We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Multi-Robot SLAM: A Vision-Based Approach

Hassan Hajjdiab[1] and Robert Laganiere[2]
[1]*Abu Dhabi University*
[2]*University of Ottawa*
[1]*Abu Dhabi, United Arab Emirates*
[2]*Ottawa, Canada*

1. Introduction

The robot is said to be truly autonomous (Dissanayake et al. (2001)), 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. 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.

Se et al. (2001) proposed an approach for a robot equipped with a trinocular stereo system (Murray & Little (1998)) and an odometer. The algorithm starts by detecting feature points (Lowe (1999)) in the three images. Then these feature points are matched using the epipolar and disparity constraints thats exist between the three cameras. Assuming known camera intrinsic parameters, the 3D position of each matched feature point is estimated. As the robot moves, the odometry readings are used to provide a rough estimate about the motion. This estimate can be employed in matching feature points among consecutive frames. The 3D position of each newly detected feature point is estimated and the motion of the robot is localized using a least-squares minimization scheme (Lowe (1991)). Finally, a map is built for the 3D positions of the detected feature points and the location graph of the robot is computed. Other SLAM methods are based on evaluating some probabilistic models of the robot motion and sensed data from the environment. It is assumed that the robot can sense landmarks relative to its local coordinate frame. The landmarks may be naturally occurring in the environment like trees or artificially added like steel poles.

Smith & Cheeseman (1986) use extended Kalman filters (EKF) to estimate the posterior distribution over the robot pose. The problem they solved can be stated as follows: Given a measurement of the environment $\mathbf{z}^t = [z_1, z_2, ..., z_t]$ and a set of control inputs $\mathbf{u}^t = [u_1, u_2, ..., u_t]$, determine the robot pose $\mathbf{s}^t$ and the location of all the k landmarks $\mathbf{m} = [m_1, m_2, ..., m_k]$. In probabilistic terms, this can be expressed as the posterior:

$$p(\mathbf{s}^t, \mathbf{m}/\mathbf{z}^t, \mathbf{u}^t) \qquad (1)$$

where the superscript ^{t} refer to a set of variables at time t.

The extended kalman filters (EKFs) are relatively slow when estimating maps with very large number of landmarks. Murphy (99) and Montemerlo et al. (03) proposed more capable approach. They proposed algorithms to solve the SLAM problem by integrating particle filter and Kalman filter representation to solve the posterior function in Equation(1).

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.

Thrun et al. (2000) presented a multi-robot SLAM algorithm based on likelihood maximization to find the maps that are maximally consistent with the sensor data. The sensor data used are laser scanners to sense the environment and odometers to estimate the robot motion. The exact initial pose of all robots relative to each other is assumed to be known. The localization process starts as follows: Each robot collects a sequence of its own odometry and sensor measurements (like laser scans). Using these measurements, each robot incrementally constructs a maximum likelihood estimate for its position and a maximum likelihood estimate for the location of surrounding objects and a posterior function to determine its location in the map. To build a joint map, the posterior estimation component is used. The relative location of robots is unknown; however, each robot in the team starts within a map of a specific robot called the team leader. Using the posterior estimation, each robot localizes it self in the team leader's map and thus a unified map for the team is built.

Simmons et al. (2000) introduced similar approach but with known approximate initial pose of the robots (within 1 meter distance and 20° orienation). Each robot in the team incrementally constructs the likelihood and posterior estimates as in the approach proposed by Thrun et al. (2000).

However, to build a joint map, each robot in the team sends its sensed data (laser range scans and odometer readings) to the team leader. Since the initial approximate pose of each robot is known, the team leader localizes the robots relative to each other.

When the initial pose of the robots is unknown, an important question is raised:

> *If two robots in the team sensing similar data in the environment, are they actually sensing the same part of the environment or are they sensing different parts that look alike?*

Liu & Thrun (2003) presented an approach to solve the multi-robot SLAM problem assuming unknown initial positions and ambiguous landmarks. Each robot in the team builds a local map similar to ones discussed in Thrun et al. (2000) and Simmons et al. (2000). The joint map is built by fusing the maps acquired by the robots into one map. First, the landmarks between different local maps are matched. Correspondences found in this matching process provide an estimate of the rotation and translation between local maps. Using this estimate, a global map of the environment may be built by estimating a posterior similar to the one defined in Equation(1) evaluated over all robot poses and all landmarks from all available data. Liu and Thrun applied the algorithm using a single vehicle equipped with laser range finder and an odometer in an outdoor environment. Features in the map are the stems of trees detected by the laser range finder. The multi-robot case is demonstrated by splitting the data acquired by the vehicle into 8 disjoint sequences and then the proposed multi-robot SLAM algorithm is applied.

In this chapter, we propose to solve the multi-robot SLAM problem by using a collection of sparse views of the scene. The originality of our approach is that it uses purely vision sensors (single camera on each robot) and that no special landmarks are assumed to exist in the environment. Moreover, no assumptions are made on the initial relative pose of the robots. The robot location is estimated from sparse views, this makes the estimation much more accurate. In contrast, method based on small incremental displacements are more affected by noise and inaccuracies and are thus more subject to error accumulation. In our proposed
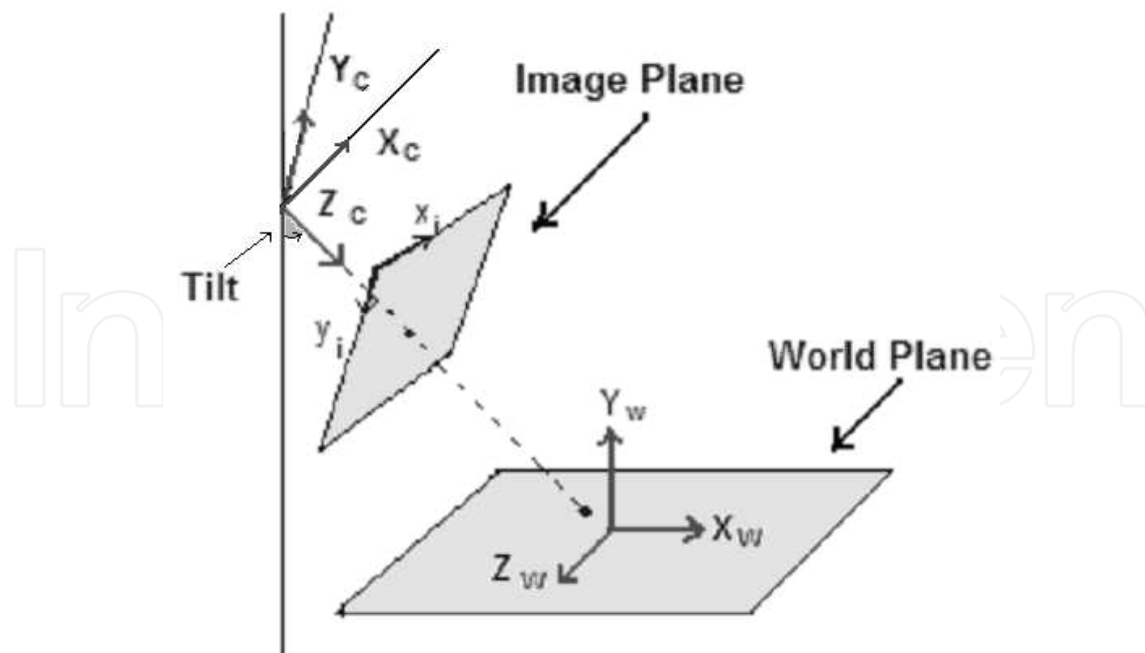
Fig. 1. Geometry of the system.

algorithm, 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 without the need for a new overlap. 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 approach 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.

2. Robot localization

2.1 Estimating the camera motion

In our work, the overhead views are used to match images and get the homography. The matching is done without depending on known patterns, moving objects on the plane or manual registration, instead the algorithm is based on features that have to be matched between views. However when a plane is observed by a tilted camera the features lying on it are subject to important perspective deformations. In the context of sparse views, this makes the features difficult to match using the usual correlation schemes. Applying the overhead view transformation will undo the perspective deformation which will make the apparent features differing only by a similarity transformation.

Figure 1 shows the structure of the coordinate systems of the world, camera and image coordinate systems. The ground plane surface is at $Y = 0$, the optical axis of the camera is aligned with the Z axis and the camera is rotated around the X axis by an angle α. The camera height from the world plane is h.

(a)   (b)

Fig. 2. Overhead view: (a) The image of a planar surface taken with tilt = 33°, (b) The corresponding generated overhead view.

The projective relation between the world plane and the corresponding overhead image point can be represented as follows:

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \mathbf{H}_O \begin{bmatrix} X_W \\ Z_W \\ 1 \end{bmatrix} \tag{2}$$

When the geometry of the system is as shown in Figure 1, the 3×3 homography matrix $\mathbf{H}_O$ can be described as follows (Laganière (2000)):

$$\mathbf{H}_O = \begin{bmatrix} f & x_0\cos(\alpha) & x_0 h\cos(\alpha) \\ 0 & f\sin(\alpha) + y_0\cos(\alpha) & y_0 h\cos(\alpha) - fh\sin(\alpha) \\ 0 & \cos(\alpha) & h\cos(\alpha) \end{bmatrix} \tag{3}$$

where f represents the focal length, x_0 and y_0 are coordinates of the principal point, α is the tilt angle and h is the height of the camera.

