Trajectory Generation and Object Tracking of Mobile Robot Using Multiple Image Fusion

TaeSeok Jin[1] and Hideki Hashimoto[2]
[1]Dept. of Mechatronics Eng., DongSeo University,
[2]Institute of Industrial Science, the University of Tokyo
[1]Korea
[2]Japan

1. Introduction

Detection of moving objects has been utilized in industrial robotic systems, for example, in the recognition and monitoring of unmanned systems that also require compression of moving images [1], [2], [3], [4]. Trajectory prediction of moving objects is required for a mobile manipulator that aims at the control and observation of motion information such as object position, velocity, and acceleration. Prediction and estimation algorithms have generally been required for industrial robots. For a simple example, in a pick-and-place operation with a manipulator, the precise motion estimation of the object on the conveyor belt is a critical factor in stable grasping. A well-structured environment, such as the moving-jig that carries the object on the conveyor belt and stops when the manipulator grasps the object, might obviate the motion estimation requirement. However, a well-structured environment limits the flexibility of the production system, requires skillful designers for the jig, and incurs a high maintenance expense; eventually it will disappear from automated production lines. To overcome these problems, to tracking a moving object stably without stopping the motion, the trajectory prediction of the moving object on the conveyor belt is necessary [5]. The manipulator control system needs to estimate the most accurate position, velocity, and acceleration at any instance to capture the moving object safely without collision and to pick up the object stably without slippage. When the motion trajectory is not highly random and continuous, it can be modeled analytically to predict the near-future values based on previously measured data [6]. However, this kind of approach requires significant computational time for high-degrees-of-freedom motion, and its computational complexity increases rapidly when there are many modeling errors. In addition, performance is highly sensitive to the change of the environment. Those state-of-the-art techniques perform efficiently to trace the movement of one or two moving objects but the operational efficiency decreases dramatically when tracking the movement of many moving objects because systems implementing multiple hypotheses and multiple targets suffer from a combinatorial explosion, rendering those approaches computationally very expensive for real-time object tracking [7].
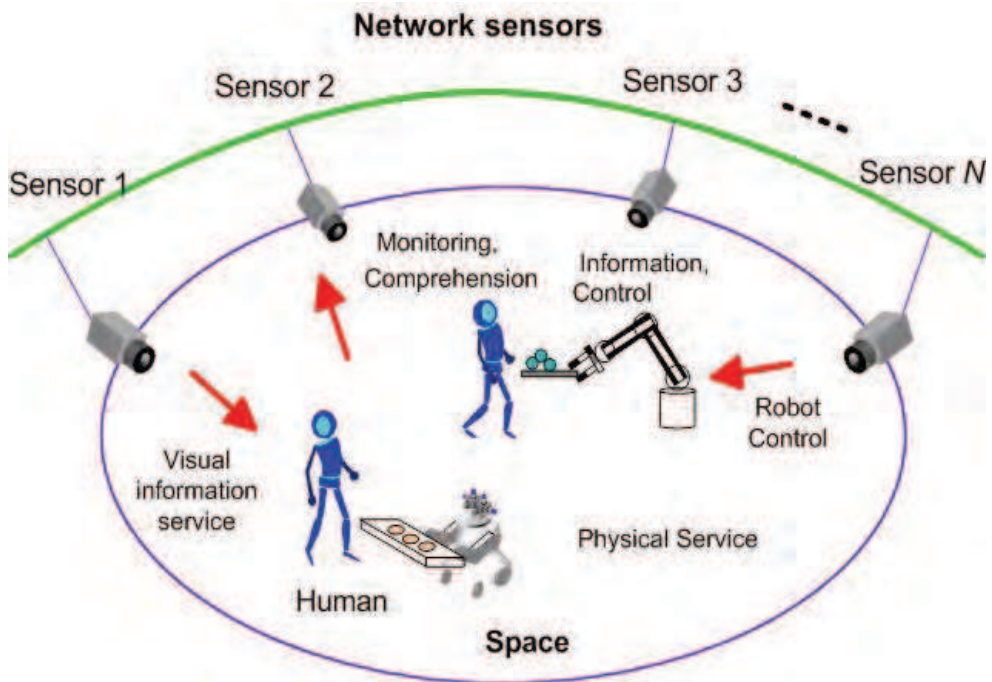
Fig. 1. Intelligent environment by distributed cameras

