2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Object Tracking with Measurements from Single or Multiple Cameras

Magnus Linderoth, Anders Robertsson, Karl Åström, and Rolf Johansson

Abstract— **To be able to determine the position of a static object in 3D space by means of computer vision, it has to be seen by cameras from at least two different view points. The same applies for measuring the position of a moving object based on images captured at one single time instant. However, if the cameras are not synchronized in time, or if a moving object is not visible in all images, one can not rely on using matching pictures for making accurate position estimates of dynamical objects.**

This paper presents a strategy to track an object with known dynamical model, using a series of images where no pair has to be captured simultaneously. It even allows tracking of a point object in 3D space using a single static camera.

I. Introduction

DYNAMICAL vision is important as a means to accomplish feedback control based on robotic work-space sensing and sensor fusion (e.g. force-vision sensor fusion). To the purpose of model-based control, the Kalman filter [4] constitutes an important component capable of handling model-based prediction and predictive control. For optimal prediction and noise suppression, the Kalman filter requires temporal updates as well as measurement updates. Provided that vision data arrive in real time with appropriate features extracted, an abundant source of error measurement is offered. As vision data deriving from moving objects may change depending on the camera positions of multiple cameras, the system should be tolerant to missing frames. Previous work on how vision and Kalman filtering can be used in robot control is presented in [5-10]. In [6] a Kalman filter was used to combine joint angles with vision to estimate the tool position of a compliant robot in contact with the environment. In [3] dynamic objects are tracked from a mobile platform using an extended Kalman filter and multiple interacting models.

A method for tracking rigid objects that were partly occluded by other objects was suggested in [11]. It does, however, require that a graphical model of the object to be tracked is available and can be projected onto the image.

One application presented in this paper is in the implementation of a ball-catching robot (Fig. 1). The system has a box with a hole mounted on a robot and cameras to see when a ball is approaching the box. The goal is to move the box so the ball always hits the hole. To do this, the ball is tracked to enable prediction of where the hole of the box should be. Even if the ball is not visible in some images or the image analysis algorithm fails to detect the ball in some images, it is desirable to be able to use the data in other images to improve the estimate of the predicted trajectory of the ball. The problem is closely related to visual servoing. It should be noted, though, that in this case vision is not used to determine the state of the controlled object (the robot). A robotic ball catcher was described in [12]. It, however, used a different tracking algorithm with an Extended Kalman Filter [13].

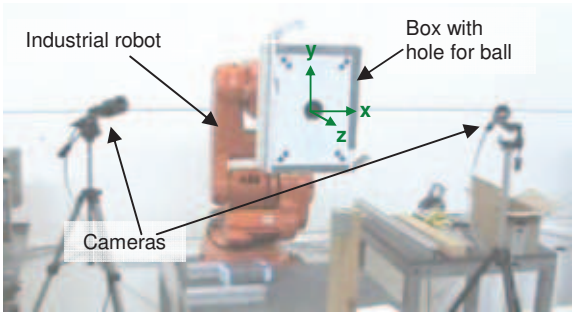

Fig. 1. Overview of a ball-catching robot.

II. Problem formulation

The goal is to track an object with known process model. This is to be done with images captured from different view points and where possibly no images are captured simultaneously. The object to be tracked is described by the discrete time state-space model

$$x(kh+h) = \Phi x(kh) + \Gamma u(kh) + v(kh), \qquad (1)$$

where h is the time step, x is the state vector and u is a known input signal. Measurements are assumed to be on the form

$$y(kh) = C(kh)x(kh) + e(kh). \qquad (2)$$

Note that the C-matrix is time dependent and will depend on the camera parameters and the image coordinates at that

Manuscript received September 13, 2009.

Corresponding author Magnus Linderoth is with the Department of Automatic Control, LTH, Lund University, Sweden (phone: +46462220847; e-mail: magnus.linderoth@control.lth.se).

Anders Robertsson and Rolf Johansson are with the Department of Automatic Control, LTH, Lund University, Sweden.