This transformation is inversible such that the overhead view can be generated from a perspective or vice versa. Figure 2(a) shows an image taken with a tilt = 33°, the corresponding overhead view image is shown in Figure 2(b).

2.2 Overhead view mosaic

In this section the composition of overhead view mosaics is discussed. The method is based on estimating the homography transformation of the set of images with respect to a certain plane in the scene (for example the ground plane). The overhead view mosaic is finally built by rectifying the images using an overhead transformation with respect to a reference image. One of the images (say image 0) is selected as a reference frame. The homographic transformation between the reference view 0 and a view i is:

$$\mathbf{x}_i = \mathbf{H}_{0i}\mathbf{x}_0 \tag{4}$$

For known camera parameters, the image to plane homography $\mathbf{H}_O$ between the reference view (view 0) and the ground plane can be calculated as described in Equation(3). The two matrices $\mathbf{H}_{0i}$ and $\mathbf{H}_O$ can be used to calculate the homography transformation between a view

i and the composted overhead view as follows:

$$\mathbf{H}_{iO} = \mathbf{H}_O^{-1}\mathbf{H}_{0i}^{-1} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (5)$$

Armed with all these homographic transformations, it is possible to produce an overhead mosaic representing the environment under study. When combining the information coming from different images, two strategies can be considered. The first one consists in determining the value of each pixel in the mosaic by selecting the pixel from the source image that has the best resolution as described by Laganière (2000). For a world point $[X, Y, 1]^T$ the mosaic pixel is selected from the source image that best samples the segment between points $[X, Y, 1]^T$ and $[X + \Delta_X, Y + \Delta_Y, 1]^T$ where Δ_X and Δ_Y are small increments in the X and Y directions respectively. The relation between a point $[X, Y, 1]^T$ in world plane and a point $[x, y, 1]^T$ in the image plane is:

$$X = h_X(x,y) = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \quad (6)$$

$$Y = h_Y(x,y) = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \quad (7)$$

The image resolution is determined by a measure called *instantaneous sampling rate*. For a world point $[X, Y, 1]^T$ the sampling rate at the X direction is the number of source images between points $[X, Y, 1]^T$ and $[X + \Delta_X, Y, 1]^T$ divided by the distance between them. The instantaneous horizontal sampling rate is as follows:

$$s_X = \sqrt{\left(\frac{\delta h_X(x,y)}{\delta x}\right)^2 + \left(\frac{\delta h_X(x,y)}{\delta y}\right)^2} \quad (8)$$

Similarly the instantaneous vertical sampling rate is:

$$s_Y = \sqrt{\left(\frac{\delta h_Y(x,y)}{\delta x}\right)^2 + \left(\frac{\delta h_Y(x,y)}{\delta y}\right)^2} \quad (9)$$

The instantaneous sampling rate is defined as follows:

$$s = s_X s_Y \quad (10)$$

The mosaic pixel at point (x,y) is selected from the source image that have the highest sampling density s.

Figure 3 shows a set of images of a ground plane scene; the reference image is Figure 3(a). Figure 4(a) shows the mosaic obtained for this set of images. As it can be seen, when obstacles are visible in the source image, these ones interfere with the ground plane representation. In order to cope with this problem, we have to discard the source pixels that contains intensities that belong to the obstacles. To do so, we propose the following algorithm:
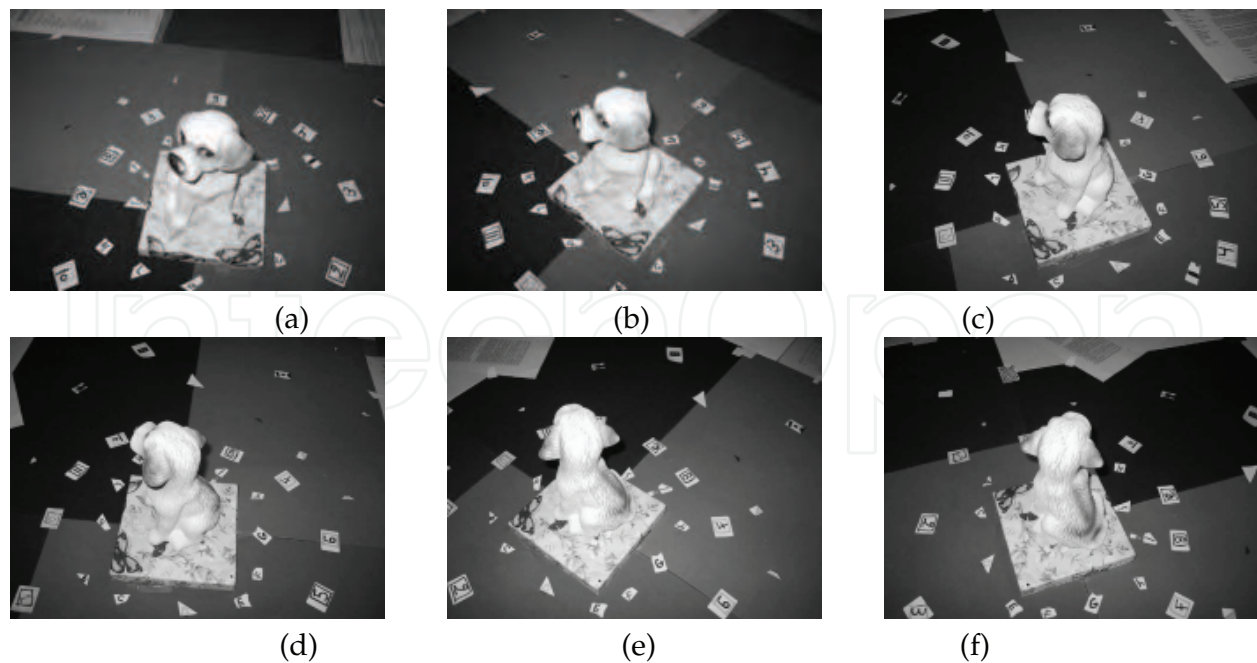
Fig. 3. Set of images of a ground plane scene.

- **Step 1**: For each point on the ground plane apply the appropriate transformation to obtain the corresponding image point in each view.
- **Step 2**: Compute the mean RGB value of the pixels and the pixel that deviates the most from this mean value is discarded.
- **Step 3**: Repeat steps 1 and 2 until half of the points are discarded.
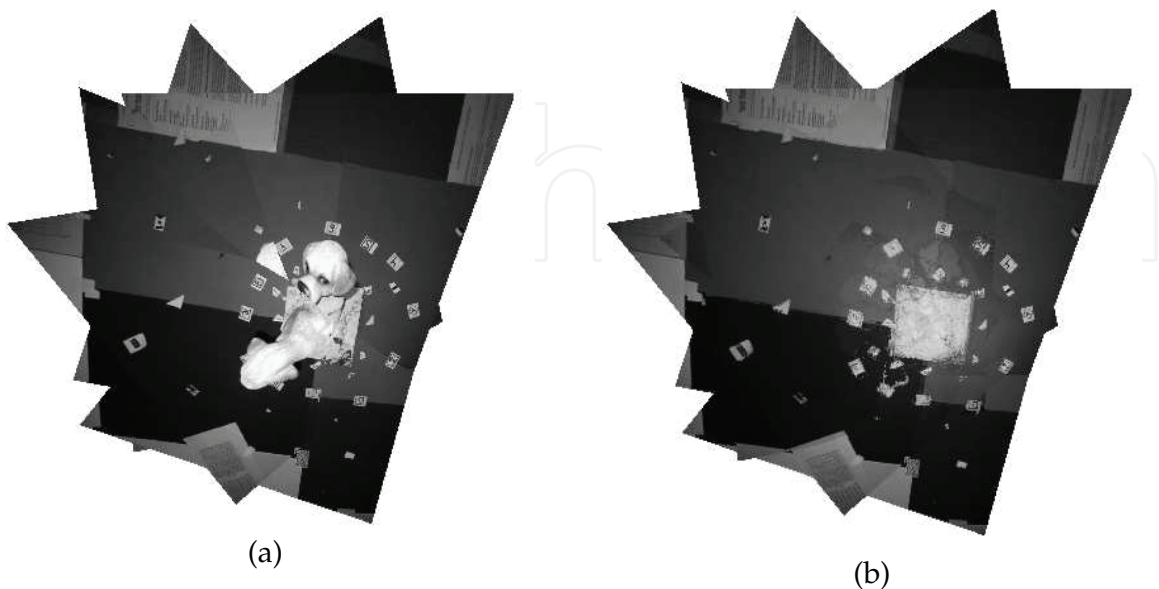- **Step 4**: The mean RGB value of the remaining points is then used in the mosaic composition.

