Paper ID
DRC1004

The 3rd TSME International Conference on Mechanical Engineering
October 2012, Chiang Rai

Multiple Objects Tracking using a Kalman Filter on a PC-Cluster based Multi-Camera System

Sangveraphunsiri, V. and Bamrungthai, P.*

Department of Mechanical Engineering, Faculty of Engineering, Chulalongkorn University
254 Phayathai Road, Pathumwan, Bangkok 10330, Thailand
Tel. +66 2218 6585, Fax. +66 2218 6583

Abstract

A multi-camera multiple objects tracking system has been presented in this paper. It has been designed as a general purpose application development platform for real-time 3-D tracking. 3-D reconstruction algorithm and Kalman filter-based tracking algorithm have been developed. The tracking algorithm consists of data association and motion prediction mechanism to identify and track each object independently. The targets are spherical objects with light inside. The Kalman filter with data association framework improves robustness of the tracking process even when object occlusion occurred. The algorithm is implemented as a software module to operate on the developed PC-cluster. The PC-cluster is a two-computer system that two computers are connected by using PCI-to-PCI interface card with fiber-optic connection. Four cameras with hardwired synchronization have been used for experiments. Experimental results show that the system can track three objects correctly. Average system frequency is 14 frames per second. The system can speed up by fine-tuning of the developed software.

Keywords*:* Multiple objects tracking, Kalman filter, Multi-camera system, PC-cluster

1. Introduction

Multiple objects tracking is a fundamental task for many applications including human-computer interaction [1], study of flying animal behavior [2], and biomechanical analysis [3]. For capturing object motion in 3-D space, multi-camera systems are widely used because of their high precision and flexibility with non-intrusiveness to the target objects. Nevertheless, occlusion and multi-view correspondence problem make the task very complex.

In this paper, the development of a multi-camera multiple objects tracking system has been presented. A multiple objects tracking algorithm using Kalman filter has been described. 3-D reconstruction algorithm is developed for solving multi-view correspondence problem. The algorithms are implemented on the developed PC-cluster system [4, 5] as a processing module. The multi-camera system on the PC-cluster has been designed as a general purpose application development platform as shown in Fig. 1. Output of the system is 3-D position of each target.

2. 3-D Reconstruction

3-D reconstruction is required to compute 3-D positions of the targets from the 2-D projected object's location in the images. In the tracking process, the 3-D positions are used for data association. After image capturing process, the following steps of the 3-D reconstruction are performed.

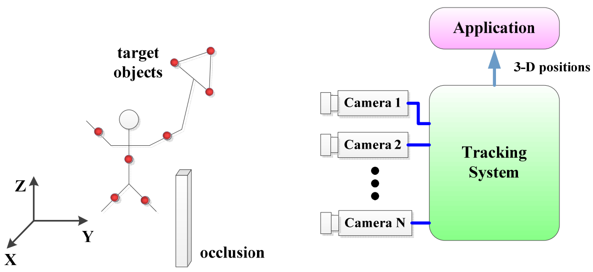

Fig. 1 System overview

2.1 Object detection

Object detection algorithm identifies the 2-D coordinates of object feature (connected component or blob [6]) in the image. Connected-component analysis with linear-time searching [7] has been used in this paper. The target objects are spherical objects with light inside. Image of the target objects from a cameras in the system is shown in Fig. 2. Object detection results from four cameras have been shown in Fig. 3.

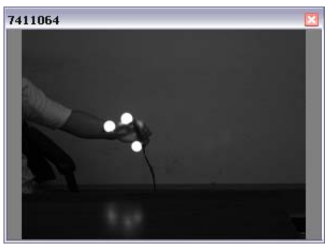

Fig. 2 Image of the target objects captured from a camera in the system

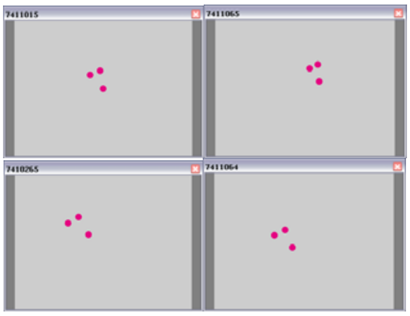

Fig. 3 Object detection results from four cameras

2.2 Multi-view correspondence

The system solves for the correspondence problem across all views by using 2-D positions obtained in the previous section. Then, 3-D position of each target is calculated from the correspondence points. As an example shown in Fig. 4, object 1 is reconstructed by using blob number (2) in image 1 and blob number (3) in image 2. The same concept is also applied for multi-camera case.