Karl Åström, is with the Department of Mathematics, LTH, Lund University, Sweden.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 230902. This document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained herein.

particular time step. The disturbances v and e are discrete-time Gaussian white noise processes with zero mean value and

$$\begin{aligned} \mathrm{E}\left[v(kh)\ v^T(kh)\right] &= R_1 \\ \mathrm{E}\left[v(kh)\ e^T(kh)\right] &= R_{12} \\ \mathrm{E}\left[e(kh)\ e^T(kh)\right] &= R_2 \end{aligned} \tag{3}$$

with a Gaussian distribution of the initial state with

$$\mathrm{E}\left[x(0)\right] = m_0 \text{ and } \mathrm{E}\left[x(0)x^T(0)\right] = R_0 \tag{4}$$

III. Methods

A. State estimation

A Kalman filter will be used to estimate the state of the tracked object. Assuming that h is used as time unit, the update law is described by (5) – (10). Here, for example, $\hat{x}(k+1|k)$ denotes the estimate of x at sample k+1 based on measurements up to sample k, and $P(k+1|k)$ the covariance of that estimate.

$$\begin{aligned} \hat{x}(k|k) &= \hat{x}(k|k-1) + \\ &\quad + K_f(k)\left(y(k) - C(k)\hat{x}(k|k-1)\right) \end{aligned} \tag{5}$$

$$\begin{aligned} K_f(k) &= P(k|k-1)C(k)^T \times \\ &\quad \times\left(R_2(k) + C(k)P(k|k-1)C(k)^T\right)^{-1} \end{aligned} \tag{6}$$

$$P(k|k) = P(k|k-1) - K_f(k)C(k)P(k|k-1) \tag{7}$$

$$\begin{aligned} \hat{x}(k+1|k) &= \Phi\hat{x}(k|k-1) + \Gamma u(k) + \\ &\quad + K(k)\left(y(k) - C(k)\hat{x}(k|k-1)\right) \end{aligned} \tag{8}$$

$$\begin{aligned} K(k) &= \left(\Phi P(k|k-1)C^T(k) + R_{12}(k)\right)\times \\ &\quad \times\left(R_2(k) + C(k)P(k|k-1)C^T(k)\right)^{-1} \end{aligned} \tag{9}$$

$$\begin{aligned} P(k+1|k) &= \Phi P(k|k-1)\Phi^T + R_1 - K(k)\times \\ &\quad \times\left(\Phi P(k|k-1)C^T(k) + R_{12}(k)\right)^T \end{aligned} \tag{10}$$

B. Homogeneous coordinates

Homogeneous coordinates [1] will be used extensively in this paper. Thus it is convenient to define "~" according to

$$\begin{aligned} &\mathbf{x_1} \sim \mathbf{x_2} \Leftrightarrow \\ &\mathbf{x_1} = \lambda\mathbf{x_2} \quad \text{for some } \lambda \neq 0 \end{aligned} \tag{11}$$

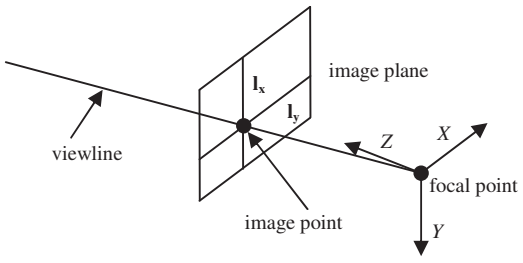

Fig. 2. Illustration of viewline. An object projected onto some image point, can be located anywhere along the corresponding viewline.

where $\mathbf{x_1}$ and $\mathbf{x_2}$ are homogeneous coordinate vectors.

The projection matrix, P_c, of a camera is a 3 by 4 matrix such that

$$\mathbf{x} \sim P_c\mathbf{X} \tag{12}$$

where $\mathbf{X}$ is a point in space and $\mathbf{x}$ is its projection onto the image plane.

IV. Transforming image data For Kalman filter

From the position of a feature point in a single image it is possible to determine a line in 3D-space along which the point must be. In this paper this line will be referred to as the *viewline* (cf. Fig. 2). The viewline parameters then have to be related to the process model states in some way to be possible to use as input to the Kalman filter.

