International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:5, 2008

A Hybrid Distributed Vision System for Robot Localization

Hsiang-Wen Hsieh, Chin-Chia Wu, Hung-Hsiu Yu, and Shu-Fan Liu

***Abstract*—**Localization is one of the critical issues in the field of robot navigation. With an accurate estimate of the robot pose, robots will be capable of navigating in the environment autonomously and efficiently. In this paper, a hybrid Distributed Vision System (DVS) for robot localization is presented. The presented approach integrates odometry data from robot and images captured from overhead cameras installed in the environment to help reduce possibilities of fail localization due to effects of illumination, encoder accumulated errors, and low quality range data. An odometry-based motion model is applied to predict robot poses, and robot images captured by overhead cameras are then used to update pose estimates with HSV histogram-based measurement model. Experiment results show the presented approach could localize robots in a global world coordinate system with localization errors within 100mm.

***Keywords*—**Distributed Vision System, Localization, Measurement model, Motion model.

I. Introduction

THE main goal of localization is to estimate the robot poses in the environment whether maps of the environment are given or not [1]. When maps of the environment are available, robots perceive the environment with range sensors and localize themselves with sensor measurements and the given maps as well.

With accurate localization and flexible path planning techniques, efficient high-level collaboration behaviors between multiple robots can then be achieved. Sensors mounted on or external to the mobile platform can both achieve localization of robots.

Laser range finder and sonar are two of the most frequently used sensors mounted on robots to perceive the environment and localize themselves. Many researchers presented probabilistic approaches proved to improve the accuracy of single robot localization with these sensor types. People or some moving objects will block laser or sonar beams when robots move in an uncontrolled environment [5]. These self-localized methods based on onboard sensors and internal encoder readings. The sensor modalities have a number of problems because they are easily confused in highly dynamic environments. Many of the approaches presented in [1] require manmade landmarks [4], which means that the robot has to localize itself with the aids of landmarks placed in the environment. This makes it impractical to implement in real world environment when moving objects, for example, people, occlude landmarks.

For Distributed Vision Systems (DVSs), sensors are regarded as cameras mounted surround the environment and robots can then be localized or routed by multiple cameras [2]. As to traditional vision-based localization techniques, there is a pre-process step in which robots capture enough reference images from different locations in the environment and stores them in the database. When the robot moves around, it captures an image and compares that with those in the database, and then tries to match the most similar one to determine the robot pose [2,6,7]. The storage size requirement of the reference images will be the key issue for large-scale environments and not valid for unstructured environments with moving objects. A Distributed Vision System design [8] utilized multiple omni-directional cameras to track robot states with image matching. Another recent DVS development for robot poses tracking and path planning utilized learning approach [9]. A simulated environment with several cameras installed was designed to teach robots how to navigate in the simulated environment.

However, once robots are entirely occluded by obstacles, localizations of robots may be failed with general design of DVS. In this paper, a hybrid DVS integrating robot odometry and vision sensor data is presented to localize robots. The presented localization technology comprises of two stages. The first stage is prediction of robot poses with odometry information from encoders while the second one is to correct the predictions of robot poses with vision data from overhead cameras. The presented localization infrastructure is illustrated in Fig. 1. The Pose Estimator receives odometry information from robot and vision data from overhead cameras and then updates the robot pose with final pose estimate. Only one reference image data and no teaching processes are required for the Pose Estimator.

The next section describes the overall localization procedure for the Pose Estimator. We introduce the detailed hybrid

Manuscript received April 30, 2008.

Hsiang-Wen Hsieh is with the Mechanical and Systems Research Laboratories, Industrial Technology Research Institute, Hsin-Chu, 310 Taiwan (phone: +886-3-5918776; fax: +886-3-5913607; e-mail: hwhsieh@ itri.org.tw).

Chin-Chia Wu is with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsin-Chu, 300 Taiwan (e-mail: chia.ece93g@ nctu.edu.tw).

Hung-Hsiu Yu is with the Mechanical and Systems Research Laboratories, Industrial Technology Research Institute, Hsin-Chu, 310 Taiwan (e-mail: godo@ itri.org.tw).

