

6D SLAM with Approximate Data Association

Andreas Nüchter, Kai Lingemann, Joachim Hertzberg
University of Osnabrück, Institute for Computer Science
Knowledge-Based Systems Research Group
Albrechtstraße 28
D-49069 Osnabrück, Germany
{nuechter|lingemann|hertzberg}@informatik.uni-osnabrueck.de

Hartmut Surmann
Fraunhofer Institute for
Autonomous Intelligent Systems (AIS)
Schloss Birlinghoven
D-53754 Sankt Augustin, Germany
hartmut.surmann@ais.fraunhofer.de

Abstract—This paper provides a new solution to the simultaneous localization and mapping (SLAM) problem with six degrees of freedom. A fast variant of the Iterative Closest Points (ICP) algorithm registers 3D scans taken by a mobile robot into a common coordinate system and thus provides relocation. Hereby, data association is reduced to the problem of searching for closest points. Approximation algorithms for this searching, namely, approximate k d-trees and box decomposition trees, are presented and evaluated in this paper. A solution to 6D SLAM that considers all free parameters in the robot pose is built based on 3D scan matching.

I. INTRODUCTION

Digital 3D models of the environment are needed in rescue and inspection robotics, facility management and architecture. The problem of automatic environment sensing and modeling is complex, because a number of fundamental scientific issues are involved. This paper focusses on how to create a consistent 3D scene into a common coordinate system from multiple scans. The proposed algorithms allow to digitize large environments fast and reliably without any intervention and to solve the simultaneous localization and mapping (SLAM) problem. Finally, robot motion on natural outdoor surfaces has to cope with changes in yaw, pitch and roll angles, turning pose estimation as well as scan matching or registration into a problem in six mathematical dimensions. This paper presents a new solution to the SLAM problem with six degrees of freedom (6D SLAM). A fast variant of the iterative closest points (ICP) algorithm registers the 3D scans into a common coordinate system and relocates the robot. Computation time is reduced by two new methods: First, we reduce the 3D data, i.e., we compute point clouds that approximate the scanned 3D surface and contain only a small fraction of that original 3D point cloud. Second, we present a fast approximation of the corresponding point for the ICP algorithm. Several approximation methods are evaluated in this paper. These extensions of ICP result in a fast and robust algorithm for generating overall consistent 3D maps, using global error minimization.

In previous work we developed the 6D SLAM algorithm [20], [27]. This paper's main contribution is to evaluate the approximate data association to speed up the algorithm. The rest of the paper is organized as follows: Section II discusses the state of the art in 3D mapping. Then we present the used 3D laser scanner and the mobile robot. Section IV describes scan matching and pose estimation, followed by the application of closest point approximation in the data

association phase. Section VI discusses the results. Section VII concludes.

II. 3D MAPPING – STATE OF THE ART

Instead of using 3D scanners, which yield consistent 3D scans in the first place, some groups have attempted to build 3D volumetric representations of environments with 2D laser range finders. Thrun et al. [15], [28], Früh et al. [12] and Zhao et al. [30] use two 2D laser range finders for acquiring 3D data. One laser scanner is mounted horizontally, the other vertically. The latter one grabs a vertical scan line, which is transformed into 3D points based on the current robot pose. Since the vertical scanner is not able to scan sides of objects, Zhao et al. use two additional, vertically mounted 2D scanners, shifted by 45° to reduce occlusions [30]. The horizontal scanner is used to compute the robot pose. The precision of 3D data points depends on that pose and on the precision of the scanner.

A few other groups use highly accurate, expensive 3D laser scanners [23], [1], [13]. The RESOLV project aimed at modeling interiors for virtual reality and tele-presence [23]. They used a RIEGL laser range finder on robots and the ICP algorithm for scan matching [7], [9]. The AVENUE project develops a robot for modeling urban environments [1], using a CYRAX laser scanner and a feature-based scan matching approach for registering of the 3D scans in a common coordinate system [25]. Nevertheless, in their recent work they do not use data of the laser scanner in the robot control architecture for localization [13]. The research group of M. Hebert has reconstructed environments using the Zoller+Fröhlich laser scanner and aims at building 3D models without initial position estimates, i.e., without odometry information [16].

Recently, different groups employ rotating SICK scanners for acquiring 3D data. Wulf et al. let the scanner rotate around the vertical axis. They acquire 3D data while moving, thus the quality of the resulting map crucially depends on the pose estimate that is given by inertial sensors, i.e., gyros [29]. In addition, their SLAM algorithms do not consider all six degrees of freedom. Nevado et al. present novel algorithms for post processing 3D scans/scenes and extracting planar models [18].

Other approaches use information of CCD-cameras that provide a view of the robot's environment [22], [8]. Yet, cameras are difficult to use in natural environments with changing light conditions. Camera-based approaches to 3D robot vision, e.g., stereo cameras and structure from motion,



Fig. 1. Left: The 3D laser range finder. Its technical basis is a SICK 2D laser range finder (LMS-200). Right: The autonomous mobile robot Kurt3D.

have difficulties providing reliable navigation and mapping information for a mobile robot in real-time. Thus some groups try to solve 3D modeling by using a planar SLAM methods and cameras, e.g., in [8].

III. AUTOMATIC 3D SENSING

A. The 3D laser range finder

The AIS 3D laser range finder (Fig. 1) [26] is built on the basis of a 2D range finder by extension with a mount and a small servomotor. The 2D laser range finder is attached in the center of rotation to the mount for achieving a controlled pitch motion. A standard servo is connected on the left side (Fig. 1) and is controlled by a computer running Linux. The 3D laser scanner operates up to 5h (Scanner: 17 W, 20 NiMH cells with a capacity of 4500 mAh, Servo: 0.85 W, 6 V with batteries of 4500 mAh) per battery pack.