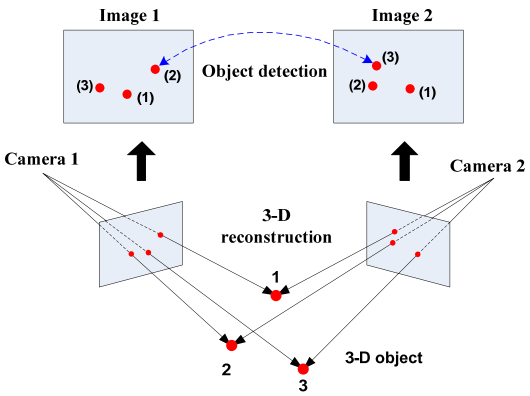

Fig. 4 Multi-view correspondence problem

Given the points p_m^i is a 2-D object position m in camera i. The algorithm for multi-view correspondence that has been developed is summarized as follows:

1) Find an unused point p_m^i.

2) Calculate the minimum distance between the ray created from the center of projection of camera i to the object point p_m^i and the ray created from the center of projection of another camera j to an unused point p_n^j.

3) If there is a minimum distance in 2) less than a specified threshold, repeat step 2) by using p_m^i and the unused point p_r^k in another camera k as object points to create the rays.

4) If there is a minimum distance in 3) less than a specified threshold, identifies point p_m^i, p_n^j, and p_r^k as correspondence points, and mark the points as used.

5) Repeat step 1) to 4) until there is no more unused point or the number of the reconstructed points reaches the predefined maximum.

2.3 Reconstruction

The 3-D position calculation from multiple 2-D correspondences in camera images is a well-studied problem in computer vision community. In this paper, linear triangulation method [8] is applied. As specified in section 2.2, the 3-D reconstruction algorithm in this paper is performed by using the correspondence points from three cameras.

Consider a multi-camera system with n cameras, given $\mathbf{P}_i$ is the i-th camera matrix, and $\mathbf{x}_i = [u_i \quad v_i \quad 1]^T$ is a point in an image of the 3-D world point $\mathbf{X}_w$ and corresponds to camera matrix $\mathbf{P}_i$. Then, we have $\mathbf{x}_i = \mathbf{P}_i\mathbf{X}_w$. By using the relation of vector cross product,

$$\mathbf{x}_i \times (\mathbf{P}_i\mathbf{X}_w) = 0 \tag{1}$$

then,

$$\left.\begin{aligned} u_i\left(\mathbf{p}_i^{3T}\mathbf{X}_w\right) - \left(\mathbf{p}_i^{1T}\mathbf{X}_w\right) &= 0 \\ v_i\left(\mathbf{p}_i^{3T}\mathbf{X}_w\right) - \left(\mathbf{p}_i^{2T}\mathbf{X}_w\right) &= 0 \\ u_i\left(\mathbf{p}_i^{2T}\mathbf{X}_w\right) - v_i\left(\mathbf{p}_i^{1T}\mathbf{X}_w\right) &= 0 \end{aligned}\right\} \tag{2}$$

where $\mathbf{p}_i^{jT}$ is the j-th row of camera matrix $\mathbf{P}_i$

From each camera, only two equations of (2) are independent. When the first two equations has been chosen, a matrix $\mathbf{L}$ can be obtained by stacking up equation (2) for n cameras $(n \geq 2)$ as

$$\mathbf{L}\mathbf{X}_w = 0 \tag{3}$$

where

$$\mathbf{L} = \begin{bmatrix} u_1\mathbf{p}_1^{3T} - \mathbf{p}_1^{1T} \\ v_1\mathbf{p}_1^{3T} - \mathbf{p}_1^{2T} \\ u_2\mathbf{p}_2^{3T} - \mathbf{p}_2^{1T} \\ v_2\mathbf{p}_2^{3T} - \mathbf{p}_2^{2T} \\ \vdots \\ u_n\mathbf{p}_n^{3T} - \mathbf{p}_n^{1T} \\ v_n\mathbf{p}_n^{3T} - \mathbf{p}_n^{2T} \end{bmatrix} \tag{4}$$

