External Cameras & A Mobile Robot: A Collaborative Surveillance System

Punarjay Chakravarty and Ray Jarvis
Monash University, Australia
punarjay@gmail.com,ray.jarvis@eng.monash.edu.au

Abstract

This paper presents a system that surveils an indoor environment through the collaborative use of overhead cameras and a mobile robot. The robot is localized both in the image plane of the external camera and a ground plane map of its environment. This simultaneous localization is used to build up a homography between the image plane of each external camera and the ground plane. The robot begins the calibration procedure by moving through a series of pre-determined waypoints in the environment. As it comes into view of each camera, it uses visual servoing to orbit the centre of the camera image and keeping within its field of view whilst avoiding obstacles. The system builds up a transformation matrix between each camera and the ground plane and proceeds to patrol the environment. When an intruder is detected in any of the external cameras, the system uses the automatically determined homography matrices to calculate the ground plane position of the person and sends out the robot to intercept the intruder. Experiments are conducted with 2 external cameras in this paper but could easily be extended to N cameras.

1 Introduction

For a mobile robot to be able to intercept intruders seen from an external camera, it needs to be able to establish the relationship between the camera's coordinate system and the coordinate system that the robot is operating in. The mapping of the robot's environment could be done from scratch, where the robot has no map to begin with and no idea of its location on the map. This problem, one of simultaneous robot localization and map-building (SLAM), where a robot needs to localize itself on an incrementally built map of a hitherto unknown environment, has been the subject of a significant amount of research during past decades [Leonard and Durrant-Whyte, 1991; Smith and Cheeseman, 1987; Thrun *et al.*, 2005]. This work does not aim to make any advances in the SLAM field and in the experiments conducted, the robot localizes itself on an apriori provided metric map using a particle filter approach [Fox *et al.*, 2001]. Most indoor environments have floorplans that could easily be digitized for use by the robot.

Camera calibration involves calibrating both its internal and external parameters to determine the projection of a 3-dimensional point in the world reference frame in the 2-dimensional camera image plane. In work done by [Chen *et al.*, 2007], the external camera parameters of an overhead camera are determined using the motion of a marker on a mobile robot. The method assumes knowledge of internal camera parameters and requires the prior setup of a database of robot motions and their associated accuracies for each camera. The system enables the pose of the robot to be calculated with respect to each external camera, which could be extended to multiple cameras by repeating the calibration procedure for each camera. However, since the robot has no idea of the placement of cameras, it still would require an operator to drive the robot to each camera in turn.

[Rekleitis *et al.*, 2006] have addressed the problems of estimating camera and robot poses in a surveillance network by combining them into a SLAM framework. Their robot carries a calibration pattern and performs a series of movements in front of each camera to ascertain the camera's internal and external parameters. These parameters provide the robot with the relative locations of landmarks (cameras) as it moves in its environment. Odometry is used to estimate the distance and direction between cameras, each sighting at a camera being a measurement (the robot does not use any other onboard sensors) and an Extended Kalman Filter is used to filter out imperfections in sensor and odometry data. The problem with this is that there is a discontinuity in the estimated robot path in the world coordinate frame when gaps between cameras are large and the odometry error accumulates. This error can be avoided if the

robot has an independent means of localizing itself in the global coordinate frame, which is what is done in our system. The calibration procedure in [Rekleitis *et al.*, 2006] determines how a point in world coordinates $[X, Y, Z]$ relates to a pixel $[u, v]$ in camera coordinates. In our case, the main motivation for calibrating the external cameras to the global coordinate frame is to enable the robot to find a route to a person (possibly not in the visual scope of the robot) detected by an external camera. The ground can be assumed to be flat (most office buildings have flat floors) and hence the ground plane position to be reached will always have the 3rd coordinate $Z = 0$. So the problem is simplified to determining a relationship between the pixel coordinate $[u, v]$ and the ground plane coordinate $[X, Y]$, i.e., a relationship between two planes. This can be done using a homography technique [Criminisi *et al.*, 1999], which requires just four corresponding points to be known both on the image and ground planes. Our method automates the process of collecting these points: it is achieved by the simultaneous localization of the robot in the image and ground planes as it moves around in its environment. In contrast with [Rekleitis *et al.*, 2006], our method uses sensors on the robot (not external cameras) for its localization on the metric map. It also obviates the need for carrying a calibration pattern that is visible from the camera; camera resolution and the size of the calibration target in their experiment restricted camera-to-robot distance to a maximum of two metres during calibration, which rather limited the height at which the cameras could be placed.

If the robot were always in the field of view of the surveillance camera, the objective of navigating it to a target could be addressed by visual servoing alone as in [Taylor, 2004], where the robot is driven to a taget in the image using visual feedback from the camera. However, this would become complicated if there were obstacles between the robot and the target. Also, having an operational environment that extended far beyond the field of view of one camera would complicate things. The robot would have to know how to get into the field of view of the camera in question, and then commence visual servoing to the target.

