

## POSE ESTIMATION OF A HUMANOID ROBOT USING IMAGES FROM A MOBILE EXTERNAL CAMERA

Marcelo Borges Nogueira\*  
Adelardo A. D. Medeiros\* Pablo J. Alsina\*

\* *Universidade Federal do Rio Grande do Norte*  
*Departamento de Engenharia de Computação e Automação*  
*UFRN-CT-DCA – Campus Universitário – 59072-970*  
*Natal-RN Brazil*

**Abstract:** This paper describes an ongoing project to move a simple humanoid robot, without an advanced embedded electronics. We used another wheeled mobile robot which already has a locating system and is equipped with a camera. We will locate the humanoid in the environment based on images, and then take the necessary actions to move it. In the future, we will also move the robot with the camera, so it will take good images of the humanoid.

**Keywords:** multi-robot cooperation, relative positioning, humanoid, pose estimation

### 1. INTRODUCTION

One of the main task of a mobile robot navigation system is to determine the robots position and orientation in its work environment.

Some of the most utilized locating sensor are encoders, sonars, cameras, compass and laser range-finders. One of the most popular and cheap is odometry. However, is is not a simple task to use odometry with legged robots, since they do not have a fixed length step. Besides, the sliding is usually a lot bigger than with wheeled robots. That is why locating humanoid robots suggests other methods. Some of these methods are very expensive (laser range-finder and GPS) and others give us sparse information (sonar)(Albert and Connell, 2004). Hence, a visual navigation system becomes attractive due to its low cost and the high processing power of current computers.

One possibility would be to equip the humanoid with one or more cameras. However this would lead to a big humanoid or to an expensive em-

bedded electronics. Besides, the locating system (hence also the camera) would have to be very accurate, since this would be the only method to locate the robot on the environment.

To build or acquire a big humanoid (adult size) is a difficult, long and expensive task. For this reason, we chose to work with a small humanoid robot, remote controlled and without an advanced embedded electronics. The humanoid robot acquired was the *Robosapien* (Wee, 2004), shown if Figure 1, from *Wow Wee*. Therefore, we chose a strategy that combines odometry and vision.

In the proposed system, we will use a second mobile wheeled robot (still under construction), equipped with a camera (Figure 2). Its function will be to process the data acquired by the camera, to calculate the relative position of the humanoid and, knowing his own position, to calculate the absolute position of the humanoid. Based on this information, we can determine which way the humanoid robot and the wheeled robot should move. After this, the wheeled robot will send

commands to the humanoid robot via infra-red. The project is in a initial phase, and will study camera calibration techniques and the humanoid movements.



Fig. 1. The Robosapien humanoid robot

We suppose that the wheeled robot (which we will call camera from now on) already have a locating system and knows how to navigate in its work environment (given the current position, the camera knows which actions to take so to reach the target position). This way, knowing the position of the humanoid relative to the camera and supposing the position of the camera in the world is known, we can calculate the position of the humanoid robot in the world. This makes possible for us to control the humanoid. To do so, we will use a heuristic technique detailed in Section 5.

The main goal is to correctly move the humanoid around the environment. However, is is also necessary to move the camera, so it will follow the humanoid through the work environment and take good images of him. The technique used to move the camera will be discussed in Section 6.

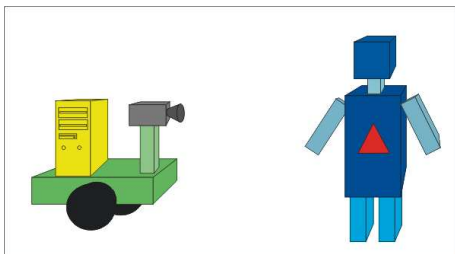


Fig. 2. Proposed system: humanoid robot observed by external mobile camera

To obtain spatial (3D) information from a 2D image it is necessary a camera calibration. In Section 2 this process will be detailed. Experiments, results and conclusions will be presented in Sections 7 and 8 respectively.

## 2. CAMERA CALIBRATION

Camera calibration is to calculate a set of values relative to the camera that makes possible to interpret images acquired by this camera. It can be divided in two steps: calculating its interior orientation parameters, or intrinsic calibration, and calculating it's exterior orientation parameters, or extrinsic calibration.