Fig. 4. Mosaic of the set of images in Figure 3 : (a) selecting points with best resolution. (b) discarding points that belong to the obstacle.

The result obtained using this algorithm is shown in Figure 4(b). An appropriate representation is thus obtained that could be used as a ground plane model of the environment. We show in the next section how it is possible to obtain the inter-image transformations ($\mathbf{H}_{ij}$) necessary to build such overhead view mosaic. This is done by matching the different available views, however taken from widely separated viewpoints.

2.3 Matching two widely separated views

The idea of this proposed technique is to use the overhead view images in the matching process. The procedure first starts by detecting corners on the original perspective images (Figure 5 (a) and (b))and the corners are then mapped to the overhead view images(Figure 5 (c) and (d)).

The two views in Figure 5 (a) and (b) are subjected to a projective deformation which makes image matching more challenging and subject to false matches. However, the two corresponding overhead views in Figure 5 (c) and (d) are only related by a similarity transformation (i.e. translation and rotation). This makes matching corner points using the overhead views more favorable. The matching process can be summerized as follows:

- **Step 1**: detect corners on the two images to be matched (Figure 5 (a) and (b))

- **Step 2**: use Equation 3 to calculate the overhead views of the two images (Figure 5 (c) and (d))

- **Step 3**: Match overhead views to calculate the similarity transformation $H_{S_{ij}}$

- **Step 4**: Homography transformation between (Figures 5 (a) and (b)) can be calculated as: $H_{ij} = H_{Oi}^{-1} H_{S_{ij}} H_{Oj}$

The reader is referred to Hajjdiab & Laganière (2004) for more details of the matching process. The inter-image homography transformation H_{ij} for two cameras C_i and C_j viewing a planar scene can be expressed as:

$$\mathbf{H}_{ij} = \mathbf{K}_j \left[\mathbf{R} - \frac{\mathbf{tn}^T}{d} \right] \mathbf{K}_i^{-1} \tag{11}$$

where $\mathbf{K}_i$ and $\mathbf{K}_j$ are the matrices containing the intrinsic parameters of cameras C_i and C_j respectively, $\mathbf{R}$ is the rotation between the two cameras, $\mathbf{n}$ is the normal to the plane under consideration and $\mathbf{t}$ is the translation. Finally d is the distance from the camera to the ground. The optical axis of the two cameras are along the z-axis as described in Figure 1, then the rotation between the two cameras can be expressed as follows (Altmann (1986)):

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta)\cos(\alpha_i) & -\sin(\theta)\sin(\alpha_i) \\ \cos(\alpha_j)\sin(\theta) & \cos(\theta)\cos(\alpha_j)\cos(\alpha_i) + \sin(\alpha_j)\sin(\alpha_i) & \cos(\alpha_j)\sin(\alpha_i)\cos(\theta) - \cos(\alpha_i)\sin(\alpha_j) \\ \sin(\alpha_j)\sin(\theta) & \sin(\alpha_j)\cos(\theta)\cos(\alpha_i) - \cos(\alpha_j)\sin(\alpha_i) & \cos(\theta)\sin(\alpha_j)\sin(\alpha_i) + \cos(\alpha_j)\cos(\alpha_i) \end{bmatrix} \tag{12}$$

where α_i and α_j are the angle of cameras C_i and C_j with respect to the ground plane.
The normal to the ground plane $\mathbf{n}$ can be expressed with respect to camera C_i as follows:

$$\mathbf{n} = \begin{bmatrix} 0 \\ -\sin(\alpha_i) \\ \cos(\alpha_i) \end{bmatrix} \tag{13}$$

The motion parameters can be estimated using SVD decomposition as proposed by Triggs (1998) as follows:

Let the coordinates of camera C_i be the reference frame, the projection matrices of cameras C_i and C_j are respectively:

$$\mathbf{P}_i = [\mathbf{I} \mid \mathbf{0}] \qquad (14)$$

$$\mathbf{P}_j = \mathbf{R} [\mathbf{I} \mid -\mathbf{t}] \qquad (15)$$

where $\mathbf{R}$ is the rotation matrix, $\mathbf{t}$ is translation vector and $\mathbf{I}$ is the 3×3 identity matrix.
The inter-image homography matrix $\mathbf{H}_{ij}$ can be decomposed as follows:

$$\mathbf{H}_{ij} = \mathbf{R} \left[\mathbf{I} - \frac{\mathbf{t}'\mathbf{n}^T}{d} \right] \qquad (16)$$

where $\mathbf{t} = -\mathbf{R}\mathbf{t}'$.
For $\frac{\mathbf{n}^T}{d} = 1$, Equation (16) can be expressed as follows:

$$\mathbf{H}_{ij} = \mathbf{R} \left[\mathbf{I} - \mathbf{t}' \right] = \mathbf{R}\mathbf{H}_{ij}^* \sim \mathbf{P}_j \qquad (17)$$

The motion can be estimated from the SVD's $\mathbf{H}_{ij}$ and $\mathbf{H}_{ij}^*$.

$$\begin{aligned} \mathbf{H}_{ij} &= \mathbf{U}\mathbf{S}\mathbf{V} \\ \mathbf{H}_{ij}^* &= \mathbf{U_1}\mathbf{S}\mathbf{V} \end{aligned} \qquad (18)$$

where $\mathbf{U}$ and $\mathbf{V}$ are 3×3 rotation matrices denoted by the columns $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$ and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$, $\mathbf{S} = diag(s_s, s_2, s_3)$ is a diagonal matrix and $\mathbf{U} = \mathbf{R}\mathbf{U}_1$.
The motion between the two cameras $(\mathbf{R}, \mathbf{t}')$ can be estimated as follows:

$$\mathbf{t}' = \frac{\beta_1}{s_1}\mathbf{v}_1 + \frac{\beta_2}{s_3}\mathbf{v}_3 \qquad (19)$$

$$\mathbf{R} = \mathbf{U}\mathbf{U}_1^T \qquad (20)$$

where $\beta_1 = \pm\sqrt{1 - s_3^2}$ and $\beta_2 = \pm\sqrt{s_1^2 - 1}$
In general, the SVD approach gives two distinct solutions. This indetermination can be resolved if additional information about the scene is known. In our case, the normal vector to the ground plane is known (Equation (13)) and is used to eliminate the ambiguity.
In the following section an experiments is performed to estimate the motion parameters between two cameras. The internal camera parameters are assumed to be known.

2.3.1 Experiment: motion estimation
In this section the details of calculating the motion between two camera is provided. The carpet example in Figures 5(a) and (b) is used here to calculate $\mathbf{R}$ and $\mathbf{t}$. The homography is calculated as described in section 2.3 as follows:

$$\mathbf{H}_{ij} = \begin{bmatrix} 0.599 & -0.883 & 213.665 \\ 0.270 & 0.946 & -39.851 \\ 0 & 0 & 1 \end{bmatrix} \qquad (21)$$

The SVD approach is applied to the matrix in Equation (21). The following two solutions were obtained:

Solution 1:

$$\mathbf{n} = \begin{bmatrix} -0.039 \\ -0.697 \\ 0.715 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 0.787 & 0.413 & 0.457 \\ -0.43 & 0.899 & -0.07 \\ -0.44 & -0.141 & 0.886 \end{bmatrix} \quad \mathbf{t} = \begin{bmatrix} -0.798 \\ -0.045 \\ -0.016 \end{bmatrix} \quad (22)$$

Solution 2:

$$\mathbf{n} = \begin{bmatrix} 0.709 \\ 0.683 \\ 0.17 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 0.613 & 0.772 & -0.164 \\ -0.787 & 0.585 & -0.189 \\ -0.05 & 0.245 & 0.968 \end{bmatrix} \quad \mathbf{t} = \begin{bmatrix} -0.289 \\ -0.505 \\ 0.548 \end{bmatrix} \quad (23)$$

The solution is the one with normal vector that satisfies the normal to the plane **n** as defined in Equation (13). In this case **Solution** 1 is selected.

The SVD algorithm is used to calculate the camera motion for the images in Figures 5, 6 and 7. The results are shown in Table 1.

In the next section, the robot localization algorithm is presented. The algorithm is applied on Table 1 and the localization results are presented.