The other possibility, with one or two cameras is that an operator mark fiducial points on the ground whose world coordinates are known, and then click on the points in the images to get the correspondences. But this would quickly become tedious with more than one camera. Also, every time the cameras were disturbed or moved accidentally, the procedure would have to be repeated. In contrast, an autonomous system could easily detect camera movement (each camera records a background model as discussed in Section 2 that could be used for this purpose) and have the robot repeat the homography based calibration procedure.

In our first publication on this subject ([Rawlinson *et al.*, 2004]), instead of using homography to establish a relationship between two planes, the system generated a lookup-table (LUT) of image points to ground coordinates as the robot wandered in the FOV of the camera. Afterwards, the operator clicked on a point in the image plane where the robot was intended to go to. The robot found the nearest point in the LUT generated earlier and headed towards the corresponding point on the ground plane. This approach, though applicable in situations where the ground is not flat, required the robot to explore large areas of the image. This would be time consuming in an environment with many cameras. Non-flat ground situations can only be envisioned in large, outdoor environments, and even in those situations, the ground overlooked by a camera could be approximated as piece-wise planar.

The rest of the paper is divided into the following sections:

Section 2 Multiple Target Tracking and Robot Localization in External Camera(s): This section talks about the detection of the robot in the image plane and the tracking of the robot and other entities in a multiple target tracking framework.

Section 3 Particle Filter Localization of the Robot in the Ground Plane This section briefly discusses robot localization in an apriori provided map of the ground plane using its laser range finder.

Section 4 Image Plane to Ground Plane Transformation & Robust Homography Estimation: The homography technique to model the transformation between the image plane of the external camera and the ground plane, given a set of corresponding points between the two planes, is detailed in this section. It further discusses a robust homography algorithm that uses Random Sampling Consensus (RANSAC) to calculate the homography in the presence of noise.

Section 5 Visual Servoing: This section describes a visual servoing method to keep the robot within the FOV of a discovered camera and to maximize the spread of points along the trajectory for homography calculation.

Section 6 Navigation: This section describes robot navigation techniques.

Section 7 Experiments: This section details experiments conducted to prove the functioning of the system.

2 Multiple Target Tracking and Robot Localization in External Camera(s)

The overhead cameras (in experiments for this paper a maximum of two cameras are used) are connected to a computer dedicated to target tracking. This computer tracks the targets, both robot and human and sends their

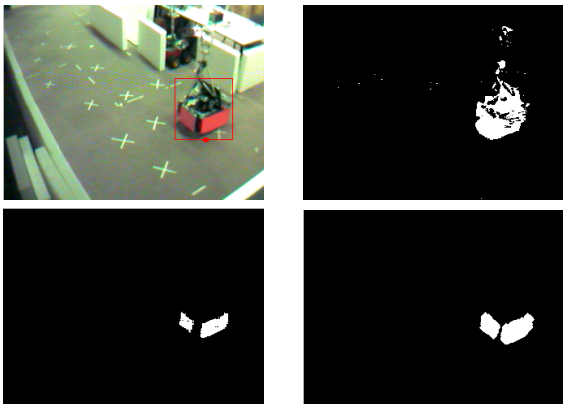


Table 1: Detection of mobile robot in external camera using its distinctive red coloured base. From top left, in raster order: input image with detected robot circumscribed with red box; red dot indicates the position of the bottom of the robot used for homography calculation, foreground image after background subtraction, red-filtered image, foreground image binary AND-ed with red-filtered image and the result dilated.

image plane coordinates to the laptop on-board the robot over a wireless LAN.

2.1 Background Subtraction

Background subtraction is used to detect entities in the image. The background is modelled at each pixel by a uni-modal Gaussian distribution that is tracked by a Kalman Filter. The RGB image is converted to greyscale and the background is modelled in a single-channel colour space to reduce computational complexity. Foreground pixels are collated into blobs using connected components analysis. The blobs are then passed to a multi-target tracker (see following subsection).

The background is only updated in regions where there is no motion (moving areas in the image are obtained by differencing successive frames) and where there are no currently operating tracks. So the information from the tracker is fed back into the background modelling so that a person who walks into the scene and then remains stationary for some time is not blended into the background. This is shown in the block diagram for the background subtraction and tracking (Figure 1).

2.2 Detection of Robot

The robot is distinguised from tracked humans using its red base. In order to make the robot distinctive, a red cardboard skirting has been bolted around its sides. The input image is filtered for a high response in the red channel and a low response in the green and blue channels. This binary red-filtered image is AND-ed with the bi-

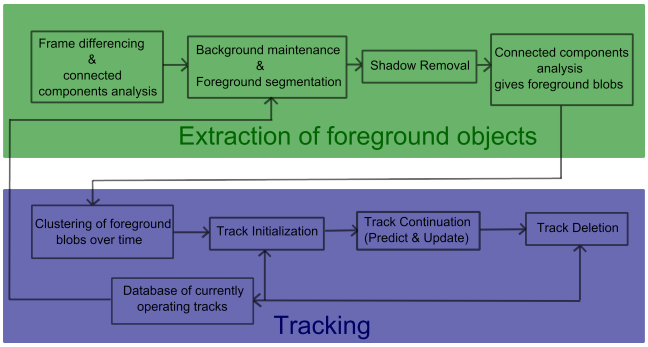


Figure 1: Block diagram showing background subtraction and tracking of foreground objects. Note that the location of the tracks feed back into the background estimation so that the background is not updated within the bounding boxes of currently operating tracks.

nary image got from background subtraction. If a high percentage of the pixels within the bounding box delineating a currently operating track are red in colour, the track is tagged as a robot. The series of figures in Table 1 show the image-processing steps for recognizing the robot in the external camera.