2. Vision systems in robotic space

2.1 Structure of robotic space

Fig. 2 shows the system configuration of distributed cameras in Intelligent Space. Since many autonomous cameras are distributed, this system is autonomous distributed system and has robustness and flexibility. Tracking and position estimation of objects is characterized as the basic function of each camera. Each camera must perform the basic function independently at least because over cooperation in basic level between cameras loses the robustness of autonomous distributed system. On the other hand, cooperation between many cameras is needed for accurate position estimation, control of the human following robot [8], guiding robots beyond the monitoring area of one camera [9], and so on. These are advanced functions of this system. This distributed camera system of Intelligent Space is separated into two parts as shown in Fig. 2. This paper will focus on the tracking of multiple objects in the basic function. Each camera has to perform the basic function independently of condition of other cameras, because of keeping the robustness and the flexibility of the system. On the other hand, cooperation between many cameras is needed for accurate position estimation, control of mobile robots to supporting human [10],[11], guiding robots beyond the monitoring area of one camera[12], and so on. These are advanced functions of this system.
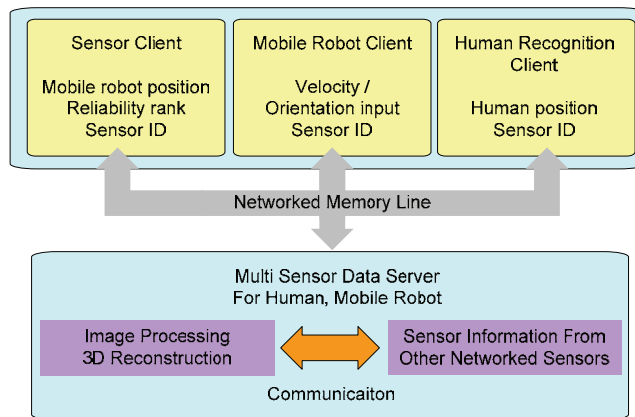
Fig. 2. Configuration of distributed camera system

2.2 Previous research for tracking

Neural networks can be classified into two categories: supervised learning- and unsupervised learning methods. In most of the previous research, the supervised learning method was adopted to overcome the nonlinear properties [10],[12]. Since the supervised learning method requires the relation between input and output [9] at all times, it is not suitable for real-time trajectory estimation for which the input-output relation cannot be obtained instantaneously in the unstructured environment. Therefore, in this study, SOM (Self Organizing Map), that is, a type of unsupervised learning method, was selected to estimate the highly nonlinear trajectory that cannot be properly predicted by the Particle filter. Also, SOM is a data-sorting algorithm, which is necessary for real-time image processing since there is so much data to be processed. Among the most popular data-sorting algorithms, VQ (Vector Quantization), SOM, and LVQ (Learning Vector Quantization), SOM is selected to sort the data in this approach since it is capable of unsupervised learning. Since VQ is limited to the very special case of a zero neighborhood and LVQ requires preliminary information for classes, neither of them is suitable for the unsupervised learning of the moving trajectory. Fig. 3 shows the estimation and tracking system for this research. The input for the dynamic model comes from either the Particle filter or SOM according to the following decision equation:

$$Predicted\ value = k \cdot Particle\ Filter_{out} + (1-k) \cdot SOM_{out}$$

where k=1 for $error \leq threshold$ and k=0 for $error > threshold$.

The threshold value is empirically determined based on the size of the estimated position error.

