Improving Simultaneous Mapping and Localization in 3D Using Global Constraints

Rudolph Triebel and Wolfram Burgard
Department of Computer Science, University of Freiburg
George-Koehler-Allee 79, 79108 Freiburg, Germany
{triebel, burgard}@informatik.uni-freiburg.de

Abstract

Recently, the problem of learning volumetric maps from three-dimensional range data has become quite popular in the context of mobile robotics. One of the key challenges in this context is to reduce the overall amount of data. The smaller the number of data points, however, the fewer information is available to register the scans and to compute a consistent map. In this paper we present a novel approach that estimates global constraints from the data and utilizes these constraints to improve the registration process. In our current system we simultaneously minimize the distance between scans and the distance of edges from planes extracted from the edges to obtain highly accurate three-dimensional models of the environment. Several experiments carried out in simulation as well as with three-dimensional data obtained with a mobile robot in an outdoor environment we show that our approach yields seriously more accurate models compared to a standard approach that does not utilize the global constraints.

Introduction

Learning maps with mobile robots is a high-dimensional state estimation problem which requires a simultaneous solution to the question of how to update the map given sensory input and an estimate of the robot's pose as well as to the question of how to estimate the pose of the robot given the map. Recently, techniques that compute the most likely map based on a graph of spatial relations have become quite popular (Lu & Milios 1997; Gutmann & Konolige 1999; Frese 2004; Konolige 2004). The advantage of such methods is that they actually do not require predefined landmarks. Rather they can cope with arbitrary representations by considering so-called constraints between the poses where observations of the environment were perceived.

The approach proposed in this paper is based on the technique for consistent pose registration by Lu & Milios (1997). The key idea of this approach is to determine constraints between positions where measurements were obtained from the odometry between successive poses and from matching range scans taken at different poses. The approach then minimizes an energy term composed of the individual constraints between the poses.

In the past, this approach has mainly been applied for two-dimensional representations of the environment and in many applications scan matching based on laser range scans has been used to determine the necessary constraints. Accordingly, the computational requirements for generating a constraint from two observations is limited. This, however, changes in the context of learning volumetric maps from three-dimensional range scans. Due to the enormous complexity of each scan, which may consist of 500,000 proximity values per scan, only a restricted number of scans can be stored and actually used to extract the constraints. This introduces two problems. First the system might become under-constrained and second, due to the reduced overlap between the scans, the quality of the individual constraints becomes low.

In this paper, we present an approach to extract global constraints from 3d range scans and to utilize these constraints during the optimization process. Our approach is motivated by the fact that many man-made environments such as buildings often contain features that can be observed in many views and at similar places. For example, the windows belonging to one and the same level of a building are typically at the same height on all sides of the building. Accordingly, the windows introduce additional constraints on range scans even if the scans have only a small or even no overlap. The key idea of the approach proposed in this paper is to extract such global constraints from three-dimensional range scans and to improve the optimization process based on these global constraints.

In the remainder of this paper we describe our approach and its implementation on a mobile robot. In practical experiments carried out with a complex outdoor data set we illustrate that the global constraints seriously improve the quality of the resulting maps especially in situations in which the individual range scans have a limited overlap.

Related Work

The problem of simultaneous mapping and localization has been studied intensively in the past. The individual approaches can be classified along multiple dimensions which include important aspects like the type of the representation of the environment and the question of how the posterior about the robot's pose and the map is represented. Extended Kalman Filter methods belong to the most popular approaches and different Variants of this technique have been proposed (Moutarlier & Chatila 1989; Leonard & Feder 1999; Guivant, Nebot, & Baiker 2000; Dissanayake *et al.* 2001; Thrun *et al.* 2004). The key idea of these techniques is to simultaneously estimate the poses of landmarks using an Extended Kalman Filter or a variant of the Extended Information Filter. As mentioned above, an alternative approach is to compute the most likely map based on a graph of spatial relations (Lu & Milios 1997; Gutmann & Konolige 1999; Frese 2004; Konolige 2004). The advantage of such methods is that they do not require predefined landmarks. Rather they can cope with arbitrary representations by considering so-called constraints between the poses where observations of the environment were perceived. However, most of these approaches rely on the assumption that the environment can be represented by a two-dimensional structure.