Shu-Fan Liu is with the Department of Information Management, Yuanpei University of Technology, Hsin-Chu, 300 Taiwan (e-mail: d877804@ alumni.nthu.edu.tw).

localization processes in Section III. Thereafter, we describe the experiment results of robot localization. We make conclusions about experiments results and future extensions in Section V.

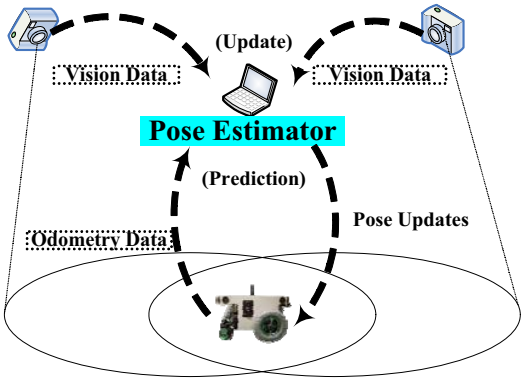


Fig. 1 Odometry and vision data integration

II. Localization Methodology

This section describes the overall localization procedure for the Pose Estimator and the coordinates transformation process from image plane to world coordinate.

A. Overall Localization Procedure

Fig. 2 depicts the overall procedure of the proposed hybrid DVS localization approach. There are four steps for robot localization: Initialization, Prediction, Update, and Resampling. In initialization stage, robot images captured from the overhead camera are converted into HSV (Hue, Saturation, Value) color space and then the corresponding normalized histograms based on color information are calculated. This is to identify the reference objects of interest to be tracked and can be done in either color calibrations for colors of interest or regions to be tracked within the image. Initial population of particles representing possible robot poses is then generated.

As to pose prediction, robot odometry data collected from encoders are used to generate robot pose prediction at time step t, say $State_t$. $State_t$ is equal to $(x_t, y_t\ \theta_t)$ which represents the position and orientation of the robot. In this stage, instead of auto-regressive models is generated for the prediction step (commonly used in vision-based tracking), but an odometry-based motion model is designed for prediction of robot poses. The predicted robot poses are on global world reference plane (WRP). These poses are then converted into image plane (IP).

After pose prediction step, particles representing possible robot poses on IP are determined. The region of interest (ROI) for each particle is determined for the calculation of weighting values. The weighting values indicate the similarity between ROI for each particle with the reference target, say robot on the image. The higher the weighting value is, the more likely certain particle (pose estimate) can represent true pose of the robot.

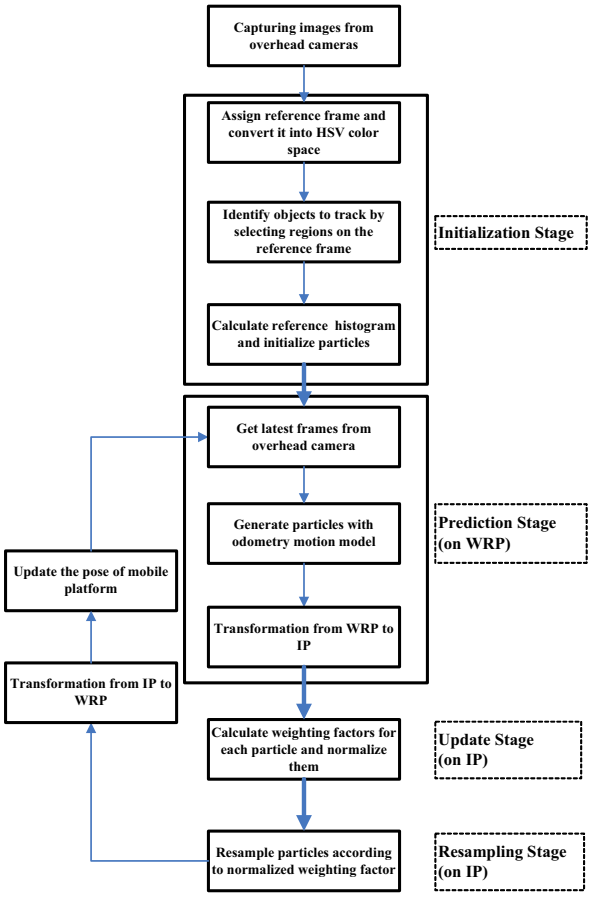


Fig. 2 Overall localization procedure

Resampling processes are then applied to generate a new set of particles according to the distribution of weighting values. Those particles with higher weighting values have higher probability to be duplicated. Therefore, the distribution of pose estimates will be converged in some small areas on IP. A dynamic resampling policy is designed and details are given in later section. All those results on IP should be converted back into WRP and the internal pose of robot at time step is updated.

