

A Volumetric Approach for Interactive 3D Modeling

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

Range image registration and surface reconstruction have been traditionally considered as two independent problems where the latter relies on the results of the former. This paper presents a new approach to surface recovery from range images where these two processes are integrated and performed in a common volumetric representation. The volumetric representation contains both implicitly represented reconstructed surface as the signed distance field and corresponding matching information in the form of the gradient of the distance field. This 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, tolerance to aforementioned types of errors and, as an application, filtering using redundant range data without loss of sharp details on the reconstructed surface.

1 Introduction

Registration and integration (geometric fusion) are two main steps in 3D modeling from multiple range images. However, the complexity of existing algorithms [12], 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. Similarly to [11, 12] we merge the integration and registration steps. Both reconstructed model and precalculated match-

ings are built incrementally. The algorithm is of linear complexity with respect to both the number of images and the number of points and is thus a good candidate for parallel implementation. This improvement leads to numerous applications:

- Interactive acquisition. Providing a partially reconstructed model to the user during the acquisition greatly facilitates the selection of the next best view and assures that the acquired images are sufficient for the reconstruction of the model.
- 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 intact the sharp details of the surface. This type of the filtering is equivalent to the frame averaging in image processing.
- Self referencing. 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.

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

2 Registration and Integration of Multiple Range Images

There are three ways to perform the registration of range images: *i*) registration of two surfaces at a time [17], usually referred to as pair-wise registration, *ii*) simultaneous registration of all images [9, 13, 14, 18], and *iii*) sequential registration of images to previously registered and merged images [3]. The first approach generally causes an accumulation of the registration error: when a pair of images is not perfectly registered, 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 a matching to all other images. As a consequence, 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. For a more detailed review of existing registration algorithms as well as a comparison of different algorithms, the reader is referred to [6, 16].

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 triangulated surface of the second one along the direction of the sensor [2, 9]. By doing so, matching is determined by the relative position of the images, rather than by the surface shape. Another commonly used approach is to take the closest point as the corresponding point [1]. A brute force matching 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 $O(N \log N)$.

Recently, a number of volumetric approaches for *integration* have been proposed [5, 8, 12, 15]. 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 [10]. A strong point of the volumetric approach is its ability to incrementally build the model by simply summing the distance fields for individual images. The *integration* algorithm proposed in [8] is of linear complexity and is thus well suited for our purpose. Nevertheless, none of those volumetric algorithms, except [12], provide the registration of range images. In this last approach the extensive use of k-d trees to match the points makes this approach unsuitable for incremental registration.

3 Incremental Registration

The best compromise between the complexity and the quality of registration is to register an image to a model re-

constructed from previously registered and merged images. However, this does not solve the complexity problem itself: whenever an image is merged to the model, the number of points and triangles grows and so does matching complexity. Since the points cannot be projected on the surface of the model, (the model is generally not a graph surface) one must *search* for the corresponding point; this leads to complexity $O(N^2)$ or $O(N \log N)$ using k-d trees [7]. Our 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 precomputing closest points in the neighbourhood of the reconstructed surface such that matching is encoded in closest points of a discrete 3D lattice. This reduces to precalculate 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

$$\nabla \sum_i d = \sum_i \nabla d. \quad (1)$$

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.

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 in 3D space where that triangle contains the closest point for each 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 of an envelope, referred to as fundamental prisms, is introduced in [4].

Not all distance fields are well suited for registration. Calculating the distance in the direction of the sensor, as proposed in [5], 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. In [11, 12] the computation of the distance field is based on the distance to the closest point and is thus very sensitive to noise (see [16], fig. 8). Hilton [8] 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 and may lead to discontinuous iso-surfaces over the edges of fundamental prisms. To ensure continuity of the distance fields, we introduce a new distance defined in the section 3.1.

Due to the sensitivity of the closest point direction 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 [12] 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. *Our solution to this problem is to compute the distance field in the direction of filtered 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 and is practically insignificant.

