FAST CORRESPONDENCE SEARCH FOR 3D SURFACE MATCHING

Devrim Akca *, Armin Gruen

Institute of Geodesy and Photogrammetry, Swiss Federal Institute of Technology (ETH) Zurich, CH-8093 Zurich, Switzerland - (agruen, akca)@geod.baug.ethz.ch

Commission V, WG V/3

KEY WORDS: Least squares 3D surface matching, Point clouds, Registration, Laser scanning, Correspondence searching

ABSTRACT:

An algorithm for least squares matching of overlapping 3D surfaces is presented. It estimates the transformation parameters between two or more fully 3D surfaces, using the Generalized Gauss-Markoff model, minimizing the sum of squares of the Euclidean distances between the surfaces. This formulation gives the opportunity of matching arbitrarily oriented 3D surfaces simultaneously, without using explicit tie points. Besides the mathematical model and execution aspects we pay particular interest to the reduction of the computational expenses. An efficient space partitioning method is implemented in order to speed up the correspondence search, which is the main portion of the computational efforts. The simultaneous matching of sub-surface patches is given as another strategy. It provides a computationally effective solution, since it matches only relevant multi-subpatches rather then the whole overlapping area. A practical example including computation times is given for the demonstration of the method.

1. INTRODUCTION

For 3D object modeling data acquisition must be performed from different standpoints. The derived local point clouds must be transformed into a common system. This procedure is usually referred to as registration. In the past, several efforts have been made concerning the registration of 3D point clouds. One of the most popular methods is the Iterative Closest Point (ICP) algorithm developed by Besl and McKay (1992), Chen and Medioni (1992), and Zhang (1994). The ICP is based on the search of nearest point-to-point or point-to-tangent plane pairs in the two sets, and estimating the rigid transformation, which aligns them. Then, the rigid transformation is applied to the points of one set, and the procedure is iterated until convergence. It does not use the local surface gradients in order to direct the solution to a minimum. Originally, it was not designed to register range data with scale factor or with higher order deformations. Several reviews and comparison studies about the ICP variant methods are available in the literature (Jokinen and Haggren, 1998; Campbell and Flynn, 2001; Rusinkiewicz and Levoy, 2001; Gruen and Akca, 2005).

The ICP, and in general all surface registration methods, requires heavily computation. Computational complexity of the original algorithm is of order O(n^2), which can take a lot of time when working with real-size data sets. Using the high performance computers or parallel computing systems (Langis et al., 2001) was proposed as a solution in order to reduce the processing time. However the main research emphasize has been given to the hardware-independent solutions.

The ICP algorithm always converges monotonically to a local minimum with respect to the mean-square distance objective function (Besl and McKay, 1992). This monotonic convergence behaviour leads slow convergence, which is typically 30-50 iterations (Besl and McKay, 1992; Zhang, 1994; Cunnington and Stoddart, 1999; Pottmann et al., 2004), even more in the extreme cases. Reducing the number of iterations is an option to accelerate the ICP. In their original publication Besl and McKay (1992) proposed an accelerated version of the ICP which updates the parameter vector using the linear or parabolic type of extrapolations. Pottmann et al. (2004) forced the parameter vector to a helical motion in the parameter space. Both of the methods change the convergence from monotonic to quadratic type. However manipulating the parameter vector without any statistical justification may cause two dangers: over-shooting the true solution, and deteriorating the orthogonality of the rotation matrix. The gradient descent types of algorithms assure substantially less number of iterations than the ICP variants (Szeliski and Lavallee, 1996; Neugebauer, 1997; Fitzgibbon, 2001). They adopted the Levenberg-Marquardt method for the estimation.

Another acceleration choice is to reduce the number of employed points. Hierarchical coarse to fine strategy is a popular approach (Zhang, 1994; Turk and Levoy, 1994; Neugebauer, 1997). They start the iteration using a lower resolution. While the algorithm approaches the solution, the resolution is hierarchically increased. Some authors used only a sub-sample of the data. The following sub-sampling strategies have been proposed: selection of points in smooth surface areas (Chen and Medioni, 1992), random sampling (Masuda and Yokoya, 1995), regular sampling (Guehring, 2001), selection of points with high intensity gradients (Weik, 1997), and selecting the points according to distribution of surface normals (Rusinkiewicz and Levoy, 2001). Godin et al. (2001) used a feature vector based random sampling. Distance minimization is performed only between pairs of points considered compatible on the basis of their viewpoint-invariant attributes, e.g. intensity, surface normal, curvature, etc. The hierarchical methods usually give satisfactory results. However sub-sampling based methods are very sensitive to data content, i.e. noise level, occlusion areas, complexity of the object, etc., and may not exploit the full accuracy potential of the registration.

