Online 3D SLAM by Registration of Large Planar Surface Segments and Closed Form Pose-Graph Relaxation

Kaustubh Pathak* **Andreas Birk** **Narunas Vaskevicius**

Max Pfingsthorn **Sören Schwertfeger** **Jann Poppinga**

Abstract

A fast pose-graph relaxation technique is presented in this article for enhancing the consistency of 3D maps created by registering large planar surface patches. The surface patches are extracted from point-clouds sampled from a 3D range-sensor. The planes based registration method offers an alternative to the state of the art algorithms and provides advantages in terms of robustness, speed, and storage. One of its features is that it results in an accurate determination of rotation, although a lack of predominant surfaces in certain directions may result in translational uncertainty in those directions. Hence, a loop-closing and relaxation problem is formulated which gains significant speed by relaxing only the translational errors, and utilizes the full translation covariance determined during pairwise registration. This leads to a fast 3D Simultaneous Localization and Mapping (SLAM) suited for online operations. The approach is tested in two disaster scenarios that were mapped at the NIST 2008 Response Robot Evaluation Exercise (RREE) in Disaster City, Texas, USA. The two data-sets from a collapsed car park and a flooding disaster consist of 26 and 70 3D scans respectively. The results of these experiments show that our approach can generate 3D maps without motion estimates by odometry, and that it outperforms Iterative Closest Point (ICP) based mapping with respect to speed and robustness.

1 Introduction

There have been many recent attempts [Surmann et al., 2003, Thrun et al., 2003, Weingarten and Siegwart, 2006, Magnusson et al., 2007, Nüchter et al., 2007] to extend the methodology of scan-matching to 3D sensors. Almost all of this research is based on matching 3D point-clouds of the environment, which are generated using a 3D range sensor and then matched using an algorithm derived from the Iterative Closest Point (ICP) algorithm

*The authors are with the Dept. of Electrical Engineering and Computer Science, Jacobs University Bremen, 28751 Bremen, Germany. (k.pathak, a.birk)@jacobs-university.de

This is the peer reviewed version of the following article:

Pathak, K., A. Birk, N. Vaskevicius, M. Pfingsthorn, S. Schwertfeger, and J. Poppinga, "Online 3D SLAM by Registration of Large Planar Surface Segments and Closed Form Pose-Graph Relaxation", Journal of Field Robotics, Special Issue on 3D Mapping, vol. 27, no. 1, pp. 52-84, 2010.

This article has been published in final form at http://dx.doi.org/10.1002/rob.20322 . This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Self-Archiving.

[Besl and McKay, 1992]. However, this algorithm, apart from being computationally expensive and slow for large point-clouds of the order of 10^4–10^6 points, also suffers from premature convergence to local minima, especially when the overlap between scene-samples decreases.

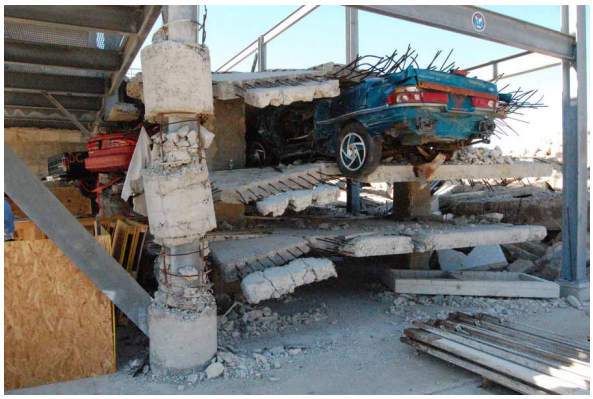
This article is concerned with 3D-Plane-SLAM, which builds upon robot pose registration by matching large planar surface patches. The core of the plane-matching registration algorithm and a derivation of its principles are introduced in [Pathak et al., 2009b] where results in an indoor lab environment are presented. In this article, two important extensions of this work on 3D registrations are made. First, it is shown that the algorithm also performs very well on data from two challenging field experiments conducted during the 2008 NIST Response Robot Evaluation Exercise (RREE) at Disaster City, Texas [TEEX, 2008]. The performance of the plane based registration is compared to ICP in two disaster scenarios, namely a collapsed multi-story car parking (Fig. 1) and a flooding disaster (Fig. 2).

Second, the plane registration algorithm is extended in this article to a proper 3D Simultaneous Localization and Mapping (SLAM) approach. This 3D-Plane-SLAM exploits the fact that the plane registration is very robust with respect to rotations as also shown in the experiments. Hence, relaxation in pose-graph SLAM can be restricted to 3D translations allowing for a closed form solution that can be efficiently computed. It is shown in the examples of the two disaster scenarios that this fast relaxation indeed leads to a quantitative and qualitative improvement of the 3D maps.

Previous work in map-building using surface-features has taken two main approaches: 1) utilization of complex neuro-fuzzy rules [Fischer and Kohlhepp, 2000, Kohlhepp et al., 2004, Kohlhepp et al., 2006] to determine surface-correspondences across samples for pose-registration; and 2) extraction of planar surfaces and keeping them as features [Weingarten and Siegwart, 2006] in EKF-SLAM, while still using point-cloud based ICP to help in registration and correspondence determination. Our new approach utilizes planar segments to systematically do surface-extraction and subsequently compute pose registration along with its covariance, while dispensing away with the use of large point-clouds altogether. Furthermore, it has distinct advantages in terms of computation speed, storage, and understandability of the resulting map. One of the main reasons for its computational efficiency is that it is based on the extraction of large surface patches and not on local surface features. The registration of two scans hence deals with a dozen or so large surface patches in contrast to the one or several orders of magnitude more features that are found by local surface analysis which is common practice in the Computer Vision and Imaging community for 3D registration. Also, typical robotics range sensors, like the actuated laser range finder (ALRF) used in our work, are too noise-prone for the standard computer vision approaches, as demonstrated and discussed in [Hähnel et al., 2003].

Both registration approaches of [Fischer and Kohlhepp, 2000, Kohlhepp et al., 2004, Kohlhepp et al., 2006] and [Weingarten and Siegwart, 2006] as well as our own [Pathak et al., 2009b] were mainly tested indoors in office like environments where large planar surfaces prevail. The two scenarios investigated here are more challenging in several respects as discussed in some detail in the sections 3.1 and 6.1. Furthermore, the plane registration is embedded in a pose graph implementation [Pfingsthorn and Birk, 2008, Pfingsthorn et al., 2007, Olson et al., 2006a] for Simulta-

(a) Front overview of the crashed car parking-lot. Note the rubble to the right.



(b) Close-up front view corresponding to the lower-right part of the façade in Fig. 1(a).

(c) Left view of the crushed car.

(d) The robot collecting data under the collapsed ceiling of the "ground floor" under the car. The yellow rotating LRF sensor is clearly visible.

Figure 1: The crashed parking lot at Disaster City in College Station, Texas [TEEX, 2008].

Figure 2: The “dwelling” scenario at Disaster City simulates a flooding disaster.

neous Localization and Mapping (SLAM), which improves the results. Especially, there is a conceptual contribution in the pose graph implementation: the plane registration is inherently very robust for rotations, thus allowing an extremely fast relaxation in a closed form solution.

The rest of this article is organized as follows. Section 2 provides an introduction to the plane based registration, which consists of the following main steps: 1) extraction of planes with the corresponding uncertainties; 2) plane matching, i.e., finding correspondences between planar surface-segments in the two scans to be matched; 3) the actual pose-registration, i.e., determining the optimal rotation and translation which aligns the corresponding set of planes, once the correspondences have been decided on. Based on the crashed car park scenario, an experimental evaluation of the performance with respect to speed and robustness of the plane based registration is presented in Sec. 3. In doing so, a detailed comparison to the performance of ICP is included. Section 4 presents the theoretical formulation of the pose-graph relaxation technique for 3D-Plane-SLAM. This is followed by results presented in Sec. 5 where it is shown that the relaxation indeed improves the map quality in the crashed car park scenario in quantitative as well as qualitative terms. Further experimental results based on a second disaster scenario known as the "dwelling" are presented in Sec. 6. A performance comparison with ICP is also included for this data set. It is shown that the plane based registration is fast and robust and that the 3D-Plane-SLAM relaxation improves the mapping results. A discussion of the merits and the limitations of plane based mapping is provided in Sec. 7. The article is concluded in Sec. 8.

2 Plane-Segment Extraction and Matching

2.1 Overview

The scan-matching based on plane-segments consists of the following steps:

1. **Planes extraction from raw point-clouds**: This procedure is based on region-growing in a range-image scan followed by a least-squares estimation of the parameters of planes [Poppinga et al., 2008]. The covariances of the plane-parameters are computed as well [Pathak et al., 2009a].
2. **Matching planes**: This step consists of finding correspondences between planar surface-segments in the two scans to be matched. These two scans are typically successive samples for normal registration but they may be also be non-successive, for example if a loop is closed.
3. **Computing pose-registration**: After the correspondences have been decided on, the optimal rotation and translation which aligns the corresponding set of planes must be found. This gives the pose change of the robot between the scans.
4. **Polygonization**: This step consists of polygonizing each plane-segment by finding and fitting the boundary of each surface-patch so that the surface can be compactly described. This step is crucial for visualization of the result– however, if only pose registration is desired, it may be omitted. It is also described in [Poppinga et al., 2008].

These steps are followed by a global relaxation described in Sec. 4 to form 3D-Plane-SLAM. We now provide a brief overview of plane-matching, i.e., the third step, because an understanding of its properties aids the understanding of why the relaxation step is designed in a certain way.

2.2 Notation

A plane $P(\hat{\mathbf{m}}, \rho)$ is given by the equation $\hat{\mathbf{m}} \cdot \mathbf{r} = \rho$, where ρ is the signed distance from the origin in the direction of the unit plane normal $\hat{\mathbf{m}}$. It can be seen that $P(\hat{\mathbf{m}}, \rho) \triangleq P(-\hat{\mathbf{m}}, -\rho)$. To achieve a consistent sign convention, we define planes as $P(\hat{\mathbf{n}}, d)$, where, $d \triangleq |\rho| \geq 0$, and $\hat{\mathbf{n}} \triangleq \sigma(\rho)\, \hat{\mathbf{m}}$, where, $\sigma(\rho) = -1$ if $\rho < 0$ and $+1$ otherwise. If $\rho = 0$, then we choose the maximum component of $\hat{\mathbf{n}}$ to be positive. The latter case is unlikely to occur in practice in the sensor-frame, because such a plane, which is parallel to the line of sight of the range sensor, is unlikely to be detected by it.

For denoting frames and relative transforms, the notation of [Craig, 2005] is used. The j-th sample is associated with a frame $\mathcal{F}_j$, in which the set of extracted planes is denoted as ${}^{j}\mathcal{P}$. Suppose we are given two samples ${}^{j}\mathcal{P}$ and ${}^{k}\mathcal{P}$. Usually they are successive, but they may be non-successive, for example during loop-closing. Using the procedure described in [Pathak et al., 2009a], one can compute an isotropic uncertainty measure σ^2 which is derived - usually, using the trace operation - from the 4×4 covariance matrix $\mathbf{C}$ of the plane parameters $\hat{\mathbf{n}}$ and d associated with it. Thus a plane-set ${}^{k}\mathcal{P}$ is an ordered set of triplets given by

$$ {}^{k}\mathcal{P} \triangleq \{\, {}^{k}\mathcal{P}_i \langle\, {}^{k}\hat{\mathbf{n}}_i,\, {}^{k}d_i,\, {}^{k}\sigma_i^2 \rangle,\ i = 1 \ldots \mathsf{N}_k \}, \tag{1} $$

2.3 Decoupling of Rotation and Translation

If the robot – more precisely, the sensor mounted on the robot – moves from $\mathcal{F}_j$ to $\mathcal{F}_k$, i.e. rotates by ${}^{j}_{k}\mathbf{R}$ and translates by ${}^{j}_{k}\mathbf{t}$ between samples j and k resolved in $\mathcal{F}_j$, then the Cartesian coordinates ${}^{j}\mathbf{p}$ and ${}^{k}\mathbf{p}$ of the same physical point observed from the two frames are related by

$$ {}^{j}\mathbf{p} = {}^{j}_{k}\mathbf{R}\ {}^{k}\mathbf{p} + {}^{j}_{k}\mathbf{t}. \tag{2} $$

Substituting the above in the plane equation, one can derive that if ${}^{j}\mathcal{P}_i$ and ${}^{k}\mathcal{P}_i$ represent the same i-th physical plane, the plane parameters resolved in the two frames are related by

$$ {}^{j}\hat{\mathbf{n}}_i = {}^{j}_{k}\mathbf{R}\, {}^{k}\hat{\mathbf{n}}_i \tag{3} $$

$$ {}^{j}\hat{\mathbf{n}}_i^{\mathsf{T}}\, {}^{j}_{k}\mathbf{t} = {}^{j}d_i - {}^{k}d_i \tag{4} $$

The above equations show that the rotation and the translation components have been nicely decoupled.

2.4 Pose-registration by plane-matching

Statement of the Problem Given noisy plane-sets ${}^{j}\mathcal{P}$ and ${}^{k}\mathcal{P}$, with correspondences between planes known and denoted by the common index i, find the optimum rotation ${}^{j}_{k}\mathbf{R}$

and the optimum translation ${}^{j}_{k}\mathbf{t}$. We provide a solution below.

2.4.1 Optimum Rotation

To find the optimum rotation between ${}^{j}\mathcal{P}$ and ${}^{k}\mathcal{P}$, we maximize the following value-function

$$\max_{{}^{j}_{k}\mathbf{R}} \zeta_r \triangleq \frac{1}{2}\sum_{i=1}^{\mathsf{N}} ({}^{j}\sigma_i^2 + {}^{k}\sigma_i^2)^{-1}\, {}^{j}\hat{\mathbf{n}}_i^{\mathsf{T}}\, {}^{j}_{k}\mathbf{R}\, {}^{k}\hat{\mathbf{n}}_i \tag{5}$$

This is the well-known Wahba's problem [Shuster, 2006]. To solve this we parameterize the rotation ${}^{j}_{k}\mathbf{R}$ with quaternions ${}^{j}_{k}\check{\mathbf{q}}$ and proceed as in [Horn, 1987]. Using the above, one can reformulate (5) as

$$\max_{{}^{j}_{k}\check{\mathbf{q}}} \zeta_r \triangleq \frac{1}{2}\, {}^{j}_{k}\check{\mathbf{q}}^{\mathsf{T}}\mathbf{K}\, {}^{j}_{k}\check{\mathbf{q}}, \tag{6}$$

where the matrix $\mathbf{K}$ is depends on ${}^{j}\hat{\mathbf{n}}_i$ and ${}^{k}\hat{\mathbf{n}}_i$, $i = 1 \dots \mathsf{N}$. The optimum quaternion ${}^{j}_{k}\check{\mathbf{q}}$ is then the eigenvector of the matrix $\mathbf{K}$ corresponding to its maximum eigenvalue $\bar{\mu}(\mathbf{K})$. The 4×4 covariance of this optimum ${}^{j}_{k}\check{\mathbf{q}}$ can be computed as

$${}^{j}_{k}\mathbf{C}_{\check{\mathbf{q}}\check{\mathbf{q}}} = -\left(\mathbf{K} - \bar{\mu}(\mathbf{K})\,\mathbf{I}_4\right)^{+}, \tag{7}$$

where, $\mathbf{X}^{+}$ represents the Moore-Penrose inverse of the matrix $\mathbf{X}$ [Kanatani, 2005]. This covariance matrix can then be transformed to roll-pitch-yaw (RPY) space by using the Jacobian of the transform between the quaternion and the RPY space. Finally, we note that at least two non-parallel pairs of planes are required to fully determine rotation.

2.4.2 Optimum Translation

The translation determination is central to the relaxation step presented later on. Therefore, we review it in detail in this section. Equation (4) can be stacked-up to give

$$\mathbf{M}\, {}^{j}_{k}\mathbf{t} = \mathbf{d}, \tag{8a}$$

where,

$$\mathbf{M}_{\mathsf{N}\times 3} \triangleq \begin{bmatrix} {}^{j}\hat{\mathbf{n}}_1^{\mathsf{T}} \\ \vdots \\ {}^{j}\hat{\mathbf{n}}_{\mathsf{N}}^{\mathsf{T}} \end{bmatrix}, \qquad \mathbf{d}_{\mathsf{N}\times 1} \triangleq \begin{bmatrix} {}^{j}d_1 - {}^{k}d_1 \\ \vdots \\ {}^{j}d_{\mathsf{N}} - {}^{k}d_{\mathsf{N}} \end{bmatrix} \tag{8b}$$

Due to its intuitive nature and fast closed-form solution, we will solve Eq. (8a) with ordinary least squares (LS). A diagonal weighting matrix $\mathbf{W}$ is defined as

$$\mathbf{\Sigma} \triangleq \begin{pmatrix} {}^{j}\sigma_1^2 + {}^{k}\sigma_1^2 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & {}^{j}\sigma_{\mathsf{N}}^2 + {}^{k}\sigma_{\mathsf{N}}^2 \end{pmatrix}, \tag{9}$$

$$\mathbf{W} \triangleq \left(\mathbf{\Sigma}^{-1}\right)^{1/2}. \tag{10}$$

Then the LS solution minimizes $\|\mathbf{W}(\mathbf{M}\, {}^{j}_{k}\mathbf{t} - \mathbf{d})\|$. If $\mathbf{M}$ is full rank, the least-squares optimum translation is ${}^{j}_{k}\mathbf{t} = \left(\mathbf{M}^{\mathsf{T}}\mathbf{W}^2\mathbf{M}\right)^{-1}\mathbf{M}^{\mathsf{T}}\mathbf{W}^2\mathbf{d}$.

Unlike rotation, we need in general $\mathsf{N} \geq 3$ mutually non-parallel planes to find ${}^{j}_{k}\mathbf{t}$. The above formula is not a good way to compute the solution because $\mathbf{M}$ may be ill-conditioned, may be rank-deficient, or $\mathsf{N} < 3$. A more general way to solve the equation is presented next. We define

$$\hat{\mathbf{M}} \triangleq \mathbf{W}\mathbf{M}, \qquad \hat{\mathbf{d}} \triangleq \mathbf{W}\mathbf{d}. \tag{11}$$

Let the singular-value decomposition of $\hat{\mathbf{M}}$ be given by $\mathbf{U}_{\mathsf{N}\times\mathsf{N}}\mathbf{\Lambda}_{\mathsf{N}\times 3}\mathbf{V}^{\mathsf{T}}_{3\times 3}$. $\mathbf{\Lambda}$ has non-negative singular values λ_i^2 arranged in descending order. The column unit vectors of $\mathbf{U}$ are denoted $\mathbf{u}_i,\ i = 1 \ldots \mathsf{N}$ and the column unit vectors of $\mathbf{V}$ are denoted $\mathbf{v}_i,\ i = 1 \ldots 3$.

Let $\mathsf{N}_{\hat{\mathbf{M}}} \leq 3$ be the *effective* rank of $\hat{\mathbf{M}}$. If the largest singular value $\lambda_1^2 < \epsilon_1$, then the effective rank is 0. The parameter ϵ_1 is dependent on machine accuracy. If $\lambda_1^2 \geq \epsilon_1$, then the effective rank is found by finding the count of all singular values $\lambda_i^2 > \lambda_1^2/\bar{c}$ in the diagonal matrix $\mathbf{\Lambda}$, where $\bar{c}$ is the maximum allowable condition number of the matrix.