3. Processing flow

3.1 Extraction of object

Classifying the moving-object pattern in the dynamically changing unstructured environment has not yet been tackled successfully [13]. Therefore, in this research, the
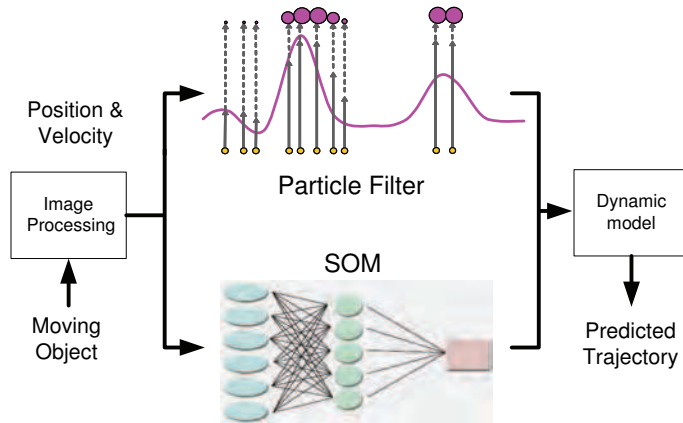
Fig. 3. Estimation and tracking system of robotic space

camera was fixed on a stable platform in order to capture static environment images. To estimate the states of the motion characteristics, the trajectory of the moving object was pre-recorded and analyzed. Fig. 4(a) and Fig. (b) represent the object images at (t-1) instance and (t) instance, respectively.

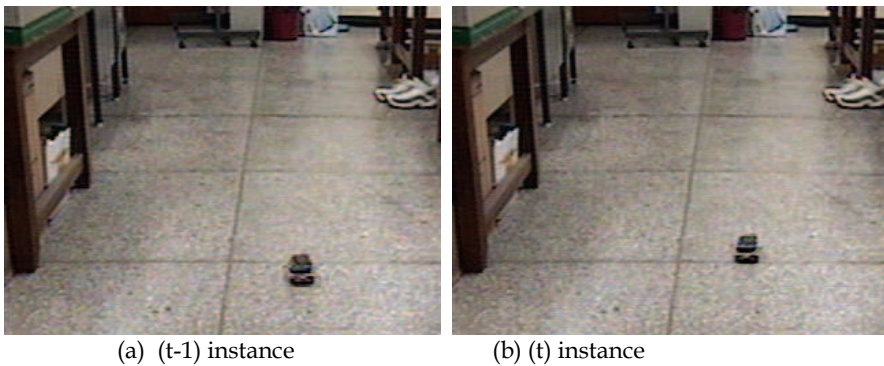(a) (t-1) instance   (b) (t) instance

Fig. 4. Detected image of moving object at each instance

As recognized in the images, most parts of the CCD image correspond to the background. After eliminating the background, the difference between the two consecutive image frames can be obtained to estimate the moving-object motion. To compute the difference, either the absolute values of the two image frames or the assigned values can be used. The difference method is popular in image pre-processing for extracting desired information from the whole image frame, which can be expressed as

$$\text{Output}(x, y) = \text{Image}1(x, y) - \text{Image}2(x, y)$$

The difference image between Fig. 4(a) and Fig. 4(b) is represented in Fig. 5. When the difference image for the whole time interval can be obtained, the trajectory of the moving object can be calculated precisely.
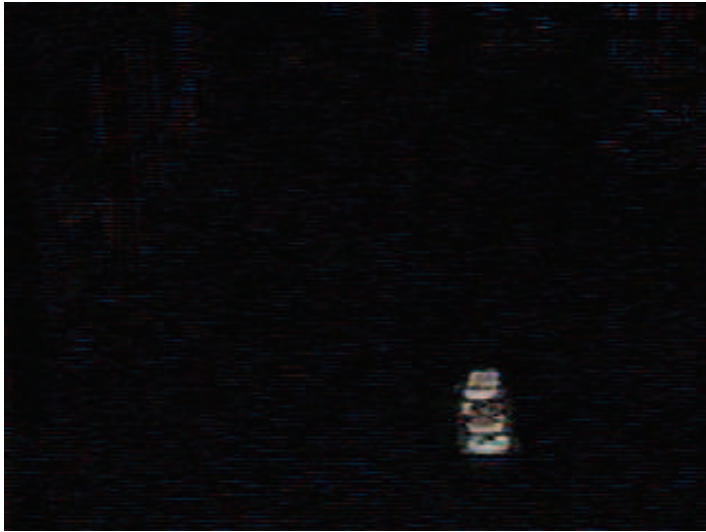
Fig. 5. Difference image between (t) and (t-1) instance images

