REAL-TIME ELLIPSE FITTING, 3D SPHERICAL OBJECT LOCALIZATION, AND TRACKING FOR THE ICUB SIMULATOR

Nicola Greggio [⊘,‡], Alexandre Bernardino [‡], Cecilia Laschi [⊘], Paolo Dario [⊘], José Santos-Victor [‡]
[⊘] *ARTS Lab - Scuola Superiore S.Anna, Polo S.Anna Valdera, Viale R. Piaggio, 34 - 56025 Pontedera, Italy*
[‡] *Instituto de Sistemas e Robótica, Instituto Superior Técnico, 1049-001 Lisboa, Portugal*
(ngreggio, alex, javs)@isr.ist.utl.pt

Keywords: Humanoid Robotics, Machine Vision, Pattern Recognition, Least-Square Fitting, Algebraic Distance

Abstract: This paper presents the implementation of real-time tracking algorithm for following and evaluating the 3D position of a generic spatial object. The key issue of our approach is the development of a new algorithm for pattern recognition in machine vision, the Least Constrained Square-Fitting of Ellipses (LCSE), which improves the state of the art ellipse fitting procedures. It is a robust and direct method for the least-square fitting of ellipses to scattered data. Although it has been ellipse-specifically developed, our algorithm demonstrates to be well suitable for the real-time tracking any spherical object, and it presents also robustness against noise. In this work we applied it to the RobotCub humanoid robotics platform simulator. We compared its performance with the Hough Transform, in terms of robustness (success/failure in the object detection) and fitting precision. We performed several tests to prove the robustness of the algorithm within the overall system. Finally we present our results.

1 INTRODUCTION

The impressive advance of research and development in robotics and autonomous systems over the past few years has led to the development of robotic platforms of increasing motor, perceptual, and cognitive capabilities. These achievements are opening the way for new application opportunities that will require these systems to interact with other robots or nontechnical users during extended periods of time. The final goal is creating autonomous machines that learn how to execute complex tasks and improve their performance throughout their lifetime. Motivated by this objective the RobotCub (ROBotic Open-Architecture Technology for Cognition, Understanding and Behavior) project has been developed (Sandini et al., 2007). This is a research initiative dedicated to the realization of embodied cognitive systems.

1.1 Related work

The detection of circular objects is fundamental in many applications, other than the developmental RobotCub scenarios. An example is in the rescue employment of robotics platforms. A rescue robot is a robot that has been designed for the purpose of aiding rescue workers. Common situations that employ rescue robots are mining accidents, urban disasters, hostage situations, and explosions. Currently, the research in rescue robotics is very fruitful (Carpin et al., 2007). In addition, the NIST implemented a simulator, USARSim (Urban Search And Rescue Simulation) in order to develop rescue robots (Wang et al., 2003). Circular ad elliptical objects can occur in body parts, such as head, and eyes. Obviously, a clear and precise object recognition is fundamental for such robots to find an accident victim as soon as possible with the highest precision as possible. Vamossy *et al* applied an ellipse detection algorithm to a rescue robot (Vamossy et al., 2003) in 2003, while, more recently, Greggio *et al* used an ellipse detection algorithm for recognizing the ball within the RoboCup context (Greggio et al., 2009).

Other work in the robotics implementation of ellipse pattern recognition techniques has been performed. Deniz *et al* used an ellipse detection algorithm for face detection (Deniz et al., 2002). In their work the authors focussed more on Human-computer

interaction. Moreover, Vincze *et al* used a RANSAC-like method to find ellipses in real-world examples (Vincze et al., 2000). They made experiments to validate the capabilities of the approach with in real contexts. Finally, Teutsch *et al* applied the ellipse recognition in industrial processes, focussing on the real-time characteristics of their approach (Teutsch et al., 2006).

1.2 Our Contribution

In this paper we implemented for the first time in a real context our least-square fitting of ellipses technique (Greggio et al., 2010). We tested our new algorithm, the B2AC (Fitzgibbon et al., 1999), and the Hough transform (Leavers, 1992) under the same experimental conditions. We choose an actual task, i.e. 3D localization of a ball, and we compared these algorithms' performance in terms of overall localization precision, and their robustness in terms of success/failure detection of the object. We used the simulation of a state of art robotics platform, the RobotCub, in order to test it at best before doing this with the real platform.

1.3 Outline

This paper is organized as follows. In section 1.3 we will discuss the state of the art problem of the least-square fitting of ellipses. Then, in section 4 we will describe the RobotCub robotics platform, in terms of its mechanics and the simulator we used. Furthermore, in section 5 we will briefly explore our vision algorithms. In sec. 6 we will describe our experimental set-up. In section 7 we will discuss our results. Finally, in section 8 we will conclude our work and explain our projects as future research.

2 LEAST SQUARE FITTING OF ELLIPSES

2.1 Least Square Fitting of Ellipses and Hough transform: The State of the Art

Two main approaches can be considered for circle detection.

The first one is to use the Hough Transform (Yuen et al., 1989). Since spatial perspective alters the perceived objects, there is the need of calibrating the camera(s). Then, a pattern recognition algorithm, such as a simple color detection, can be applied and subsequently the Hough circle transform can be applied in order to estimate all the ball's features.

However, this approach can be complex to be implemented, and even elevate resource consumption. First, it requires the camera calibration. Moreover, it can be argued that using a Hough Transform, for instance, by augmenting the image's resolution the computational burden increases as well. Finally, the Hugh transform needs to be set well, in terms of the accumulator threshold at the center detection stage parameter.