Recently, several authors investigated the problem of constructing 3d-models of buildings. For example, Stamos & Leordeanu (2003) construct 3d-models by combining multiple views obtained with a 3d range scanner. Früh & Zakhor (2004) present a technique to learn accurate models of large-scale outdoor environments by combining laser, vision, and aerial images. Thrun, Burgard, & Fox (2000) use two 2d range scanners. The first is oriented horizontally whereas the second points toward the ceiling. By registering the horizontal scans the system generates accurate three-dimensional models. In a more recent work by Thrun *et al.* (2003) several range scanners were used to learn models of underground mines. Nüchter, Surmann, & Hertzberg (2003) developed a robot that is able to autonomously explore non-planar environments and to simultaneously acquire the three-dimensional model. Compared to these approaches, which directly compare range scans to estimate the pose of the vehicles, the algorithm proposed in this paper extracts global constraints from the range scans and utilizes these constraints in an optimization process similar to that of Lu & Milios (1997). As a result, our algorithm can cope with fewer scans and even with a smaller overlap between the individual scans.

Network-based Pose Optimization

Suppose we are given a set of N partial views $V_1, \ldots, V_N$ of a scene in 3D. A view V_n is defined by a set of $s(n)$ points in 3D, where $s(n)$ is defined as the *size* of view V_n. We will denote these points as $\mathbf{z}_1^n, \ldots, \mathbf{z}_{s(n)}^n$. Each view V_n is taken from a position $\mathbf{x}_n \in \mathbb{R}^3$ and an orientation $(\varphi_n, \vartheta_n, \psi_n)$. The tuple $(\mathbf{x}_n, \varphi_n, \vartheta_n, \psi_n)$ is denoted as the *robot pose* $\mathbf{p}_n$. The goal now is to find a set of poses that minimize an appropriate error function based on the observations $V_i, \ldots, V_N$.

Following the approach described by Lu & Milios (1997), we formulate the pose estimation problem as a system of error equations that are to be minimized. We represent the set of robot poses as a constraint network, where each node corresponds to a robot pose. A link l in the network is defined by a set of nodes that are connected by l. It represents a constraint between the connected nodes. Note that this is an extension of the link definition given by Lu & Milios (1997) who only consider links between pairs of nodes. In the following, we will deal with two different kinds of network links, namely

- links that connect exactly two consecutive robot poses (so-called *local links*) and
- links that connect several, not necessarily consecutive, robot poses (denoted as *global links*).

3D vs. 2D Registration

Lu & Milios (1997) represent a local constraint between poses $\mathbf{p}_i$ and $\mathbf{p}_j$ as the difference D_{ij} in position and orientation. Let us assume that R_{ij} is the rotation matrix corresponding to the difference $(\varphi_j - \varphi_i, \vartheta_j - \vartheta_i, \psi_j - \psi_i)$ in the Euler angles of $\mathbf{p}_i$ and $\mathbf{p}_j$. The term D_{ij} is considered a random variable with mean $\bar{D}_{ij}$ and covariance C_{ij}. Assuming that an estimate of $\bar{D}_{ij}$ is given by local scan matching in terms of a rotation $\bar{R}_{ij}$ and a translation $\bar{\mathbf{t}}_{ij}$, we can express each scan point $\mathbf{z}_h^i$ from view V_i in the local coordinate system of pose $\mathbf{p}_j$ as

$$\bar{\mathbf{z}}_h^i = R_j^{-1}(\bar{R}_{ij}(R_i\mathbf{z}_h^i + \mathbf{x}_i) + \bar{\mathbf{t}}_{ij} - \mathbf{x}_j), \qquad (1)$$

where R_j is the rotation defined by $(\varphi_j, \vartheta_j, \psi_j)$ and R_i the one defined by $(\varphi_i, \vartheta_i, \psi_i)$.

