ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume I-3, 2012
XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia

OCTREE-BASED SIMD STRATEGY FOR ICP REGISTRATION AND ALIGNMENT OF 3D POINT CLOUDS

D. Eggert, S. Dalyot

Institut für Kartographie und Geoinformatik
Leibniz Universität Hannover
Appelstr. 9A, 30167 Hannover, Germany
{eggert, dalyot} @ikg.uni-hannover.de
http://www.ikg.uni-hannover.de

Commission III/2

KEY WORDS: LIDAR, Matching, Processing, Application, Three-dimensional, Registration, Point Cloud, Performance

ABSTRACT:

Matching and fusion of 3D point clouds, such as close range laser scans, is important for creating an integrated 3D model data infrastructure. The Iterative Closest Point algorithm for alignment of point clouds is one of the most commonly used algorithms for matching of rigid bodies. Evidently, scans are acquired from different positions and might present different data characterization and accuracies, forcing complex data-handling issues. The growing demand for near real-time applications also introduces new computational requirements and constraints into such processes. This research proposes a methodology to solving the computational and processing complexities in the ICP algorithm by introducing specific performance enhancements to enable more efficient analysis and processing. An Octree data structure together with the caching of localized Delaunay triangulation-based surface meshes is implemented to increase computation efficiency and handling of data. Parallelization of the ICP process is carried out by using the Single Instruction, Multiple Data processing scheme - based on the Divide and Conquer multi-branched paradigm - enabling multiple processing elements to be performed on the same operation on multiple data independently and simultaneously. When compared to the traditional non-parallel list processing the Octree-based SIMD strategy showed a sharp increase in computation performance and efficiency, together with a reliable and accurate alignment of large 3D point clouds, contributing to a qualitative and efficient application.

1 INTRODUCTION

In recent years we have experienced a growing market demand for range sensors required for vast applications, as well as the availability of numerous processing algorithms of range data. Similarly to automatic image mosaicking widely used and applicable today for creating a wide coverage imagery data, it is only logical that there is also a growing need for fast and fully automatic solutions of registering and aligning of range data. Registration and alignment, and thus fusion, of 3D point clouds, acquired for example via close range laser scans, is important for creating an integrated 3D model data infrastructure. Since evidently the different scans are acquired from different positions and might present different data characterization and accuracy complex data-handling and processing issues are enforced to such registration and alignment process. Also, the growing demand for fast processing time applications introduces new computational requirements and constraints.

Registration and alignment of range scans consists of matching and estimation, usually based on rigid transformation. The challenge in such a process is to perform it automatically and robustly independent of different geometric constraints, such as small spatial overlap of the scans and their different characteristics. The Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992) and its variants (Rusinkiewicz and Levoy, 2001) is perhaps the most commonly used algorithm for matching of rigid bodies. ICP is based on coupling-up pairs of corresponding points (query- and search-points), considered the nearest ones existing in the matched models, and thus estimating rigid body transformation that aligns models best. This transformation is applied to one model while the procedure continues iteratively until convergence is achieved, e.g., spatial alignment. ICP can be referred to as a low-level method for finding rigid body transformation (i.e., relative motion) between two (or more) point clouds of the same scene acquired from different viewpoints. The algorithm works in an iterative fashion to find the correct closest correspondences and spatial transformation.

The main challenge in extracting the transformation is the matching of the points in one model to those of the other, referred to as query- and search-points, respectively. k Nearest Neighbor (kNN) computation involved with the ICP algorithm is perhaps the major computational 'bottleneck': most of the computation time is usually taken up by finding corresponding points, rather than computing the transformation. (Besl and McKay, 1992) reported that 95% of the run-time is consumed for searching these correspondences. Furthermore, the aligned range-scans most evidently will present different distribution of data, in terms of density and level of detail (LOD), in addition to data and measurement errors and noise. These factors will result in a condition where there is no explicit one-to-one correspondence of individual points in the point clouds, e.g., each query-point has an associated search-point that was sampled from the same object spatial position, thus impairing the quality of the registration and alignment process.