* Corresponding author. http://www.photogrammetry.ethz.ch

The main computationally expensive part of the ICP is the exhaustive search for the correspondence. For a review of existing surface correspondence algorithms we refer to Planitz et al. (2005). Besl and McKay (1992) reported that 95% of the run-time is consumed for searching the correspondence. Speeding up the correspondence computation is another option in order to accelerate the ICP. Point-to-projection methods provide very fast solutions, since they reduce the problem to a 2D search when the sensor acquisition geometry or calibration parameters are known (Blais and Levine, 1995; Jokinen and Haggren, 1998). Recently Park and Subbarao (2003) gave a mixed method combining the accuracy advantage from the point-to-plane technique and speed advantage from the point-to-projection technique. In addition they gave an overview over the three mostly employed techniques, i.e. point-to-point, point-to-(tangent) plane, and point-to-projection. Projection to multi-z-buffers is another technique (Benjemaa and Schmitt, 1997). The multi-z-buffer technique provides a 3D space partitioning by segmenting the overlapping areas into z-buffer zones according to known depth direction. In general point-to-projection methods can solve the correspondence problem very quickly, but acquisition geometry and sensor calibration parameters must be known in advance. On the other hand, they only give approximations, and resulting registration is not as accurate as the point-to-point or point-to-plane methods.

Searching the correspondence is an algorithmic problem in fact, and can be substantially optimized by employing the special search structures. Search structures accelerate the registration by restricting the search space to a subpart of the data. The *k*-D tree (*k* dimensional binary search tree) was introduced by Bentley (1975), and is likely the most well-utilized nearest neighbor method (Zhang, 1994; Eggert et al., 1998; Greenspan and Yurick, 2003). The *k*-D tree is a binary search tree in which each node represents a partition of the *k*-dimensional space. The root node represents the entire space, and the leaf nodes represent subspaces containing mutually exclusive small subsets of the relevant point cloud. The space partitioning is carried out in a recursive binary fashion, i.e. letting at each step the direction of the cutting plane alternate between *yz*-, *xz*- and *xy*-plane. The average performance of the *k*-D tree search is of order (*n*log*n*), and the memory requirement is of order O(*n*). However constructing a *k*-D tree is quite complicated task and consumes a significant time of span, which is typical for all kind of tree-search algorithms. The Oct-tree, which is the 3D analogy of the quad-tree, was also used (Jackins and Tanimoto, 1980; Szeliski and Lavallee, 1996; Pulli et al., 1997). Brinkhoff (2004) investigated the usage of hash trees and R-trees, which have originally been developed for spatial database systems. Recently Wang and Shan (2005) applied the space partitioning technique to a relational database for effective management of LIDAR data. They ordered the 3D cells based on the principle of Hilbert space-filling curve, which provides fast access and spatial query mechanisms.

Pre-computed 3D distance map is another solution (Danielsson, 1980). Unfortunately, storing the complete uniform distance map at the desired accuracy can be expensive in term of memory requirement. Szeliski and Lavalle (1996) used an approximate but efficient pre-computed distance map named octree spline whose resolution increases hierarchically near the surface. Greenspan and Godin (2001) developed a nearest neighbor method, which calculates the spherical neighborhoods of each point in the preprocessing step, and tracks the evolution of point correspondence across iterations. Jost and Huegli (2003) combined a coarse to fine strategy with a fast closest point search by employing a nearest neighbor algorithm. They gave an extensive overview on the fast implementation of the ICP as well.

In our previous work an algorithm for least squares matching of overlapping 3D surfaces was given (Gruen and Akca, 2004; Gruen and Akca, 2005). It estimates the transformation parameters between two or more fully 3D surfaces, using the Generalized Gauss-Markoff model, minimizing the sum of squares of the Euclidean distances between the surfaces. This formulation gives the opportunity of matching arbitrarily oriented 3D surfaces simultaneously, without using explicit tie points. Our mathematical model is a generalization of the least squares image matching method, in particular the method given by Gruen (1985).

In this study we focus on the computational aspects in order to optimize the run-time. We implemented a rapid method for searching the correspondence, which is the main portion of the computational effort. We opt for a space partitioning method given by Chetverikov (1991) called boxing, since it is easy to implement and time-effective for constructing the box and accessing the data. In the original publication it was given for 2D point sets. We straightforwardly extend it to the 3D case. We combined our 3D boxing structure with a hierarchical local and adaptive nearest neighborhood search. Our second acceleration strategy is the simultaneous matching of sub-surface patches, which are selected in cooperative surface areas. It provides a computationally effective solution, since it matches only relevant multi-subpatches rather then the whole overlapping areas.

The details of the mathematical modeling of the proposed method and the execution aspects are explained in the following section. The two acceleration strategies are given in the third section. A practical example for the demonstration of the feasibility of the method is presented in the fourth section.