3.2 Target regions encoded in a state vector using particle filter

Particle filtering provides a robust tracking framework, as it models uncertainty. Particle filters are very flexible in that they not require any assumptions about the probability distributions of data. In order to track moving objects (e.g. pedestrians) in video sequences, a classical particle filter continuously looks throughout the $2D$-image space to determine which image regions belong to which moving objects (target regions). For that a moving region can be encoded in a state vector. In the tracking problem the object identity must be maintained throughout the video sequences. The image features used therefore can involve low-level or high-level approaches (such as the colored-based image features, a subspace image decomposition or appearance models) to build a state vector. A target region over the $2D$-image space can be represented for instance as follows:

$$r = \{l,\ s,\ m,\ \gamma\}$$

where l is the location of the region, s is the region size, m is its motion and γ is its direction. In the standard formulation of the particle filter algorithm, the location l, of the hypothesis, is fixed in the prediction stage using only the previous approximation of the state density. Moreover, the importance of using an adaptive-target model to tackle the problems such as the occlusions and large-scale changes has been largely recognized. For example, the update of the target model can be implemented by the equation

$$\overline{r}_t = (1-\lambda)\overline{r}_{t-1} + \lambda \mathrm{E}[r_t]$$

where λ weights the contribution of the mean state to the target region. So, we update the target model during slowly changing image observations.

4. Tracking moving objects

4.1 State-space over the top-view plan

In a practical particle filter [5],[6] implementation, the prediction density is obtained by applying a dynamic model to the output of the previous time-step. This is appropriate when the hypothesis set approximation of the state density is accurate. But the random nature of the motion model induces some non-zero probability everywhere in state-space that the object is present at that point. The tracking error can be reduced by increasing the number of hypotheses (particles) with considerable influence on the computational complexity of the algorithm. However in the case of tracking pedestrians we propose to use the top-view information to refine the predictions and reduce the state-space, which permits an efficient discrete representation. In this top-view plan the displacements become Euclidean distances. The prediction can be defined according to the physical limitations of the pedestrians and their kinematics. In this paper we use a simpler dynamic model, where the actions of the pedestrians are modeled by incorporating internal (or personal) factors only. The displacements $M^t_{topview}$ follows the expression

$$M^t_{topview} = A(\gamma_{topview})M^{t-1}_{topview} + N$$

where A(.)is the rotation matrix, $\gamma_{topview}$ is the rotation angle defined over top-view plan and follows a Gaussian function $g(\gamma_{topview};\sigma_\gamma)$, and N is a stochastic component. This model proposes an anisotropic propagation of M: the highest probability is obtained by preserving the same direction. The evolution of a sample set is calculated by propagating each sample according to the dynamic model. So, that procedure generates the hypotheses.

