Multiple Moving Person Tracking Based on the IMPRESARIO Simulator

Hyun-deok Kim, TaeSeok Jin

Abstract**—**In this paper, we propose a real-time people tracking system with multiple CCD cameras for security inside the building. To achieve this goal, we present a method for 3D walking human tracking based on the IMPRESARIO framework incorporating cascaded classifiers into hypothesis evaluation. The efficiency of adaptive selection of cascaded classifiers has been also presented. The camera is mounted from the ceiling of the laboratory so that the image data of the passing people are fully overlapped. The implemented system recognizes people movement along various directions. To track people even when their images are partially overlapped, the proposed system estimates and tracks a bounding box enclosing each person in the tracking region. The approximated convex hull of each individual in the tracking area is obtained to provide more accurate tracking information.

We have shown the improvement of reliability for likelihood calculation by using cascaded classifiers. Experimental results show that the proposed method can smoothly and effectively detect and track walking humans through environments such as dense forests.

Index Terms**—**Person Tracking, Detection, CCD camera, Image processing, Subtraction.

I. INTRODUCTION

Real-time human tracking information is very useful source for security application as well as people management such as pedestrian traffic management, tourist flows estimation. To recognize and track moving people is considered important for the office security or the marketing research. Many of such measurements are still carried out on manual works of persons. Therefore it is necessary to develop the automatic method of counting the passing people.

Several attempts have been made to track pedestrians. Segen and Pingali [1] introduced a system in which the pedestrian silhouette is extracted and tracked. The system runs in real-time, however, the algorithm is too heavy to track many people simultaneously and can not deal well with temporary occlusion. Masoud and Papanikolopoulos [2] developed a real-time system in which pedestrians were modeled as rectangular patches with a certain dynamic behavior. The system had robustness under partial or full occlusions of pedestrians by estimating pedestrian parameters. Rossi and Bozzoli [3] avoided the occlusion problem by mounting the camera vertically in their system in order to track and count passing people in a corridor, but assumed that people enter the scene along only two directions (top and bottom side of the image). Terada [4] proposed a counting method which segmented the human region and road region by using the three dimensional data obtained from a stereo camera. However, this system also assumed only simple movement of pedestrians.

And also, it is difficult for these approaches to establish consistent label without overlapping of the monitoring areas among different cameras. Feature matching approaches based on the color or others are the simplest scheme to establish consistent labeling. However, color feature matching is not reliable when the disparity is large in location and orientation. In this paper, color appearance based object representation for the distributed vision system in the Intelligent Space is described. At first, vision system in Intelligent Space will be explained. Then, this paper will show how to learn the object color appearance model, track the multi-object under occlusions, and achieve the correspondence among different cameras [5].

In this paper, we propose a real-time people tracking system with multiple CCD cameras for security inside the building. The camera is mounted from the ceiling of the laboratory so that the image data of the passing people are fully overlapped. The implemented system recognizes people movement along various directions. To track people even when their images are partially overlapped, the proposed system estimates and tracks a bounding box enclosing each person in the tracking region. The approximated convex hull of each individual in the tracking area is obtained to provide more accurate tracking information. This paper is organized as follows: Section II describes the system architecture of the proposed people counting system. In Section III and IV present the real-time tracking system and tracking following detection, respectively. Experimental results and discussions are described in Section VI. Finally, conclusions are presents in Section VII.

Manuscript received February 13, 2008; revised June 18, 2008. Hyun-deok Kim is with the Dept. of Electronics Eng. , JinJu National University, Jinju, Korea (Tel: +82-55-751-3331, Email: hdkim@jinju.ac.kr), TaeSeok Jin is with the Dept of Mechatronics Eng. , DongSeo University, Busan, Korea (Tel: +82-51-320-1541, Email: jints@dongseo.ac.kr)

II. SYSTEM ARCHITECTURE

Fig. 1 shows a scene of the walking people through the corridor outside the building. There are incoming and outgoing individuals in the scene. Multiple cameras unit is hung from the ceiling of the laboratory so that the walking people can be observed and tracked in a tracking area in front of the door. The images captured by the cameras are processed and the number of the passing people is calculated.