Then the best rank $\mathsf{N}_{\hat{\mathbf{M}}}$ approximation of $\hat{\mathbf{M}}$ is

$$\tilde{\mathbf{M}} = \sum_{i=1}^{\mathsf{N}_{\hat{\mathbf{M}}}} \lambda_i^2 \hat{\mathbf{u}}_i \hat{\mathbf{v}}_i^{\mathsf{T}}, \qquad \mathsf{N}_{\hat{\mathbf{M}}} \leq 3. \tag{12}$$

The span of the orthogonal unit vectors $\hat{\mathbf{u}}_i,\ i = 1 \ldots \mathsf{N}_{\hat{\mathbf{M}}}$ gives the best approximation for the range-space of $\hat{\mathbf{M}}$. Therefore, the closest we can get to $\hat{\mathbf{d}}$ is $\tilde{\mathbf{d}} = \sum_{i=1}^{\mathsf{N}_{\hat{\mathbf{M}}}} (\hat{\mathbf{u}}_i \cdot \hat{\mathbf{d}})\hat{\mathbf{u}}_i$, which gives the corresponding translation estimate

$$\begin{aligned} {}^{j}_{k}\mathbf{t} &= \sum_{i=1}^{\mathsf{N}_{\hat{\mathbf{M}}}} \lambda_i^{-2} (\hat{\mathbf{u}}_i \cdot \hat{\mathbf{d}})\hat{\mathbf{v}}_i & (13) \\ &\triangleq \hat{\mathbf{M}}^{+}\hat{\mathbf{d}}, \qquad \hat{\mathbf{M}}^{+} \triangleq \sum_{i=1}^{\mathsf{N}_{\hat{\mathbf{M}}}} \lambda_i^{-2} \hat{\mathbf{v}}_i \hat{\mathbf{u}}_i^{\mathsf{T}}. & (14) \end{aligned}$$

This is also the minimum 2-norm solution of the LS problem regardless of the rank of $\mathbf{M}$ [Golub and Loan, 1980].

Note that for directions $\hat{\mathbf{v}}_i,\ i = (\mathsf{N}_{\hat{\mathbf{M}}}+1) \ldots 3$, we have no information about the translation. One option is to keep these components 0 and inject large uncertainty along those directions in the covariance matrix. However, if an odometry estimate ${}^{j}_{k}\mathbf{t}_{\mathrm{O}}$ and its covariance matrix ${}^{j}_{k}\mathbf{C}_{\mathbf{tt},\mathrm{O}}$ are available, we can use them *only* for these missing components. In this case, we have

$$\begin{aligned} {}^{j}_{k}\mathbf{t} &= \hat{\mathbf{M}}^{+}\mathbf{d} + \sum_{i=\mathsf{N}_{\hat{\mathbf{M}}}+1}^{3} ({}^{j}_{k}\mathbf{t} \cdot \hat{\mathbf{v}}_i)\hat{\mathbf{v}}_i, \\ &\triangleq \hat{\mathbf{M}}^{+}\mathbf{d} + \mathbf{M}_{\mathrm{O}}\, {}^{j}_{k}\mathbf{t}_{\mathrm{O}}, \quad \mathbf{M}_{\mathrm{O}} \triangleq \sum_{i=\mathsf{N}_{\hat{\mathbf{M}}}+1}^{3} \hat{\mathbf{v}}_i \hat{\mathbf{v}}_i^{\mathsf{T}}. \end{aligned} \tag{15}$$

The covariance of odometry is in general only very roughly known because it depends on the vehicle model, unknown slippage, etc. A fusion of odometry translation and plane-matching

translation using this covariance should hence not be done per default. But the nice thing about this solution is that it automatically detects the directions for which the translation is uncertain, and it resorts to odometry – if available – only for these directions. This is, for example, of interest when the robot moves along a long corridor where the end cannot be seen, and hence there is high uncertainty along the corridor direction. In such cases where the translation is not fully determined in some directions, the uncertainty in translation increases, which then needs to be mitigated by loop-closing and relaxation, i.e., by proper SLAM, as introduced in Sec. 4.

To round-off the solution, we can write the estimate of the covariance matrix for translation as follows

$$\begin{aligned} {}^{j}_{k}\mathbf{C}_{\mathbf{tt}} &= \hat{\mathbf{M}}^{+}\mathbf{W}\mathbf{\Sigma}\mathbf{W}^{\mathsf{T}}(\hat{\mathbf{M}}^{+})^{\mathsf{T}} + \mathbf{M}_{\mathrm{O}}\,{}^{j}_{k}\mathbf{C}_{\mathbf{tt},\mathrm{O}}\,\mathbf{M}_{\mathrm{O}}^{\mathsf{T}}, \\ &= \mathbf{M}^{+}\mathbf{\Sigma}\,(\mathbf{M}^{+})^{\mathsf{T}} + \mathbf{M}_{\mathrm{O}}\,{}^{j}_{k}\mathbf{C}_{\mathbf{tt},\mathrm{O}}\,\mathbf{M}_{\mathrm{O}}^{\mathsf{T}} \end{aligned} \tag{16}$$

where the last equation comes from simplification using (10) and (11). This covariance plays a central role in the pose-graph relaxation.

2.5 Determining Correspondences

It is assumed in Sec. 2.4 that $\mathsf{N}_{k\leftrightarrow k+1}$ pairs of corresponding planes have already been found. However, detecting that $P_{k,i}$ is the same physical plane patch as $P_{k+1,j}$ is a difficult problem. We denote correspondence by $P_{k,i} \leftrightarrow P_{k+1,j}$ or simply as $i \leftrightarrow j$. Many similarity metrics like size, shape, color, geometric consistency, etc. can be used to infer correspondence.

In [Pathak et al., 2009b], we have introduced an algorithm for solving this problem in a geometric manner, namely by determining the correspondence-set which maximizes the global rigid body motion constraint. It makes use of the fact that the number of planes in a scene is much less than the size of a typical point-cloud. The algorithm scales as $O(\mathsf{N}^4)$, where N is the average number of planes in the two scans being matched. For details, the reader is referred to the aforementioned article.

3 A Performance Comparison of Iterative Closest Point (ICP) and Plane Registration

3.1 The Collapsed Car Park Scenario

In this section, a first comparison with ICP is presented based on data from a disaster scenario– a collapsed car parking-lot in Disaster City, Texas. The data was collected in November 2008 during the annual NIST Response Robot Evaluation Exercise (RREE) [TEEX, 2008]. As shown in Fig. 1(d), the Jacobs University rescue robot used in the RREE is equipped with an actuated laser range-finder (ALRF). The ALRF has a horizontal field of view of 270° of 541 beams. The sensor is rotated (pitched) by a servo from $-90°$ to $+90°$ at a spacing of 0.5°. This gives a 3D point-cloud of a total size of $541 \times 361 = 195,301$ points per sample. The maximum range of the sensor is about 30 meters. The mobile robot was

teleoperated and stopped occasionally to take scans. The time to take one full scan is about $T_{\text{scan}} \approx 32$ seconds. This data acquisition is currently the bottleneck in our 3D mapping but only due to the low cost sensor used for this purpose. By using a much faster but also more costly 3D LRF like a Velodyne HDL-64E, this data acquisition time could be reduced by two orders of magnitude.

A total of 26 usable scans were taken in this scenario. Figure 3 shows the view of the front camera of the robot at the locations where the scans where taken. The whole front camera movie of the moving robot can be downloaded from the web (see Sec. 7.3). Note that the conditions in which the robot operated include a high presence of dust, rubble, and so on, which limit the accuracy of motion estimates by odometry. The front camera pictures are taken with a wide angle lens with an horizontal opening angle of approximately 85°. The camera hence only covers a fraction of the data gathered by the ALRF with its 270° horizontal field of view. Figure 4 shows an example scan as point cloud from two different perspectives and as a gray scale range image, which provides a rough overview of the data contained in the scan. The range-images of all 26 scans of the collapsed car park scenario are shown in Fig. 5.

3.2 ICP Registration

A point-to-point Iterative Closest Point (ICP) registration is used as a comparison baseline for the plane-registration. Our implementation of ICP is straightforward, with a kd-tree for fast nearest neighbor operations, but no other specialized optimizations. A standard stopping criterion based on the percentage of the decrease of the mean squared error (MSE) from time step t to $t+1$ is used, i.e., the registration is considered to have converged if $(MSE_t - MSE_{t+1})/MSE_t < T_{\text{stop}}$ with $T_{\text{stop}} = 10^{-4}$. Subsampling of the point clouds is used to improve the runtime. The experiments are done with subsampling by factor 3 and 10. As discussed later on, these rates of subsampling have little effect on the quality of the results; they just speed up the processing by ICP. ICP is run on the scan pairs without any initial guess on the robot's motion. As discussed later on, using odometry significantly worsens the results. In fact, ICP fails to do any registration if the highly inaccurate odometry is used in this scenario.

The run times for the registrations vary significantly between scan pairs as shown in Table 1. The average time for registration of a scan pair with subsampling by 3 is about 18 minutes. It is still 1:45 minutes with subsampling by 10. This is significantly larger than the run-times of the plane based registration as shown in Sec. 3.3. But what is more important than the run-times – which may be improved by specialized versions of ICP – is that ICP fails to register a significant number of scan pairs.

The results of the pairwise registrations were manually inspected. The results of this qualitative assessment are also shown in Table 1. As also discussed in more detail later on, the qualitative assessment did not differ for the different resolutions of the subsampling. A registration was only judged to have failed if there are severe displacements between the two supposedly registered scans. The results clearly fall into two very distinctive categories: either ICP succeeds well or it clearly fails. An example of a successful ICP registration is

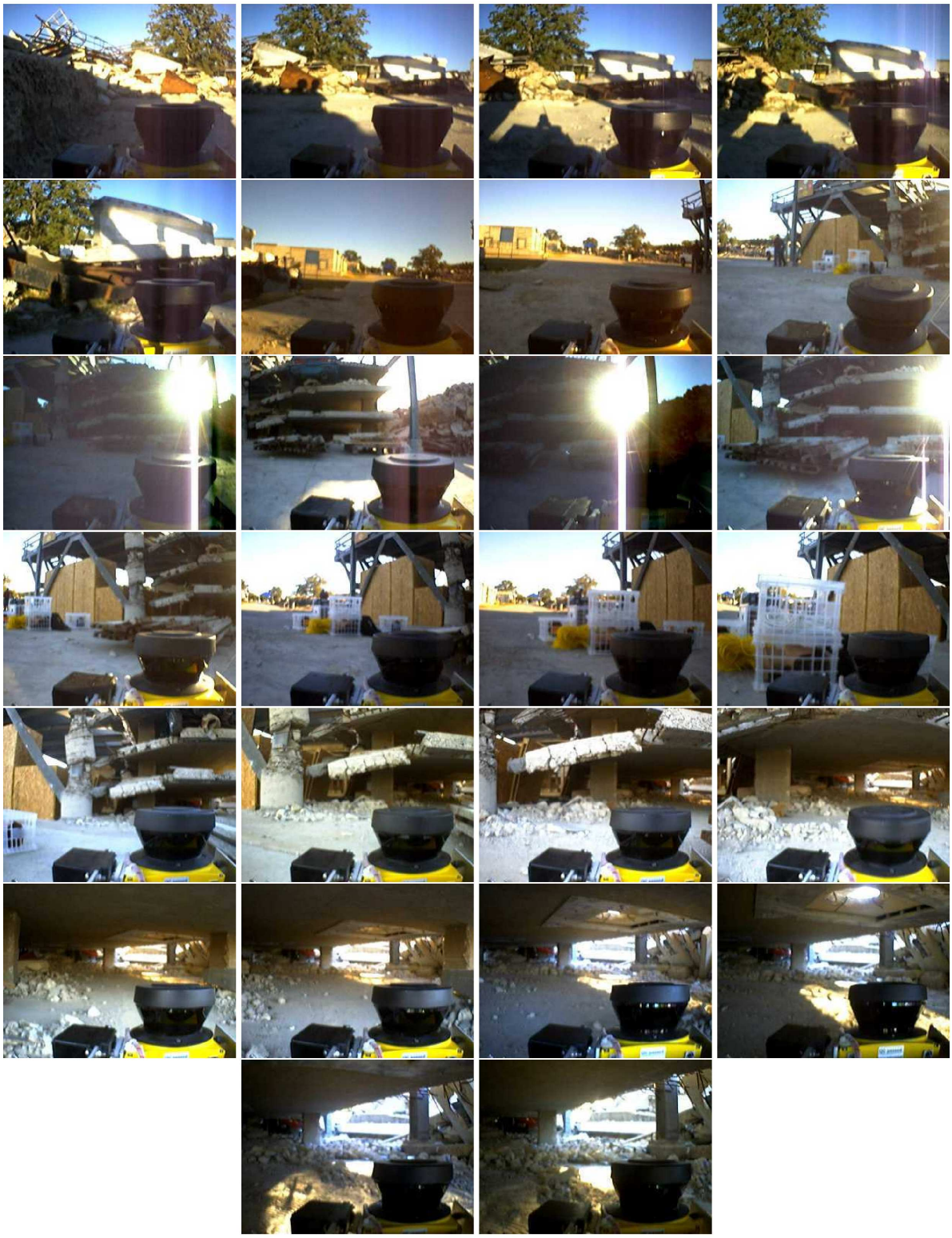
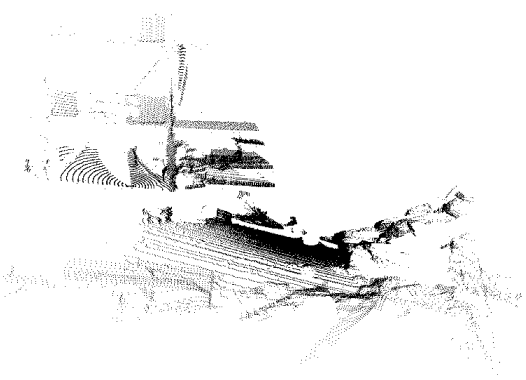
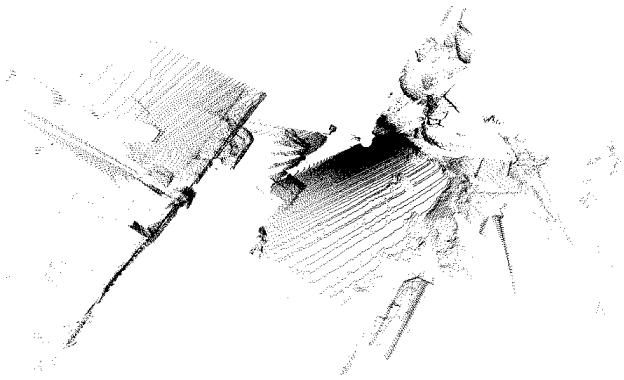


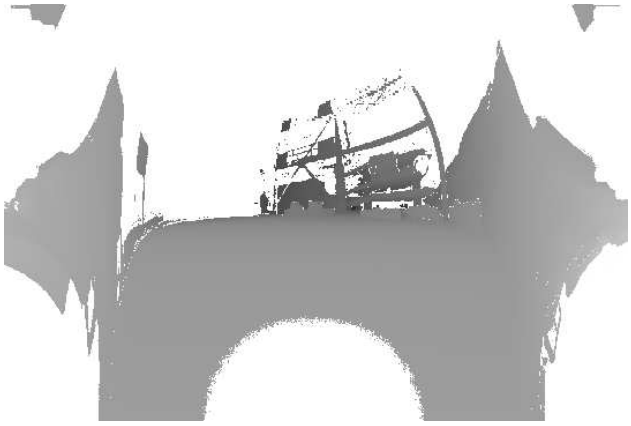
Figure 3: The front camera view of the scenes 0–25 at the locations where the robot took a scan. The first image with index 0 is shown at the top left.



(a) scan 7, point cloud, front view



(b) scan 7, point cloud, top view



(c) scan 7, range image

Figure 4: An example scan of the collapsed car park scenario as point cloud and as gray scale range image.

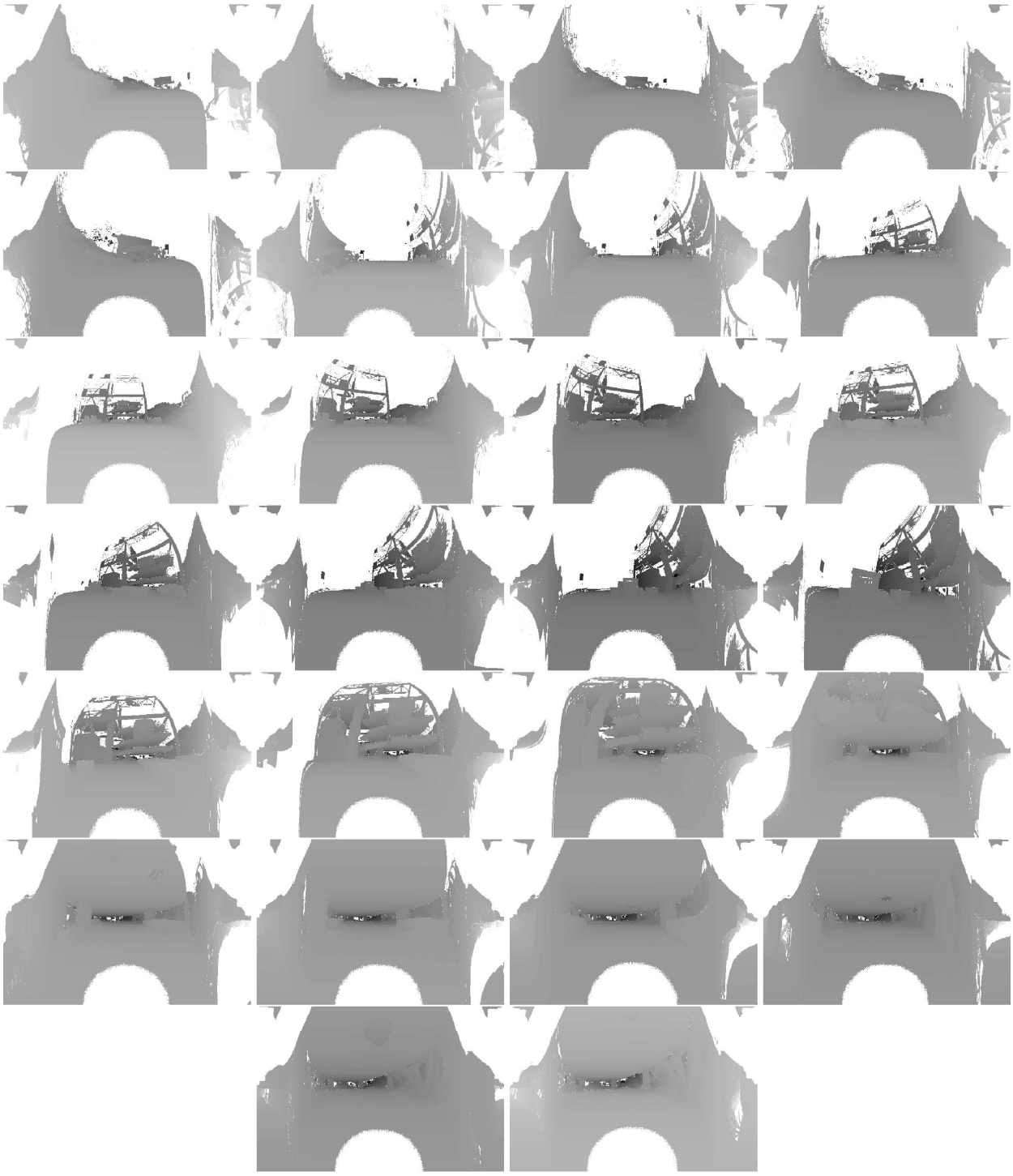
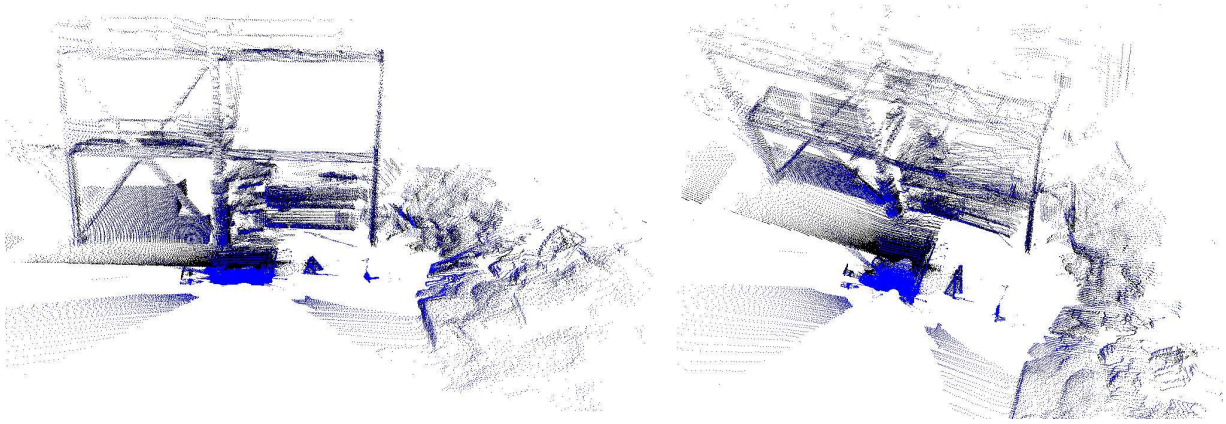


Figure 5: The 26 scans of the collapsed car park as gray-scale range-images. The first scan with index 0 is shown at the top left.