3.1 Calculating the Distance Field in the Direction of Filtered Normals

The normal at each vertex of a triangulated surface is computed as the average of the 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 not only to vertices but to the triangulated surface. For those 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

$$\mathbf{p} = \mathbf{p}_c + d \mathbf{n}_c, \quad (2)$$

where \mathbf{n}_c is the normal at 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). As mentioned earlier, the envelope is partitioned in fundamental prisms, an example

of which is shown in figure 2. Fundamental prisms are bounded by two iso-triangles $T(d_{min})$ and $T(d_{max})$ and by three edge-surfaces which are bilinear patches defined using two vertices of a triangle and associated normals (see Figure 2).

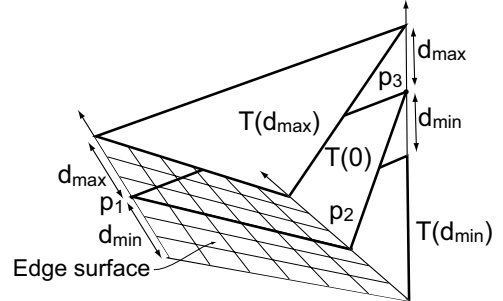


Figure 2. Fundamental prism.

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 the 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 is nothing but the normalized direction of the closest point:

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

In practice, both fields (distance and gradient) are calculated independently for each fundamental prism. To do so efficiently, the bounding box is calculated for each prism and the distance is calculated using equation 3 for each lattice point in the bounding box. Furthermore, one must verify that the point is located inside the prism. This reduces to verifying that the point is contained in a triangle $T(d)$ as explained above.

Since the normals at vertices of triangles are not parallel in general, a line passing through a vertex in the direction of the normal at the same vertex may intersect facing edge

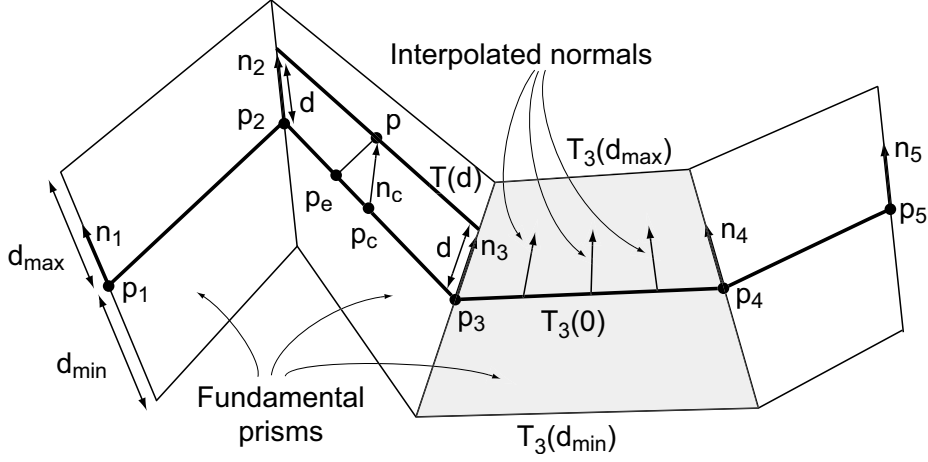


Figure 1. Volumetric envelope.

surface. This limits the maximal distance inside a prism because the equation 3 have infinitely many solutions at the point of intersection. Therefore, the calculated distance is accepted only if it is smaller than this maximal distance. Note that it is always possible to choose (filter) normals so that those intersections occur outside the envelope, for example by taking all normals to be parallel to sensors directions.

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 and its associated matching information in the form of a vector field 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 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.

$$w(\mathbf{p}) = w_1 b_1 + w_2 b_2 + w_3 b_3. \quad (8)$$

An example of the distance and vector field is shown in figure 3.

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. Both algorithms are described in pseudo-code below.