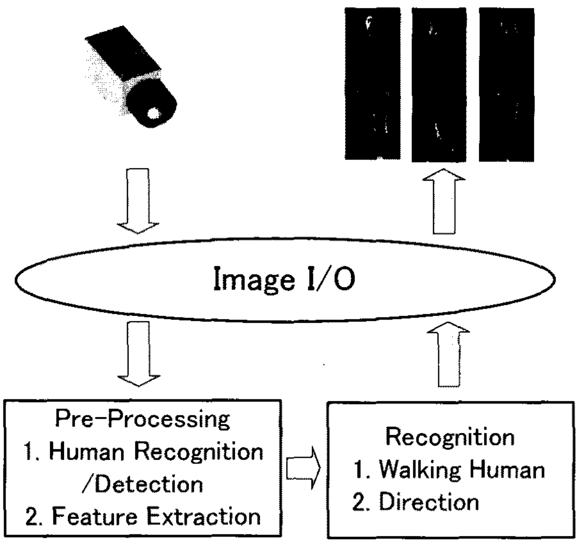


Fig. 1 Object Finding Process

To cope with inherently dynamic phenomena (people enter the scene, move across the field of view of the camera, and finally cross the counting line), the people recognizing and tracking problem has been decomposed into the following three steps: [3][6]

- Determine whether any potentially interesting objects have entered into the scene (Alerting phase);
- Track their motion until the counting line is reached (Tracking phase);
- Establish how many people correspond to tracked objects (Interpretation phase).

A. Software Solution

The accuracy and relatively high frequency requirements of the final application dictated that a software system be selected that provided for simultaneous control and acquisition of data from multiple instruments in a coordinated fashion. The core design architecture of the SpatialAnalyzer package provided all of the above capabilities. To ease operator burdens, a simplified interface was developed that enables an operator to control multiple laser trackers from a single user interface. The operator can control all measurement functions by interacting with just one Windows dialog. This interface, referred to as "TransTrack", allows one user to control the three trackers in a coordinated fashion to track a six degree-of-freedom transformation. The TransTrack application then controls three other slave applications that are essentially minimal tracker control modules.

These individual control modules have been named "MiniTrack". Given the large working area and the complexities associated with three laser trackers operating simultaneously, text-to-speech technology is used heavily in the TransTrack application. System health is monitored, and the computer will speak to the operator when error conditions occur. For example, if tracker number three's beam is broken, the system will say, "Tracker three, beam broken. Stop motion and click recover". At this point, the operator takes action, then resumes the process.

B. Starting the Tracker and Initializing the System

The system is started by running TransTrack on the desktop computer. The three MiniTrack applications are then started on three separate notebook machines. Three notebooks are used so that that trackers may be located in distant regions of the workspace. Upon starting, each Mini-Track application performs initialization procedures on its respective tracker, and communicates its network parameters to TransTrack. At start-up, each tracker is identified as a primary, secondary, or tertiary instrument. Essentially, the primary instrument will determine the origin of the moving reference system, the secondary will determine an axis direction of the moving system, and the tertiary will determine final orientation by locking a specified plane of the moving system [7].