A. Extracting viewlines

To determine the viewline for a point in an image, first two lines through the point in the image are chosen. The easiest way to do this is to take one horizontal and one vertical line. Each of these lines uniquely defines a plane through the focal point and the projection of this line onto the image plane. Finally the viewline can be determined as the intersection of these planes. A line $\mathbf{l}$ in an image can be represented as the points fulfilling the equation

$$\mathbf{l^T x} = 0, \tag{13}$$

where $\mathbf{l} = (l_1 \quad l_2 \quad l_3)^T$ and $\mathbf{x} = (x \quad y \quad 1)^T$.

A plane $\boldsymbol{\pi}$ in space can similarly be represented by the points fulfilling

$$\boldsymbol{\pi}^\mathbf{T}\mathbf{X} = 0, \tag{14}$$

where $\boldsymbol{\pi} = (\pi_1 \quad \pi_2 \quad \pi_3 \quad \pi_4)^T$ and $\mathbf{X} = (X \quad Y \quad Z \quad 1)^T$.

Moreover, if the plane $\boldsymbol{\pi}$ is projected onto the line $\mathbf{l}$, then

$$\mathbf{x} \sim P_c\mathbf{X}, \tag{15}$$

where P_c is the projection matrix of the camera. Inserting (15) into the equation for the image line (13) we get

$$0 = \mathbf{l^T x} \sim \mathbf{l^T} P_c\mathbf{X} = (P_c^T\mathbf{l})^T\mathbf{X}. \tag{16}$$

Hence it can be can concluded that the points that project onto the line $\mathbf{l}$ reside in the plane defined by

$$\boldsymbol{\pi} = P_c^T\mathbf{l}. \tag{17}$$

If the coordinate of a feature point in an image is (x, y), a convenient choice of lines through this point is $\mathbf{l_x} = (-1 \quad 0 \quad x)^T$ and $\mathbf{l_y} = (0 \quad -1 \quad y)^T$ (cf. Fig. 2) corresponding to the planes defined by

$$\begin{aligned} \boldsymbol{\pi}_\mathbf{x} &= P_c^T\mathbf{l_x} \\ \boldsymbol{\pi}_\mathbf{y} &= P_c^T\mathbf{l_y} \end{aligned} \tag{18}$$

The viewline can then be described as the intersection of the planes $\boldsymbol{\pi}_\mathbf{x}$ and $\boldsymbol{\pi}_y$.

B. Flying ball example

To illustrate how the viewlines are transformed into Kalman

filter input data, a simple example of a dynamical model is used; a ball flying under the influence of gravity and without air friction. The state vector is

$$x = \begin{pmatrix} x_d & y_d & z_d & \dot{x}_d & \dot{y}_d & \dot{z}_d \end{pmatrix}^T, \quad (19)$$

where the first three states represent the position and the last three states represent the velocity in the coordinate system of Fig.1. The process model matrices of (1) are

$$\Phi = \begin{pmatrix} 1 & 0 & 0 & h & 0 & 0 \\ 0 & 1 & 0 & 0 & h & 0 \\ 0 & 0 & 1 & 0 & 0 & h \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \Gamma = \begin{pmatrix} 0 \\ -gh^2/2 \\ 0 \\ 0 \\ -gh \\ 0 \end{pmatrix} \quad (20)$$

with $u(kh) = 1$ for all k

C. Including constraints into the Kalman filter

The planes in (18) each put the constraint

$$\boldsymbol{\pi}^{\mathbf{T}}\mathbf{X} = 0 \quad (21)$$

on the feature point position **X**. This can be rewritten as a constraint on the state vector (19) of the process:

$$0 = \boldsymbol{\pi}^{\mathbf{T}}\mathbf{X} = \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 \end{pmatrix}\begin{pmatrix} x_d & y_d & z_d & 1 \end{pmatrix}^T \Leftrightarrow$$
$$\begin{pmatrix} \pi_1 & \pi_2 & \pi_3 \end{pmatrix}\begin{pmatrix} x_d & y_d & z_d \end{pmatrix}^T = -\pi_4 \Leftrightarrow$$
$$-\pi_4 = cx$$
$$\text{where } c = \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 & 0 & 0 & 0 \end{pmatrix} \quad (22)$$
$$\text{and } x = \begin{pmatrix} x_d & y_d & z_d & \dot{x}_d & \dot{y}_d & \dot{z}_d \end{pmatrix}^T$$

If $\boldsymbol{\pi}$ is normalized so $\sqrt{\pi_1^2 + \pi_2^2 + \pi_3^2} = 1$, $-\pi_4$ can be interpreted as the signed length of the orthogonal projection of the feature point position onto the direction $\begin{pmatrix} \pi_1 & \pi_2 & \pi_3 \end{pmatrix}$.