2.3 Multiple Target Tracking

Foreground pixels obtained as a result of background subtraction are collated into blobs using connected components analysis, which are associated with other blobs in neighbouring frames. Blobs that have been associated for a number of consecutive frames are identified as tracks, whose trajectories are smoothed with the Kalman filter.

A bank of N Kalman filters are used to track N targets. A new filter is initialized on each new connected component in the foreground image provided that its bounding box does not coincide with an already existing track. The filter state $\widehat{x_t}$ at time t includes the bounding box parameters $(x, y, width, height)$ along with their first and second derivatives. Each filter, in addition maintains the following data:

1. A Colour histogram of the hue channel in HSV colour space of the foreground component inside the bounding box
2. Track unassociation count: The number of times in the past frames that the filter has not been associated with a measurement
3. Flag indicating whether the target is robot/human

The robot is distinguished by detecting its red base as mentioned in Section 2.

2.4 Data Association

The data association problem, one of associating the set of latest measurements with filters, is achieved in a mod-

ified nearest neighbour fashion [Thrun *et al.*, 2002]. A measurement is assigned to the closest (Euclidean distance) operating filter such that the correlation distance d_{corr} (equation 1) in colour histograms between the measurement and the filter is below a threshold. Measurements already associated with a filter are not associated again with any other filter, in what is known as a "hard" assignment.

$$d_{corr}(H_1, H_2) = \frac{\sum_i H_1'(i) H_2'(i)}{\sqrt{\sum_i (H_1'(i))^2 (H_2'(i))^2}} \quad (1)$$

where $H_k'(i) = H_k(i) - (1/N)(\sum_j H_k(j))$ and N is the number of bins in the histogram.

Once a measurement is assigned to a filter, its Track unassociation count is set to zero.

2.5 Track Initialization

The track initialization procedure initializes a new Kalman filter around un-associated measurements that have been clustered over a few frames.

2.6 Track Deletion

If a track is consistently not associated with any measurements, i.e., its track unassociation count rises above a threshold, it is deactivated.

2.7 Track Continuation

The track continuation procedure implements the following

1. Predict : The filter state vector and error covariance matrix are propagated forward in time using the prediction equations of the Kalman filter.
2. Update : The associated measurement is used to update the state and error covariance using the Kalman update equations. In addition, the colour histogram of the filter is updated using the histogram update equation 2.

$$H_{updated}(i) = \alpha H_{measurement}(i) + (1 - \alpha) H_{track}(i) \quad (2)$$

where $H_{updated}$ is the updated value in bin i of the track's histogram H_{track} using the histogram $H_{measurement}$ from the latest associated measurement.

The tracking mechanism is outlined in the block diagram in Figure 1.

3 Particle Filter Localization of the Robot in the Ground Plane

A map of the ground plane is provided to the robot and a particle filter is used to localize the robot from scratch (global localization) on this map. The particle filter [Rekleitis, 2002] uses a set of probability weighted particles to approximate the probability density function (pdf) that describes the location of the robot on the map. For each particle, the expected laser range finder readings for that particle's location on the map (obtained by ray-tracing) are compared against the laser readings coming from the laser range sensor at that time. Particles are weighted according to the agreement between the observed and expected laser range readings. Particles with higher weights are propagated to the next cycle of the algorithm and lower weighted particles die out, leading to a clustering of particles around the actual position of the robot. This is a well-established method and this paper does not discuss this algorithm further. For more details on particle filter based localization, the reader is asked to refer to our earlier paper on this subject [Rawlinson *et al.*, 2004] or a tutorial on robot localization using particle filtering [Rekleitis, 2002].

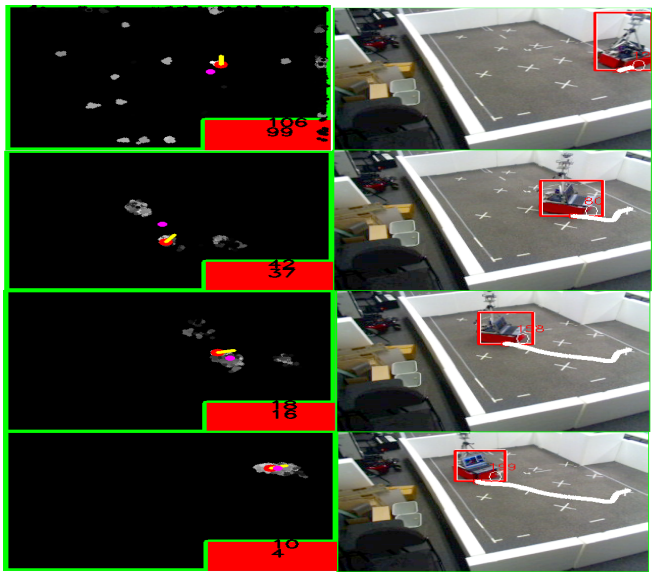
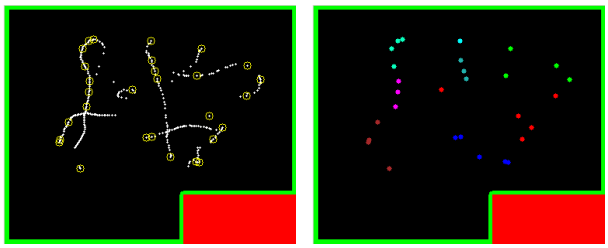