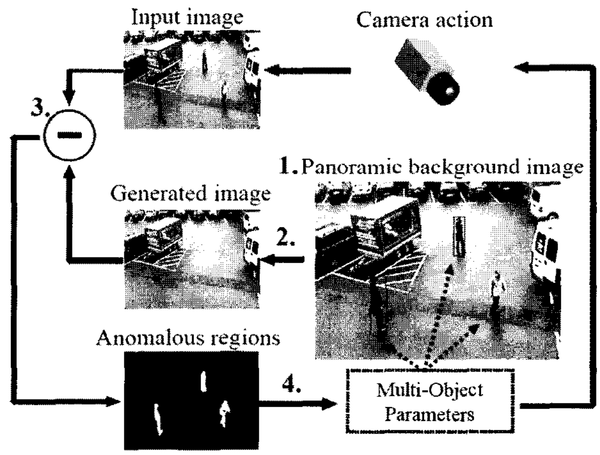


Fig. 2 Object detection and tracking using an PTZ Camera

C. Locating the Trackers

The first step after starting the system is to locate all of the trackers relative to a known reference coordinate system. This known reference system may be derived from CAD design data or nominal coordinate information. Any number of nominal points may be used to locate the trackers. The TransTrack application provides a simple, intuitive check-list interface to aid in this location procedure.

There is a simple tabular interface where each column represents an instrument and each row represents a point. The user simply positions a retro-reflector from one instrument in a particular target nest, then clicks in the corresponding cell in a table. The point is measured and that element is marked as completed. Once a sufficient set of common targets is measured, the user selects the "Locate" option, and all the trackers are located relative to the nominal coordinate data points. If there are any cases of insufficient data, the user is told specifically where additional data is needed. The entire location process should take no more than five minutes once the trackers are warmed up and online[8].

D. Tracking the Objects

With the trackers located in the reference frame, the object transform can now be tracked. First, the SMR's are tracked to their respective tracking points as shown in Figure 3. From the three tracking points a coordinate frame is built and updated as TransTrack receives data from the trackers. The point from Tracker 1 becomes the origin of the tracking frame. Tracker 2 provides a point which lies along the X-axis, and the point from Tracker 3 is set to lie in the X-Y plane. As data is received by TransTrack, the tracking frame, or transform, is updated as shown in Figure 4. TransTrack also maintains a set of diagnostics including the distance between the three tracked points, and timing information pertinent to the acquisition time differences[9].

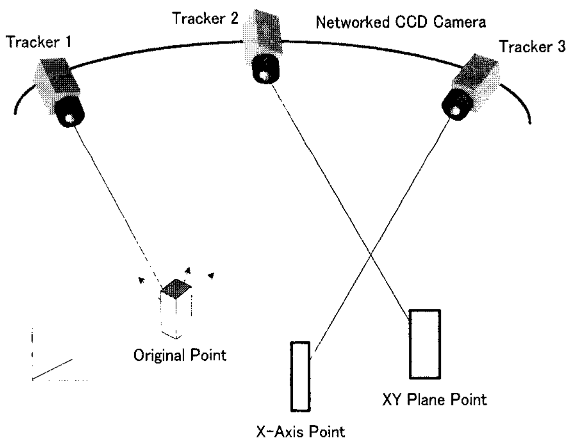


Fig. 3 Tracking the Transform

III. REAL-TIME TRACKING ALGORITHME

A. IMPRESARIO GUI

This chapter describes Impresario's application developing interface (API) which can be used to extend Impresario's functionality by developing new macros[10]. In order to be able to understand this guide and successfully develop own macros, general knowledge is needed about:

- Concepts and usage of Impresario,
- Concepts and usage of the LTI-Lib,
- Object oriented programming in C++ including class inheritance, data encapsulation, polymorphism, and template usage,
- DLL (Dynamic Link Library) concepts and programming on Windows platforms.

To build macro projects a compiler which supports the ANSI C++ standard is required. For convenience this development kit contains project files for Microsoft's Visual Studio .NET 2003. Impresario and the delivered macro DLLs were developed with this environment. Different compilers haven't been tested yet but it should be possible to produce executable code as well. The development kit also contains two Perl scripts which help to create new macro projects and new macro classes. To be of use a Perl interpreter has to be installed on the system.

B. Directory Structure

By default the development kit is installed in the directory macrodev as a subdirectory of the Impresario software. It contains the following subdirectories:

Doc: This directory contains the documentation your are currently reading.