Pair	**Subsampled by 3 time (min:sec)**	**Subsampled by 10 time (min:sec)**	**Success**
$0 \rightarrow 1$	02:24	0:11	$\times$
$1 \rightarrow 2$	17:32	1:37	$\surd$
$2 \rightarrow 3$	08:48	1:04	$\surd$
$3 \rightarrow 4$	14:38	1:40	$\times$
$4 \rightarrow 5$	02:42	0:09	$\surd$
$5 \rightarrow 6$	12:02	1:16	$\times$
$6 \rightarrow 7$	02:54	0:18	$\times$
$7 \rightarrow 8$	09:19	1:03	$\times$
$8 \rightarrow 9$	07:15	0:56	$\surd$
$9 \rightarrow 10$	10:26	1:05	$\surd$
$10 \rightarrow 11$	10:49	0:58	$\surd$
$11 \rightarrow 12$	08:51	0:37	$\surd$
$12 \rightarrow 13$	09:52	1:19	$\surd$
$13 \rightarrow 14$	15:25	1:18	$\surd$
$14 \rightarrow 15$	08:24	0:47	$\surd$
$15 \rightarrow 16$	10:06	1:19	$\times$
$16 \rightarrow 17$	21:30	2:18	$\surd$
$17 \rightarrow 18$	12:13	1:19	$\surd$
$18 \rightarrow 19$	13:31	1:16	$\surd$
$19 \rightarrow 20$	21:00	1:40	$\times$
$20 \rightarrow 21$	29:22	2:24	$\times$
$21 \rightarrow 22$	57:40	5:33	$\surd$
$22 \rightarrow 23$	72:57	5:42	$\times$
$23 \rightarrow 24$	28:21	2:60	$\times$
$24 \rightarrow 25$	06:53	0:40	$\times$
Average	18:13	1:44	14 / 25

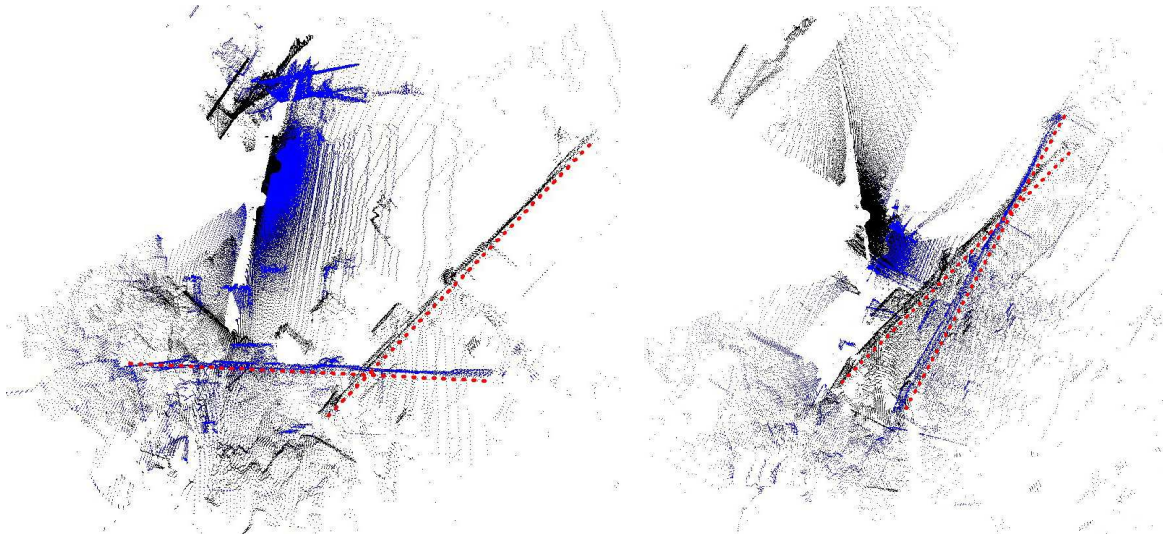
Table 1: ICP performance on the crashed car park data set. $\surd$ - succeeded, $\times$ - failed



(a) reg. scans 16-17, front view

(b) reg. scans 15-16, perspective view

Figure 6: An example of a successful pairwise registration by ICP of scan 16 (blue) and 17 (black). Note that the robot mainly made a forward motion.



(a) failed reg. scans 6-7, top view

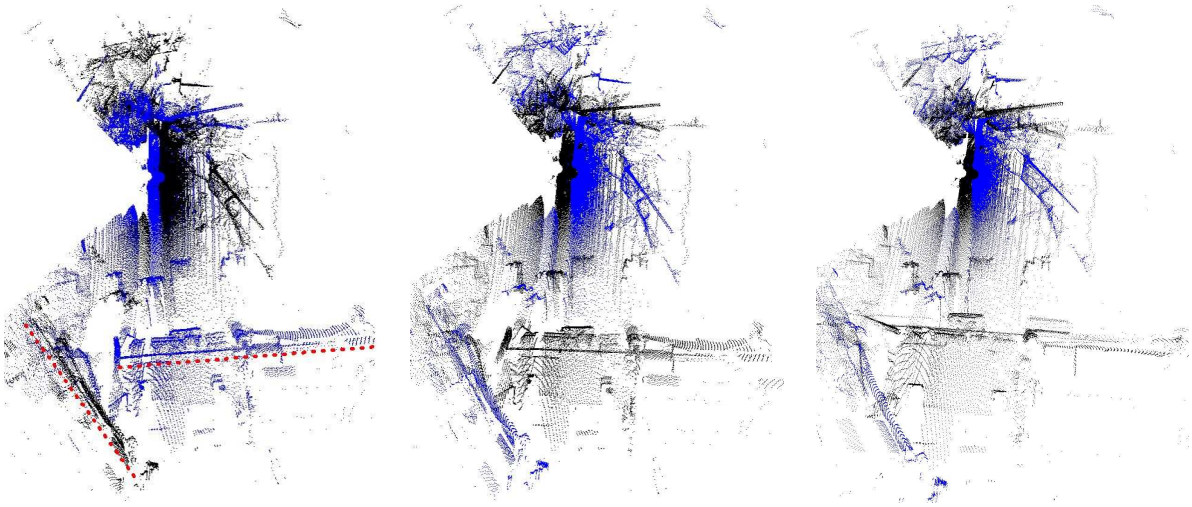
(b) failed reg. scans 15-16, top view

Figure 7: Two examples of unsuccessful pairwise registrations by ICP: scan pairs 6 (blue),7 (black), and 15 (black),16 (blue). The misaligned façade – indicated by dashed red lines – is clearly recognizable in this top view. ICP seemingly could not cope with the larger rotations of the robot in these cases.

shown in Fig. 6. Typical failed registrations are shown in Fig. 7.

In total, 11 out of 25 ICP registrations are clearly unsuccessful. It is notable that the failures occur when the robot made relatively large rotations. When the robot mainly made a forward motion, ICP performs well. It can be intuitively explained as follows. Given a point $\mathbf{p}_A$ in point cloud A, let its nearest neighbor in point cloud B be p_B. The likelihood that $\mathbf{p}_A$ and $\mathbf{p}_B$ physically correspond in the real world is much higher if A and B are just transformed due to a translation. In case there is a rotation as well, the likelihood that the nearest neighbors are actual correspondences is much smaller.

In this context, the lack of proper odometry is of importance. Given good initial estimates, especially for the rotation, ICP is known to perform better. ICP has to cope with rather large motions in this data-set, which also affects its run-time. The Disaster City scenario with significant amounts of rubble and dust renders odometry – like in most unstructured environments – completely useless. It is to be noted that the usage of odometry information in the ICP experiments as initial guesses for the iterations does not improve the results. On the contrary, it significantly worsens the outcomes. Not a single registration is successful when the recorded odometry data is used in the ICP. This is because the ICP registrations are successful when the robot mainly moves forward with some minor rotations. But due to slip, odometry typically perceives larger changes in orientation at many times. This again leads to virtual rotations, hence worsening the starting conditions for ICP when odometry is used.



(a) no subsampling (b) subsampled by 3 (c) subsampled by 10

Figure 8: When the ICP registration fails, it leads to similar wrong results, independent of subsampling. The situation is illustrated above with the results for the failed registration of scans 4 and 5 with subsampling by 3 and 10, and with no subsampling respectively.

Last but not the least, on selected scans where the ICP registration failed with subsampling, we tested whether the usage of the full data in a scan would improve the result. The run-time per registration increased by a factor of about 78 compared to subsampling by 3. But the qualitative result remained the same as illustrated in Fig. 8. The misalignment is always very similar, irrespective of whether subsampling by 3, 10, or no subsampling is used. The

qualitative similarity of the results is also indicated by the estimated robot paths shown in Fig. 9.

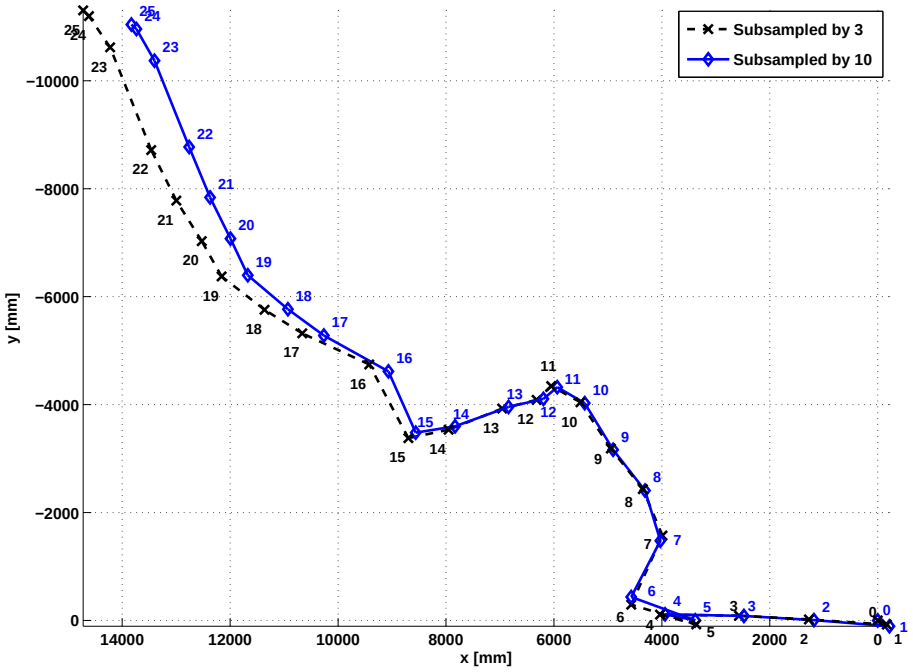


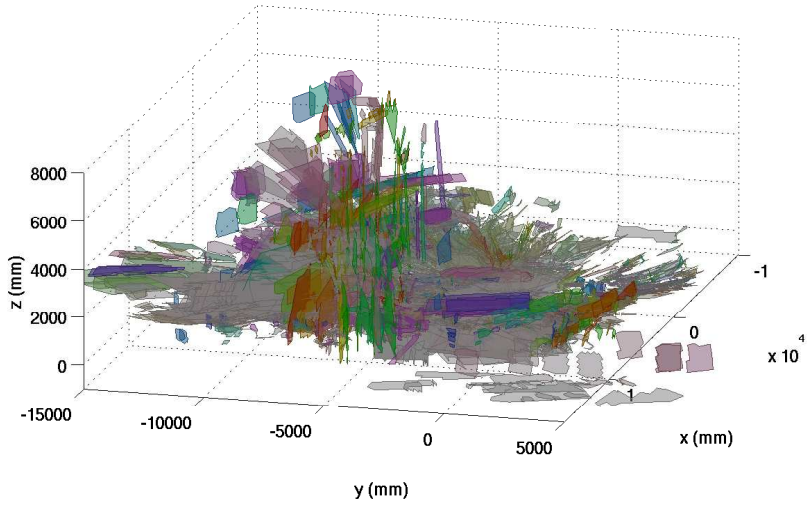
Figure 9: The robot path according to the ICP registrations of the scans. The results for the different amounts of subsampling are very similar.

For the sake of completeness, a 3D map based on recorded odometry only is shown in Fig. 10. It is clearly unusable. The plane extraction method from [Poppinga et al., 2008] is used as the representation basis of the scans for illustration purposes only; the corresponding point cloud representation is not suited for visualization due to its lack of structure.

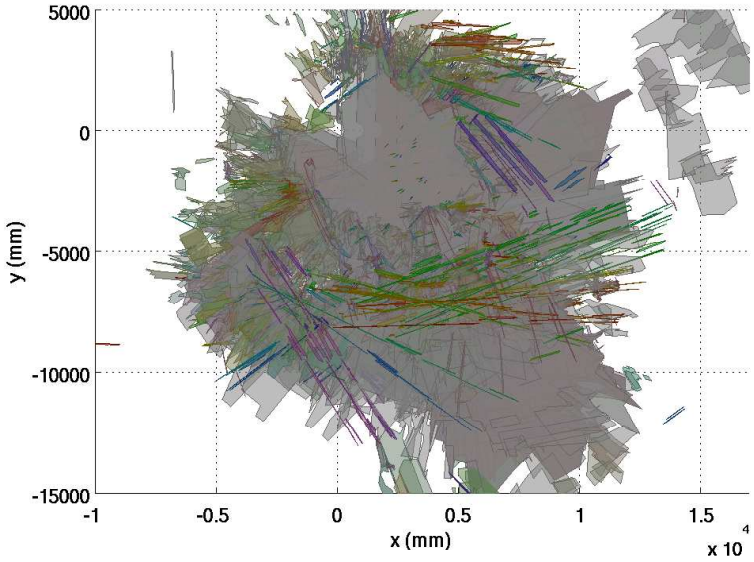
Figure 11 shows the 3D map as result of ICP registrations. Some structures are recognizable. It is a significant improvement over the odometry map shown in Fig. 10. But overall, the map has significant errors. The façade appears in several different locations in the map, the lower floor under which the robot drove is misrepresented, and parts of the large rubble pile next to the scenario are distributed all over the map. The computation time to generate the whole map with subsampling by 10 is 39:28 minutes and it is about 7 hours for subsampling by 3.

3.3 Results of the Plane-Matching Registration

The plane matching registration performs on the same data-set better than ICP in terms of processing speed as well as robustness. Table 2 summarizes the results. The values are for an AMD Turion 2 1.6 GHz with 960 MB RAM running OpenSUSE 10.3. The computation time is on average 2.68 seconds for the plane extraction and 5.42 seconds for the plane

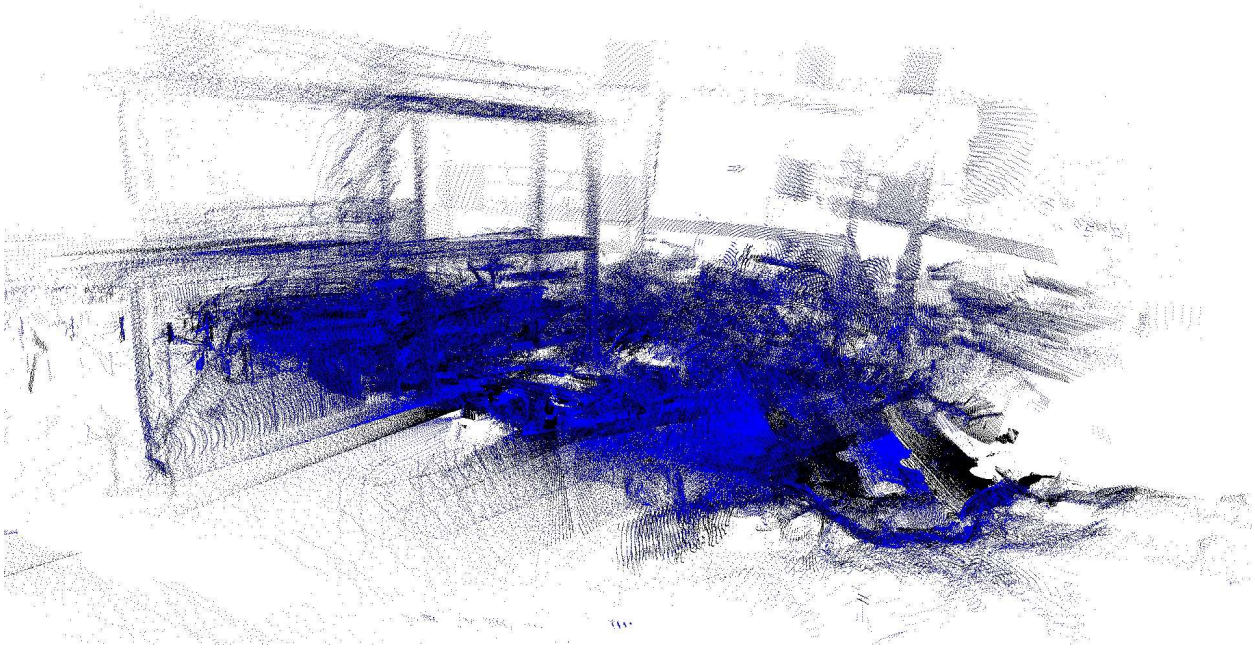


(a) Perspective View

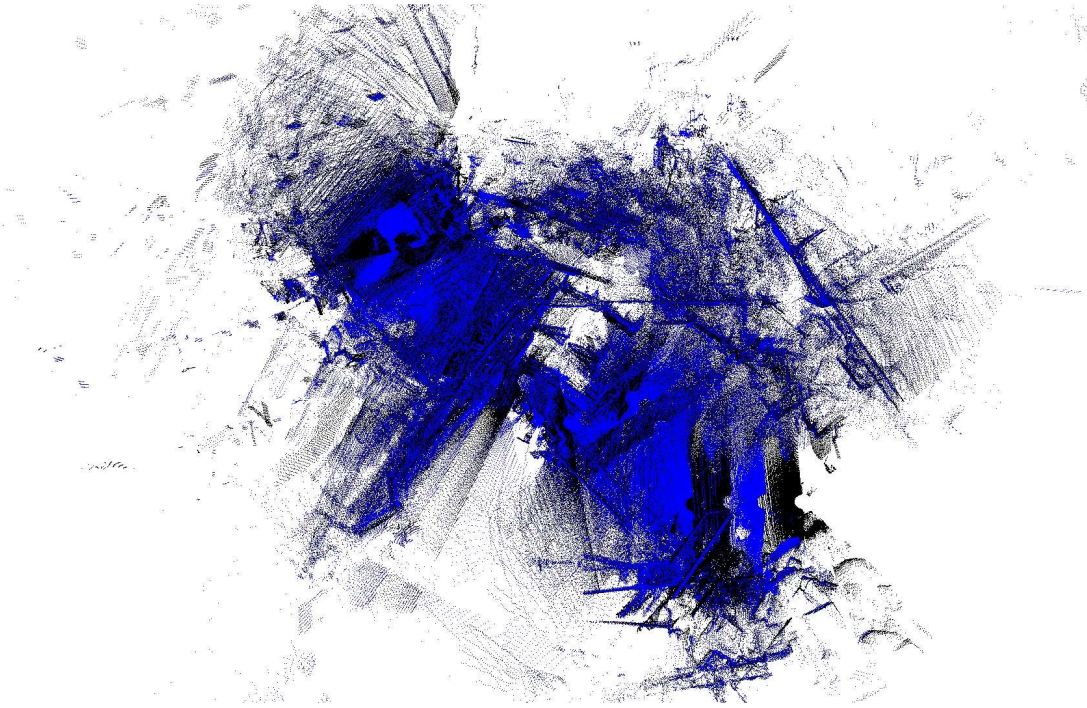


(b) Top View

Figure 10: A 3D "map" created using only odometry. The plane based representation of the scans is used for illustration purposes only. The planes are colored according to their normal directions for easier visualization, and the colors do not carry any other semantic meaning. Such a map is clearly not usable.