The aim of this research is to solve the above mentioned computational and processing complexities introduced in the ICP algorithm. This is achieved by introducing specific performance enhancements. To resolve the kNN computational 'bottleneck' we propose the use of an Octree data structure, which divides the search space into convex voxels, enabling a much faster kNN search and data-handling capabilities. To resolve the non-explicit one-to-one correspondence, we employ a point-to-plane matching scheme - instead of the original point-to-point one. This is carried out by implementing localized Delaunay triangulation based surface meshes. The localized Triangulated Irregular

Network (TIN) meshes are cached during processing and reused when required. In order to parallelize the ICP process and execute the entire process more efficiently, the use of the Single Instruction, Multiple Data (SIMD) processing scheme with the Divide and Conquer (D&C) multi-branched paradigm is implemented. It was found that execution speed increased when compared to non-parallel point-list processing.

2 RELATED WORK

Spatial point-based matching and modeling algorithms for rigid bodies depend mainly on the geometry type of features that are needed to be matched, their topological relations, models volumes, and the semantic attributes (Mount et al., 1999). The majority of ICP algorithm implementations utilize a 6-parameter rigid body transformation without uniform scale factor, mainly due to the fact that the aligned scans exist on the same space. The parameters of the rigid body transformation are generally estimated by the use of closed-form solutions, mainly Singular Value Decomposition (SVD) (Horn et al., 1988) and Least Squares Matching (LSM) (Gruen and Akca, 2004), while usually calculating a robust and global maxima estimation regarding the rigid body spatial transformation. Still, local minima solution might also exist in case of large amount of data-outliers, noise and wrong (or missing) preliminary rough registration parameters. (Zhang, 1994) showed that these closed form solutions cannot fully consider the statistical point error models, so if necessary an alternative or combined algorithm is required. Several recent strategies were suggested for localization and elimination of outliers (rejection of pairs) and occlusions (such as: (Dalley G., 2002), (Fitzgibbon, 2003), and (Guehring, 2001)).

ICP requires heavy computational complexity, such that the original algorithm is of order $\mathcal{O}(n^2)$. A parallel computing system was proposed by (Langis et al., 2001), while (Qiu et al., 2009) proposed the use of Graphical Processing Units (GPUs) to reduce processing time and accelerate convergence. Still, the main emphasize in both strategies has been put on hardware independent solutions only. Several attempts also addressed the monotonic convergence behavior of the ICP by suggesting quadratic-type solution by using linear or parabolic types of extrapolations, as well as helical motion in the parameter space (Fitzgibbon, 2003). Still, such solutions might suggest an over shooting of the true solution, and deteriorating the orthogonality of the rotation matrix. Another acceleration solution aiming at reducing the number of points used (sub-sampling of data by random or regular sampling), such as the hierarchical coarse to fine strategy (Guehring, 2001), alters the resolution (lowest to highest) as the process approaches convergence. Though such strategies can give satisfactory results, sub-sampling strategies are very sensitive to noise level, occlusion areas, complexity of the object, etc., while they do not fully exploit the highest achievable accuracy level of the registration.

Search strategies accelerate the alignment by restricting the search space to a subpart of the data. The k-D tree (k Dimensional binary tree) is perhaps still the most utilized nearest neighbor search method with ICP (Greenspan and Yurick, 2003). Though the average performance of the k-D tree search suggests more efficiency, it is of order $\mathcal{O}(n \log n)$; also, constructing a k-D tree can be quite a complicated task with large point-cloud while consuming significant amount of time. An option might suggest the use of Octree search criteria, which is the 3D analogy of the quad tree. (Wang and Shan, 2005) showed that based on the principle of Hilbert space filling curves, this search criteria provides fast access and spatial query and neighborhood mechanisms. (Strand et al., 2007) also proved that this data-structure enabled qualitative processing in terms of large odometry errors existed among the different point-clouds. (Szeliski, 1996) used an approximate precomputed distance map named Octree spline whose resolution increases hierarchically near the surface. Still, storing the complete uniform distance map at the desired accuracy can be expensive in terms of memory requirements.

3 METHODOLGY AND STRATEGY

3.1 ICP Mathematical Considerations

Two - or more - 3D point-cloud representations are given, which are considered as representing different scans or viewpoints of the same object. The point-clouds - congruent or partially congruent - can be depicted as two 3D functions - subject and reference, denoted as $g(x, y, z)$ and $f(x, y, z)$, respectively. Each function is a discrete 3D approximation of the model. Due to the existence of random errors, an error vector $e(x, y, z)$ is added to describe the association (cross-relations) and magnitude of correspondence of the models (Akca and Gruen, 2005). This error vector can be described as the shift surface of the two models (subject and reference), mostly composed of local random errors but also of global systematic ones.