In order to understand the additional problems introduced by moving from 2d to 3d, let us consider the pose differences D_{ij} in the situation in which $\mathbf{x}_i = \mathbf{0}$ and $\mathbf{x}_j = \mathbf{0}$. If the scan matching is perfect, we obtain $\bar{\mathbf{t}}_{ij} = \mathbf{0}$. Under the assumption that the common points in both scans can be matched perfectly, this results in

$$\mathbf{z}_h^j = R_j^{-1}\bar{R}_{ij}R_i\mathbf{z}_h^i. \qquad (2)$$

Here we can see a major difference between the 2D case and the 3D case. For rotations in 2D it is guaranteed that $\bar{R}_{ij} = R_{ij}$. This can be seen by setting two of the Euler angles to 0. In 3D however, this is not true in general. In practical experiments we found out that in 3d the approximation of R_{ij} by $\bar{R}_{ij}$ obtained from scan matching introduces a linearization error that often prevents the optimization process from converging.

Definition of Local Constraints

To cope with this problem, we consider a quadratic error function and in this way avoid linearization errors. More specifically, we compute the sum of squared errors between corresponding points from both views V_i and V_j. This means that the error defined between poses $\mathbf{p}_i$ and $\mathbf{p}_j$ can be expressed as

$$l(\mathbf{p}_i, \mathbf{p}_j) \quad := \quad \sum_{v=1}^{C} \|R_i\mathbf{z}_{c_1(v)}^i + \mathbf{x}_i - (R_j\mathbf{z}_{c_2(v)}^j + \mathbf{x}_j)\|^2 \qquad (3)$$

where $(c_1(1), c_2(1)), \ldots, (c_1(C), c_2(C))$ is the set of C correspondences between points from the views V_i and V_j.

Since this error function is non-linear in the pose parameters $\mathbf{p}_i$ and $\mathbf{p}_j$ we cannot use the closed-form solution generally used in the context of two dimensions.

Definition of Global Constraints

In general, a global constraint between different robot poses can be defined in many possible ways. For example, if a set of 3D landmarks $\lambda_1, \ldots, \lambda_G$ with known poses is given, then each of these landmarks constitutes a global constraint on all robot poses from which a partial view onto the landmark was taken. The evaluation function γ_j that corresponds to the global constraint g_j is then defined by the squared Euclidean distance between the view V_i and the landmark λ_j.

In this paper, we will not assume the existence of known landmarks. We rather define the global constraints based on the object(s) seen from the different views. This stems from the observation that many real world objects, such as buildings are highly self-similar. For example, if a building is seen from two sides, it is very likely that specific features that are extracted from both views (e.g., windows, the edge between walls and the roof etc.) are on the same absolute height in both views. In general, many different types of features are possible, where high-level features such as windows, doors etc. constrain the algorithm to be applicable for specific objects like buildings. We therefore rely on low-level features, in particular 3D edges.

Before we discuss how to extract edges from range scans, we will first describe how we actually use the edges to generate global constraints. Assume we are given a set of edge features $E_i = \{e^i_1, \ldots, e^i_M\}$ for each view V_i. We define a global constraint as a plane P that has a sufficient *support* by edges detected in different views. Here the support supp(P) of a plane P is defined by all edges e that lie entirely inside a given corridor around the P. Given the support we calculate the error imposed by a global constraint g between the poses $\mathbf{p}_{i_1}, \ldots, \mathbf{p}_{i_k}$ as

$$g(\mathbf{p}_{i_1}, \ldots, \mathbf{p}_{i_K}) \;:=\; \sum_{k=1}^{K} \sum_{e \in E_{i_k} \cap \ \mathrm{supp}(P)} d(R_{i_k} e + \mathbf{t}_{i_k}, P) \quad (4)$$

Here, d defines the squared distance of the rotated and translated edge e to the plane P. In our implementation, this equals the sum of squared Euclidean distances between the transformed vertices of e and P.

Pose estimation

As stated above, we formulate the pose estimation as a minimization problem for a given set of error equations. Suppose the network consists of L local and G global constraints. The aim now is to find a set of poses $\mathbf{p}_1, \ldots, \mathbf{p}_N$ that minimizes the overall error function f:

$$\begin{aligned} f(\mathbf{p}_1, \ldots, \mathbf{p}_N) \;:=\; & \sum_{i=1}^{L} l_i(\mathbf{p}_{\nu_1(i)}, \mathbf{p}_{\nu_2(i)}) \cdot \alpha \; + \\ & \sum_{i=1}^{G} g_i(\mathbf{p}_{\nu_1(i)}, \ldots, \mathbf{p}_{\nu_{K(i)}(i)}) \cdot (1 - \alpha) \quad (5) \end{aligned}$$

Here, α is introduced as a factor that weighs between local and global constraints. To solve this non-linear optimization problem we f by using the Fletcher-Reeves gradient descent algorithm.