(a) perspective view



(b) top view

Figure 11: The 3D map of the collapsed car park as result of ICP registration. Some structures are roughly recognizable. But overall, the map contains gross errors.

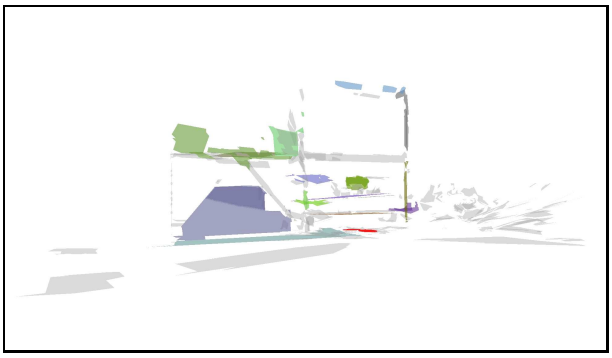
Scan k	Extraction (sec.)	Polygon. (sec.)	Registration k→k+1 (sec.)	success	Total (sec.)
0	2.48	3.38	4.34	√	10.20
1	2.47	1.85	4.85	√	9.17
2	2.46	2.07	2.39	√	6.92
3	2.40	1.94	2.60	√	6.94
4	2.55	2.91	4.87	√	10.33
5	2.49	2.17	5.61	√	10.27
6	2.40	2.08	3.66	√	8.14
7	2.32	1.53	3.20	√	7.05
8	2.53	1.70	5.44	√	9.67
9	2.60	1.51	7.02	√	11.13
10	2.51	1.30	13.64	√	17.45
11	2.59	1.95	13.82	√	18.36
12	2.54	1.61	7.68	√	11.83
13	2.56	2.02	6.51	√	11.09
14	2.72	1.88	9.23	√	13.83
15	2.63	1.86	8.31	√	12.80
16	2.65	2.30	7.48	√	12.43
17	2.67	2.29	6.57	√	11.53
18	2.96	2.74	2.86	√	8.56
19	2.90	4.19	2.85	√	9.94
20	2.91	4.28	2.95	√	10.14
21	3.00	3.91	1.97	√	8.88
22	3.09	2.48	2.18	√	7.75
23	3.00	3.21	2.86	√	9.07
24	3.12	4.08	2.59	√	9.79
25	3.09	4.25	–	–	–
Average	2.68	2.52	5.42	25/25	10.62

Table 2: The performance of the plane matching registration.

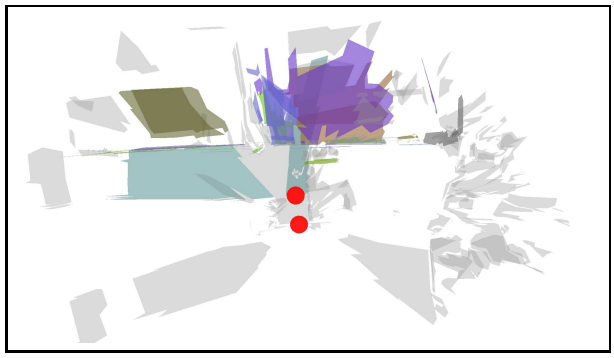
registration, i.e., 8.1 seconds in total for the registration of two scans. In case a plane based visualization instead of point clouds is used, additional computation of the polygonization of the boundaries of the plane patches is required. This takes 2.52 seconds, i.e., the overall computation time is then 10.62 seconds. This is – without any subsampling – significantly faster than ICP, which requires on an average 18:13 and 1:44 minutes with subsampling by factor 3 and factor 10 respectively.

The aspect even more crucial than the run-time is the robustness of the plane matching registration. The pairwise registrations by plane matching were manually inspected using the same criteria as for the results of ICP. None of the plane matching registrations showed any of the gross failures that could be observed in 11 cases with ICP. Figure 12 shows several examples of the registrations. Not only did the cases where the robot only moved forward worked out well, e.g., scan pair 16-17, but the cases with larger amounts of rotation in the

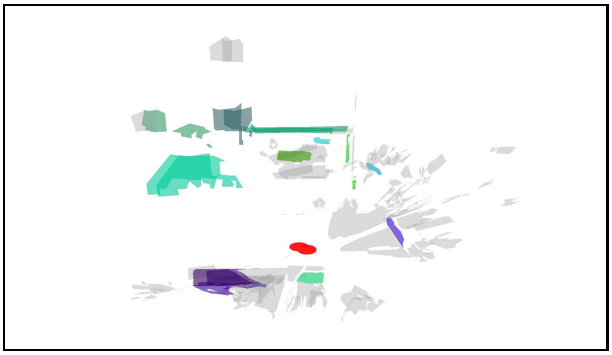
robot's motion worked out well also, e.g., scan pairs 6-7 and 15-16, in which ICP failed (see also Fig. 7).



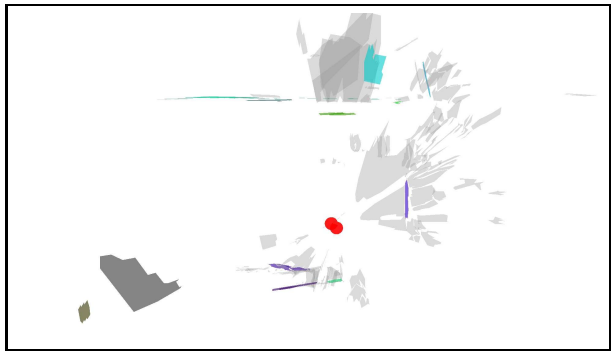
(a) scan pair 16-17 (ICP: √), front view



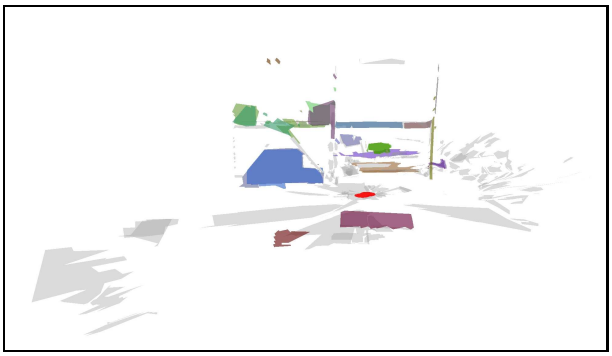
(b) scan pair 16-17 (ICP: √), top view



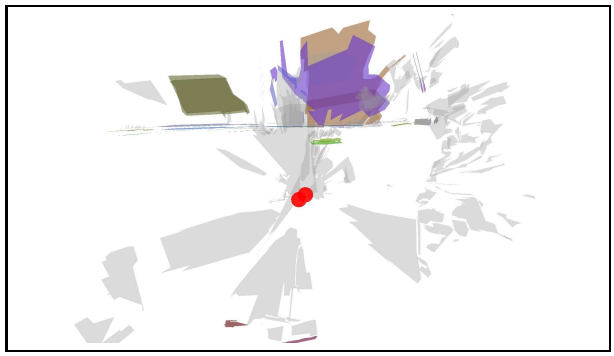
(c) scan pair 6-7 (ICP: ×), front view



(d) scan pair 6-7 (ICP: ×), top view



(e) scan pair 15-16 (ICP: ×), front view



(f) scan pair 15-16 (ICP: ×), top view

Figure 12: Examples of the successful pairwise registrations by the plane based approach. ICP performs also well on the pair 16-17 (see also Fig. 6), but it fails, for example, on pair 6-7 and 15-16 (see also Fig. 7).

Figure 13 shows the resulting 3D map of the collapsed car park scenario. The overall structure can be well recognized. It is a significant improvement over the odometry and ICP based maps (Figs. 10 and 11). Though the 3D map represents the scene quite well, it is not perfect. The remaining errors in this map are presented and discussed in some detail in Sec. 5. They consist of certain corresponding planes that have some translational offsets parallel to the planes themselves. This occurs because of an occasional lack of prominent planes to match along certain directions, and sometimes due to the determination of incorrect

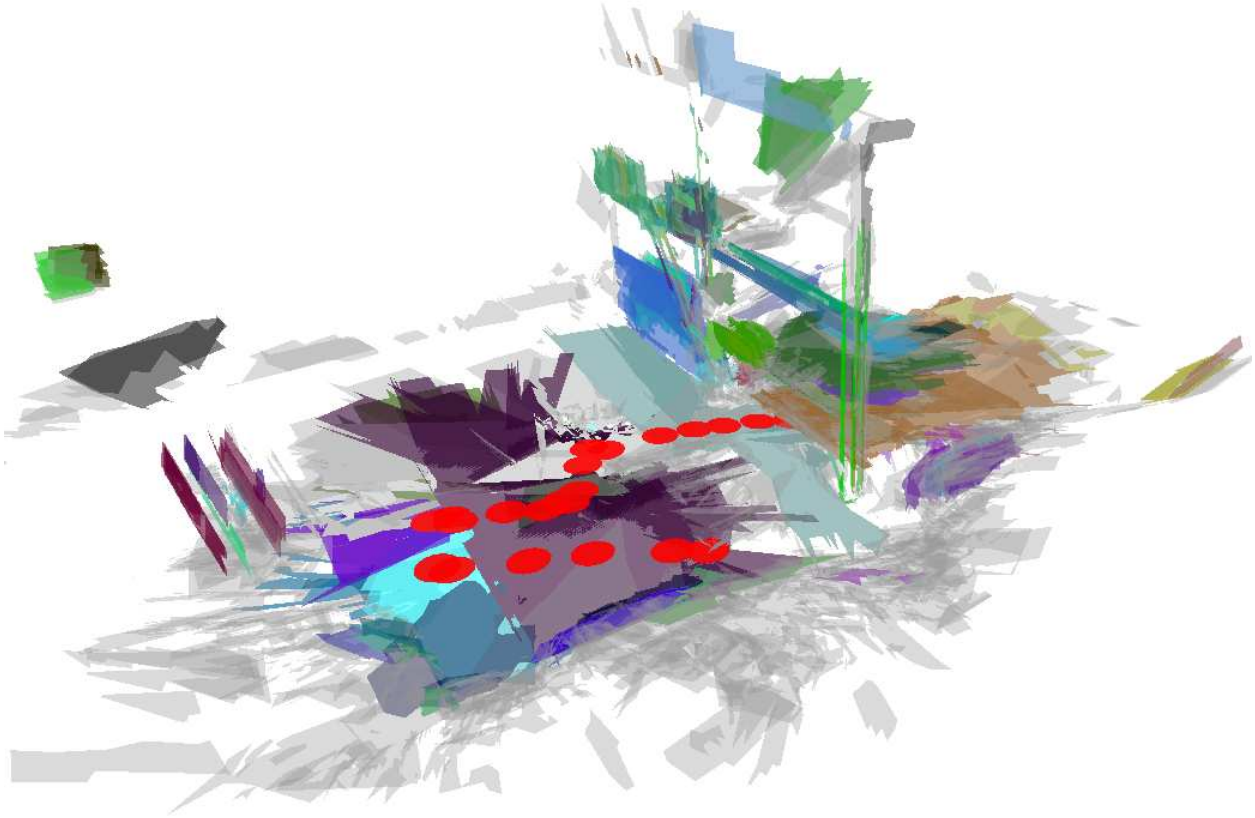


Figure 13: The 3D map created using the plane registration. Matched planes are drawn in the same color and unmatched planes are grayed out. The estimated robot positions are shown by red circles.

correspondences when the scenes have several planar patches lying within the same infinite plane. However, the fact that corresponding planes remain parallel shows that the rotation was accurately determined; numerical data to back up this claim is also presented in Sec. 5.3.

A closed-form pose-graph relaxation for 3D-Plane-SLAM is introduced in the next Sec. 4. As shown in Sec. 5, the 3D-Plane-SLAM further improves the the quality of the 3D map in the collapsed parking lot scenario. The performance of 3D-Plane-SLAM, i.e., plane based registration and pose-graph relaxation, is then further evaluated with respect to computation speed and robustness in Sec. 6, where a second data set from another Disaster City scenario is used.

4 Fast Relaxation for 3D-Plane-SLAM

4.1 Pose Graph Maps

The planar surface registration is now embedded into a 3D SLAM approach, which – as shown later on in sections 5 and 6 by experiments in the two disaster scenarios – is extremely fast and improves the map quality. First, our notation is introduced. The robot takes 3D scans from several locations $i = 0 \dots \mathsf{N}$. Each location defines a robot-attached frame $\mathcal{F}_i$ with origin $\mathcal{O}_i$. For each pair of adjacent frames $\mathcal{F}_i$ and $\mathcal{F}_{i+1}$, planes are extracted from the point-clouds and matched, thus providing the pose registration between these frames. These matches are written $i \rightarrow i+1$.

It is sometimes also possible to match certain *non-adjacent* frames $\mathcal{F}_i$, $\mathcal{F}_j$ such that $|j-i| > 1$. These are called loop-closing matches, and are written in reverse order as $j \rightarrow i$, where $j > (i+1)$. The pose-registration computed by plane-matching consists of a relative rotation ${}^j_i\mathbf{R}$ and translation ${}^j_i\mathbf{t} \triangleq \overrightarrow{\mathcal{O}_j\mathcal{O}_i}$, resolved in frame $\mathcal{F}_j$.

A general consensus on efficient data structures for 3D map representations does not exist. Many representations, such as plain point clouds in kd-trees [Nuechter et al., 2004], 3D occupancy grids in oct-trees [Surmann et al., 2003, Nüchter et al., 2007], or surface models as features in an extended Kalman filter [Weingarten and Siegwart, 2006], are in use in current literature. A graph-based representation called pose-graph [Olson et al., 2006a, Pfingsthorn et al., 2007, Pfingsthorn and Birk, 2008, Takeuchi and Tsubouchi, 2008] is used here, although other similar graph data-structures are in use as well [Kohlhepp et al., 2007]. The pose graph consists of nodes $\mathcal{N}_i$, $i = 0 \dots \mathsf{N}$, and directed edges ${}^j_i\mathcal{E}$, $i \neq j$, also denoted as $j \rightarrow i$. Each node $\mathcal{N}_i$ corresponds to a sensor frame $\mathcal{F}_i$ with origin $\mathcal{O}_i$. Each directed edge ${}^j_i\mathcal{E}$ contains the relative rotation ${}^j_i\mathbf{R}$ and translation ${}^j_i\mathbf{t}$ along with their estimated covariance.

4.2 Relaxation

After a successful loop detection through spatial proximity to a previously visited node, the map has to be relaxed to take advantage of the newly available information. The most general formulation relaxes both rotation as well as translation errors. However, even in relatively general approaches formulated in 2D, some simplifications of the rotational coupling are done

to speed up the computation, e.g., by neglecting the contribution of the rotation matrix in the computation of Jacobians [Olson et al., 2006b, Grisetti et al., 2007].

As shown earlier, our plane matching can decouple the determination of rotation and translation components. Additionally, only two pairs of non-parallel corresponding planes need to be found to determine the rotation registration between two samples, whereas for translation, this number is three. Typically, for a large FOV sensor, the number of high-evidence corresponding plane pairs found between two samples is around 10. Therefore, the rotation is in general more accurately determined than translation in plane-matching. This can be exploited for pose-graph SLAM by relaxing only the translation error. Since this part is linear, a fast non-iterative closed-form least-squares solution can be obtained. A similar approach was taken in 2D by [Duckett et al., 2002], where ignoring rotation relaxation was justified by assuming the availability of a global orientation sensor, e.g. a compass. In our work, the justification is based on the rotational accuracy of the plane-matcher.

In addition to this theoretical motivation, this property can also be experimentally observed. As discussed in more detail in sections 5.3 and 6.3, tables 4 and 5 show the rotation errors that are determined through loop closing in the two disaster scenarios. The orientation differences between node i and j are once shown based on the propagated changes through the path from i to j, i.e., $i \rightarrow (i+1) \rightarrow ... \rightarrow j$ (Cumul.), and once based on the loop closing result, i.e., the registration $j \rightarrow i$ (Direct). The differences are negligible as indicated by the small χ^2 distances computed based on the covariance matrix of the direct registration.

Proceeding as in [Lu and Milios, 1997], the translation ${}^{j}_{i}\mathbf{t}$ can be thought of as a random vector with mean ${}^{j}_{i}\bar{\mathbf{t}}$ and the inverse of its 3×3 covariance matrix being denoted as ${}^{j}_{i}\mathbf{C}^{-1}_{\mathbf{tt}}$. This covariance comes from the plane-matcher. If the scans corresponding to two nodes are not matchable, we set the inverse of the covariance ${}^{j}_{i}\mathbf{C}^{-1}_{\mathbf{tt}} \equiv \mathbf{0}$.