A rigid n-parameter 3D transformation, which satisfies for instance a LSM goal function between these surfaces, has to be estimated by measuring the sum of the squares of the Euclidean distances between the representations, e.g., 3D functions. The n-parameter goal function can be extended or reduced depending on the deformation(s) between the surfaces - as required by the situation at hand (since this research aims at efficiency strategies, a 3-parameter transformation denoted by translation only was used). To perform Least Squares estimation, i.e., linearization by Taylor series, only the linear terms are retained, depicted in Equations (1) and (2), where, w_i defines a certain parameter's differentiation, i.e., $i \in [t_x, t_y, t_z]$, denoting the translations.

$$f(x,y,z) - e(x,y,z) = g(x,y,z) + \frac{\partial g^0(x,y,z)}{\partial x}dx + \frac{\partial g^0(x,y,z)}{\partial y}dy + \frac{\partial g^0(x,y,z)}{\partial z}dz \quad (1)$$

$$dx = \frac{\partial x}{\partial w_i};\;\; dy = \frac{\partial y}{\partial w_i};\;\; dz = \frac{\partial z}{\partial w_i} \quad (2)$$

Each observation formula is related to a linear combination of the parameters, which basically are variables of a deterministic unknown (Gauss-Markov theorem, (Plackett, 1950)) with unbiased minimum variance estimation for the parameters. On each iteration the subject representations is transformed 'toward' the reference one while updating the values in the vector of transformation parameters. Consequently, the design matrix A, which holds the derivatives of the unknown parameters, and the discrepancies vector l are recalculated based on the new 'position' of data. The iterations continue until all differential values of the n-elements in the solution vector $\overline{x}$ are smaller than a certain predefined threshold (or any other statistical combination of thresholds). The standard deviations of the calculated transformation parameters are supposedly reliable, mainly due to the fact that the representations used in the process have stochastic properties, and due to the nature of the LSM process (Gruen, 1996).

3.2 Mutual Points Correspondences

Since mutual homologous query- (subject) and search- (reference) points does not explicitly exist in both data representations, the point-to-plane (tangent) strategy was used in this research;

e.g., searching for the 'closest' search-point to the query-point (after transformation) that upholds the continuous function of the reference representation, i.e., $f(x, y, z)$. Consequently, this tangent search criteria process upholds the following constraints, by implementing Least Squares target function minimization:

1. The search-point upholds a surface function existing in a local plane in the reference representation (a TIN mesh was used in this research, e.g., triangular-plane, represented by heights Z_1, Z_2, and, Z_3); and,
2. Query-point after transformation $g^t(x, y, z)_i$ and search-point $f(x, y, z)_i$ form a vector that has to be a tangent to the local plane function (shortest path transformed from query-point to local plane). Thus, first order derivative in both x (2^{nd}constraint) and y (3^{rd}constraint) are generated.

The schematics of this point-to-plane (tangent) strategy are given in Figure 1. It can be described as if these criteria of constraints shift the vector between the points $g^t(x, y, z)_i$ (dark-grey) and $f(x, y, z)_i$ (black) over the local triangle plane towards its final position, thus minimizing the Least Squares target function until converging to the final positioning of $f(x, y, z)_i$.

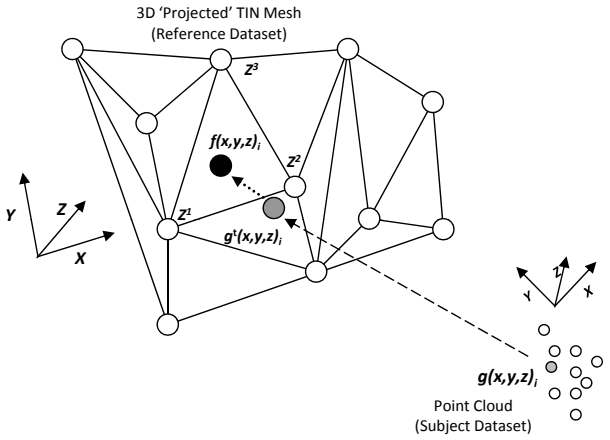

Figure 1: Schematic explanation of Least Squares target function minimization via three constraints