Figure 2: Montage showing localization of robot in the ground plane and simultaneous tracking in the image plane of external camera. Particles scattered throughout map to begin with. Over time, particles cluster around correct robot position.

4 Image Plane to Ground Plane Transformation & Robust Homography Estimation

Robot movement in view of an external camera gives it a set of corresponding points in the camera image plane and the ground plane because it is simultaneously localized in both planes. These points are utilized to build a homography or mapping between the image and ground planes. Once the homography is deduced, any point on



(a) (b)

Figure 3: Extraction of perimeter points (circled) from robot trajectory (in a map of the ground plane) shown in white. The 2nd figure shows clustering of these perimeter using k-means clustering, with a different colour per cluster. One point from each of these clusters is randomly selected to estimate the homography. This process ensures well-spaced points are used for homography estimation.

the image plane can be mapped on to the ground plane. Consequently, the robot can be sent to intercept any person viewed in the external camera because the homography gives it a corresponding destination position on the ground plane.

It has been empirically established that the homography calculation works best if the points are well spread out. From the robot trajectory (shown in white in Figure 3(a)), a perimeter point extraction algorithm is first used to extract all the points on the periphery of the trajectory (peripheral points are shown using yellow circles in Figure 3(a)). The perimeter point algorithm involves dividing the set of points into equi-angular segments centred around its centroid and choosing the farthest point in each segment to get the perimeter points. From this point set on the perimeter, 8 points are chosen to be sufficiently apart by performing k-means clustering to cluster the points into 8 clusters and selecting one point from each cluster randomly. The 8 clusters are shown in Figure 3(b). Note that the robot trajectory (white) seems discontinuous, because only those parts of the trajectory are used where the robot is confident of its localization.

Let the points in the ground plane be represented using the homogeneous vector $X = (X, Y, Z)$, and their counterparts in the image plane by $x = (x, y, 1)$.

The homogeneous equation system for n points is as follows:

$$A\widetilde{h} = \widetilde{0} \tag{3}$$

where

$A = [A_1|A_2]_{2nx9}$

and

$A_1 =$

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_i & y_i & 1 \\ & & \vdots & & & \end{bmatrix},$$

$A_2 =$

$$\begin{bmatrix} -x_iX_i & -y_iX_i & -X_i \\ -x_iY_i & -y_iY_i & -Y_i \\ & & \vdots \end{bmatrix}$$

and

$\widetilde{h} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}'$

For each pair of points on the image plane and the ground plane that the robot traverses, we fill up two rows of the matrix A, which becomes a $2n$x9 matrix, where n is the number of points. The elements of the transformation vector $\widetilde{h}$ are obtained from the eigen vector corresponding to the least eigen value of A^TA. The Singular Value Decomposition (SVD) of A:

$$A = UDV^T \tag{4}$$

results in the eigenvalues being arranged in decreasing order along the diagonal of the matrix D and the corresponding eigenvectors along the columns of the matrix V. The values of the eigenvector associated with the smallest eigenvalue make up the values of the vector $\widetilde{h}$. $\widetilde{h}$ is the null-vector of A, i.e., the column of V associated with a zero-valued eigenvalue in D.

$$X = Hx \tag{5}$$

The elements of $\widetilde{h}$ are re-arranged to form the elements of the matrix H in equation 5. The ground plane location corresponding to any arbitrary point in the image plane can now be found using this equation.

4.1 Robust Homography Estimation

The 8 points picked from the robot trajectory are subject to noise and localization error. To calculate a homography matrix that is robust to these errors, we use Random Sampling Consensus (RANSAC) which enables model estimation from data that may contain outliers. Our adaptation of RANSAC goes through a number of iterations to calculate the best possible homography model. At each iteration, 8 perimeter points are chosen, one from each cluster obtained by k-means clustering (see Section 4 and Figure 3(b)) and are then used to calculate the homography. If the quality of homography is better than the previous iteration, then the model is preserved; otherwise the previous model is kept. This process is repeated until either one of 2 things happens: the number of iterations exceeds the pre-defined limit or the difference in the error measure between the current and previous iterations falls below a threshold. This error measure is the quality of homography estimation and is determined from the Hausdorff distance between the

original points in ground plane trajectory and the points obtained by projecting the image plane points onto the ground plane using the calculated homography matrix. The forward hausdorff distance between two point sets A and B is given by the maximin function in equation 6 and the reverse hausdorff distance, by the equation 7, where $d(a, b)$ is the Euclidean distance between points a and b. The sum of $D(A, B)$ and $D(B, A)$ is taken as our error measure, called the hausdorff error measure. The hausdorff error is a much better measure of finding the shape similarity of two point sets (indeed it is used for image comparison purposes by [Huttenlocher and Rucklidge, 1993]) as opposed to a measure like the sum of the errors for all re-projected points. The smaller the hausdorff distance between two point sets, the more similar they are in shape.