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
  Register image  $I_i$  using  $\mathbf{f}_{int}$  and  $d_{int}$ 
  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  $d_{int}$ 
   $i \leftarrow i + 1$ 
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

```

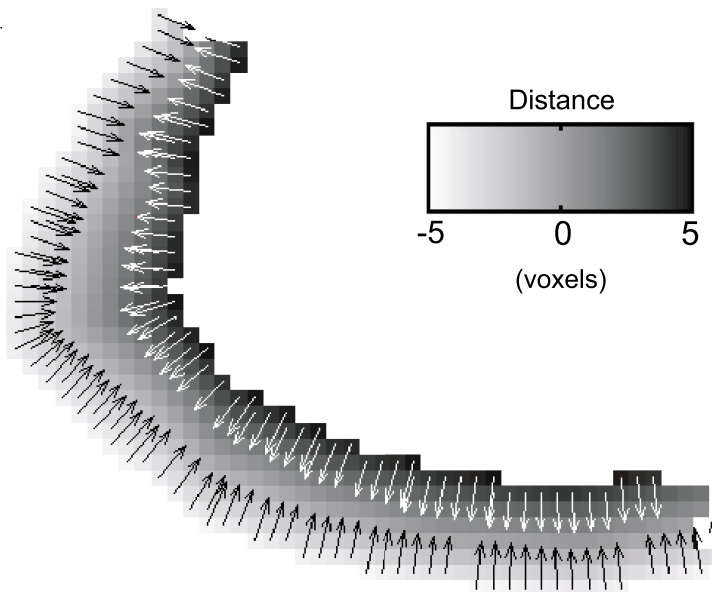


Figure 3. 2D slice through the integration fields. The distance field d_{int} is depicted as shades of gray, while the directions of gradient field $d_{int} \mathbf{f}_{int}$ are depicted as arrows.