For each query-point i in dataset g a nearest search-point from dataset f is paired-up as long as the three constraint criteria are fulfilled. Consequently, with all the point pairs extracted, a rigid transformation is achieved. The ICP process is carried out iteratively until predefined statistical criteria are achieved, which are based on:

1. Value differences of the n-parameter in each consecutive iteration (can also be denoted as spatial difference);
2. Number of iterations - a large number of iterations indicates that both representations functions are different, i.e., matching might be impossible. This number is derived by several factors, such as: number of points, congruency percentage of models, etc.; and,
3. Number of homologous counterpart points - the larger the number, the more robust the matching. A number smaller than a certain threshold indicates that a solution might not be reliable enough.

3.3 Point Cloud TIN Mesh

In order to determine the mutual points correspondences based on the point-to-plane strategy a 3D surface of the reference model has to be established, based on the scan points. The most obvious solution might suggest generating the complete 3D surface of the reference model in advance and to use it during the process. In various experiments we have found that there is no fast - and more importantly - reliable surface reconstruction method for arbitrary point-based laser scans. The mesh created is not an explicit one and existing data outliers, even a small number, produced an unusable 3D surface meshes. Therefore we have decided to create local meshes from the kNN points in the reference model for each query-point. The local meshes are created on demand during the process. At first, the kNN of the query point has to be found in the reference model. It was determined that $k = 32$ neighbors produced good results; this number ascertains that a reliable triangle mesh is constructed, representing the scanned surface: wide enough for the projection task, while avoiding harming computation time. Since the reference dataset is represented by an Octree, the kNN are determined as follows:

1. Find the Octree cell c containing the query-point q and at least k points, while the child s of c (which also contains q) contains less than k points ($\mathcal{O}(1)$); and,
2. Do a linear kNN-search for all points contained in cell c ($\mathcal{O}(k)$).

The second step entails the projection of the kNN points found in c from 3D to 2D space. For that, we compute an oriented bounding box of these points, where the box axes are the eigenvectors of the covariance matrix of the k points. As shown in Figure 2, once the oriented bounding box is computed, the points are projected onto the largest plane of that bounding box, while the correspondences between the original and projected points are stored. The 2D points are triangulated using an ordinary Delaunay triangulation algorithm. Finally, the triangulation is transformed (projected backward) from 2D to 3D space by using the previously stored correspondences. The result is a local 3D surface mesh of the given k nearest neighbor points.

Finally, as described in 3.2, the query-point is projected orthogonally onto the plane of each of the mesh's triangles that were generated. Only if the projected point falls into the bounds of a certain triangle while validating the three constraints, it can be considered valid. Overall, three cases might exist, where the query-point q:

1. Cannot be projected on any of the triangles of the mesh;
2. Can be projected on exactly one triangle; and,
3. Can be projected on several triangles.

In the first case, the query-point is skipped for the current iteration. The second case represents a unique transformation, while the third case is somewhat ambiguous. The ambiguity is solved by choosing the projection with the smallest Euclidean distance between the query-point and the projected point (plane). In order to avoid meshes with outliers, which might be expressed by large triangles, a median triangle area threshold is applied, which is defined by all triangles containing search-points found within the first iteration.

3.4 Caching

Since the process described in 3.3 is computationally intensive, we propose a caching strategy of the already built 3D meshes. Once a mesh is created, it is stored within the Octree, or more precisely within the Octree cell c originally used for the kNN determination. Thus, the next time before a kNN search of a specific query-point is executed, the cache of the corresponding