Equation (3) can be solved by the least-squares method. The solutions are unit singular vector which is correspond to the smallest singular value of matrix $\mathbf{L}$.

3. Multiple Objects Tracking

3.1 The discrete Kalman filter

Because of its recursive solution to estimate state of a linear discrete time random process in an optimal manner, Kalman filter [9, 10] is applied for object state estimation in the tracking process in this paper.

State equation describes how the current state evolved from the previous state,

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k \tag{5}$$

where $\mathbf{A}_k$ is the process model that relates the state $\mathbf{x}_k$ from time k to $k+1$ in the absence of an external disturbance and $\mathbf{w}_k$ is the process noise vector.

Measurement equation describes how the measurement is derived from the current state,

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{6}$$

where $\mathbf{H}_k$ is the measurement model describing the noiseless relationship between the measurement $\mathbf{z}_k$ and the state vector $\mathbf{x}_k$ at time k and $\mathbf{v}_k$ is the measurement noise vector.

The set of equations for the prediction and adjustment used in the tracking loop are list below.

Prediction equations:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_{k-1} \hat{\mathbf{x}}_{k-1}^- \tag{7}$$

$$\mathbf{P}_k^- = \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \tag{8}$$

$$\hat{\mathbf{z}}_k = \mathbf{H}_k \hat{\mathbf{x}}_k^- \tag{9}$$

Adjustment equations:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{10}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \tag{11}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \tag{12}$$

In this paper, the target objects are treated as particles and the state vector

$$\mathbf{x} = [x,\ y,\ z,\ \dot{x},\ \dot{y},\ \dot{z}]^T \tag{13}$$

is used for describing 3-D position and velocity of the targets. The measurement $\mathbf{z}$ is the same as the state vector because object is tracked in 3-D space directly.

The process model with constant velocity motion assumption has been used in this paper,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

with Δt being the time step.

3.2 Kalman filter based multiple objects tracking algorithm

Flow of the tracking process of the system can be shown in Fig. 5. Starting with the captured image, an object detection algorithm identifies the 2-D coordinates of every blob's centroid in the image. For every set of synchronized frames, the system solves the correspondence problem across all views. Then, 3-D reconstruction is performed for every set of correspondence points. 3-D tracking of each object has been done independently with object's state prediction and data association and update algorithm. Finally, the output of the system is 3-D position of each target.

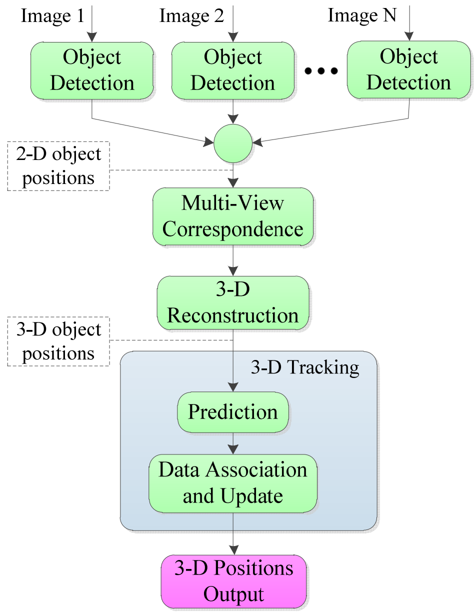

Fig. 5 Flow of the tracking process

Data association step in the tracking process is the nearest neighbor searching between the predicted object positions (dotted circles) and the current 3-D reconstruction points (filled circles) as shown in Fig. 6.