2. LEAST SQUARES 3D SURFACE MATCHING (LS3D)

Assume that two different partial surfaces of the same object are digitized/sampled point by point, at different times (temporally) or from different viewpoints (spatially). $f(x,y,z)$ and $g(x,y,z)$ are conjugate regions of the object in the *left* and *right* surfaces respectively. In other words $f(x,y,z)$ and $g(x,y,z)$ are discrete 3D representations of the *template* and *search* surfaces. The problem statement is estimating the final location, orientation and shape of the search surface $g(x,y,z)$, which satisfies minimum condition of the least squares matching with respect to the template $f(x,y,z)$. In an ideal situation one would have

$$f(x,y,z) = g(x,y,z) \tag{1}$$

Taking into account the noise and assuming that the template noise is independent of the search noise, Equation (1) becomes

$$f(x,y,z) - e(x,y,z) = g(x,y,z) \tag{2}$$

where $e(x,y,z)$ is a true error vector. Equation (2) are observation equations, which functionally relate the observations $f(x,y,z)$ to the parameters of $g(x,y,z)$. The matching is achieved by least squares minimization of a goal function, which represents the sum of squares of the Euclidean distances between the surfaces. The final location is estimated with respect to an initial position of $g(x,y,z)$, the approximation of the conjugate search surface $g^0(x,y,z)$.

To express the geometric relationship between the conjugate surface patches, a 7-parameter 3D similarity transformation is used:

$$[\,x \;\; y \;\; z\,]^{\mathrm{T}} = [\,t_x \;\; t_y \;\; t_z\,]^{\mathrm{T}} + m\mathbf{R}[\,x_0 \;\; y_0 \;\; z_0\,]^{\mathrm{T}} \tag{3}$$

where $\mathbf{R} = \mathbf{R}(\omega,\varphi,\kappa)$ is the orthogonal rotation matrix, $[t_x \; t_y \; t_z]^{\mathrm{T}}$ is the translation vector, and m is the uniform scale factor. This parameter space can be extended or reduced, as the situation demands it.

In order to perform least squares estimation, Equation (2) must be linearized by Taylor expansion.

$$\begin{aligned} f(x,y,z) - e(x,y,z) = g^0(x,y,z) &+ \frac{\partial g^0(x,y,z)}{\partial x}\mathrm{d}x \\ &+ \frac{\partial g^0(x,y,z)}{\partial y}\mathrm{d}y + \frac{\partial g^0(x,y,z)}{\partial z}\mathrm{d}z \end{aligned} \tag{4}$$

with

$$\mathrm{d}x = \frac{\partial x}{\partial p_i}\mathrm{d}p_i \;\;,\;\; \mathrm{d}y = \frac{\partial y}{\partial p_i}\mathrm{d}p_i \;\;,\;\; \mathrm{d}z = \frac{\partial z}{\partial p_i}\mathrm{d}p_i \tag{5}$$

where $p_i \in \{t_x,\; t_y,\; t_z,\; m,\; \omega,\; \varphi,\; \kappa\}$ is the i-th transformation parameter in Equation (3). Differentiation of Equation (3) gives:

$$\begin{aligned} \mathrm{d}x &= \mathrm{d}t_x + a_{10}\,\mathrm{d}m + a_{11}\,\mathrm{d}\omega + a_{12}\,\mathrm{d}\varphi + a_{13}\,\mathrm{d}\kappa \\ \mathrm{d}y &= \mathrm{d}t_y + a_{20}\,\mathrm{d}m + a_{21}\,\mathrm{d}\omega + a_{22}\,\mathrm{d}\varphi + a_{23}\,\mathrm{d}\kappa \\ \mathrm{d}z &= \mathrm{d}t_z + a_{30}\,\mathrm{d}m + a_{31}\,\mathrm{d}\omega + a_{32}\,\mathrm{d}\varphi + a_{33}\,\mathrm{d}\kappa \end{aligned} \tag{6}$$

where a_{ij} are the coefficient terms, whose expansions are trivial. Using the following notation

$$g_x = \frac{\partial g^0(x,y,z)}{\partial x},\;\; g_y = \frac{\partial g^0(x,y,z)}{\partial y},\;\; g_z = \frac{\partial g^0(x,y,z)}{\partial z} \tag{7}$$

and substituting Equations (6), Equation (4) results in the following:

$$\begin{aligned} -e(x,y,z) = {} & g_x\,\mathrm{d}t_x + g_y\,\mathrm{d}t_y + g_z\,\mathrm{d}t_z \\ & + (g_x a_{10} + g_y a_{20} + g_z a_{30})\mathrm{d}m \\ & + (g_x a_{11} + g_y a_{21} + g_z a_{31})\mathrm{d}\omega \\ & + (g_x a_{12} + g_y a_{22} + g_z a_{32})\mathrm{d}\varphi \\ & + (g_x a_{13} + g_y a_{23} + g_z a_{33})\mathrm{d}\kappa \\ & - (f(x,y,z) - g^0(x,y,z)) \end{aligned} \tag{8}$$

