A Volumetric Approach for the Registration and Integration of Range Images: Towards Interactive Modeling Systems

Dragan Tubić Patrick Hébert Denis Laurendeau
Computer Vision and Systems Laboratory,
Laval University,
Ste-Foy, Québec, Canada, G1K 7P4

E-mail: (tdragan, hebert, laurendeau)@gel.ulaval.ca

Abstract

This paper presents a new approach for registering and integrating range images where these two processes are merged and performed in a common volumetric representation. The proposed approach allows both simultaneous and incremental registration where matching complexity is linear with respect to the number of images. This improvement leads to incremental modeling from range image acquisition to surface reconstruction. It is shown that the approach is tolerant to initial registration errors as well as to measurement errors while keeping the details of the initial range images. The paper describes the formalism of the approach. Experimental results demonstrate performance advantages and tolerance to aforementioned types of errors for free form objects.

1 Introduction

Registration and integration (geometric fusion) are two main steps in 3D modeling from multiple range images. However, the complexity of existing algorithms [7], especially registration algorithms, is too high to allow real-time, interactive modeling. This paper takes a step towards real-time modeling systems by providing a method for incremental registration and integration of range images. The algorithm is of linear complexity with respect to both the number of images and the number of points; it is thus a good candidate for parallel implementation. This improvement leads to numerous applications. Firstly, providing a partially reconstructed model to the user during acquisition facilitates the selection of the next best view and assures that the acquired images are sufficient for the reconstruction of the model. Secondly, 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 intact the sharp details of the surface. Finally, registration of range images can be used to reference the sensor with respect to its environment. This application is important in mobile robotics since the algorithm provides an up-to-date volumetric model of the scene and the rigid transformation between subsequent views.

2 Registration and Integration of Multiple Range Images: Problem Statement

There are three ways to perform the registration of range images: registration of two surfaces at the same time - usually referred to as pair-wise registration, simultaneous registration of all images, and sequential registration of images to previously registered and merged images. The first approach generally causes an accumulation of the registration error: when a pair of images is not perfectly registered, then the registration error propagates to the next pair. On the opposite, simultaneous registration does not suffer from this problem, but registering a single image requires finding matches in all other images. As a result, the algorithm complexity grows exponentially with the number of images. The third solution is a compromise between the two others. In this case the registration error is reduced and the number of pairwise matchings is equal to the number of images.

The main performance problem of registration is linked to the matching step. The simplest approach to achieve matching between two images is to select points in one image and to project them on the surface obtained from the second one along the direction of the sensor [2]. By doing so, the matching is determined by the relative positions of the images, rather then by the surface shape. Another commonly used approach is to take the closest point as the corresponding point [1]. A brute force algorithm requires $O(N^2)$ operations, where N is the number of points in each image. Using more sophisticated approaches based on k-d trees reduces complexity to $O(NlogN)$.

Recently, a number of volumetric approaches for integration have been proposed [4, 5, 7]. 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 surface can be recovered from such a representation by extracting the zero-set of the distance field, usually using the Marching Cubes algorithm [6]. A strong point of the volumetric approach is its ability to incrementally build the model by summing the distance fields for individual images. The *integration* algorithm proposed in [5] is of linear complexity and thus well suited for our purpose. Except [7], none of those volumetric algorithms provide the registration of range images. However, the use of k-d trees makes this last approach unsuitable for incremental registration.

3 Incremental Registration: Solutions

The main idea of incremental registration is to build matching information incrementally in the same way as the distance field is built for integration. This consists in precomputing closest points in the neighbourhood of the reconstructed surface (on a discrete lattice of points), so that the matching information can be obtained simply as the value from the closest voxel centre. To do so, we 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 is true:

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

The gradient of the integration (summed) field, and thus direction towards the closest point of the reconstructed model can be computed incrementally in the same way as the distance field itself. This solves 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 of points.

For the performance problem related to the number of points we note that the distance field is always calculated relative to a single triangle on the surface, and that there is a connected region where for each point, this triangle contains the closest point. Partitioning a surface envelope in these regions allows independent computation of the field for each triangle. This reduces the complexity to be proportional to the number of triangles. Such a partition, referred to as fundamental prisms, is introduced in [3].

Not all distance fields are well suited for registration. Calculating the distance in the direction of the sensor, as proposed in [4], results in gradient values, and therefore in matching directions, that are determined only by the direction of the sensor. In [7] the computation of the distance field is based on the distance to the closest point and is thus very sensitive to noise. Hilton [5] calculates the distance relative to the triangles in the direction of their normal. He also uses normal volumes to partition the space, so that inside such a volume, the distance is calculated relative to a single triangle; this may lead to discontinuous iso-surfaces. To ensure continuity of the distance fields, a new distance is defined in section 3.1.

Due to the sensitivity of the direction of the closest point to noise, matching errors occur whenever the distance between two surfaces is large because many points on one surface are attracted by outliers. To limit this behaviour, Masuda [7] restricts the search for the closest points to those points located within the distance equal to a voxel diagonal. However, this severely limits the maximum acceptable initial registration errors. Our solution to this problem is to compute the distance field in the direction of filtered normals rather then 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 and practically unsignificant.