Libs: This directory stores third party libraries which may be used during development of new macro projects. By default, it contains a compiled version of the LTI-Lib which is necessary at least for image input and output.

Projects: Main directory for macro projects. It contains a workspace file for Visual Studio .NET 2003 named *macrodevelopment.sln* and two projects whereas the *Sample* project serves as template for new projects. The *Macros* project contains the source code for most macros delivered with the base version of Impresario.

Tools: Contains two Perl scripts *createProject.pl* and *createMacro.pl* to create a new macro project and a new macro class respectively. The subdirectories within this folder contain template files used by the scripts.

C. Creating a New Macro

In Impresario every macro is described by its input ports, output ports, and parameters. The visual appearance of a macro in the GUI is depicted in the following figure. The input ports are colored yellow, the output ports are colored red, and the list of parameters is available in a separate window. Internally a macro is represented by a C++ class which is derived from the class *CMacroTemplate. CMacroTemplate* defines the common interface to Impresario. Therefore the two files *macrotemplate.h* and *macrotemplate.cpp* have to be included in every macro project[10].

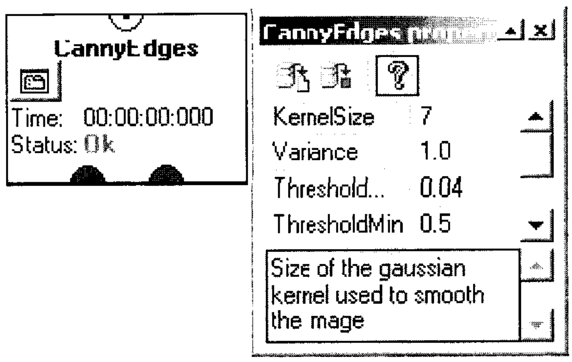


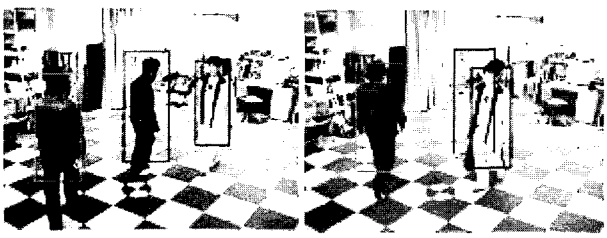
Fig. 4 Appearance of a standard macro property window

IV. DETECTION AND TRACKING

In this paper, a walking people tracking system of the type tracking following detection is built and tested using IMPRESARIO and LTI-LIB which is an open source software library that contains a large collection of algorithms from the filed of computer vision. As explained in previous section, systems of this type are more appropriate in scenes where inter-person occlusion is rare, e.g. the surveillance of large outdoor areas from a high camera perspective, than in narrow or crowded indoor scenes. This is due to the necessity of the tracked persons to be separated from each other most of the time to ensure stable tracking results.

The general structure of the system as it appears in IMPRESARIO is shown in Fig. 4. Image source can either be a live stream from a webcam or a prerecorded image sequence.

The first processing step is the segmentation of moving foreground objects by background subtraction (see Fig. 7). The background model is calculated from the first *initialFrameNum* frames of the image sequence, so it has to be ensured that these frames show only the empty background scene without any persons or other moving objects. Alternatively, a pre-building background model (using the IMPRESARIO-macro **trainBackgroundModel**) can be loaded that must have been created under exactly the same lighting conditions. The automatic adjustment of the camera parameters has to be deactivated to prevent background colors from changing when people with dark or bright clothes enter the scene.



(a)separated view (b) overlapped view

Fig. 5 Example output of tracking system

After reducing the noise of the resulting foreground mask with a sequence of morphological operations, person candidates are detected image regions are passed on to the **peopleTracking** macro, where they are used to update the internal list of tracked objects. A tracking logic handles the appearing and disappearing of people in the camera field of view as well as the merging and splitting of regions as people occlude each other temporarily in the image plane. Each person is described by a one or two-dimensional Temporal Texture Template for re-identification after an occlusion or, optionally, after re-entering the scene. The tracking result is displayed using the **drawTrackingResults** macro. The appearance models of individual persons can be visualized with the macro **extractTracking Templates**[10].