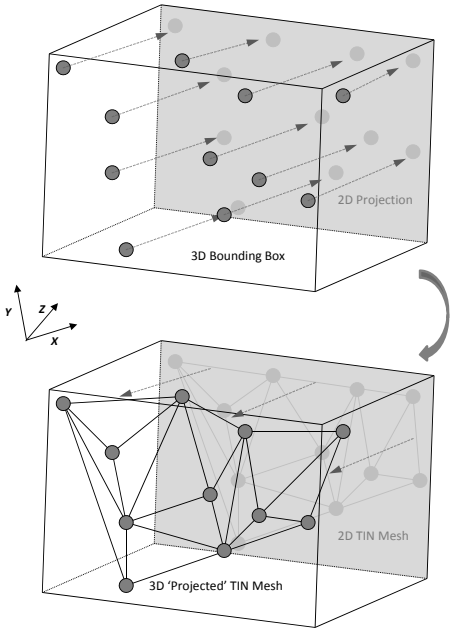

Figure 2: Schematic explanation of 3D 'projected' TIN mesh creation

Octree cell is searched for existing meshes the query-point can be projected on. If the query-point can be projected onto a mesh from the cache the correspondence is stored. Otherwise, if no usable cached mesh is found, all steps described in 3.3 are carried out and a new mesh is built, which will be cached for further reuse.

Due to memory limitations, the size of the cache is limited to a fixed number of meshes per Octree cell. In our experiments, we used a maximum cache size of ten meshes, while new meshes are preferred over old ones. Once the cache for a particular cell contains ten meshes, and a new mesh is required to be cached, the oldest mesh from the cache is removed, while the new one is stored. Since the reference points and their corresponding meshes are not subject to transformations, the cache is preserved over consecutive iterations.

3.5 SIMD Strategy

To reduce computation time we introduce an advanced Octree data structure to achieve fast kNN determination as well as caching for the reuse of existing meshes. As shown in (Eggert and Paelke, 2010), the kNN determination as one of the major computational bottlenecks can be handled via SIMD-based strategies, such as GPGPU (General-Purpose computing on Graphics Processing Units) or utilizing multiple CPU cores in parallel. Thus, we propose a Single Instruction, Multiple Data (SIMD) strategy to exploit modern hardware capabilities. In contrast to the Single Instruction, Single Data (SISD) architecture, SIMD applies the same operation(s)/instruction(s) to not only one data unit, but to multiple ones. Therefore, the processed data has to be independently processable. Moreover, the processing itself has to be designed for parallel execution. As in the proposed ICP process, the transformation for each query-point is determined independently to finally derive a global one. Hence, the process, as well as the data, is suitable for a parallel processing strategy, such as the strategy proposed here. As shown in Figure 3, all instructions involving a single query-point are executed in parallel:

1. kNN determination
2. Oriented Bounding Box (OBB) determination
3. 3D -> 2D point projection
4. 2D Delaunay triangulation
5. 2D -> 3D triangulation projection
6. Point-to-plane projection
7. Transformation calculation

After all individual transformations are calculated, a global transformation is derived from all individual ones. Since all intensive computational parts are parallelized, the process easily scales in respect to the number of available processing units. Meaning that doubling the number of processing units almost cuts the needed computation time in half.