3.1 Calculating the Distance Field in the Direction of Filtered Normals

At each vertex of a triangulated surface the normal is computed as an average of normals of all triangles containing the vertex and filtered by averaging with normals at vertices of adjacent triangles. This procedure yields normals at vertices but not at points located inside triangle boundaries, an information that is required in order to match points to the triangulated surface. For interior points we linearly interpolate the normals at vertices. Thus, for any point $\mathbf{p}$ on a triangle, the normal is obtained as

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

where b_1, b_2 and b_3 are barycentric coordinates of $\mathbf{p}$ and $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ are normals at the vertices of the triangle.

Using this definition of the normals, we define the closest point $\mathbf{p}_c$ to point $\mathbf{p}$ as the point which verifies

$$\mathbf{p} = \mathbf{p}_c + d\mathbf{n}_c, \tag{2}$$

where $\mathbf{n}_c$ is the normal at the point $\mathbf{p}_c$ and d is the distance between $\mathbf{p}$ and $\mathbf{p}_c$. 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 figure 1.

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 (see figure 1). An example of a fundamental prism in 3D is shown in figure 2a.

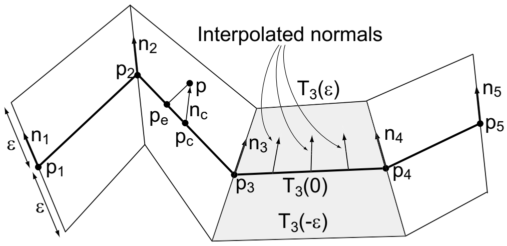
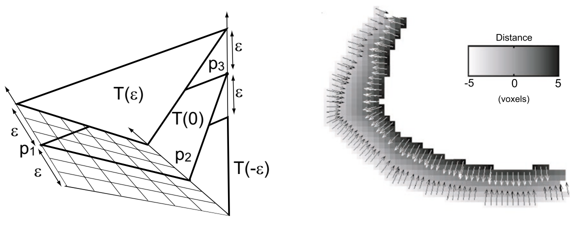


Figure 1. Volumetric envelope.



a) b)

Figure 2. a) Fundamental prism, b) Distance field.

As defined in Eq. 2, the distance for any point located inside a fundamental prism with respect to the associated triangle (generator triangle) is computed as the solution of the following system:

$$[\mathbf{p}_1 + d\mathbf{n}_1 - \mathbf{p}, \mathbf{p}_2 + d\mathbf{n}_2 - \mathbf{p}, \mathbf{p}_3 + d\mathbf{n}_3 - \mathbf{p}] = 0. \quad (3)$$

In Eq. 3 square brackets denote the scalar triple product, $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are vertices of the triangle and $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ their normals. If point $\mathbf{p}$ is in the prism then it is located in a triangle $T(d)$ whose vertices are $\mathbf{p}_1 + d\mathbf{n}_1, \mathbf{p}_2 + d\mathbf{n}_2, \mathbf{p}_3 + d\mathbf{n}_3$. Barycentric coordinates b_1, b_2 and b_3 of the point $\mathbf{p}$ in this triangle 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

$$\mathbf{p}_c = b_1\mathbf{p}_1 + b_2\mathbf{p}_2 + b_3\mathbf{p}_3. \quad (4)$$

The gradient of the distance map $\mathbf{f}$ is nothing else but the normalized direction of the closest point:

$$\mathbf{f}(\mathbf{p}) = \nabla d(\mathbf{p}) = \frac{\mathbf{p} - \mathbf{p}_c}{||\mathbf{p} - \mathbf{p}_c||} \quad (5)$$

In practice, the distance field and its gradient are calculated independently for each fundamental prism. To do so efficiently, a bounding box is calculated for each prism and the distance is calculated using equation 3 for each point inside the bounding box. Furthermore, one must verify that the point is located inside the prism.

3.2 Incremental Update of the Fields

The sum of fields for multiple images contains both the implicit representation of the surface as a distance field d_{int}, and its associated matching information in the form of a vector field $\mathbf{f}_{int}$ corresponding to the gradient of the distance field. We refer to these fields as to *integration fields*. As mentioned above, the integration fields are obtained by averaging fields for individual images:

$$\mathbf{f}_{int}(\mathbf{p}) = \left[\sum_i^N \mathbf{f}_i(\mathbf{p})\omega_i(\mathbf{p})\right] / \left[\sum_i^N \omega_i(\mathbf{p})\right], \quad (6)$$

$$d_{int}(\mathbf{p}) = \left[\sum_i^N d_i(\mathbf{p})\omega_i(\mathbf{p})\right] / \left[\sum_i^N \omega_i(\mathbf{p})\right], \quad (7)$$

where $\mathbf{f}$ denotes the unit vector (gradient) field, d is the signed distance field, and ω represents the confidence level for the measured points, usually expressed as the cosine of the angle between the direction of the sensor and the surface normal. To preserve the continuity of the fields, the weights ω_i should be interpolated in the same way as the normals, using barycentric coordinates. An example of the distance and vector fields is shown in figure 2b.