The second one is to use ellipse specific pattern recognition algorithms, such as (Maini, 2006), (Fitzgibbon et al., 1999). By processing a ball thinking of it as it were an ellipse, we overcome the distortion problems. Circles in man-made scenes are almost always distorted when projected onto the camera image plane, therefore generating ellipses.

Some approaches based on Least Square (LS) come out in recent years (Fitzgibbon et al., 1999), (Gander et al., 1994). The principal reason is because of its computational costs. There are two main kinds of LS techniques: those based on the minimization of the algebraic distance between (Algebraic Distance Least Square, ADLS) the data points and the ideal curve (intended as the deviations of the implicit equation from the expected value, i.e. zero, at each given point) and those based on the minimization of the geometric distance (intended as the orthogonal, or shortest, distances from the given points to the geometric feature to be fitted) of the same data points (Geometric Distance Least Square, GDLS). ADLSs suffer of high curvature bias (Kanatani, 1994) with the the non-invariance to Euclidean transformation (Zhang, 1997). However, GDLSs suffer of being dependent of iterative algorithms (Rosin and West, 1995) as do cluster/voting (CV) techniques, therefore making them not suitable for real-time applications (Fitzgibbon et al., 1999). This is a notable drawback, because iterative algorithms do not have a fixed computational time. Nevertheless, algebraic fitting algorithms may guarantee a direct one-step convergence. We will focus on this way, starting from the work of Fitgibbon *et Al.*, called B2AC, which will be described in the next section (Fitzgibbon et al., 1999).

2.2 Least Square Fitting of Ellipses Algorithm

A central conic can be expressed by a second order equation in its implicit form, as follows in the eq. (1):

$$F(x,y) = ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (1)$$

This can also be expressed in the vectorial form:

$$F_a(\mathbf{x}) = \mathbf{x} \cdot \mathbf{a} = 0 \tag{2}$$

where $\mathbf{a} = [a,b,c,d,e,f]^T$ is the vector of the equation coefficients, and $\mathbf{x} = [x^2, xy, y^2, x, y, 1]$ is the vector of the points' coordinates, both relative to the conic section.

Considering that we have this set of data points:

$$T = \{(x_i, y_i) : i = 1...N\} \tag{3}$$

our aim is to minimize the sum of the squared distances of the curve 1 to he given points (3). In other words, by assuming $F(\mathbf{a}, \mathbf{p_i})$ as the *algebraic distance* from the point $\mathbf{p_i} = (x_i, y_i)$ to the conic expressed by (2) the following non-linear minimization problem has to be solved (DeSouza and Kak, 2002):

$$min_a(\sum_{i=1}^{N} F(\mathbf{a}, \mathbf{p_i})) = min_a(\sum_{i=1}^{N} F(\mathbf{a} \cdot \mathbf{p_i})^{\mathbf{2}}) \tag{4}$$

In (Fitzgibbon et al., 1999) Fitzgibbon *et Al.* demonstrated that solving the problem with the following constraints gives rise to a unique exact solution:

$$\begin{cases} \min \|\mathbf{D} \cdot \mathbf{a}\|^2 \\ \mathbf{a^T} \cdot \mathbf{C} \cdot \mathbf{a} = \mathbf{1} \end{cases} \tag{5}$$

where

$$\mathbf{D} = \begin{pmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^2 & x_N y_N & y_N^2 & x_N & y_N & 1 \end{pmatrix} \tag{6}$$

and

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{7}$$

Now, by using the Lagrange mulltiplier λ and differentiating we obtain:

$$\begin{cases} 2\mathbf{D^T D a} - 2\lambda \mathbf{C a} = \mathbf{0} \\ \mathbf{a^T} \cdot \mathbf{C} \cdot \mathbf{a} = \mathbf{1} \end{cases} \tag{8}$$

or in the form:

$$\begin{cases} \mathbf{S a} = \lambda \mathbf{C a} \\ \mathbf{a^T} \cdot \mathbf{C} \cdot \mathbf{a} = \mathbf{1} \\ \mathbf{S} = \mathbf{D^T D} \end{cases} \tag{9}$$

Finally, Fitzgibbon and colleagues demonstrated that $\tilde{a_i} = \mu_i \mathbf{u_i}$ is a unique solution of the system equations in (5) (Fitzgibbon et al., 1999), where:

$$\mu_i = \sqrt{\frac{1}{\mathbf{u_i^T C u_i}}} = \sqrt{\frac{\lambda_i}{\mathbf{u_i^T S u_i}}} \tag{10}$$

Therefore, the correspondent affine anti-transformation (Maini, 2005) needs to be performed after having found the optimal solution $\tilde{a_6}$.

Some improvements to the original method (Fitzgibbon et al., 1999) have been made within the last years. One deserves particular noticing. In (Maini, 2005) it has been proposed to compute the following affine transformation to the input points before applying the Fitzgibbon's *et Al.* algorihm (Fitzgibbon et al., 1999) :

$$\tilde{x} = \frac{x - x_m}{s_x} - 1 \qquad \tilde{y} = \frac{y - y_m}{s_y} - 1 \tag{11}$$

where

$$x_m = \min_{i=1}^{N} x_i \qquad y_m = \min_{i=1}^{N} y_i \tag{12}$$

and

$$s_x = \frac{\max_{i=1}^{N} x_i - \min_{i=1}^{N} x_i}{2} \quad s_y = \frac{\max_{i=1}^{N} y_i - \min_{i=1}^{N} y_i}{2} \tag{13}$$

2.3 Least Square Fitting of Ellipses: Drawbacks and Our improvement

In 2006 Maini criticized the ill-conditioning of the scatter matrix $\mathbf{S} = \mathbf{D^T D}$ (9) and proposed an affine transformation for solving it by recentering the ellipse points within a square with side length equals to 2 (Maini, 2006). Moreover, in (Maini, 2006) it has been reported that the algorithm in (Fitzgibbon et al., 1999) has a specific source of errors not mentioned in the paper, and that this causes numerical instabilities, giving rise to the fact that the closer the data points are to the ellipse (i.e. the less noise is present), the more difficult is to locate a unique solution. This results in the impossibility of having a solution (i.e. a precise and unique ellipse curve equation) when the data points lie exactly on, or too close to, the ideal ellipse curve. In (Maini, 2005), and (Maini, 2006), a resampling procedure has been proposed, that perturbs the data points with gaussian noise in the case of they are too close to the ellipse. However, this requires an excessive computational burden. In fact, he claims that the procedure must be applied *an adequate number of times M* in order to make the algorithm effectively robust. This makes this approach it not suitable for real-time applications.

In this work we propose the application of a new pattern recognition algorithm for the least square of ellipses we presented in (Greggio et al., 2010) that improves the original formulation (Fitzgibbon et al., 1999). Our solution takes advantage of the improvements given in (Maini, 2006) in terms of the ill-

conditioning of the scattered matrix **S** (9), and implements an alternative solution to the problem of the impossibility of having a solution when the data points lie too close to the ideal ellipse curve. Moreover, our approach is twofold, because on one hand it allows to overcome this instability problem, while on the other hand it can be always applied because of its extremely low computational complexity.

3 LCSE: LEAST CONSTRAINED SQUARE-FITTING OF ELLIPSES

3.1 Instability of the exact ellipse solution

In this section we propose a technique that overcomes the previous problems. Instead of perturbing the original points with gaussian noise for many times, we decided to perturb the ellipse's polar transformation by adding a periodic symmetric function. We apply the data perturbation only if the case of not stable numerical solution, as in (Maini, 2006). However, due to our low computational burden, it does not affect the total computation sensibly, and can therefore be used any time is required. The scheme is illustrated in Fig. 1.

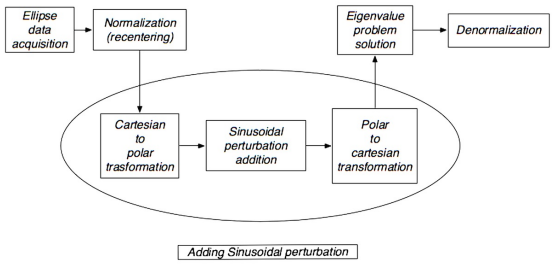


Figure 1: The RobotCub's Head. On the left image the head without the cover is shown, while in the right image the cover is shown.

The procedure is then described as follows:

- **(a)** Application of the affine transformation (Maini, 2006).
- **(b)** Transformation from the cartesian coordinates (x,y) to the polar ones (ρ,θ). The point i results:

$$\begin{aligned}
&\rho_i = \sqrt{x_i^2 + y_i^2} \\
&\theta_i = \arctan(\frac{y_i}{x_i}); \qquad with: \ x_i \geq 0, y_i \geq 0 \\
&\theta_i = \frac{\pi}{2} - \arctan(\frac{x_i}{y_i}); \quad with: \ x_i \leq 0, y_i \geq 0 \\
&\theta_i = \pi + \arctan(\frac{y_i}{x_i}); \quad with: \ x_i \leq 0, y_i \leq 0 \\
&\theta_i = \frac{3\pi}{2} - \arctan(\frac{x_i}{y_i}); \quad with: \ x_i \geq 0, y_i \leq 0
\end{aligned} \tag{14}$$

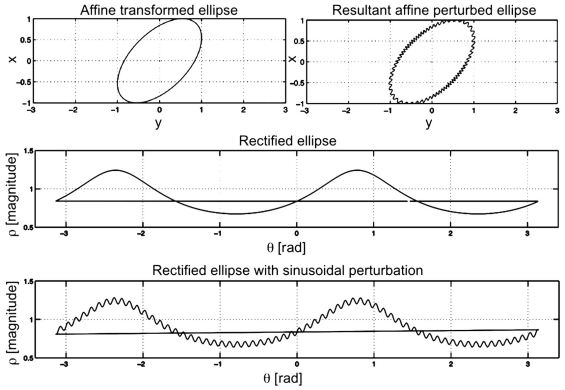


Figure 2: the original ellipse after the recentering procedure (top-left), that represented in polar coordinates (middle), the polar transformed ellipse with the sinusoidal perturbation (bottom), and the resultant perturbed ellipse (top-right).

Our aim is to move the points around their initial position, but maintaining the ellipse average over the whole polar representation period (2π) within its polar representation. Any symmetric periodic function with period taken as integer multiplier of 1 added to the original scattered data leaves the ellipse polar average unaltered. Therefore, we choose the sinusoidal function, being continuous, easy to be implemented, with zero average and infinitely derivable.

- **(c)** We choose the amplitude equals to $A = 0.001$ and the frequency equals to $f = 1000Hz$. The point i obeys to:

$$\hat{\rho}_i = \rho_i + A \cdot sin(2\pi f \theta_i); \tag{15}$$

- **(d)** When the ellipse is remapped in cartesian coordinates, its results equally slightly perturbed inside and outside its ideal curve, which is the curve that best interpolates these data. It results, for the point i:

$$\begin{aligned}
\hat{x}_i &= \hat{\rho}_i \cdot \cos(\theta_i); \\
\hat{y}_i &= \hat{\rho}_i \cdot \sin(\theta_i);
\end{aligned} \tag{16}$$

- **(e)** Now the ellipse is ready to be fitted by building the design matrix (7), and by solving the eigenvalues problem (9).
- **(f)** Finally, the affine denormalization transformation of the point **(a)** has to be applied.

Fig. 2 shows the original ellipse after the recentering procedure (top-left), that represented in polar coordinates (middle), the polar transformed ellipse with the sinusoidal perturbation (bottom), and the resultant perturbed ellipse (top-right).

3.2 Computational burden analysis

Now we analyze the computational complexity of the algorithm proposed in (Maini, 2006), and our technique. Low computational complexity means higher frame rates in real-time applications, and therefore faster control loops. This results essential essential in many actual applications (Vincze, 2001) (Kwolek, 2004) (Nelson and Khosla, 1994). Our new approach is able to eliminate the numerical instability that affects the original algorithm (Fitzgibbon et al., 1999) as (Maini, 2006) does, but greatly faster. We consider N being the number of points composing the ellipse scattered data.

Now we will describe:

a. The resampling procedure proposed in (Maini, 2006);

b. Our new approach;

c. The final comparison between these two algorithms.

→ a. Resampling procedure - (Maini, 2006): The complete procedure has been explained in (Maini, 2005). For each point a gaussian noise component is added. Therefore, this operation goes with $O(N)$. Therefore the sequence of operations 2, 3, 4, 5 has to be performed. This process goes with $O(6N)+O(42N)$, repeated for M times. Thus, the resultant complexity is $O(49MN)$. Finally, an averaging procedure through all the ellipse data has to be performed, which makes the overall process going with $O(MN)+O(49MN)=O(50MN)$.

→ b. Add sinusoidal perturbation - our new method: This adds the sinusoidal perturbation to the ellipse data after having been transformed into polar coordinates (originally, they are expressed in cartesian representation). Thus there are three operations to be performed: the first one is the transformation of all the data points from cartesian to polar representation. This takes $O(2N)$. After that, the addition of the sinusoidal perturbation takes $O(N)$ operations. Then, the polar coordinates are remapped into cartesian ones, taking $O(2N)$. Therefore, the whole operation goes with $O(5N)$.

→ c. Computational comparison and improvement. In (Maini, 2006) it has been suggested $50 \leq M \leq 200$. Moreover, in (Maini, 2006) it has been reported that EDFE performed better performances than B2AC for $M \geq 200$. However, it is clear that repeating the resampling procedure more than 200 times costs a very high computational burden. Even if M were equal to 50, our procedure is 500 times faster than (Maini, 2006). In fact, by comparing (Maini, 2006) versus our procedures for eliminating the numerical instability, i.e. the passages 8 and 9, respectively, it is possible to see that our procedure is faster of $O(50MN)/O(5N) \Rightarrow 10M = 10 \cdot 50 = 500$ times.

4 THE ICUB ROBOTIC PLATFORM

The robot is composed of 53 degrees of freedom (DOFs). Most of them are directly actuated, such as the shoulders, others are under-actuated, such as the hands (Metta et al., 2005).

In vision, the robotic head design plays an important role. Both eyes can tilt (i.e. to move simultaneously up and down), pan (i.e. to move simultaneously left and right), and verge (i.e. to converge or diverge, with respect to the vision axes). The pan movement is driven by a belt system, with the motor behind the eye ball.

An exhaustive explanation about a kinematic and a dynamic analysis for the upper body structure can be found in (Nava et al., 2008).

4.1 The ODE iCub Simulation

On the one side, the simulator information is not exhaustive, but it is a good approximation for the software debugging before using it on the real robot. On the other side, our algorithm claims to overcome the original Fitzgibbon's approach drawback of failing in detecting the ellipse when the curve lies on the ideal curve (i.e. is case of noise absence.) (Maini, 2006). It is clear that image segmentation in the real robot will never, or very seldom, produce perfect ellipses after image segmentation, due to all the imperfection within the real word (light gradients, light contrasts, color gradients, not regular object shapes, etc), therefore testing this ellipse pattern recognition algorithm to the real robot will not produce comprehensive results. Contrariwise, the simulator does not present these artifacts, or at least it limits them.

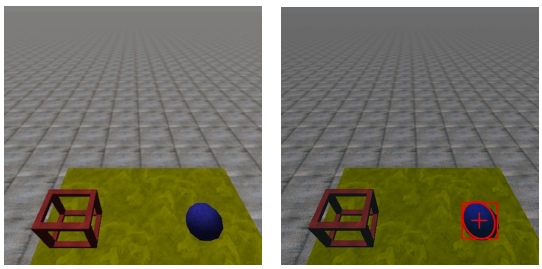
Tikhanoff *et al.* developed a completely open source simulator for the iCub (Tikhanoff et al., 2008), based entirely on the *ODE* (Open Dynamic Engine). We use this simulator in order to test our algorithms.

Fig. 4 shows a print screen of the simulator.

5 AI032-CUB: THE ROBOT CONTROLLING TOOL

5.1 The Vision Module

The vision module receives the images from the two cameras mounted on the iCub head. In order to detect the ball, and all its features, we implemented a simple but efficient image processing algorithm. We identify the ball by means of a color filter.



(a) The left camera output. (b) The object recognized within the left camera.

Figure 3: The input image, as seen by the robot within the simulator with the egocentric view (a) and the same image with the superimposition of an ellipse, drawn by using the characteristic parameters obtained by computing the LCSE (b).

For the identification of the blob corresponding to the ball, we use a *connected components* labeling algorithm. We assume the largest blob is the ball, so we look for the blob with the largest area. Subsequently, we proceeded by applying our LS technique (Greggio et al., 2010) to the found blob, in order to detect all the parameters of the curve that describes the boundary of the blob. In Fig. 3(a) the input to the left camera is presented, i.e. the experimental scenario, while in Fig. 3(b) output of the algorithm is presented.

5.2 The Motor Control Module

In addition, we implemented a tracking algorithm in a closed loop. The information received from the vision module are then processed sent to the motors of the iCub's head by means of a velocity control scheme. that directly commands the head of the robot, Then, using the forward robot's kinematics and the encoders' information we are able to reconstruct the target object's center of gravity (COG) spatial position. [1]. In Fig. 4 a screenshot is depicted, that shows an operative situation in which the simulator tracked the ball.

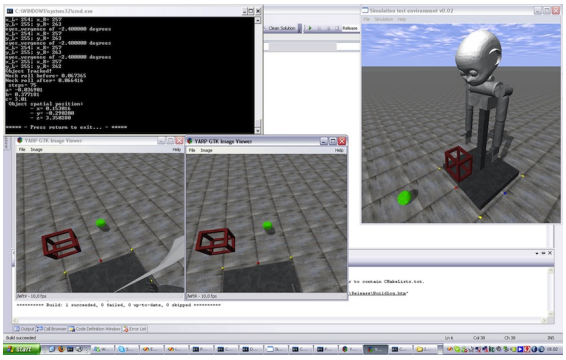


Figure 4: A screenshot depicting the moment in which the simulated robot tracked the ball position in the 3D surrounding environment. Therefore, our program uses the encoders information to triangulate the position of the centroid of the object within the simulated space.

5.3 The Kinematics Module

Then, the Denavit-Hartemberg convention for the object's COG coordinates is analyzed. Tab. III shows the parameters of the D-H symbols notation.

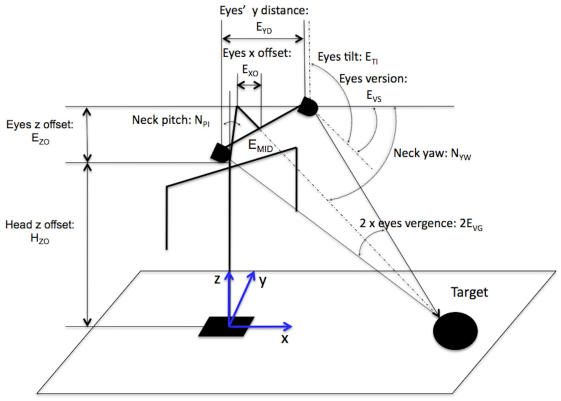


Figure 5: Schematization of the iCub's kinematics. This is not all the kinematics, of course. We focussed on the head and neck's joints.

[1] The reference system is centered on the floor plane, at the center of the pole that sustains the robot. The x axis evolves along the front of the robot, the y axis runs along the left of the robot, and the z axis evolves along its height.

Lenght [*SMU*]	Symbol	Meaning
0.35	E_{YD}	Eyes y distance: Related to the y axis, it represents the distance between the eyes along their axis
0.36	E_{XO}	Eyes x offset: Related to the x axis, it represents the distance between the neck pitch axis and the eyes axis
3.50	E_{ZO}	Eyes z offset: Related to the z axis, it represents the distance between the neck yaw axis and the eyes axes
3.01	H_{ZO}	Head z offset: Related to the z axis, it represents the distance between the origin of the reference system and the neck pitch joint

Table 1: Body kinematics lenghts.

Link	a_i	α_i	d_i	θ_i
L_1	0	$\pi/2$	h_{L1}	0
L_2	0	-$\pi/2$	0	θ_1^*
L_3	l_{L3}	$\pi/2$	h_{L3}	θ_2^*
L_4	0	-$\pi/2$	0	θ_3^*
L_5	d_0	0	0	θ_4^*

Table 2: Body kinematics Denavit-Hartenberg parameters.

Here, the angles represent:

- θ_1^*: Neck pitch - positive up
- θ_2^*: Neck yaw - positive left
- θ_3^*: Eyes tilt - positive up
- θ_4^*: Eyes version - positive left

Then, d_0 represents the target's COG distance from the eyes' middle axis point E_{MID} (see Fig. 5). This is evaluated with a simple geometrical relationship, as follows:

$$d_0 = \frac{E_{YD}}{2}\tan\left(\frac{\pi}{2} - E_{VG}\right) = \frac{E_{YD}}{2}\tan^{-1}(E_{VG}) \quad (17)$$

The target's COG coordinates $(x_{COG}, y_{COG}, z_{COG})$ are evaluated as follows:

$$\begin{bmatrix} x_{COG} \\ y_{COG} \\ z_{COG} \\ 1 \end{bmatrix} = \left[T_0^5\right] \begin{bmatrix} x_5 \\ y_5 \\ z_5 \\ 1 \end{bmatrix} \quad (18)$$

with:

$$T_0^5 = \begin{bmatrix} \cos\theta_1^*\cos\theta_2^*\cos\theta_3^*\cos\theta_4^* + & -\sin\theta_4^*\cos\theta_1^*\cos\theta_2^*\cos\theta_3^* + \\ -\cos\theta_1^*\sin\theta_2^*\sin\theta_4^* & -\cos\theta_1^*\sin\theta_2^*\cos\theta_4^* \\ \sin\theta_2^*\cos\theta_3^*\cos\theta_4^* + & \sin\theta_2^*\sin\theta_4^*\cos\theta_3^* + \\ \cos\theta_2^*\sin\theta_4^* & \cos\theta_2^*\cos\theta_4^* \\ \sin\theta_1^*\cos\theta_2^*\cos\theta_3^*\cos\theta_4^* + & -\sin\theta_1^*\sin\theta_4^*\cos\theta_2^*\cos\theta_3^* + \\ -\sin\theta_1^*\sin\theta_2^*\sin\theta_4^* & -\sin\theta_1^*\sin\theta_2^*\cos\theta_4^* \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -\sin\theta_3^*\cos\theta_1^*\cos\theta_2^* & d_0\cos\theta_1^*\cos\theta_2^*\cos\theta_3^*\cos\theta_4^* + \\ & -d_0\sin\theta_4^*\cos\theta_1^*\sin\theta_2^* + \\ & l_{L3}\cos\theta_1^*\cos\theta_2^* \\ -\sin\theta_2^*\sin\theta_3^* & -d_0\cos\theta_3^*\cos\theta_4^*\sin\theta_2^* + \\ & d_0\cos\theta_2^*\sin\theta_4^* \\ & l_{L3}\sin\theta_2^* \\ -\sin\theta_1^*\sin\theta_3^*\cos\theta_2^* + & d_0\cos\theta_2^*\cos\theta_3^*\cos\theta_4^*\sin\theta_1^* + \\ \cos\theta_1^*\cos\theta_3^* & -d_0\sin\theta_1^*\sin\theta_2^*\sin\theta_4^* \\ & l_{L3}\sin\theta_1^*\cos\theta_2^* + \\ & h_{L1}\cos\theta_1^* + h_{L1} \\ 0 & 1 \end{bmatrix}$$

6 EXPERIMENTS

We performed three types of experiments:

a. The robot has to localize a green cylinder (obtained as a section of the ball used in the experiment (b)), and having a negligible height) in front of it; the cylinder goes away along the x-axis direction at each trial;

b. The robot has to evaluate the ball's radius while an occlusion hides the object;

c. The robot has to localize a green ball in front of it.

Localization is intended in terms of 3D cartesian coordinates. At each trial the Hough transform, the B2AC, and the LCSE algorithms are used in order to evaluate the ball's center of mass (COM) within the 2D camera images. Therefore this information is triangulated with the encoders' values in order to determine the ball spatial position.

Since there is a prospective error, introduced by the spatial perspective, the ball is not seen as a 2D circle by the two camera, hence distorted, bringing about

to an artifact during these scenarios experiments that is not due to the goodness of the three tested algorithms. In order to reduce this effect we tried to isolate the perspective error by performing the experiment (a). For each scenario we performed at least 30 trials. The robot stands up and remains in the same position.

We tested both the precision in localization and the percentage of success/failure in detection.

7 RESULTS AND DISCUSSION

In the scenario *a* and *b* the error between the real and the evaluated cylinder's and ball's position is determined, while in the scenario *c* the error between the real and evaluated ball's radius is calculated.

7.1 Error Propagation Evaluation

We evaluated the error propagation for the position detection as follows. All of the terms are measured in *simulator measure unit (SMU)*. The err_{pixel} is the absolute error relative to the value of one square pixel. In order to evaluate it we referred to the known ball's radius. By knowing it (as a fixed value, i.e. 0.17 SMU) and by evaluating it at each measure we can estimate the value of a square pixel in SMU (this is the image resolution at the object distance) as the ratio between the known radius and the one estimated with each of the three algorithms considered (i.e. Hough transform, B2AC, and LCSE).

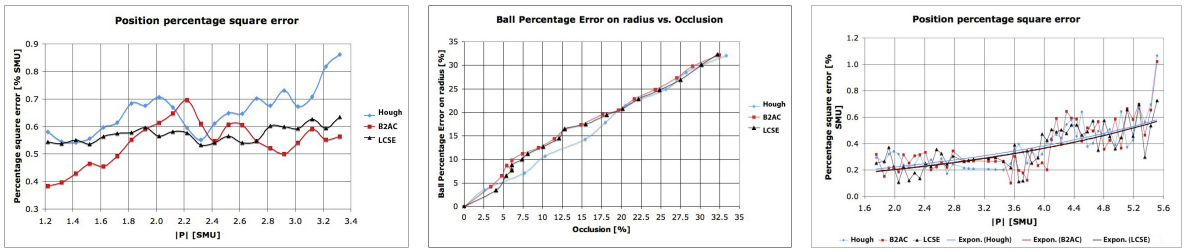
The errors of the encoders can be considered negligible within the simulator. Since there is no documentation on the encoders' resolution within the simulator, we considered the accuracy of their information approximated to their last digit, which is the forth one (therefore negligible). Finally the errors due robot's lengths need to be considered. Again, there is no information about the error the lengths of the robot's parts have been expressed with. Therefore, in order to fix their accuracy we analyzed the simulator's source code. So far, we found that the lengths of the robot's parts were expressed with the second digit of approximation. Hence, we approximated them as 0.01 SMU.

7.2 Scenarios' Evaluation

As a first result, Fig. 6(a) shows the results of the scenario *a*. With exception for the quadratic error within the range $[2.15-2.35]$, the Hough Transform gives rise to the highest error. Both the ellipse fitting algorithms cause lower error. Specifically, the B2AC algorithm is the most precise in terms of quadratic error, within the ranges $[1.2-1.9]$, and $[2.7-3.4]$. However, it presents several discontinuities, and a total non-linear characteristic emerges, even following the Hough Transform approach's error (but keeping almost lowest). The LCSE seems to be not the lowest error prone, but it has a very regular characteristic of the function of the distance. By increasing the distance it fits the B2AC error curve well, while keeping little bit higher.

The experiment of the scenario *b* shows a great linearity between the occlusion of the ball and the error on its radius evaluation. Fig. 6(b) illustrates the results of this experiment. Here, the Hough Transform gets better results within the range [5 % - 20 %] of occlusion, where P_r is the residual number of pixels, and P_t is the total number of target object pixels, determined with no occlusion), then almost superimposing with the other two approaches after he 20 % of occlusion. The characteristic is quite linear for all the techniques adopted, with the exception of the cited range, in terms of a slight decrease from the linear ideal line for the Hough Transform and a slight increment for both ellipse detection approaches. Subsequently, the error introduced by spatial perspective is mapped as a function of the object's distance from the eyes axis midpoint. We isolate the perspective error by comparing the absolute error obtained within the tests in the scenario *a* and *c*, as absolute errors. It is worth noting that in order to compare these errors, the cylinder and and the ball we used have the same radius (0.17 SMU) within the trials. Therefore the percentage perspective error has been evaluated as the ratio between the absolute perspective error and the module of the distance between the eyes axis midpoint and the object. The two ellipse recognition techniques are more sensitive than the Hough Transform to the spatial perspective.

Finally, the scenario *c* is discussed. In spite of the fact that the ellipse detection approaches give rise to a bigger spatial perspective error than the Hough Transform, the precision given within the overall system is superior than the one obtained with the Hough Transform. In 6(c) this is showed. We did not filter the results, in order to keep them as natural as possible. By acting in this way, the noise affects the trend of the curves most. Therefore, we inserted three trend lines (one for each technique, each of them with exponential characteristic) in order to evidence the most fruiting approach. Here, the B2AC's and the LCSE's trend lines appear superimpose, so that it is not possible distinguishing them from each other. However, the Hough Transform's trend line shows of this technique is the most error prone for balls' spatial position detection in image processing.



(a) (b) (c)

Figure 6: Cylinder's position error as function of the distance while considering the perspective effect negligible (a), ball Percentage Error on radius, in % of the radius value (b), and percentage square error, measured in % of the simulator measure unit (c).

7.3 Some aspects of the Hough transform

It is worth mentioning, that the Hough transform depends on some parameters in order to be well set. Of most importance is the *accumulator threshold at the center detection stage* (*ATCDS*). The smaller its value is, the more false circles may be detected, but the higher it is, the less circles may be detected. We tested that the smaller the *ATCDS* is, the more instability is produced on the Hough computation. In the first case, an *ATCDS* lower setting causes that other curvatures, e.g. artifacts on the ball's border caused for instance by inaccuracies of the color filtering, may be detected as additional objects. These inaccuracies, in fact, can be interpreted as small circles by the Hough transform, giving rise to wrong results. However, setting *ATCDS* too high can cause the opposite problem. In our case there was only one circular object within the image (i.e. the ball), but wrong setting values (e.g. *ATCDS* too high) were sufficient to not detect it. This was true even if the detected ball was the only color blob after the erosion of the image, and even if it was substantially big too not be negligible (i.e. not to be misinterpreted as a color artifact). This means that one has to find the right value in every condition, in terms of the best compromise between sensibility (intended as the capability of detecting all the possible circles in the image) on one hand, and the stability (intended as the accuracy of the algorithm of not detecting false circles) on the other hand. Therefore, we looked for the biggest value that is able to perform all the experiments without avoiding the detection of the circles and maintaining the best possible stability. In our experiments we set $ATCDS = 2$. In table **??** there are some *ATCDS* values: each one represents the maximum worth able to perform the experiments at some fixed occlusion and distance values.

However, both B2AC and LCSE algorithms do not present a similar drawback, permitting them to be used in any situation without any previous setting. This can be considered a great advantage, since they do not require any *a priori* information of the scene to be analyzed. This is twofold, because allows not only to build a robust and $scene-independent$ technique, but also it fits with the concept of cognitive robotics perfectly.

ATCDS	3	2.6	2.3	2.1	2	2
Occlusion [%]	5	10	15	20	25	30
ATCDS	3.4	3	2.6	2.4	2.1	2
Distance [SMU]	1.2	1.6	2	2.4	2.8	3.2

Table 3: Maximum value for the accumulator threshold for getting stability in our experiments.

8 CONCLUSIONS

In this work we presented the first implementation of the LCSE ellipse square fitting algorithm, and we applied it to a humanoid robotics platform. Moreover, we implemented a real-time tracking algorithm to localize an object with the Robot's stereo vision, and subsequently we used it to determine the 3D position of the object's centroid in the environment. We compared the Hough Transform, the B2AC, and the LCSE performances in terms of localization precision and failure in detection in presence of induced artifacts (such as the ball occlusion by another object) and as function of the distance of the target. We found that the B2AC and LCSE give rise to overall more precise results than the Hough Transform. In the near future we plan to apply our techniques to the real iCub robotics platform, in order to compare and validate our results with the real robot, and not only with the ODE simulator. Then, we will make our code freely available within the iCub repository.

ACKNOWLEDGEMENTS

We thank Dr. Andrea Cini for his contribution in writing the DH matrix for the robot forward kinematics.

This work was supported by the European Commission, Project IST-004370 RobotCub and FP7-231640 Handle, and by the Portuguese Government - Fundação para a Ciência e Tecnologia (ISR/IST pluriannual funding) through the PIDDAC program funds and through project BIO-LOOK, PTDC / EEA-ACR / 71032 / 2006.

REFERENCES

Carpin, S., Lewis, M., Wang, J., Balarkirsky, S., and Scrapper, C. (2007). Usarsim: a robot simulator for research and education. In *IEEE International Conference on Robotics and Automation*.

Deniz, O., Castrillon, M., Lorenzo, J., Guerra, C., Hernandez, D., and Hernandez, M. (2002). Casimiro: A robot head for human-computer interaction. *Robot and Human Interactive Communication. Proceedings. 11th IEEE International Workshop on*, (ISBN: 0-7803-7545-9):319– 324.

DeSouza, G. N. and Kak, A. C. (2002). Vision for mobile robot navigation: A survey. *IEEE Transaction PAMI*, 24:237–267.

Fitzgibbon, A., Pilu, M., and Fisher, R. (1999). Direct least square fitting of ellipses. *IEEE Trans. PAMI*, 21:476–480.

Gander, W., Golub, G., and Strebel, R. (1994). Fitting of circles and ellipses least squares solution. Technical report tr-217, Institut für Wissenschaftliches Rechen, ETH, Zurich, Switzerland.

Greggio, N., Bernardino, A., Laschi, C., Santos-Victor, J., and Dario, P. (2010). An algorithm for the least square-fitting of ellipses. *IEEE 22th International Conference on Tools with Artificial Intelligence (ICTAI 2010), Arras, France*.

Greggio, N., Silvestri, G., Menegatti, E., and Pagello, E. (2009). Simulation of small humanoid robots for soccer domain. *Journal of The Franklin Institute - Engineering and Applied Mathematics*, 346(5):500–519.

Kanatani, K. (1994). Statistical bias of conic fitting and renormalization. *IEEE Trans. Patt. Anal. Mach. Intell.*, 16:320–326.

Kwolek, B. (2004). Real-time head tracker using color, stereovision and ellipse fitting in a particle filter. *INFORMATICA*, 15(2):219–230.

Leavers, V. F. (1992). Shape detection in computer vision using the hough transform. *Springer-Verlag*.

Maini, E. S. (2005). Robust ellipse-specific fitting for real-time machine vision. *BVAI*, pages 318–327.

Maini, E. S. (2006). Enhanced direct least square fitting of ellipses. *IJPRAI*, 20(6):939–954.

Metta, G., Sandini, G., Vernon, D., Caldwell, D., Tsagarakis, N., Beira, R., Santos-Victor, J., Ijspeert, A., Righetti, L., Cappiello, G., Stellin, G., and Becchi, F. (2005). The robotcub project - an open framework for research in embodied cognition. *Humanoids Workshop, IEEE –RAS International Conference on Humanoid Robots*.

Nava, N., Tikhanoff, V., Metta, G., and Sandini, G. (2008). Kinematic and dynamic simulations for the design of robocub upper-body structure. *ESDA*.

Nelson, B. J. and Khosla, P. K. (1994). The resolvability ellipsoid for visual servoing. *Proc. of the 1994 Conf. on Computer Vision and Pattern Recognition (CVPR94)*.

Rosin, P. L. and West, G. A. (1995). Non parametric segmentation of curves into various representations. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17:140–153.

Sandini, G., Metta, G., and Vernon, D. (2007). The icub cognitive humanoid robot: An open-system research platform for enactive cognition. *50 Years of AI, M. Lungarella et al. (Eds.), Festschrift, LNAI 4850*, pages 359–370.

Teutsch, C., Berndt, D., Trostmann, E., and Weber, M. (2006). Real-time detection of elliptic shapes for automated object. *Machine Vision Applications in Industrial Inspection XIV. Edited by Meriaudeau, Fabrice; Niel, Kurt S. Proceedings of the SPIE*, 6070:171–179.

Tikhanoff, V., Fitzpatrick, P., Nori, F., Natale, L., Metta, G., and Cangelosi, A. (2008). The icub humanoid robot simulator. *International Conference on Intelligent RObots and Systems IROS*, Nice, France.

Vamossy, Z., Moolnar, A., Hirschberg, P., Toth, A., and Mathe, B. (2003). Mobile robot navigation projects at bmf nik. *International Conference in Memoriam John von Neumann*.

Vincze, M. (2001). Robust tracking of ellipses at frame rate. *Pattern Recognition*, 34:487–498.

Vincze, M., Ayromlou, M., and Zillich, M. (2000). Fast tracking of ellipses using edge-projected integration of cues. *Pattern Recognition. Proceedings. 15th International Conference on*, 4(ISBN: 0-7695-0750-6):72–75.

Wang, J., Lewis, M., and Gennari, J. (2003). Usar: A game-based simulation for teleoperation. *Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society, Denver, CO - Oct. 13-17*, pages 493–497.

Yuen, H. K., Illingworth, J., and Kittler, J. (1989). Detecting partially occluded ellipses using the hough transform. *Image Vision and Computing*, 7(1):31–37.

Zhang, Z. (1997). Parameter estimation techniques: a tutorial with application to conic fitting. *Image Vision and Computing*, 15:59–76.