$$D(A, B) = max_{a \epsilon A} \left[min_{b \epsilon B} \left[d(a, b) \right] \right] \quad (6)$$

$$D(B, A) = max_{b \epsilon B} \left[min_{a \epsilon A} \left[d(b, a) \right] \right] \quad (7)$$

Note that our version of the RANSAC algorithm does not refine the model at each iteration by re-calculating it from all inliers (including interior points in the data set). In experiments conducted, this did not improve the modelling, probably because the points in the interior do not contribute constructively to model estimation.

5 Visual Servoing

It has been experimentally determined the trajectory points need to be well spread out in the image and non-linear for better homography accuracy. To this end, an algorithm is needed that keeps the robot within the confines of the camera who's homography is being calculated and also maximizes the spread of the robot trajectory. We could keep the robot wandering from obstacle to obstacle (including the virtual obstacle of the image boundary) and continue calcating the homography till we were satisfied with the quality (which could be the Hausdorff error measure in Section 4.1), but this would be wasteful in both time and robot resources (imagine the robot had 10 cameras to calibrate).

To keep the robot orbiting the image centre, we use an algorithm that is inspired by the chemotaxis behaviour of the silkworm moth as it tries and finds the source of a pheromone plume emitted by its mate. The moth has two antennae, and to keep within a plume blowing downwind, it turns in the direction of the antenna picking up a higher concentration of pheromones. It casts from side to side and follows an oscillatory trajectory until it finds the plume source. This behaviour has been duplicated in chemical sensing robots in [Kuwana *et al.*, 1996].

In our version of the chemotaxis algorithm, we use virtual antennae cast on either side of the robot, in the direction of its orientation in the image plane. These virtual antennae sample a binary image (of the same size as the image from the camera) with an ellipse in the middle of it, within which the robot is to be confined. The robot is kept on the margins of the ellipse (and hence on the margins of the camera FOV) by turning towards the side of the antenna which samples the inside of the ellipse (a value of 255). An elipse is chosen because it is a shape without edges that occupies the maximum space within a 320x240 image that enables the robot to maintain a smooth trajectory around the periphery of the image. In the case when both antennae sample the inside of the ellipse, the robot turns in a direction towards the antenna that is more distant from the image centre. The angle by which it turns depends on a function (given in Equation 8) of its distance from the image centre.

$$\psi_{servoGoal} = \psi_{max}(1 - (d_1/d_{max})) + \psi_{random} \quad (8)$$

where d_1 is the distance of the robot from the image centre, d_{max} is the maximum allowable distance from the image centre, ψ_{max} is the maximum angle it is allowed to turn by and ψ_{random} is a random turn angle.

This turn angle, $\psi_{servoGoal}$ is then passed on to the goal seeking part of the behaviour arbitration scheme (discussed in Section 6). Whilst the robot is being visually servoed towards the image periphery, obstacles on the ground plane are avoided using the avoid behaviour of the same scheme. Whenever one or both of its antennae sample the inside of the ellipse, i.e., it is within the camera FOV, its latest ground plane position is recorded. When the robot nears the edge of the ellipse, with one antenna inside and one outside, it turns in the direction of the antenna inside the ellipse so that both its antennae enter the ellipse again. Should it for any reason move completely outside the camera FOV, it navigates to the last known ground plane position recorded whilst it was still within the ellipse.

Figure 4 shows a path that the robot might take using this algorithm in the absence of any ground plane obstacles. The ellipse in the image shows the area that the robot is to be restricted to in the camera FOV. The gradient in ellipse colour from maroon to red to light pink displays the fact that the robot turns by different values toward the periphery of the ellipse depending on its distance from the centre. Note that in experiments the Euclidean distance of the robot from the centre of the image determines this turn angle (given by Equation 8), and its gradient expands outwards in a circular fashion, but for purposes of illustration, the gradient has been shown to be ellipsoid around the centre in this figure.

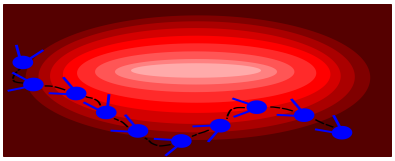


Figure 4: Visual Servoing the robot to keep it within camera FOV and on its periphery (along the ellipse). Virtual antennae help to move the robot within the ellipse and the colour gradient indicating regions of same turn angle ensures the robot favours a trajectory that is spread out along the image margins.