B. Coordinates Transformation Process

The coordinates transformation is designed to transform pose estimates of particles from WRP to IP (at Prediction stage) or from IP to WRP (after Resampling stage). In order to complete the coordinate transformation process, overhead camera's intrinsic and extrinsic parameters in the environment should be calculated in advance. Extrinsic parameters represent the camera pose relative to the origin of WRP.

A separate calibration steps are performed to retrieve camera intrinsic and extrinsic parameters and Bouget camera toolbox [3] is used. Intrinsic parameters are independent of the position and orientation of the cameras in space. They will not be updated if camera's focus is not changed. A checkerboard patterns is used for the calibration and then extrinsic parameters

can be retrieved, which are the position and orientation of the camera written in the form of a rigid body transformation. Images of the testing environment are captured for camera calibrations and the camera's intrinsic and extrinsic parameters are obtained at Step 1, shown in Fig. 3. From Step 2 to Step 3, the transformation matrix that relates IP and WRP is obtained.

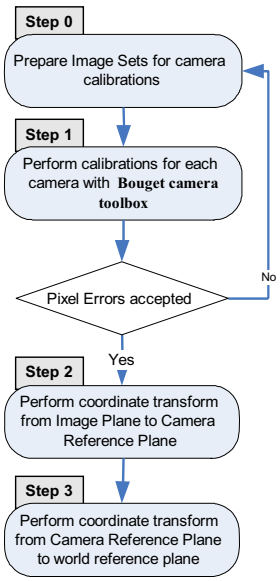


Fig. 3 Coordinate transformation process

The obtained camera parameters are inputs for derivations of transformation matrix from IP to WRP. To have one unique transformation solution from IP to WRP, additional input of the robot height should be given. Equation (1) represents relations between camera reference plane (CRP) and WRP. R_c and T_c are extrinsic parameters obtained and Z_w indicates a constant value, robot height. Equation (2) is the transformation from CRP to IP, where K represents intrinsic parameters. f_x and f_y are the camera's focal distance in units of horizontal and vertical pixels, respectively. (u_0, v_0) is coordinates of the camera's optical center. XX_c, XX_i, and XX_w are coordinates on CRP, IP, and WRP, respectively. With these inputs and equation (1) & (2), equation (3) is derived that directly relates IP to WRP.

$$XX_c = R_c XX_w + T_c \tag{1}$$

$$\begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c / Z_c \\ Y_c / Z_c \\ 1 \end{bmatrix} \tag{2}$$

$$XX_w = R_c^- XX_c - R_c^- T_c = R_c^- K \begin{bmatrix} X_i Z_c \\ Y_i Z_c \\ Z_c \end{bmatrix} - R_c^- T_c \tag{3}$$

Where,

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$XX_c = [X_c; Y_c; Z_c]$$
$$XX_i = [X_i; Y_i]$$
$$XX_w = [X_w, Y_w, Z_w]$$

III. Hybrid DVS Localization

Fig. 2 depicts the overall localization procedure that comprises of Prediction, Update and Resampling stages. The presented hybrid DVS localization utilized particle filter algorithm as illustrated in Fig. 4. Step 1 is the prediction of robot poses that generates a particle population representing pose estimates for the robot. An odometry-based motion model is designed for the prediction. For the update stage at Step 2, no sonar range sensors are equipped on the testing robot as most researches did, but external overhead vision sensors in the environment instead. An HSV histogram-based model is utilized to evaluate the quality of pose predictions from Step 1. Resampling process is applied at Step 3 that generates new particle population according to particles' weighting values from Step 2.

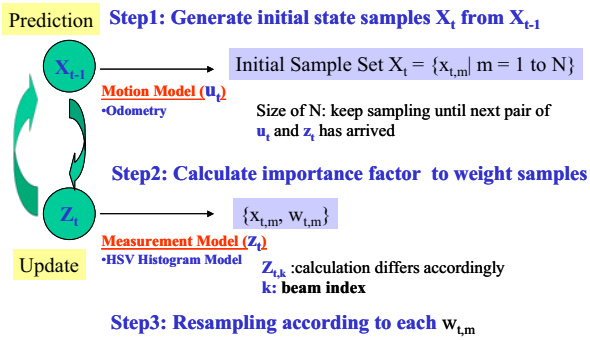


Fig. 4 Particle filter algorithm for robot pose estimate

A. Prediction Stage: Motion Model Design

Normally the motion model of a wheel-based robot can be modeled by velocity or odometry based motion commands. In this paper, an odometry based motion model is designed for robot pose prediction. Suppose $X_{t-1} = (x, y, \theta)^T$ is the robot pose estimate at time step $t-1$ and $u_t = (\hat{x}_{t-1}, \hat{x}_t)$ is the control command based robot odometry at time step $t-1$. $\hat{x}_{t-1} = (\hat{x}, \hat{y}, \hat{\theta})^T$ and $\hat{x}_t = (\hat{x}', \hat{y}', \hat{\theta}')^T$ are odometry measurements from encoders of the robot. $X_t = (x', y', \theta')^T$ is

the pose estimate at time step t with X_{t-1} and u_t. In other words, X_t is predicted by X_{t-1} plus relative motion measured by robot odometry and noise terms. Fig. 5 illustrates the relative motion of robot from pose A to pose B, respectively.

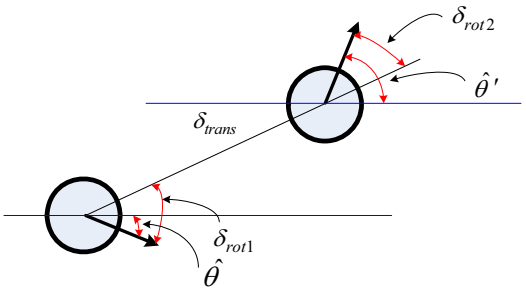


Fig. 5 Motion model for mobile platform

δ_{trans} is the translational distances between A and B. δ_{rot1} and δ_{rot2} are rotational motions from the control commands. These values can be calculated with equation (4) to (6). $X_t = (x', y', \theta')^T$ in equation (7) represents pose estimate at time step t. X_t is calculated iteratively with pose estimates at previous time step, control command, and noise terms . $\hat{\delta}_{trans}$, $\hat{\delta}_{rot1}$, and $\hat{\delta}_{rot2}$ represent the predicted translational distance and rotational motion with noise terms added. α_i ($i = 1 \sim 4$) are parameters used to control the calculation of variances, V_j ($j = 1 \sim 3$). Normal distribution, $N(0, V_j)$, is selected to draw random numbers for the noise terms.

$$\delta_{rot1} = ATan2(\hat{y}' - \hat{y}, \hat{x}' - \hat{x}) - \hat{\theta} \tag{4}$$

$$\delta_{trans} = \sqrt{(\hat{x} - \hat{x}')^2 + (\hat{y} - \hat{y}')^2} \tag{5}$$

$$\delta_{rot2} = \hat{\theta}' - \hat{\theta} - \delta_{rot1} \tag{6}$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \cos(\theta + \hat{\delta}_{rot1}) & 0 & 0 \\ 0 & \sin(\theta + \hat{\delta}_{rot1}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} \hat{\delta}_{trans} \\ \hat{\delta}_{trans} \\ \hat{\delta}_{rot1} + \hat{\delta}_{rot2} \end{bmatrix} \tag{7}$$