Each image with a measurement of a feature position gives two constraints, each specified by a row vector c in (22), one in the x direction and one in the y direction of the image. If several images are available, this can be handled simply by adding rows in the C-matrix of (2):

$$C = \begin{pmatrix} c_{ax} \\ c_{ay} \\ c_{bx} \\ c_{by} \\ \vdots \end{pmatrix} \quad (23)$$

where a and b denote different images and x and y the different measurement directions. Similarly the measurement vector y of (2) is composed of the negative fourth component of the planes $\boldsymbol{\pi}$ in (21):

$$y = \begin{pmatrix} -\pi_{4ax} & -\pi_{4ay} & -\pi_{4bx} & -\pi_{4by} & \ldots \end{pmatrix}^T \quad (24)$$

If no measurement at all is available at some time step, the last term of (5), (7), (8), and (10) simply disappears.

D. Measurement time offset

This section describes a way to handle the case when there is a small known difference between the measurement time and a sampling instant of the time-discrete model.

Assume that, instead of at the Kalman filter sampling time t_k, an image is captured at $t_i = t_k + \Delta t$. Using the flying ball example (19), and the approximation $x(t_i) \approx x(t_k) + \Delta t \cdot \dot{x}(t_k)$, (22) results in the constraint

$$\begin{pmatrix} \pi_1 & \pi_2 & \pi_3 \end{pmatrix}\left(\begin{pmatrix} x_d(t_k) \\ y_d(t_k) \\ z_d(t_k) \end{pmatrix} + \Delta t \begin{pmatrix} \dot{x}_d(t_k) \\ \dot{y}_d(t_k) \\ \dot{z}_d(t_k) \end{pmatrix}\right) = -\pi_4, \quad (25)$$

which can be written in the same form as (22):

$$-\pi_4 = cx$$
$$\text{where } c = \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 & \Delta t \cdot \pi_1 & \Delta t \cdot \pi_2 & \Delta t \cdot \pi_3 \end{pmatrix} \quad (26)$$
$$\text{and } x = \begin{pmatrix} x_d & y_d & z_d & \dot{x}_d & \dot{y}_d & \dot{z}_d \end{pmatrix}^T$$

E. Initialization

Initial values of $\hat{x}$ in (8) and P in (10) have to be chosen. A lower bound on appropriate values of $P(0)$ is determined by the requirement that $\hat{x}(0)$ should have negligible influence on $\hat{x}(k)$ for as small k as possible. A good guideline for this is to make sure that $R_2(k) + C(k)P(0)C^T(k) \approx C(k)P(0)C^T(k)$.

An upper bound on appropriate values of $P(0)$ is determined by the finite numerical accuracy of computers. When $R_2(k) + C(k)P(k)C^T(k) \approx C(k)P(k)C^T(k)$ and $R_{12}(k) + \Phi(k)P(k)C^T(k) \approx \Phi(k)P(k)C^T(k)$ the first and third term of (10) are approximately equal, but of opposite sign. This means that two matrices with large elements are subtracted, resulting in values close to zero. Hence the calculations are very sensitive to round-off approximations.

Following the above recommendations, the value of $\hat{x}(0)$ should be of little importance. Still, in practical applications there usually is a known approximate region in the state space, in which the tracked object is expected to be when it is first detected, and this can be used to choose an appropriate initial value.

As an example, assume that the goal is to track a ball thrown by a human. The initial position should clearly be set to some point in the field of view of the cameras, since this is where it will first be detected. Choosing $P(0)$ corresponding to a standard deviation that is much larger than the size of the field of view makes sure that the actual chosen initial value is of little importance. Assuming that nothing is known about the direction of the throw, a reasonable estimate of the initial velocity is zero. Choosing the initial standard deviation to be a value much bigger than what a human thrower can produce, makes sure convergence is quick when measurements arrive.