4.2 Estimation of region size

The size of the search region represents a critical point. In our case, we use the *a-priori* information about the target object (the pedestrian) to solve this tedious problem. We assume an averaged height of people equal to 160 cm, ignoring the error introduced by this approximation. That means, we can estimate the region size s of the hypothetical bounding box containing the region of interest $r = \{l, s, m, \gamma\}$ by projecting the hypothetical positions from top-view plan in Fig. 6.
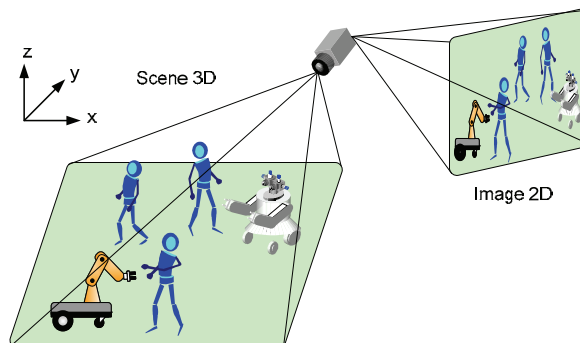

Fig. 6. Approximation of Top-view plan by image plan with a monocular camera

A camera calibration step is necessary to verify the hypotheses by projecting the bounding boxes. So this automatic scale selection is a useful tool to distinguish regions. In this way for each visual tracker we can perform a realistic partitioning (bounding boxes) with consequent reduction in the computational cost. The distortion model of the camera's lenses has not been incorporated in this article. Under this approach, the processing time is dependent on the region size.

4.3 Object model update

In multi-motion tracking, the hypotheses are verified at each time step by incorporating the new observations (images). A well known measure of association (strength) of the relationship between two images is the normalized correlation.

$$dc_{i,j} \ = \ corr_{nor}(t \arg et_i \ ; \ hypothesis_{i,j})$$

where i : target region, and j : an hypothesis of the target region i. The observation of each hypothesis is weighted by a Gaussian function with variance σ.

$$h^{(i;j)} \ = \frac{1}{\sqrt{2\pi}\sigma_{dc}} e^{\frac{-(1-dc_{i,j})^2}{2\sigma_{dc}^2}}$$

where $h^{(i;j)}$ is the observation probability of the hypothesis j tracking the target i. The obvious drawback of this technique is the choice of the region size (defined in previous section) that will have a great impact on the results. Larger region sizes are less plagued by noise effects.

5. Experiments

To compare the tracking performance of a mobile robot using the algorithms of the Particle filter and SOM, experiments of capturing a micro mouse with random motion by the mobile robot were performed. As can be recognized from Fig. 7, SOM based Particle Filter provided better performance to the mobile robot in capturing the random motion object than the other algorithms. As shown in Fig. 7, the mobile robot with SOM based Particle Filter has a smooth curve to capture the moving object. As the result, the capturing time for the moving object is the shortest with SOM based Particle Filter. Finally, as an application experiments were performed to show the tracking and capturing a mobile object in robotic space.

Fig. 8 shows the experimental results for tracking a moving object that is an 8x6[cm] red-colored mouse and has two wheels with random velocities in the range of 25-35[cm/sec]. First, mobile robot detects the moving object using an active camera. When a moving object is detected within view, robot tracks it following the proposed method.
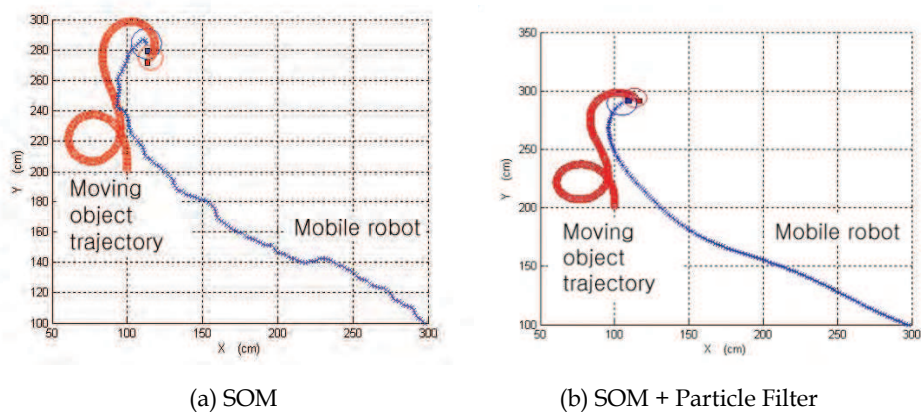
(a) SOM   (b) SOM + Particle Filter