To relax the graph we need to minimize the cost

$$\min_{{}^{j}_{i}\mathbf{t}} \sum_{j=0}^{\mathsf{N}} \sum_{i=j+1}^{\mathsf{N}} \left({}^{j}_{i}\mathbf{t} - {}^{j}_{i}\bar{\mathbf{t}}\right)^{\mathsf{T}} {}^{j}_{i}\mathbf{C}^{-1}_{\mathbf{tt}} \left({}^{j}_{i}\mathbf{t} - {}^{j}_{i}\bar{\mathbf{t}}\right) \tag{17}$$

with the additional constraints on ${}^{j}_{i}\mathbf{t}$ for all possible independent loops containing it. As ${}^{j}_{i}\mathcal{E}$ and ${}^{i}_{j}\mathcal{E}$ contain the same information, only one is considered in the summation – which is considered, depends on the direction the graph is traversed.

Since unconstrained optimization is easier to solve, we reformulate the problem by resolving all quantities with respect to the global frame – since the rotations are considered known and certain, this is possible. The position of $\mathcal{N}_i$ resolved in global coordinates is denoted by ${}^{g}_{i}\mathbf{t}$. Furthermore,

$${}^{j}_{i}\bar{\mathbf{g}} \triangleq {}^{g}_{j}\mathbf{R}\ {}^{j}_{i}\bar{\mathbf{t}} \tag{18}$$

$${}^{j}_{i}\mathbf{\Sigma}^{-1}_{\mathbf{tt}} \triangleq {}^{g}_{j}\mathbf{R}\ {}^{j}_{i}\mathbf{C}^{-1}_{\mathbf{tt}}\ {}^{g}_{j}\mathbf{R}^{\mathsf{T}} \tag{19}$$

Then the previous cost function can be made unconstrained as follows

$$\min_{{}^{g}_{\star}\mathbf{t}} \sum_{j=0}^{\mathsf{N}} \sum_{i=j+1}^{\mathsf{N}} \left({}^{g}_{i}\mathbf{t} - {}^{g}_{j}\mathbf{t} - {}^{j}_{i}\bar{\mathbf{g}}\right)^{\mathsf{T}} {}^{j}_{i}\mathbf{\Sigma}^{-1}_{\mathbf{tt}} \left({}^{g}_{i}\mathbf{t} - {}^{g}_{j}\mathbf{t} - {}^{j}_{i}\bar{\mathbf{g}}\right)$$

We further define

$$\mathbf{x} \triangleq \begin{bmatrix} {}^{g}_{1}\mathbf{t} \\ \vdots \\ {}^{g}_{\mathsf{N}}\mathbf{t} \end{bmatrix} \in \mathbb{R}^{3\mathsf{N}}, \tag{20}$$

and also,

$${}^{j}_{i}\mathbf{I} \triangleq \begin{bmatrix} \mathbf{0}_{3\times3} & \cdots & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \cdots \end{bmatrix} \tag{21}$$

In the above, $\mathbf{I}_3$ is the identity matrix, $-\mathbf{I}_3$ appears in the jth 3×3 block of ${}^{j}_{i}\mathbf{I}$, and $\mathbf{I}_3$ appears in the ith block. Note that the size of ${}^{j}_{i}\mathbf{I}$ is $3 \times 3\mathsf{N}$. The cost function can again be rewritten as

$$\min_{\mathbf{x}} \sum_{j=0}^{\mathsf{N}} \sum_{i=j+1}^{\mathsf{N}} \left({}^{j}_{i}\mathbf{I}\,\mathbf{x} - {}^{j}_{i}\bar{\mathbf{g}}\right)^{\mathsf{T}} {}^{j}_{i}\mathbf{\Sigma}^{-1}_{\mathbf{tt}} \left({}^{j}_{i}\mathbf{I}\,\mathbf{x} - {}^{j}_{i}\bar{\mathbf{g}}\right) \tag{22}$$

Differentiating the above by $\mathbf{x}$, setting the resulting expression to zero and solving for $\mathbf{x}$ gives

$$\mathbf{G}\,\mathbf{x} = \mathbf{b}, \text{where,} \tag{23}$$

$$\mathbf{G} \triangleq \sum_{j=0}^{\mathsf{N}} \sum_{i=j+1}^{\mathsf{N}} {}^{j}_{i}\mathbf{I}^{\mathsf{T}}\, {}^{j}_{i}\mathbf{\Sigma}^{-1}_{\mathbf{tt}}\, {}^{j}_{i}\mathbf{I}, \qquad \mathbf{b} \triangleq \sum_{j=0}^{\mathsf{N}} \sum_{i=j+1}^{\mathsf{N}} {}^{j}_{i}\mathbf{I}^{\mathsf{T}}\, {}^{j}_{i}\mathbf{\Sigma}^{-1}_{\mathbf{tt}}\, {}^{j}_{i}\bar{\mathbf{g}}. \tag{24}$$

After solving for $\mathbf{x}$, we can back-calculate

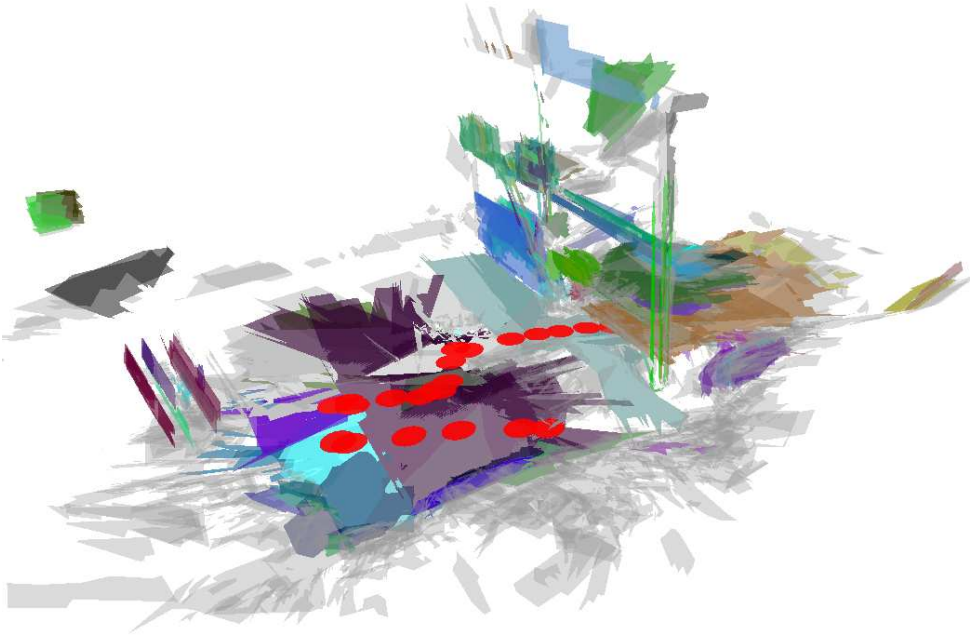
$${}^{j}_{i}\mathbf{t} = {}^{g}_{j}\mathbf{R}^{\mathsf{T}}\left({}^{g}_{i}\mathbf{t} - {}^{g}_{j}\mathbf{t}\right). \tag{25}$$

5 3D-Plane-SLAM in the Collapsed Parking-Lot Scenario

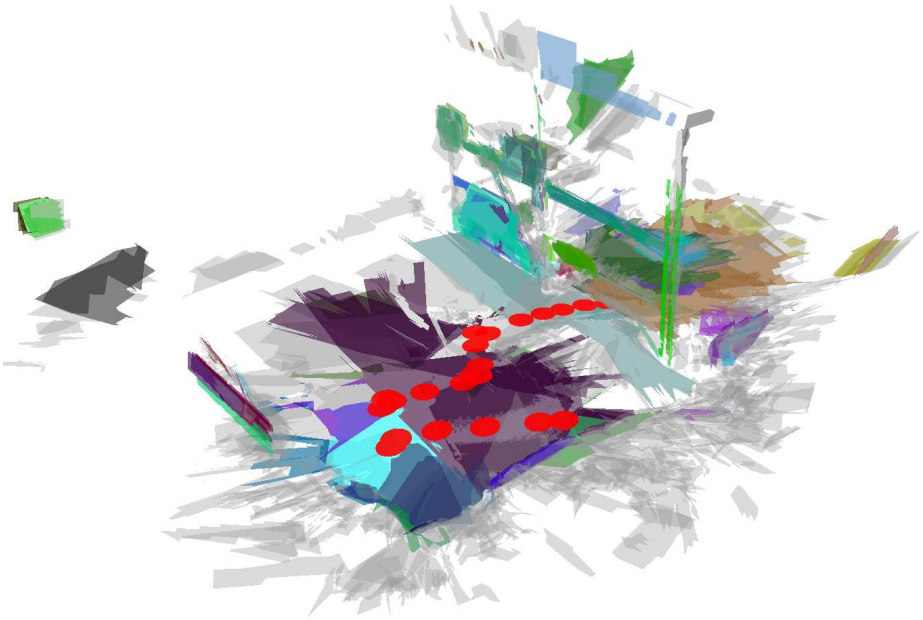
5.1 Loop Closing and Closed Form Pose Graph Relaxation

As previously described in Sec. 3, in contrast to ICP and using just odometry, the plane based registration leads to a meaningful 3D map of the collapsed car parking scenario. The map (Figs. 14(a) and 15(b)) captures the overall structure of the scene. But there is still room for improvement. Some corresponding planes have translational offsets parallel to the planes themselves, some examples are shown in Fig. 18(c). These errors occur in case of a lack of prominent planes to match along certain directions and sometimes due to the determination of incorrect correspondences when the scenes have several planar patches lying within the same infinite plane. The errors are relatively small, at least compared to the cases of unsuccessful registrations of ICP (Sec. 3, Fig. 7), but they nevertheless accumulate. It is hence of interest to use proper SLAM. An experimental evaluation of the 3D-Plane-SLAM introduced in the previous Sec. 4, i.e., the embedding of the plane based registration into a pose graph with closed-form relaxation, is presented in this section. The relaxation is very fast and it leads to quantitative as well as qualitative improvements of the results.

Figure 16 shows the computed rotation of the robot along its path, and Fig. 17 shows the estimated robot paths for plane-registration only and for full 3D-Plane-SLAM. The latter figure also shows the numbering of the pose estimates. Proximity based loop detection is used. It leads to a total of 8 loops that can be successfully matched, out of which 7 are short, consisting of 3 nodes each. There is one large loop $12 \rightarrow 2$, consisting of 10 nodes.

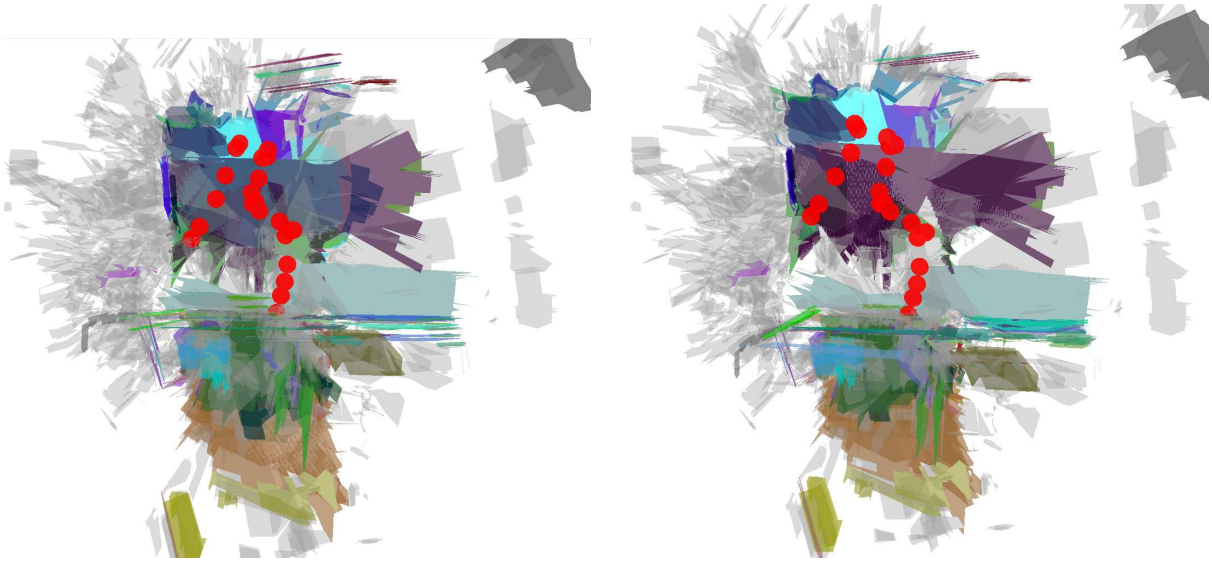


(a) Perspective view before relaxation. The rubble pile on the figure-bottom consists of small gray unmatched planes (see also Fig. 1(a)). Note the misalignment of the car-chassis back (see also Fig. 18) and the planes on the left of the figure. All the misalignments are parallel to the planes, indicating that the rotation error is negligible.



(b) Perspective view after relaxation. Note the good alignment of planes parallel to the façade.

Figure 14: Perspective view showing the improvement from relaxation. The robot position is shown by red circles. Matched planes are shown with the same color and unmatched planes are grayed out. A video of the point-cloud model is available on the website given in Sec. 7.3.



(a) Top view before relaxation.

(b) Top view after relaxation

Figure 15: Top views of the map, before and after relaxation.

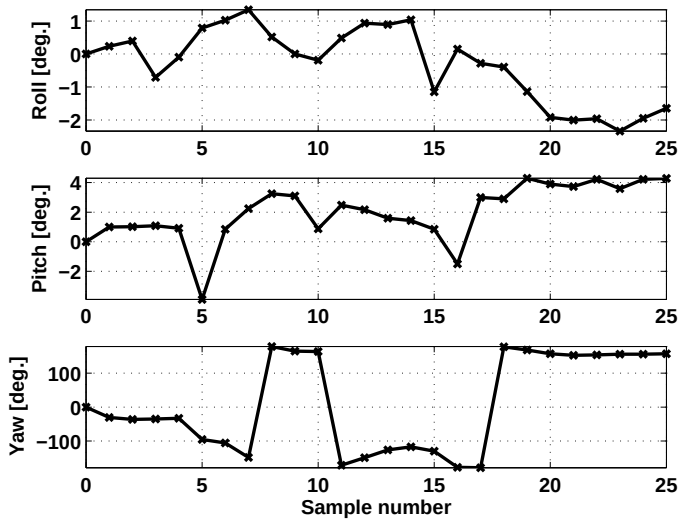


Figure 16: The robot rotation trajectory. Note that the apparent jumps in the yaw are simply the wrapping of angle value from $-180°$ to $180°$. The rotation was not relaxed.

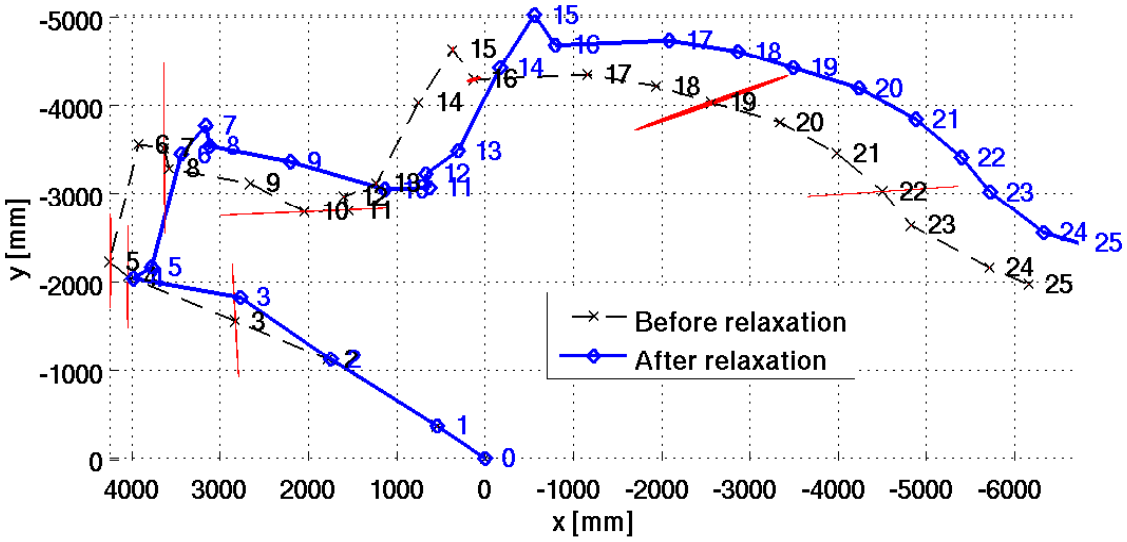
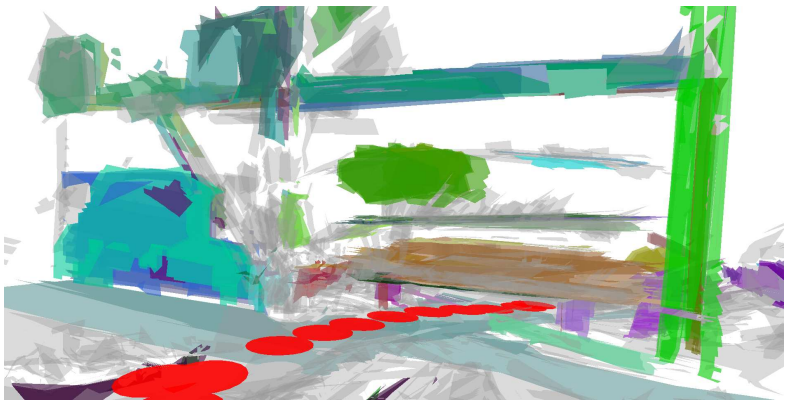


Figure 17: The robot path before and after relaxation. In the unrelaxed path, the 10σ uncertainty ellipses are plotted in red for pair-wise matching result. This shows that most of the time there is only one principal direction of uncertainty. The motion along the z direction is quite small; it amounts to about 200 mm in total.

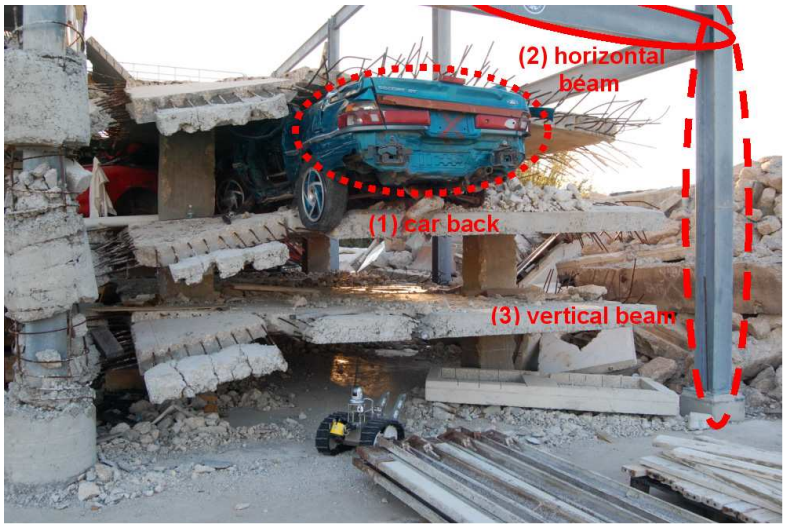
5.2 Computation Time and Map Improvement

The results of applying the algorithm described in Sec. 4 to the 3D pose-graph is shown in Figs. 14(b) and 15(b), where the whole map is shown in perspective and top view. The results are also shown in a close-up in Fig. 18(d), which should be compared with the camera photographs of Fig. 18(b). The reader can immediately locate various ground-truth structures and notice that their alignment has improved significantly after relaxation, i.e., there is a clear qualitative improvement in the map. There is also a quantitative improvement in terms of reduced uncertainty: the accumulated Mahalanobis distance is significantly reduced by the relaxation, as can be seen in Table 3. The table also gives the run-times for the various steps involved: plane-extraction, plane-matching, polygonization, and relaxation.