```

    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
For  $i = 1$ :number of images
    Register image  $I_i$ 
End
Until convergence

```

While the first algorithm is intended for use during the acquisition, the second one is used for registration and integration after all the images have been acquired.

4 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. Resolution of the synthetic images was 150×150 while the resolution of the 3D lattice was $128 \times 128 \times 128$.

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 level of noise while keeping the registration errors constant. The result shown in figure 4 indicates 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 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 5 shows the average and the maximum error distribution for the Beethoven model before (a) and after (b) registration.

Finally, the influence of noise on the convergence speed of the algorithm is illustrated in Figure 6 for both 12 noiseless images and 12 images corrupted by noise of variance

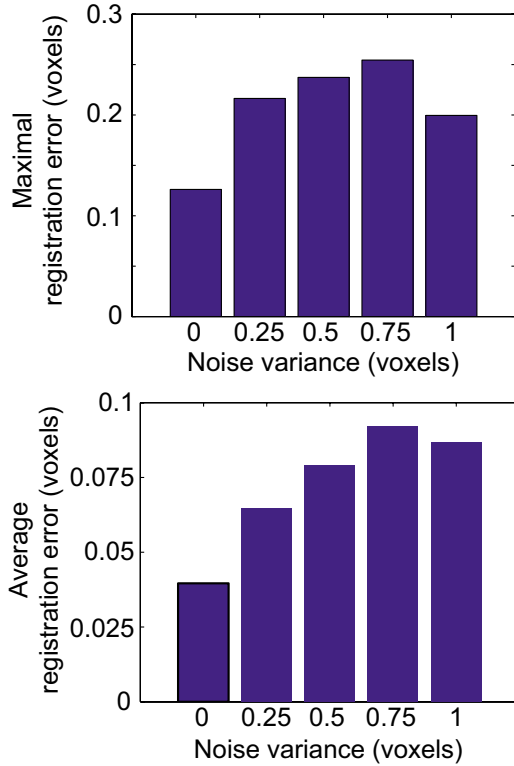
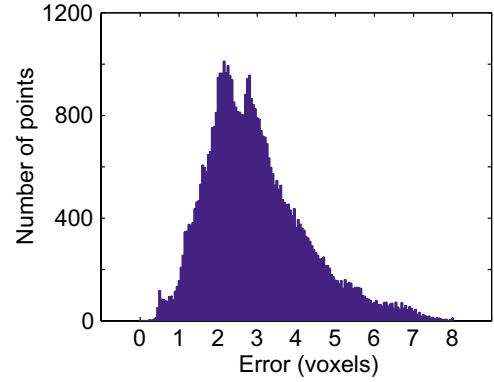


Figure 4. Registration error vs. level of noise.

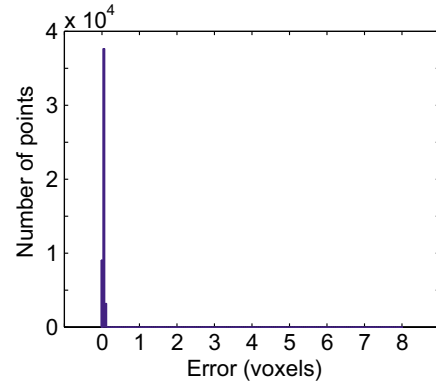
1 voxel. The diagrams on the right side show change of the norm of rotation the matrix while the plots on the left side show the change of the translation vector as a function of the number of iterations. These diagrams show that the convergence of the algorithm is practically unaffected by noise.

Since the reconstruction field is an average of all individual fields, the 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 another viewpoint using the above algorithm is shown in figure 7. The second image covers the left side of the first image. Note that averaging makes sense only if the images are well registered. Also note that registering a very large number of images, say 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 almost impossible for current algorithms. Another problem is that a frame rate of 30 or 60 images per second clutters disk space rapidly. By doing the reconstruction and filtering on-line, as it is proposed in this paper, the redundant data can be discarded as soon as its field is summed in the integration field.

Illustration of reconstruction and registration of the syn-



(a)



(b)

Figure 5. Distribution of registration error for 12 noiseless images of a Beethoven statue. (a) Registration error before registration. (b) Residual registration error after registration