The area of $180^\circ(\text{h}) \times 120^\circ(\text{v})$ is scanned with different horizontal (181, 361, 721) and vertical (128, 256, 400, 500) resolutions. A plane with 181 data points is scanned in 13 ms by the 2D laser range finder (rotating mirror device). Planes with more data points, e.g., 361, 721, duplicate or quadruplicate this time. Thus a scan with 181×256 data points needs 3.4 seconds. In addition to the distance measurement the 3D laser range finder is capable of quantifying the amount of light returning to the scanner. Scanning the environment with a mobile robot is done in a stop-scan-go fashion.

B. The mobile robot

Kurt3D (Fig. 1) is based on a mobile robot platform with a size of 45 cm (length) \times 33 cm (width) \times 26 cm (height) and a weight of 15.6 kg. Equipped with the 3D laser range finder the height increases to 47 cm and weight increases to 22.6 kg. Kurt3D's maximum velocity is 5.2 m/s (autonomously controlled 4.0 m/s). Two 90W motors are used to power the 6 wheels, whereas the front and rear wheels have no tread pattern to enhance rotating. Kurt3D operates for about 4 hours with one battery (28 NiMH cells, capacity: 4500 mAh) charge. The core of the robot is a Pentium-III-600 MHz with 384 MB RAM and real-time Linux. An embedded 16-Bit CMOS microcontroller is used to control the motor.¹

¹Videos of an exploration with the autonomous mobile robot can be found at: <http://www.ais.fraunhofer.de/ARC/kurt3D/index.html>

IV. RANGE IMAGE REGISTRATION AND ROBOT RELOCALIZATION

Multiple 3D scans are necessary to digitalize environments without occlusions. To create a correct and consistent model, the scans have to be merged into one coordinate system. This process is called registration. If robot carrying the 3D scanner were precisely localized, the registration could be done directly based on the robot pose. However, due to the unprecise robot sensors, self localization is erroneous, so the geometric structure of overlapping 3D scans has to be considered for registration.

The following method for registration of point sets is part of many publications, so only a short summary is given here. The complete algorithm was invented in 1992 and can be found, e.g., in [7]. The method is called *Iterative Closest Points (ICP) algorithm*.

Given two independently acquired sets of 3D points, M (model set, $|M| = N_m$) and D (data set, $|D| = N_d$) which correspond to a single shape, we aim to find the transformation consisting of a rotation \mathbf{R} and a translation \mathbf{t} which minimizes the following cost function:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\|^2. \quad (1)$$

$w_{i,j}$ is assigned 1 if the i -th point of M describes the same point in space as the j -th point of D . Otherwise $w_{i,j}$ is 0. Two things have to be calculated: First, the corresponding points, and second, the transformation (\mathbf{R}, \mathbf{t}) that minimize $E(\mathbf{R}, \mathbf{t})$ on the base of the corresponding points.

The ICP algorithm calculates iteratively the point correspondences. In each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation (\mathbf{R}, \mathbf{t}) for minimizing equation (1). The assumption is that in the last iteration step the point correspondences are correct. Besl et al. prove that the method terminates in a minimum [7]. However, this theorem does not hold in our case, since we use a maximum tolerable distance d_{\max} for associating the scan data. Such a threshold is required, given that the 3D scans overlap only partially. Fig. 2 (top) shows three frames, i.e., iteration steps, of the ICP algorithm. The bottom part shows the start poses (x, z, θ_y) from which a correct matching is possible, here with only three degrees of freedom.

A. Calculation of the rotation and translation

In every iteration the optimal transformation (\mathbf{R}, \mathbf{t}) has to be computed. Eq. (1) can be reduced to

$$E(\mathbf{R}, \mathbf{t}) \propto \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_i + \mathbf{t})\|^2, \quad (2)$$

with $N = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j}$, since the correspondence matrix can be represented by a vector containing the point pairs.

Four methods are known to minimize eq. (2) [17]. In earlier work [20], [27] we used a quaternion based method [7], but the following one, based on singular value decomposition (SVD), is robust and easy to implement, thus we give a brief overview

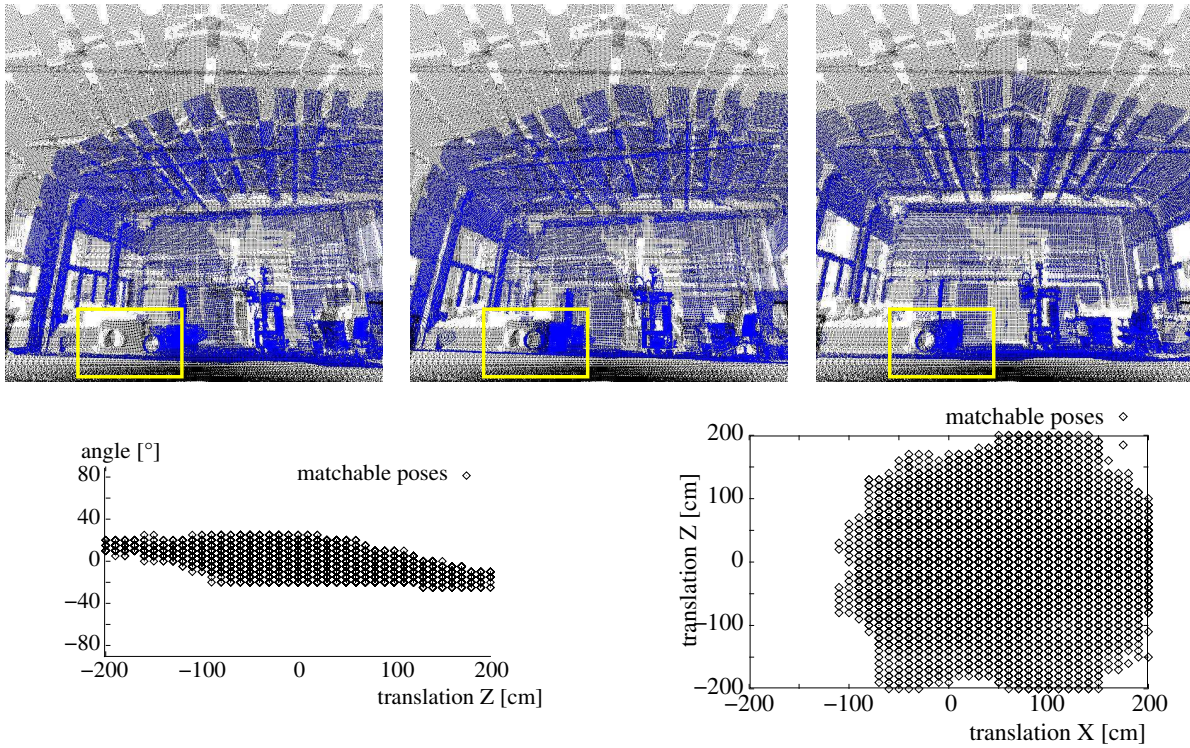


Fig. 2. Top row: Left: Initial odometry based pose of two 3D scans. Middle: Pose after five ICP iterations. Right: final alignment, pairwise matching. Bottom row: The poses are marked in (x, z, θ_y) from which a correct alignment of two 3D scans is possible.

