

Real Time Surface Registration for PET Motion Tracking

Jakob Wilm¹, Oline V. Olesen^{1,2,3}, Rasmus R. Paulsen¹, Liselotte Højgaard²,
Bjarne Roed³, and Rasmus Larsen¹

¹ Informatics and Mathematical Modelling, Technical University of Denmark
Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, Denmark
<http://imm.dtu.dk/>

² Department of Clinical Physiology, Nuclear Medicine & PET, Rigshospitalet,
Copenhagen University Hospital, University of Copenhagen

³ Siemens Healthcare, Siemens A/S, Denmark

Abstract. Head movement during high resolution Positron Emission Tomography brain studies causes blur and artifacts in the images. Therefore, attempts are being made to continuously monitor the pose of the head and correct for this movement. Specifically, our method uses a structured light scanner system to create point clouds representing parts of the patient's face. The movement is estimated by a rigid registration of the point clouds. The registration should be done using a robust algorithm that can handle partial overlap and ideally operate in real time. We present an optimized Iterative Closest Point algorithm that operates at 10 frames per second on partial human face surfaces.

Keywords: motion tracking, registration, ICP.

1 Introduction

High resolution medical imaging modalities are highly sensitive to patient movement during image acquisition. The effect of patient movement during a dynamic Positron Emission Tomography (PET) recording is shown in Fig. 1. Acquisition times vary from several minutes to a few hours.

Since the invention of PET in the 1970's [1], resolution has been ever increasing. Our motion tracking method is targeted at the Siemens High Resolution Research Tomograph (HRRT), which features a spatial resolution below 2 mm [2]. This is well below the average drift that is observed in healthy subjects during a 45 min period [3]. The problem of patient movement is often mitigated by the use of thermoplastic head restraints or vacuum pillows. However, it is not possible to completely avoid motion using these methods [4].

A popular method for real time motion tracking is the use of an optical tool tracker such as Polaris Vicra [5]. The difficulty with such systems is to keep the optical tool well attached to the patients head, and in the field of view (FOV) of the tool tracker. This is particularly cumbersome in the narrow gantry of the HRRT PET scanner.

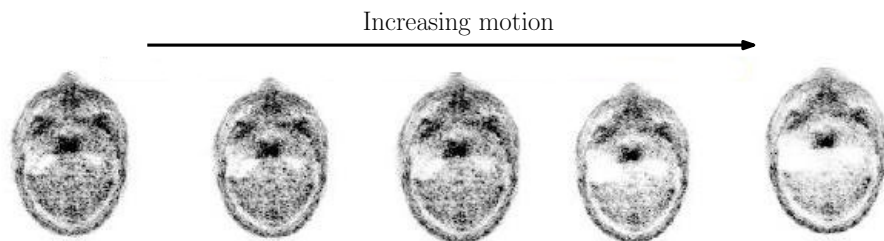


Fig. 1. A simulation of the effect of patient movement on a dynamic PET recording with radioactive tracer [11-C] Verapamil. Notice the white regions of no diagnostic value that appear with increased motion. Modified from [6].

We have previously presented a structured light system for motion tracking [7]. In contrast to the optical tool tracker, our method 1) does not need any optical tool and preparation of the subject, 2) fits into the narrow gantry of the HRRT PET scanner and 3) can potentially be built into future medical scanners.

In this paper, we focus on the registration of surfaces emerging from the current system. Ideally such registrations should be performed in real time, enabling online motion correction. In a clinical PET environment, this has many advantages over post processing, in that it allows for interventions based on the results, and improves clinical work flow.

The standard framework for rigid registration of point clouds is the Iterative Closest Point (ICP) algorithm. It was introduced by Chen and Medioni [8] and in a similar version by Besl and McKay [9]. It has gained wide spread popularity due to its simple formulation and low complexity. Besl and McKay use a point to point error metric. They also propose an extrapolation scheme to reduce the number of necessary iterations.

Several authors have proposed enhancements to the speed and robustness of the algorithm. Most of these are usage specific. An overview of ICP variants is given by Rusinkiewicz and Levoy in [10]. In the same paper they introduce the concept of normal space sampling to emphasize feature rich regions of the surfaces. This paper demonstrates the usefulness of such feature extraction methods.