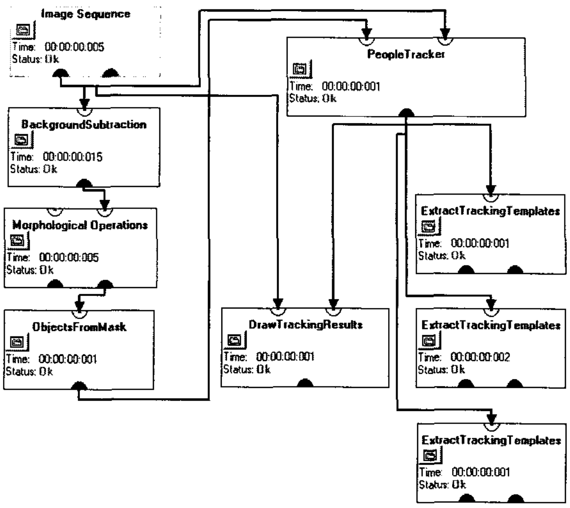


Fig. 6 IMPRESARIO for walking human tracking

V. TRACKING RESULTS

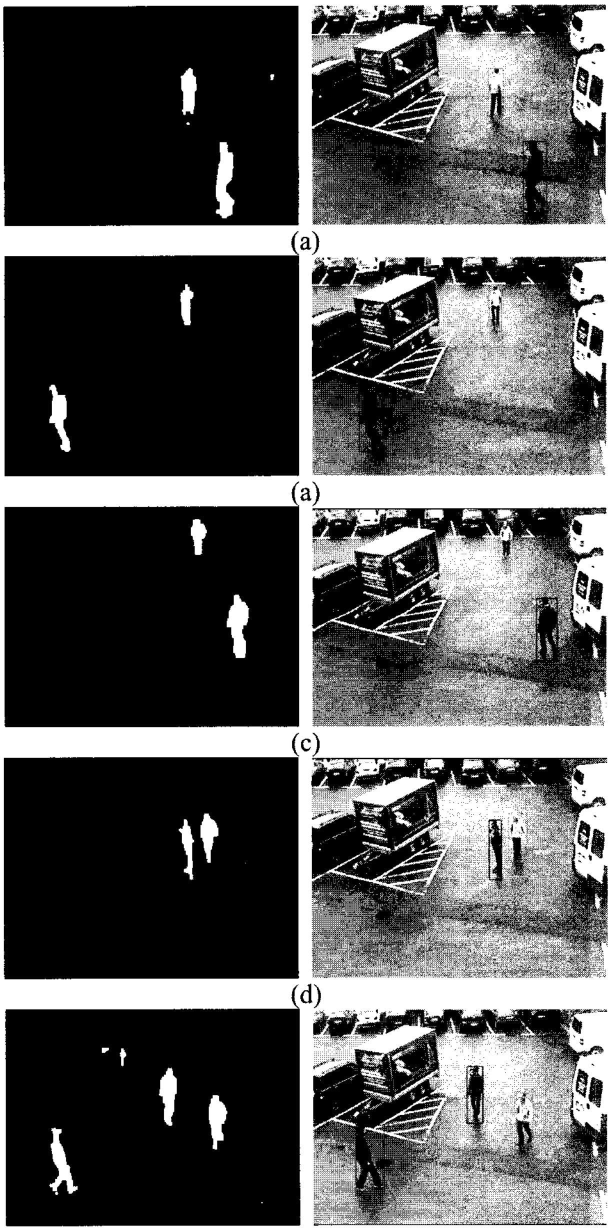
At each camera a background subtraction module extracts foreground regions from the live camera images. The centroids of foreground regions are later used by the tracking agent in connection with camera models to discover the 3-D position of objects of interest.