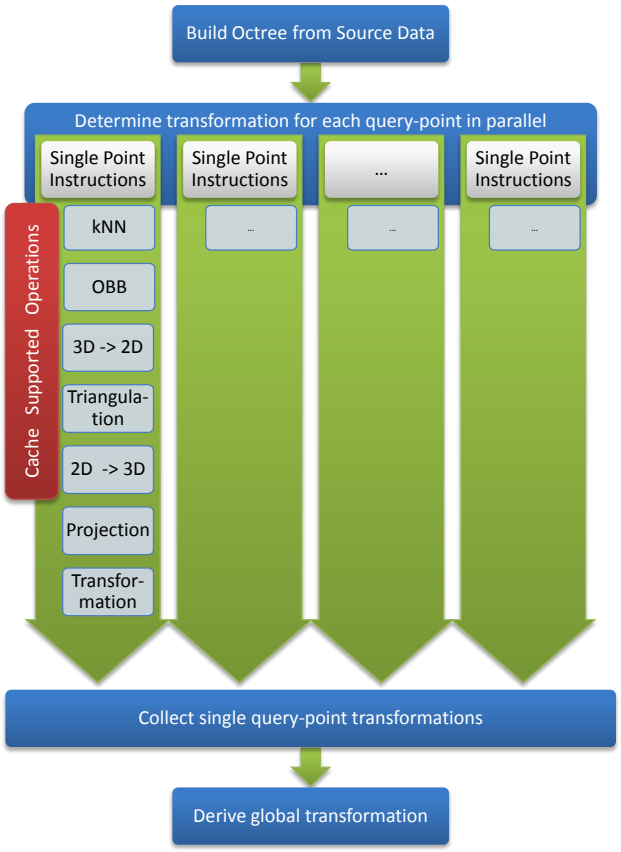

Figure 3: SIMD strategy

Nonetheless, using a SIMD-based processing implementation involves the use of suitable (thread-safe) data structures. While read only operations are usually thread safe, all operations altering the data structure (e.g., adding or removing elements) are not. Since the Octree containing the reference points, as well as the point list containing the subject dataset, are not altered during the process, no precautions have to be taken. In contrast, the cache is extensively altered by adding new and removing old meshes. As the experiments were carried out using a Java-based implementation, a *ConcurrentLinkedQueue* has been used as a thread-safe cache data structure. The other crucial altering data structure is the map storing the single transformation for each point during an iteration. Here, Java provides the thread safe *ConcurrentHashMap*, which serves our needs.

4 EXPERIMENTAL RESULTS AND DISCUSSION

Our experiments were carried out using several different datasets: the well known Stanford bunny model and the Welfenschloss model, which was scanned both by terrestrial and mobile laser scanner. The first experiment presented here involved the registration and alignment of the Stanford bunny model. Since the strategy presented here is designed for an ICP workflow for the registration of a dense as well as a coarse point datasets, we simplified the bunny model, originally containing 36,000 vertices, to a model containing 18,000 vertices. The second experiment presented here involved the registration and alignment of the Welfenschloss model from different scans, each presenting different point-of-view, density and LOD: the mobile laser scan is a more coarse model containing 32,000 vertices, while the terrestrial one is much denser with 168,000 vertices.

The proposed algorithm was implemented using the Java programming language, while SIMD was realized using Java's fork/join framework. The experiments were conducted on an Intel core i7 870 machine with four physical and eight logical cores running at 2.93 GHz.

Since the two bunny datasets are virtually the same model with different LOD, an arbitrary synthetic shift was applied to one of the models. The applied shift was in the range of 20% on the x-axis, 10% on the y-axis, and 0.5% on the z-axis in respect to the models' bounding box. This demonstrates quite a lrage displecemnt of the models, which enable to evaluate the robustness of the proposed methodology in converging to the right alignment. The starting point constellation of the models is depicted in Figure 4 (top-left). The proposed registration and alignemnt algorithm converged after 17 iterations, with a spatial displacement difference threshold of 0.0002m between consecutive iterations (less than 0.2% of the average bounding box axis length). The registration and alignment result, depicted in Figure 4 (bottom-right), achived an accuracy of 98.5% on the $x-$ and y-axis, and 97.5% on the z-axis in respect to the synthetic shift applied. Iterations 6 and 14 are also depicted in Figure 4 (top-right) and (bottom-left), repectivelly.

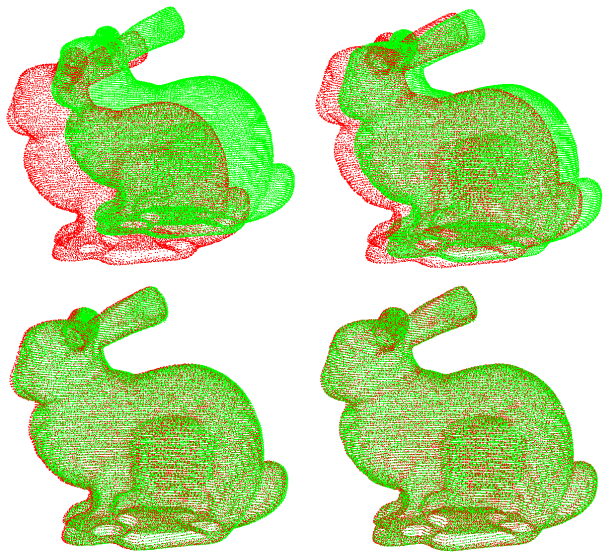

Figure 4: ICP process applied to the Stanford bunny point cloud with coarse subject (red) and dense reference (green)

In contrast to the bunny datasets, the two Welfenschloss datasets present a different origin, and thus a different point of view. Hence, there is no reference to which the registration and alignment result can be compared to. First, the two models are manually roughly registered to serve as start point, as depicted in Figure 5 (top). The registration process converged after 30 iterations, with a spatial displacement difference threshold of 0.01m between consecutive iterations (being less than 0.0001% of the average bounding box axis length). The registration and alignement result is depicted in Figure 5 (bottom).