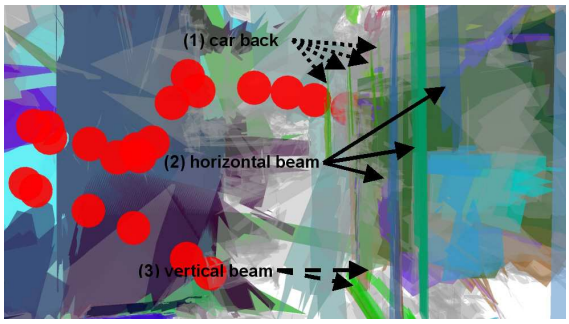
Compared to the state of the art in 3D mapping, the approach can be considered to be fast. This holds especially from a practical viewpoint, since in comparison to the previously mentioned total time $T_{\text{scan}} \approx 32$ seconds for taking one scan: the core registration, i.e., mean extraction and plane-matching, takes $0.25\,T_{\text{scan}}$. When adding the polygonization step, which is needed mainly for visualization, the total time of mean extraction, plane-matching, and polygonization is $0.33\,T_{\text{scan}}$. Hence, It is easy to process a registration of the previous two scans while a new one is being acquired. Currently, a speed-up of the data acquisition by at least two orders of magnitude is possible with more costly off-the-shelf equipment. Even then, the approach is still very well suited for online processing. The approximately 10 seconds that it takes to register two scans are less than the time it takes the robot to



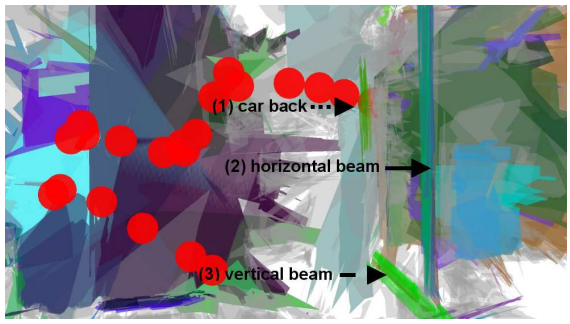
(a) Closeup view showing the transition of the robot from outside the structure to underneath the collapsed floor.



(b) Ground-truth structures.



(c) Zoomed-in top-view before relaxation



(d) Zoomed-in top-view after relaxation

Figure 18: There are several ground-truth structures that can be easily identified in the planar models before and after relaxation.

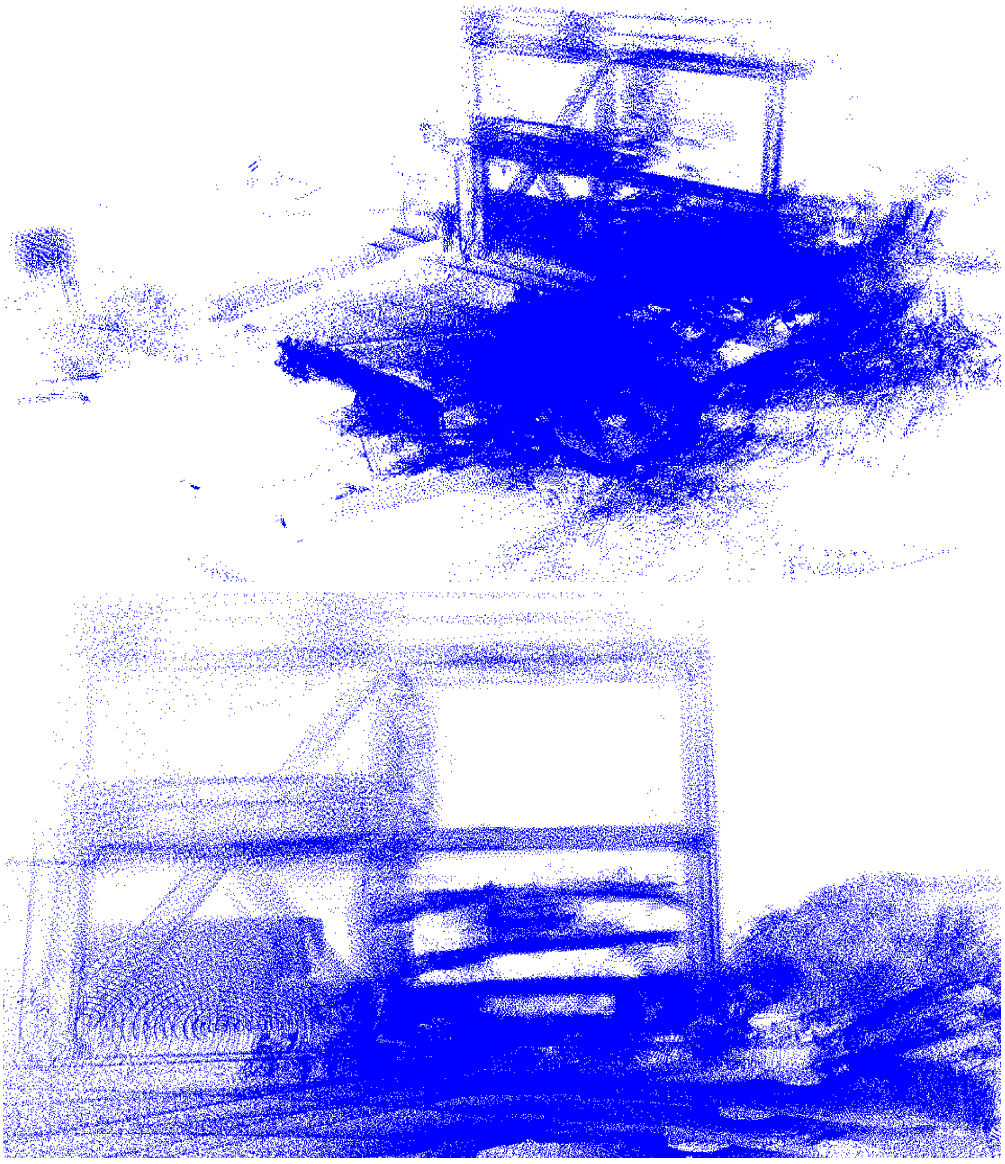


Figure 19: The final 3D map as point cloud. The point-clouds have been aligned based on the results of plane-matching registration and loop-closing for 26 scans. As can be seen, there is a higher level of detail but the points are too dense in several regions for a meaningful visualization.

Table 3: Mean statistics for 3D-Plane-SLAM on the 26 scans of the Collapsed Parking Lot scenario

Planes-extraction per scan	2.68	sec.
Nr. of planes unfiltered per scan	99	
Nr. of planes after filtering per scan	19	
Plane-matching and registration per scan-pair	5.42	sec.
Polygonization per scan	2.52	sec.
Time for relaxation of pose-graph with 26 nodes and 33 edges	0.01	sec.
Mahalanobis distance in Eq. (22) after/before relaxation	$869724/(2.36 \times 10^{10})$	

move from one scan spot to the next one under the challenging locomotion conditions of this scenario.

The 3D SLAM proper, i.e., loop closing and relaxation, comes almost for free with the plane based registration. Due to its angular precision, the relaxation can be restricted to translations only, allowing for a closed-form solution that can be computed in a few milliseconds. The SLAM improves the map as shown in Figs. 14(b) and 18(d). As discussed in more detail in Sec. 7, the 3D map can still be displayed based on the collected point-clouds, e.g., to see a high level of detail. This is shown in Fig. 19, which was created by applying the pose-registration computed by plane-matching and relaxation to the point-clouds. The figure also illustrates the difficulty of visualization of large point-clouds.

5.3 The Accuracy of Rotation Estimates

To experimentally validate the claim that the plane-matching leads to accurate rotation results, the rotation computations for the loop-closing edges are listed in Table 4. As an example, for the loop $12 \rightarrow 2$, the row marked "Direct" shows the result of matching $\mathcal{F}_{12}$ with $\mathcal{F}_2$ directly. The row marked "Cumul." shows the result of finding the relative transform between $\mathcal{F}_{12}$ and $\mathcal{F}_2$ by cumulatively computing the relative rotation by traversing all the intermediate edges as follows

$$ {}^{2}_{12}\mathbf{R} = {}^{2}_{3}\mathbf{R}\,{}^{3}_{4}\mathbf{R} \ldots {}^{10}_{11}\mathbf{R}\,{}^{11}_{12}\mathbf{R} \tag{26} $$

It can be seen that the maximum difference is 2.5° for roll and that the yaw angles are quite close. More importantly, the Mahalanobis distance χ^2 based on the covariance matrix of the direct transform is quite small, which shows that the difference is not statistically significant.

The high accuracy of rotational estimates is also supported by theory and experiments presented in [Pathak et al., 2009b] where the plane registration is introduced. Additional relaxation of the rotations requires non-linear optimizations based on computationally more costly iterative methods. The relaxation method of 3D-Plane-SLAM is therefore not suited for the pose graphs generated by ICP registrations; this holds at least for the two data-sets used in this article. Also the loop detection is critical in this respect. The simple proximity based method in 3D-Plane-SLAM requires reasonable initial 3D maps from registration, which is not the case for the ICP registrations in the two Disaster City scenarios.

Loop	Type	Roll	Pitch	Yaw	χ^2 dist.
$3 \rightarrow 1$	Direct	$0.612°$	$0.340°$	$4.197°$	
	Cumul.	$0.857°$	$-0.136°$	$4.475°$	$1.11\,10^{-4}$
$8 \rightarrow 6$	Direct	$-2.258°$	$0.561°$	$76.026°$	
	Cumul.	$-2.253°$	$0.553°$	$75.879°$	$6.575\,10^{-6}$
$10 \rightarrow 8$	Direct	$1.164°$	$1.741°$	$14.73°$	
	Cumul.	$0.474°$	$2.358°$	$14.870°$	$2.668\,10^{-4}$
$12 \rightarrow 2$	Direct	$-3.760°$	$1.595°$	$112.708°$	
	Cumul.	$-1.238°$	$2.717°$	$112.79°$	$2.324\,10^{-3}$
$14 \rightarrow 12$	Direct	$0.927°$	$0.284°$	$-32.166°$	
	Cumul.	$0.820°$	$0.399°$	$-32.239°$	$9.198\,10^{-6}$
$18 \rightarrow 16$	Direct	$0.280°$	$-4.664°$	$5.246°$	
	Cumul.	$0.274°$	$-4.419°$	$5.318°$	$1.985\,10^{-5}$
$21 \rightarrow 19$	Direct	$-0.301°$	$-0.0180°$	$15.164°$	
	Cumul.	$-0.207°$	$0.172°$	$15.568°$	$6.336\,10^{-5}$
$25 \rightarrow 23$	Direct	$-0.786°$	$-1.156°$	$-1.271°$	
	Cumul.	$-0.588°$	$-0.654°$	$-1.375°$	$9.197\,10^{-5}$

Table 4: Estimated rotations between nodes i and j once based on the propagated changes through the path from i to j (Cumul.) and once based on the registration of j with i (Direct). The differences are statistically negligible as shown by the small Mahalanobis distances χ^2.

6 Registration and Relaxation in the "Dwelling" Scenario

6.1 The Scenario

The performance with respect to computation speed and robustness of 3D-Plane-SLAM, i.e., plane based registration and closed form relaxation, is evaluated in this section based on a second set of experiments at the 2008 Response Robot Evaluation Exercise (RREE) in Disaster City [TEEX, 2008]. The scenario known as the "dwelling" deals with a small house hit by a flooding disaster. It consists of a half-collapsed porch, two mainly intact rooms with large amounts of rubble, damaged furniture, etc., and two kinds of courtyard structures including some collapsed roof parts. The scans also partly cover some pieces of a large concrete rubble pile in the direct neighborhood of the scenario.

In all 70 usable scans were taken. Figure 20 shows the view of the front camera of the robot at the locations where the scans where taken. Note that these are difficult conditions for the locomotion, i.e., a high presence of dust, rubble, and so on. Also, the floor is not level; this is partially due to rubble and partially due to artificial ramps that were added during RREE as locomotion and sensing challenges. The roll and pitch of the robot thus significantly varies during the mission.

As mentioned before, the front camera pictures has a wide angle lens with an horizontal opening angle of approximately $85°$, i.e., it only covers a fraction of the field of view of the actuated Laser Range Finder (ALRF). Figure 21 shows the first scan in the sequence as

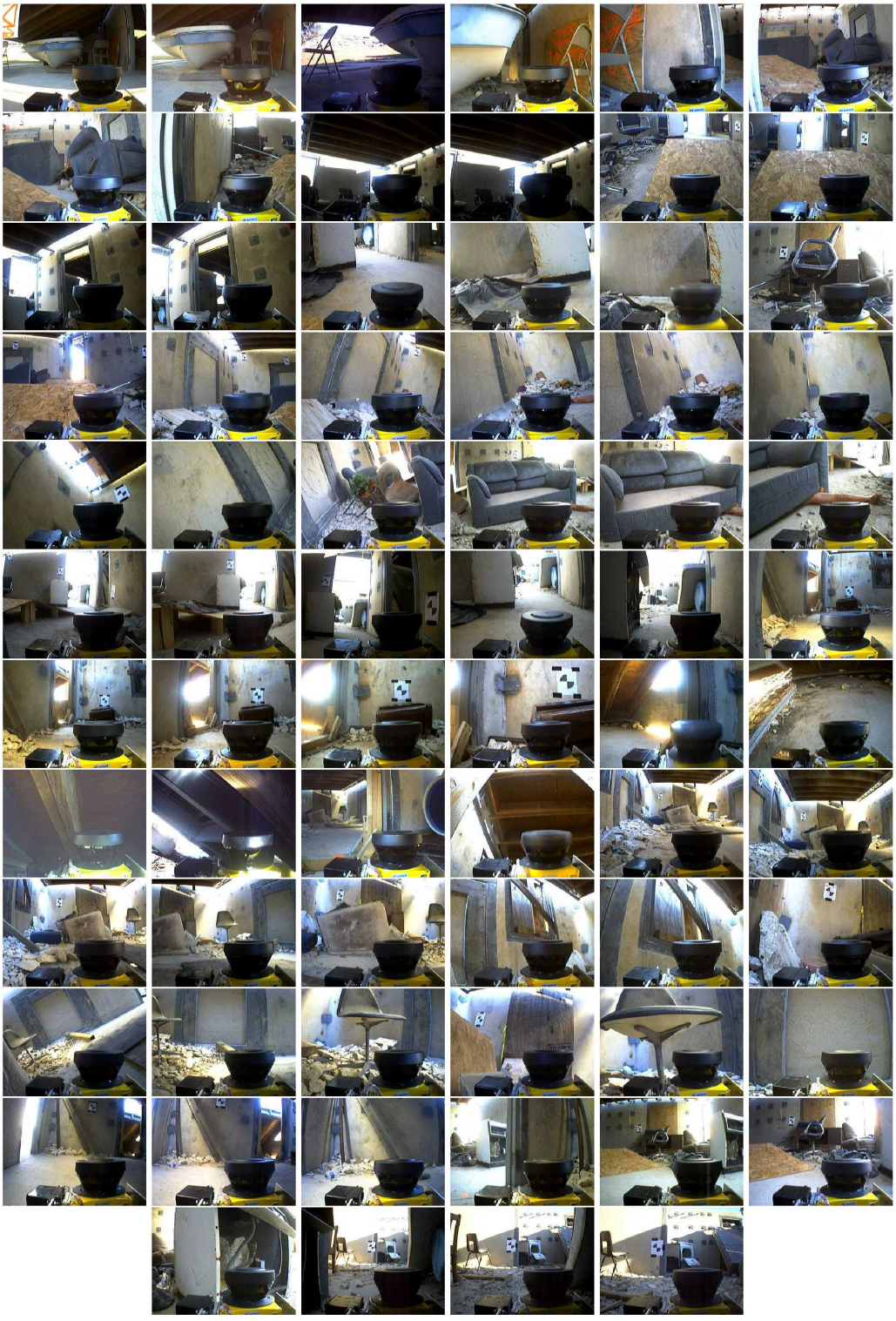
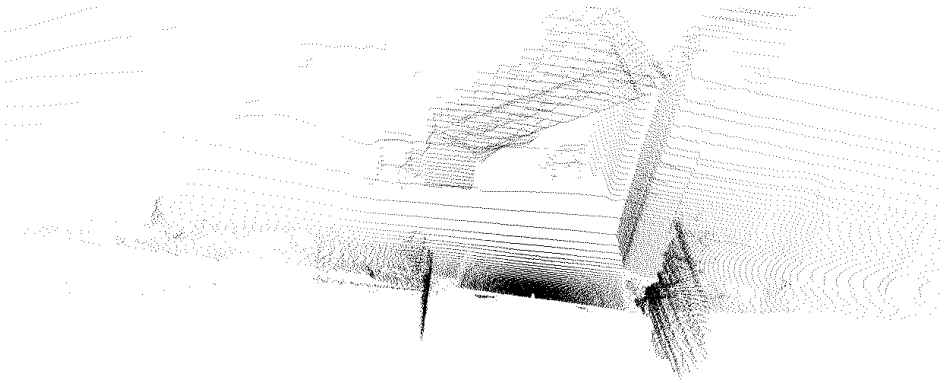


Figure 20: The front camera view of locations where the scans 0 to 69 were taken in the "dwelling" scenario.

a point cloud from two different perspectives and as a gray-scale range-image. The other 69 scans of the mission are shown as range-images in Fig. 22. The ALRF on the robot has a horizontal field of view of 270° of 541 beams. In this experiment, the servo pitches the ALRF from −90° to +90° at a spacing of 2°, i.e., with 4 times less vertical resolution than in the collapsed car park experiment. Each 3D point-cloud has hence a total size of $541 \times 91 = 49,231$ points. This setting was chosen during the response robot exercise to allow for faster scanning, namely in about 8 seconds per scan. Thus, the overall mission time in this scenario with 70 scans was about the same as for the mission with 26 scans at the collapsed car park.



(a) scan 0, point cloud, perspective view

(b) scan 0, range image

Figure 21: The first scan of the 70 scans of the dwelling scenario as a point cloud and as a gray-scale range-image.

The scenario is mainly an indoors environment but it provides several challenges. First of all, there are significant changes in roll and pitch of the robot. Second, the floor is covered with a lot of rubble and broken furniture, i.e., with many non-planar structures. This holds also to some extent for the walls. The ceiling is partially broken and it has a regular, repetitive structure of horizontal support beams, i.e., a significant potential for mis-registration of similar looking sub-patterns. Last but not the least, the robot moves through four narrow doors. This results in a lot of occlusions: most range data visible in one sample disappears in the next.