3.3 Image Registration

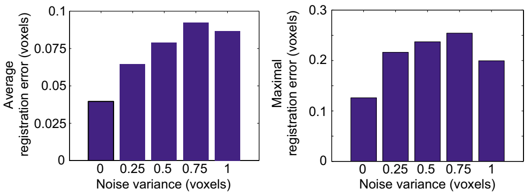
Once the integration fields are computed, the registration of a single image is performed by aligning all measured points $\mathbf{p}$ with $\mathbf{p} + d_{int}(\mathbf{p}_v)\mathbf{f}_{int}(\mathbf{p}_v) + \langle \mathbf{f}_{int}(\mathbf{p_v}), (\mathbf{p} - \mathbf{p}_v)\rangle$, where $\langle \mathbf{f}_{int}(\mathbf{p_v}), (\mathbf{p} - \mathbf{p}_v)\rangle$ compensates for the distance between the point $\mathbf{p}$ and the closest voxel center $\mathbf{p}_v$. 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. The first algorithm is described in pseudo-code below.

```
Incremental Registration
Initialize fields using first image: $\mathbf{f}_{int} \leftarrow \mathbf{f}_1$, $d_{int} \leftarrow d_1$, $i \leftarrow 2$
Repeat until no images left
    Repeat until convergence
        Find matching points for image $I_i$ using $\mathbf{f}_{int}$ and $d_{int}$
        Calculate and apply transformation on image $I_i$
    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 $d_{int}$, $i \leftarrow i + 1$
```

4 Results

In order to assess the performance of the algorithm it is very convenient to firstly 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 centred 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 final position of each point in the registered model to its exact position. Resolution of the synthetic images was 150 × 150 while the resolution of the 3D lattice was 128 × 128 × 128.

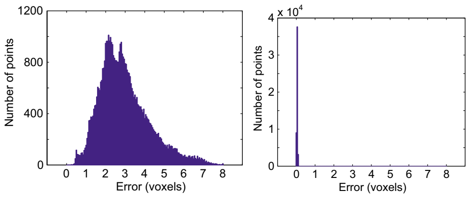
To provide an experimental evidence supporting the claim that the filtering of normals makes the algorithm less sensitive to noise, the residual registration error was measured for varying level of noise while keeping the registration errors constant. The result shown in figure 3 indicates that the noise has a minor impact on the performance of the algorithm.



a) b)

Figure 3. Registration error vs. level of noise.

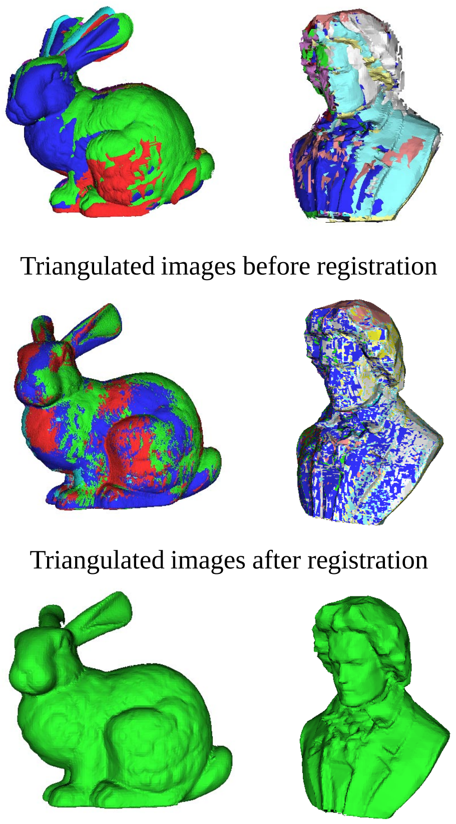
One might argue that the discretization of the fields should result in a less accurate registration. This is true but, since the model is reconstructed on a discrete lattice of points, registration errors smaller then the voxel size are invisible. Therefore it is sufficient to reduce the registration errors below voxel size. Experimental results confirm that this is accomplished by the proposed algorithm. Figure 4 shows the actual registration error distribution for the Beethoven model before (a) and after (b) registration.



a) b)

Figure 4. Registration error distribution.

Reconstruction and registration of the synthetic and real data are shown in figure 5 for the Beethoven model and the model of a rabbit from the Stanford image repository. The execution time of the unoptimized algorithm, on a 1.2GHz PC, is 2 seconds per image containing approximately 10000 triangles each.



Triangulated images before registration

Triangulated images after registration

Reconstructed models

Figure 5. Reconstruction and registration.

References

[1] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[2] G. Blais and M. Levine. Registering multiview range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:820–824, 1995.

[3] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, and W. Wright. Simplification envelopes. In *SIGGRAPH '96 Conference Proceedings*, pages 119–128, August 1996.

[4] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96 Conference Proceedings*, pages 303–312, August 1996.

[5] A. Hilton and J. Illingworth. Geometric fusion for a handheld 3d sensor. *Machine vision and applications*, 12:44–51, 2000.

[6] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH '87 Conference Proceedings*, 21(4):163–169, 1987.

[7] T. Masuda. Generation of geometric model by registration and integration of multiple range images. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling*, pages 254–261, May 2001.