6 Navigation

The robot navigates through its environment using a mixture of global path-planning and reactive local obstacle avoidance. Global path planning involves using the distance transform algorithm to find a path from the robot's current location to its destination, which could be the location of an intruder that the system wants the robot to intercept. The Distance Transform algorithm gives a path that is broken down into regularly spaced waypoints. For patrol, the user selects a series of waypoints for the robot to navigate to. Given a waypoint, the robot uses a reactive behavioural arbitration scheme to drive the robot smoothly to the waypoint whilst avoiding obstacles. Goal finding and obstacle avoidance are implemented as two competing behaviours that generate dynamics (rate of change of bearing of the robot). Their weighted summations are used to produce a control output for the robot, which is a rate of change of bearing from the robot's current bearing. This dynamical systems approach to behaviour arbitration, first published by [Althaus and Christensen, 2002] provides a smooth control output despite the possible contrasting requests of the two behaviours. More details on the distance transform and the behavioural arbitration scheme are available in [Jarvis, 1994] and [Althaus and Christensen, 2002; Chakravarty *et al.*, 2006] respectively.

7 Experiments

This section details the experiments that have been performed to demonstrate all the ideas discussed in the previous sections. It is divided into two subsections. In the first subsection, 10 experiments demonstrate the ability of the Visual Servoing algorithm to keep the robot within the camera FOV for 2 cameras and successfully calculating the homographies for those cameras to the ground plane. The second subsection describes 3 experiments in which the robot patrols the environment and then intercepts intruders observed by either of the external cameras. During the course of these experiments,

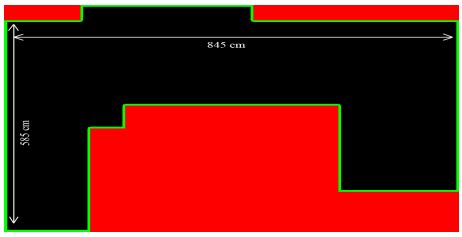


Figure 5: Map used for experiments in section 7.

the intruder keeps moving about from one camera to the other and the system is able to direct the robot to the corresponding ground plane locations.

7.1 Visual Servoing Experiments

10 experiments were performed in which the robot was asked to first, globally localize its position on the map of the ground plane, and once localized, navigate to a series of operator-defined patrol waypoints. The robot environment was viewed by 2 non-overlapping cameras. Once the robot was "viewed" from an external camera, it began the visual servoing to get a maximally spread trajectory in the camera view. When there were enough points in the trajectory, the homography between the camera in question and the ground plane was calculated using the Robust Homography procedure. Once the first camera was calibrated, the robot resumed waypoint navigation until it is sighted in the second camera, following which the same procedure was repeated.

To ascertain the accuracy of the homography, the transformation matrix calculated in each of the 10 experiments was used to transform 8 "ground truth points" which were marked as crosses on the ground plane and whose ground plane positions were manually measured. These points were individually clicked on, in their image from the external camera and the homography matrix calculated was used to project the points to the ground plane. These ground plane coordinates were then compared with the manually measured coordinates to get error measures. The mean and maximum errors for the homographies calculated for cameras 1 and 2 in each of the 10 experiments are given in Figure 6. The robot did not once lose its localization in any of the 10 experiments. The maximum error in homography for the 8 points whose ground truth positions are known over 10 experiments for camera 1 is 71.46 cm, which occurs in experiment 10 while the maximum error for camera 2 is 86.80 cm, which occurs in experiment 1. The mean error for camera 1 ranges between 37.84 cm and 49.91 cm and the mean error for camera 2 ranges between 40.76 cm and 52.43 cm. This sub-metre accuracy is acceptable for our purposes, which is to navigate a robot to an intruder when seen by an external camera, experiments for which are described in the next section. Once within about a

Expt Idx	Cam 1		Cam 2	
	Max Error	Mean Error	Max Error	Mean Error
1	62.53	46.07	86.80	50.56
2	54.14	48.81	61.73	47.90
3	62.39	40.92	52.17	40.76
4	66.88	46.61	57.96	40.86
5	71.23	50.96	76.03	51.95
6	67.08	49.91	72.05	52.43
7	57.50	43.03	55.46	46.75
8	52.57	37.84	59.87	52.22
9	52.41	38.79	57.02	51.43
10	71.46	45.66	86.04	52.59

Figure 6: Maximum and mean re-projection error values (cm) for 8 ground truth points in cameras 1 & 2 in 10 experiments where the robot was visually servoed to remain inside the FOV of the external camera.

metre of the person, the robot's on-board sensors could take over for further pursuit.

7.2 Interception of Intruders

In the previous subsection, 10 experiments were conducted in which the robot autonomously calculated the homographies between the image and ground planes for 2 cameras positioned in unknown locations in the environment. Here this calculated homography is utilized by the robot to navigate to the ground plane position of a person detected in either of the two cameras.

In a series of experiments, the robot is tasked with the following:

1. Localizing from scratch and navigating a set of operator-defined waypoints during patrol.
2. Detecting intrusions using 2 external cameras looking down on this environment.
3. Using homography to find the corresponding ground plane location of intrusion.
4. Driving to the intrusion spot and intercepting the intruder.