V. Outlier detection

Before a measurement of a feature is used in the Kalman filter, a check can be made to verify that the measurement is close to what is expected, based on the state estimate. Otherwise the measurement is discarded.

Each measurement is associated with a measurement vector, $y(k)$, a measurement covariance, $R_2(k)$, and a matrix, $C(k)$, relating the measurement to the state according to $y(k) = C(k)x(k) + e(k)$, cf. (2). To simplify notation, the time indices are omitted in the rest of this section. Further it is assumed that the covariance, P, of the state estimate, $\hat{x}$, is known. Based on the state estimate, the expected measurement and its variance become

$$\begin{aligned} \hat{y} &\equiv C\hat{x} \\ m_{\hat{y}} &\equiv \mathrm{E}[\hat{y}] = \mathrm{E}[C\hat{x}] = C\,\mathrm{E}[\hat{x}] = Cm_x \\ R_{\hat{y}} &\equiv \mathrm{E}[(\hat{y}-m_y)(\hat{y}-m_y)^T] = \\ &= \mathrm{E}[(C\hat{x}-Cm_x)(C\hat{x}-Cm_x)^T] = \\ &= C\,\mathrm{E}[(\hat{x}-m_x)(\hat{x}-m_x)^T]C^T = CPC^T \end{aligned} \tag{27}$$

Now let $d = y - \hat{y}$ be the difference between the actual measurement and the expected measurement. Assuming that the errors in y and $\hat{x}$ are uncorrelated the covariance of d is

$$R \equiv \mathrm{cov}(d) = R_2 + R_{\hat{y}}. \tag{28}$$

The variance in the direction of d is

$$\sigma^2 = \frac{d^T}{\|d\|} R \frac{d}{\|d\|} = \frac{d^T R d}{d^T d}, \tag{29}$$

where $\|d\|$ is the 2-norm of d. The measurement is considered to be an outlier if

$$\|d\| > p\sigma, \tag{30}$$

i.e. if the distance between y and $\hat{y}$ is larger than p standard deviations, where p is a tuning parameter.

VI. Results

The tracker was implemented and tested on series of images of a flying dart, captured by two cameras. The images had a resolution of 656×480 pixels and were captured every 20 ms. The darts were detected in real-time with a processing time of less than 4 ms per image on a desktop PC. The measured positions of the dart, as seen by the two cameras, are shown in Fig. 3.

A. Using two cameras alternately

Figs. 4 – 7 show the result when the tracker acted on the data in Fig. 3. To demonstrate that the dart can be tracked without any pair of images being captured simultaneously, half of the images were left out, so the images from Camera 1 were used only at time steps $k = 1,3,5...$ and the images from Camera 2 were used only at time steps $k = 2,4,6...$.

Fig. 4 shows the position estimates with uncertainty ellipsoids around each estimate. The state estimate was initialized to $(x, y, z) = (0, 0, 3)$ and quickly converged to a

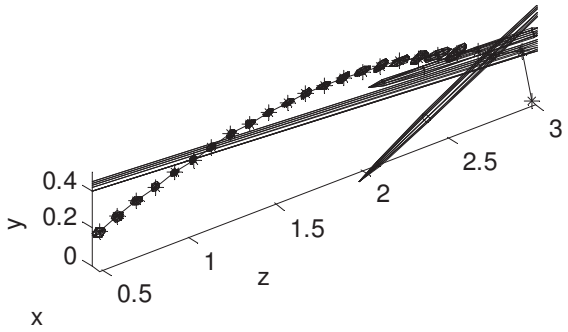

Fig. 4. Tracked positions of the dart with uncertainty ellipsoids at every position estimate.