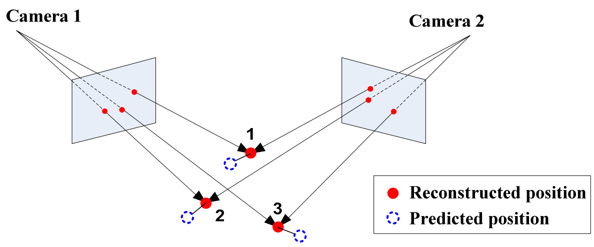

Fig. 6 Data association in the tracking process

Given the 3-D positions of each object is obtained from the 3-D reconstruction process, the multiple objects tracking algorithm developed in this paper is summarized as follows:

1) Set states to initial value for the first time.
2) Predict states and measurements by using equations (7) – (9).
3) Calculate 3-D distance between the current 3-D reconstruction positions and the predicted positions obtained in 2).

4) For each reconstruction points, if the minimum result obtained from step 3) is less than a specified threshold, associate the reconstructed point with the predicted point that makes the minimum distance.
5) Update states with the associated measurements obtained in 4) by using equations (10) – (12).
6) After step 4) and 5), if there are some of tracked points that is not associated with 3-D reconstructed point, check for the number of occlusions.
   a. If it does not exceed the maximum missing steps, set it as temporary occluded point.
   b. If it has exceeded the maximum missing steps, delete it from the tracking point set.
7) If there is some reconstructed points left after step 6), set it as new tracked point.

4. The PC-Cluster

The multi-camera system in this paper operates on the developed PC-cluster that consists of two computers. By using PCI-to-PCI interface card with fiber-optic cable for connection, the two computers work as a single processing system. Schematic of the system with four cameras is shown in Fig. 7.

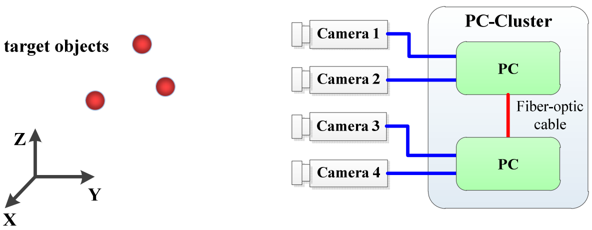

Fig. 7 Schematic of the developed multi-camera system based on a PC-cluster

One computer is main or local computer for overall system operations control such as camera settings, parameters settings for calibration and tracking, getting command from user, and displaying graphical output on monitor. The other is remote computer used for image capture and object detection in images obtained from the attached cameras. Each computer runs the developed application software with processing modules as shown in Fig. 8.

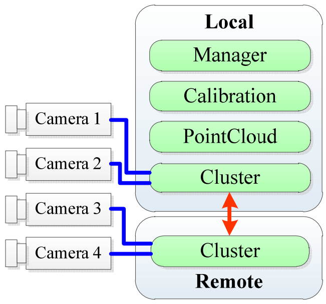

Fig. 8 Processing modules in the PC-cluster

There are four processing modules in this work.
1) Cluster module is used for image capture, cameras control, object detection, and PC-to-PC communication.
2) PointCloud module is used for 3-D reconstruction and tracking of multiple objects. It is an implementation of the algorithms that have been described in section 2 and 3.
3) Calibration module is used for camera calibration that is a separate offline step before starting 3-D reconstruction and tracking process.
4) Manager module is used for camera synchronization, export output file, and system event management.

All modules are implemented in C++ as dynamic-link libraries (DLLs) on Windows XP operating system.

5. Experimental Results

In the experiments, four sets of IEEE1394a monochrome cameras with 8mm lenses have been used. All cameras are calibrated by using Tsai's camera calibration method [11]. Hard-wired synchronization is used for multi-camera capture which one camera has been chosen as primary camera. The images are captured with the size of 640$\times$480 pixels. The main computer of the PC-cluster is implemented on a 2.83 GHz Intel Core 2 Quad computer. And the remote computer runs on 3.4 GHz Pentium 4 processor.