Three experiments were recorded in which the robot was able to detect intrusions using two external cameras, determine intrusion locations on the ground plane and navigate to the required location. Each camera had its own homography for transforming intruder positions on its image plane to the common ground plane map of the environment. The camera views were not overlapping. In experiments 1 and 2, the intruder remained in view of one camera for the duration of the experiment and was intercepted by the robot. During experiment 3, the intruder repeatedly moved from one part of the environment to another, so at times, he was visible from the first camera and other times, from the second. The system was able to correctly navigate the robot to the intruder's location. Figures 8-13 show two of these interception instances. The remaining instances in this and the other two experiments can be viewed on the video files. The figures in this sequence are a montage of the two external camera views (stitched side by side, with camera 1 to the left of camera 2) and the localization image. The localization image is inserted into this stitched image as a semi-transparent layer so that the view from the external cameras is not totally obscured. The localization view shows the particle cluster around the correct location of the robot. The best pose of the robot is shown as a red arrow. When an intruder enters the scene and his ground plane location is found by the system using the homography for that camera, a path to the person, calculated using the distance transform is superimposed in white on the localization image. The trajectory of the targets being tracked in the external camera views is also shown. The robot, recognized by its red base is always shown using a white trajectory. Any other targets, assumed to be intruders have their trajectories in different colours. In this sequence of images, there is only one intruder and his trajectory is shown in green.

A brief description of the figures is descriptive of the experiment (download video from http://www.ecse.monash.edu.au/centres/irrc/video/acra09/CollaborativeSurveillanceInterceptionVideo.avi):

Figure 7: The robot is localizing from scratch (global localization) using the Particle Filter algorithm.

Figure 8: The robot is now confident of its position and has begun its patrol through the environment. It is visible in external camera 1.

Figure 9: The robot continues its patrol along a series of pre-set waypoints in the environment.

Figure 10: An intruder is tracked in camera 1. This information is passed on to the robot, which uses the homography matrix of camera 1 to determine the corresponding location on the ground plane. The Distance Transform algorithm is then used to plan a path from the robot's current location to the location of the intruder. This path is shown in white in the localization image.

Figure 11: The robot has turned around and is approaching the intruder.

Figure 12: The intruder moves away from the robot, and goes to another part of the environment, visible from camera 2. He is tracked in the image plane of camera 2 and the system uses the homography matrix of this camera to infer his corresponding ground plane location. The robot now plans a path to this new location.

Figure 13: The robot has intercepted the intruder at

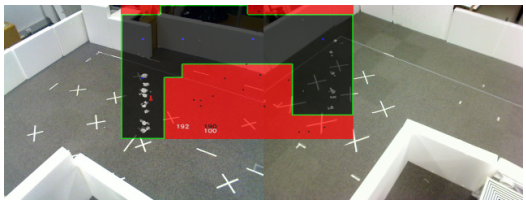


Figure 7: Figure shows views from cameras 1 and 2 stitched side by side. Robot appears in view of camera 1. Inset (in semi-transparent layer) shows robot location on ground plane. The robot is performing global localization.

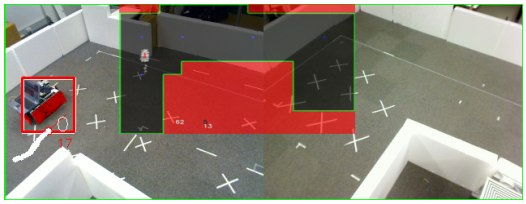


Figure 8: Robot appears in view of camera 1. The robot has finished global localization and is now doing position tracking and begins patrol.

this new location.

These experiments demonstrate the ability of the robot to use its autonomously calculated homography to navigate to a person detected by the external cameras. The system is correctly able to track the intruder in either of the external cameras and use the calculated homography for the relevant camera to navigate the robot to him. When the intruder moves between cameras, the system is able to calculate a new ground plane position for him and redirect the robot there.

8 Conclusion

A collaborative surveillance system has been presented in this paper that utilizes a mobile robot to establish a mapping between camera image planes and a ground plane coordinate system. The robot is localized in the image plane of the external cameras and is also localized on a ground plane map. This simultaneous localization gives the system a series of corresponding points on the

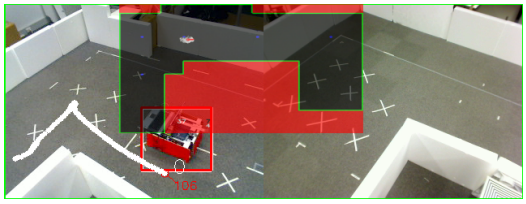


Figure 9: Robot continues patrol.

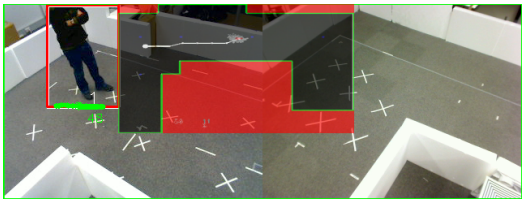


Figure 10: Intruder appears in view of camera 1. Inset robot localization view shows, in addition to the robot's location, the path to the intruder on the ground plane.

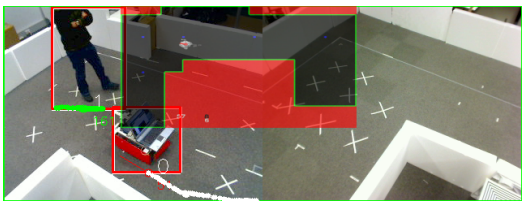


Figure 11: Robot turns around and approaches intruder.

