Building Local Safety Maps for a Wheelchair Robot using Vision and Lasers

Aniket Murarka, Joseph Modayil, and Benjamin Kuipers
Department of Computer Sciences
The University of Texas at Austin
Email: {aniket, modayil, kuipers}@cs.utexas.edu

Abstract

To be useful as a mobility assistant for a human driver, an intelligent robotic wheelchair must be able to distinguish between safe and hazardous regions in its immediate environment. We present a hybrid method using laser range-finders and vision for building local 2D metrical maps that incorporate safety information (called local safety maps). Laser range-finders are used for localization and mapping of obstacles in the 2D laser plane, and vision is used for detection of hazards and other obstacles in 3D space. The hazards and obstacles identified by vision are projected into the travel plane of the robot and combined with the laser map to construct the local 2D safety map. The main contributions of this work are (i) the definition of a local 2D safety map, (ii) a hybrid method for building the safety map, and (iii) a method for removing noise from dense stereo data using motion.

1. Introduction

We present a hybrid laser and vision based method for building local 2D metrical maps that incorporate safety information for mobile robots.

The intended application of this work is to develop an autonomous robotic wheelchair to serve as a mobility assistant for people who have movement disabilities but normal perception and cognition. The wheelchair should have the ability to autonomously explore and navigate the environment with the driver maintaining executive control. The expected environmental settings are urban regions like university campuses which have slow moving traffic and (generally) follow the Americans with Disabilities Act (ADA) standards and guidelines [17], such that most places be accessible via sidewalks and wheelchair ramps. The platform for our work is Vulcan, a wheelchair robot shown in Figure 1 that has been (mostly) developed in our lab. It is equipped with two laser range-finders, a stereo camera [18], and optical encoders (for odometry). One of the primary requirements for such an application is safety. The agent should be capable of distinguishing between safe and hazardous regions in its surroundings and representing this information in a local map. Maintaining local maps also helps the robot navigate and explore the environment autonomously [1].

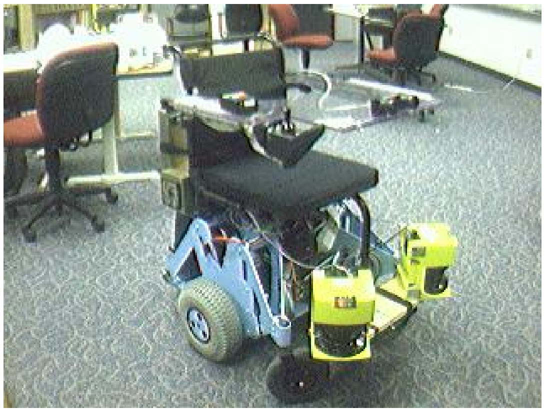


Figure 1. The wheelchair robot has a stereo camera and two laser range-finders.

Great progress has been made in recent years on the problem of simultaneous localization and mapping (SLAM) in unknown environments particularly when using laser range-finders [15]. Existing SLAM algorithms can localize the robot to within a few centimeters and construct accurate metrical maps of local environments. However the maps constructed by the lasers only show obstacles that intersect the plane of their beams. Obstacles above or below that plane are invisible, as are hazards such as drop-offs. Vision, on the other hand, can provide accurate (though noisy) 3D range information and hence be used to identify the remaining obstacles and hazards in the robot's surrounding space.

Our approach to building safe local metrical maps utilizes these respective strengths of lasers and vision. We use laser range-finders for incremental localization and mapping of obstacles in the 2D laser plane, and vision for detection of hazards and other obstacles in 3D space. Dense stereo vision and visual feature based methods are used to build a 3D point cloud model of the surrounding space.

[1] A local map of the robot's surroundings can also serve as an interface between the robot and its human driver, e.g, to pass directions from human to robot, or for the robot to ask the driver for advice in cases of uncertainty.

Table 1. Features of urban environments relevant for mobile robot safety.

Features	Examples
Fixed Obstacles	Walls, tables, stairs
Dynamic Obstacles	People, doors
Invisible Obstacles	Glass doors, glass walls
Overhangs	Table tops, tree branches
Drop offs	At sidewalk curbs, staircases
Rough surfaces	Gravel paths, lawns
Inclines	Wheelchair ramps, sidewalks up/down a hill
Narrow regions	Doorways, narrow sidewalks

The 3D point cloud is used to identify safe and hazardous regions that are then projected into the robot's 2D travel plane. The combination of the 2D laser metrical and projected visual data is used to create a *local 2D safety map* of the robot's surrounding with safe and hazardous regions defined. A key contribution of this work is a robust method for removing noise from dense stereo data using motion.

Note that once a local map of a region is constructed the robot does not keep the 3D model of the world around and instead works with the 2D safety map significantly reducing computation. In section 2, we show that for safe navigation of a mobile robot such as the wheelchair, this 2D safety map is an adequate representation.

For proper exploration and path planning the robot needs a global map of the world. We use the HSSH (Hybrid Spatial Semantic Hierarchy) [8], to build a global map from the local safety maps that the robot constructs as it moves through the world. The HSSH factors the problem of global mapping into four problems: (a) local metrical mapping of small regions, i.e, building the local safety maps in our case; (b) local topology extraction from the local metrical map; (c) global topological mapping, accomplished by applying topological axioms to local topologies and, (d) global metrical mapping, which is accomplished by utilizing the skeleton provided by the global topological map to combine the local metrical maps into a single frame of reference. A key advantage to using the HSSH framework, is that only local metrical maps that are bounded in size are required. This leads to significant computational savings as it limits the amount of visual processing and does not require accurate metrical localization over long distances.

The paper is organized as follows. The next section discusses environmental characteristics and the safety map. Sections 3 and 4 explain the laser and vision based methods for building safety maps. Section 5 presents results followed by related work in section 6. Section 7 concludes.

2. The Environment and the Safety Map

To build safe maps, we first identify the characteristics of the urban environment relevant to the robot and then present a suitable representation for the local safety map.

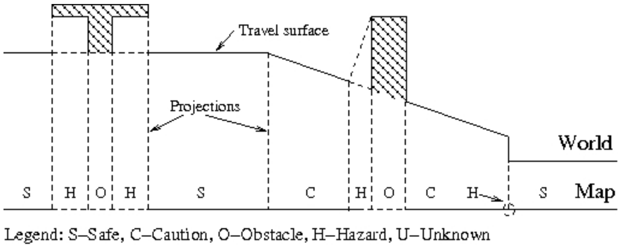


Figure 2. Features (obstacles, hazards) in the world are projected to create a 2D safety map.

2.1. Characterization of the Environment

Table 1 classifies the major features of the urban environment (relevant for safety) in which we want the robot to function. Based on this set of features the robot can classify parts of the world as being: 1. Obstacles. 2. Hazards, consisting of overhangs, drop offs, steep inclines, and very rough surfaces. 3. Caution areas, consisting of inclines, narrow regions, and rough surfaces. 4. Unknown areas, consisting regions with insufficient data. 5. Safe areas.

Although, in general the surfaces over which a robot travels can be inclined, in this paper we only consider environments where the travel surfaces are level, i.e, not inclined. This is mostly true of indoor office environments and many outdoor places as well. Therefore in this paper we treat inclines as hazards. Furthermore, we only consider fixed obstacles and ignore rough surfaces.

2.2. The Local 2D Safety Map

Based on the characterization of the environment we propose to represent the local safety map as a 2D plane with various features of the world appropriately projected onto this plane and annotated as being an obstacle, a hazard, an area of caution, unknown, or safe. The projection is not straightforward and requires the robot to take into consideration the angle of the travel surface locally, as Figure 2 shows [2]. Given that the robot is localized at all times, such a map captures all the structure and information required for safe navigation by the robot in its *local* surroundings.

Given the relationship between the safety map and the world, it is easy to see how a 3D model of the environment can be projected to obtain the safety map - in fact for level travel surfaces the projection is trivial. In the following sections, we describe in detail how we use lasers and vision to obtain a 3D model (in our case a 3D point cloud) to construct a safety map of the local environment of a robot.

[2] Since in this paper we work with level travel surfaces, inclines are treated as hazards (as opposed to caution areas as shown in figure 2).

3. 2D Localization and Mapping using Lasers

For localization in small regions, we exclusively use the laser range-finders. The algorithm used is a variant of the one presented in [15] and is generally accurate to within a few centimeters. Mapping is performed simultaneously with localization and results in accurate 2D metrical maps of the local environment. In particular, at any instant this process gives us a set of localized robot poses, a collection of 2D points corresponding to obstacles in the plane of the laser beams, which we call the 2D laser metrical map, and also a 2D occupancy grid map of the world with uncertain (unknown) regions marked.

4. 3D Landmarks from Stereo

We use the stereo camera to build a 3D cloud of point landmarks in the world. *The point landmarks can correspond either to individual pixels or to visual features in a stereo image pair.* To get the 3D locations of points corresponding to pixels a dense stereo vision algorithm is used (these algorithms also work at subpixel resolution). To get locations of points corresponding to features we implement our own method based on SIFT features [9]. The reason we use both feature and pixel based methods is to increase the amount of information available on the environment (for safety). It is possible that features might be detected in areas where dense stereo algorithms fail (e.g., in low texture areas) and vice versa.

The 3D point location estimates obtained using either the feature based method or the dense stereo algorithm are noisy and also contain many false positives, e.g., points in free space. Therefore we develop a probabilistic framework to track point landmark locations to reduce noise and to match point landmarks across frames to remove false positives (*note that the probabilistic framework is independent of how the point landmarks are generated - whether from pixels or features*). Only point landmarks that are observed consistently over many frames are made permanent and the rest are discarded removing false positives. To the best of the authors knowledge, the probabilistic framework described here has not been used earlier to reduce noise and remove false positives from landmarks produced using *dense stereo methods* and is one of the contributions of this work. However, similar methods have been applied before to landmarks produced using feature based methods.

4.1. Stereo Range and Error Analysis

The idealized geometry of the stereo camera obtained after calibration (we use calibration software that comes with the camera [18]) is shown in Figure 3. A point p's 3D coordinates in the camera reference frame $\mathbf{x}_p^c = (x_p^c, y_p^c, z_p^c)^T$, can be obtained given its location in the image and its disparity ($d_p = c_p^L - c_p^R$), i.e., $\mathbf{x}_p^c = \lambda(\mathbf{z_p})$ where $\mathbf{z}_p = (r_p^R, c_p^R, d_p)^T$ [3]. To determine the disparity of a point that corresponds to a pixel/feature in one image we find the pixel/feature in the other image that corresponds to the same point. This is accomplished by matching each pixel/feature in one image (say left image) to all pixels/features in the other image and picking the best matches. For dense stereo, i.e., pixel-to-pixel matches, we use a multi-resolution stereo matching algorithm that comes with the camera [7]. For SIFT features we use each feature's image coordinates and visual properties (scale, orientation, and local neighborhood descriptors) as a basis for finding matches.

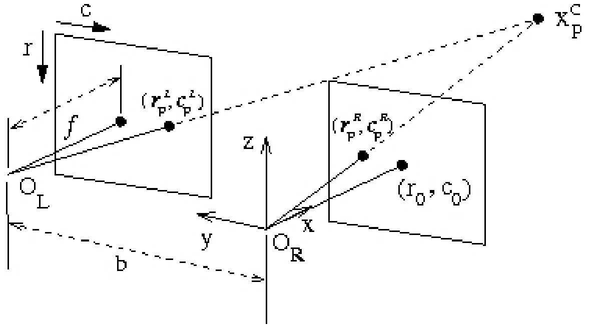


Figure 3. Geometry of the (L)eft and (R)ight stereo imagers observing a point in the world.

Localization gives the position of the robot and hence the camera (for simplicity we assume the camera and robot frames coincide) in the global frame of reference $\mathbf{x}_R^g = (x_R^g, y_R^g, \theta_R^g)^T$, allowing us to calculate the point's position in the global frame of reference, $\mathbf{x}_p^g = \kappa(\mathbf{x}_p^c, \mathbf{x}_R^g)$. To estimate error in the point's location, we model the error in the point's image coordinates with a Gaussian distribution with covariance $\Sigma_p = diag(\sigma_r^2, \sigma_c^2, \sigma_d^2)$, [10]. Then the error in the point's position in the global reference frame, i.e., the covariance Σ_p^g in its global position, can be approximated by first order error propagation (and assuming negligible localization error) as follows, [10]:

$$\Sigma_p^g = \left[\frac{\partial\kappa}{\partial\mathbf{x}_p^c}\right]\left[\frac{\partial\lambda}{\partial\mathbf{z}_p}\right]\Sigma_p\left[\frac{\partial\lambda}{\partial\mathbf{z}_p}\right]^T\left[\frac{\partial\kappa}{\partial\mathbf{x}_p^c}\right]^T \qquad (1)$$

4.2. Tracking Landmarks

In this section we develop the probabilistic framework by which we reduce noise in the point landmark location estimates and remove false positives. To build a map of landmarks the robot has to solve two problems at every frame: (i) determine associations/matches between the points observed in the current frame and existing (temporary or permanent) landmarks, and (ii) update existing landmark location estimates based on the observed locations of the matching points. Points that do not match any existing landmarks are used to initialize new *temporary* landmarks. Hence, when the robot first wakes up in the world, it does not know of any existing landmarks, and so *all* points are used to initialize temporary landmarks. Over time, if enough points do not match the temporary landmarks they are considered to be false positives and removed from the landmark database, otherwise they are made into permanent landmarks. A nice thing about the probabilistic method is that it combines the use of point locations and visual properties for matching into a single framework. However, the method assumes localization error is negligible. While it does not seem to affect the results much, it is an avenue for further work. We consider both problems beginning with (ii).

Updating Landmark Location Estimates. Let the *true* location of a landmark, l_i, in the global frame of reference be $\mathbf{x}_{l_i}$, and of a point p_j, observed in the current time step (or frame) t, be $\mathbf{x}^t_{p_j}$ [3]. Each landmark location *estimate* is modeled by a Gaussian probability distribution. Let $\widehat{\mathbf{x}}^t_{l_i} = \{\mu^t_{l_i}, \Sigma^t_{l_i}\}$ be the parameters of the distribution over landmark l_i's location *estimate* at time t. We use a Kalman filter [4], to update the landmark's location estimate based on the observed locations of matching points - in other words, to reduce the amount of error in the landmark location estimates. Given that point p_j is associated with l_i at time t, the landmark location estimate from $t-1$ can be updated as follows:

$$\begin{aligned}(\Sigma^t_{l_i})^{-1} &= (\Sigma^{t-1}_{l_i})^{-1} + (\Sigma^t_{p_j})^{-1} \\ \mu^t_{l_i} &= \Sigma^t_{l_i}\left((\Sigma^{t-1}_{l_i})^{-1}\mu^{t-1}_{l_i} + (\Sigma^t_{p_j})^{-1}\mathbf{x}^t_{p_j}\right) \qquad (2)\end{aligned}$$

Associating Points with Landmarks. We develop a Bayesian framework for associating points with landmarks. The problem is to pick from all points observed at time t, the point that has the highest probability of being associated with landmark l_i, based on point and landmark locations and visual properties.

We describe the relevant visual properties of point p_j observed at time t by a vector: $\mathbf{d}^t_{p_j}$. These properties are the visual properties of the feature/pixel, e.g., scale and orientation for SIFT features (neighborhood descriptors are not used) that the point corresponds to in one of the stereo images. The visual properties $\mathbf{d}^t_{l_i}$ of a landmark l_i at t, are taken to be the visual properties of a feature/pixel associated with the landmark in the past. The feature/pixel chosen is one whose viewing direction (in the global frame of reference) when it was seen, is closest to the current viewing direction to the landmark.

To make the process tractable, let $P_i = \{p_1, p_2, ..., p_{n_i}\}$ be the set of points that can possibly be associated with landmark l_i. This initial set can be obtained in various ways, for example by collecting all points less than a certain Euclidean distance from the landmark. Since we assume that each point can be associated with exactly one landmark, let $a_i \in P_i$ denote the point with which landmark l_i is associated. The point p^* most likely to be associated with l_i is then given by,

$$p^* = \arg\max_{[p_j \in P_i]} p(a_i = p_j \mid I_X, I_D) \qquad (3)$$

where I_X and I_D represent location and visual property terms, $I_X = \{\widehat{\mathbf{x}}^{t-1}_{l_i}, \{\mathbf{x}^t_{p_k}\}_{k=1..n_i}\}$, $I_D = \{\mathbf{d}^t_{l_i}, \{\mathbf{d}^t_{p_k}\}_{k=1..n_i}\}$.

Equation 3 can be simplified using Bayes rule and making an independence assumption between location and visual properties,

$$\begin{aligned}p(a_i = p_j \mid I_X, I_D) &\propto p(I_X, I_D \mid a_i = p_j)\, p(a_i = p_j) \\ &\propto p(I_X \mid a_i = p_j)\, p(I_D \mid a_i = p_j) \qquad (4)\end{aligned}$$

Also, given no prior information we assume $p(a_i = p_j)$ to be uniform for all $p_j \in P_i$. The left term in equation 4 can be simplified as follows,

$$\begin{aligned}p(I_X | a_i = p_j) \;&=\; p(\mathbf{x}^t_{p_j} \mid \widehat{\mathbf{x}}^{t-1}_{l_i}, \{\mathbf{x}^t_{p_k}\}_{\forall k \neq j}, a_i = p_j) \\ &\quad .\, p(\widehat{\mathbf{x}}^{t-1}_{l_i} \mid \{\mathbf{x}^t_{p_k}\}_{\forall k \neq j}, a_i = p_j) \\ &\quad .\, p(\{\mathbf{x}^t_{p_k}\}_{\forall k \neq j} \mid a_i = p_j) \qquad (5) \\ &\propto\; p(\mathbf{x}^t_{p_j} \mid \widehat{\mathbf{x}}^{t-1}_{l_i}, a_i = p_j) \qquad (6)\end{aligned}$$

where the values of the last two terms in (5) can be shown to be common across all probability terms in (3) and hence not relevant when finding the arg max. Also, the first term in (5) can be assumed to be independent of $\{\mathbf{x}^t_{p_k}\}_{\forall k \neq j}$ given $a_i = p_j$, to give (6).

The RHS of (6) can be further evaluated by marginalizing over the true landmark location x_{l_i},

$$\begin{aligned}&p(\mathbf{x}^t_{p_j} | \widehat{\mathbf{x}}^{t-1}_{l_i}, a_i = p_j) \\ &= \int p(\mathbf{x}^t_{p_j} \mid \mathbf{x}_{l_i}, \widehat{\mathbf{x}}^{t-1}_{l_i}, a_i = p_j) \\ &\qquad .\, p(\mathbf{x}_{l_i} \mid \widehat{\mathbf{x}}^{t-1}_{l_i}, a_i = p_j)\, d\mathbf{x}_{l_i} \qquad (7) \\ &= \int N_{\mathbf{x}^t_{p_j}}(\mathbf{x}_{l_i}, \Sigma^t_{p_j}) \cdot N_{\mathbf{x}_{l_i}}(\mu^{t-1}_{l_i}, \Sigma^{t-1}_{l_i})\, d\mathbf{x}_{l_i} \qquad (8) \\ &\propto e^{-\frac{1}{2}(\mathbf{x}^t_{p_j} - \mu^{t-1}_{l_i})^T (\Sigma^t_{p_j} + \Sigma^{t-1}_{l_i})^{-1} (\mathbf{x}^t_{p_j} - \mu^{t-1}_{l_i})} \qquad (9)\end{aligned}$$

The left term in the integral in (7) is a generative model. It is the probability distribution over the observed global point locations given the true location of the landmark. For a Gaussian generative model it reduces as shown. The right term under the integral in (7) is the probability distribution over the true location given the parameters of the estimated Gaussian distribution and reduces to the estimated Gaussian distribution itself. Combining equations (6) to (9) we get,

$$\begin{aligned}&p(I_X \mid a_i = p_j) \propto \\ &\quad e^{-\frac{1}{2}(\mathbf{x}^t_{p_j} - \mu^{t-1}_{l_i})^T (\Sigma^t_{p_j} + \Sigma^{t-1}_{l_i})^{-1} (\mathbf{x}^t_{p_j} - \mu^{t-1}_{l_i})} \qquad (10)\end{aligned}$$

[3] We drop the superscript g in this section as all locations are in the global frame of reference unless otherwise stated.

We proceed in a similar fashion for the right term in (4) to get,

$$p(I_D \mid a_i = p_j) \propto e^{-\frac{1}{2}(\mathbf{d}^t_{p_j} - \mathbf{d}^t_{l_i})^T \Sigma_d^{-1} (\mathbf{d}^t_{p_j} - \mathbf{d}^t_{l_i})} \quad (11)$$

where Σ_d is a known set of constant parameters.

If we combine equations 3, 4, 10, and 11 and take the negative log we get that the point p^* most likely to be associated with l_i is given by,

$$p^* = \arg\min_{[p_j \in P_i]} (\mathbf{d}^t_{p_j} - \mathbf{d}^t_{l_i})^T \Sigma_d^{-1} (\mathbf{d}^t_{p_j} - \mathbf{d}^t_{l_i}) + (\mathbf{x}^t_{p_j} - \mu^{t-1}_{l_i})^T (\Sigma^t_{p_j} + \Sigma^{t-1}_{l_i})^{-1} (\mathbf{x}^t_{p_j} - \mu^{t-1}_{l_i}) \quad (12)$$

The first term on the RHS is the square of the Mahalanobis distance between the observed point location at t and estimated landmark location at $t-1$. This implies that points with a lower Mahalanobis distance to the landmark will be favored. The second term, which is also the square of a Mahalanobis distance, ensures that points with visual properties similar to that of the landmark are favored.

No Matches and New Landmarks. To consider the possibility of no point being associated with landmark l_i, we set a threshold on the maximum Mahalanobis distance allowed between the locations of a point and a landmark. To set the threshold, we think of the Mahalanobis distance as the Euclidean distance measured in units of standard deviation [2], and set the maximum distance, in standard deviation units, that a point can be from a landmark in order for a match to be considered (the threshold can be found by looking up a χ^2 distribution table). An added benefit is that points, that do not lie within the Mahalanobis distance threshold of *any* landmark, are used to initialize new landmarks.

Removing False Positives. As mentioned, newly initialized landmarks are temporary to begin with. Only after a minimum number of points match the landmark within a given time period, is the landmark made permanent. If enough matches are not found, the landmark is removed. This removes false positives from the set of observed points.

The "Mahalanobis Effect". False positives are effectively removed by the two constraints of minimum number of required point matches and maximum Mahalanobis distance between locations. Not using either of these, particularly the Mahalanobis metric, leads to unusable maps. Figure 4 shows rather dramatically what happens when, instead of the Mahalanobis distance, standard Euclidean distance is used as a metric. Both figures show a laser generated map of the lab overlaid with a map built using only SIFT features. In the SIFT feature map on the left, false positives are eliminated which results in the visual landmarks lining up well with the laser map. In the map on the right, Euclidean distance fails to eliminate false positives leading to an unusable visual feature map. It is possible to use a tighter threshold when working with Euclidean distance but then this leads to a map with very few landmarks that is not adequate for safety.

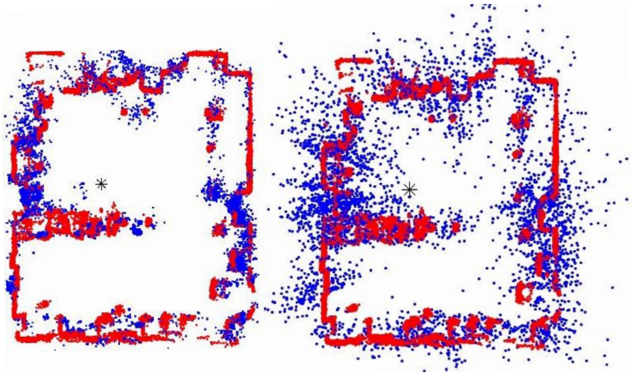


Figure 4. Left: Laser map (lines) overlaid with SIFT map (dots) made using Mahalanobis distance. Right: Laser map overlaid with SIFT map made without Mahalanobis distance.

Another implication of using the Mahalanobis distance is that it is possible to track landmarks using only location information. The use of visual properties such as scale and orientation for SIFT features may improve matching slightly but the final map is virtually indistinguishable. This is also apparent for the case of landmarks corresponding to pixels, where we don't have any visual properties and still get good results, even despite the fact that landmarks corresponding to pixels are denser than those corresponding to features. Although, the Mahalanobis metric is well known in robotics [1], and tracking literature [12], to the best of our knowledge this is the first application of the metric to tracking point landmarks corresponding to pixels, i.e., generated using dense stereo processing methods.

5. Results

We describe the results of testing our system in two environments: indoors in our lab, and outdoors on a wheelchair ramp. We describe (1) the creation of local safety maps using our system and their evaluation, (2) the quality of 3D reconstruction, and (3) a failure mode of the system.

First, we consider local safety maps of our lab. From left to right, Figure 5 shows four maps of the lab: (a) a local metrical map built using lasers, (b) a local metrical map built using dense stereo, (c) a local *safety* map built by merging and automatically annotating the laser and dense stereo maps, and (d) a human annotated safety map. For this particular data trace we do not show the SIFT feature map as the information in the dense stereo map subsumes that in the SIFT map (as opposed to that expected in section

4). However this may not be true in general and different features may provide extra data.

Figure 5(c) shows a local safety map of the lab annotated according to the specifications in section 2.2 and created as follows: (i) Obstacles: Parts of the lab detected by both lasers and vision. These are black in the figure and include walls and some furniture. (ii) Hazards: Parts of the lab invisible to lasers but detected by vision are treated as overhangs (i.e., hazards). They are dark grey in the figure. For example, the lower right hand corner of the figure has a table, whose top is treated as an overhang and legs are treated as obstacles (since the lasers can see them). Figure 6 shows in detail how the table is perceived differently by different sensors. (iii) Safe areas: Parts of the lab that are clear to the lasers and where vision sees the ground plane and detects neither obstacles nor hazards are considered safe. They are in white. (iv) Unknown areas: Parts of the world that are unknown to the lasers, and for which vision has no information are marked unknown. They are a light shade of grey and constitute most of the area outside the lab walls. At present we do not detect areas of caution.

To do a quantitative evaluation of the safety map, we have the robot collect a *new* data trace of the same environment and create another merged safety map. This new safety map is converted into a grid map, and a person manually annotates the grid map for safety. This human annotated map is taken to be ground truth (Figure 5(d)). We compare the laser metrical map, dense stereo map, and merged safety map, with the human annotated map. To compare the laser map and the dense stereo map we annotate them in a manner similar to that described for the merged safety map above. For the dense stereo map the process is almost the same except we don't distinguish between hazards and obstacles. For the laser map, since the lasers cannot see the ground, clear areas are considered safe.

We then determine the number of the following types of cells - A: True positives, i.e., cells marked safe by both humans and robots; B: False positives, i.e., cells marked safe by robots but unsafe by humans; C: False negatives, i.e., cells marked unsafe by robots but safe by humans. We measure three standard statistics: Precision = A/(A+B); Recall = A/(A+C); f1 = 2A/(2A+B+C). Precision gives the portion of the total number of cells that the robot says are safe, that are actually safe. Recall gives the portion of the total number of cells that are actually safe, that are marked safe by the robot. The f1 statistic considers both precision and recall and so is an overall measure of the systems performance. All three statistics lie between 0 and 1.

Table 2. Safety related statistics for the laser map, dense stereo map, and merged map.

Map	Precision	Recall	f1
Laser	0.78	0.99	0.87
Stereo	0.94	0.86	0.90
Merged	0.94	0.89	0.92

The results are in Table 2. Lasers have low precision most likely because they consider areas below overhanging objects to be safe. Lasers have higher recall than stereo vision because despite everything they still are more accurate sensors, with lower noise, than stereo vision. Because of higher noise vision considers some safe areas to be unsafe whereas lasers rarely do so. Vision has high precision because most obstacles and overhanging objects are visible to stereo. The merged map has high precision because stereo helps it see most objects and higher recall than vision because lasers help. Finally, the f1 statistic shows that the merged map performs best overall.

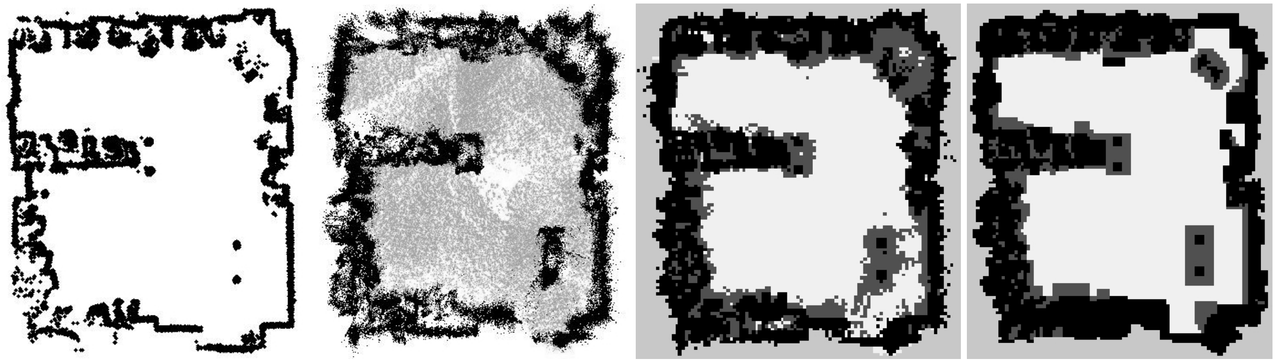
Figure 7 shows a picture of an outdoor wheelchair ramp and a 3D reconstruction of the ramp's railing using dense stereo. Qualitatively, the 3D reconstruction obtained is quite good and shows, amongst other things, the effectiveness of the Mahalanobis metric. Although not shown here due to lack of space, this environment also demonstrates a failure mode of our system. Due to low texture the dense stereo algorithm is not able to get reliable depth information (a common failing of many dense stereo methods) for the ground plane of the ramp. This illustrates that we cannot rely only on stereo vision for safety and need to use other visual cues, such as color.

6. Related Work

The idea of using Kalman filters to track visual features is well known. Harris [6], and, Se, et. al, [13], use independent Kalman filters to track visual features through the environment. Localization is done through a combination least squares and a separate Kalman filter to track robot pose. However, neither methods use dense stereo vision.

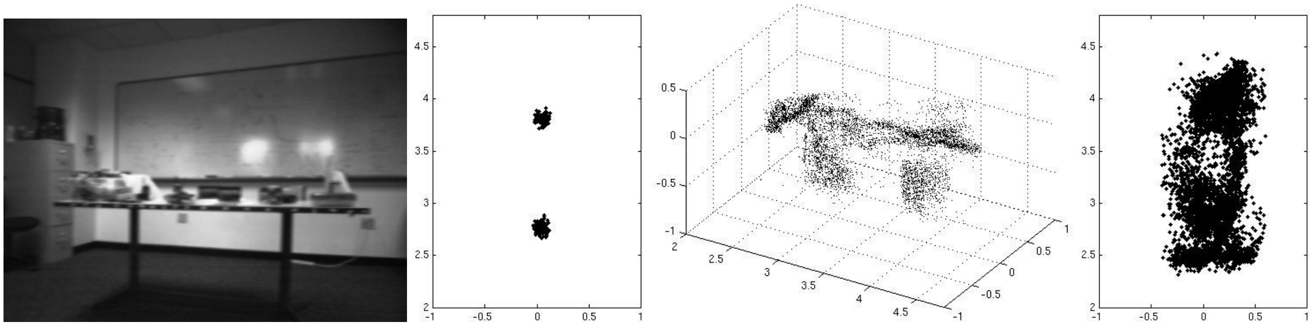
A problem that frequently accompanies Kalman filters is that of data association. Applying probabilistic methods (amongst them the Mahalanobis metric) for solving data association is well established, particularly in the tracking literature. For example, Reid [12], solves the data association problem associated with tracking multiple moving objects (e.g. airplanes) using radar (and other sensors) by maintaining multiple hypotheses and computing their likelihoods. In the robot mapping literature, Kalman filters and the Mahalanobis distance have been used for extensively for landmark based SLAM [1]. However, landmarks in such applications are usually sparse and the incidence of false positives is much lower. Recently, Sim, et. al, [14], have used Kalman filters for tracking SIFT features in a Rao-Blackwellized particle filter based SLAM framework.

Using dense stereo for building maps and safety has also been investigated. Murray and Little [11], present methods for building 2D occupancy grid maps of the world using



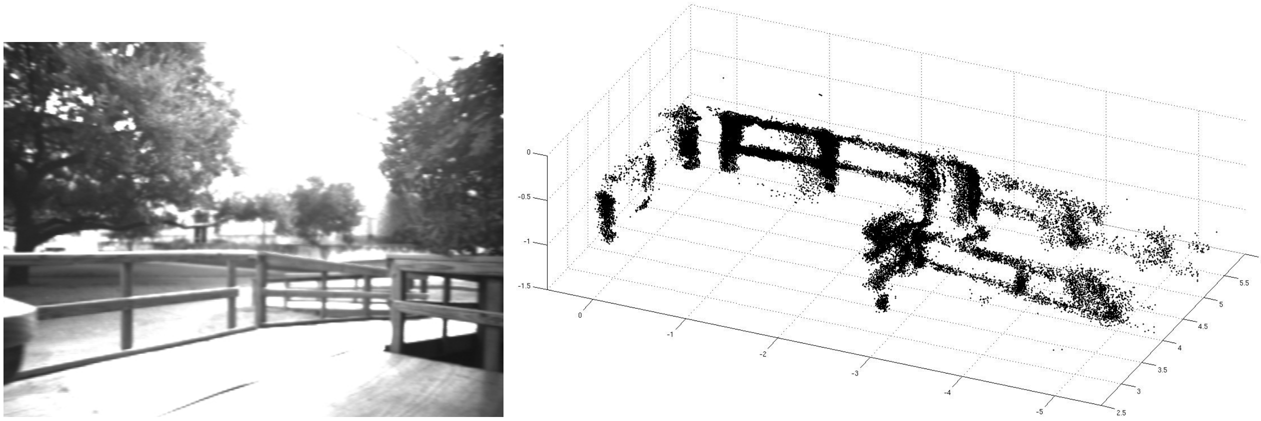
(a) (b) (c) (d)

Figure 5. Four maps of the lab: (a) Laser metrical map with obstacles in black. (b) Dense stereo metrical map with obstacles in black, ground plane in grey. (c) *Safety* map created by merging laser and dense stereo maps with obstacles in black, hazards in dark grey, unknown areas in light grey, safe areas in white. (d) Human annotated "ground truth" safety map.



(a) (b) (c) (d)

Figure 6. Different sensors perceive a table differently: (a) A picture of the table. (b) Lasers' perception of the table - only the table legs are seen. (c) 3D point cloud model of the table showing that a stereo camera sees almost the entire table. (d) Projection of the point cloud onto the 2D travel plane.



(a) (b)

Figure 7. (a) A picture of a wheelchair ramp. (b) 3D point cloud of ramp railing constructed using dense stereo.

dense stereo but in doing so, do not use a lot of the 3D information available. Gutmann, et. al, [5], create grid maps using dense stereo vision for a humanoid robot. Each cell in the grid is marked as floor or obstacle and annotated with the floor and obstacle heights. Ye and Borenstein [19], build a similar kind of elevation map using a 2D laser range-finder tilted towards the ground. The methods however do not recognize hazards (drop offs, overhangs) explicitly.

Since using stereo depth information is not sufficient, people have looked at other visual cues. Ulrich and Nourbakhsh [16], assume that their robot starts on safe ground and then use the color of terrain already traversed to classify pixels in new images as belonging either to the ground or to an obstacle. The method is simple yet effective.

7. Conclusions and Future Work

We make three main contributions in this work. First, we provide a definition of a local 2D safety map. Second, we present a hybrid method for building the 2D safety map and evaluate its performance in different environments. Third, we provide a new method for removing noise from dense stereo data using motion and demonstrate its effectiveness at 3D reconstruction. The planned application of this work is to build an autonomous robotic wheelchair which can navigate and explore urban environments in interaction with its human driver.

There are many ways in which this work can be extended. In the short term, future work will consist of speeding up the system and auto calibrating the sensors. Our current implementation works offline and further work will explore the use of efficient data structures to make the system work online. For calibration we intend to use techniques that take advantage of multiple sensors (such as encoders, lasers and vision) on a single robot to auto-calibrate the sensors against each other [20]. A longer term extension to this work will include using additional visual cues for safety - such as color [16]. We can also do deeper inference on the data provided by lasers and vision to obtain safety information for unknown areas. Finally, we plan to extend the work to environments where travel surfaces are non-level.

Acknowledgements. This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by grants from the National Science Foundation (IIS-0413257 and IIS-0538927), from the National Institutes of Health (EY016089), and by an IBM Faculty Research Award. The authors would like to thank the reviewers for their comments.

References

[1] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localisation and map building (slam) problem. *IEEE Tran. of Robotics and Automation*, pages 229–241, 2001.

[2] R. Duda, P. Hart, and D. Stork. *Pattern Classification (2nd Edition).* Wiley-Interscience, 2000.

[3] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* Prentice-Hall, 2002.

[4] A. Gelb. *Applied Optimal Estimation.* MIT Press, 1974.

[5] J.-S. Gutmann, M. Fukuchi, and M. Fujita. A floor and obstacle height map for 3d navigation of a humanoid robot. In *IEEE Intl. Conf. on Robotics and Automation*, 2005.

[6] C. Harris. Geometry from visual motion. *Active vision*, pages 263–284, 1993.

[7] L. Iocchi and K. Konolige. A multiresolution stereo vision system for mobile robots. In *AIIA (Italian AI Association) Workshop*, Padova, Italy, 1998.

[8] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli. Local metrical and global topological maps in the Hybrid Spatial Semantic Hierarchy. In *IEEE Intl. Conf. on Robotics and Automation*, 2004.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 60(2), 2004.

[10] L. Matthies and S. Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, 3(3):239–248, 1997.

[11] D. Murray and J. Little. Using real-time stereo vision for mobile robot navigation. In *Workshop on Perception for Mobile Agents at CVPR*, Santa Barbara, CA, 1998.

[12] D. Reid. An algorithm for tracking multiple targets. *IEEE Tran. on Automatic Control*, 24(6), 1979.

[13] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Intl. Journal of Robotics Research*, 21(8), 2002.

[14] R. Sim, P. Elinas, M. Griffin, and J. Little. Vision-based slam using the rao-blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR)*, Edinburgh, Scotland, 2005.

[15] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *IEEE Intl. Conf. on Robotics and Automation*, 2000.

[16] I. Ulrich and I. Nourbakhsh. Appearance-based obstacle detection with monocular color vison. In *Proc. of the AAAI National Conf. on Artificial Intelligence*, Austin, Texas, 2000.

[17] U.S. Department of Justice. ADA standards for accessible design. http://www.usdoj.gov/crt/ada/stdspdf.htm, 1994.

[18] Videre Design. http://www.videredesign.com, 2006.

[19] C. Ye and J. Borenstein. A new terrain mapping method for mobile robot obstacle negotiation. In *Proc. of the UGV Tech. Conf. at the 2003 SPIE AeroSense Symp.*, 2003.

[20] Q. Zhang and R. Pless. Constraints for heterogenous sensor auto-calibration. In *IEEE Workshop on Real-Time 3D Sensors and their Use*, 2004.