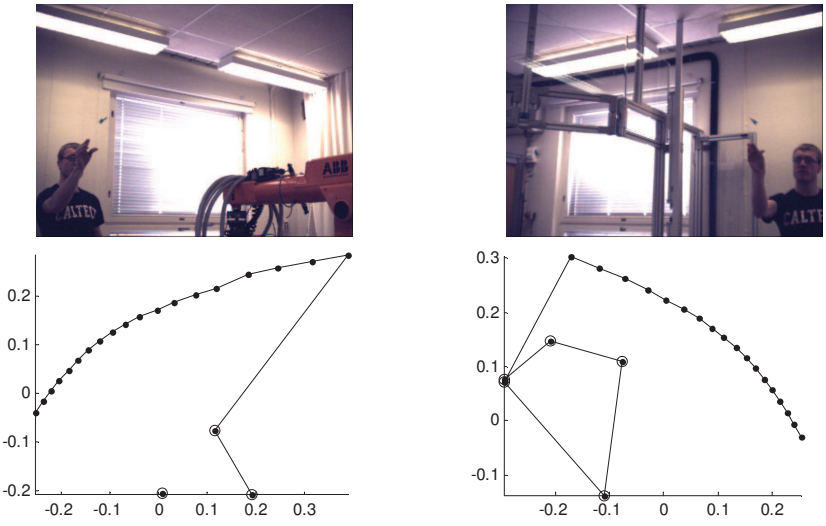

Fig. 3. (Upper) Example images of a thrown dart. (Lower) Sequence of measured dart positions. Measurements classified as outliers by (30) are marked with circles.

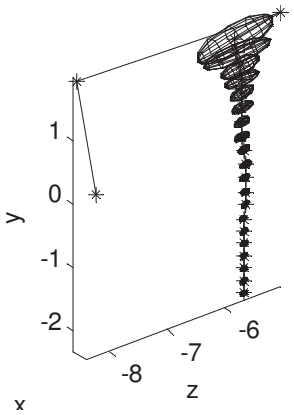

Fig. 5. Tracked velocities of the dart with uncertainty ellipsoids at every velocity estimate

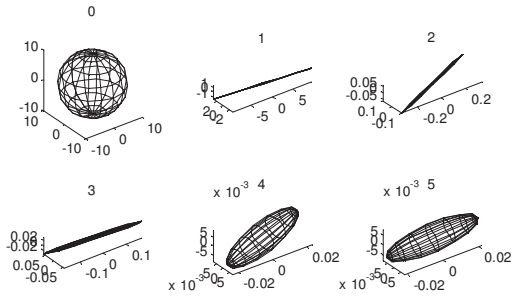

Fig. 6. Uncertainty ellipsoids of the position estimate at the first time steps. The number above the graph indicates the number of used measurements. Note the significantly different scales in the different graphs.

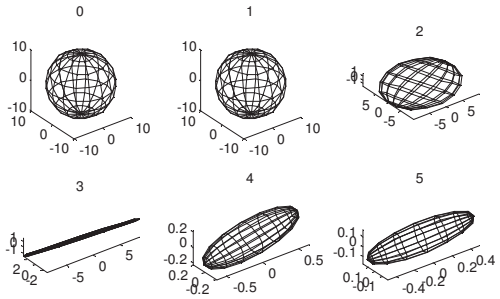

Fig. 7. Uncertainty ellipsoids of the velocity estimate at the first time steps. The number above the graph indicates the number of used measurements. Note the significantly different scales in the different graphs.

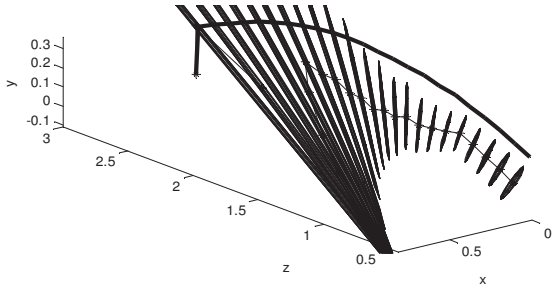

Fig. 8. Tracked position of a dart with measurements only from a single static camera. The tracked trajectory is shown as a thin line and ellipsoids indicate the uncertainty at each time step. The tracked trajectory computed using all images from both cameras is plotted with a thick line for reference.