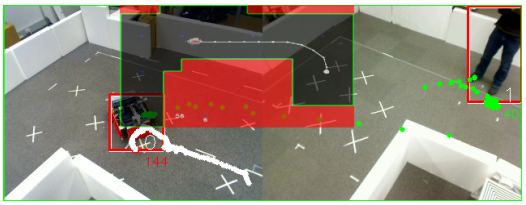


Figure 12: Intruder has now moved to camera 2. Robot turns around. Inset shows path to new position of intruder.

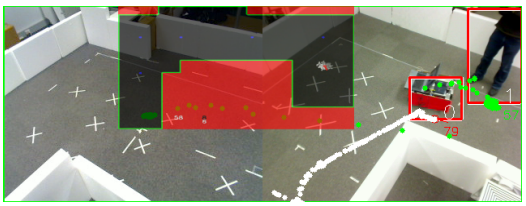


Figure 13: Robot intercepts intruder.

image and ground planes, which it uses to calculate a homography or mapping between the two. The robot starts navigating a series of pre-defined waypoints. As soon as it enters the field of view of a camera, it commences a visual servoing procedure that makes it orbit the centre of the image and prevents it from leaving the image margins. Once calibration is completed for one camera, it proceeds to the next camera. Experiments in this paper have been conducted with two cameras. Ten experiments were conducted for both cameras where the system-calculated homographies were compared with homographies calculated using ground truth points. The error was determined to be half-metre or less; adequate for navigating our robot to an intruder. Three further experiments were carried out in which the robot successfully intercepted intruders using the calculated homographies. In one of the experiments, the intruder kept moving between the two cameras and the system was successfully able to make the robot pursue the intruder. In future work, the accuracy of the robot detection in the image plane will be improved, which will in turn improve the accuracy of the homography. Also, our approaches for people tracking from a moving panoramic camera [Chakravarty and Jarvis, 2008] and a moving perspective camera [Chakravarty *et al.*, 2006] will be integrated into this system so that once the robot gets within the vicinity of the intruder, on-board sensing could take over the tracking.

References

[Althaus and Christensen, 2002] Philipp Althaus and Henrik Christensen. Behaviour coordination for navigation in office environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[Chakravarty and Jarvis, 2008] Punarjay Chakravarty and Ray Jarvis. People tracking from a moving panoramic camera. In *Australian Conference on Robotics and Automation (ACRA)*, 2008.

[Chakravarty *et al.*, 2006] Punarjay Chakravarty, David Rawlinson, and Ray Jarvis. Person tracking, pursuit & interception by mobile robot. In *Australasian Conference of Robotics and Automation (ACRA)*, 2006.

[Chen *et al.*, 2007] Huiying Chen, Kohsei Matsumoto, Jun Ota, and Tamio Arai. Self-calibration of environmental camera for mobile robot navigation. *Robotics and Autonomous Systems*, 55(3):177–190, 2007.

[Criminisi *et al.*, 1999] Antonio Criminisi, Ian D. Reid, and Andrew Zisserman. A plane measuring device. *Image Vision Computing.*, 17(8):625–634, 1999.

[Fox *et al.*, 2001] D. Fox, S. Thrun, W. Burgard, and F. Dellaert. *Sequential monte carlo methods in practice*, chapter Particle filters for mobile robot localization. Springer Verlag, 2001.

[Huttenlocher and Rucklidge, 1993] D.P. Huttenlocher and W.J. Rucklidge. A multi-resolution technique for comparing images using the hausdorff distance. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 705–706, 1993.

[Jarvis, 1994] Ray Jarvis. *Recent Trends in Mobile Robots*, chapter Distance Transform Based Path Planning For Robot Navigation, pages 3–31. World Scientific, 1994.

[Kuwana *et al.*, 1996] Yoshihiko Kuwana, Isao Shimoyama, Yushi Sayama, and Hirofumi Miura. Synthesis of pheromone-oriented emergent behavior of a silkworm moth. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1722–1729, 1996.

[Leonard and Durrant-Whyte, 1991] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. *IEEE International Workshop on Intelligent Robots and Systems*, pages 1442–1447, 1991.

[Rawlinson *et al.*, 2004] D. Rawlinson, P. Chakravarty, and R. Jarvis. Distributed visual servoing of a mobile robot for surveillance applications. In *Australasian Conference for Robotics and Automation*, 2004.

[Rekleitis *et al.*, 2006] Ioannis M. Rekleitis, David Meger, and Gregory Dudek. Simultaneous planning, localization, and mapping in a camera sensor network. *Robotics and Autonomous Systems*, 54(11):921–932, 2006.

[Rekleitis, 2002] I.M. Rekleitis. A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, McGill University, 2002.

[Smith and Cheeseman, 1987] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1987.

[Taylor, 2004] Geoff Taylor. Automatic video surveillance and robotic early response: An evaluation of objectvideo vew. Technical Report MECSE-30-2004, Monash University, 2004.

[Thrun *et al.*, 2002] S. Thrun, M. Montemerlo, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people tracking. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. MIT Press, 2005.