It should be noted that the global minimum for the error function f is not unique. This is because both local and global constraints are only defined with respect to the relative displacements between the robot poses and the global minimum of f is invariant with respect to affine transforms of the poses. In practice, this problem can be solved by fixing one of the robot poses at its initial value. Then the other poses are optimized relative to this fixed pose.

Approximating Correspondences and Planes

When formulating the error functions in Eqns. (3) and (4) we assumed some prior knowledge about the local and global constraints. In the case of the local constraints, we assumed that the correct correspondences between points in two different views are given. For the global constraints we assumed that the optimal coordinates for the planes are given. However, in practice both assumptions do not hold. Therefore we approximate both the correspondences and the planes for the global constraints.

Obtaining the Correspondences

For a given set of initial robot poses $\mathbf{p}_i$ and $\mathbf{p}_j$ we calculate the correspondences of points between consecutive views V_i and V_j by applying a variant of the Iterative Closest Point scan matching algorithm (ICP) described in (Besl & McKay 1992). After convergence of the ICP, we obtain the correspondences directly from the last ICP iteration. In general, ICP is not guaranteed to converge to a global optimum. In fact, the convergence highly depends on the initial values for the poses. This means that the poses that minimize Eqn. (3) are in general not globally optimal. It is not even guaranteed that they are a better approximation to the global optimum than the initial poses are, because the ICP may diverge in some cases.

Obtaining the Planes

Similar to the correspondences for the local constraints, we approximate the planes for the global constraints from a set of initial poses $\mathbf{p}_1, \ldots, \mathbf{p}_N$. From the resulting set of edge features $E_1, \ldots, E_N$ we compute a set of planes so that their normal vectors are orthogonal to at least two edges. By doing this we obtain for example the surface planes of a building. These planes constrain all views that have edges close the planes to be aligned to the planes.

Dependency between Constraints and Poses

Both the approximation of the correspondences and the planes is dependent on the initial state of the robot poses. This means that a good first estimate of the poses yields a good approximation of the local and global constraints. Conversely, a good approximation of the constraints reduces the probability of running into local minima during the pose optimization process. In other words, we have a circular dependency between constraints and robot poses. Therefore, the idea is to estimate both constraints and poses iteratively. Note that such an approach is applied in many other iterative approximation methods such as Expectation Maximization (EM), where a set of hidden random variables, that are

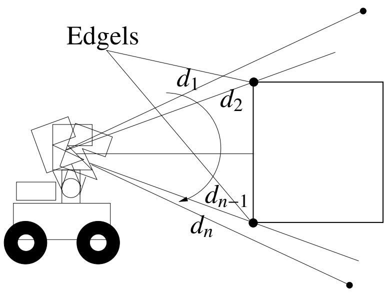

Figure 1: Detection of edgels. These are defined as scan points in a vertical scan line that have a distance d_i from the view point which is smaller than the difference of a neighbor's distance and a given threshold τ, i.e.: $d_i < d_{i-1} - \tau$ or $d_i < d_{i+1} - \tau$

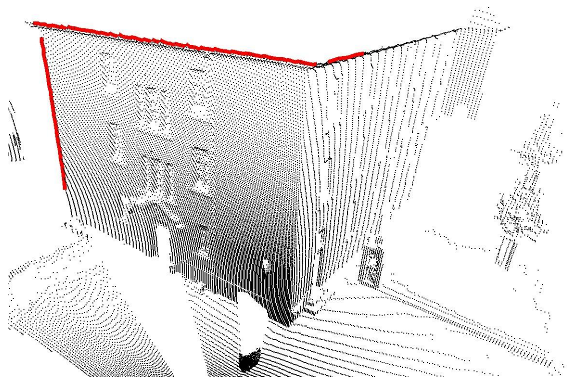

Figure 2: Extracted edge features extracted from a single 3d scan of a building.

dependent on the parameters, is estimated and fed back to the optimization of the parameters. In contrast to the maximum likelihood estimation process applied in an EM framework, our algorithm is not guaranteed to converge. However, in our experiments we found that by introducing the global constraints the iterative optimization gets more stable with respect to convergence.