For partial surfaces, Turk and Levoy suggested to reject point pairs that match to a mesh boundary [11]. A similar effect might be achieved by rejecting point pairs that are far apart [10]. In most real world situations, including our own, one of these techniques is necessary, because the algorithm would otherwise create correspondences between points that are not present in both surfaces.

As several authors have noted, the determination of nearest neighbors is by far the most time consuming step in the algorithm. Different acceleration structures are employed to reduce computation time. These can be grouped into space partitioning structures such as quad-trees [12], kD-trees [13] or Voronoi diagrams and data partitioning like Elias algorithm and bucket sorting into a uniform grid. Using these methods, the complexity of nearest neighbor searches can be reduced from the $O(N^2)$ of brute force searching to an expected $O(N \log N)$ in kD-trees [13] and $O(N)$ for the uniform grid. They do, however necessitate

a single computation of the search structure for each ICP run. Greenspan and Yurick show, that in many cases, approximate neighbors are sufficient for the ICP algorithm to converge [14]. In kD-trees such approximate lookups can be done using the highly optimized ANN library by Mount and Arya [15]. A caching kD-tree implementation by Nüchter et al. was reported to halve computation times in ICP [16].

The use of both kD-trees and the uniform grid is discussed in this paper. We finally present an optimized version of the ICP algorithm that is particularly well suited for fast and robust registration of human face scans.

2 Methods

Our structured light scanner uses two cameras and a Pico projector that are mounted just above the patient tunnel. The setup is shown in Fig. 2. A sequence of patterns is projected onto the patient’s face. Based on the distortion of these patterns, and using a Phase Shifting Interferometry (PSI) algorithm, point clouds are created. The scanner’s FOV is set to cover the region around the eyes, which in most cases is free of soft tissue deformations during image acquisition.

The system creates point cloud representations of a part of the patient’s face. These are reconstructed to surfaces and inter frame motion is estimated using a rigid registration with the ICP algorithm.

We process our data to create a single polygonal surface from the input of the two cameras. The reconstruction and merging of the two point clouds is done in one step using the Markov Random Field surface reconstruction algorithm [17]. This reconstruction method extracts an isosurface from a regularized distance field, and has been shown to be particularly well suited for human surface scans [18]. We call one such reconstructed surface a single frame. The frames contain approximately 50 k triangles and 30 k points

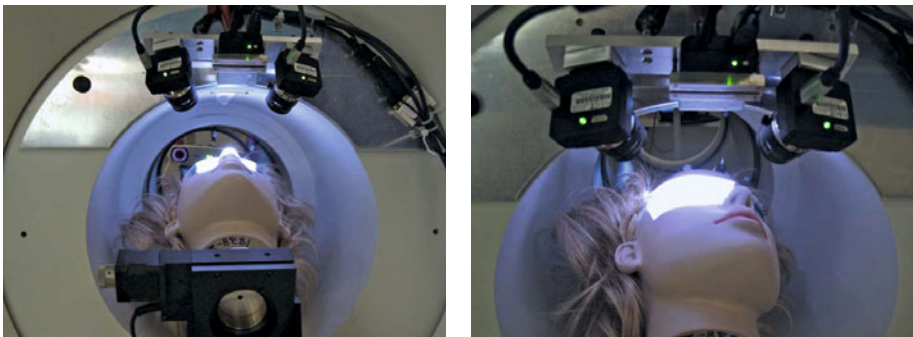


Fig. 2. Photographs of a mannequin head inside the HRRT PET scanner with the SL system in the front mounted to the gantry

To perform motion tracking, every frame must be registered to some reference frame. To evaluate our algorithm, we use the simulated registration problem shown in Fig. 3. It shows a surface resembling part of a mannequin head and it was preprocessed as described above. We transform the same surface using known parameters and add Gaussian noise individually to the target and reference. This way, we know the correct point correspondences, and we evaluate the registration results objectively as they are compared to ground truth, instead of the value of the objective function. For some experiments we remove parts of both surfaces, as shown, to create partial overlap.

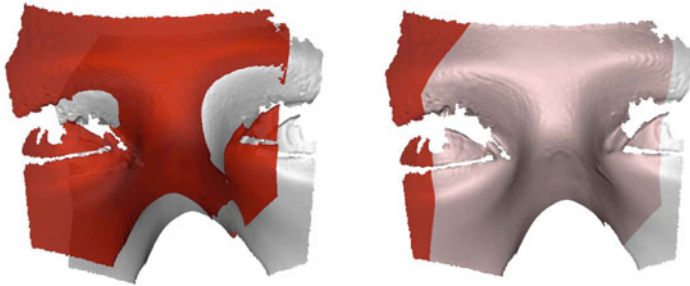


Fig. 3. A simulated registration scene with a mannequin head. There are approximately $N = 30\text{ k}$ points on each surface. The left side shows the situation before registration, the right side after. We have used a known transformation. Parts of both surfaces were removed to simulate the effect of partial overlap.

We state the problem of aligning the point set $\mathbf{P} = \{p_i\}$ for $i = 1 \dots N_p$, to another point set, $\mathbf{Q} = \{q_j\}$ for $j = 1 \dots N_q$, where individual point correspondences are not known.

In order to address the issue of missing point correspondence, the ICP algorithm iteratively performs the following steps until convergence

1. Matching: every data point in \mathbf{P} is matched to a point in the model point set \mathbf{Q} to form the nearest neighbor pair (p_i, \hat{q}_i) .
2. Minimization: the error metric is minimized.
3. Transformation: data points are transformed using the minimization result.

The point to point error metric is

$$E = \sum_{i=1}^N \|\mathbf{R}p_i + \mathbf{T} - \hat{q}_i\|^2, \quad (1)$$

where \mathbf{R} is an orthogonal rotation matrix and \mathbf{T} is a translation vector.

The minimization of E has several closed form solutions, using either quaternions [19], orthonormal matrices [20] or the singular value decomposition (SVD) [21]. We use the SVD because it results in the fewest calculations.

We consider sub sampling of the surfaces for two reasons. Firstly, it reduces the overall computational cost of ICP registration. Most time is spend on nearest neighbor searches with a complexity of at most $O(N^2)$ for the brute force approach, and we expect that halving the number of points can quarter computation times at best. The other reason is, that an appropriate sampling strategy will improve ICP convergence.

By bucketing the points according to their normal direction, and sampling across these buckets, the relative sampling density in feature rich regions is increased. A reasonable choice is to divide the unit sphere into 27 equally sized regions to obtain the normal buckets. A similar exploitation of differential properties is sampling across points of different curvature values. We use a method for curvature estimation on point clouds due to Pauly et al. [22]. The authors demonstrate, that the variance in a point's neighborhood can be used to approximate the surface mean curvature. For our surfaces a local neighborhood size of 100 points divided into 10 buckets works well.

For point clouds that have only partial overlap, point pairs involving the border of the target might be rejected [11]. This is illustrated conceptually in Fig. 4. Border points are identified as those defining edges that are only part of one triangle. With our data, this is easy to implement, because connectivity information is available for the data. Also, the computational cost is very low.

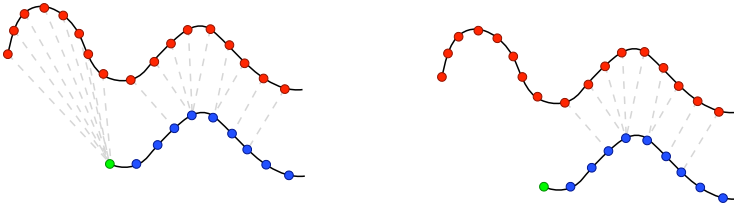


Fig. 4. In surfaces with partial overlap, many erroneous matches to the border of the target might occur. In border rejection, these pairs are discarded. Modified from [10].

Statistical approaches to remove erroneous matches require no triangulation of the target. Albeit, we don't consider them as robust as the border point rejection, because the latter will let surfaces *slide in place* even if the degree of overlap is small.

An important observation concerning the algorithm is that, in many iterations, it will make small updates in approximately the same direction. This will in some cases allow us to extrapolate the results of preceding iterations to move quicker.

Using unit quaternions to represent the current rotation, the transformation may be represented by a seven dimensional vector with six degrees of freedom:

$$\mathbf{q} = [q_w \ q_x \ q_y \ q_z \ T_x \ T_y \ T_z]^T$$

Letting \mathbf{q}_k denote the total transformation state vector at the k th iteration, one may define the change in every iteration as

$$\Delta_k = \mathbf{q}_k - \mathbf{q}_{k-1}$$

The angle between the last two directions is

$$\theta_k = \text{acos} \left(\frac{\Delta_k \cdot \Delta_{k-1}}{\|\Delta_k\| \|\Delta_{k-1}\|} \right)$$

If the angles θ_k and θ_{k-1} are small, extrapolation might be considered. It is noteworthy that the computational overhead of extrapolation is insignificant and that, it can be combined with all other ICP modifications presented here. Details of the technique are described in the paper of Besl and McKay [9].

As noted before, the most time consuming part of ICP is the determination of nearest neighbors. The points on the reconstructed surfaces are close to equally spaced, which is the result of re-meshing in our surface reconstruction. For this reason, we consider bucket sorting the points into a uniform grid as an alternative to the popular kD-tree approach. Naturally, bucketing works best, when the data points are distributed uniformly such that every cell is occupied by exactly one point, in which case the algorithm provides $O(N)$ complexity. Point matches are made by spiraling out from the query point cell and calculating the distances to all points encountered. When the closest point so far is closer than all unvisited cells, the search is terminated. It should be noted that the performance of bucketing can rapidly decrease in the case of outliers, which cause the data bounding box to grow. For our data, this does not present an issue, because the Markov Random Field surface reconstruction removes outliers in the data.

3 Results and Discussion

Results for uniform sampling, normal-space sampling and curvature sampling are shown in Fig. 5. For this evaluation the scene shown in Fig. 3 is used with full overlap. Therefore the results converge towards zero. It is seen, that normal space and curvature space sampling perform well and yield faster convergence than normal uniform sampling. Because normal sampling requires fewer computations in our case, we use it in our implementation.

We compare the border rejection scheme with dynamic rejection of all pairs whose distance is either in the upper 10% or more than 2.0 times the standard deviation. Here we use the scene depicted in Fig. 3 in which model and data have only partial overlap. The results of using these rejection methods are seen in Fig. 6. It is observed, that for this scene, a rejection strategy is absolutely necessary. Whilst the border rejection converges slower than the statistical methods we consider it more robust because of its parameter independence.

The result of using quaternion extrapolation on our registration problem is shown in Fig. 7. Extrapolation occurs at several instances throughout the registration, and it reduces the number of iterations significantly. Because extrapolation introduces basically no overhead, we use it in our implementation.

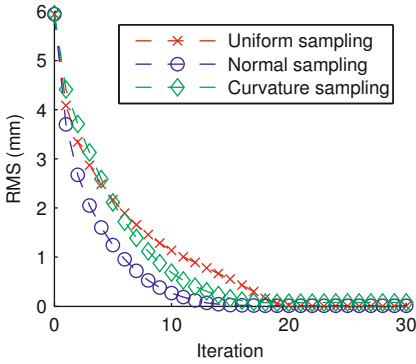


Fig. 5. Convergence for different sampling strategies. The surfaces with full overlap were used. In all cases, 1/4 of the available points were used.

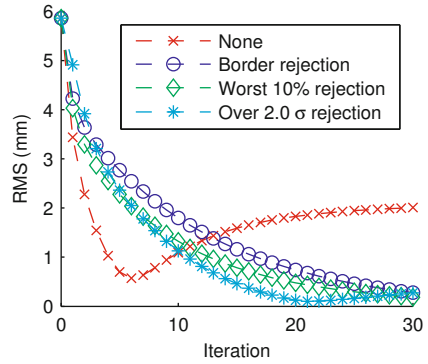


Fig. 6. Convergence for different rejection strategies on the surfaces with partial overlap

Nearest neighbor search methods are compared in Fig. 8. We use the ANN library by Arya and Mount for exact kD-tree searches. Both target and reference contain the full surfaces. It is seen that bucketing in a uniform grid results in nearest neighbor search times lower than what is achieved with the highly optimized ANN library. Both methods are much faster than the brute force approach.