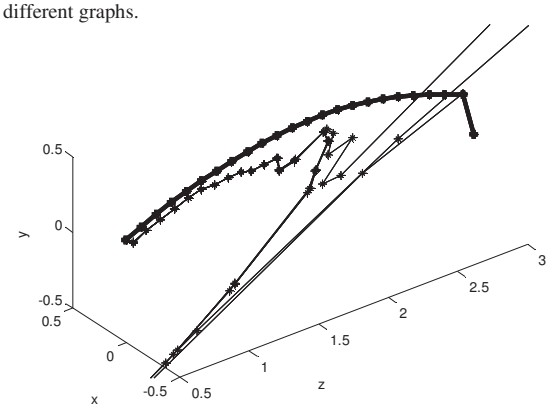

Fig. 9. The thin lines show estimated trajectories based on the same measurement data, but with different initializations. The trajectory estimated using all images from both cameras is plotted with a thick line for reference.

smooth parabola. The uncertainty ellipsoids of the first time steps have been left out in Fig. 4, since they are very large, but they can be seen in Fig. 6. Note the significant difference in scale between the different graphs. The variance (10) was initialized to 10^2 in all directions. When the first measurement was received, the ellipsoid almost reduced to a line, parallel to the viewline of the dart in that image. In the next few time steps the ellipsoid shrunk significantly at every step and stabilized after 4 measurements, Fig. 5 showing the velocity estimates with uncertainty ellipsoids around each estimate. The state estimate was initialized to $(\dot{x}, \dot{y}, \dot{z}) = (0,0,-8)$ and its variance was set to 10^2 in all directions. The first uncertainty ellipsoids are left out, but are shown in Fig. 7. The sizes of the ellipsoids were hardly affected by the first measurements, and a reasonable estimate was obtained after four measurements. This conforms well to the intuition of the problem. To have a good velocity estimate in the directions perpendicular to the viewlines of one camera, images from that camera must be captured at two different times. In order for the estimate to be good in the direction parallel to these viewlines, images have to be captured at two different times from some other direction too.

The initial values of the state estimate were chosen to be close to what is expected when a person is standing in front of a dart board, throwing a dart towards it in a setup like the one in Fig. 1.

B. Using images from only one camera

It is actually possible to track objects using only a single fixed camera, since the object is observed from slightly different angles as it moves across the image. Fig. 8 shows the result when only measurements from the left part of Fig. 4 were used. The uncertainty ellipsoids were initially very oblong, since the position in the direction of the viewlines was very uncertain. As the dart moved and got observed from different angles, the estimate variance decreased drastically. The estimated trajectory based on the images from both cameras is plotted with a thick line for reference. The uncertainty ellipsoids point toward the reference track, indicating that the state variance estimate is a good measure of the uncertainty.

In Fig. 9 the different thin curves represent tracked positions based on exactly the same measurement data, but with different initial values of the Kalman filter. The state was initialized to values in different directions from the actual value and up to 70 m away. The standard deviation of the initial estimate was set to 100 m in all directions. After 6 - 9 measurements all tracks converged to approximately the

same values, showing that the tracker is robust to bad initial values of the Kalman filter, even in this ill-conditioned setup with only one camera.

C. Ball catcher

The tracker was used in the implementation of a ball-catching robot shown in Fig. 1. The experiments were performed on an open robot control system for an industrial manipulator (ABB IRB 140) at the RoboticsLab, Dep. of Automatic Control, LTH, Lund University. The accompanying video submitted with this paper shows the robot as it catches thrown darts and balls. The video can also be downloaded from the web [14].

VII. Discussion

No explicit triangulation was done in the proposed procedure. However, if the rows in C of Eq. (23) correspond to measurements in many different directions over time, an estimate with low variance in all directions can be obtained.

A benefit of this approach is that it easily handles any number of cameras. In (23) and (24) two more rows are simply added to C and y for every camera that captured an image at the same sampling instant. Another advantage is that a measurement can be used even if there are no valid measurements from any other cameras, so that no 3D position estimate could be made from the data from only that sampling instant. It was shown that a dart can be tracked using a single camera. Accuracy is, however, significantly improved if two or more cameras at different positions are used.

If a continuous-time model of the process is available, the procedure can be generalized to completely aperiodic measurements. The expressions (1) – (10) then have to be recalculated at every measurement to reflect the actual time that passed since the previous measurement.

Each row of C and y in (23) and (24) describes a hyperplane in the state space. The above proposed procedure can thus be extended to any feature that can be expressed as a hyperplane in the state space. When tracking a dart, like in Sec. VI, the model (20) could be extended to track the orientation of the dart, along with the position, to allow more accurate tracking. To do this, an Extended Kalman Filter [13] may be needed to capture the nonlinear dynamics. In the robotic context the dart-catching application is an example of eye-hand coordination [2].