of the SVD-based algorithms. It was first published by Arun, Huang and Blostein [2]. The difficulty of this minimization problem is to enforce the orthonormality of matrix \mathbf{R} . The first step of the computation is to decouple the calculation of the rotation \mathbf{R} from the translation \mathbf{t} using the centroids of the points belonging to the matching, i.e.,

$$\mathbf{c}_m = \frac{1}{N} \sum_{i=1}^N \mathbf{m}_i, \quad \mathbf{c}_d = \frac{1}{N} \sum_{i=1}^N \mathbf{d}_i \quad (3)$$

and

$$\mathbf{M}' = \{\mathbf{m}'_i = \mathbf{m}_i - \mathbf{c}_m\}_{1, \dots, N}, \quad (4)$$

$$\mathbf{D}' = \{\mathbf{d}'_i = \mathbf{d}_i - \mathbf{c}_d\}_{1, \dots, N}. \quad (5)$$

After replacing (3), (4) and (5) in the error function, $E(\mathbf{R}, \mathbf{t})$ eq. (2) becomes:

$$\begin{aligned} E(\mathbf{R}, \mathbf{t}) &\propto \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i - \underbrace{(\mathbf{t} - \mathbf{c}_m + \mathbf{R}\mathbf{c}_d)}_{=\tilde{\mathbf{t}}}\|^2 \\ &= \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\|^2 \end{aligned} \quad (6a)$$

$$- \frac{2}{N} \tilde{\mathbf{t}} \cdot \sum_{i=1}^N (\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i) \quad (6b)$$

$$+ \frac{1}{N} \sum_{i=1}^N \|\tilde{\mathbf{t}}\|^2. \quad (6c)$$

In order to minimize the sum above, all terms have to be minimized. The second sum (6b) is zero, since all values refer

to centroid. The third part (6c) has its minimum for $\tilde{\mathbf{t}} = \mathbf{0}$ or

$$\mathbf{t} = \mathbf{c}_m - \mathbf{R}\mathbf{c}_d. \quad (7)$$

Therefore the algorithm has to minimize only the first term, and the error function is expressed in terms of the rotation only:

$$E(\mathbf{R}, \mathbf{t}) \propto \sum_{i=1}^N \|\mathbf{m}'_i - \mathbf{R}\mathbf{d}'_i\|^2. \quad (8)$$

Theorem: The optimal rotation is calculated by $\mathbf{R} = \mathbf{V}\mathbf{U}^T$. Herby the matrices \mathbf{V} and \mathbf{U} are derived by the singular value decomposition $\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$ of a correlation matrix \mathbf{H} . This 3×3 matrix \mathbf{H} is given by

$$\mathbf{H} = \sum_{i=1}^N \mathbf{m}'_i{}^T \mathbf{d}'_i = \begin{pmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{pmatrix}, \quad (9)$$

with $S_{xx} = \sum_{i=1}^N m'_{ix} d'_{ix}$, $S_{xy} = \sum_{i=1}^N m'_{ix} d'_{iy}$, \dots . The analogous algorithm is derived directly from this theorem.

Proof: Since rotation is length preserving, i.e., $\|\mathbf{R}\mathbf{d}'_i\|^2 = \|\mathbf{d}'_i\|^2$ the error function (8) is expanded

$$E(\mathbf{R}, \mathbf{t}) \propto \sum_{i=1}^N \|\mathbf{m}'_i\|^2 - 2 \sum_{i=1}^N \mathbf{m}'_i \cdot \mathbf{R}\mathbf{d}'_i + \sum_{i=1}^N \|\mathbf{d}'_i\|^2.$$

The rotation affects only the middle term, thus it is sufficient to maximize

$$\sum_{i=1}^N \mathbf{m}'_i \cdot \mathbf{R}\mathbf{d}'_i = \sum_{i=1}^N \mathbf{m}'_i{}^T \mathbf{R}\mathbf{d}'_i. \quad (10)$$

Using the trace of a matrix, (10) can be rewritten to obtain

$$\text{trace} \left(\sum_{i=1}^N \mathbf{R} \mathbf{d}'_i \mathbf{m}'_i{}^T \right) = \text{trace} (\mathbf{R} \mathbf{H}),$$

With \mathbf{H} defined as in (9). Now we have to find the matrix \mathbf{R} that maximizes trace $(\mathbf{R} \mathbf{H})$.