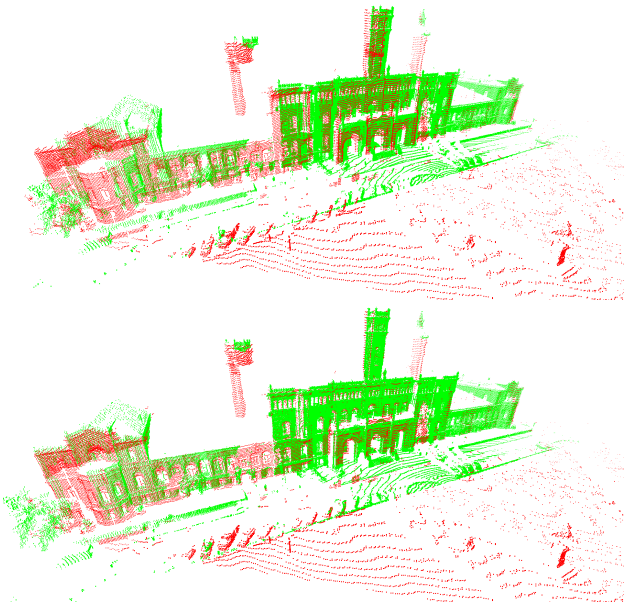

Figure 5: ICP process applied to the Welfeschloss point cloud with coarse subject from the mobile laser scanner (red), and dense reference from terrestrial laser scanner (green)

The statistics and figures clearly show that the proposed algorithm is robust and reliable. To present advantages of the proposed performance enhancements, namely Octree data structure, caching, and parallel processing, the ICP process was also implemented in its straight-forward form (referred to here as list-based, i.e., "classic" ICP implementation with point-to-plane strategy), with and without mesh caching, and with various number of processing threads. The duration of all processes are depicted in Figure 6 for the bunny model (left), and for the Welfenschloss model (right). The graphs show the runtime in seconds of all processes in the y-axis in respect to the number of used threads (SIMD level) in the x-axis (both in logarithmic scale). The list-based ICP processes with caching disabled (red) and caching enabled (green) are depicted in respect to the proposed methodology with caching disabled (blue) and caching enabled (pink), respectively. First, it is clear that caching accelerates running time for both processes, suggesting efficiency without harming the process reliability. Second, it is clear that our proposed methodology preforms best, suggesting an improvement of up to 65 times faster than the straight forward "classic" implementation (list-based, non-cached, single threaded), as in the Welfenschloss experiment, when optimizing the number of threads to 8 SIMD. It is worth noting that when the number of threads was increased to 16, all processes showed some decline in their running time, which is the result of an overhead with existing configuration of our working environment (providing 8 processing cores). A more powerful computing system (providing a larger number of processing cores) will suggest even better results. It should be noted that the list-based process with caching enabled did not always converge. The reason might be the insufficient caching strategy: since there is

no way to georeference the cached meshes a global cache has to be used, which might introduce a large number of false point-to-plane correspondences. In contrast, an Octree based data structure, as used in the proposed methodology, enables local (georeferenced) mesh caching. In terms of number of iterations and matching results all processes (except for the list-based cached) terminated with similar values.