Our custom registration algorithm builds upon the preceding discussion and results of sampling, border rejection and extrapolation. It samples 1/4 of the points from across 27 normal buckets. Point pairs involving the model border

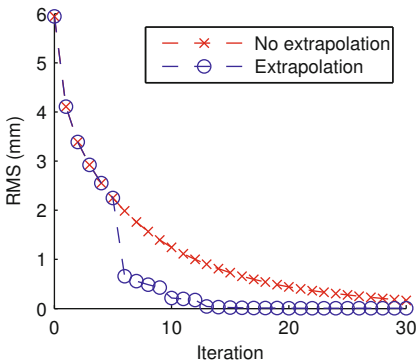


Fig. 7. Convergence behaviour for the test scene with and without extrapolation in transformation space

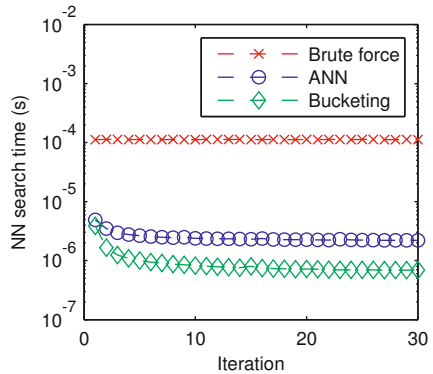


Fig. 8. Nearest neighbor search times versus iteration count for the test scene with $N = 29,678$ points in both model and data

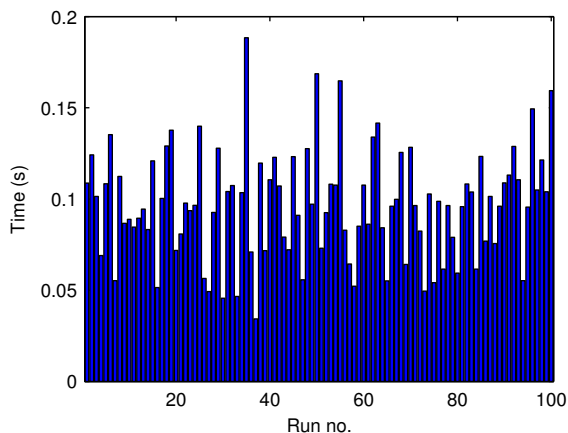


Fig. 9. Alignment times for 100 runs of our robust ICP algorithm with different random transformations. The measurements have mean 0.1 s and standard deviation 0.03 s.

are rejected. Extrapolation in quaternion space is performed to speed up the algorithm. Matching is done using bucketing in a uniform grid. The custom registration algorithm is run 100 times with different transformations. In every run, the rotation axis and angle are drawn from a uniform distribution. Rotation angles vary from -5 to $+5$ degrees. Translations are also uniformly distributed in the interval -5 mm to $+5$ mm. In every run, the algorithm is terminated when the change in RMS value is below $1 \mu\text{m}$. Measurements are performed on a 2.16 GHz machine. The results are seen in Fig. 9.

The registration times have mean 0.1 s and standard deviation 0.03 s. This makes our implementation suitable for real time motion tracking in our PET application.

4 Conclusion

We have presented a method for motion tracking during PET scans. It builds upon a structured light system, surface reconstruction and a highly optimized version of the ICP algorithm. The algorithm is specifically designed for fast alignment of facial scans. Quantitative experiments show that it is able to run at approximately 10 fps. This makes our implementation suitable for real time motion tracking in a clinical situation. In the future, this will result in much fewer artifacts on high resolution PET. Our system may be easily adapted to other imaging modalities in which motion artifacts are a major source of error. For the optimized ICP algorithm, we see many other applications that reach beyond the medical imaging field.