6.2 Registration Comparison with ICP

The challenges of this scenario are also reflected in the performance of ICP, which – in contrast to the crashed car park scenario – more or less completely fails here. As shown in

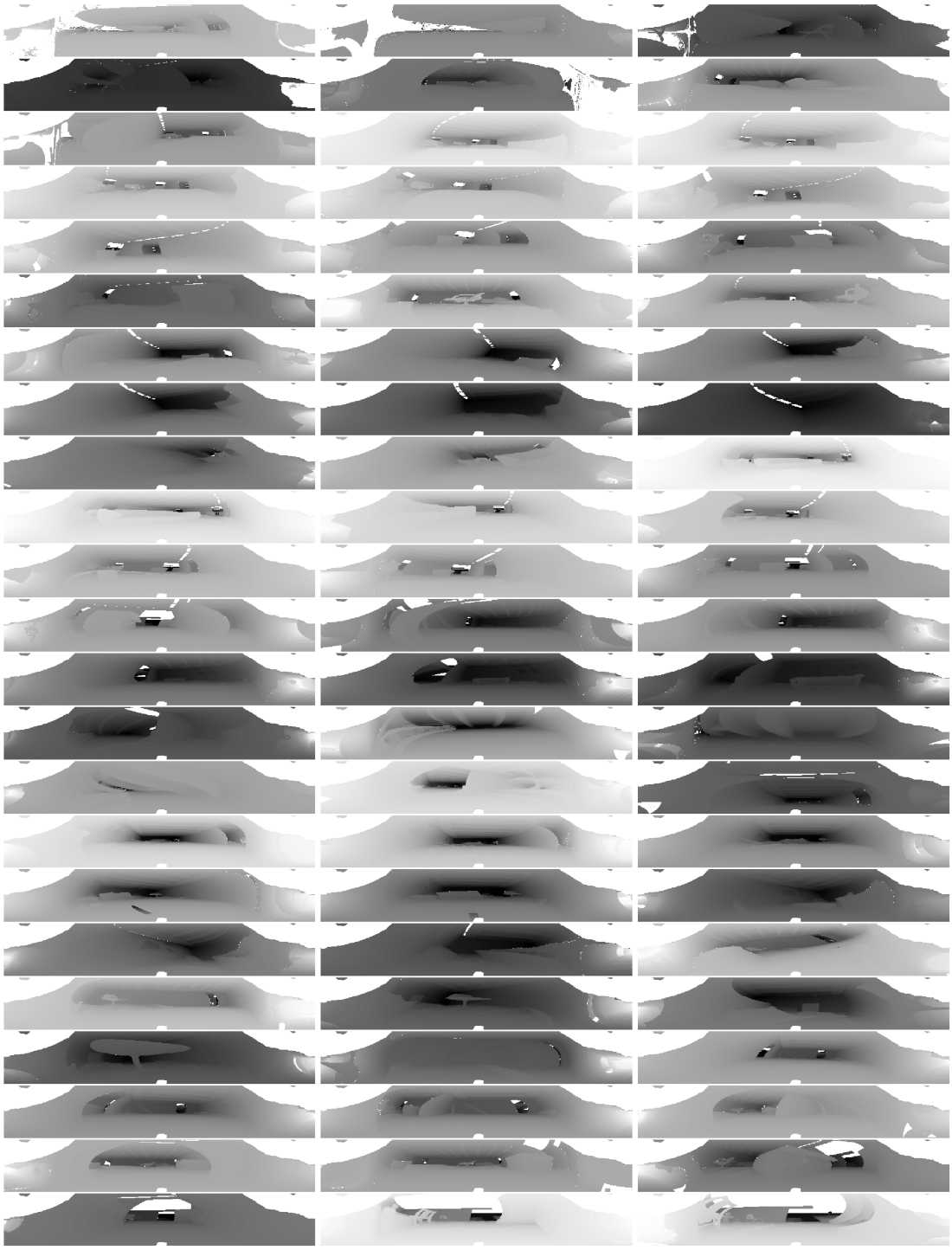
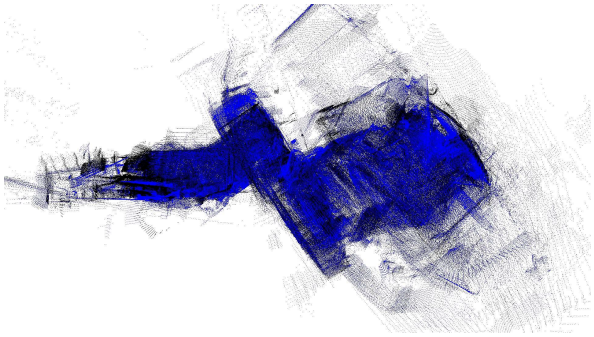
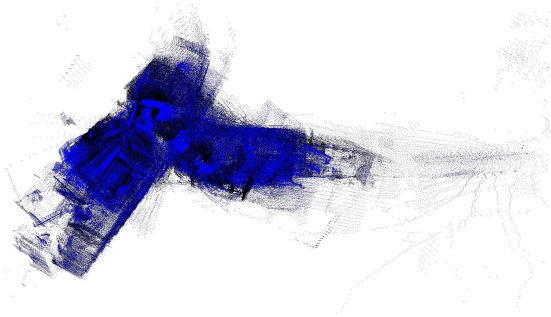


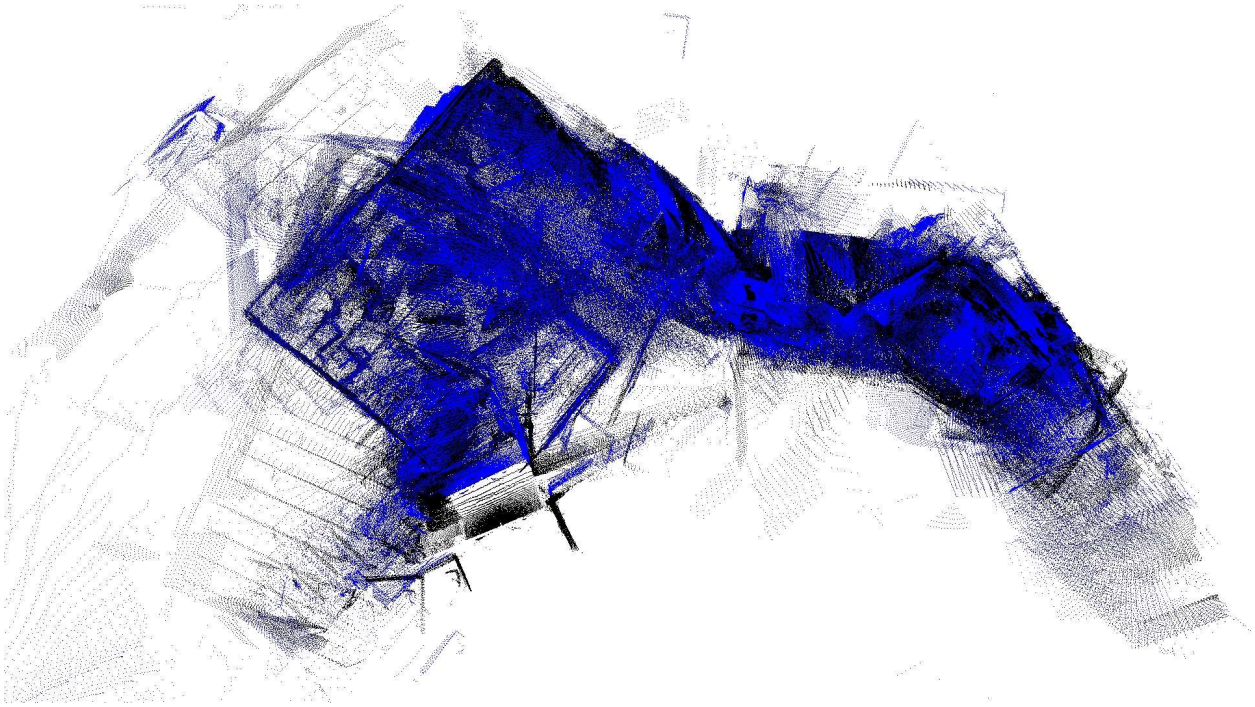
Figure 22: The other 69 range-images of the "dwelling" scenario. The first range-image is shown in Fig. 21(b).



(a) front view



(b) side view



(c) top view

Figure 23: The result of ICP registration in the "dwelling" scenario. The estimations of the relative orientations between scans pairs are quite often grossly wrong, leading to significant misrepresentations along the roll, pitch and yaw axes.

Loop	Type	Roll	Pitch	Yaw	χ^2 dist.
$17 \rightarrow 12$	Direct	$1.570°$	$-18.605°$	$-144.878°$	
	Cumul.	$3.302°$	$-18.109°$	$-146.096°$	$1.44\,10^{-3}$
$22 \rightarrow 17$	Direct	$1.345°$	$-18.037°$	$-111.323°$	
	Cumul.	$5.481°$	$-18.059°$	$-111.088°$	$5.23\,10^{-3}$
$27 \rightarrow 22$	Direct	$8.075°$	$9.011°$	$155.593°$	
	Cumul.	$9.813°$	$9.093°$	$158.078°$	$2.8\,10^{-3}$
$22 \rightarrow 8$	Direct	$8.677°$	$-5.032°$	$124.810°$	
	Cumul.	$9.545°$	$-3.736°$	$124.159°$	$8.7\,10^{-4}$
$60 \rightarrow 55$	Direct	$-4.499°$	$-4.187°$	$88.720°$	
	Cumul.	$2.080°$	$-7.966°$	$89.966°$	$1.8\,10^{-2}$

Table 5: Estimated rotations between nodes i and j, once based on the propagated changes through the path from i to j (Cumul.), and once based on the registration of j with i (Direct).

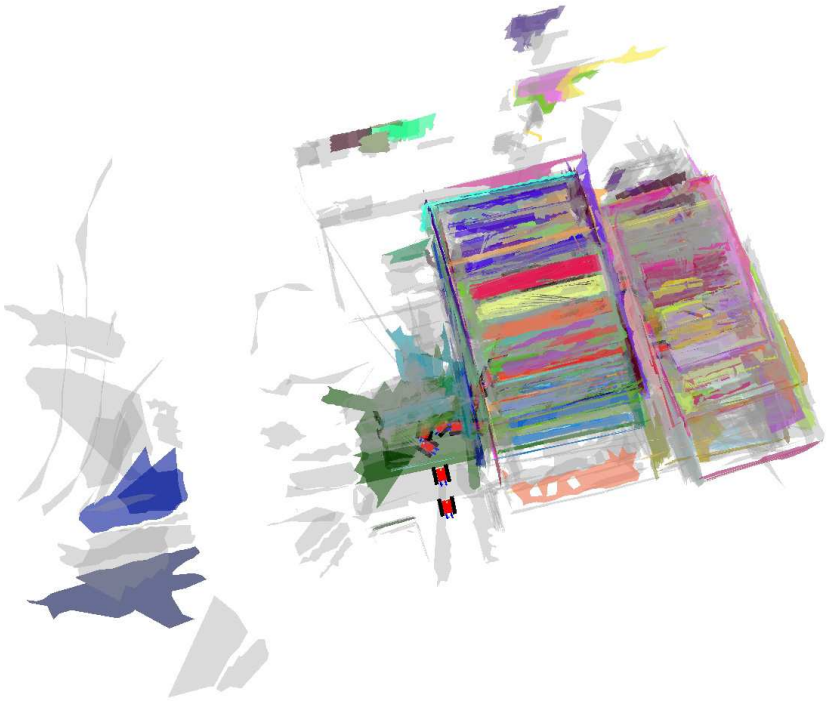
Fig. 23, the ICP registrations drift significantly in the estimations of the robot's roll, pitch and yaw. One likely reason for this is the lack of any meaningful odometry measurements that can be used to seed ICP with good initial guesses. As discussed before in Sec. 3.2, this is not unexpected in this scenario since the orientation of the robot often changes significantly between scans. Additionally, a pairwise inspection of registered scans indicates that ICP often also performs poorly on translations in this scenario; especially, sequences with motions through doors, with the concomitant occlusions, seem to pose difficulties.

6.3 Performance of 3D Plane SLAM

In contrast to ICP, the plane based registration provides a good initial basis for pose graph SLAM. The registration of the 70 scans succeeds in the sense that there are no gross misalignments, especially with respect to the relative orientation of registered scans. There are a few translational errors; the two most severe ones are discussed later on in this section in the context of qualitative improvements through the relaxation. The run-time is about 8 seconds on an average per full registration, including plane extraction and the polygonization for visualization. It is thus slightly faster than in the crashed car park scenario.

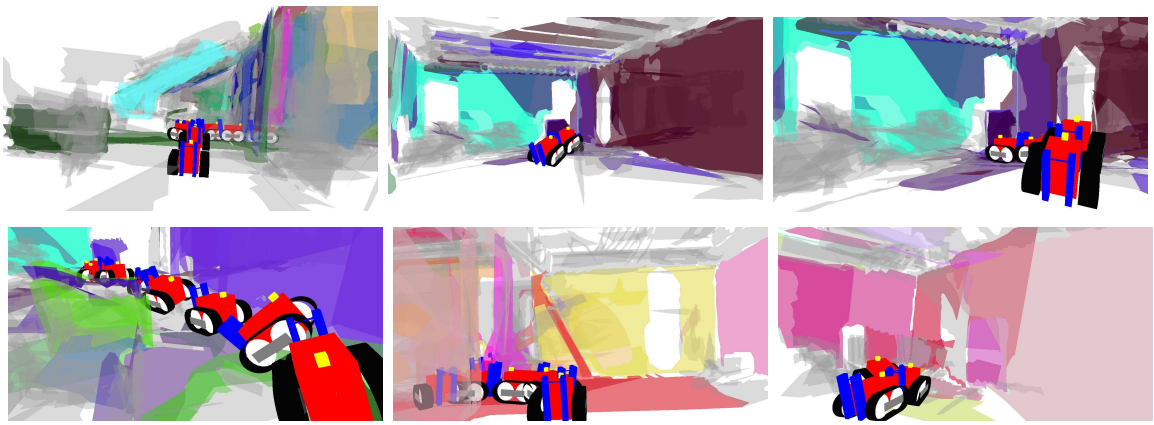
A proximity based loop detection is used again. It leads to a selection of 5 loops consisting of scan-pairs $(17, 12), (22, 17), (27, 22), (22, 8), (60, 55)$, where members of each pair are written in reverse order to emphasize that a scan is being matched against a non-sequential past scan. The related poses are shown in Fig. 27 where the path estimates are shown with and without relaxation. The SLAM proper, i.e., the closed form relaxation generates again hardly any computational overhead as it is computed with a run-time of 34.11 milliseconds. On relaxation, the overall consistency of the map increases, which is again reflected in quantitative as well as in qualitative terms.

The cost-function (Eq. (22)) is reduced from $954,808$ to 785.38. Its effect on the motion estimates can be seen in Fig. 27. During plane-matching, no prior roll/pitch estimates were available. The decrease in uncertainty is indicated by the significant decrease in the accu-



(a) top view

(b) front view



(c) perspective views, inside the building

Figure 24: The final map of the "dwelling" disaster scenario as result of 3D-Plane-SLAM. The robot is shown at the scan-locations by an avatar which is to scale.

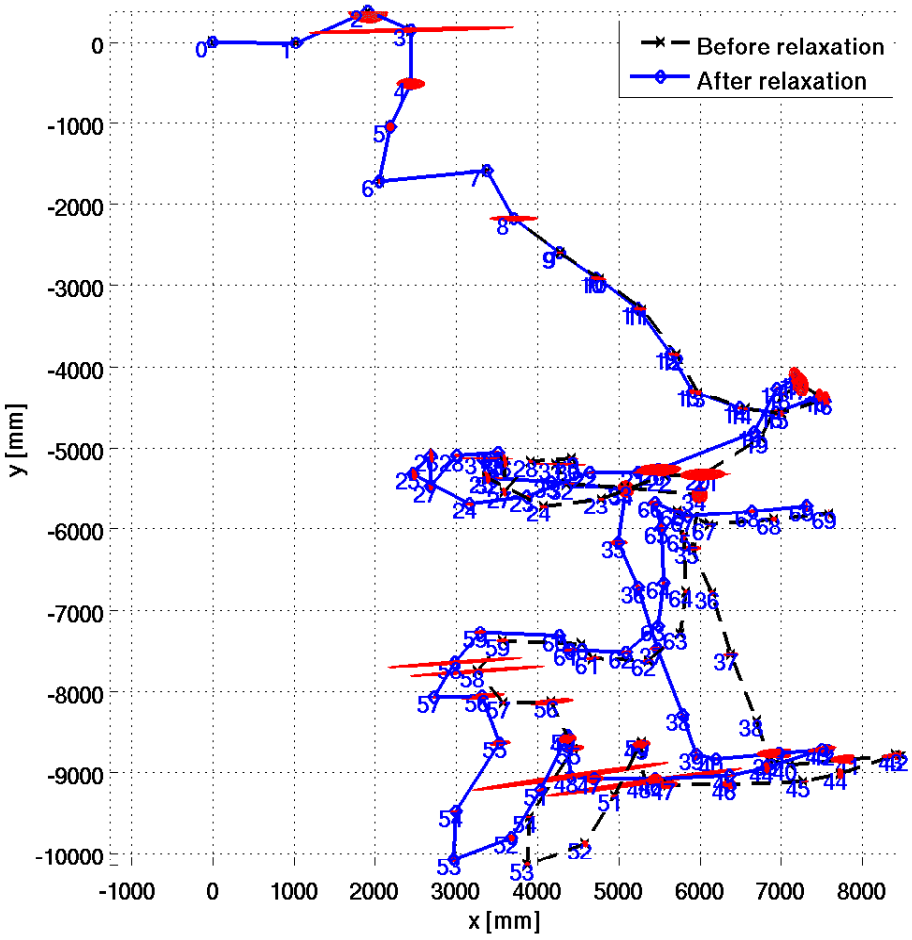


Figure 25: The **xy** path of the robot before and after relaxation. The pair-wise uncertainty 10σ ellipses, which form the basis for the relaxation, are shown in red. The **z** direction motion of the robot was in the range ± 500 mm.