Fig. 7. Tracking trajectory by SOM and SOM based particle filter
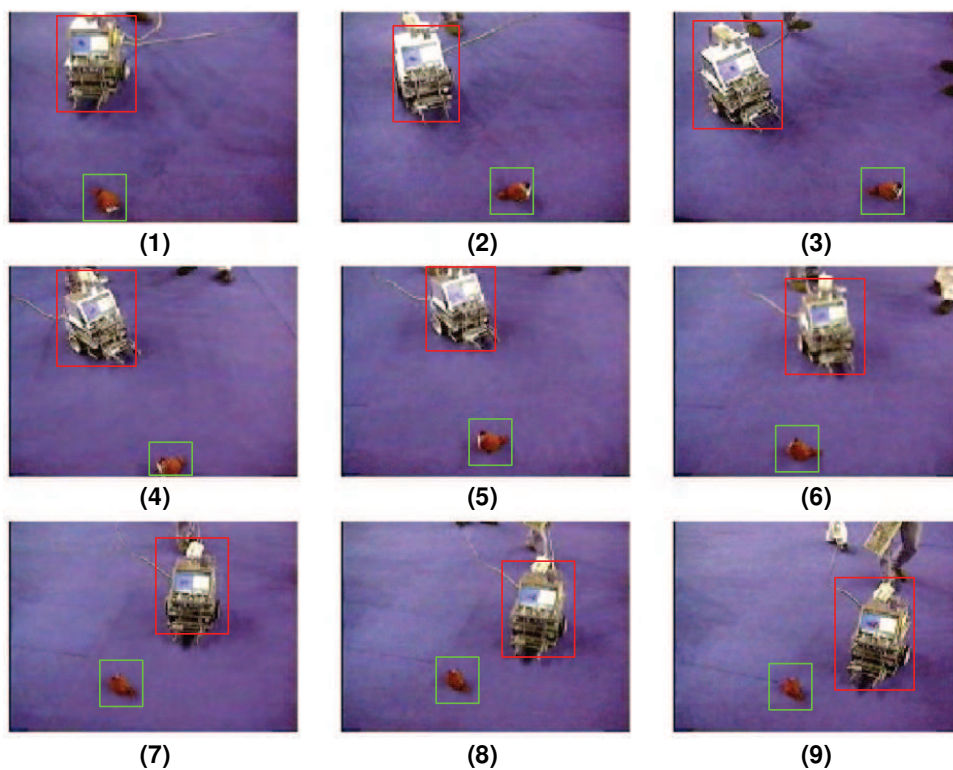

Fig. 8. Experimental results for tracking a moving object

6. Conclusions

In this paper, the proposed tracking method adds an adaptive appearance model based on color distributions to particle filtering. The color-based tracker can efficiently and successfully handle non-rigid and fast moving objects under different appearance changes. Moreover, as multiple hypotheses are processed, objects can be tracked well in cases of occlusion or clutter. This research proposes estimation and tracking scheme for a moving object using images captured by multi cameras. In the scheme, the state estimator has two algorithms: the particle filter that estimates the states for the linear approximated region, and SOM for the nonlinear region. The decision for the switchover is made based on the size of the position estimation error that becomes low enough for the linear region and becomes large enough for the nonlinear region. The effectiveness and superiority of the proposed algorithm was verified through experimental data and comparison. The adaptability of the algorithm was also observed during the experiments. For the sake of simplicity, this research was limited to the environment of a fixed-camera view. However, this can be expanded to the moving camera environment, where the input data might suffer from higher noises and uncertainties. As future research, selection of a precise learning pattern for SOM in order to improve the estimation accuracy and the recognition ratio, and development of an illumination robust image processing algorithm, remain.