Assume that the singular value decomposition of \mathbf{H} is

$$\mathbf{H} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T,$$

with \mathbf{U} and \mathbf{V} orthonormal 3×3 matrices and $\mathbf{\Lambda}$ a 3×3 diagonal matrix without negative elements. Suppose

$$\mathbf{R} = \mathbf{V} \mathbf{U}^T.$$

\mathbf{R} is orthonormal and

$$\begin{aligned} \mathbf{R} \mathbf{H} &= \mathbf{V} \mathbf{U}^T \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T \\ &= \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T \end{aligned}$$

is a symmetric, positive definite matrix. Arun, Huang and Blostein provide a lemma to show that

$$\text{trace} (\mathbf{R} \mathbf{H}) \geq \text{trace} (\mathbf{B} \mathbf{R} \mathbf{H})$$

for any orthonormal matrix \mathbf{B} . Therefore the matrix \mathbf{R} is optimal. Proving the lemma is straightforward using the Cauchy-Schwarz [2]. Finally, the optimal translation is calculated as (cf. eq. (6c) and (7))

$$\mathbf{t} = \mathbf{c}_m - \mathbf{R} \mathbf{c}_d.$$

B. ICP-based 6D SLAM

To digitalize environments, multiple 3D scans have to be registered. After registration, the scene has to be globally consistent. A straightforward method for aligning several 3D scans is *pairwise matching*, i.e., the new scan is registered against the scan with the largest overlapping areas. The latter one is determined in a preprocessing step. Alternatively, in [9] an *incremental matching* method is introduced, i.e., the new scan is registered against a so-called *metascan*, which is the union of the previously acquired and registered scans. Each scan matching is limited in precision. Both methods accumulate the registration errors such that the registration of a large number of 3D scans leads to inconsistent scenes and to problems with the robot localization.

1) *Closing the loop*: After matching multiple 3D scans, errors have accumulated and a closed loop will be inconsistent. Our algorithm detects a closing loop by registering the last acquired 3D scan with earlier acquired scans, e.g., the first scan. If a registration is possible, the computed error is distributed over all 3D scans. A second step minimizes the global error with the following algorithm.

2) *Diffusing the Error*: Pulli presents a registration method that minimizes the global error and avoids inconsistent scenes [21]. The registration of one scan is followed by registration of all neighboring scans, such that the global error is distributed. Other matching approaches with global error minimization have been published, e.g., [5] and [10]. Benjema et al.

establish point-to-point correspondences first and then use randomized iterative registration on a set of surfaces [5]. Eggert et al. compute motion updates, i.e., a transformation (\mathbf{R}, \mathbf{t}) , using force-based optimization, with data sets considered as connected by groups of springs [10].

Based on the idea of Pulli we designed a relaxation method called *simultaneous matching*. Thereby, the first scan is the masterscan and determines the coordinate system. This scan is fixed. The following three steps register all scans and minimize the global error, after a queue is initialized with the first scan of the closed loop:

- 1) The current scan is the first scan of the queue. This 3D scan is removed from the queue.
- 2) If the current scan is not the master scan, then a set of neighbors (set of all scans that overlap with the current scan) is calculated. This set of neighbors forms one point set M . The current scan forms the data point set D and is aligned with the ICP algorithms. One scan overlaps with another iff more than 250 corresponding point pairs exist.
- 3) If the current scan changes its location by applying the transformation (translation or rotation), then each single scan of the set of neighbors that is not in the queue is added to the end of the queue. If the queue is empty, terminate; else continue at (1).

In contrast to Pulli's approach, our method is totally automatic and no interactive pairwise alignment has to be done. Furthermore the point pairs are not fixed [21]. The accumulated alignment error is spread over the whole set of acquired 3D scans. This diffuses the alignment error equally over the set of 3D scans [27].

C. Computing point correspondences

The time complexity of the algorithm described above is dominated by the time for determining the closest points (brute force search $\mathcal{O}(n^2)$ for 3D scans of n points). Several enhancements have been proposed [6], [7], [24]. We have implemented *kd-trees* as proposed by Simon et al. Fig. 3 shows two slices taken from a *kd-tree*.

1) *kd-trees*: *kd-trees* are a generalization of binary search trees. Every node represents a partition of a point set to the two successor nodes. The root represents the whole point cloud and the leafs are a disjunct partition of the set. These leafs are called buckets (cf. Fig. 4). Furthermore, every node contains the limits of the represented point set. An efficient implementation of a *kd-tree* is given in [19].

Searching in *kd-trees* is done recursively. A given 3D point \mathbf{p}_q needs to be compared with the separating plane in order to decide on which side the search must continue. This procedure is executed until the leafs are reached. There, the algorithm has to evaluate all bucket points. However, the closest point may be in a different bucket, iff the distance to the limits is smaller than the one to the closest point in the bucket. In this case backtracking has to be performed. Fig. 4 shows a backtracking case, where the algorithms has to go back to the root. The test is known as Ball-Within-Bounds test [6], [11], [14].

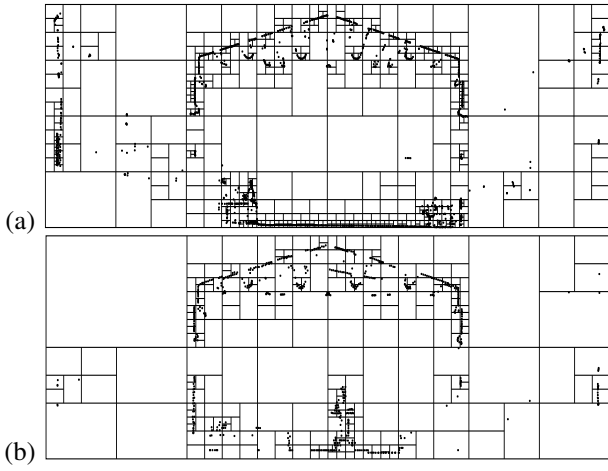


Fig. 3. A kd -tree of scanned 3D data ($k = 3$) of Fig. 2 (top) first/black 3D scan. Two (x, y) -projections of slices at depths $z = 100$ cm (a) and $z = 550$ cm (b) are given.

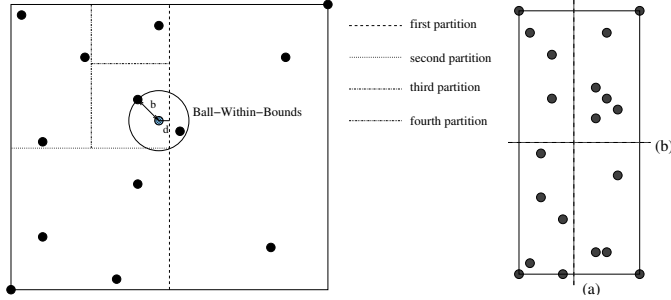


Fig. 4. Left: Construction of a kd tree. Right: The optimized kd -tree uses splits along the longest axis to ensure compact volumes.