(a)
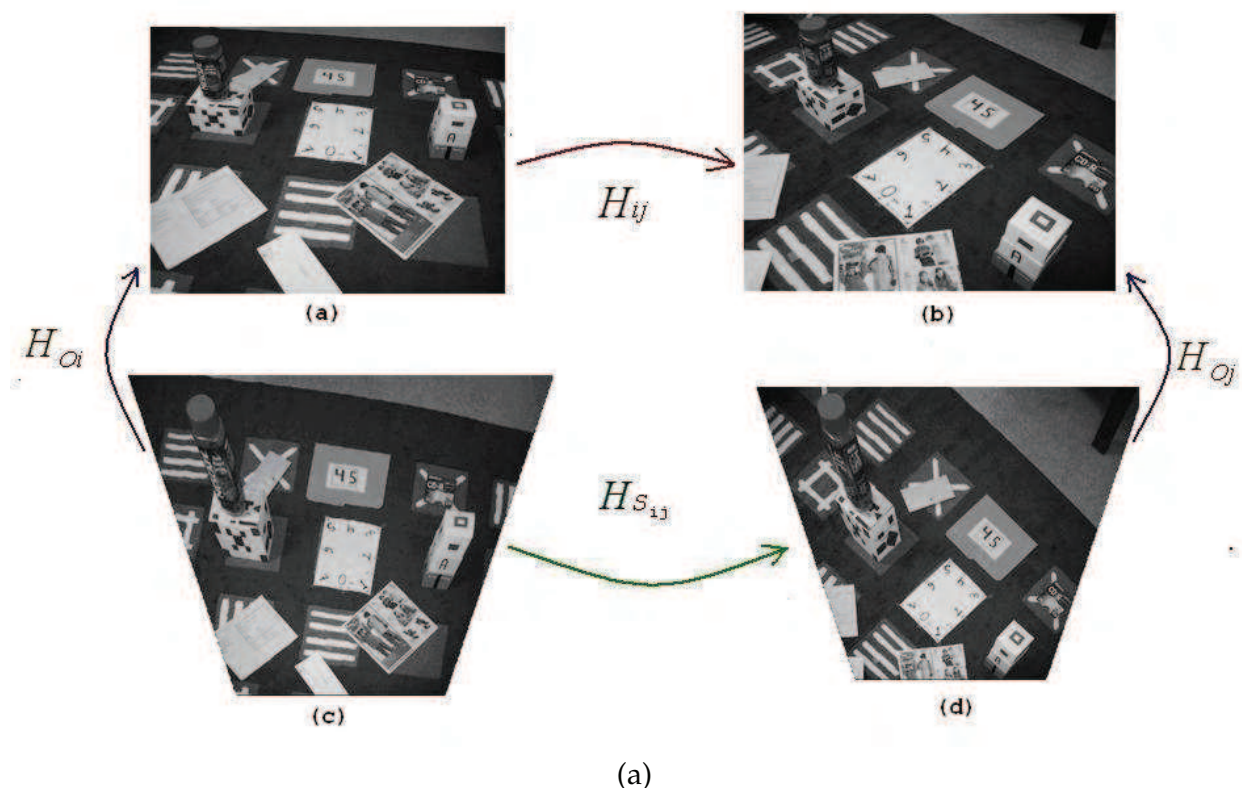
Fig. 5. Two views of a *carpet* scene taken with camera tilt of 45°. (a) The first image (Image i)(b) The second image (Image j). (c) and (d) The overhead view transformations of images in i and j respectively mapped.
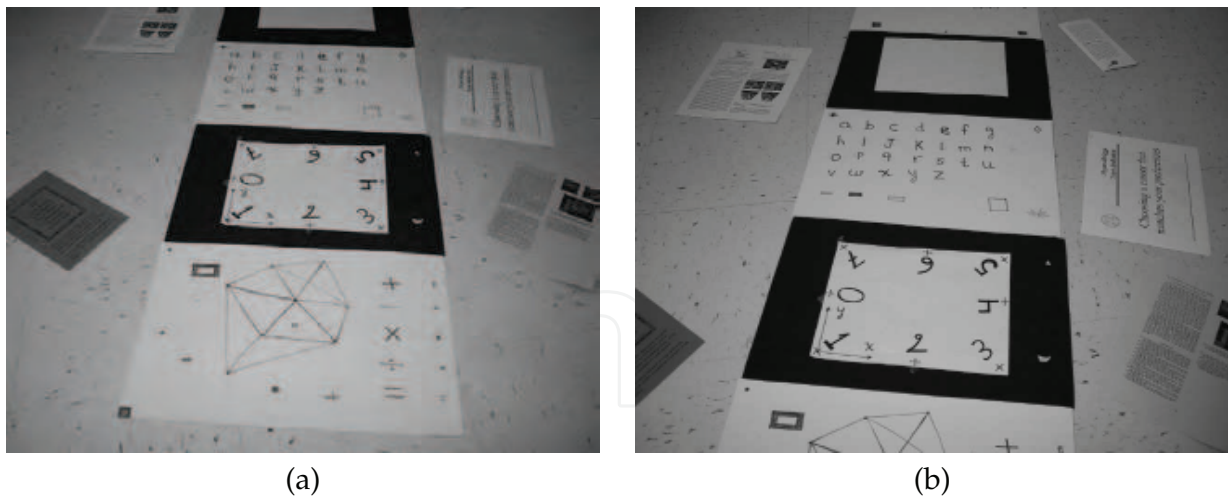
(a)   (b)

Fig. 6. Floor images (a)and (b) images for a floor taken at tilt of 45°.

.

Scene	Motion estimated from $\mathbf{H}_{ij}$
carpet	$\mathbf{R} = \begin{bmatrix} 0.787 & 0.413 & 0.457 \\ -0.43 & 0.899 & -0.07 \\ -0.44 & -0.141 & 0.886 \end{bmatrix}$, $\mathbf{t} = [-0.798 \ -0.045 \ -0.016]^T$, $\mathbf{n} = [-0.039 \ -0.697 \ 0.715]^T$
floor	$\mathbf{R} = \begin{bmatrix} 0.999 & 0.025 & 0.028 \\ -0.026 & 0.999 & 0.016 \\ -0.028 & -0.016 & 0.999 \end{bmatrix}$, $\mathbf{t} = [-0.082 \ -0.256 \ -0.209]^T$, $\mathbf{n} = [-0.013 \ -0.661 \ 0.75]^T$
vase	$\mathbf{R} = \begin{bmatrix} 0.852 & 0.474 & 0.221 \\ -0.424 & 0.873 & -0.237 \\ -0.306 & 0.108 & 0.945 \end{bmatrix}$, $\mathbf{t} = [-0.289 \ 0.239 \ 0.15]^T$, $\mathbf{n} = [-0.01 \ -0.377 \ 0.926]^T$

Table 1. Camera motion estimation using the homography transformations.

2.4 Locating the Robots

The localization problem is formalized as shown in Figure 8. This can be parameterized by the triplet $\Gamma = [\rho, \phi_1, \phi_2]$. Where ρ is the Euclidean distance between the 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

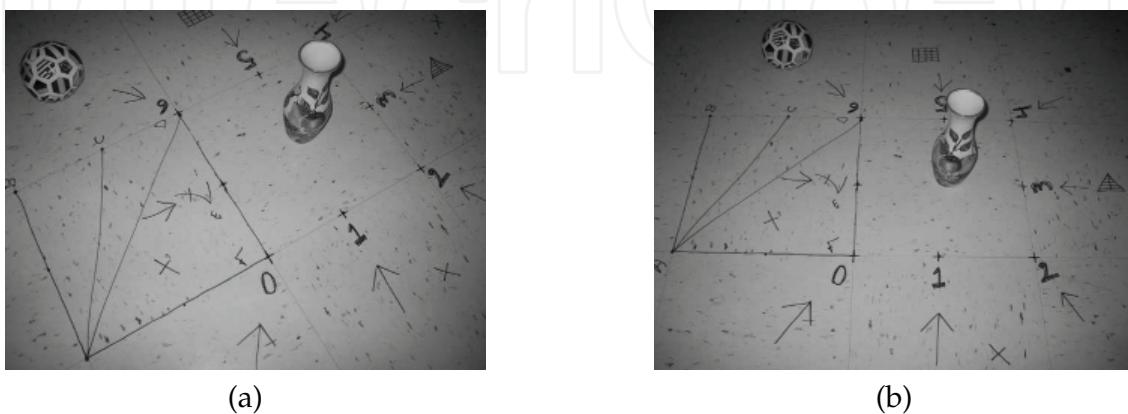(a)   (b)

Fig. 7. Vase images: (a) tilt $= 22^o$ and (b) tilt $= 33^o$.
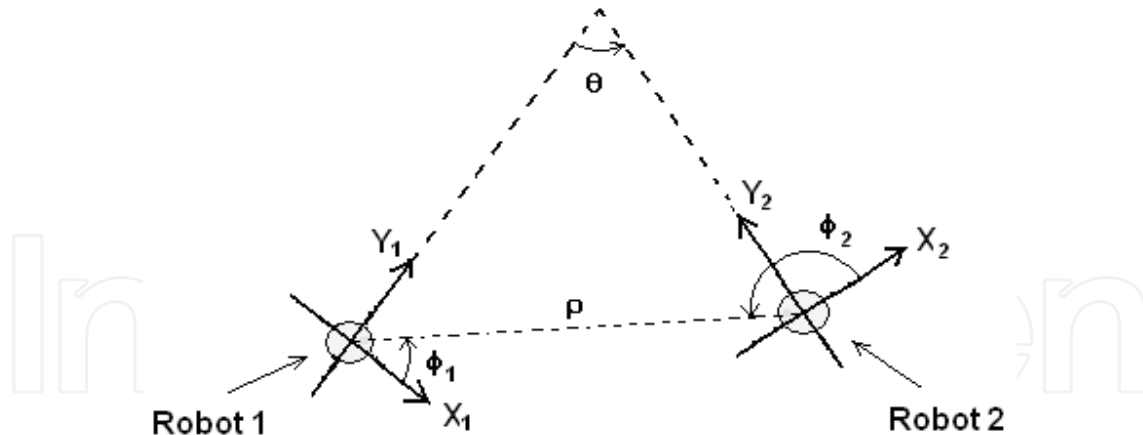
Fig. 8. The Robot angles.

