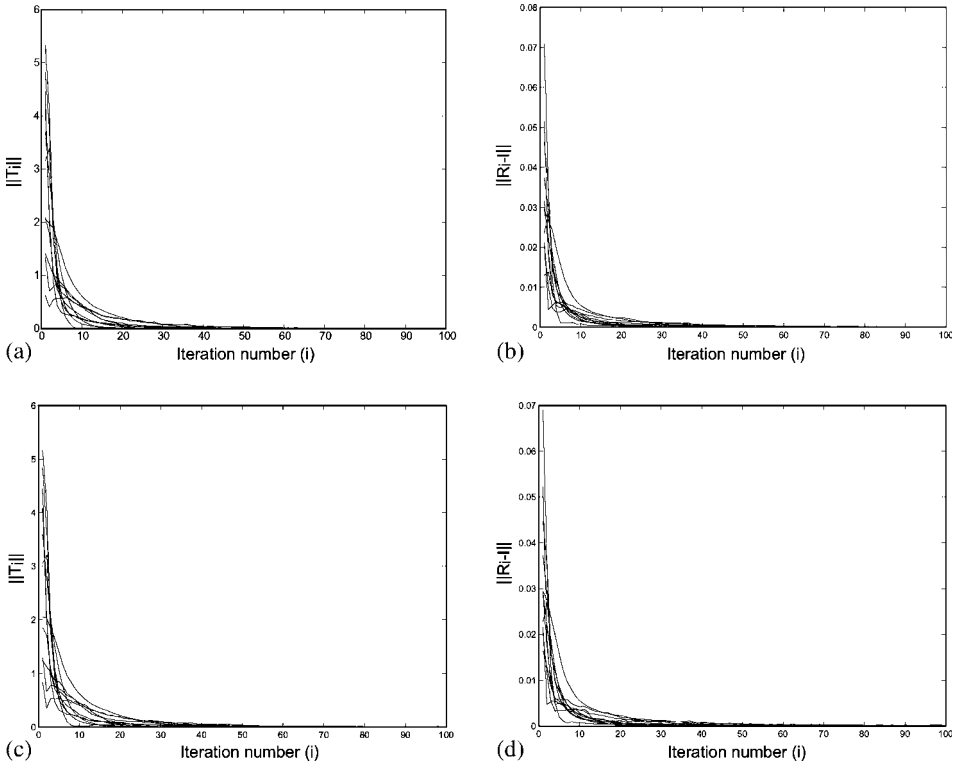


Fig. 13. Evolution of transformation parameters as a function of the number of iterations. Top row: Evolution of translation (a) and rotation (b) for 12 noiseless images of Beethoven. Bottom row: Evolution of translation (c) and rotation (d) for 12 images corrupted by noise of the standard deviation of 1 voxel.

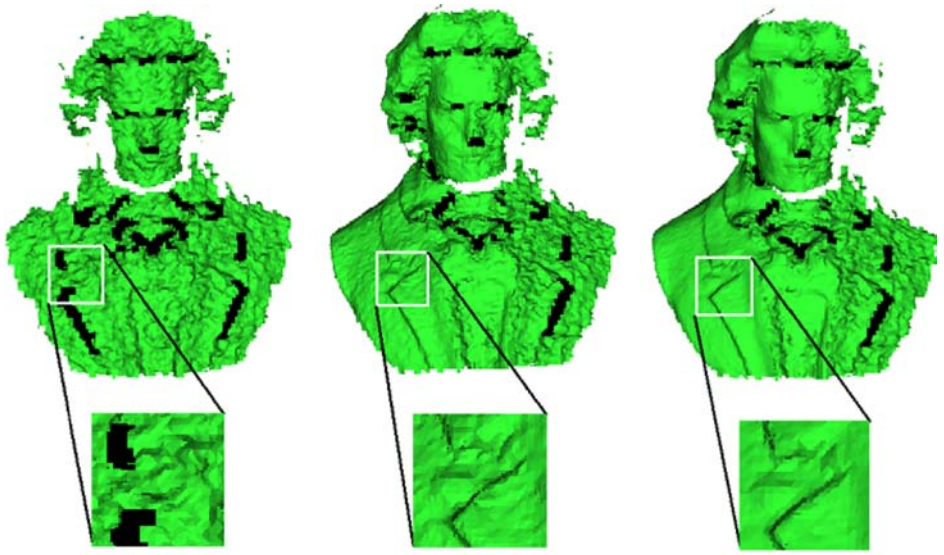


Fig. 14. Example of filtering. (a) Reconstruction from a single image. (b) Reconstruction from 10 registered and averaged images (left side). (c) Reconstruction from 50 registered and averaged images (left side). Since the registration algorithm provides an accurate pose for each image, there is no loss of fine details on the reconstructed surface.

noise. Note that averaging makes sense only if the images are well registered. Also note that registering a very large number of images, for example a few thousands, with a registration algorithm whose complexity is $O(M^2)$ with respect to the number of images, is very difficult if not impossible for current algorithms. Another potential problem is that a frame rate of 30 or 60 images per second clutters disk space rapidly. By performing the reconstruction and filtering online, the redundant data can be discarded as soon as its field is summed in the integration field.

Illustration of reconstruction and registration of the synthetic and real data is shown in Fig. 15 for the Beethoven model and the model of a rabbit from the Stanford image repository. Another example, a model of a duck, reconstructed from 12 range images obtained at NRCC is shown in Fig. 16. The envelope size for all three models has been fixed to 3 voxels.

The execution time of the unoptimized algorithm, including I/O, on a 1.2 GHz PC, is 2 seconds per image of Beethoven model containing approximately 8000 triangles each. For the rabbit model whose images contain around 80,000 triangles, the execution time was below 15 seconds per image. Most of the time is spent on the computation of the transformation: 3 s for the calculation of the field and 12 for the iterative pose refinement. Although we did not implement this option, a subset of all measured points could be chosen as control points thus accelerating the computation of the transformation.

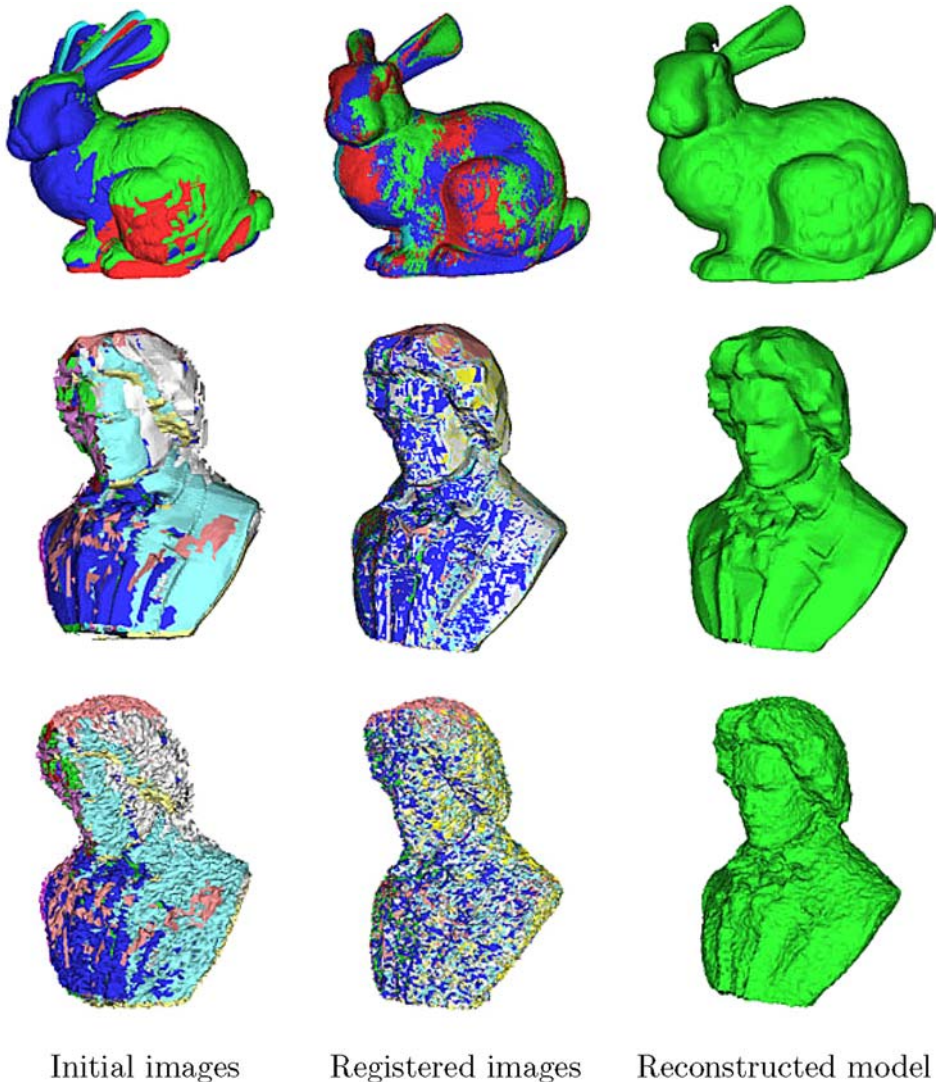


Fig. 15. Examples of reconstruction and registration. Top row: real range data from Stanford image repository. Middle row: registration and reconstruction using noiseless synthetic data. Bottom row: registration and reconstruction using noisy images (standard deviation of noise 1 voxel).

6. Conclusion

We have presented two algorithms suitable for online as well as for offline registration and integration of range images. For these two algorithms, measurements are accumulated in an integration field which includes not only the implicit representation of the surface but also registration information in a vector field. This makes the

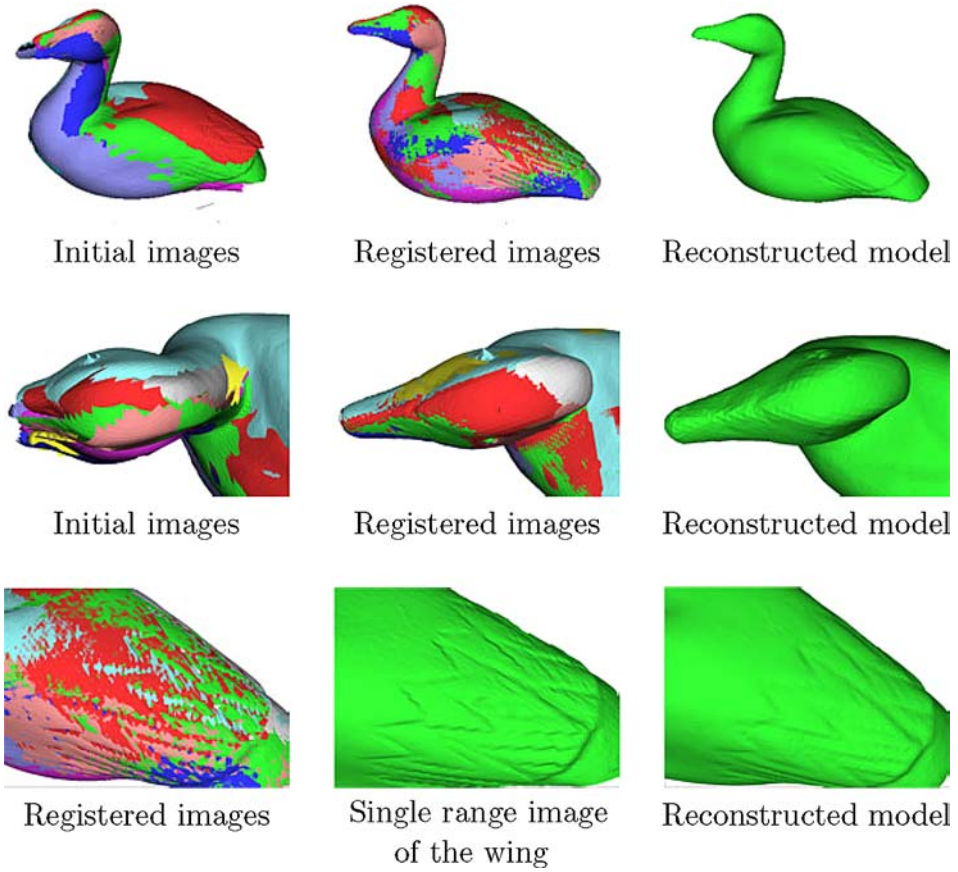


Fig. 16. Example of reconstruction and registration from real range data. Top row: side view of the whole model. Middle row: top-view of the head detail. Bottom row: zoom on the wing.

algorithms of linear complexity with respect to both the number of images and the number of triangles built from the images. Adding a new image with N triangles requires the computation of N fundamental prisms. Since correspondence is stored in the created field, registration can be performed efficiently requiring $O(1)$ operations per measurement. The correspondences are found in the direction of filtered normals thus making the registration less sensitive to noise and producing a more uniform distribution of corresponding points on the reconstructed surface.

Based on incremental registration and surface integration, the first algorithm is well suited for interactive modeling. Actually, it is possible to integrate a model from a continuous flow of range images that are not well aligned. Redundancy of range measurements can thus be exploited to recover a higher quality model. The second algorithm is a variation of the first algorithm; it is based on simultaneous registration and can be applied *a posteriori* when data acquisition is completed. Even though the

simultaneous registration does not accumulate registration errors and is thus expected to give better results, we observed only subtle differences between the recovered models. Nevertheless, simultaneous registration is more computationally expensive while the gain in the quality of registration is not necessarily significant. A quantitative comparison of the two registration strategies should be made, but this topic is beyond the scope of this paper.

One apparent drawback common to all volumetric approaches is the limited resolution imposed by memory requirements and computational complexity. The number of voxels, and consequently the number of operations, grows as a power of 3 with the resolution. Moreover, representing a surface, which is a 2D object, in a 3D volume leads to very inefficient memory usage since most of the voxels remain unused. It thus appears more difficult to build precise models for very large objects. One could recover surface sections independently in a set of local overlapping volumes but this approach is not ideal since efficiency is lost, at least partially. There are several proposed solutions for the memory requirement issue such as run-length encoding or octrees for locally adapting the resolution. However, these solutions are more or less sophisticated data structures that do not convey geometric properties of the shape they represent. More efficient and useful compression schemes are currently being explored.

Acknowledgments

The authors wish to express their gratitude to the Fonds de Recherche sur la Nature et les Technologies and the Natural Sciences and Engineering Research Council of Canada for supporting this research.

References

[1] P. Besl, N. McKay, A method for registration of 3-d shapes, IEEE Trans. Pattern Anal. Mach. Intell. 14 (2) (1992) 239–256.
[2] G. Blais, M. Levine, Registering multiview range data to create 3d computer objects, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 820–824.
[3] Y. Chen, G. Medioni, Object modelling by registration of multiple range images, Internat. J. Image Vision Comput. 10 (3) (1992) 145–155.
[4] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, W. Wright, Simplification envelopes, in: SIGGRAPH '96 Conf. Proc., August, 1996, pp. 119–128.
[5] B. Curless, M. Levoy, A volumetric method for building complex models from range images, in: SIGGRAPH '96 Conf. Proc., 1996, pp. 303–312.
[6] G. Dalley, P. Flynn, Range image registration: a software platform and empirical evaluation, in: Proc. Third Internat. Conf. 3D Digital Imaging Modeling, 2001, pp. 246–253.
[7] D.W. Eggert, A.W. Fitzgibbon, R.B. Fisher, Simultaneous registration of multiple range views for use in reverse engineering of cad models, Comput. Vision Image Understanding 69 (3) (1998) 253–272.
[8] J.H. Friedman, J.L. Bentley, R.A. Finkel, An algorithm for finding best matches in logarithmic expected time, ACM Trans. Math. Software 3 (3) (1977) 209–226.
[9] M. Greenspan, G. Godin, A nearest neighbor method for efficient icp, in: Proc. Third Internat. Conf. 3D Digital Imaging Modeling, May, 2001, pp. 161–168.