Where,

$$\hat{\delta}_{rot1} = \delta_{rot1} - N(0, V_1)$$
$$\hat{\delta}_{trans} = \delta_{trans} - N(0, V_2)$$
$$\hat{\delta}_{rot2} = \delta_{rot2} - N(0, V_3)$$
$$V_1 = \alpha_1 |\delta_{rot1}| + \alpha_2 \delta_{trans}$$
$$V_2 = \alpha_3 \delta_{trans} + \alpha_4 (|\delta_{rot1}| + |\delta_{rot2}|)$$
$$V_3 = \alpha_1 |\delta_{rot2}| + \alpha_2 \delta_{trans}$$

B. Update Stage: HSV Histogram-Based Measurement Model and Resampling

Each X_t represents a particle (pose estimate) on WRP. Coordinate Transformation Process in Section II-B are then applied to it from WRP to IP as shown in Fig. 6. Once all the particles have been transformed to IP, a measurement model is then applied to calculate weighting values. An HSV Histogram-Based measurement model is applied to determine the weighting value for each particle. For images captured from the vision sensor, they are converted to HSV color space for subsequent HSV histogram value calculations. As Hue ranges from 0.0 to 1.0, the corresponding colors vary from red, yellow, green, cyan, blue, and magenta, back to red. Saturation varies from 0.0 to 1.0, the corresponding colors vary from unsaturated to saturated (no white component). For Value (brightness), varies also from 0.0 to 1.0, the corresponding colors become increasing brighter [12]. Let nH, nS, and nV denote number of color sectors for individual HSV planes from image. Total number of discrete histogram bins, that represent particle's color distribution, is defined as $nH * nS + nV$.

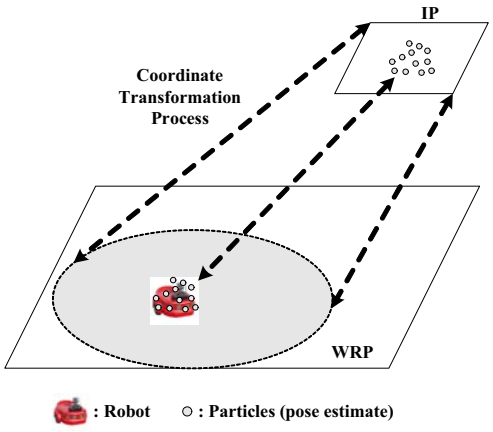
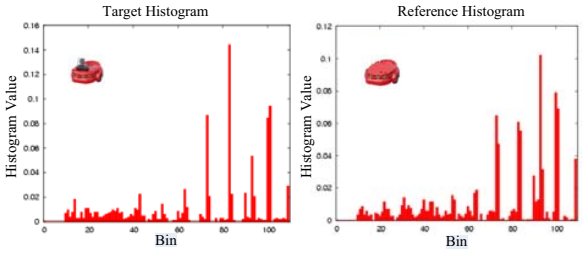


Fig. 6 Particle space transformation

Fig. 7 shows bins and histogram values that represent the HSV color distribution for one particle. Fig. 7 (a) is the target histogram (a particle enclosed by rectangle checking area, ROI) while (b) is the reference histogram (representing robot template). Comparison of these two histograms yields the weighting value for the particle. The weighting value indicates similarity of the projected particles and the reference model. If color distribution within the region of interest (ROI) covering the particle is similar to that of reference image, higher weighting value is obtained. To specifically calculate the similarities between these two distributions, Bhattacharyya distance [11] is adopted to measure similarity of two discrete probability distributions r and s.



(a) (b)

Fig. 7 Histogram-based measurement model

Equation (8) is Bhattacharyya distance calculation. The calculated distance value is the weighting value for each particle. These weighting values are then used to determine the priority and resampling policy for particle population at next time step.

$$Dsq(r,s) = \sum_{x \in X} \sqrt{r(x)s(x)} \tag{8}$$

The number of particles required for robot poses estimates at each time step determines accuracy of pose estimates and total image frames the system can process per second. A resampling policy is designed to flexibly adjust the number of particles according to the calculated weighting values. Let $Dsq_i^t(r,s_i)$ be the calculated weighting value of the i^{th} particle at time step t, $numParticles$ be the initial variable that represents number of particles, and $pFactor^t$ be the adjustment variable for particle numbers at time step t. The adjusted particle numbers at time step t, $adjParticles^t$, can then be calculated as equation (9), where $pFactor^t$ is determined by equation (10):

$$adjParticles^t = numParticles^t * pFactor^t \tag{9}$$

$$pFactor^t = \begin{cases} 1, & if\ \max(Dsq_i^t) > \lambda \quad (i=1...numParticles^t) \\ \sqrt{1/\max(Dsq_i^t)}, & otherwise \end{cases} \tag{10}$$

Where, $(0 < \lambda <= 1)$

IV. Experiments Discussions

Two general web cameras, LogicTech pro5000, and one wheel-based mobile platform equipped with encoders are selected for the experiments. To verify the effectiveness of the presented hybrid DVS localization methodology, three experiment scenarios are given. The first experiment is to compare the robot pose from encoders and the pose estimates from presented localization methodology, the second one is to verify if the presented hybrid DVS can still accurately localize the mobile platform at some corner positions that are far away from the cameras, and the last experiment is conducted to compare localizations results from two cameras. Cameras configurations on the testing arena are shown as Fig. 8. Two cameras are located at different pan/tilt angles and are calibrated separately to obtain intrinsic and extrinsic parameters for the Coordinates Transformation Process. Each camera is capable of localizing the mobile platform at the same time so that the robot can be simultaneously localized with two cameras.

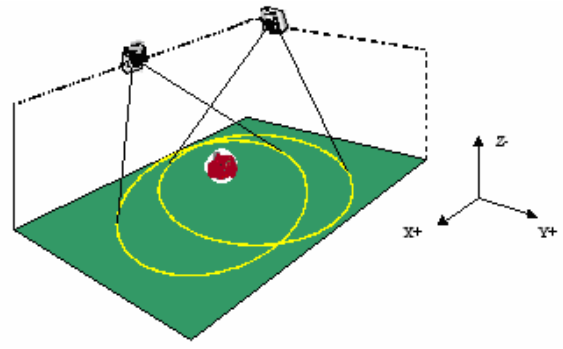


Fig. 8 Vision sensors configuration

Fig. 9 shows the predetermined routing path for the first experiment. The numbers, 0, 1, 2, 3, and 4 indicate positions where rotational commands are given to the mobile platform. The routing sequence is: 0→ 1 → 2 → 0 → 3 →4 → 0. The starting node 0 is around the origin (X = 98.00, Y= 140.00) on WRP. Fig. 9 shows the estimated path generated from robot encoders (Original) and our approach (Corrected). Results show that when the robot moves back to the origin on WRP, the routing path estimated from robot encoders indicate the robot pose is (X =650.00, Y = -500.00). The localization errors are 845.17mm. The pose estimates from hybrid DVS is (X = 91.06, Y = 80.83) and resultant localization errors are 59.54mm.

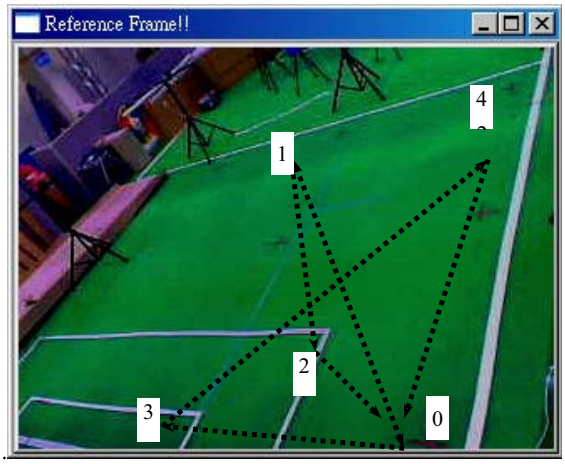


Fig. 9 Testing routing plan 1

The comparisons of recorded paths directly from encoders (labeled as Original) and the presented approach (labeled as Corrected) are shown in Fig. 10. Arrows on the figure indicate large differences exist between these recorded paths. It is evident that hybrid DVS integrating vision sensors and encoders on the robot can significantly correct accumulated localization errors.