camera coordinate systems on *Robot*1 and *Robot*2 on a 2D coordinate systems. The location of *Robot*2 with respect to *Robot*1 is at (x_1, y_1) defined as follows:

$$x_1 = -t_{x_1} \tag{24}$$

$$y_1 = \sqrt{t_{y_1}^2 + t_{z_1}^2} \sin(\alpha_1 + \beta_1) \tag{25}$$

$$\rho_1 = \sqrt{x_1^2 + y_1^2} \tag{26}$$

where $\beta_1 = \tan^{-1} \frac{t_{y_1}}{t_{z_1}}$

Similarly, the location of *Robot*1 with respect to *Robot*2 is at (x_2, y_2) defined as follows:

$$x_2 = -t_{x_2} \tag{27}$$

$$y_2 = \sqrt{t_{y_2}^2 + t_{z_2}^2} \sin(\alpha_2 + \beta_2) \tag{28}$$

$$\rho_2 = \sqrt{x_2^2 + y_2^2} \tag{29}$$

where $\beta_2 = \tan^{-1} \frac{t_{y_2}}{t_{z_2}}$

Finally, the location vector Γ is defined as follows:

$$\Gamma = [\rho, \phi_1, \phi_2] = \left[\frac{\rho_1 + \rho_2}{2}, tan^{-1}\left(\frac{y_1}{x_1}\right), tan^{-1}\left(\frac{y_2}{x_2}\right) \right] \tag{30}$$

The value of ρ in Equation (30) is calculated in term of camera height units. For a known camera height h in cm, the value of Γ can be expressed as follows:

scene	h	$d_{measured}$	Γ	$|\Delta_d|$
carpet	78 cm	61.5 cm	$[0.79, -3.18^o, 138.76^o]$	0.12 cm
floor	80 cm	27 cm	$[0.33, -75.99^o, 101.76^o]$	0.6 cm
vase	78 cm	32 cm	$[0.40, 44.09^o, 102.06^o]$	0.8 cm

Table 2. The calculated robot position from the estimated homographies

$$\Gamma = [\rho h \text{ cm}, \phi_1, \phi_2] \tag{31}$$

The information in Table 1 is used to localize the robots in the *carpet*, *floor* and *vase* scenes. The localization results are shown in Table 2. The camera height h and the measured distance between the cameras for each scene are shown in columns 2 and 3 respectively. The location vectors Γ are calculated as described in this section and the results are shown in column 4, the discrepancy between the calculated and the measured locations are shown in column 5.

3. SLAM Experiments

In this section two experimental examples are provided for a robot traversing a work site. The objective is that the robot is capable of calculating its pose with respect to its previous position.

3.1 Single-robot SLAM

Figure 9 shows a set of images collected by a robot moving on a planar surface, the tilt with the ground is set manually to 33° and the height of the cameras is measured to be $h = 82$ cm. First, the matching algorithm discussed in the previous section is applied and the inter-image homographies are calculated between consecutive robot locations. The homography transformations calculated are: $\mathbf{H}_{01}$, $\mathbf{H}_{12}$, $\mathbf{H}_{23}$, $\mathbf{H}_{34}$, $\mathbf{H}_{45}$, $\mathbf{H}_{56}$ and $\mathbf{H}_{67}$. These homographies are used to incrementally build a location graph. At each new location, the robot pose is calculated with respect to its previous position, the resulting location graph is shown in Figure 10. To estimate the accumulated error in estimating the pose between consecutive locations, the pose between the initial location (Robot0) and last location (Robot 7) is estimated by concatenating the calculated inter-image homography transformations as follows:

$$\mathbf{H}_{07} = \mathbf{H}_{67}\mathbf{H}_{56}\mathbf{H}_{45}\mathbf{H}_{34}\mathbf{H}_{23}\mathbf{H}_{12}\mathbf{H}_{01} \tag{32}$$

Any error in estimating any of the matrices in the right hand side of Equation(32) will influence the estimation of $\mathbf{H}_{07}$. In Figure 10 the value of r_1 represents the measured distance between Robot 7 and Robot 0, the estimated distance using $\mathbf{H}_{07}$ is represented by r_2.
The error in estimating the distance between Robot 0 and Robot7 can be evaluated by $d_r = |r_1 - r_2|$ as shown in Figure 10. The value of d_r is $11.5 cm$ 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 as described in Section 2.2. The result is shown in Figure 11.
In the next section, our single-robot SLAM algorithm is generalized to the multiple-robot case.

3.2 Multi-robot SLAM

In this section we provide an example of two mobile robots *RobotA* and *RobotB* moving in the same work site. The camera tilt on robots *RobotA* and *RobotB* are set manually to 33° and 45° respectively, the height of the camera from the ground plane is $55 cm$. Figure 12 shows the images captured by *RobotA* and Figure 13 shows the 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 a local location graph and a local map of the environment as discussed in the single-robot SLAM case, the local locations graphs are shown in Figure 14(a) and (b). When an overlap occurs between the two robots, a joint location graph
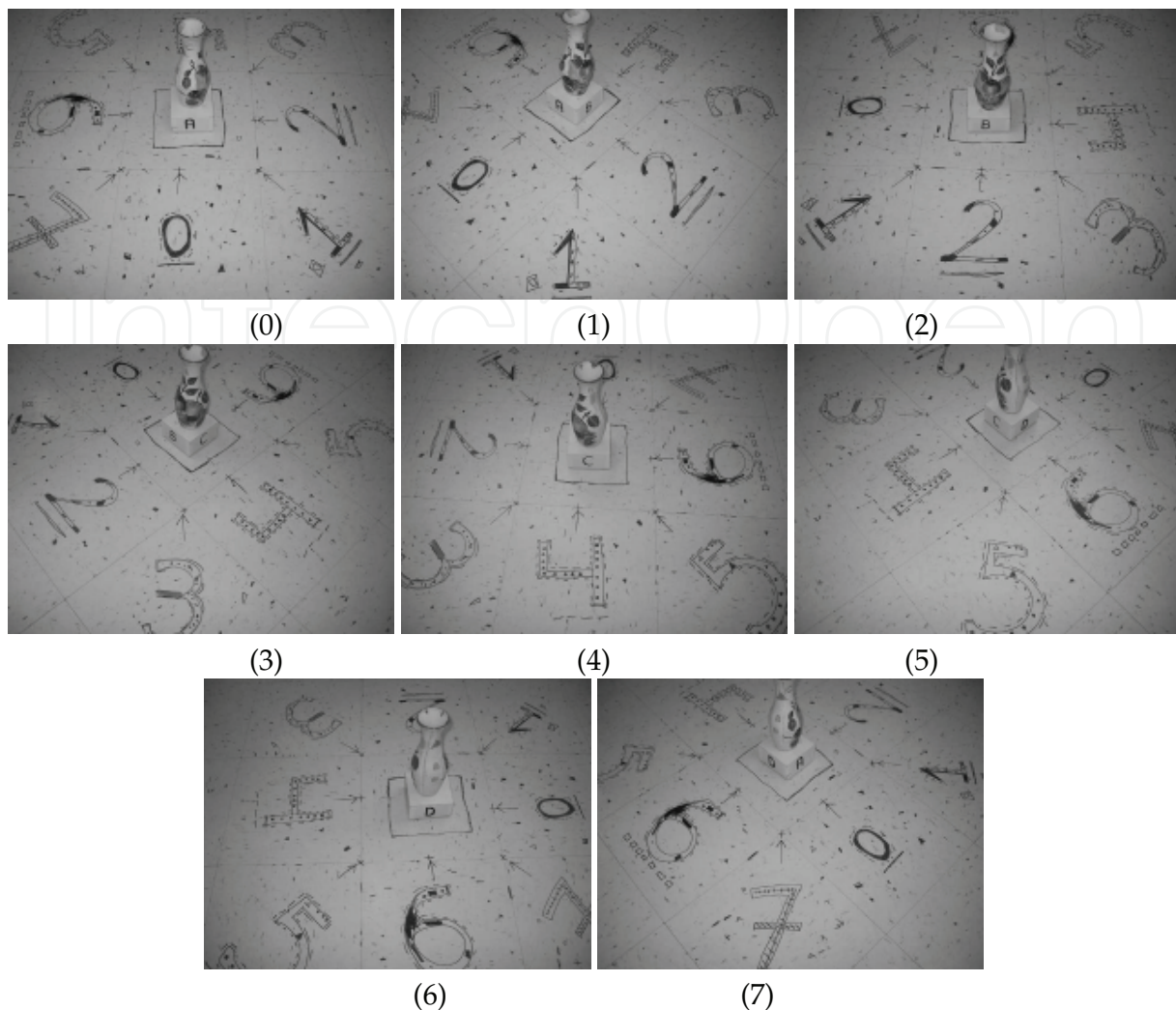
Fig. 9. Single-robot SLAM: Image set collected by the robot.

and environment map can be build by calculating the relative pose of the two robots. The approach used to determine the overlap between two robots is presented in the next section. At location $A7$ for $Robot A$ and location $B8$ for $Robot B$ the two robots are viewing the same scene, this overlap can be verified by comparing image $A7$ in Figure 12 and image $B8$ in Figure 13. 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 14(c) shows the joint location graph which relates all the robot locations with $A7$ selected as the reference frame. The joint map of the environment is shown in Figure 15.

3.3 Overlapping view detection

In order to determine if two cameras are viewing the same scene, the images have to be compared. One approach is to calculate the inter-image homography as discussed in Section 2.3 and then determine the validity of this transformation. The validity may be checked by mapping the corner points from one image to the other. If there exist a large number of matches then this transformation is considered to be a valid transformation and accepted otherwise it is rejected.
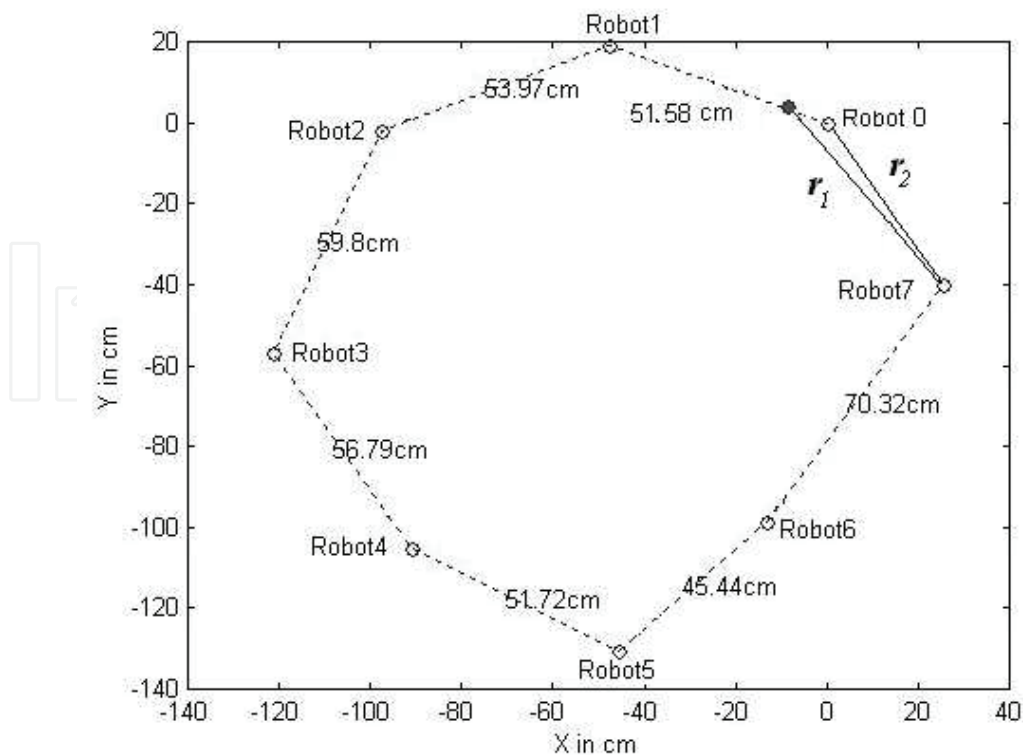
Fig. 10. Single-robot SLAM: Location graph.

For two robots, this approach requires calculating the inter-image homography for every possible combination of images. For N images captured by each robot, a complexity of $O(N^2)$ of inter-image homography calculations is required. To improve the efficiency, the homography between any two images is only calculated if there is a potential for match. To do so, we propose a method based on color histogram distance.

The color histograms are commonly used to compare images based on their overall appearance (Pass et al. (1996)). Because they are computationally efficient, they have been used in many applications like image retrieval (Brown et al. (1995); Flicker (1995); Olga & Stonebraker (1995); Pentland et al. (1996)) and object identification(Swain & Ballard (1991)).

The histogram **h** of an image is a vector consisting of n bins as follows:

$$\mathbf{h} = [h_1, \ldots, h_n] \tag{33}$$

Each bin $h_i \in \mathbf{h}$, $1 \leq i \leq n$, contains the number of pixels of color i in the image. Two images **I** and **I**$'$ of histograms **h** and **g** can be compared based on the inter-bin distance $d(\mathbf{h}, \mathbf{g})$ between their histograms. Images **I** and **I**$'$ are considered similar if $d(\mathbf{h}, \mathbf{g})$ is minimized. The histogram is trivially computed compared to the inter-image homography. We propose an algorithm to use histogram distance to determine the most likely overlap and then use-inter image homography transformation to validate such an overlap. The algorithm is stated as follows:

Let $\mathbf{A} = \{\mathbf{I}_1, \ldots, \mathbf{I}_{N_A}\}$ and $\mathbf{B} = \{\mathbf{I}'_1, \ldots, \mathbf{I}'_{N_B}\}$ be the sets of images collected by *RobotA* and *RobotB* respectively. With set **A** contains N_A images and set **B** contains N_B images. The problem to solve is to find, using color histogram distance, an image $\mathbf{I}_a \in \mathbf{A}$ and an image $\mathbf{I}'_b \in \mathbf{B}$ such that $\mathbf{I}_a$ and $\mathbf{I}'_b$ are most likely the overlap images if such an overlap exists between the two robots.

Fig. 11. Single-robot SLAM: generated map of the site

Let $Hist(\mathbf{I})^1$ denotes the histogram of the overhead view of image $\mathbf{I}$ and C_{Hist} and C_{Homog} denote the cost of computing the histogram distance d and the inter-image homography between two images. The algorithm we propose is summarized as follows:

```
Step 1:  For k = 1 to N_A
             For l = 1 to N_B
                 calculate d(Hist(I_k), Hist(I'_l))
                 If d is minimized then
                         I_a = I_k and I'_b = I'_l
                 End
             End
         End
Step 2: Calculate the inter-image homography H_ab between images I_a and I'_b
Step 3: Accept/reject H_ab based on the validation test.
```

In Step 1, the histogram distances between all the images acquired by the two robots are calculated. The outcome is the two images ($\mathbf{I}_a$ and $\mathbf{I}'_b$) whose histogram distance is the global minimum. In Step 2, the inter-image homography ($\mathbf{H}_{ab}$) is calculated for the two images.

[1]throughout this chapter all the histograms are calculated using the overhead view images

Fig. 12. Image set collected by the *Robot A*.

(B₀)   (B₁)   (B₂)

(B₃)   (B₄)   (B₅)

(B₆)   (B₇)   (B₈)

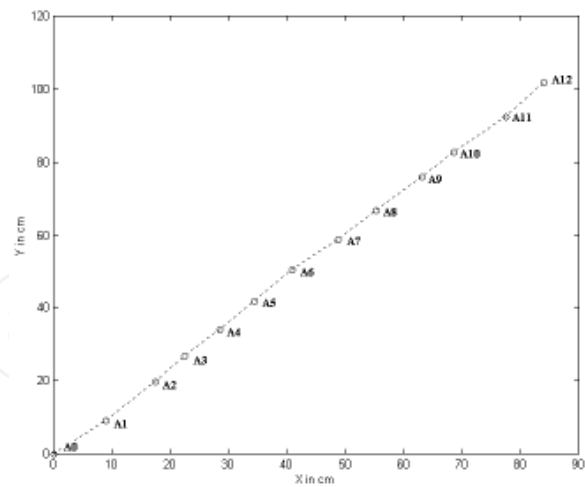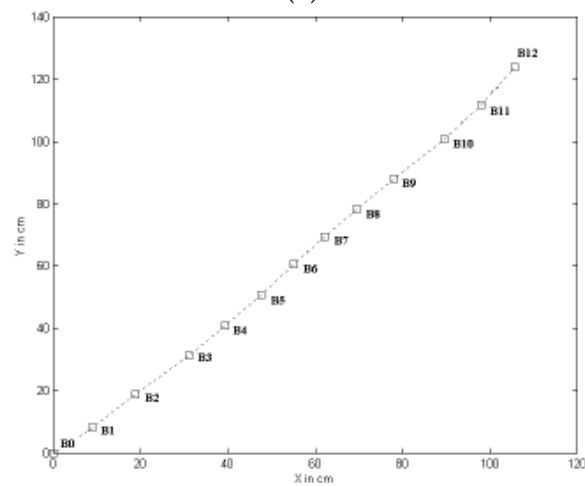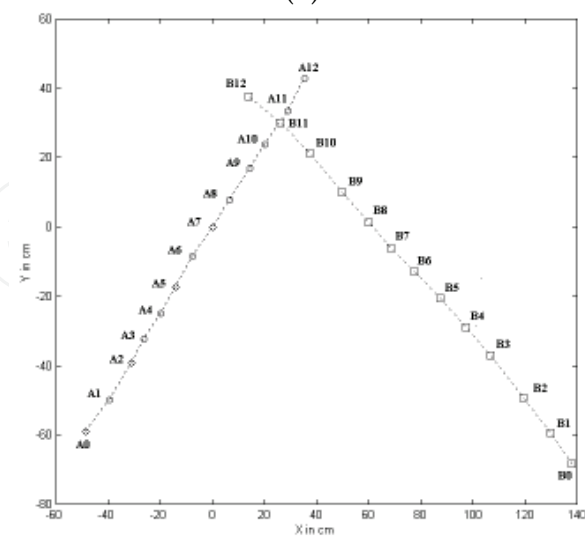(B₉)   (B₁₀)   (B₁₁)

(B₁₂)
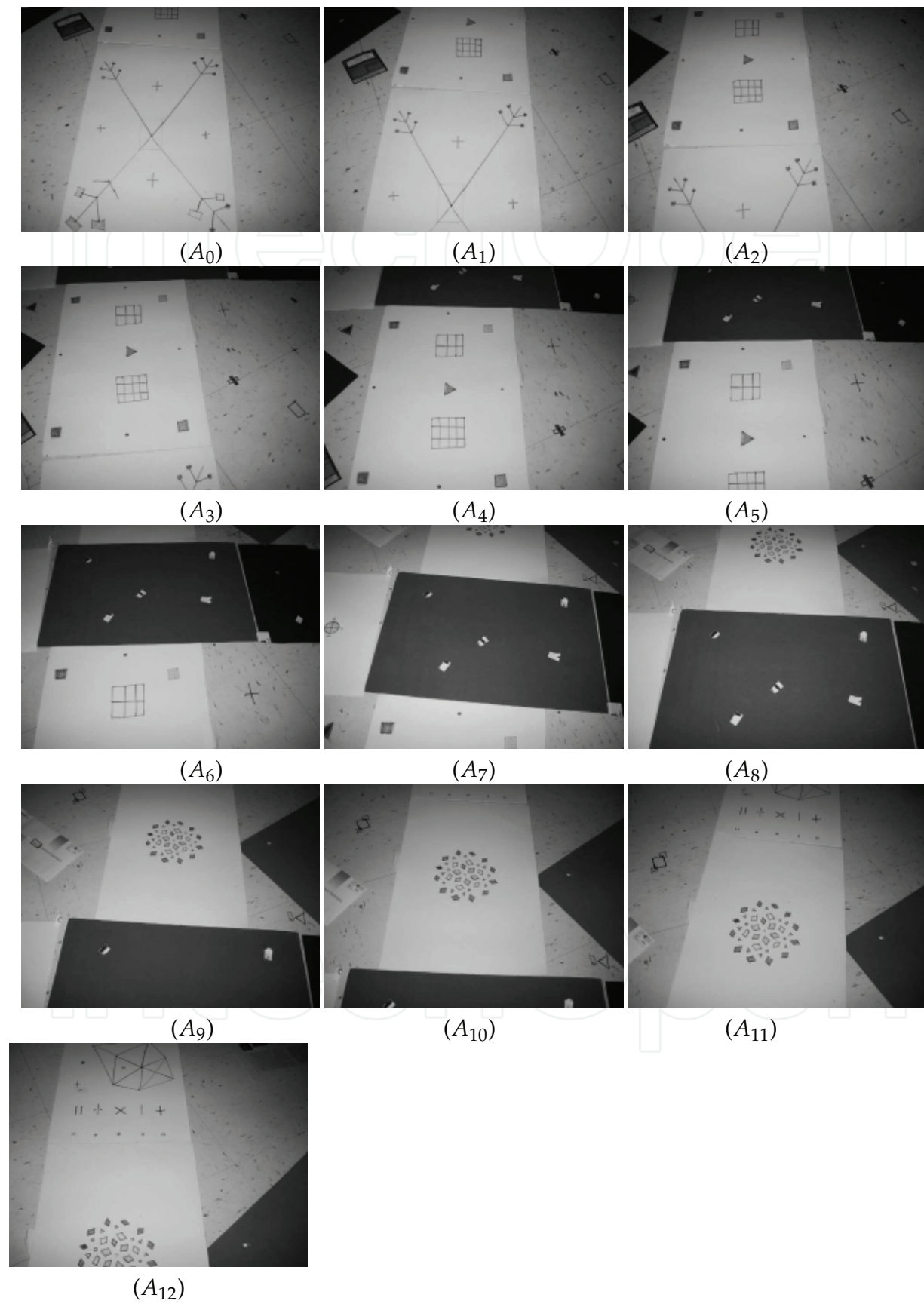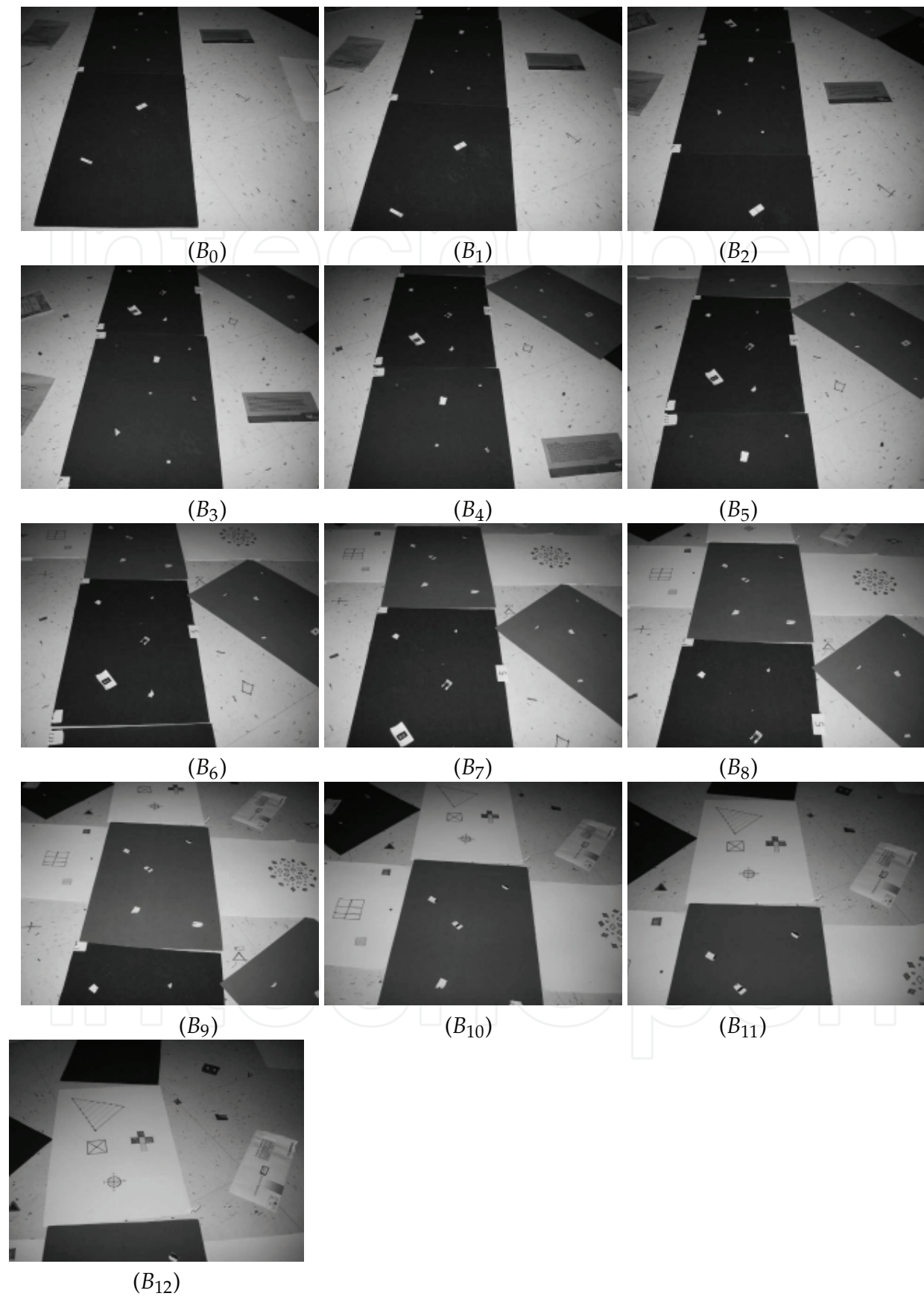
Fig. 13. Image set collected by the *Robot B*.