2) *The optimized kd-tree*: The objective of optimizing kd -trees is to reduce the expected number of visited leaves. Three parameters are adjustable, namely, the direction and place of the split axis as well as the number of points in the buckets. Splitting the point set at the *median* ensures that every kd -tree entry has the same probability [11]. The median can be found in linear time, thus the time complexity for constructing the tree is not affected. Furthermore, the split axis should be oriented *perpendicular* to the longest axis to minimize the amount of backtracking (see Fig. 4). Friedman and colleagues prove that a bucket size of 1 is optimal [11]. Nevertheless, in practice it turned out that a slightly larger bucket size is faster as given in Fig. 5.

3) *Point reduction*: To gain an additional speedup, we have proposed a point reduction. During scanning surfaces close to the scanner are sampled with more data points. These areas are subsampled using a median and reduction filter. Fig. 6 shows the result, details of the algorithm can be found in [20].

V. APPROXIMATE DATA ASSOCIATION

A. Approximate kd-trees

Since the ICP algorithm, and therefore our 6D SLAM method, extensively computes nearest neighbours, approximating the nearest neighbours will speed up the algorithm. S. Arya and D. Mount introduce the following notion for approximating the nearest neighbor [3]: Given an $\varepsilon > 0$, then the point

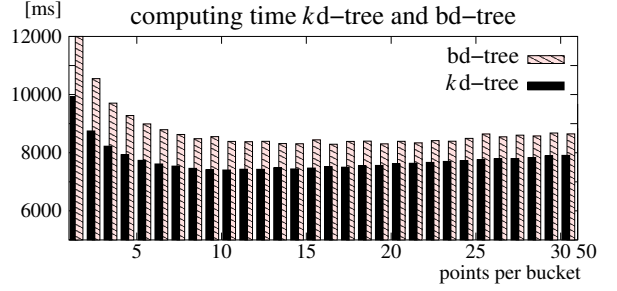


Fig. 5. Computing time in milliseconds for a 3D scan matching dependent on the bucket size of a kd - and bd -tree.

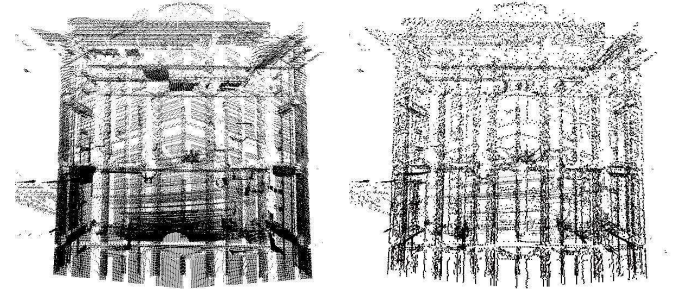
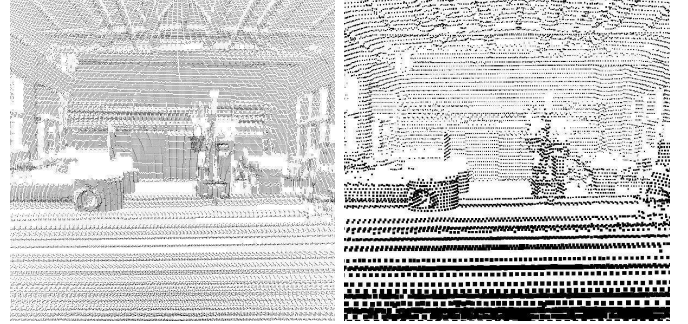


Fig. 6. Left: Two views of a 3D scene. Right: Subsampled version (points have been enlarged). Two different views are presented (bottom view from top).

$\mathbf{p} \in D$ is the $(1 + \varepsilon)$ -approximate nearest neighbour of the point \mathbf{p}_q , iff

$$\|\mathbf{p} - \mathbf{q}\| \leq (1 + \varepsilon) \|\mathbf{p}^* - \mathbf{q}\|,$$

where \mathbf{p}^* denotes the true nearest neighbour, i.e., \mathbf{p} has a maximal distance of ε to the true nearest neighbour. Using this notation in every step the algorithm records the closest point \mathbf{p} . The search terminates if the distance to the unanalyzed leaves is larger than

$$\|\mathbf{p}_q - \mathbf{p}\| / (1 + \varepsilon).$$

Fig. 7 (left) shows an example where the gray cell needs not to be analyzed, since the point \mathbf{p} satisfies the approximation criterion.

B. Approximate box decomposition trees

Arya et al. [4] have presented an optimal algorithm for approximate nearest neighbor search. They use a balanced box decomposition tree (bd -tree) as their primary data structure. This tree combines two important properties of geometric data structures: First, as in the kd -tree case, the set of points is

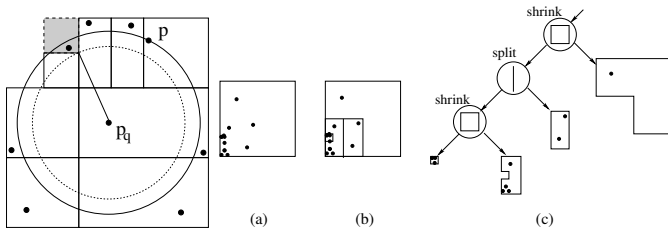


Fig. 7. Left: The $(1 + \varepsilon)$ -approximate nearest neighbor. The solid circle denotes the ε environment of \mathbf{p}_g . The search algorithm need not analyze the gray cell, since \mathbf{p} satisfies the approximation criterion. Middle and right: (a) Given point set. (b) decomposition into buckets. (c) Tree layout. Fig. adapted from [3], [4].