The key idea in the implemented segmentation algorithm is to subtract the still background from the current image yielding people or recently moved objects. The critical part of this approach is to adapt the background estimation over time in such a way that gradual lighting changes or moved objects do not result in a seriously erroneous background estimate.

Most of the background subtraction algorithms estimate the background color of each pixel in the image continously. Our tracking system uses the background estimation process developed in [9]. The process uses a mixture of Gaussians to estimate the background color per pixel. This provides a more robust foreground region extraction compared to single Gaussian approaches.

We show the result of tracking in the following figures. Background models that estimate the background color per pixel tend to detect shadows as foreground (false positives), if the underlying color space has an intensity component as in Fig. 7. To counter this problem the background color can be estimated only on a chromatic color space. But this does not always solve the problem, since a number of foreground objects might not be detected (false negatives) or an object dissolves into several regions as in Fig 7(5). The implemented segmentation algorithm uses both chromatic and intensity information to ensure a low number of false positives and negatives. The process of segmentation is illustrated in Figure 7:

A subject walked, bended and stretched in an observed area with changing orientations of his/her body. Fig. 7 shows the example images of the tracking result. In each image, the tracking result is drawn by the colored rectangle with dots corresponding to samples. We first run a background subtraction algorithm on each of the camera views, and then, apply an image segmentation algorithm to the foreground regions. The segmentation algorithm differentiates between different objects even though they might occur in the same connected component as found by the background subtraction algorithm, but, of course oversegments the component into many pieces. We next match regions along epipolar lines in pairs of cameras views. The mid-points of the matched segments along the epipolar lines of each stereo pair are back-projected to yield 3D points, which are then projected onto the ground plane.



(a) (a) (c) (d) (e)

Fig. 7 Tracked walking human and extract tracking templates

These ground points are then used to form an object location probability distribution map using Gaussian kernels for a single image pair. The probability distribution maps are the combined using outlier-rejection techniques to yield a robust estimate of the 2D position of the objects, which is then used to track them. From these foreground regions the RG color histogram, the bounding box, the centroid, and the size are computed and broadcasted appropriately packaged and time stamped.

Respect to the recognition process, it was observed that, having the object correctly located and tracked, people was positively recognized in almost all the cases. To indicate that a person has been positively recognized, a bounding box is drawn around its centroid. The whole human analyzed were positively detected the most of the cases in frontal and back views (see Fig 7). Although the geometrical structure changes in an appreciable way for lateral views, the overall recognition process provides the correct result in the majority of the cases. The negative recognition behavior was tested also with positive results.

VI. ACKNOWLEDGMENTS

This work is resulted from the project of Human Resources Development for Industrial Demand financially supported by the Ministry of Education and Human Resources Development(MOE).

VII. CONCLUSIONS

In this paper, we proposed a method for 3D walking human tracking based on the IMPRESARIO framework incorporating cascaded classifiers into hypothesis evaluation. The efficiency of adaptive selection of cascaded classifiers has been also presented. We have shown the improvement of reliability for likelihood calculation by using cascaded classifiers.

This realizes robust and accurate human head tracking. We confirmed the effectiveness of our method by experiments on tracking of a human head in an outdoor environment.

RT-IMPRESARIO is a system for tracking objects in real-time video streams (video conferences) and allowing hyperlink anchors to be associated with these tracked objects. We have described here the further use of IMPRESARIO for applying automated object tracking to stored video streams, thereby allowing automated markup of archived video data with hyperlinks.

Extending IMPRESARIO from real-time to archived video requires a link layer to capture and maintain the link anchors as they are tracked from frame to frame; in the real-time mode, this information is available at each instant, but lost as each frame progresses to the next. We demonstrated such a link layer on top of the basic IMPRESARIO tracker using COTS software. namely, the *Wired Sprites* of Apple.s *QuickTime* standard.