(a)

(b)

(c)

Fig. 14. Multi-Robot Localization: (a) location graph generated by *Robot A* (b) location graph generated by *Robot B* (c) the joint location graph
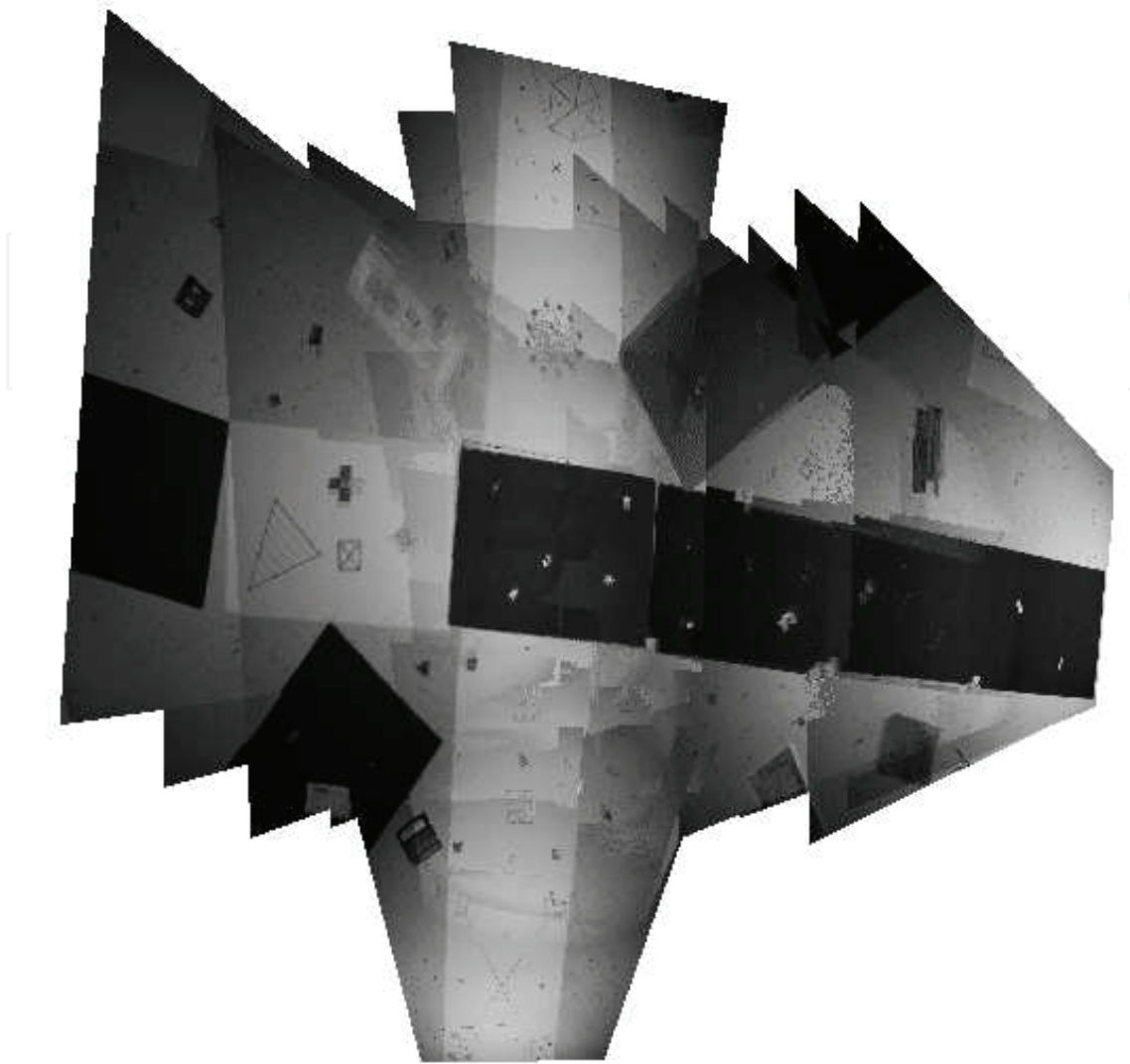
Fig. 15. The map generated from images collected by *Robot A* and *Robot B*.

Finally, in Step 3, $\mathbf{H}_{ab}$ is validated, if there exist a high number of matches between corners in image $\mathbf{I}_a$ and corners in image $\mathbf{I}'_b$ then the homography $\mathbf{H}_{ab}$ is accepted.

4. Summary

We have presented a vision-based technique to solve the multi-robot SLAM problem. The different robot locations are computed by finding the transformations that relate together the captured images of the scene. The camera motions are estimated from the computed inter-image homographies and each robot is localized with respect to a local map. To build a global map, the inter-image homographies have to be calculated between different robots in the team. To do so, an overlapping view detection technique is presented. The technique is based on color histogram distances between images collected by the team of robots. When two images from two robots have similar color histograms, a potential overlapping between the two robots is assumed. The inter-image homography is calculated to verify if this overlapping is valid. If the such an overlapping exist, then it is possible to build a common map of the environment inside which the robots are evolving.

5. References

Altmann, S. (1986). *Rotations, Quaternions, and Double Groups*, Oxford University Press, Oxford.

Brown, M., Foote, J., Jones, G., Jones, K. & Young, S. (1995). Automatic content-based retrieval of broadcast news, *ACM Multimedia conference*.

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

Flicker, M. (1995). Query by image and video content: The qbic system, *IEEE computer*, Vol. 28 of *9*, pp. 23–32.

Hajjdiab, H. & Laganière, R. (2004). Vision-based multi-robot simultaneous localization and mapping, *in Proceedings of the 1st Canadian Conference on Computer and Robot Vision, CCCRV'04*, London,Ontario,Canada.

Laganière, R. (2000). Composing a bird's eye view mosaic, *Vision Interface* pp. 382–386.

Liu, Y. & Thrun, S. (2003). Gaussian multi-robot slam, *submitted to NIPS* .

Lowe, G. (1991). Fitting parameterized three-dimensional models to images, *IEEE Trans. Pattern Analysis and Mach. Intell, PAMI*, Vol. 13, pp. 441–450.

Lowe, G. (1999). Object recognition from local scale invariant features, *Proceedings of the 7th International conference on Computer Vision*, Kerkyra, Greece, pp. 1150–1157.

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

Murphy, K. (99). Bayesian map learning in dynamic environments, *NIPS* .

Murray, D. & Little, J. (1998). Using real-time stereo vision for mobile robot navigation, *Proceedings of the IEEE Workshop on Perception for Mobile Agents*, Santa Barbara, CA.

Olga, V. & Stonebraker, M. (1995). Chabot: Retrieval from a relational database of images, *IEEE computer*, Vol. 28 of *9*, pp. 40–48.

Pass, G., Zabih, R. & Miller, J. (1996). Comparing images using color coherence vectors, *In Proceedings of ACM Multimedia 96*, Boston, Ma, USA, pp. 65–73.

Pentland, A., Picard, R. & Sclaroff, S. (1996). Photobook: Content-based manipulation of image database, *International Journal of Computer Vision* 18(3): 233–254.

Se, S., Lowe, D. & Little, J. (2001). Vision-based mobile robot localization and mapping using scale-invariant features, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, pp. 2051–2058.

Simmons, R., Apfelbaum, D., Burgard, W., Fox, M., Moors, D., Thrun, S. & Younes, H. (2000). Coordination for multi-robot exploration and mapping, *In Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, TX.

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

Swain, M. & Ballard, D. (1991). Color indexing, *International Journal of Computer Vision* 7(1): 11–32.

Thrun, S., Burgard, W. & Fox, D. (2000). A real-time algorithm for mobile robot mapping with application to multi-robot and 3d mapping, *in Proceedings of the IEEE International Conference on Robotics and Automation*.

Triggs, B. (1998). Autocalibration from planar scenes, *In Proc. European Conference on Computer Vision*, pp. 89–105.
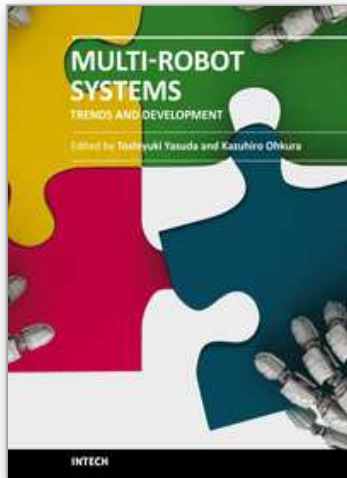
Multi-Robot Systems, Trends and Development

Edited by Dr Toshiyuki Yasuda

This book is a collection of 29 excellent works and comprised of three sections: task oriented approach, bio inspired approach, and modeling/design. In the first section, applications on formation, localization/mapping, and planning are introduced. The second section is on behavior-based approach by means of artificial intelligence techniques. The last section includes research articles on development of architectures and control systems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hassan Hajjdiab and Robert Laganiere (2011). Multi-Robot SLAM: A Vision-Based Approach, Multi-Robot Systems, Trends and Development, Dr Toshiyuki Yasuda (Ed.), ISBN: 978-953-307-425-2, InTech, Available from: http://www.intechopen.com/books/multi-robot-systems-trends-and-development/multi-robot-slam-a-vision-based-approach

INTECH
open science | open minds