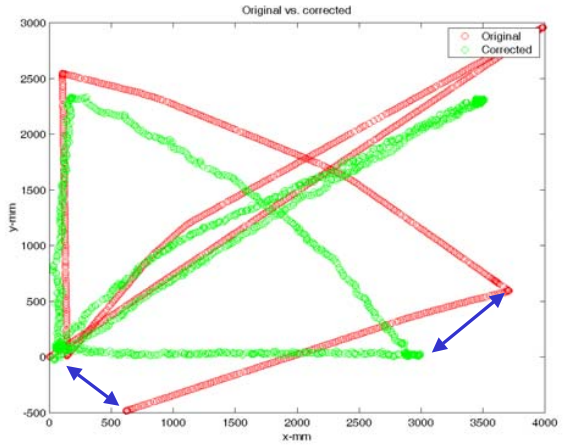


Fig. 10 Estimated path generated

The second experiment is designed to move the mobile platform around a 5m x 5m testing arena as shown in Fig. 11. Five positions (labeled as 0~4) are selected to record the pose estimates from hybrid DVS. Four of them are around the arena corners and the rest one is in the middle. The routing sequence is: 0 → 1 → 2 → 3 → 4 → 2 → 0. Table I summarizes the comparisons of physical position measured directly on the arena and pose estimates from hybrid DVS. The largest localization error is 201.31mm estimated at position 4. All the other four errors are all within 100mm and the average localization error is 79.69mm.

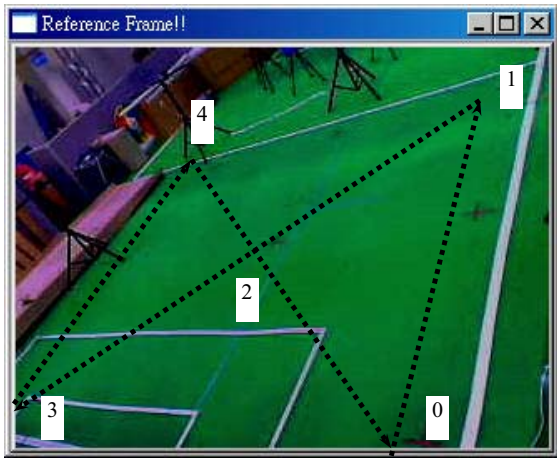


Fig. 11 Testing routing plan 2

TABLE I
LOCALIZATION RESULTS: TESTING ROUTING PLAN 2

No.	Physical Position		Estimated Results		Localization Errors
	X	Y	X	Y	
0	0.00	0.00	57	17	72.68
1	5000.00	0.00	4882.00	30.00	59.01
2	2500.00	2500.00	2448.00	2467.00	48.10
3	0.00	5000.00	79.00	4960.00	17.36
4	5000.00	5000.00	4852.00	4927.00	201.31
Avg. Localization Errors					79.69

The third experiment is to localize the robot with two cameras simultaneously at each time step. This scenario is to verify whether two cameras located at different locations can simultaneously localize the robot moving on the testing arena. The robot moves in S shape as shown in Fig. 12. Again, the robot moves from origin, make straight and circular motions and then go back to the starting point. Fig. 12 shows the localization results from the two overhead cameras. The maximum localization differences between cameras are around 150mm at outer region of the arena. This is reasonable because of possible distortions of images captured from general web cameras. This result can be very useful for deployments of multiple cameras to help localize mobile platform at even larger areas. The localization differences among cameras can potentially be reduced if cameras installed are pointed at similar target regions of the testing arena.

To sum up, the proposed localization methodology that integrates odometry and vision sensor information can greatly reduce the localization errors. The errors can be controlled within 100mm for most regions of the testing arena. For simultaneous localizations from multiple cameras, localization results can be further improved once cameras are deployed at suitable locations. Moreover, a weighting-based inference mechanism can be further designed and incorporated into hybrid DVS to fuse pose estimates from different cameras. This is the next step to enhance the capability of hybrid DVS for mobile platform localization and navigation.