In the context of the Gauss-Markoff model, each observation is related to a linear combination of the parameters, which are variables of a deterministic unknown function. The terms $\{g_x,\; g_y,\; g_z\}$ are numeric first derivatives of this function $g(x,y,z)$. Equation (8) gives in matrix notation

$$-\boldsymbol{e} = \mathbf{A}\boldsymbol{x} - \boldsymbol{l} \;\;,\;\; \mathbf{P} \tag{9}$$

where $\mathbf{A}$ is the design matrix, $\boldsymbol{x}^{\mathrm{T}} = [\mathrm{d}t_x \;\, \mathrm{d}t_y \;\, \mathrm{d}t_z \;\, \mathrm{d}m \;\, \mathrm{d}\omega \;\, \mathrm{d}\varphi \;\, \mathrm{d}\kappa]$ is the parameter vector, and $\boldsymbol{l} = f(x,y,z) - g^0(x,y,z)$ is the discrepancy vector that consists of the Euclidean distances between the template and correspondent search surface elements. The template surface elements are approximated by the data points, on the other hand the search surface elements are represented in two different kind of piecewise surface forms (planar and bi-linear) optionally. In general both surfaces can be represented in any kind of piecewise form.

With the statistical expectation operator E{} and the assumptions $\mathrm{E}\{\boldsymbol{e}\} = 0$, $\mathrm{E}\{\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\} = \sigma_0^2\mathbf{P}_{ll}^{-1}$ Equation (9) is a Gauss-Markoff estimation model, where $\mathbf{P} = \mathbf{P}_{ll}$ is *a priori* weight matrix.

The unknown transformation parameters are treated as stochastic quantities using proper *a priori* weights. This extension gives advantages of control over the estimating parameters. We introduce the additional observation equations on the system parameters as

$$-\boldsymbol{e}_b = \mathbf{I}\boldsymbol{x} - \boldsymbol{l}_b \;\;,\;\; \mathbf{P}_b \tag{10}$$

where $\mathbf{I}$ is the identity matrix, $\boldsymbol{l}_b$ is the (fictitious) observation vector for the system parameters, and $\mathbf{P}_b$ is the associated weight coefficient matrix. The least squares solution of the joint system Equations (9) and (10) gives as the Generalized Gauss-Markoff model the unbiased minimum variance estimation for the parameters

$$\hat{\boldsymbol{x}} = (\mathbf{A}^{\mathrm{T}}\mathbf{P}\mathbf{A} + \mathbf{P}_b)^{-1}(\mathbf{A}^{\mathrm{T}}\mathbf{P}\boldsymbol{l} + \mathbf{P}_b\boldsymbol{l}_b) \qquad \text{(solution vector)} \tag{11}$$

$$\hat{\sigma}_0^2 = (\boldsymbol{v}^{\mathrm{T}}\mathbf{P}\boldsymbol{v} + \boldsymbol{v}_b^{\mathrm{T}}\mathbf{P}_b\boldsymbol{v}_b)/r \qquad \text{(variance factor)} \tag{12}$$

$$\boldsymbol{v} = \mathbf{A}\hat{\boldsymbol{x}} - \boldsymbol{l} \qquad \text{(residuals vector for surface observations)} \tag{13}$$

$$\boldsymbol{v}_b = \mathbf{I}\hat{\boldsymbol{x}} - \boldsymbol{l}_b \qquad \text{(residuals vector for parameter observations)} \tag{14}$$

where ^ stands for the Least Squares Estimator, and r is the redundancy. Since the functional model is non-linear, the solution is obtained iteratively. In the first iteration the initial approximations for the parameters must be provided. After the solution vector (Equation 11) is solved, the search surface $g^0(x, y, z)$ is transformed to a new state using the updated set of transformation parameters, and the design matrix $\mathbf{A}$ and the discrepancies vector $\boldsymbol{l}$ are re-evaluated. The iteration stops if each element of the alteration vector $\hat{\boldsymbol{x}}$ in Equation (11) falls below a certain limit: $|\mathrm{d}p_i| < c_i$.

The numerical derivative terms $\{g_x,\, g_y,\, g_z\}$ are defined as local surface normals $\boldsymbol{n}$. Their calculation depends on the analytical representation of the search surface elements. Two first degree C^0 continuous surface representations are implemented: triangle mesh form, which gives planar surface elements, and optionally grid mesh form, which gives bi-linear surface elements. The derivative terms are given as *x-y-z* components of the local normal vectors: $[g_x \; g_y \; g_z]^{\mathrm{T}} = \boldsymbol{n} = [n_x \; n_y \; n_z]^{\mathrm{T}}$.

The proposed method provides mechanisms for internal quality control by means of theoretical precision and correlation measures. Typical convergence rate for a good data configuration case is 5-6 iterations. A simple weighting scheme adapted from the Robust Estimation Methods is used to localize the occluded parts and the outliers. For the details of the method we refer to Gruen and Akca (2005).

3. ACCELERATION STRATEGIES

3.1 Fast correspondence search with boxing structure

The computational effort increases with the number of points in the matching process. The main portion of the computational complexity is to search the corresponding elements of the template surface patch on the search surface, whereas the parameter estimation part is a small system, and is quickly solved using Cholesky decomposition followed by back-substitution. Searching the correspondence is guided by an efficient boxing structure (Chetverikov 1991), which partitions the search space into boxes. For a given surface element, the correspondence is searched only in the box containing this element and in the adjacent boxes.

Let points $A_i = \{x_i, y_i, z_i\} \in$ S, $i = 0,1,\ldots,N$-1, represent the object $S \in \mathbb{R}^3$, and are kept in list L_1 in spatially non-ordered form. The boxing data structure (Fig. 1) consists of a rearranged point list L_2 and an index matrix $I = I_{u,v,w}$ whose elements are associated to individual boxes: $u,v,w = 0,1,\ldots,M$-1. The items of L_2 are coordinates of N points placed in the order of boxes. I contains integers indicating the beginnings of the boxes in L_2.

Initialization. Defining the box size.

Step 1. Recall min, max$\{x_i, y_i, z_i\}$ of data volume.

Step 2. Define number of boxes along *x-y-z* axes. For the sake of simplicity, they are given same (M) here.

Pass 1. Computing I.

Step 1. Allocate an MxMxM size accumulator array $B = B_{u,v,w}$ which is to contain the number of points in each box.

Step 2. Scan L_1 and fill B. For any point A_i the box indices are as follows:

$$u_i = \left\lfloor \frac{x_i - x_{\min}}{D_X} \right\rfloor, \quad v_i = \left\lfloor \frac{y_i - y_{\min}}{D_Y} \right\rfloor, \quad w_i = \left\lfloor \frac{z_i - z_{\min}}{D_Z} \right\rfloor \tag{15}$$

where $\lfloor\ \rfloor$ stands for the truncation operator, and D_X, D_Y and D_Z are dimensions of any box along the *x-y-z* axes respectively.

Step 3. Fill I using the following recursive formula:

$$I_{0,0,0} = 0.$$

For all $(u,v,w) \neq (0,0,0)$,

$$I_{u,v,w} = \begin{cases} I_{u,v,w-1} + B_{u,v,w-1} & \text{if} \quad w > 0 \\ I_{u,v-1,M-1} + B_{u,v-1,M-1} & \text{else if} \quad v > 0 \\ I_{u-1,M-1,M-1} + B_{u-1,M-1,M-1} & \text{else} \end{cases} \tag{16}$$

Pass 2. Filling L_2.

Step 1. For all u, v and w, set $B_{u,v,w} = 0$.

Step 2. Scan L_1 again. Use Equation (15), I and B to fill L_2. In L_2, the first point of the (u,v,w)-th box is indexed by I while the address of the subsequent points is controlled via B whose value is incremented each time a new point enters the box. Finally, release the memory area of B.

The memory requirement is of order O(N) for L_2 and O(M^3) for I. For the sake of clarity of the explanation, L_2 is given as a point list containing the *x-y-z* coordinate values. If one wants to keep the L_1 in the memory, then L_2 should only contain the access indices to L_1 or pointers, which directly point to the memory locations of the point coordinates.

Access procedure.

Step 1. Using Equation (15), compute the indices u_i, v_i and w_i of the box that contains point A_i.

Step 2. Use the boxing structure to retrieve the points bounded by the (u,v,w)-th box. In L_2, I indexes the first point, while the number of points in the box is given by the following formula:

$$\begin{cases} I_{u,v,w+1} - I_{u,v,w} & \text{if} \quad w < M-1 \\ I_{u,v+1,0} - I_{u,v,M-1} & \text{else if} \quad v < M-1 \\ I_{u+1,0,0} - I_{u,M-1,M-1} & \text{else if} \quad u < M-1 \\ N - I_{M-1,M-1,M-1} & \text{else} \end{cases} \tag{17}$$

The access procedure requires O(q) operations, where q is the average number of points in the box. One of the main advantages of the boxing structure is a faster and easier access mechanism than the tree search-based methods provide.

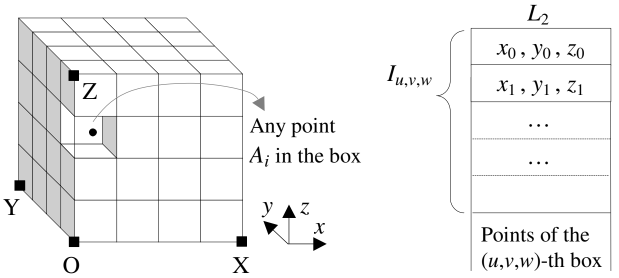

Figure 1. 3D Boxing. (Left) Boxing bounds all the data points, (Right) the boxing data structure.

The boxing structure, and in general all search structures, are designed for searching the nearest neighborhood in the static point clouds. In the LS3D surface matching case, the search surface, which the boxing structure is established for, is transformed to a new state by the current set of transformation parameters. Nevertheless there is no need neither to re-establish the boxing structure nor to update the I and L_2 in each iteration. Only positions of those four points (Fig. 1) are updated in the course of iterations: O$=\{x_{\min}, y_{\min}, z_{\min}\}$, X$=\{x_{\max}, y_{\min}, z_{\min}\}$, Y$=\{x_{\min}, y_{\max}, z_{\min}\}$, Z$=\{x_{\min}, y_{\min}, z_{\max}\}$. They uniquely define the boxing structure under the similarity transformation. The access procedure is the same, except the following formula is used for the indices calculation:

$$u_i = \left\lfloor \frac{OA_i \cdot OX}{\|OX\| D_X} \right\rfloor, \quad v_i = \left\lfloor \frac{OA_i \cdot OY}{\|OY\| D_Y} \right\rfloor, \quad w_i = \left\lfloor \frac{OA_i \cdot OZ}{\|OZ\| D_Z} \right\rfloor \tag{18}$$

where $\cdot$ stands for vector dot product. If the transformation is a similarity rather than a rigid body, the D_X, D_Y and D_Z values must also be updated in the iterations.

In our implementation, the correspondence is searched in the boxing structure during the first few iterations, and in the meantime its evolution is tracked across the iterations. Afterwards the searching process is carried out only in an adaptive local neighborhood according to the previous position and change of correspondence. In any step of the iteration, if the change of correspondence for a surface element exceeds a limit value, or oscillates, the search procedure for this element is returned to the boxing structure again.

3.2 Simultaneous multi-subpatch matching

The basic estimation model can be implemented in a multi-patch mode, that is the simultaneous matching of two or more search surfaces $g_i(x, y, z)$, i=1,…,k to one template $f(x, y, z)$.

$$-\boldsymbol{e}_i = \mathbf{A}_i \boldsymbol{x}_i - \boldsymbol{l}_i \quad , \quad \mathbf{P}_i \tag{19}$$

Since the parameter vectors $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_k$ do not have any joint components, the sub-systems of Equation (19) are orthogonal to each other. In the presence of auxiliary information those sets of equations could be connected via functional constraints, e.g. as in the Geometrically Constrained Multiphoto Matching (Gruen, 1985; Gruen and Baltsavias, 1988) or via appropriate formulation of multiple (>2) overlap conditions.

An ordinary point cloud includes enormously redundant information. A straightforward way to register such two point clouds could be matching of the whole overlapping areas. This is computationally expensive. We propose multi-subpatch mode as a further extension to the basic model, which is capable of simultaneous matching of sub-surface patches, which are interactively selected in cooperative surface areas. They are joined to the system by the same 3D transformation parameters. This leads to the observation equations

$$-\boldsymbol{e}_i = \mathbf{A}_i \boldsymbol{x} - \boldsymbol{l}_i \quad , \quad \mathbf{P}_i \qquad (20)$$

with $i = 1, \ldots, k$ subpatches. They can be combined as in Equation (9), since the common parameter vector $\boldsymbol{x}$ joints them to each other. The individual subpatches may not include sufficient information for the matching of whole surfaces, but together they provide a computationally effective solution, since they consist of only relevant information rather than using the full data set.

4. EXPERIMENTAL RESULTS

A practical example is given to show the capabilities of the method. The experiment was carried out using own self-developed C/C++ software that runs on an Intel® P4 2.53Ghz PC. The object is a chapel, which is located in Wangen, Germany. It is around 20x9 meters in size. The date set consists of 14 point clouds, which were acquired by the IMAGER 5003 (Z+F) terrestrial laser scanner. Each scan file contains 2.64 million points. The average point spacing is around 1-2 cm.

Fourteen consecutive matching processes were performed using the simultaneous multi-subpatch approach of the LS3D matching method. The results are given in Table 1. The initial approximations of the unknowns were provided by interactively selecting 3 common points on both surfaces prior to matching. The scale factor m was fixed to unity by infinite weight value $((\mathbf{P}_b)_{ii} \rightarrow \infty)$. The iteration criteria values c_i were selected as 0.1 mm for the translation vector and 10^{cc} for the rotation angles.

One of the scans was selected as the reference, which defines the datum of the common coordinate system. Since it is a closed object, there is need for a global registration, which distributes the residuals evenly among all of the scans, and also considers the closure condition, i.e. matching of the last scan to the first one. For this purpose we used the block adjustment by independent model solution, which was formerly proposed for global registration of laser scanner point clouds, but for the case of retro-reflective targets as tie points (Scaioni and Forlani, 2003). In the LS3D matching processes, the final correspondences were saved to separate files. Then all these files were given as input to a block adjustment by independent model procedure, which concluded with 1.6 mm *a posteriori* sigma value. Visual inspection of the final model showed the success of matching in all overlapping areas. The final model contains ca. 11.5 million triangles (Fig. 2).

A comparison against the non-accelerated versions is made for the matching experiments #4, #5, #6 and #7 (Table 2). The non-accelerated version exhaustively searches the correspondence in a large portion of the search surface during the first few iterations. In the following iterations it uses the same adaptive local neighborhood search as in the accelerated version. The results of both Tables 1 and 2 refer to the multi-subpatch approach. For a fair comparison approximately the same number of points were employed in the template and search surfaces. It is apparent that the accelerated version substantially decreases the processing times (here by factors 2 to 3). This is the sole effect of the space partitioning technique.

Table 1. Numerical results of "Chapel" example.

#	No. of template points	No. of search points	Iter.	No. of patches	Time (sec.)	Sigma naught (mm)
1	106,577	458,015	8	9	13.3	3.5
2	12,090	219,732	8	5	3.8	3.3
3	33,929	779,130	7	6	16.5	2.8
4	113,374	144,610	8	7	14.2	2.8
5	189,969	342,388	7	5	22.5	2.8
6	29,432	441,624	7	9	4.6	3.9
7	52,816	243,666	11	7	9.4	3.6
8	117,929	493,070	7	5	27.1	3.3
9	69,756	353,357	8	3	10.6	3.1
10	106,656	271,633	6	4	15.1	2.8
11	40,007	239,615	12	4	5.8	3.7
12	48,384	389,649	8	6	5.9	3.0
13	49,427	471,845	14	4	10.8	3.9
14	7,394	963,379	10	4	4.2	3.8

Table 2. Numerical results of non-accelerated version.

4	113,625	144,610	9	7	26.5	2.8
5	189,970	342,388	7	5	64.4	2.8
6	29,431	441,624	8	9	11.5	3.9
7	52,816	243,666	13	7	24.7	3.6

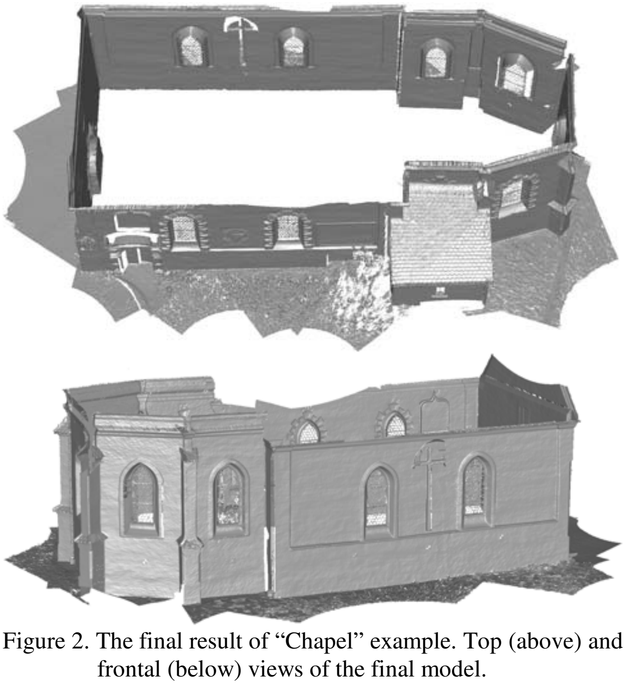

Figure 2. The final result of "Chapel" example. Top (above) and frontal (below) views of the final model.

5. CONCLUSIONS

An algorithm for the least squares matching of overlapping 3D surfaces is presented. Our proposed method, the Least Squares 3D Surface Matching (LS3D), estimates the transformation parameters between two or more fully 3D surfaces, using the Generalized Gauss Markoff model, minimizing the sum of squares of the Euclidean distances between the surfaces. The mathematical model is a generalization of the least squares image matching method and offers high flexibility for any kind of 3D surface correspondence problem. The least squares concept allows for the monitoring of the quality of the final results by means of precision and reliability criterions.

Two acceleration strategies are given as well. The multi-subpatch approach of the LS3D is combined with an efficient space partitioning technique. The practical example shows that our proposed method can provide successful matching results in reasonable processing times. The use of our space partitioning technique alone leads to a speed up of computing times by factor 2-3. Another aspect of our experiment is that registration of the point clouds of medium or large sized objects can be performed automatically without using retro-reflective or other special kinds of targets.

ACKNOWLEDGEMENTS

The data set “Chapel” is courtesy of Zoller+Fröhlich GmbH. The first author is financially supported by an ETHZ Internal Research Grant, which is gratefully acknowledged.

REFERENCES

Benjemaa, R., and Schmitt, F., 1997. Fast global registration of 3D sampled surfaces using a multi-z-buffer technique. *3DIM*, Ottawa, pp. 113-120.

Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. *Comm. of the ACM*, 18(9), 509-517.

Besl, P.J., and McKay, N.D., 1992. A method for registration of 3D shapes. *IEEE PAMI*, 14(2), 239-256.

Blais, G., Levine, M.D., 1995. Registering multiview range data to create 3D computer objects. *IEEE PAMI*, 17(8), 820-824.

Brinkhoff, T., 2004. Spatial access methods for organizing laserscanner data. *IAPRS*, 35(B4), pp. 98-102.

Campbell, R.J., and Flynn, P.J., 2001. A survey of free-form object representation and recognition techniques. *CVIU*, 81(2), 166-210.

Chen, Y., and Medioni, G., 1992. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3), 145-155.

Chetverikov, D., 1991. Fast neighborhood search in planar point sets. *Pattern Recognition Letters*, 12(7), 409-412.

Cunnington, S.J., and Stoddart, A.J., 1999. N-view point set registration: a comparison. *British Machine Vision Conference*, Nottingham, pp. 234-244.

Danielsson, P.-E, 1980. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14(3), 227-248.

Eggert, D.W., Fitzgibbon, A.W., and Fisher, R.B., 1998. Simultaneous registration of multiple range views for use in reverse engineering of CAD models. *CVIU*, 69(3), 253-272.

Fitzgibbon, A.W., 2001. Robust registration of 2D and 3D point sets. *British Machine Vision Conf.*, Manchester, pp. 411-420.

Guehring, J., 2001. Reliable 3D surface acquisition, registration and validation using statistical error models. *3DIM*, Quebec, pp. 224-231.

Godin, G., Laurendeau, D., and Bergevin, R., 2001. A method for the registration of attributed range images. *3DIM*, Quebec, pp. 179-186.

Greenspan, M., and Godin, G., 2001. A nearest neighbor method for efficient ICP. *3DIM*, Quebec, pp. 161-168.

Greenspan, M., and Yurick, M., 2003. Approximate K-D tree search for efficient ICP. *3DIM*, Banff, pp.442-448.

Gruen, A., 1985. Adaptive least squares correlation: a powerful image matching technique. *South African Journal of Photogrammetry, Remote Sensing and Cartography*, 14(3), 175-187.

Gruen, A., Baltsavias, E.P., 1988. Geometrically Constrained Multiphoto Matching. *PE & RS*, 54(5), 633-641.

Gruen, A., Akca, D., 2004. Least squares 3D surface matching. *IAPRS*, 34(5/W16), (on CD-ROM).

Gruen, A., Akca, D., 2005. Least squares 3D surface and curve matching. *ISPRS Journal of Photogrammetry & Remote Sensing*, 59(3), 151-174

Jackins, C.L., and Tanimoto, S.L., 1980. Oct-trees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14(3), 249-270.

Jokinen, O., and Haggren, H., 1998. Statistical analysis of two 3-D registration and modeling strategies. *ISPRS Journal of Photogrammetry & Remote Sensing*, 53(6), 320-341.

Jost, T., and Huegli, H., 2003. A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images. *3DIM*, Banff, pp. 427-433.

Langis, C., Greenspan, M., and Godin, G., 2001. The parallel iterative closest point algorithm. *3DIM*, Quebec, pp. 195-204.

Masuda, T., and Yokoya, N., 1995. A robust method for registration and segmentation of multiple range images. *CVIU*, 61(3), 295-307.

Neugebauer, P.J., 1997. Reconstruction of real-world objects via simultaneous registration and robust combination of multiple range images. *Int. J. of Shape Modeling*, 3(1-2), 71-90.

Park, S.-Y., and Subbarao, M., 2003. A fast point-to-tangent plane technique for multi-view registration. *3DIM*, Banff, pp. 276-283.

Planitz, B.M., Maeder, A.J., and Williams, J.A., 2005. The correspondence framework for 3D surface matching algorithms. *CVIU*, 97(3), 347-383.

Pottmann, H., Leopoldseder, S., and Hofer, M., 2004. Registration without ICP. *CVIU*, 95(1), 54-71.

Pulli, K., Duchamp, T., Hoppe, H., McDonald, J., Shapiro, L., and Stuetzle, W., 1997. Robust meshes from multiple range maps. *3DIM*, Ottawa, pp. 205-211.

Rusinkiewicz, S., and Levoy, M., 2001. Efficient variants of the ICP algorithm. *3DIM*, Quebec, pp. 145-152.

Scaioni, M., and Forlani, G., 2003. Independent model triangulation of terrestrial laser scanner data. *IAPRS*, 34(5/W12), 308-313.

Szeliski, R., and Lavallee, S., 1996. Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines. *IJCV*, 18(2), 171-186.

Turk, G., and Levoy, M., 1994. Zippered polygon meshes from range images. *Proc. SIGGRAPH’94*, Orlando, pp. 311-318.

Wang, J., and Shan, J., 2005. Lidar data management with 3-D Hilbert space-filling curve. *ASPRS Annual Meeting*, Baltimore.

Weik, S., 1997. Registration of 3D partial surface models using luminance and depth information. *3DIM*, Ottawa, pp. 93-100.

Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 13(2), 119-152.