thetic and real data is shown in figure 8 for the Beethoven model and the model of a rabbit from the Stanford image repository. The execution time of the unoptimized algorithm, including I/O, on a 1.2GHz PC, is 2 seconds per image containing approximately 10000 triangles each.

5 Conclusion

We presented volumetric algorithms for registration and integration of range images. The algorithms are of linear complexity with respect to the number of images and the number of triangles. The impacts of our approach are numerous, but perhaps, the most important one is on sensor design. Online registration and filtering by averaging allow design of less accurate 3D scanners without sophisticated referencing systems that still can produce high quality models. As a matter of fact, we believe that the future of 3D scanners is not only in the design of very accurate scanners

that use accurate referencing, but also in high-speed scanners that provide high-quality models not by the quality of range images but rather by their quantity (redundancy).

The most important drawback of the proposed approach, as well as of any volumetric approach, is the limited resolution imposed by memory requirements. Representing a surface, which is a 2D object, in a 3D volume leads to very inefficient memory usage since most of the voxels are unused. There are several proposed solutions for this problem such as run-length encoding [5] or octrees. However, those methods are just more sophisticated data structures and they do not reflect any geometric properties of the shape they represent. We believe that more efficient and useful compression schemes are possible; this will be the next step in our research.

6 Acknowledgements

The authors wish to express their gratitude to Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR) and Natural Sciences and Engineering Research Council of Canada (NSERC) supporting this research.

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] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *International Journal of Image and Vision Computing*, 10(3):145–155, April 1992.
- [4] 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.
- [5] 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.
- [6] G. Dalley and P. Flynn. Range image registration: A software platform and empirical evaluation. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling*, pages 246–253, May 2001.
- [7] M. Greenspan and G. Godin. A nearest neighbor method for efficient icp. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling*, pages 161–168, May 2001.
- [8] A. Hilton and J. Illingworth. Geometric fusion for a hand-held 3d sensor. *Machine vision and applications*, 12:44–51, 2000.
- [9] O. Jokinen. Area-based matching for simultaneous registration of multiple 3-d profile maps. *Computer Vision and Image Understanding*, 71(3):431–447, September 1998.
- [10] 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.
- [11] T. Masuda. A unified approach to volumetric registration and integration of multiple range images. In *Proceedings of the 14th International Conference On Pattern Recognition*, pages 977–981, August 1998.
- [12] 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.
- [13] P. J. Neugebauer. Reconstruction of real-world objects via simultaneous registration and robust combination of multiple range images. *International Journal of Shape Modeling*, 3(1,2):71–90, 1997.
- [14] H. G. R. Bergevin, M. Soucy and D. Laurendeau. Towards a general multi-view registration technique. *Pattern Analysis and Machine Intelligence*, 18(5):540–547, May 1996.
- [15] G. Roth and E. Wibowoo. An efficient volumetric method for building closed triangular meshes from 3-d image and point data. In W. Davis, M. Mantei, and V. Klassen, editors, *Graphics Interface*, pages 173–180, May 1997.
- [16] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling*, pages 145–152, May 2001.
- [17] G. Turk and M. Levoy. Zippered polygon meshes from range images. *SIGGRAPH '94 Conference Proceedings*, 26:311–318, 1994.
- [18] J. Williams and M. Bennamoun. Simultaneous registration of multiple corresponding point sets. *Computer Vision and Image Understanding*, 81(1):117–142, January 2001.

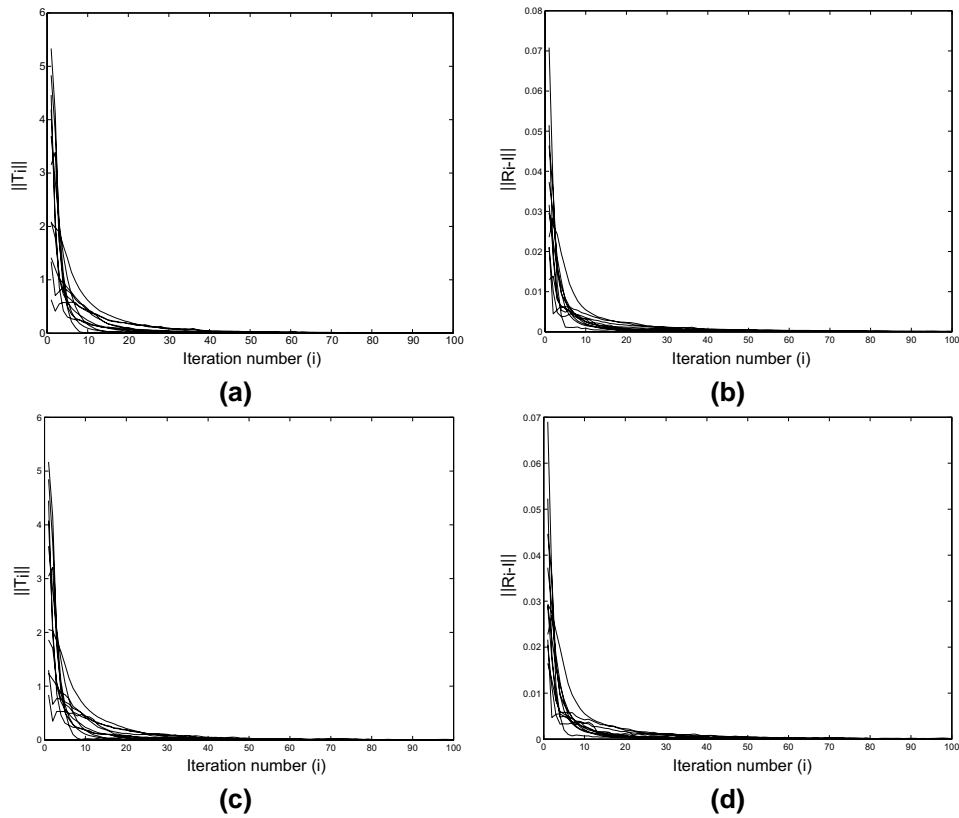


Figure 6. Change of transformation parameters as a function of the number of iterations. Top row: Change of translation (a) and rotation (b) for 12 noiseless images of Beethoven. Bottom row: Change of translation (c) and rotation (d) for 12 images corrupted by noise of variance 1 voxel.



Figure 7. 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).

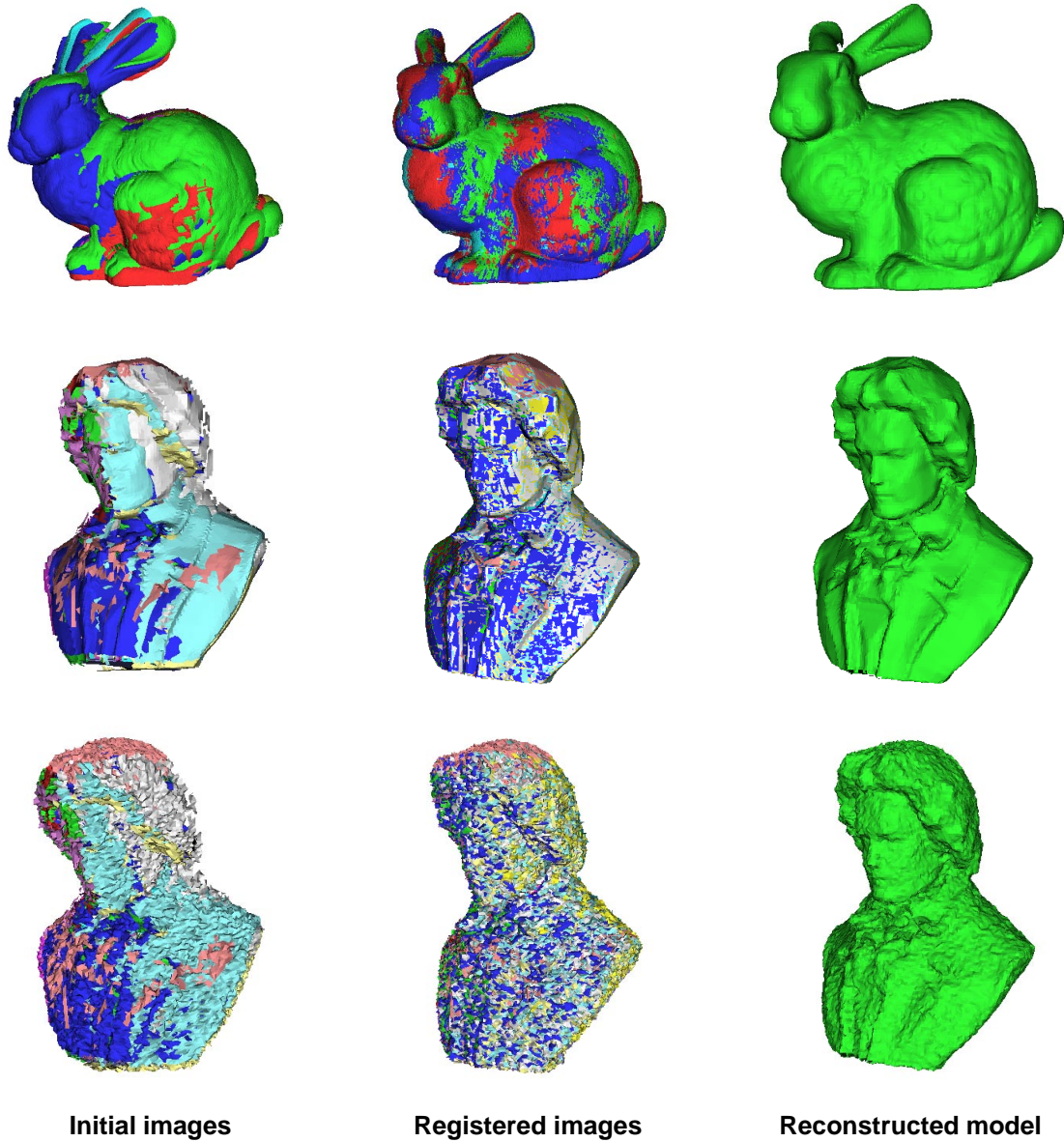


Figure 8. Examples of reconstruction and registration. Top row: real range data from Stanford image repository. Middle row: registration and reconstruction using noiseless synthetic data. Bottom row: registration and reconstruction using noisy images (noise variance 1 voxel).