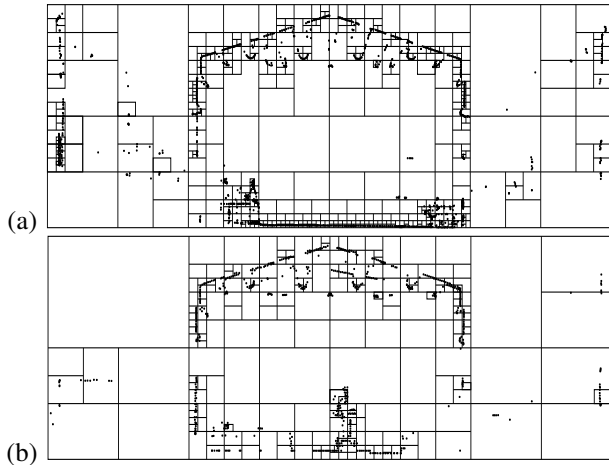


Fig. 8. A bd-tree of scanned 3D data ($k = 3$) of Fig. 2 (top) first/black 3D scan. Two (x, y) -projections of slices at depths $z = 100$ cm (a) and $z = 550$ cm are given (b).

exponentially reduced. Second, the aspect ratio of the tree edges are bounded by a constant. Not even the optimized k d-tree is able to make this assurance, but quadtrees show this characteristic [4]. The actual box decomposition search tree is composed of splits and shrinks. Fig. 7 (c) shows the general structure and Fig. 8 presents two slices within this search tree.

The search procedure of bd-trees is similar to the one of k d-trees. The approximate search is discontinued (cf. Fig. 7) if the distance to the unanalyzed leaves is larger than

$$\|\mathbf{p}_q - \mathbf{p}\| / (1 + \varepsilon).$$

VI. RESULTS

This section focuses on three aspects. Firstly, we evaluate the quality of scan matching with approximate nearest neighbour search. Secondly, we investigate the performance of approximate k d-trees and approximate bd-trees. Finally, we reproduce results from a robot run given in [27] to demonstrate the general performance of the approach.

To evaluate the quality of the scan matching we restrict the problem to three degrees of freedom. We acquired two 3D scans and measured the pose shift by a reference system, i.e., a meter rule. Fig. 2 (bottom) shows the starting poses from which a correct scan matching is possible. Fig. 10 indicates the initial positions that result in a correct scan matching for different values of ε and the bucket size b . Comparing the figures, we conclude that the approximation

TABLE I

COMPUTING TIME AND NUMBER OF ICP ITERATIONS TO ALIGN ALL 32 3D SCANS (PENTIUM-IV-2400). IN ADDITION THE COMPUTING TIME FOR THE SLAM ALGORITHM (CLOSED LOOP DETECTION AND SIMULTANEOUS MATCHING) IS GIVEN. ABOUT 1 MIN PER 3D SCAN IS NEEDED TO DO ERROR DIFFUSION IN 6D [27].

points & search method	time	iter.
all pts. & brute force	144 h 5 min	2080
all pts. & k D-tree	12 min 23 s	2080
all pts. & Apx- k D-tree	10 min 1 s	2080
red. pts. & Apx- k D-tree	1 min 32 s	2176
6D SLAM with reduced pts. & Apx- k D-tree	38 min	16000

does not significantly influence the scan matching, due to the large number of used points and to the iterative nature of the algorithm.

The performance of the proposed tree search is given in Fig. 5 and 9. In case of no approximation (Fig. 5) the k d-tree outperforms the bd-tree. The optimal time is reached with 10 points per bucket. In case of approximation, only in a few cases, i.e., 18 out of 124 experiments, the bd-tree is faster than the k d-tree. Nevertheless, one notices that with increasing ε the computation time for the scan matching is reduced drastically (up to a factor of 2).

The proposed algorithms have been applied to a data set acquired on the Fraunhofer Campus Birlinghoven campus. 32 3D scans, each containing 302820 (721×420) range data points, were taken by the mobile robot Kurt3D. The robot had to cope with a height difference between the two buildings of 1.05 meter, covered, in the first case, by a sloped driveway in open outdoor terrain, and, in the second case, by a ramp of 12° inside the building. The 3D model was computed after acquiring all 3D scans. Table I summarizes the computation time of our 6D SLAM algorithms. Refer to the website <http://www.ais.fraunhofer.de/ARC/3D/6D/> for a computed animation and video through the scanned 3D scene. Furthermore, the algorithms have been evaluated at the RoboCup Rescue 2004 competition in Lisbon and precise, reliable, on time 3D maps have been generated (see <http://www.ais.fhg.de/ARC/kurt3D/rr.html>).

VII. CONCLUSION

This paper has presented a new solution to the simultaneous localization and mapping (SLAM) problem with six degrees of freedom. The method is based on the ICP scan matching algorithm. The paper investigates approximate data association using k d-trees and bd-trees. k d-trees empirically outperform bd-trees with and without approximation. Approximation does not significantly deteriorate the quality of scan registration. From that we know of no other superior approach, scan matching based on reduced point sets and approximate k d-trees is currently the most performant method at hand for scan matching in 6D SLAM.

ACKNOWLEDGEMENTS

The work was done during the authors' time at the Fraunhofer Institute for Autonomous intelligent Systems. We would like to thank Kai Pervözl and Matthias Hennig and the whole Kurt3D RoboCup Rescue team of Fraunhofer AIS.