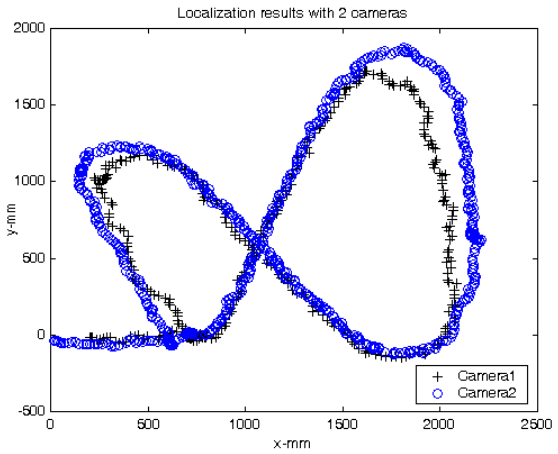


Fig. 12 Localization results: Camera1 vs. Camera2

V. Conclusion

In this paper, a Hybrid Distributed Vision System for robot localization is presented. Robot localization replies not only on vision information from overhead cameras but control commands from mobile platform.The experiments showed that with integration of encoders, vision sensor information and flexible resampling policy, the system could dynamically adjust the number of particles and localize mobile platform with localization errors small than 100mm. The localization speed of the proposed system is around 15 fames per second with 300 particles in average. Furthermore, the proposed system can be easily expanded to localize more than one robot. Future extensions of the paper are interactive dynamic path planning and intelligent inference mechanism for the integration of pose estimates from multiple cameras to increase the accuracy of pose estimates.

References

[1] J. Borenstein, B. Everett, and L. Feng, 1996, "Navigating Mobile Robots: Systems and Techniques," A. K. Peters, Ltd., Wellesley, MA.
[2] J. Gaspar, N. Winters, and J. Santos-Victor, 2000, "Vision-based navigation and environmental representations with an omnidirectional camera," IEEE Transaction on Robotics and Automation, Vol. 16(6).
[3] J. Y. Bouget, "Camera calibration toolbox for Matlab," http://www.vision.caltech.edu/bougetj/calib_doc.
[4] H. Everett, D. Gage, G. Gilbreth, R. Laird, and R. Smurlo, 1994, "Realworld issues in warehouse navigation," In Proceedings of the SPIE Conference on Mobile Robots IX, Vol. 2352.
[5] D. Fox, W. Burgard, S. Thrun, and A. B. Cremers, 1998, "Position estimation for mobile robots in dynamic environments," In Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence.
[6] T. Wilhelm, H. J. B¨ohme, and H. M. Gross, August 2004, "A multi-modal system for tracking and analyzing faces on a mobile robot," Robotics and Autonomous Systems,Vol. 48, pp. 31–40.
[7] J. Wolf, W. Burgard, and H. Burkhardt, 2005, "Robust vision-based localization by combining an image retrieval system with monte carlo localization," IEEE Transactions on Robotics, 21(2), pp. 208–216.
[8] E. Menegatti, G. Gatto, E. Pagello, T. Minato, H. Ishiguro, "Distributed Vision System for robot localization in indoor environments," Proc. of the 2nd European Conference on Mobile Robots ECMR'05 September 2005 Ancona – Italy, pp. 194-199.
[9] T. Sogo, H. Ishiguro and T. Ishida, 2001, ''Mobile robot navigation by a distributed vision system,'' New Generation Computing, Springer-Verlag.
[10] J. Bruce, M. Veloso, "Fast and accurate vision-based pattern detection and identification", Dept. of Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA.
[11] Y. Rui and Y. Chen, "Better proposal distributions: Object tracking using unscented particle filter," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, Hawaii, volume II, 2001, pp. 786–793.
[12] J. Sangoh, "Histogram-Based Color Image Retrieval," Available: http://scien.stanford.edu/class/psych221/projects/02/sojeong.

Hsiang-Wen Hsieh was born in 1973, Taiwan. He graduated from dept. of industrial management National Central University, Taiwan, 1998 as a master degree and received his PhD from dept. of Industrial Engineering and Engineering Management, National Tsing-Hua University, Taiwan, 2004. His research interests include engineering data analysis in semiconductor, image processing, and probabilistic robotics. Currently, he is a researcher at intelligent robotics control division, MSL/ITRI, Taiwan.