References

1. Ter-Pogossian, M.M., Phelps, M.E., Hoffman, E.J., Mullani, N.A.: A Positron-Emission Transaxial Tomograph for Nuclear Imaging (PETT). *Radiology* 114, 89–98 (1975)
2. Olesen, O.V., Sibomana, M., Keller, S.H., Andersen, F., Jensen, J., Holm, S., Svarer, C., Højgaard, L.: Spatial resolution of the HRRT PET scanner using 3D-OSEM PSF reconstruction. In: *Nuclear Science Symposium Conference Record 2009*, pp. 3789–3790 (2009)
3. Dinelle, K., Blinder, S., Cheng, J.C., Lidstone, S., Buckley, K., Ruth, T.J., Sossi, V.: Investigation of Subject Motion Encountered During a Typical Positron Emission Tomography Scan. In: *Nuclear Science Symposium Conference Record 2006*, pp. 3283–3287 (2006)
4. Green, M.V., Seidel, J., Stein, S.D., Tedder, T.E., Kempner, K.M., Kertzman, C., Zeffiro, T.A.: Head Movement in Normal Subjects During Simulated PET Brain Imaging with and without Head Restraint. *Journal of Nuclear Medicine* 35(9), 1538–1546 (1994)
5. Rahmim, A.: Advanced Motion Correction Methods in PET. *Iranian Journal of Nuclear Medicine* 13(24), 1–17 (2005)
6. Anton-Rodriguez, J.M., Sibomana, M., Walker, M.D., Huisman, M.C., Matthews, J.C., Feldmann, M., Keller, S.H., Asselin, M.: Investigation of Motion Induced Errors in Scatter Correction for the HRRT Brain Scanner (to appear)
7. Olesen, O.V., Jørgensen, M.R., Paulsen, R.R., Højgaard, L., Roed, B., Larsen, R.: Structured Light 3D Tracking System for Measuring Motions in PET Brain Imaging. In: *Proceedings of SPIE 7625 76250X* (2010)
8. Chen, Y., Medioni, G.: Object Modeling by Registration of Multiple Range Images. In: *1991 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2724–2729 (1991)
9. Besl, P.J., McKay, N.D.: A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2), 239–256 (1992)
10. Rusinkiewicz, S., Levoy, M.: Efficient Variants of the ICP Algorithm. In: *Third International Conference on 3-D Digital Imaging and Modeling*, p. 145 (2001)
11. Turk, G., Levoy, M.: Zipped Polygon Meshes from Range Images. In: *ACM SIGGRAPH 1994*, pp. 311–318 (1994)
12. Finkel, R.A., Bentley, J.L.: Quad Trees A Data Structure for Retrieval on Composite Keys. *Acta Informatica* 4(1), 1–9 (1974)
13. Friedman, J.H., Bentley, J.L., Finkel, R.A.: An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software* 3(3), 209–226 (1977)
14. Greenspan, M., Yurick, M.: Approximate K-D Tree Search for Efficient ICP. In: *Fourth International Conference on 3-D Digital Imaging and Modeling, 3DIM 2003*, pp. 442–448 (2003)
15. Arya, S., Mount, D.: ANN: A Library for Approximate Nearest Neighbor Searching. In: *2nd CGC Workshop on Computational Geometry* (1997)
16. Nüchter, A., Lingemann, K., Hertzberg, J.: Cached k-d tree search for ICP algorithms. In: *Sixth International Conference on 3-D Digital Imaging and Modeling, 3DIM 2007*, pp. 419–426 (2007)
17. Paulsen, R.R., Larsen, R.: Anatomically Plausible Surface Alignment and Reconstruction. In: *Proceedings of Theory and Practice of Computer Graphics 2010, Eurographics UK* (2010)

18. Paulsen, R.R., Bærentzen, J.A., Larsen, R.: Markov Random Field Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 636–646 (2009)
19. Horn, B.K.P.: Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A* 4(4), 629–642 (1987)
20. Horn, B.K.P., Hilden, H.M., Negahdaripour, S.: Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A* 5(7), 1127–1135 (1988)
21. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9(5), 698–700 (1987)
22. Pauly, M., Gross, M., Kobbelt, L.P.: Efficient Simplification of Point-Sampled Surfaces. In: *IEEE Visualization, VIS 2002*, pp. 163–170 (2002)