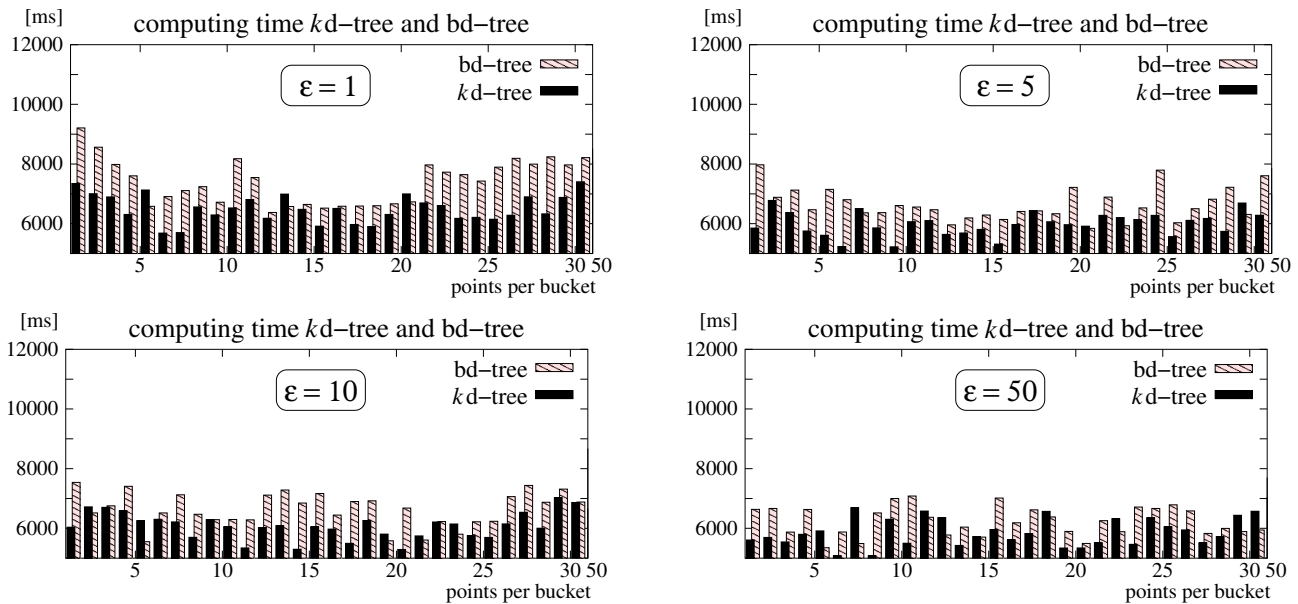


Fig. 9. Computing time in milliseconds for a 3D scan matching depending on the bucket size of approximate kd - and approximate bd -tree. Values for $\epsilon = 1, 5, 10, 50$ are given. In the majority of cases, the approximate kd -tree outperforms the approximate bd -tree.

REFERENCES

- [1] P. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer. AVENUE: Automated Site Modeling in Urban Environments. In *Proceedings of the third International Conference on 3D Digital Imaging and Modeling (3DIM '01)*, Quebec City, Canada, May 2001.
- [2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least square fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698 – 700, 1987.
- [3] S. Arya and D. M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *Proceedings of the 4th ACM-SIAM Symposium on Discrete Algorithms*, pages 271 – 280, 1993.
- [4] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An Optimal Algorithms for Approximate Nearest Neighbor Searching in Fixed Dimensions. *Journal of the ACM*, (45):891 – 923, 1998.
- [5] R. Benjema and F. Schmitt. Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique. In *Proceedings IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling (3DIM '97)*, Ottawa, Canada, May 1997.
- [6] J. L. Bentley. Multidimensional binary search trees used for associative searchin. *Communications of the ACM*, 18(9):509 – 517, September 1975.
- [7] P. Besl and N. McKay. A method for Registration of 3–D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239 – 256, February 1992.
- [8] P. Biber, H. Andreasson, T. Duckett, and A. Schilling. 3D Modeling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04)*, Sendai, Japan, September 2004.
- [9] Y. Chen and G. Medioni. Object Modelling by Registration of Multiple Range Images. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA '91)*, pages 2724 – 2729, Sacramento, CA, USA, April 1991.
- [10] D. Eggert, A. Fitzgibbon, and R. Fisher. Simultaneous Registration of Multiple Range Views Satisfying Global Consistency Constraints for Use In Reverse Engineering. *Computer Vision and Image Understanding*, 69:253 – 272, March 1998.
- [11] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transaction on Mathematical Software*, 3(3):209 – 226, September 1977.
- [12] C. Früh and A. Zakhor. 3D Model Generation for Cities Using Aerial Photographs and Ground Level Laser Scans. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR '01)*, Kauai, Hawaii, USA, December 2001.
- [13] A. Georgiev and P. K. Allen. Localization methods for a mobile robot in urban environments. *IEEE Transaction on Robotics and Automation (TRO)*, 20(5):851 – 864, October 2004.
- [14] M. Greenspan and M. Yurick. Approximate K-D Tree Search for Efficient ICP. In *Proceedings of the 4th IEEE International Conference on Recent Advances in 3D Digital Imaging and Modeling (3DIM '03)*, pages 442 – 448, Banff, Canada, October 2003.
- [15] D. Hähnel, W. Burgard, and S. Thrun. Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot. In *Proceedings of the fourth European workshop on advanced mobile robots (EUROBOT '01)*, Lund, Sweden, September 2001.
- [16] M. Hebert, M. Deans, D. Huber, B. Nabbe, and N. Vandapel. Progress in 3–D Mapping and Localization. In *Proceedings of the 9th International Symposium on Intelligent Robotic Systems, (SIRS '01)*, Toulouse, France, July 2001.
- [17] A. Lorusso, D. Eggert, and R. Fisher. A Comparison of Four Algorithms for Estimating 3-D Rigid Transformations. In *Proceedings of the 5th British Machine Vision Conference (BMVC '95)*, pages 237 – 246, Birmingham, England, September 1995.
- [18] M. M. Nevado, J. G. Garcia-Bermejo, and E. Z. Casanova. Obtaining 3d models of indoor environments with a mobile robot by estimating local surface directions. *Robotics and Autonomous Systems*, 48:131 – 143, August 2004.
- [19] A. Nüchter. *Autonome Exporation und Modellierung von 3D-Umgebungen*, GMD Report 157. GMD, Sankt Augustin, 2002.
- [20] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with an Application in autonomous mine mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1998 – 2003, New Orleans, USA, April 2004.
- [21] K. Pulli. Multiview Registration for Large Data Sets. In *Proceedings of the 2nd International Conference on 3D Digital Imaging and Modeling (3DIM '99)*, pages 160 – 168, Ottawa, Canada, October 1999.
- [22] S. Se, D. Lowe, and J. Little. Local and Global Localization for Mobile Robots using Visual Landmarks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '01)*, Hawaii, USA, October 2001.
- [23] V. Sequeira, K. Ng, E. Wolfart, J. Goncalves, and D. Hogg. Automated 3D reconstruction of interiors with multiple scan-views. In *Proceedings of SPIE, Electronic Imaging '99, The Society for Imaging Science and Technology /SPIE's 11th Annual Symposium*, San Jose, CA, USA, January 1999.
- [24] D. Simon, M. Hebert, and T. Kanade. Real-time 3–D pose estimation using a high-speed range sensor. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '94)*, volume 3, pages 2235 – 2241, San Diego, CA, USA, May 1994.
- [25] I. Stamos and P. Allen. 3-D Model Construction Using Range and Image Data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR '00)*, USA, June 2000.
- [26] H. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg. A 3D laser range finder for autonomous mobile robots. In *Proceedings of the*

the 32nd International Symposium on Robotics (ISR '01), pages 153 – 158, Seoul, Korea, April 2001.