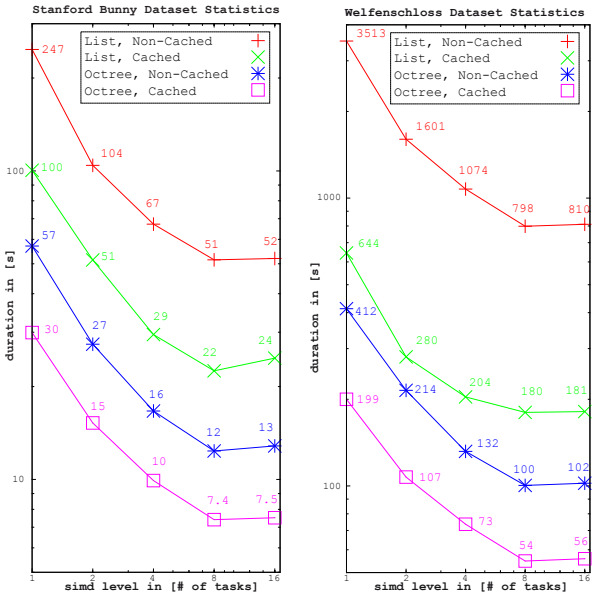

Figure 6: Duration statistics

5 CONCLUSIONS AND FUTURE WORK

This paper proposed an ICP methodology for the registration and alignment of 3D point clouds. The computational complexities introduced by the straight forward ICP are tackled by the application of specific performance enhancements, such as the use of an Octree data structure, caching strategies for the reuse of existing data, as well as the SIMD parallelization scheme. The overall results of the experiments showed a sharp decrease in running times - up to 65 times faster than the straight forward algorithm, while maintaining a reliable and robust solution. This methodology present a step forward towards an ICP process that is much more efficient with near real-time computing capabilities.

Our future work will involve the extension of the SIMD feature to make use of the GPGPU capabilities of modern graphic card - and not merely the use of multiple CPU cores. Furthermore, the caching strategy is planned to be fine tuned in order to increase the robustness of the solution and further improve the computation speed.

ACKNOWLEDGEMENTS

This joint research project was financially supported by the State of Lower-Saxony and the Volkswagen Foundation, Hannover, Germany.

REFERENCES

Akca, D. and Gruen, A., 2005. Recent advances in least squares 3d surface matching. In: Optical 3-D Measurement Techniques VII, pp. 197–206.

Besl, P. J. and McKay, N. D., 1992. A method for registration of 3-d shapes. IEEE Trans. Pattern Analysis and Machine Intelligence 14(2), pp. 239–256.

Dalley G., F. P., 2002. Pair-wise range image registration: A study in outlier classification. Computer Vision and Image Understanding 87(1–3), pp. 104–115.

Eggert, D. and Paelke, V., 2010. Relevance-driven acquisition and rapid on-site analysis of 3d geospatial data. In: Proceedings of the Joint International Conference on Theory, Data Handling and Modelling in GeoSpatial Information Science, Vol. 38number 2, pp. 118–123.

Fitzgibbon, A. W., 2003. Robust registration of 2d and 3d point sets. Image Vision Comput pp. 1145–1153.

Greenspan, M. and Yurick, M., 2003. Approximate k-d tree search for efficient icp. In: 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on, pp. 442–448.

Gruen, A., 1996. Least Squares Matching: a Fundamental Measurement Algorithm. Whittles, chapter Close Range Photogrammetry and Machine Vision, pp. 217–255.

Gruen, A. and Akca, D., 2004. Least squares 3d surface matching. In: IAPRS, 34(5/W16).

Guehring, J., 2001. Reliable 3d surface acquisition, registration and validation using statistical error models. In: 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on, pp. 224–231.

Horn, B. K. P., Hilden, H. and Negahdaripour, S., 1988. Closed-form solution of absolute orientation using orthonormal matrices. Journal of the Optical Society America 5(7), pp. 1127–1135.

Langis, C., Greenspan, M. and Godin, G., 2001. The parallel iterative closest point algorithm. In: 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on, pp. 195–202.

Mount, D. M., Netanyahu, N. S. and Moigne, J. L., 1999. Efficient algorithms for robust feature matching. Pattern Recognition pp. 17–38.

Plackett, R. L., 1950. Some theorems in least squares. Biometrika 37(1/2), pp. 149–157.

Qiu, D., May, S. and Nüchter, A., 2009. Gpu-accelerated nearest neighbor search for 3d registration. In: Proceedings of the 7th International Conference on Computer Vision Systems: Computer Vision Systems, ICVS '09, pp. 194–203.

Rusinkiewicz, S. and Levoy, M., 2001. Efficient variants of the icp algorithm. In: Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling, pp. 145–152.

Strand, M., Erb, F. and Dillmann, R., 2007. Range image registration using an octree based matching strategy. In: Mechatronics and Automation, 2007. ICMA 2007. International Conference on, pp. 1622–1627.

Szeliski, R., 1996. Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines. International Journal of Computer Vision 18, pp. 171–186.

Wang, J. and Shan, J., 2005. Lidar data management with 3-d hilbert space-filling curve. In: ASPRS Annual Meeting, Baltimore, March 7-11.

Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces.