Real-Time 3D Ball Trajectory Estimation for RoboCup Middle Size League using a Single Camera

H.Silva, A.Dias, J.M. Almeida, A. Martins, E.P. Silva

INESC PORTO/ISEP
ISEP – Instituto Superior de Engenharia do Porto
ESEIG – Escola Superior de Estudos Industriais e de Gestão
Rua Dr Antonio Bernardino de Almeida 431 4200-072 Porto, Portugal
{hsilva,adias,jma,aom,eaps}@lsa.isep.ipp.pt
http://www.lsa.isep.ipp.pt

Abstract. This paper proposes a novel architecture for real-time 3D ball trajectory estimation with a monocular camera in Middle Size League scenario. Our proposed system consists on detecting possible multiple ball candidates in the image, that are filtered in a multi-target data association layer. Validated ball candidates have their 3D trajectory estimated by Maximum Likelihood method (MLM) followed by a recursive refinement obtained with an Extended Kalman Filter (EKF). Our approach was validated in real RoboCup scenario, evaluated recurring to ground truth information obtained by alternative methods allowing overall performance and quality assessment.

Keywords: Monocular Vision, 3D Trajectory Estimation, Maximum Likelihood Method, Extended Kalman Filter, RoboCup Middle Size League

1 Introduction

This work addresses an important problem in RoboCup Middle Size League (MSL)[5], which is the real-time ball trajectory estimation in 3D space, using only a perspective monocular camera. Our MSL Team ISePorto robots, have as primary mean of perception, three perspective monocular cameras. However, due to a restrictive number of factors mainly related to the lack of camera synchronization together with the currently available cameras field of view, makes stereo vision not our most suitable option.

Considering previous restrictions, 3D ball estimation by a single camera is a demanding task, since current MSL robots are able to kick a soccer ball at very high velocities (more than 10 m/s). This is particularly challenging when the ball is kicked into the air at high velocity and describes a 3D trajectory that must be estimated in real-time.

Our camera system provides a sequence of observations which are non-linearly related to the ball position in 3D. The estimation of parameters from a sequence of non-linear observations is a non-linear filtering problem which can be tackled using recursive algorithms such as the Extended Kalman Filter and the Unscented Kalman filter [2]. These methods are appropriate for real-time applications since they recursively update the estimates every time a new observation is received. However, they require a

Fig. 1. Image sequence of a typical 3D ball trajectory in Robocup Middle Size League Scenario, (game between Cambada and TechUnited)

good initialization. Unfortunately, the available sensor does not provide an accurate estimate of ball position and velocity after a kick. This prevents a widespread use of the previous methods as stand-alone techniques.

To overcome this difficulty we combine two methods: the Maximum Likelihood method (MLM) which is able to provide an initial estimate of the trajectory and the Extended Kalman filter (EKF) which is able to recursively update the trajectory using new information. The MLM uses the first few frames of the ball trajectory to obtain the initial estimate and the EKF tracks the ball in the following frames, both methods are combined in a Multi-Target data association layer, that follows validated ball candidates presented in the retrieved images. This association layer uses image target dynamics in robot world coordinates to distinguish between possible ball candidates and false positive ones.

The 3D ball estimation system is based on the analysis of 3D motion parameters estimated from 2D image observations: image ball coordinates are detected using vision system described in [9]. Physical ball motion model and camera projective model are combined in order to obtain a set of measurements, non-linearly related to the 3D ball position and velocity of the ball at the initial time instant. After, a ML estimator is applied to the ball trajectory, pure parabolic trajectory is assumed. The number of available data measurements influences ML estimator error, so by analysis of a cost function together with the number of available data measurements it's possible to define when the ML estimator error is below a given threshold.

This paper is organized as follows:

– Section 2 presents related work associated to the problem; some of the methods already address the real time concerns.
– Section 3 formulates the problem and the describes the architecture proposed in this paper.
– Section 4 presents experimental results and a comparison with ground truth information.
– Section 5 presents conclusions and discusses future work.

2 Related Work

The 3D ball estimation problem, in Robocup Scenario was first approached by Voigtlander et al.[10], who used a mixture of Perspective and Catadioptric cameras to estimate the 3D ball trajectory. The 3D-ball position was given by basic stereo triangulation extended by a 2D Kalman filter approach to the 3D scenario. Motivated by the same challenge Birbach et al. [2] developed a monocular vision system with aide of a inertial sensor to track a moving ball. The pose of the camera is obtained by integrating gyro and accelerometers measurements, to minimize the error caused by inertial measurements, visual landmarks of the RoboCup field are used to correct the camera pose estimation. The camera provides the direction towards the ball through it's image position, depth information comes from the effect of gravity over time, which directly influences the ball radius in the image. This paper also compares results obtained using a ML estimator approach with an Unscented Kalman filter(UKF), but unlike our approach there is no integration of both methods. There are also other approaches that performs online 3D trajectory estimation using EKF [7], however this technique only works if the camera is on a stationary position, becoming a straight forward solution.

Furthermore, 3D trajectory ball estimation is also being applied in other types of systems, namely the ones related to computer game study of sports. The work realized by Chen et al. [4], estimates 3D ball motion from 2D frames in basketball games. It incorporates physical characteristics of ball motion to reconstruct the 3D trajectories and estimate the player shooting locations. Another 3D ball estimation work related to sport analysis was developed by Yu et al. [11], who conducted ball estimation analysis in soccer videos.

Other approach is presented by Benovoli et al. [1], that uses MLM to help estimate the ballistic target problem. Some of the work proposed by Ribnick et al. [8], also uses MLM in the estimation of 3D projectile positions and velocities using monocular cameras.

In the literature, there are other application scenarios related to different types of sensors namely radar systems, Farina [6], also applies ML estimation techniques to track airborne objects.

3 Implementation of 3D Ball Estimation Method

3.1 3D Ball Estimation Architecture

This work is based on achieving 3D ball estimation with a single camera for use in real time applications, such as Robocup Middle Size League.

The 3D ball estimation is achieved using a robot that has a differential traction wheelbase with an additional rotating upper body housing the onboard computer, wireless communications module, inertial and magnetic sensors and a kicking device. The kicker mechanism has two fixed cameras, one used mainly for close range ball detection and the other for 3D aerial ball estimation. Has mentioned previously, due to having a short baseline, stereo vision is not our most suitable option. On top of the upper body, the robot has a rotating head with a camera for long range target tracking.

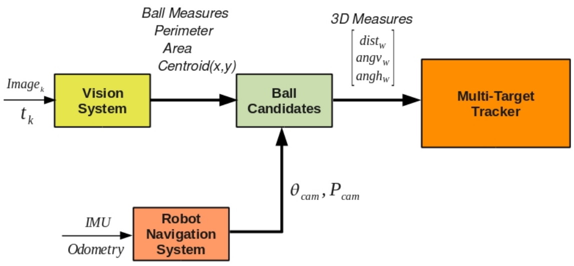

Fig. 2. Ball estimation Architecture

Overall system architecture is described in figure 2. The vision system starts by acquiring images at a pre-determinate frame rate (in this case 30 fps 640*x*480 image resolution).

Afterwards, image segmentation is conducted and relevant object (ball) information is extracted from the image, according to the work described in detail by Silva et al. [9]. Basically, image ball information is retrieved using image blob extraction techniques together with edge detection that uses weighted least squares circle estimation, in order to obtain a more robust ball estimation. This information, contains all relevant ball information measures such as: ball perimeter area, bounding box, ball centroid and ball radius, all of this in 2D image plane.

Due to the fact that MSL League has a very dynamic background environment, more than one ball candidate may appear in the same image, thus the system must be able to cope with multiple ball target candidates,

Additionally, information is provided by the robot Navigation System, namely: odometry information provided by wheel encoders and gyro information given by the robot Inertial Measurement Unit. This information is fused to obtain the camera position and bearing (P_{cam},θ_{cam}) in World coordinates, see figure (1,2). The World frame is centered in the middle of the Robocup Field with x axis oriented from our team side to the opponents side.

$$\theta_{cam} = \theta_{kicker} + \theta_{robot} \tag{1}$$

$$P_{cam} = P_{robot} + R(\theta_{cam}) \times P_{camkicker} \tag{2}$$

Having the 2D ball information fused with the robot information, we can obtain 3D information of the ball related to World frame but in other coordinate frame whose center is the robot middle point (at the wheelbase).

The 3D ball candidates information is managed by a Multi-target Tracker framework. This framework starts by initializing a ML estimator for each of the possible ball candidates. After the ML estimator converges, we can initialize a EKF-filter to track the ball. The objective of this framework is twofold: First, we use the tracker to eliminate erroneous measures of balls that appear in the public, by analysing it's dynamic related to the robot motion. Second, we are able to initialize the EKF-filter only when

ML estimator has good convergency and analyse it's behavior when we update the ball information in consecutive frames.

3.2 Camera Model

Considering a pinhole camera projection model, image points relate to the camera reference frame by means of the camera intrinsic parameters. To obtain these parameters an offline calibration [3] is performed, by acquiring measures to a known 3D target.

The intrinsic parameters enables us to deal with camera lens distortion factor and allows to obtain the camera internal parameters (3).

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} fkx & k_\theta & x_0 & 0 \\ 0 & fky & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix} \tag{3}$$

Where (x, y) is the point projection in the image plane, f the camera focal length and (x_0,y_0) the principal point, kx and ky are scaling factors converting from space metrics to pixels in image and k_θ is an additional skew factor that usually is set to 0.

Lens distortion (such as occurring in wide angular lens) can be corrected prior by correcting the relevant pixel through a pre-calculated look up table.

We can relate the image reference frame with the camera in the kicker coordinated frame by means of some pre-determinated (calibrated) camera parameters ($R_{\theta cam}$, $T_{\theta cam}$), (4). These angles are have their initial point in the camera but they are aligned with the World coordinate frame.

$$P_{img} = A \left[R_{\theta cam} | T_{\theta cam}\right] P_{camkicker} \tag{4}$$

After the ball estimation points in the kicker coordinate frame are obtained it's possible to calculate the horizontal and vertical angle ($ang_h w$, $ang_v w$) in polar coordinates by means of simple perspective projective geometry. In figure 3, one can observe the different coordinate frames, from the image plane to world coordinate frame.

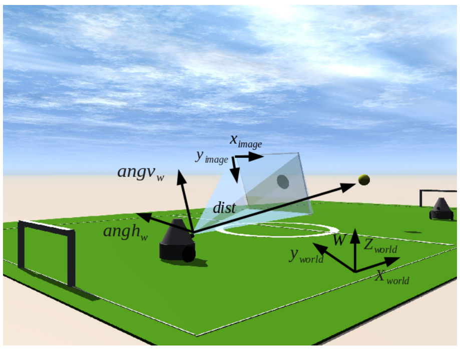

Fig. 3. System Reference Frames

In order to obtain the distance to the ball, we need to acquire bearing measures of two points of the ball. The estimated ball radius and added with the estimated ball center is used to obtain the other ball point. Since the ball radius is previously known and it's equal to a FIFA size 5 ball, the distance to the ball is thus given by, (6).

$$\theta_{ball_{cxy}} = \arctan\left(\frac{\cos(\arctan\left(\frac{y-C_{py}}{F_{cy}}\right))\tan(angh_w)}{\cos(angv_w)}\right) \tag{5}$$

$$dist_w \simeq \left(\frac{R_{ball}}{angh_w - angh_{w\,plimit}}\right) \tag{6}$$

Where $angh_{w\,plimit}$ is the polar coordinate horizontal angle to a image point of the edge of the ball at the right of the ball center.

3.3 Parabola Model

Ball trajectory is considered to be a parabola in the x, y and z World coordinate frame (see figure 3). The ball trajectory can be formally represented by a dynamical system given by ordinary differential equations as a function of time T that follow classical mechanics such as gravitation.

Trajectory equations are given by (7):

$$\begin{cases} x(T) = x_0 + v_x T \\ y(T) = y_0 + v_y T \\ z(T) = z_0 + v_z T - \frac{1}{2} g T^2 \end{cases} \tag{7}$$

Where g is the gravity acceleration and v_x, v_y and v_z are the velocity components in relation to x,y and z. The trajectory described by the ball belongs to a plane which determines an angle with the x axis having a start point $P_{init}(x_0, y_0, z_0)$ with $(z_0 = 0)$ and $T = 0$, a maximum height $P_{max}(x, y, z)$ and an impact point $P_{end}(x, y, z)$ for $T \neq 0$ at $z(T) = 0$.

$$\begin{gathered} \frac{\mathrm{d}z(T)}{\mathrm{d}T} = v_z - gT = 0 \\ T_{Zmax} = \frac{V_z}{g} \end{gathered} \tag{8}$$

Where T_{Zmax} is the ball maximum height.

3.4 Maximum Likelihood Method

In 3D ball estimation, ball trajectory can be identified by the initial point $P_0(x_0, y_0, z_0)$ and by it's initial velocity vector, whose components are v_x, v_y and v_z, if we obtained the kicking instance t_0. Taking in consideration all k images frames, measurements of distance ($dist_w(k)$), vertical angle ($angv_w(k)$) and horizontal angle ($angh_w(k)$), collected into the vector measurement:

$$z(k) = \left[dist_w(k)\ angv_w(k)\ angh_w(k)\right]^T + n_k \tag{9}$$

with a measurement noise which is Gaussian, zero-mean and with the following covariance matrix:

$$R = \begin{bmatrix} \sigma^2_{dist_w} & 0 & 0 \\ 0 & \sigma^2_{angv_w} & 0 \\ 0 & 0 & \sigma^2_{angh_w} \end{bmatrix} \tag{10}$$

$\sigma^2_{dist_w}$ $\sigma^2_{angv_w}$ $\sigma^2_{angh_w}$ are the variances of the ball position measurement errors. The MLM calculates the unknown vector parameter $x = [x_0, y_0, z_0, v_x, v_y, v_z]^T$. The unbiased estimator $\hat{x}$ is obtained by solving the minimization problem $\hat{x} = argmin_x \Lambda(x)$, where $\Lambda(x)$ represents the ML estimator functional and T the camera frame rate.

$$\Lambda(x) = argmin_x \sum_{k=1}^{N} [Z_k - h(x)]^T R^{-1} [Z_k - h(x,k)] \tag{11}$$

$$h(x,k) = \begin{cases} \sqrt{(xo + vxt(k) - Pcx)^2 + (yo + vyt(k) - Pcy)^2 + (zo + vzt(k) - \frac{1}{2}gt(k)^2 - Pcz)^2} \\ \arcsin \frac{zo+vzt(k)-(1/2)gt(k)^2-Pcz}{(xo+vxt(k)-Pcx)^2+(yo+vyt(k)-Pcy)^2+(zo+vzt(k)-\frac{1}{2}gt(k)^2-Pcz)^2)} \\ \arctan \frac{yo+vyt(k)-Pcy}{xo+vxt(k)-Pcx} \end{cases} \tag{12}$$

3.5 3D-Extended Kalman Filter

To perform 3D ball estimation, we used an Extended Kalman Filter approach, that was initialized using the ML estimator fit.

This is useful because the EKF employs a model that only possesses an approximate knowledge of the state. Unlike the linear Kalman Filter algorithm, the filter must be accurately initialized at the start of operation to ensure that we obtain a valid model. If this is not done, the estimates computed by the filter will simply be meaningless. So, we initialized $\hat{X}$, using the MLM estimator and thus calculate the ball trajectory, based on those estimations.

We developed two different approaches to the EKF filter, one where the State was the target (ball) current position and velocity, and the other one which is presented in this paper, where the state is the ball initial position and velocity(x0, y0, z0 , vx0, vy. We preferred the initial position approach to the EKF filter, because estimating the initial position of the ball, is much smoother than tracking at each time instant, whose measure are more noisy.

Filter State is:

$$\hat{X} = \left[\hat{X_{x0}}\ \hat{X}_{y0}\ \hat{X}_{z0}\ \hat{X}_{v_x}\ \hat{X}_{v_y}\ \hat{X}_{v_z}\right]$$

With the Predict Step:

$$\hat{X}_{k|k-1} = \hat{X}_{k-1}$$

$$\hat{P}_{k|k-1} = \hat{P}_{k-1} + Q$$

and Update Step:

$$\tilde{y}_k = z_k + h(\hat{X}_{k|k-1})$$

$$S_k = H_k P_{k|k-1} H_k^T + R$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k \tilde{y}_k$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

4 Experimental Setup

Experimental setup is divided into two parts: First, the ISePorto robot setup and second the external stereo vision system used to produce the 3D ground-truth analysis.

Robot setup Tests were conducted using the ISePorto Robot in a middle size field. The robot was positioned fixed at the goal. Robot vision hardware is composed by three USB cameras, working at 30 FPS with VGA (640x480) resolution connected to a Intel Pentium Dual Core PC 2 GHz and 2GB memory running a RT-Linux operating system. High level software is composed by several modules, being one of this modules the acquisition and image processing system [9], implementing the ball detection algorithm.

Ground-Truth 3D Ball Tracking In order to evaluate the quality of the 3D ball estimation from the ISePorto robot, an exogenous system of cameras in a stereo baseline was introduced as a ground-truth(GT), figure 4. Cameras were positioned with a baseline of 13 meters and connected via GigEthernet to a Intel Pentium Dual Core PC 2 GHz running a Linux operating system. The cameras used are Jai CB080GE working at 20 FPS with 1024x768 resolution with external sync trigger.

In performed tests, the ball is moving during most of the time. Therefore, first step in ball detection by ground-truth cameras is to extract moving objects by a background frame subtraction operation. The pixel-wise subtraction extracts all the moving objects in the frame obtaining not only a ball but also other type of objects. We eliminate all objects that are not possible ball candidates by performing colour threshold segmentation in HSV space

5 Results

We performed multiple tests to evaluate the ML estimator fit and EKF estimation, in several parabolic kicks. Some of those results are presented in this section. First, we evaluated the results obtained by the robot MLE estimator versus the Ground-truth, using the images acquired by the stereo vision GT system. In figure 4, the red line is ML estimator from the ground truth and in black is the ML estimator from the robot.

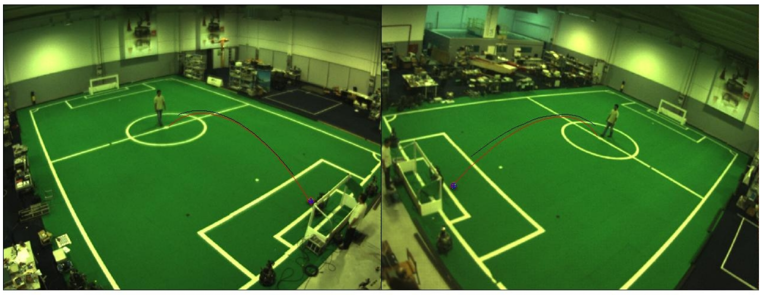

Fig. 4. MLE fit from Ground-Truth stereo compared to the Robot MLE estimator

In figure 5, the ground-truth error for a parabolic ball trajectory is displayed. One can observe that ground-truth ball does not describe a true parabolic trajectory, but shows some side effects. In figure 6, one can observe results obtained by the robot MLE estimator for different observations measures.

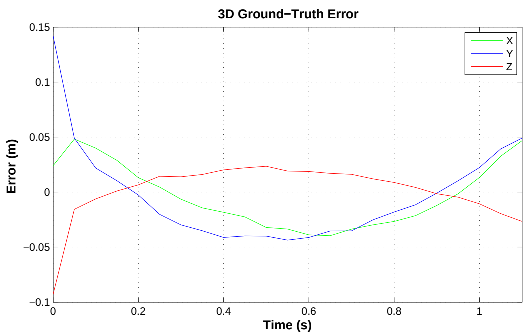

Fig. 5. Ground-truth error

In figure 7, one can observe the 3D ball trajectory estimation analysis. On top, an upper view of the parabola trajectories are displayed in World coordinates. It shows the converges of the 3D ball trajectory has the MLE estimator of the robot gets more ball observation measures. It also displays the convergency between robot and GT MLE

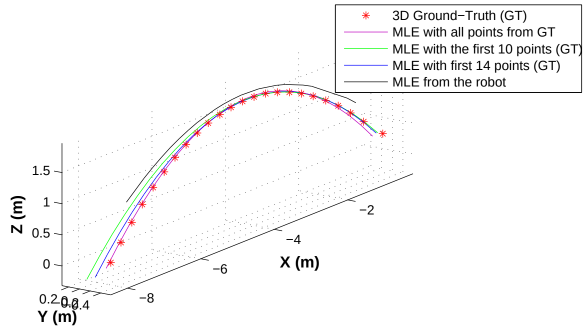

Fig. 6. Ground-Truth and MLE from the Robot with different initialization points

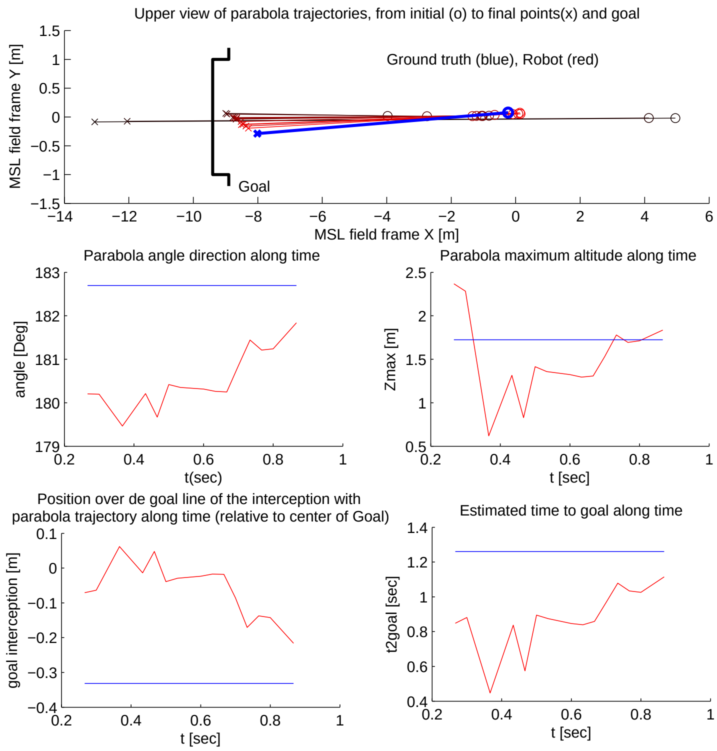

Fig. 7. 3d Ball Estimation Trajectory Analysis

estimators. In the bottom left of the figure, one can observe the position over the goal line of the described parabola trajectory. It's possible to observe compared to figure 5, that the ball describes a small trajectory deviation at the beginning of the trajectory.

From figure 8 one can observe the MLE estimator behavior as a initialization mechanism to the EKF filter. This results represents another benchmark test performed with the robot and ground-truth..

In figure 8,results obtained by the ML estimator together with EKF estimation for the same kick are displayed. The ball described two parabolic trajectories in the image. The difference between the ML estimator and the EKF estimation does not exceed 2 cm standard deviation.

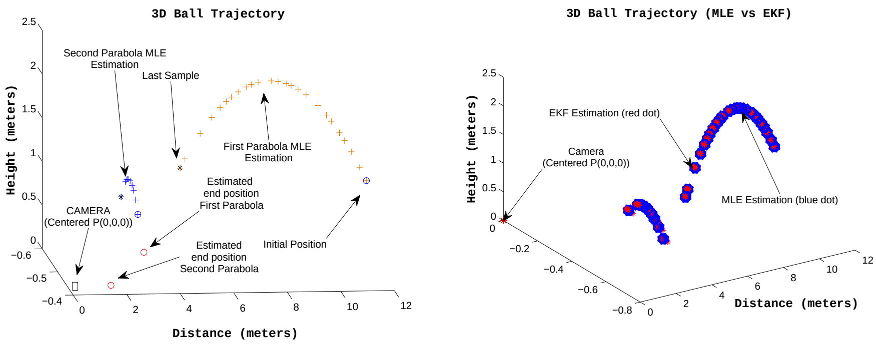

Fig. 8. Parabolic 3D ball Estimation Trajectory ML estimator and EKF

6 Conclusions

This paper describes a real-time 3D ball estimation using a single camera for Robocup league applications. We started by implementing a Maximum Likelihood Estimator for parabolic trajectories, since air ball trajectories in RoboCup scenario are similar to a projectile estimation problem.

As an input data to the ML estimator ,a ball detection algorithm from our ISePorto MSL Team, was used together with robot navigation system information, in order to obtain 3D ball estimates.

Using these ball estimates, a optimization procedure that minimizes the error between the ML estimator and the ball observed measures, was conducted.

However, our ML estimator fit works in batch computation and requires at least 8 measures to have a good estimate of the ball trajectory, this is equivalent to almost 0.5 seconds of our camera frame rate time. In order to try to extend the ML estimator approach to be computed in real-time, we developed an Extended Kalman Filter that used the ML estimator fit has initial condition. The obtained results displayed in the results section, show the innovation of the EKF is always well inside the 95% confidence bound and is unbiased. Furthermore, we state that the EKF and ML estimator both estimate ball trajectory with very small standard deviation. All results were evaluated using a ground-truth stereo vision system, proving that results obtained by the robot using only a single camera are similar to the ones obtained by external synchronize stereo vision system.

Acknowledgements

This work was supported by Portuguese FCT under grant SFRH/BD/47468/2008 and PCMMC project with reference PTDC/EEA-CRO/100692/2008, and by Porto Politechnic Institute under grants SFRH/PROTEC/49796/2009, SFRH/PROTEC/49795/2009 and SFRH/PROTEC/49716/2009.

References

1. A. Benavoli, A. Farina, and L. Ortenzi. Mle in presence of equality and inequality nonlinear constraints for the ballistic target problem. In *Radar Conference, 2008. RADAR '08. IEEE*, pages 1 –6, 26-30 2008.
2. Oliver Birbach, Jörg Kurlbaum, Tim Laue, and Udo Frese. Tracking of Ball Trajectories with a Free Moving Camera-Inertial Sensor. In Luca Iocchi, Hitoshi Matsubara, Alfredo Weitzenfeld, and Changjiu Zhou, editors, *RoboCup 2008: Robot Soccer World Cup XII*, volume 5399 of *Lecture Notes in Artificial Intelligence*, pages 49–60. Springer, 2009.
3. J. Y. Bouguet. Camera calibration toolbox for matlab, 2008.
4. Hua-Tsung Chen, Ming-Chun Tien, Yi-Wen Chen, Wen-Jiin Tsai, and Suh-Yin Lee. Physics-based ball tracking and 3d trajectory reconstruction with applications to shooting location estimation in basketball video. *Journal of Visual Communication and Image Representation*, 20(3):204 – 216, 2009.
5. Andre Dias, Jose Almeida, Alfredo Martins, and E.P Silva. Traction characterization in the robocup middle size league. *IEEE Robotica 2007 - 7th Conference on Mobile Robots and Competitions*, 2007.
6. A. Farina, A. Di Lallo, L. Timmoneri, T. Volpi, and B. Ristic. Crlb and ml for parametric estimate: new results. *Signal Process.*, 86(4):804–813, 2006.
7. R. Herrejon, S. Kagami, and K. Hashimoto. Online 3-d trajectory estimation of a flying object from a monocular image sequence. In *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, IROS'09, pages 2496–2501, Piscataway, NJ, USA, 2009. IEEE Press.
8. Evan Ribnick, Stefan Atev, and Nikolaos P. Papanikolopoulos. Estimating 3d positions and velocities of projectiles from monocular views. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):938–944, 2009.
9. H Silva, J Almeida, L Lima, A Martins, and E.P Silva. A real time vision system for autonomous systems: Characterization during a middle size match robocup 2007. *Robot Soccer World Cup XI Lecture Notes in Computer Science*, 5001, 2008.
10. Arne Voigtlander, Sascha Lange, Martin Lauer, and Martin Riedmiller. Real-time 3d ball recognition using perspective and catadioptric cameras, 2005.
11. Junqing Yu, Yang Tang, Zhifang Wang, and Lejiang Shi. Playfield and ball detection in soccer video. In *Proceedings of the 3rd international conference on Advances in visual computing - Volume Part II*, ISVC'07, pages 387–396, Berlin, Heidelberg, 2007. Springer-Verlag.