Feature Extraction

As described above, our algorithm relies on the extraction of 3D edge features from each partial view V_i. These edge features are then used to determine a set of planes representing the global constraints. In the following, we describe the details of the feature extraction process.

Edges from Views

For a given view V_n, the edge features are extracted as follows:

1. We detect a set of edge points (*edgels*). These are defined as scan points in a vertical scan line for which one vertical neighbor point is far away from the view point compared to the edgel itself (see Fig. 1).
2. Then we calculate the tangent vectors $\mathbf{t}_j$ for each edgel $\mathbf{e}_j$. Here, $\mathbf{t}_j$ is defined by the first principle direction for a set of edgels in the vicinity of $\mathbf{e}_j$.
3. Next, we cluster the edgels twice. The first clustering is done with respect to the tangent directions. We use a spherical histogram to find tangent vectors that point into similar directions. The obtained clusters are then clustered wrt. the positions of the edgels in space. This is done using a region growing technique.
4. Finally, we connect all edgels in one cluster to a polyline.

Fig. 2 shows an example of a set of edge features extracted from a real 3d scan. In this figure, only polylines with a minimum length of $3m$ are shown.

Planes from Edges

Once the edge features have been extracted from the partial views, we search for planes that are orthogonal to at least two edges, which may not be parallel and at the same time close to each other, in the remainder denoted as equal. Note that this problem is different from clustering edges into planes, because one edge can lie in several planes. This means that standard clustering methods – like EM-based clustering (Liu *et al.* 2001) – cannot be applied.

In our current system we instead use a variant of the RANSAC algorithm (Fischler & Bolles 1981). We start by randomly sampling an edge pair (e_1, e_2) from the set of all pairs of non-equal edges. Here, an edge pair is sampled with a probability proportional to the sum of the length of both edges. This way we obtain with a higher likelihood planes that have a high support. Next, we find the plane P_m that is orthogonal to both edges of the pair. Here we have to consider two different cases, namely that e_1 and e_2 are parallel, but far away from each, or that e_1 and e_2 are not parallel. In the first case, an orthogonal vector to e_1 and e_2 is not uniquely defined. Therefore, we define P_m in this case as the plane that minimizes the squared distance to all edgels from e_1 and e_2. In the second case, the normal vector of P_m is defined by the cross product of the main directions of e_1 and e_2.

The next step includes the calculation of the support of the resulting plane P_m. After this we apply a hill climbing strategy to obtain more general planes by fitting a new plane P'_m into all edges from the support of plane P_m. This is done by finding a normal vector $\mathbf{v}^*$ so that

$$\mathbf{v}^* = \arg\min_{\mathbf{v}} \|A\mathbf{v}\|, \ \|\mathbf{v}\| = 1 \qquad (6)$$

where A is a $k \times 3$ matrix consisting of all tangent vectors corresponding to the edges from the support of P_m.

The minimum in Eqn. (6) is determined by computing the singular value decomposition (SVD) of A. The optimal normal vector $\mathbf{v}^*$ is then obtained as the last column of the matrix V where $A = UDV^T$, assuming the singular values in D are sorted in descending order.

Experimental Results

To evaluate our algorithm we implemented it and tested it on real data as well as in simulation runs. The goal of the experiments reported here is to illustrate that the incorporation

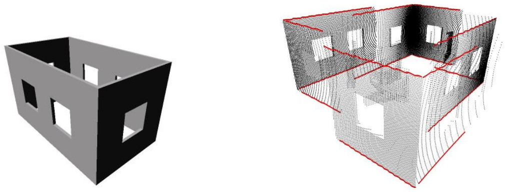

Figure 3: Simulated scene used to verify the performance of the algorithm. left: 3D view of the scene; right: 4 generated partial views shown in an explosion view drawing.

of global constraints increases the accuracy of the resulting models.

Real World Experiment

In order to analyze the performance of our registration algorithm in practice, we tested it with a data set taken from a real world scene. The data set consisted of 6 different three-dimensional scans taken from a building that is about 70m long, 20m wide and 11m high from the ground to the roof edge.

Fig. 4 shows the result of the view registration using global constraints. As can be seen, 5 different planes were detected by our algorithm. These planes were used as global constraints in the network of robot poses. The figure also depicts close-up-views of several parts of the model. Shown are the results obtained with our approach and with local constraints only. As can be seen from the figure, the edges detected in the partial views have been aligned more accurately to the planes using our approach. Especially at the roof plane we can see that the views are all at the same level. To quantify the improvement we measured the variance σ_α of the absolute angular deviation of the edges from the corresponding plane tangent. For the global-constraint based registration we obtained $\sigma_\alpha = 0.75$. In contrast, the local constraints only produce a value of $\sigma_\alpha = 2.56$.

Quantitative Results

To more systematically evaluate the quality of the models obtained with our algorithm and in comparison with other approaches we performed a series of experiments using the simulated scene shown in Figure 3. The object is 3m wide, 5m long and 3m high. In the particular experiment reported here we generated 4 different scans. These scans are shown in Figure 3 in an exploded view drawing. For the initial poses we added Gaussian noise to the poses known from the ground truth. The noise added to the angles had a variance that was different from the noise variance added to the positions. We ran two different kinds of experiments where the noise variances were (0.0005|0.05) and (0.0008|0.08) respectively. For both noise levels we started the optimization with 10 different initial sets of poses. We evaluated both the registration method using only local constraints and the one using local and global constraints.

The results for the low variance case are summarized in Table 1. Shown are the average deviations in angle μ_ϵ and position μ_x from the ground truth. As can be seen, the incorporation of global constraints improves the registration process. Especially the angular deviation is smaller when using global constraints. This is because the global planar constraints correct smaller errors that arise from the local scan matcher. These errors are mainly encountered in the rotations.

Table 1: Comparison of performance for different registration algorithms with respect to average angular deviation μ_ϵ (in radians) and positional deviation μ_x (in meters)

Method	μ_ϵ	μ_x
local constr.	0.0175	0.1595
global + local constr.	0.0071	0.1109

In the case of a high noise variance, the algorithm that uses only local constraints always diverged. This is because the local scan matching could not find enough correspondences and therefore diverged. However, in some of these cases convergence could be achieved by adding the global constraints.

In a further experiment we demonstrate the performance of our approach in situations where few overlap is given between the individual partial views. Again, we used the scene shown in fig. 3. We ran the registration process using different numbers of partial views, ranging from 6 down to 2, where the overlap decreased with the number of views. Fig. 5 plots the rotational deviations from the ground truth. As can be seen, our algorithm performs significantly better ($\alpha = 0.05$) in situations with fewer overlap between the individual views.

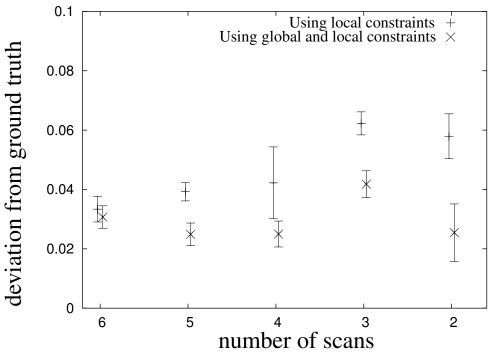

Figure 5: Statistical analysis of the registration process with respect to the angular rotation from the ground truth

Conclusions

This paper describes an approach to learn accurate volumetric models from three-dimensional laser range scans. The key idea of our approach is to extract global constraints from the individual scans and to utilize these constraints in during the alignment process. Our algorithm has been implemented and tested on real data and in simulation. Experimental results demonstrate that our approach yields more accurate models especially in situations in which only a few scans with little overlap are given.