5.1 Robot tracking

In the first experiment, Mitsubishi PA10-7C robot arm is used for moving the target objects. The experimental setup is shown in Fig. 9. The targets are three spherical objects with 20 mm diameter. They are arranged in an L-shape as shown in Fig. 10.

Fig. 9 Mitsubishi PA10-7C robot arm holding the target objects

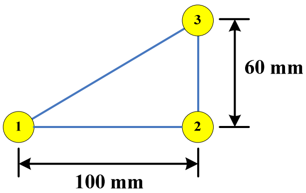

Fig. 10 Three objects arrangement and its dimensions

The robot moves in planar circular motion in XY plane of the robot coordinate system (Fig 11). It moves in a circle of radius 100 mm with angular velocity 0.05 rad/sec. The 3-D positions of each target obtained from the multi-camera system are plotted in Fig. 12 – 13. The result shows that the developed system can identify and track each object correctly.

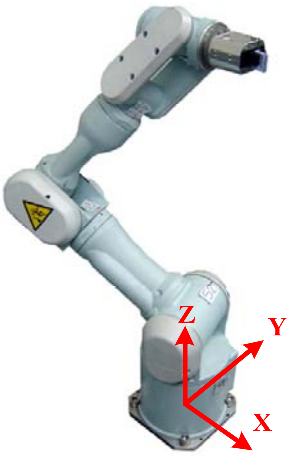

Fig. 11 Robot coordinate system

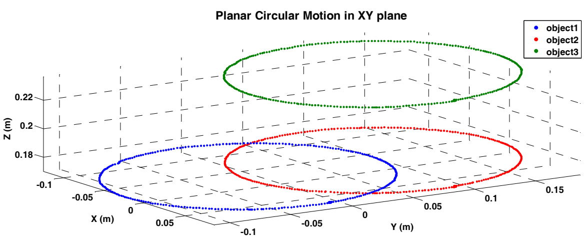

Fig. 12 Planar circular motion in XY plane (3-D view)

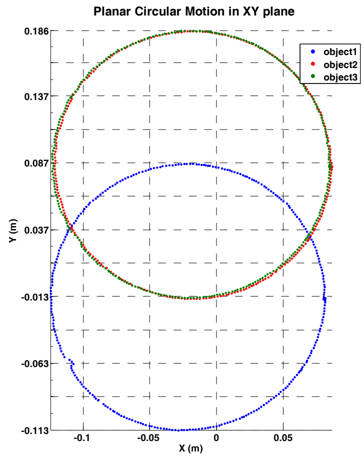

Fig. 13 Planar circular motion in XY plane (Top view)

5.2 Surface scan

The system can be applied for reverse engineering application if the target objects move on the surface to be measured. In the second experiment, three target objects (Fig. 14) have been moved independently on the helmet prototype surface, shown in Fig. 15, to capture its shape into computer system.

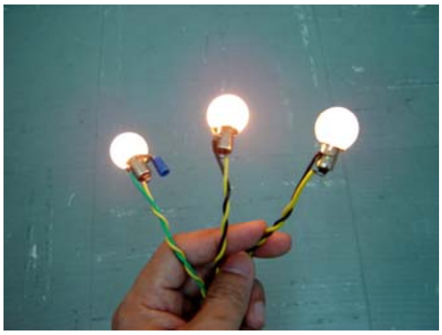

Fig. 14 Target objects used for surface scan

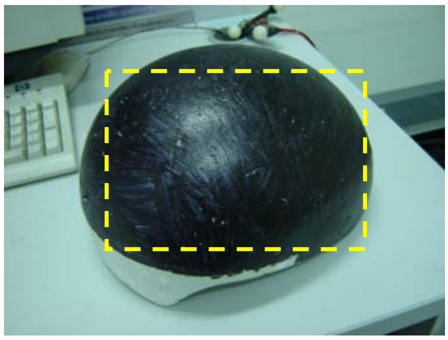

Fig. 15 Helmet prototype and the measurement region

The result is point cloud of the helmet as shown in Fig. 16 - 17. It shows the correctness of the tracked position that can capture shape of the helmet.