7. Acknowledgments

8. References

[1] Senior, A.: Tracking with Probabilistic Appearance Models. In Proc. ECCV workshop on Performance Evaluation of Tracking and Surveillance Systems (2002) 48-55.

[2] Bierlaire, M., Antonini, G., Weber, M.: Behavioural Dynamics for Pedestrians," in K. Axhausen (Ed.), Moving through nets: the physical and social dimensions of travel, Elsevier (2003) 1-18.

[3] Nummiaro, K., Koller-Meier, E., Van Gool, L.J.: Object Tracking with an Adaptive Color-Based Particle Filter. DAGM-Symposium Pattern Recognition (2002) 353-360.

[4] P. K. Allen, A. Tmcenko, B. Yoshimi, and P. Michelman, "Trajectory filtering and prediction for automated tracking and grasping of a moving object," *IEEE International Conference on Robotics and Automation* (1992) 1850-1856.

[5] Yi Ma, J. Kosecka, and S. S. Sastry, "Vision guided navigation for a nonholonomic mobile robot," *IEEE Transaction on Robotics and Automation*, vol. 15, no. 3, pp. 521-536, 1999.

[6] Choo, K., Fleet, D.J.: People tracking using hybrid Monte Carlo filtering. In Proc. Int. Conf. Computer Vision, vol. II (2001) 321-328.

[7] Anderson, B., Moore, J.: Optimal Filtering. Prentice-Hall, Englewood Cliffs (1979).

[8] Kitagawa, G.: Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. Journal of Computational and Graphical Statistics, Vol. 5 (1996) 1-25.

[9] Yi-Yuan Chen and Kuu-young Young, "An intelligent radar predictor for high-speed moving-target tracking," *TENCON '02. Proceedings. IEEE Region 10 Conference on*

Computers, Communications, Control and Power Engineering, vol. 3, pp. 1638 -1641, 2002.

[10] J. M. Roberts, D. J. Mills, D. Charnley, and C. J. Harris, "Improved Kalman filter initialization using neuro-fuzzy estimation," *Int'l. Conf. on Artificial Neural Networks*, pp. 329-334, 1995.

[11] Norlund, P., Eklundh, J.O.: Towards a Seeing Agent. Proc. of First Int. Workshop on Cooperative Distributed Vision (1997) 93-120.

[12] Atsushi, N., Hirokazu, K., Shinsaku, H., Seiji, I,: Tracking Multiple People using Distributed Vision Systems. Proc. of the 2002 IEEE Int. Conf. on Robotics & Automation (2002) 2974-2981.

[13] Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: finder: Real-Time Tracking of the Human Body. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19 (1997) 780-785.

[14] Gardner, W.F., Lawton, D.T.: Interactive model based vehicle tracking. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.18 (1996) 1115-1121.

[15] Swain, M.J., Ballard, D.H.: Color indexing. Int. J. of Computer Vision, Vol.7, No.1, (1991)11-32.

Sensor and Data Fusion

Edited by Nada Milisavljevic

Data fusion is a research area that is growing rapidly due to the fact that it provides means for combining pieces of information coming from different sources/sensors, resulting in ameliorated overall system performance (improved decision making, increased detection capabilities, diminished number of false alarms, improved reliability in various situations at hand) with respect to separate sensors/sources. Different data fusion methods have been developed in order to optimize the overall system output in a variety of applications for which data fusion might be useful: security (humanitarian, military), medical diagnosis, environmental monitoring, remote sensing, robotics, etc.

How to reference

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

TaeSeok Jin and Hideki Hashimoto (2009). Trajectory Generation and Object Tracking of Mobile Robot Using Multiple Image Fusion, Sensor and Data Fusion, Nada Milisavljevic (Ed.), ISBN: 978-3-902613-52-3, InTech, Available from:
http://www.intechopen.com/books/sensor_and_data_fusion/trajectory_generation_and_object_tracking_of_mobile_robot_using_multiple_image_fusion

INTECH
open science | open minds