The performance of the ball catcher was limited by the accuracy of the image analysis and the speed and acceleration of the robot, which were limited to 0.5 m/s and 13 m/s^2 respectively. The diameters of the ball and the hole were 5 cm and 6 cm respectively. The catch rate was approximately 50 %. Performance was the lowest for trajectories that were far from normal to the box front or that forced the robot to move far from its initial position.

VIII. Conclusions

This paper describes a strategy for tracking dynamical objects with computer vision. A novel way of mapping image space data to a Kalman filter in 3D space is described. It provides a simple way to use data from an arbitrary number of pictures captured simultaneously. In combination with a dynamical model of the tracked object, it enables determination of the position of the object in 3D space without any two images being captured at the same time. It also allows tracking of objects in 3D space, using only a single static camera.

A method to determine whether a measurement is an outlier, based on the covariances of the measurement and the position estimate, is presented.

References

[1] J. Denavit, and R. S. Hartenberg, A kinematic notation for lower-pair mechanisms based on matrices. *J. Applied mechanics*, pages 215-221, June 1955.

[2] B. Espiau, F. Chaumette, and P. Rives, A new approach to visual servoing in robotics, *IEEE Trans. Robotics and Automation*, pp. 313–326, June 1992.

[3] Z. Jia, A. Balasuriya, and S: Challa, Sensor Fusion Based 3D Target Visual Tracking for Autonomous Vehicles with IMM, *Proc. 2005 IEEE Int. Conf. Robotics and Automation*, pp. 1829-1834, Barcelona, Spain.

[4] R. E. Kalman, A new approach to linear filtering and prediction problems, *Trans. ASME—J. Basic Engineering, 82 (1960),* pp. 35–45.

[5] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry, An Invitation to 3-D Vision, 2004, Springer-Verlag, New York.

[6] P. Marayong, G. D. Hager, and A. M. Okamura, Control Methods for Guidance Virtual Fixtures in Compliant Human-Machine Interfaces. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2008)*, pp. 1166-1172, Nice, France.

[7] T. Olsson, J. Bengtsson, A. Robertsson, and R. Johansson, Visual Position Tracking using Dual Quaternions with Hand-Eye Motion Constraints. In *Proc. of the 2003 IEEE Int. Conf. Robotics and Automation*, pp. 3491-3496, Taipei, Taiwan, September 14-19, 2003.

[8] T. Olsson, R. Johansson, and A. Robertsson. Force/Vision Based Active Damping Control of Contact Transition in Dynamic Environments. In *Proc. 10th IEEE Int. Conf. Computer Vision, Workshop on Dynamical Vision*, Beijing, October 2005.

[9] T. Olsson, R. Johansson and A. Robertsson, High-speed visual robot control using an optimal linearizing intensity-based filtering approach, *Proc. 2006 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS2006)*, October 9-15, 2006, Beijing, China, pp. 1212-1217.

[10] T. Olsson, R. Johansson, and A. Robertsson, Flexible Force-Vision Control for Surface Following using Multiple Cameras, *Proc. 2004 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS2004)*, Sept. 28-Oct. 2, 2004, Sendai, Japan, pp. 798-803.

[11] Y. Sato, J. Takamatsu, H. Kimura, K. Ikeuchi, Recognition of a Mechanical Linkage Based on Occlusion-Robust Object Tracking. In *Proc. IEEE Conf. Multisensor Fusion and Integration for Intelligent Systems, 2003*, pp. 329- 334, Tokyo, Japan.

[12] U. Frese, B. Bäuml, S. Haidacher, G. Schreiber, I Schaefer, M. Hähnle, G. Hirzinger. Off-the-shelf Vision for a Robotic Ball Catcher, *Proc. 2001 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Oct. 29-Nov. 03, 2001, Maui, Hawaii, pp. 1623-1629.

[13] S. F. Schmidt, Applications of state space methods to navigation problems, in Advanced Control Systems, C. T. Leondes, ed., vol. 3, Academic Press, New York, 1966, pp. 293–340.

[14] Video of dart and ball catching robot. http://www.control.lth.se/user/magnusl/ball_and_dart_catcher_August2009_0001.wmv