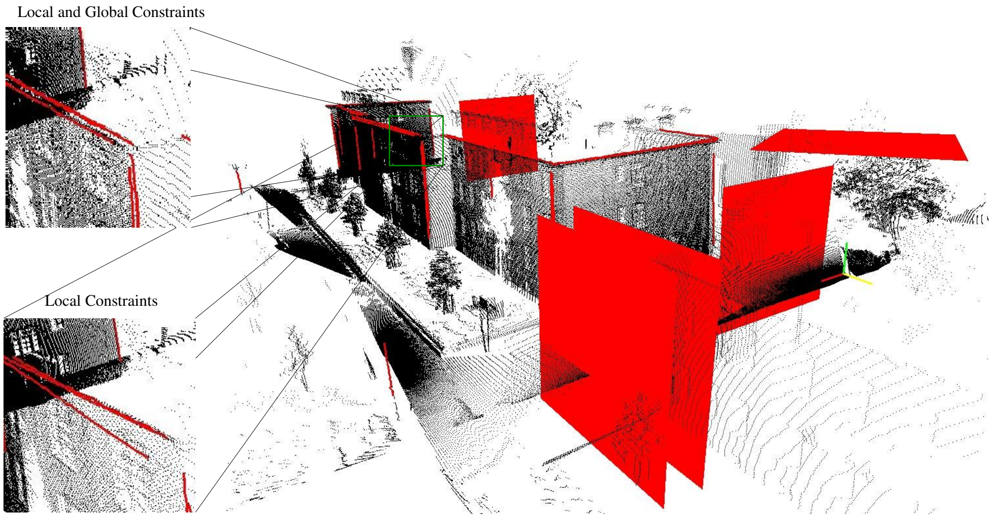

Figure 4: Registered data set from an outdoor scene. Shown are the range data and the edges used to extract the planes as global structures.

Acknowledgments

This work has partly been supported by the German Research Foundation under contract number SFB/TR8.

References

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

Dissanayake, G.; Newman, P.; Clark, S.; Durrant-Whyte, H.; and M., C. 2001. A solution to the simultaneous localisation and map building (slam) problem. *IEEE Transactions on Robotics and Automation*.

Fischler, M. A., and Bolles, R. C. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Assoc. Comp. Mach.* 24(6):381–396.

Frese, U. 2004. *An $O(\log n)$ Algorithm for Simulateneous Localization and Mapping of Mobile Robots in Indoor Environments*. Ph.D. Dissertation, University of Erlangen-Nürnberg.

Früh, C., and Zakhor, A. 2004. An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision* 60:5–24.

Guivant, J.; Nebot, E.; and Baiker, S. 2000. Autonomous navigation and map building using laser range sensors in outdoor applications. *Journal of Robotics Systems* 17(10):565–583.

Gutmann, J., and Konolige, K. 1999. Incremental mapping of large cyclic environments. In *International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*.

Konolige, K. 2004. Large-scale map-making. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*.

Leonard, J., and Feder, H. 1999. A computationally efficient method for large-scale concurrent mapping and localization. In *Proceedings of the Ninth International Symposium on Robotics Research*.

Liu, Y.; Emery, R.; Chakrabarti, D.; Burgard, W.; and Thrun, S. 2001. Using EM to learn 3D models with mobile robots. In *Proceedings of the International Conference on Machine Learning (ICML)*.

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

Moutarlier, P., and Chatila, R. 1989. An experimental system for incremental environment modeling by an autonomous mobile robot. In *Proc. of the International Symposium on Experimental Robotics*.

Nüchter, A.; Surmann, H.; and Hertzberg, J. 2003. Planning robot motion for 3d digitalization of indoor environments. In *Proc. of the 11th International Conference on Advanced Robotics (ICAR)*.

Stamos, I., and Leordeanu, M. 2003. Automated feature-based range registration of urban scenes of large scale. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thrun, S.; Hähnel, D.; Ferguson, D.; 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. of the IEEE International Conference on Robotics & Automation (ICRA)*.

Thrun, S.; Liu, Y.; Koller, D.; Ng, A.; Ghahramani, Z.; and Durant-Whyte, H. 2004. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research* 23(7-8):693–704.

Thrun, S.; Burgard, W.; and Fox, D. 2000. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*.