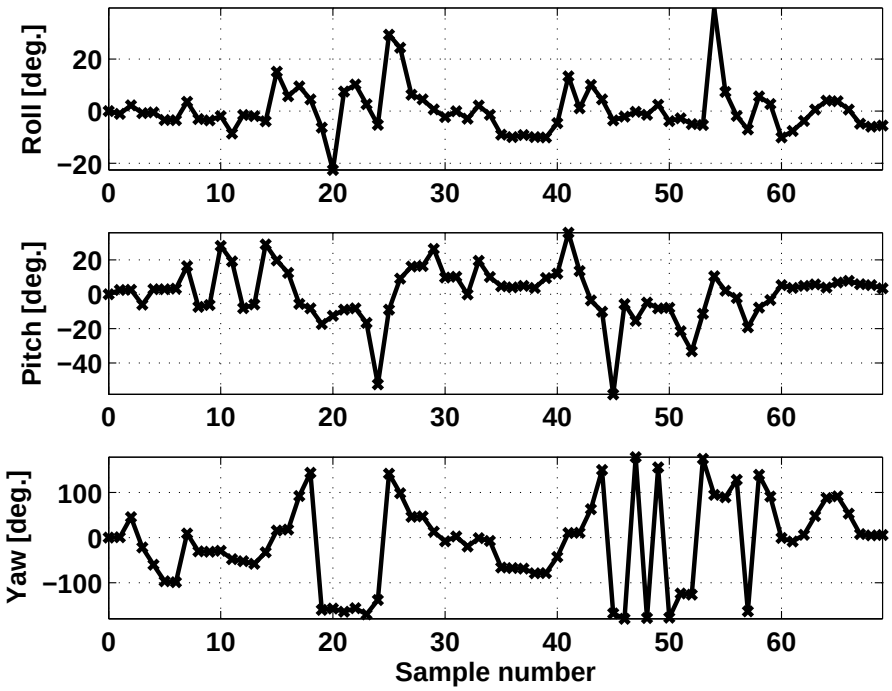


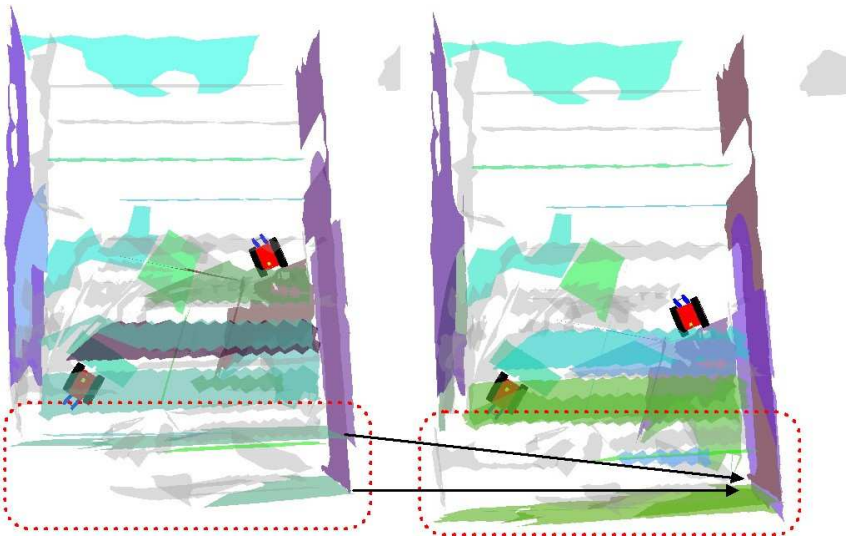
Figure 26: The computed roll, pitch, and yaw of the robot. Note the high roll and pitch at certain samples.

mulated Mahalanobis distance. The claim that the plane-matching leads to quite accurate rotation results is also supported in this scenario. The rotation computations for the loop-closing edges are listed in Table 4. Given, for example, the loop $17 \rightarrow 12$, the row marked "Direct" shows the result of matching $\mathcal{F}_{17}$ with $\mathcal{F}_{12}$ directly. The row marked "Cumul." shows the result of finding the relative transform between $\mathcal{F}_{17}$ and $\mathcal{F}_{12}$ by cumulatively computing the relative rotation by traversing all the intermediate edges as follows

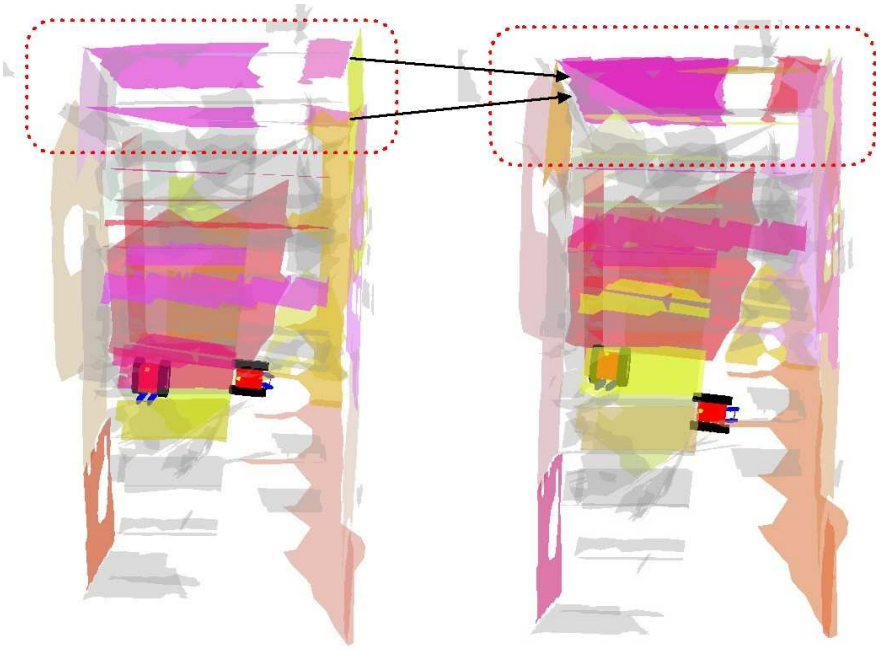
$$ {}^{12}_{17}\mathbf{R} = {}^{12}_{13}\mathbf{R}\,{}^{13}_{14}\mathbf{R}\,{}^{14}_{15}\mathbf{R}\,{}^{15}_{16}\mathbf{R}\,{}^{16}_{17}\mathbf{R} \tag{27} $$

The maximum differences are $6.5°$ for roll, $4°$ for pitch, and $2.5°$ for the yaw. This is slightly worse than in the collapsed car parking scenario but also to be expected. First of all, there are significant higher variations in roll and pitch during the robot's motions in this scenario. Second, a significant larger number of scans - namely 70 instead of 26 - is used in the dwelling scenario. Nevertheless, the fast, closed form relaxation is still a very reasonable alternative to full non-linear optimization in this case. This is also indicated by the Mahalanobis distance χ^2 based on the covariance matrix of the direct transform: at worst it is 10^{-2} and hence indicates no strong statistical significance of the differences.

The positive effects of the relaxation can also be seen in qualitative terms. Two quite apparent translational errors in the plane based registration are shown in Fig. 25. A pairwise manual inspection of the registrations, including the 5 loop pairs, suggested that these two cases had by far the largest translational error before relaxation. In both cases, a main wall is off by about 70 cm, i.e., about 2.3% of the ALRF's maximum range. As shown in Fig. 24, both these errors are corrected by the relaxation.



(a) scan pair 22-8, before (left) and after (right) relaxation



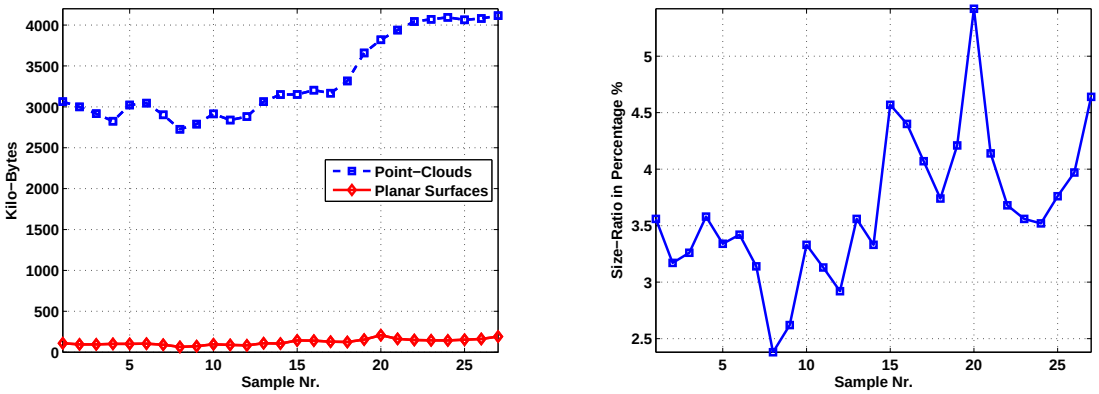
(b) scan pair 60-55, before (left) and after (right) relaxation

Figure 27: The result of relaxation can also be qualitatively seen in this scenario. The main effects are on the two pairs shown above where each time an error of about 70 cm in translation is corrected, causing the walls in room 1 (scan pair 22-8) and room 2 (scan pair 60-55) to be properly aligned.

7 A Comparison of Point Cloud Based Maps and Surfaces Based Maps

7.1 Planar Surface Patches for Map Representation

Point clouds are the predominant form of representation for 3D mapping. In this work we have used a surface based representation in the form of planar patches. This representation of 3D ALRF data is also the basis for the scan registration in our approach. Our registration method can be used for localization only, or – especially as it is capable of delivering proper covariance matrices – as a basis for SLAM. Note that the representation of the 3D map itself is completely independent of this. Obviously, one may use the plane registration for localization and simply fuse the 3D ALRF data as point clouds into a 3D map. If one prefers, the presented algorithms can hence be used to generate "standard" point cloud maps.



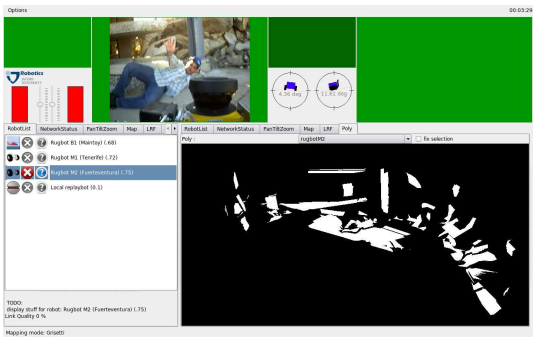
(a) Data sizes for point clouds and plane representations per scan.

(b) Data sizes of the plane representations relative to the point clouds sizes per scan.

Figure 28: The surface representation in form of polygons leads to significantly smaller amounts of data. This is especially beneficial in teleoperation scenarios when the data has to be transmitted over wireless links.

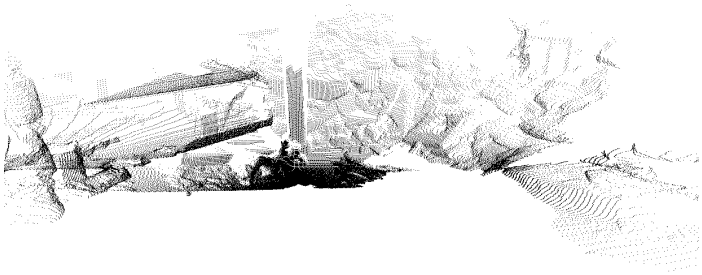
The representation used here has other advantages as well. First of all, it can be used for a significant data reduction. Figure 28 shows the data sizes of the two different representations per scan. For the point clouds, the average size is 3.235 MByte. For the surface representation, it is only 124 KByte. The planes require on average only 3.8% of the size of the point clouds. Within the whole set of all scans, the compression ranges from a maximum of 2.38% to a minimum of 5.42%. This compression makes a significant difference especially in teleoperation scenarios when wireless links are used. Also, it makes it faster and more efficient to visualize the map.

Secondly, the planar polygon representation is well suited for 3D maps even in the quite unstructured scenarios of Disaster City. Point clouds admittedly provide a higher level of detail, but point cloud maps tend to become too dense for a meaningful visualization (Fig. 19). A polygon based map, in contrast, always provides a very good overview of the scene. In addition, even rather complex, non-planar objects are reasonably well represented

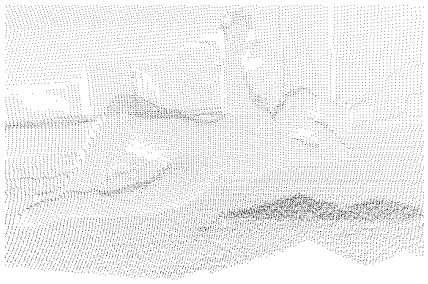


(a) The robot control GUI.

(b) Polygon representation of the victim.



(c) The corresponding point cloud.



(d) Close up of the victim in the point cloud.

Figure 29: A human "victim" in the collapsed car parking scenario. The polygon representation is much more efficient than the point cloud in terms of memory requirements.

and recognizable. Figure 29 shows a human disaster "victim", who can be clearly identified in the polygon representation.

7.2 Limitations and Open Questions

Surface based mapping has some potential but there are still some open questions. There is the obvious question as to which extent planar patches are suitable for registration of 3D scans in more general environments. The two disaster scenarios presented in this article are quite unstructured, but they nevertheless still represent urban environments. It is hence of interest to consider higher order surfaces, e.g. quadrics, as representations for large surface patches in non-urban settings. The determination of an appropriate level of detail is also an open question.

It is of foremost interest for 3D-Plane-SLAM to concentrate on the extraction of a few, large planes per scan that give high evidence correspondences for accurate registration. However, to make actual use of the 3D map, especially in 3D path-planning and navigation, more fine grained surface representations may be of interest. We postulate that surface patches are a good candidate for these purposes, especially compared to point clouds. The main reason for this is that large planes with polygon boundaries are very well suited for computational geometry algorithms employed in 3D path-planning and navigation.

Another open question is the implementation of loop-closing in scenarios where the planed

based registration does not provide a reasonable initial 3D map. The loop detection used here is based on spatial proximity: it presumes that the computed robot's path corresponds reasonably well with reality. If this is not the case, $O(n^2)$ pairwise registrations of all n scans may be needed to determine possible loops.

In addition to these general issues, there are some more concrete limitations of the presented 3D-Plane-SLAM, especially with respect to the relaxation method used here. The closed-form solution for pose-graph relaxation is based on the assumption that the determination of the rotations is accurate. Though this accuracy is quite high as shown in the experiments in this article, it is nevertheless not perfect. Orientation errors accumulate and they can become sufficiently severe for long sequences of registered scans. In this case, a more complex non-linear optimization is required.

7.3 Multimedia

Several animated movies of point cloud and polygon maps generated with our algorithms can be downloaded from the following URL: `http://robotics.jacobs-university.de/projects/3Dmap/`. The website also contains multimedia material of the scenarios, e.g., the front camera movies.

8 Conclusions

Two contributions in the context of a novel approach to 3D mapping are presented. First of all, it is shown that a new algorithm for registering large planar surface patches can be applied to quite unstructured environments. Experimental data from two disaster scenarios is presented, which was collected at the 2008 NIST Response Robot Evaluation Exercise (RREE) in Disaster City, Texas. Second, we show how to embed the plane registration in pose-graph SLAM. In doing so, the angular precision of the plane registration is exploited and the relaxation is formulated as a closed-form solution.

We demonstrated that the presented algorithm is quite fast and robust in comparison to Iterative Closest Point (ICP). The registration of scans is done in about 10 seconds, including plane extraction and polygonization for visualization. This is less than it takes the robot to move in the scenarios from one scan spot to the next. Therefore, online mapping is feasible. 3D-Plane-SLAM, i.e., the embedding of the registration into a pose-graph with loop-closing and relaxation, comes almost for free. It takes only a few milliseconds of computation time and improves the 3D maps in quantitative and qualitative terms.

9 Acknowledgments

This work was supported by the *Deutsche Forschungsgemeinschaft* (German Research Foundation).

References

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

[Craig, 2005] Craig, J. J. (2005). *Introduction to Robotics – Mechanics and Control*. Prentice Hall.

[Duckett et al., 2002] Duckett, T., Marsland, S., and Shapiro, J. (2002). Fast on-line learning of globally consistenst maps. *Autonomous Robots*, 12(3):287–300.

[Fischer and Kohlhepp, 2000] Fischer, D. and Kohlhepp, P. (2000). 3D geometry reconstruction from multiple segmented surface descriptions using neuro-fuzzy similarity measures. *Journal of Intelligent and Robotic Systems*, 29:389–431.

[Golub and Loan, 1980] Golub, G. H. and Loan, C. F. V. (1980). An analysis of the total least squares problem. *SIAM Journal of Numerical Analysis*, 17(6):883–893.

[Grisetti et al., 2007] Grisetti, G., Stachniss, C., Grzonka, S., and Burgard, W. (2007). A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA.

[Hähnel et al., 2003] Hähnel, D., Burgard, W., and Thrun, S. (2003). Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, 44(1):15–27.

[Horn, 1987] Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642.

[Kanatani, 2005] Kanatani, K. (2005). *Statistical Optimization for Geometric Computation*. Dover Publications, Inc., New York. ISBN 0486443086.

[Kohlhepp et al., 2006] Kohlhepp, P., Bretthauer, G., Walther, M., and Dillmann, R. (2006). Using orthogonal surface directions for autonomous 3D-exploration of indoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3086–3092.

[Kohlhepp et al., 2004] Kohlhepp, P., Pozzo, P., Walther, M., and Dillmann, R. (2004). Sequential 3D-SLAM for mobile action planning. *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 1:722–729 vol.1.

[Kohlhepp et al., 2007] Kohlhepp, P., Strand, M., Bretthauer, G., and Dillmann, R. (2007). The elastic view graph framework for autonomous, surface-based 3D-SLAM Part I: Concept and local layer. *at - Automatisierungstechnik*, 55:136–145.

[Lu and Milios, 1997] Lu, . F. and Milios, . E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349.

[Magnusson et al., 2007] Magnusson, M., Lilienthal, A., and Duckett, T. (2007). Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827.

[Nüchter et al., 2007] Nüchter, A., Lingemann, K., and Hertzberg, J. (2007). 6D SLAM– 3D mapping outdoor environments. *Journal of Field Robotics*, 24(8/9):699–722.

[Nuechter et al., 2004] Nuechter, A., Surmann, H., Lingemann, K., Hertzberg, J., and Thrun, S. (2004). 6D SLAM with an application in autonomous mine mapping. *Robotics*

and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, 2:1998–2003.

[Olson et al., 2006a] Olson, E., Leonard, J., and Teller, S. (2006a). Fast iterative alignment of pose graphs with poor initial estimates. In Leonard, J., editor, *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2262–2269.

[Olson et al., 2006b] Olson, E., Leonard, J., and Teller, S. (2006b). Fast iterative alignment of pose graphs with poor initial estimates. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2262–2269.

[Pathak et al., 2009a] Pathak, K., Vaskevicius, N., and Birk, A. (2009a). Revisiting uncertainty analysis for optimum planes extracted from 3D range sensor point-clouds. In *IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan.

[Pathak et al., 2009b] Pathak, K., Vaskevicius, N., Poppinga, J., Pfingsthorn, M., Schwertfeger, S., and Birk, A. (2009b). Fast 3D mapping by matching planes extracted from range sensor point-clouds. In *International Conference on Intelligent Robots and Systems (IROS)*. St. Louis, MO, USA.

[Pfingsthorn and Birk, 2008] Pfingsthorn, M. and Birk, A. (2008). Efficiently communicating map updates with the pose graph. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.

[Pfingsthorn et al., 2007] Pfingsthorn, M., Slamet, B., and Visser, A. (2007). A scalable hybrid multi-robot SLAM method for highly detailed maps. In *RoboCup 2007: Proceedings of the International Symposium*, LNAI. Springer.

[Poppinga et al., 2008] Poppinga, J., Vaskevicius, N., Birk, A., and Pathak, K. (2008). Fast plane detection and polygonalization in noisy 3D range images. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Nice, France.

[Shuster, 2006] Shuster, M. D. (2006). The generalized Wahba problem. *The Journal of the Astronautical Sciences*, 54(2):245–259.

[Surmann et al., 2003] Surmann, H., Nuechter, A., and Hertzberg, J. (2003). An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3-4):181–198.

[Takeuchi and Tsubouchi, 2008] Takeuchi, E. and Tsubouchi, T. (2008). Multi sensor map building based on sparse linear equations solver. *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2511–2518.

[TEEX, 2008] TEEX (2008). NIST Response Robot Evaluation Exercise. `http://www.teex.com/teex.cfm?pageid=USARprog\&area=USAR\&templateid=1538`.

[Thrun et al., 2003] Thrun, S., D. Haehnel, D. F., Montemerlo, M., Triebel, R., Burgard, W., Baker, C., Omohundro, Z., Thayer, S., and Whittaker, W. (2003). A system for volumetric robotic mapping of abandoned mines. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. Taipei, Taiwan.

[Weingarten and Siegwart, 2006] Weingarten, J. and Siegwart, R. (2006). 3D SLAM using planar segments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing.