Vision-based Multi-Robot Simultaneous Localization and Mapping

Hassan Hajjdiab and Robert Laganière
VIVA Research Lab
School of Information Technology and Engineering
University of Ottawa, Ottawa, Canada, K1N 6N5
laganier@site.uottawa.ca

In this paper we present a vision-based approach for the multi-robot Simultaneous Localization and Mapping (SLAM) problem. We study the case of a team of robots equipped with a single camera and collaborating in the same worksite. We propose to calculate the location of the robots by using a collection of sparse views of the planar surface on which these robots are moving. The camera motions are estimated using inter-image homographies computed from the matching of overhead transformed views. Results of map generated from the estimated robot locations are presented.

Keywords: sparse view matching, robot localization, homography, camera pose estimation

1 Introduction

The ability of a mobile robot to localize itself in a worksite is a key prerequisite towards developing autonomous robots. This problem has gained considerable attention in the robotics community in the recent few years.

Many authors have proposed various approaches to estimate robot location with some a priori information about the environment. Krotkov [5] assumes that the robot has a map of its working environment. The map marks the positions of vertically oriented objects in the scene such as doors, desks, etc. The robot is then localized by establishing a correspondence between the landmark directions and points in the map. Sim and Dudek [17] use learned land marks to perform position estimation of a mobile robot. The landmarks are detected from a preliminary traversal of the environment and then saved in a database. The mobile robot position is estimated by matching the landmarks extracted from the image with the landmarks saved in the database. Dudek and Zhang [2] use multi-layer neural network to estimate the pose of the robot. The neural network is trained in the preliminary phase using a set of training images at known location and orientation. Werman et al. [10] introduced a robot localization method based on image measurements that are invariant to the intrinsic parameters of the camera. The measurements depend only on the extrinsic parameters i.e rotation and translation of the camera with respect to some world coordinate system.

A robot is said to be truly autonomous, if it has the ability to start at an unknown location in an unknown environment and then simultaneously build a map of the environment and localize it self in the map [3]. Thus the robot has to solve the simultaneous localization and mapping (SLAM) problem. The information used are sequences of relative observations captured by the mobile robot. Many approaches has been proposed to solve the SLAM problem, Smith and Cheeseman [18] use extended Kalman filters (EKF) to estimate the posterior distribution over the robot pose. Murphy [12] and Montemerlo et al. [9] introduced algorithms to solve the SLAM problem by integrating particle filter and Kalman filter representation. When a team of robots are sharing the same worksite, the SLAM problem becomes more challenging. The robots has to build a joint map of the environment and be able to localize their positions in the joint map in order to coordinate the navigation and minimize the overlap in information. Simmons et al. [14] presented a multi-robot SLAM algorithm based on likelihood maximization to find the maps that are maximally consistant with the sensor data. The exact initial pose of all robots relative to each other is assumed to be known. Thrun et al. [16] introduced similar approach but with known approximate initial pose of the robots (within 1 meter). Liu and Thrun [7] presented a Bayesian approach to solve the multi-robot SLAM problem assuming unknown initial positions and ambiguous land marks.

In this paper, we propose to solve the multi-robot SLAM problem by using a collection of sparse views of the scene. We assume no known initial relative pose of the robots and no known landmarks. Each robot in the team starts at an arbitrary unknown location and incrementally builds a local map of the environment with the ability to localize itself in the map. When an overlap occurs between any two robots, a joint map can be built between them and the two robots are able to localize themselves in the joint map for all their previous as well as future locations. Under this approach a joint map of the team can be built if each robot has at least one overlap with any other robot in the team.

In our algorithm we simply assume that each robot is equipped with a single camera and the robots are operating on a planar surface. We also assume that the height of the camera with respect to the ground plane as well as its orientation are known. Note that the tilt angle does not have to be accurate since this one is simply used as an initial approximation and will be re-estimated by the algorithm. However, a good accuracy in the height measure could be required since robot localization will be specified in terms of height units.

The rest of the paper is structured as follows. In Section 2, we present our matching algorithm. In Section 3, the inter-image homography calculation is presented. In Section 4, the multi-robot SLAM method is introduced. Section 5 presents some experimental results and, finally, Section 6 is a conclusion.

2 Sparse overhead view matching

In order to calculate the displacement between two robot positions, we have to match the images captured at these two locations. Based on the geometry of the system of cameras, it will then become possible to estimate the relative camera motion and therefore to accurately position the robots in their environment. The geometry of the system can be described as follows. The ground plane lies on the XZ plane and the optical axis of the camera is aligned with the Z axis. The camera, at a height h, and tilted by rotating it about the X axis by an angle α. In this case, the 3×3 homography matrix H_B of the projective relation, $[x, y, 1]^\intercal = H_B[X_W, Z_W, 1]^\intercal$, between the world plane and the corresponding overhead image point can be described as follows:

$$H_B = \begin{bmatrix} f & u_0 c_\alpha & u_0 h c_\alpha \\ 0 & f s_\alpha + v_0 c_\alpha & v_0 h c_\alpha - f h s_\alpha \\ 0 & c_\alpha & h c_\alpha \end{bmatrix} \quad (1)$$

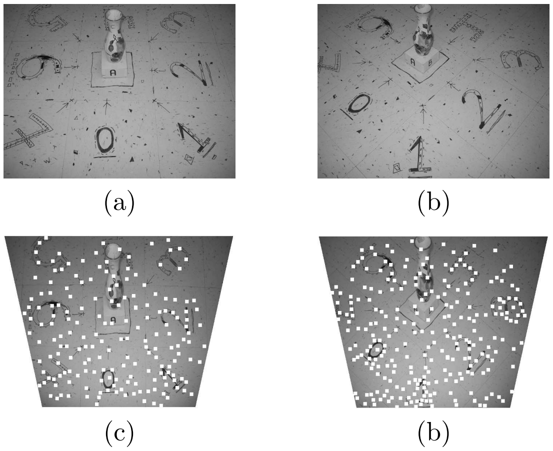
where f represents the focal length of the camera, u_0 and v_0 are the principal point coordinates, and with $c_\alpha = \cos\alpha$, $s_\alpha = \sin\alpha$. This transformation is invertible such that the overhead view can be generated from a perspective or vice versa. Figure 1 shows an example of images on which this kind of transformation has been applied to. It also shows that once features are detected on the original images, these ones are then mapped to their corresponding locations on the overhead views.

Indeed, in order to match two views of a scene, feature points must be detected. To this end, we used the Harris corner detector [4]. Working in the overhead view space has the virtue of eliminating the perspective distortion that deforms the visual patterns in each view. Consequently, to match points in these overhead views, only a rotationally invariant measure is required. Assuming that intensity variations due to the changes in viewpoints are not significant, differential invariants can give a robust characterization of the points of interest. Since we have at our disposal color images, we selected the Hilbert's color invariants, these having the advantage of requiring only first order derivatives. These derivatives are computed using Gaussian filters. Montesinos et al. [11] proposes to use an invariant vector of 8 such components. In our case, we just use the Gaussian filtered color intensities, $R_\sigma, G_\sigma, B_\sigma$, and their corresponding gradient magnitudes, $|\nabla_\sigma R|, |\nabla_\sigma G|, |\nabla_\sigma B|$.

In order to compute the similarity between two such vectors, the components must be resized since color intensities and gradient magnitudes have different ranges of values. Moreover, since the stability of the gradient magnitude over viewpoint variation is not very good, we normalize these values using a sigmoidal function, i.e.:

$$s(t) = \frac{1}{1 + e^{-\mu(t - t_o)}} \quad (2)$$

This non-linear normalization leads to a more qualitative characterization of the gradient where pixels are classified (in a fuzzy way) as point with low or high color gradient magnitude. The value t_o is the threshold that defines the point of transition between low and high magnitudes and μ controls the fuzzyness of the classification. The gradient magnitude thus transformed is similar to the measure of *edgeness* as defined in [20]. This leads us to the following



(a) (b)

(c) (b)

Figure 1: (a) and (b) Two images of the scene. (c) and (d) The overhead view transformations on which feature points have been mapped.

normalized invariant vector:

$$I^{\nu}(X) = \left[\frac{R_\sigma}{255}, \frac{G_\sigma}{255}, \frac{B_\sigma}{255}, s(|\nabla_\sigma R|), s(|\nabla_\sigma G|), s(|\nabla_\sigma B|) \right]^T \qquad (3)$$

Euclidean distance can then be used to measure the similarity between two color image points. However, because of their limited discrimination power, matching with invariants leads to several false matches. Therefore, we need to introduce an additional matching measure. The choice we made is based on the observation that while the transformation between two images of a plane is a general homography, the transformation between the two generated overhead views is an isometric transformation, i.e., it is composed of a rotation and a translation. The transformation has three degrees of freedom; two for translation and one for rotation, it can therefore be computed from two point correspondences. This isometric transformation H_S can be described as follows:

$$H_S = \begin{bmatrix} \cos\theta & -\sin\theta & T_x \\ \sin\theta & \cos\theta & T_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (4)$$

with θ being the angle between two cameras. An invariant in this transformation is the Euclidean distance between two points. Indeed, the line length between the two overhead views is preserved. Following a RANSAC-like scheme, we randomly select two match pairs and the length of the line segments that join the two selected points in each image is compared. If the difference in length is sufficiently low, then the points are considered to be good candidate matches. These candidate matches are then further validated by considering the intensity profiles of the candidate segments. Tell and Carlsson [19] also used a comparison between the affinely invariant Fourier features of the intensity profiles between randomly selected pairs of image interest points. We use a similar approach; however, in our algorithm, and because of the isometry that separates the two views, each line segment is simply divided into $k+1$ points. Cross correlation between the intensity profiles of the two lines is then computed as follows:

$$l_c = \frac{\sum_{i=0}^{k}[(A_i - \bar{A})(B_i - \bar{B})]}{\sqrt{[\sum_{i=0}^{k}(A_i - \bar{A})^2][\sum_{i=0}^{k}(B_i - \bar{B})^2]}} \qquad (5)$$

where arrays A and B contain the pixel intensity values of the $k+1$ points of the two segments (we used $k = 8$) . If the correlation coefficient l_c exceeds a given value then the two segments, and therefore their corresponding end points, are assumed to match.

The transformation between the two overhead view images can then be calculated using the resulting set of matches. A best-fit finds the best isometric transformation.

3 The inter-image homography

The above matching phase allows to obtain the geometric relation between two overhead views. Using the system geometry, we have also been able to compute the transformation between an image and its corresponding overhead view. We now would like to chain all these transformations in order to obtain the transformation that relates two perspective views of the work surface. That is to say that the transformation between two perspective views of a plane is also a homography and is given by:

$$H_{ij} = H_{Bi}^{-1} H_{Sij} H_{Bj} \qquad (6)$$

where H_{Bi} and H_{Bj} are the overhead homography transformations for views i and j respectively, and H_{Sij} is the overhead isometric transformation between views i and j. However, in order to obtain reliable camera position from H_{ij}, this one needs to be refined. This can be done by transferring, using H_{ij}, feature points in one view to the other and searching in a window of size $W \times W$ for the closest detected features. We also impose that the $\mathbf{I}^{\nu}$ difference between the two matched corners has to be below certain threshold. The matched corners thus obtained are then used to refine the homography H_{ij} [1].

4 Robot Localization

4.1 Estimating the camera motion

For a two-camera system, Tsai and Hung [15] showed that the inter-image planar homography can be decomposed as follows:

$$H_{ij} = K \left[R - \frac{Tn^{\mathtt{T}}}{d} \right] K^{-1} \tag{7}$$

where K is the intrinsic parameters of the camera. R is the rotation between the two cameras, n is the normal to the plane under consideration and T is the translation vector. Finally d is the distance from the camera to the ground. The matrix $\frac{Tn^T}{d}$ is therefore defined to be:

$$\frac{Tn^{\mathtt{T}}}{d} = \begin{bmatrix} 0 & \frac{-T_x \sin(\alpha_i)}{d} & \frac{T_x \cos(\alpha_i)}{d} \\ 0 & \frac{-T_y \sin(\alpha_i)}{d} & \frac{T_y \cos(\alpha_i)}{d} \\ 0 & \frac{-T_z \sin(\alpha_i)}{d} & \frac{T_z \cos(\alpha_i)}{d} \end{bmatrix} \tag{8}$$

The value $R - Tn^{\mathtt{T}}$, with d arbitrarily set to 1 and α being the camera tilt angle, is given by Equation (9) with $c_\theta = \cos\theta$ and $s_\theta = \sin\theta$.

If the homography H_{ij} between the two images is known then:

$$R - Tn^{\mathtt{T}} = K^{-1} H_{ij} K = \begin{bmatrix} a_i & b_i & c_i \\ d_i & e_i & f_i \\ g_i & h_i & i_i \end{bmatrix} \tag{10}$$

The tilt α_j of camera j can be calculated as $\alpha_j = \tan^{-1}(\frac{g_i}{d_i})$ and the angle between the two cameras is given by $\theta = \cos^{-1}(a_i)$. To calculate the rest of the parameters, consider the inverse of Equation (9) as follows:

$$[R - Tn^{\mathtt{T}}]^{-1} = [K^{-1} H_{ij} K]^{-1} = \begin{bmatrix} a_j & b_j & c_j \\ d_j & e_j & f_j \\ g_j & h_j & i_j \end{bmatrix} \tag{11}$$

The tilt α_i can be calculated as $\alpha_i = \tan^{-1}(\frac{g_j}{d_j})$. The translation vector with respect to the camera i, $T_i = [T_{x_i}, T_{y_i}, T_{z_i}]$ can be obtained by combining Equations (9) and (10). Similarly, the translation vector with respect to the camera j, $T_j = [T_{x_j}, T_{y_j}, T_{z_j}]$ can be obtained from Equation (11), a double solution is obtained as follows:

$$\begin{aligned} T_{x_{i1}} &= \tfrac{b_i + \sin(\theta)\cos(\alpha_i)}{\sin(\alpha_i)} \\ T_{x_{i2}} &= \tfrac{-c_i - \sin(\theta)\sin(\alpha_i)}{\cos(\alpha_i)} \end{aligned} \tag{12}$$

$$\begin{aligned} T_{y_{i1}} &= \tfrac{e_i - \cos(\theta)\cos(\alpha_j)\cos(\alpha_i) - \sin(\alpha_j)\sin(\alpha_i)}{\sin(\alpha_i)} \\ T_{y_{i2}} &= \tfrac{-f_i - \cos(\theta)\sin(\alpha_i)\cos(\alpha_j) - \cos(\alpha_i \sin(\alpha_j)}{\cos(\alpha_i)} \end{aligned} \tag{13}$$

$$\begin{aligned} T_{z_{i1}} &= \tfrac{h_i - \cos(\theta)\cos(\alpha_i)\sin(\alpha_j) + \cos(\alpha_j)\sin(\alpha_i)}{\sin(\alpha_i)} \\ T_{z_{i2}} &= \tfrac{-i_i + \cos(\theta)\sin(\alpha_j)\sin(\alpha_i) + \cos(\alpha_j \cos(\alpha_i)}{\cos(\alpha_i)} \end{aligned} \tag{14}$$

$$\begin{aligned} T_{x_{j1}} &= \tfrac{b_j + \sin(\theta)\cos(\alpha_j)}{\sin(\alpha_j)} \\ T_{x_{j2}} &= \tfrac{-c_j - \sin(\theta)\sin(\alpha_j)}{\cos(\alpha_j)} \end{aligned} \tag{15}$$

$$\begin{aligned} T_{y_{j1}} &= \tfrac{e_j - \cos(\theta)\cos(\alpha_i)\cos(\alpha_j) - \sin(\alpha_i)\sin(\alpha_j)}{\sin(\alpha_j)} \\ T_{y_{j2}} &= \tfrac{-f_j - \cos(\theta)\sin(\alpha_j)\cos(\alpha_i) - \cos(\alpha_j \sin(\alpha_i)}{\cos(\alpha_j)} \end{aligned} \tag{16}$$

$$\begin{aligned} T_{z_{j1}} &= \tfrac{h_j - \cos(\theta)\cos(\alpha_j)\sin(\alpha_i) + \cos(\alpha_i)\sin(\alpha_j)}{\sin(\alpha_j)} \\ T_{z_{j2}} &= \tfrac{-i_j + \cos(\theta)\sin(\alpha_i)\sin(\alpha_j) + \cos(\alpha_i \cos(\alpha_j)}{\cos(\alpha_j)} \end{aligned} \tag{17}$$

The translation vector is finally obtained by simply taking the average of these two solutions.

4.2 Locating the Robots

The localization problem is formalized as shown in Figure 2. Location are parameterized by the triplet $\Gamma = [\rho, \phi_1, \phi_2]$. Where ρ is the Euclidean distance between two robots, ϕ_1 is the angle of *Robot*2 with respect to *Robot*1 and ϕ_2 is the angle of *Robot*1 with respect to *Robot*2.

The robot locations with respect to one another may be expressed by projecting the two 3D camera coordinate systems on *Robot*1 and *Robot*2 2D coordinate systems. The location of *Robot*2 with respect to *Robot*1 is at (x_1, y_1) defined as follows:
$x_1 = -T_{x_i}$
$y_1 = \sqrt{T_{y_i}^2 + T_{z_i}^2} \sin(\alpha_i + \beta_1)$
Where $\beta_1 = \tan^{-1} \frac{T_{y_i}}{T_{z_i}}$
Similarly, the location of *Robot*1 with respect to *Robot*2 is at (x_2, y_2) defined as follows:
$x_2 = -T_{x_j}$
$y_2 = \sqrt{T_{y_j}^2 + T_{z_j}^2} \sin(\alpha_j + \beta_2)$
Where $\beta_2 = \tan^{-1} \frac{T_{y_j}}{T_{z_j}}$
Finally, the location vector Γ is defined as follows:

$$\Gamma = [\rho, \phi_1, \phi_2] = \left[\sqrt{x_1^2 + y_1^2}, \tan^{-1} \frac{y_1}{x_1}, \tan^{-1} \frac{y_2}{x_2} \right] \tag{18}$$

For example, the images in Figure 1 were taken with the camera set manually to a tilt of 33^o, the height of the camera from the ground plane is $80cm$.

$$R - Tn^T = \begin{bmatrix} c_\theta & -s_\theta c_{\alpha_i} + T_{x_i} s_{\alpha_i} & -s_\theta s_{\alpha_i} - T_{x_i} c_{\alpha_i} \\ c_{\alpha_j} s_\theta & c_\theta c_{\alpha_j} c_{\alpha_i} + s_{\alpha_j} s_{\alpha_i} + T_{y_i} s_{\alpha_i} & c_{\alpha_j} s_{\alpha_i} c_\theta - c_{\alpha_i} s_{\alpha_j} - T_{y_i} c_{\alpha_i} \\ s_{\alpha_j} s_\theta & s_{\alpha_j} c_\theta c_{\alpha_i} - c_{\alpha_j} s_{\alpha_i} + T_{z_i} s_{\alpha_i} & c_\theta s_{\alpha_j} s_{\alpha_i} + c_{\alpha_j} c_{\alpha_i} - T_{z_i} c_{\alpha_i} \end{bmatrix} \tag{9}$$

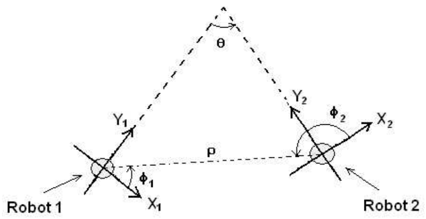


Figure 2: The Robot angles.

The homography between first and the second images H_{ij} is calculated as describe is Section 2 , the parameters α, θ and T expressed in height units were calculated as described in Section 4 to be as follows: $\alpha = 34.3$, $\theta = 43^o$ and $T = [-0.66, 0.21, 0.14]$. The location triplet Γ was calculated to be $\Gamma = [52cm, 22.7^o, 158.5^o]$.

5 Experimental Results

5.1 Single-Robot SLAM

Figure 3(a) shows a set of images collected by a robot moving on a planar surface. First, the matching algorithm discussed in Section 2 is applied and the inter-image homographies are calculated between consecutive robot locations. These homographies are used to incrementally built a location graph by calculating the robot pose with respect to its previous position as discussed in Section 4; the resulting location graph is shown in Figure 3(b). To estimate the accumulated error in estimating the pose between consecutive location, the pose between the initial location and last location is estimated. The error in estimated the distance between *Robot*0 and *Robot*7 can be estimated by $d_j = |r_1 - r_2|$ as shown in Figure 3(b). The value of d_j is $11.5cm$ for a total displacement error of 2.95%.

The global overhead view map of the environment can be built by combining the overhead transformation and the inter-image homographies [6] as shown in Figure 4(a). Due to the parallax effect [13] points above the ground plane are not mapped correctly (see the 'vase' on Figure 4(a)). To built a map that shows only the ground points, we use the following procedure. For each point on the ground plane, the appropriate transformation is applied to obtain the corresponding image point in each view. The mean RGB value is then computed and the image point that deviates the most from this mean value is discarded. This procedure is repeated until half of the image points have been discarded. The mean RGB value of the remaining points is then used in the mosaic composition. If a sufficient number of widely distributed views is used, this algorithm should eliminate the images of the obstacles from the ground model, the map obtained is shown in Figure 4(b).

5.2 Multi-Robot SLAM

In this section we provide an example of two mobile robots *RobotA* and *RobotB* moving in the same work site. The cameras tilt on robot *RobotA* and *RobotB* are set manually to 33^o and 45^o respectively, the height of the camera from the ground plane is $55cm$. Figure 5 shows 4 of the 13 the images captured by *RobotA* and Figure 6 shows 4 of the 13 images captured by *RobotB*. The two robots start at arbitrary unknown locations (location $A0$ for *RobotA* and location $B0$ for *RobotB*). Each of the robots starts building its relative location graph and a local map of the environment as discussed on the single-robot SLAM case. The obtained locations graphs are shown in Figure 7(a) and (b). When an overlap occurs between the two robots, a joint location graph and environment map can be built by calculating the relative pose of the two robots. At location $A7$ for *RobotA* and location $B9$ for *RobotB* the two robots are viewing the same scene, this overlap can be verified by comparing image 7 in Figure 5 and image 9 in Figure 6. The relative pose of the two robots is estimated by the inter-image homography between the two images and a joint location graph can be built. Figure 7(c) shows the joint location graph with relates all the robot locations with $A7$ selected as the reference frame. The joint map of the environment is shown in Table 1.

6 Conclusions

We have presented a technique for the multi-robot SLAM problem. The different robot locations are computed by finding the transformations that relate together the captured images of the scene. To overcome the difficulty that represents the matching of the available sparse views, we have proposed to first transform the images into overhead views; this

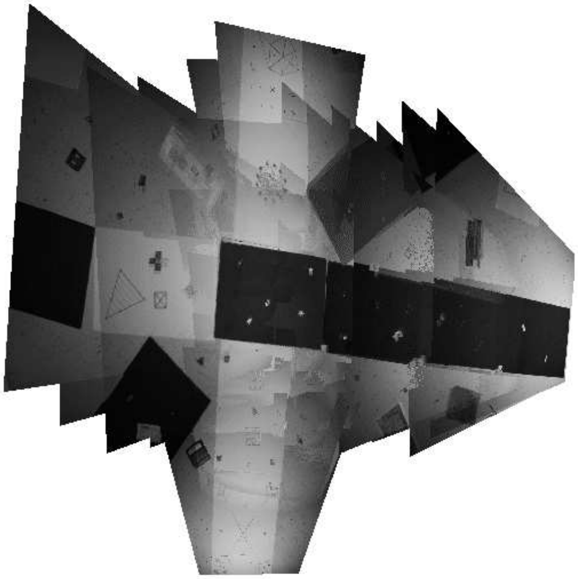
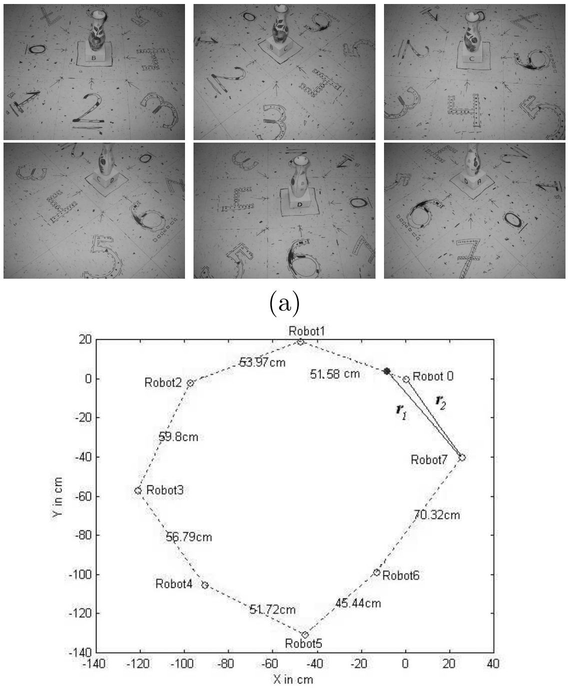


Table 1: The map generated from images collected by *Robot A* and *Robot B*.



(a)

(b)

Figure 3: Single-robot SLAM (a) Image set collected by the robot (b) Location graph.

transformation being obtained from the geometry of the system. A first estimate of the inter-image homography is obtained by using a best-fit approach. This one is then refined by aligning detected image features based on gaussian intensity differences. The camera motions are estimated from the computed homographies and the robots are localized. This allowed us to build maps of the environment inside which the robots are evolving.

References

[1] I. Reid A. Crismini and A. Zisserman. A plane measuring device. *Proc of BMVC*, 1997.

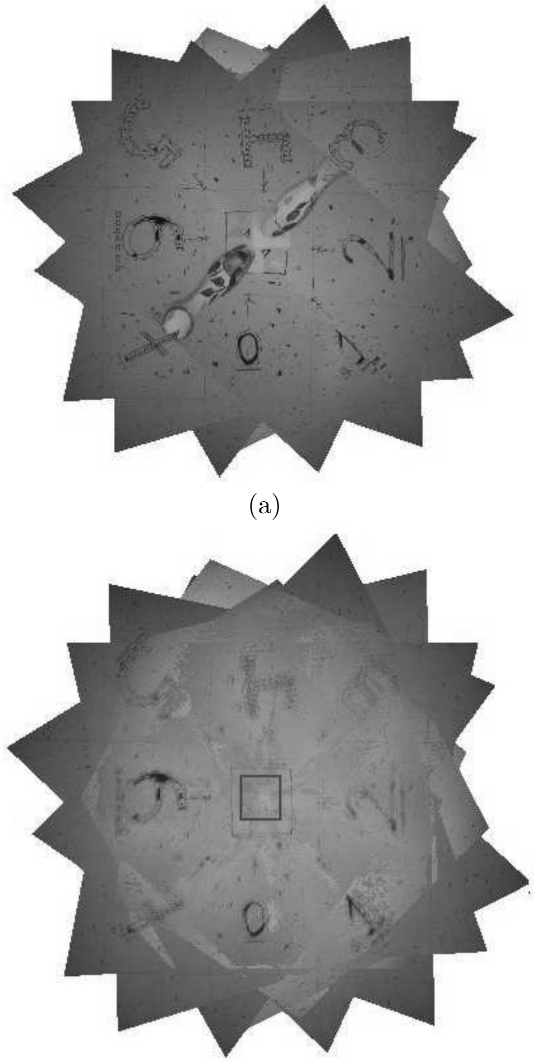
[2] G. Dudek and C. Zhang. Vision-based robot localization without explicit object models. *Int. Conf. on Robotics and Automation*, 1996.

[3] H.F. Durrant-Whyte S. Clark M. Csobra G. Dissanayake, P. Newman. A solution to the simultaneous localisation and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[4] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey vision Conf*, pages 147–151, 1988.

[5] E. Krotkov. Mobile robot localization using single image. *in Proc. 1989 IEEE Int. Conf. on Robotics and Automation*, pages 978–983, 1989.

[6] R. Laganière. Composing a bird's eye view mosaic. *Vision Interface*, pages 382–386, May 2000.



(a)

(b)

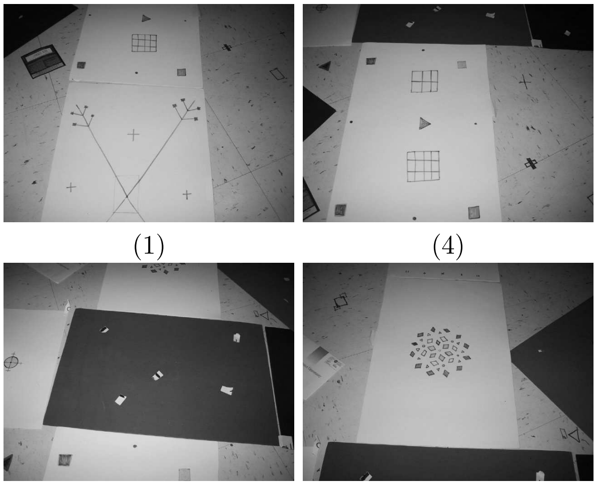
Figure 4: single-robot SLAM (a) generated map of the site (b) map with obstacle images eliminated.

[7] Y. Liu and S. Thrun. Gaussian multi-robot slam. *submitted to NIPS*, 2003.

[8] M.I.A Lourakis and S.C. Orphanoudakis. Visual detection of obstacles assuming a locally planar ground. *In Proc. 3rd ACCV*, 2:527–534, 1998.

[9] D. Koller M. Montemerlo, S. Thrun and B. Wegbreit. Fastslam2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *IJCAI*, 03.

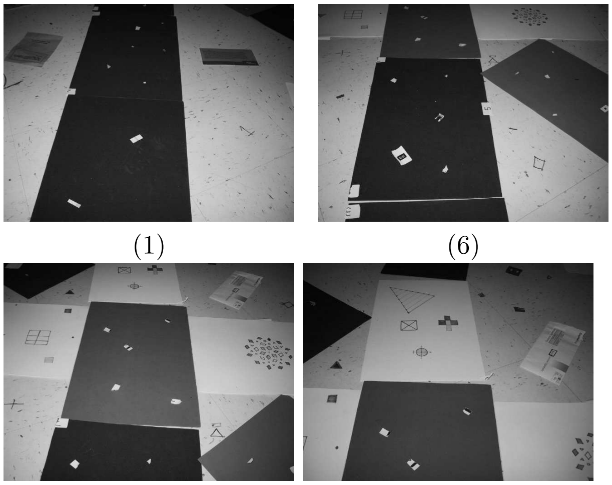
[10] S.D. Roy M. Werman, S. Banerjee and M. Qiu. Robot localization using uncalibrated camera invariants. *CVPR*, pages 353–359, 1999.



(1) (4)

(7) (10)

Figure 5: Four images of the set collected by the *Robot A*.



(1) (6)

(9) (11)

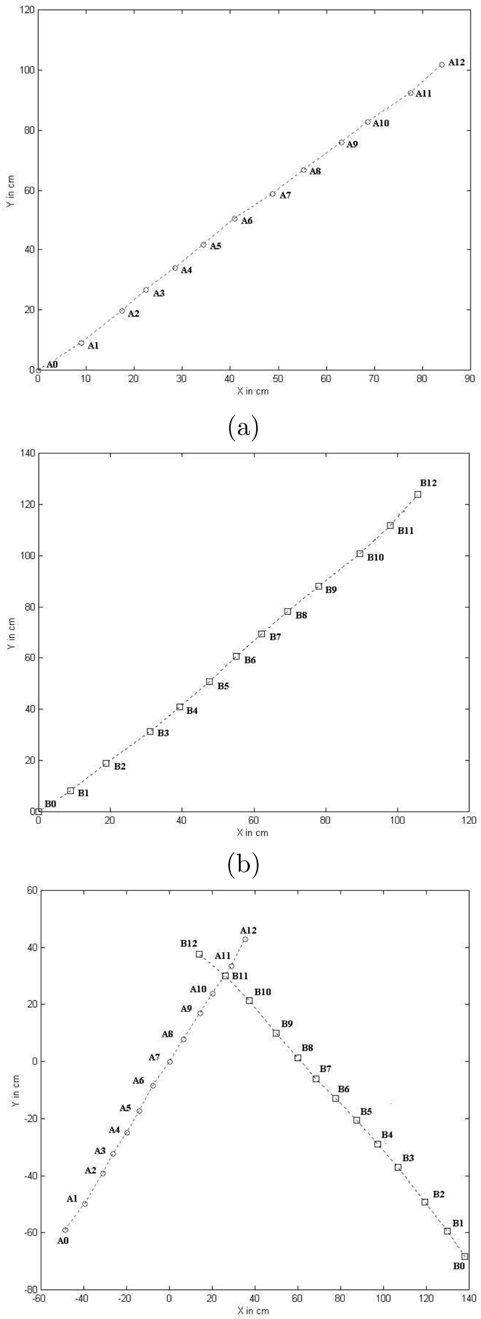
Figure 6: Four images of the set collected by the *Robot B*.

[11] P. Montesinos, V. Gouet, R. Deriche, and D. Pel. Matching color uncalibrated images using differential invariants. *Image and Vision Computing*, 18(9):659–672, June 2000.

[12] K. Murphy. Bayesian map learning in dynamic environments. *NIPS*, 99.

[13] P. Anandan R. Kumar and K. Hanna. Direct recovery of shape from multiple views. *In Proc 12th ICPR*, 555:111–222, 1994.

[14] W. Burgard-M. Fox D. Moors S. Thrun R. Simmons, D. Apfelbaum and H. Younes. Coordina-



(a)

(b)

(c)

Figure 7: Multi-Robot Localization: (a) location graph generated by *RobotA* (b) location graph generated by *RobotB* (c) the joint location graph

tion for multi-robot exploration and mapping. *AAAI*, 00.

[15] T.S. Huang R. Y. Tsai and W.L. Zhu. Estimating three-dimentional motion parameters of a rigid planar patch. *IEEE ASSP*, 30(4):525–534, Aug 1982.

[16] W. Burgard S. Thrun and D. Fox. A real-time algorithm for mobile robot mapping with application to multi-robot and 3d mapping. *ICRA*, 00.

[17] R. Sim and G. Dudek. Mobile robot localization from learned landmarks. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems (IROS)*, Victoria, BC, Oct. 1998.

[18] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *Int. J. Robotics Research*, 5(4), 1986.

[19] D. Tell and S. Carlsson. Wide baseline point matching using affine invariants computed from intensity profiles. In *ECCV*, pages 814–828, 2000.

[20] J. Weng, N. Ahuja, and T. Huang. Two-view matching. In *Int. Conf. on Computer Vision*, pages 65–73, 1988.