- [27] H. Surmann, A. Nüchter, K. Lingemann, and J. Hertzberg. 6D SLAM A Preliminary Report on Closing the Loop in Six Dimensions. In *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV '04)*, Lisbon, Portugal, July 2004.
- [28] S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with application to multi robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, San Francisco, CA, USA, April 2000.

- [29] O. Wulf, K. O. Arras, H. I. Christensen, and B. A. Wagner. 2D Mapping of Cluttered Indoor Environments by Means of 3D Perception. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4204 – 4209, New Orleans, USA, April 2004.
- [30] H. Zhao and R. Shibasaki. Reconstructing Textured CAD Model of Urban Environment Using Vehicle-Borne Laser Range Scanners and Line Cameras. In *Second International Workshop on Computer Vision System (ICVS '01)*, pages 284 – 295, Vancouver, Canada, July 2001.

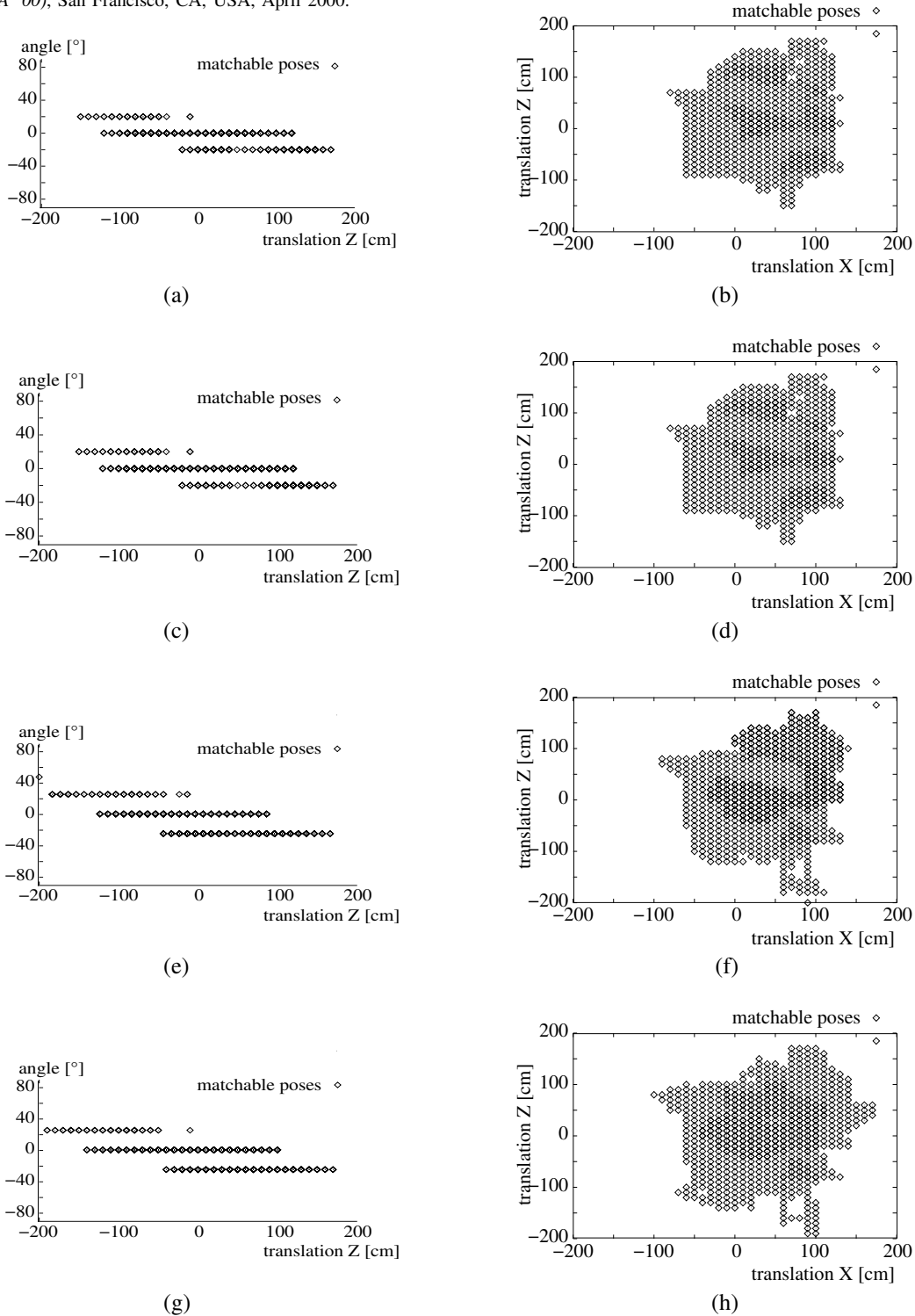


Fig. 10. Initial poses are marked in (x, z, θ_y) from which a correct alignment of two 3D scans is possible. The scans of Fig. 2 have been used. (a) and (b): $\epsilon = 1$ and $b = 10$. (c) and (d): $\epsilon = 1$ and $b = 20$. (e) and (f): $\epsilon = 10$ and $b = 10$. (g) and (h): $\epsilon = 50$ and $b = 5$. The rows represent second angle projections of the 3D space of matchable poses.