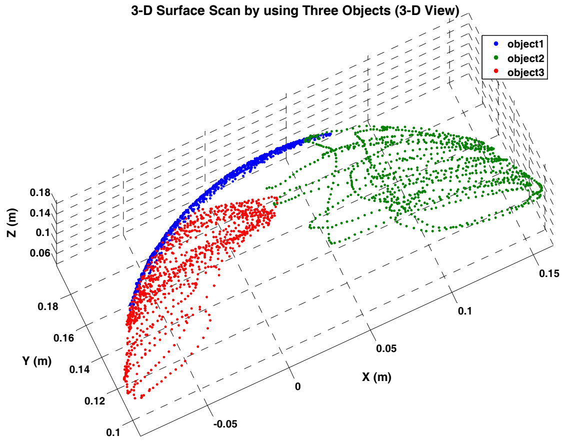

Fig. 16 Point cloud of the helmet prototype (3-D view)

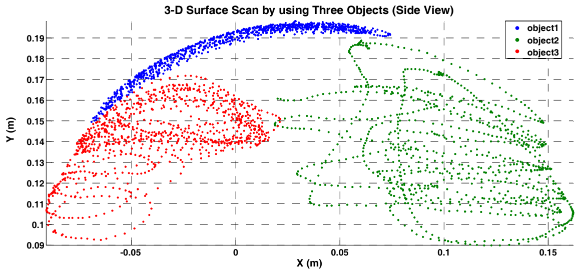

Fig. 17 Point cloud of the helmet prototype (Side view)

6. Conclusions

This paper presents the development of a multi-camera system for 3-D multiple objects tracking. The 3-D reconstruction algorithm and the Kalman filter-based tracking algorithm are developed. They are implemented on the PC-cluster (two-computer system) as a processing module (PointCloud Module). The system has been tested by using three objects. The experimental results show that the system can track each object correctly. The system can be applied for many applications if application specific module has been added to the system.

7. Acknowledgement

This work has been supported by Chulalongkorn University under Chulalongkorn University Centenary Academic Development Project.

8. References

[1] Dorfmüller-Ulhaas, K. (2003). Robust optical user motion tracking using a Kalman filter, *Technical Report 2003-6*, University of Augsburg.

[2] Straw, A.D., Branson, K., Neumann, T.R., and Dickinson, M.H. (2011). Multi-camera real-time three-dimensional tracking of multiple flying animals, *J. R. Soc. Interface*, 8: 395–409.

[3] Cerveri, P., Rabuffetti, M., Pedotti, A., Ferrigno G. (2003). Real-time human motion estimation using biomechanical models and non-linear state-space filters, *Med. Biol. Eng. Comput*, 41: 109 – 123.

[4] Uttamang, K., and Sangveraphunsiri, V. (2007). Multi-camera system using PC-cluster for real-time 3-D pose estimation, *Thammasat Int. J. Sc. Tech*, 12 (April-June 2007): 52-63.

[5] Bamrungthai, P., and Sangveraphunsiri, V. (2010). A PC-Cluster based Multi-Camera System for Real-Time Multiple Objects Tracking, *Asian International Journal of Science and Technology in Production and Manufacturing Engineering (AIJSTPME)*, 3(4): 57-64.

[6] Jain, R., Kasturi, R., and Schunck, B.G. (1995). *Machine Vision*, Singapore: McGraw-Hill.

[7] Chang, F., Chen, C.J., Lu, C.J. (2004). A Linear-Time Component-Labeling Algorithm using Contour Tracing Technique, *Comput. Vis. Image Underst*, 93(2): 206-220.

[8] Hartley, R., and Zisserman, A. (2003). *Multiple view geometry in computer vision*, 2nd ed, Cambridge, UK: Cambridge University Press.

[9] Kalman, R.E. (1960). A new Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME – Journal of Basic Engineering*, 82: 35-45.

[10] Zarchan, P., and Musoff, H. (2009). *Fundamentals of Kalman Filtering: A Practical Approach*, AIAA, Reston.

[11] Tsai, R. (1987). A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology using Off-The-Shelf TV Cameras and Lenses, *IEEE Journal of Robotics and Automation*, 3(4): 323-344.