In the future, we extend our work to the multiple people tracking. This framework of multiple camera tracking allows us to track multiple objects without establishing correspondences among objects observed in each camera. Since the output of our likelihood function does not represent a probability of likelihood theoretically, we plan to adopt the likelihood calibration technique described in [2] to improve the performance of tracking with cascaded classifiers. The initial detection of human heads and employing additional classifiers to improve estimation accuracy of head orientation are also left for future works.

REFERENCES

[1] Ren C. Ruo and Kuo L. Su, "A Review of High-level Multisensor Fusion: Approaches and applications," Proc. Of IEEE Int'l. Conf. On Multisensor Fusion and Integration for Intelligent Systems, pp. 25-31, Taipei, Taiwan, 1999.

[2] Jang M. Lee, B. H. Kim, M. H. Lee, M. C. Lee, J. W. Choi, and S. H. Han, "Fine Active Calibration of Camera Position/Orientation through Pattern Recognition," Proc. of IEEE Int'l. Symp. on Industrial Electronics, pp. 100-105, Slovenia, 1999.

[3] Hong, L., Lynch, A., "Recursive temporal-spatial information fusion with applications to target identification," Aerospace and Electronic Systems, IEEE Transactions on, Vo. 29 Issue. 2, p. 435-445. 1993.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. R. Statist. Soc., vol. 39, pp. 1–38, 1977.

[5] TaeSeok Jin, Hideki Hashimoto, "Multi-Object Tracking using the Color-Based Particle Filter in ISpace with Distributed Sensor Network" International Journal of Fuzzy Logic and Intelligent Systems, Vol. 5, No. 1, pp. 46-51, March 2005.

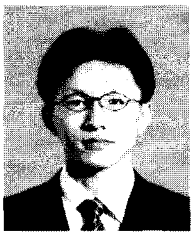
[6] TaeSeok Jin, Hideki Hashimoto, "Human Tracking using Multiple-Camera-Based Global Color Model in Intelligent Space" International Journal of Fuzzy Logic and Intelligent Systems, Vol. 6, No. 1, pp. 39-46, March 25, 2006.

[7] Akihiro Sugimoto, Kiyotake Yachi, and Takashi Matsuyama. "Tracking human heads based on interaction between hypotheses with certainty". In Proc. of Scandinavian Conference on Image Analysis, pages 617–624, 2003.

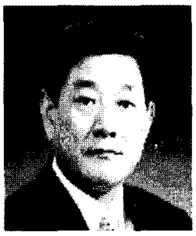
[8] Paul Viola and Michael Jones. « Rapid object detection using a boosted cascade of simple features". In Proc. of International Conference on Computer Vision and Pattern Recognition, pages I:511–518, 2001.

[9] Ya-DongWang, Jian-KangWu, and Ashraf A. Kassim. "Particle filter for visual tracking using multiple cameras". In Proc. of IAPR Conference on Machine Vision Applications, pages 298-301, 2005.

[10] Karl-Friedrich Kraiss. "Advanced Man-machine Interaction-fundamental and Implementation. Springer, 2007.



Tae-Seok Jin
He received the Ph.D. degrees from Pusan National University, Busan, Korea, in 2003, in electronics engineering. He is currently an assistant professor at DongSeo University. From 2004 to 2006, he was a Postdoctoral Researcher at the Institute of Industrial Science, The University of Tokyo, Japan. His research interests include network sensors fusion, mobile robots, computer vision, and intelligent control. Dr. Jin is a Member of the KFIS, IEEK, ICASE, and JSME.



Hyun-Deok Kim
received the Ph.D. degrees from DongA University, Busan, Korea, in 1996, in electrical engineering.
Since 1989 He has been a professor at Jinju National University. Now, He is a member of KIMICS. His research interests include digital signal processing, neural network.