[10] O. Hall-Holt, S. Rusinkiewicz, Stripe boundary codes for real-time structured light range scanning of moving objects, in: Proc. Eighth IEEE Internat. Conf. Comput. Vision, vol. 2, 2001, pp. 359–366.
[11] A. Hilton, J. Illingworth, Geometric fusion for a hand-held 3d sensor, Mach. Vision Appl. 12 (1) (2000) 44–51.
[12] B.K.P. Horn, Closed-form solution of absolute orientation using unit quaternions, J. Opt. Soc. Am. A 4 (4) (1987) 629–642.
[13] O. Jokinen, Area-based matching for simultaneous registration of multiple 3-d profile maps, Comput. Vision Image Understanding 71 (3) (1998) 431–447.
[14] J. Lasenby, W.J. Fitzgerald, C.J.L. Doran, A.N. Lasenby, New geometric methods for computer vision, Internat. J. Comput. Vision 36 (3) (1998) 193–213.
[15] W.E. Lorensen, H.E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, in: SIGGRAPH '87 Conf. Proc., vol. 21(4), 1987, pp. 163–169.
[16] T. Masuda, A unified approach to volumetric registration and integration of multiple range images, in: Proc. 14th Internat. Conf. Pattern Recogn., 1998, pp. 977–981.
[17] T. Masuda, Generation of geometric model by registration and integration of multiple range images, in: Proc. Third Internat. Conf. 3D Digital Imaging Modeling, 2001, pp. 254–261.
[18] T. Masuda, Registration and integration of multiple range images by matching signed distance fields for object shape modeling, Comput. Vision Image Understanding 87 (2003) 51–65.
[19] P.J. Neugebauer, Reconstruction of real-world objects via simultaneous registration and robust combination of multiple range images, Internat. J. Shape Modeling 3 (1,2) (1997) 71–90.
[20] K. Pulli, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, W. Stuetzle, Robust meshes from range maps, in: Proc. Internat. Conf. Recent Adv. 3-D Digital Imaging Modeling, Ottawa, Canada, May, 1997, pp. 205–211.
[21] H.G.R. Bergevin, M. Soucy, D. Laurendeau, Towards a general multi-view registration technique, Pattern Anal. Mach. Intell. 18 (5) (1996) 540–547.
[22] G. Roth, E. Wibowoo, An efficient volumetric method for building closed triangular meshes from 3-d image and point data, in: W. Davis, M. Mantei, V. Klassen (Eds.), Graphics Interface, May, 1997, pp. 173–180.
[23] S. Rusinkiewicz, M. Levoy, Efficient variants of the icp algorithm, in: Proc. Third Internat. Conf. 3D Digital Imaging Modeling, May, 2001, pp. 145–152.
[24] W. Schroeder, E. Forgber, S. Estable, Scannerless laser range camera, Sensor Rev. 19 (4) (1999) 285–291.
[25] D. Simon, M. Hebert, T. Kanade, Real-time 3-d pose estimation using a high-speed range sensor, in: Proc. IEEE Internat. Conf. Robotics Automation (ICRA '94), vol. 3, May, 1994, pp. 2235–2241.
[26] W. Sun, A. Hilton, R. Smith, J. Illingwort, Layered animation of captured data, Visual Comput.: Internat. J. Comput. Graphics 17 (8) (2001) 457–474.
[27] G. Turk, M. Levoy, Zippered polygon meshes from range images, in: SIGGRAPH '94 Conf. Proc., vol. 26, 1994, pp. 311–318.
[28] M. Wheeler, Y. Sato, K. Ikeuchi, Consensus surfaces for modeling 3d objects from multiple range images, in: Proc. Internat. Conf. Comput. Vision, January, 1998, pp. 917–924.
[29] J. Williams, M. Bennamoun, Simultaneous registration of multiple corresponding point sets, Comput. Vision Image Understanding 81 (1) (2001) 117–142.