The intrinsic calibration is an off-line procedure, that is, it is made before the control system of the robot is working. Besides, it needs to be done just once, since the interior parameters of a camera (usually) do not change over time. Hence, time constraints are not critical in this calibration phase. On the other hand, the extrinsic calibration phase has a serious time restriction, since it must run at each control sampling time.

Suppose  $\{R\}$  the world's 3D coordinate system, in centimeters,  $\{C\}$  the camera's 3D coordinate system, in centimeters,  $\{I\}$  the image's 2D coordinate system, in centimeters and  $\{F\}$  the image's 2D coordinate system, in pixels. Using the pinhole camera model we can find: (Zhang, 2000):

$$C_z \begin{bmatrix} FP \\ 1 \end{bmatrix} = A \cdot {}^C T_R \cdot [RP \ 1]^T \quad (1)$$

where  $FP = [F_x \ F_y]^T$  is a point relative to  $\{F\}$ ,  $RP = [R_x \ R_y \ R_z]^T$  is a point relative to  $\{R\}$ ,  $C_z$  is the z coordinate of a point relative to  $\{C\}$ ,  ${}^C T_R$  is the transformation matrix between  $\{C\}$  and  $\{R\}$  and  $A$  is the intrinsic parameters matrix, given by:

$$A = \begin{bmatrix} \alpha & \gamma & F_{x_0} \\ 0 & \beta & F_{y_0} \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where  $(F_{x_0}, F_{y_0})$  are the image center coordinates, in pixels,  $\alpha$  and  $\beta$  are respectively ratios between the scale factors in the images axes  $x$  and  $y$  and the focal distance of the camera and  $\gamma$  is the distortion of the angle formed by the axes.

### 2.1 Intrinsic Camera Calibration

Intrinsic camera calibration means finding the intrinsic properties of the camera (matrix  $A$  show in equation 2). There are several methods in the literature about intrinsic camera calibration. An analysis of these methods is made by Nogueira (Nogueira, 2005).

Many calibration methods include observing one or more images which contains objects, in which we can find target points. Some methods do not need the object to be known, neither the displacement made by the camera between the observations (Faugeras *et al.*, 1992; Longuet-Higgins, 1981). Others need to known at least the observed object. Besides, the target points in the object

can or can not be coplanar, depending on the method. After an analysis, we decided that the method that best suits the proposed system is the one proposed by Zhang (Zhang, 2000) due to his flexibility, robustness and low computational cost.

In this method, it is necessary that the camera observes a flat pattern of known measures from at least two different positions. However, it is not needed that we know the displacement made by the camera between the observations. The coordinate system of the pattern must be chosen so that the  $z$  coordinate of each point on the pattern will be zero. An example of such a pattern can be observed in Figure 6. Based on these information, after finding the target points on the pattern (the method we used to do this will be discussed in Section 3), the Zhang method will give us an initial guess of the intrinsic parameters matrix. This initial guess has an error, due to non-linear distortions caused by the lenses.

From this initial guess, we can conduct a linear optimization so to reduce the error between the observed points and the ones calculated by the Equation 1.

## 2.2 Extrinsic Camera Calibration

The extrinsic parameters are the position and orientation of the camera relative to the world's system coordinate (matrix  ${}^C T_R$  in Equation 1). For this reason, this phase is also called pose estimation.

There are several different ways to solve this problem (Nogueira, 2005). In this paper we will discuss the so called PnP, or Perspective- $n$ -Point methods, which consist in calculating the pose of an object using  $n$  target points belonging to the object and that appear in an image of this object.

To solve the PnP problem we need at least 3 target points. However, in this case, there will be multiple solutions (as much as 8), what demands a further processing to choose one of these solutions (Haralick *et al.*, 1991). To find only one solution we need at least 4 target points. According to the way it is solved, the P4P can be classified in: solution by optimization and analytic solution. Among the existing P4P methods, we chose the one proposed by Kamata *et al.* (Kamata *et al.*, 1992).

The P4P (*Perspective Four-Points problem*) method proposed by Kamata has the advantage of having a analytic solution. This is a very important feature, due to the time restriction in this phase.

Kamata's method consists of, from four coplanar points known relative to  $\{R\}$ :  ${}^R P_0$  to  ${}^R P_3$ , and the corresponding 2D image points  ${}^F P_0$  to  ${}^F P_3$

(the method to detect these points in the image will be shown in Section 4), calculate the matrix  ${}^C T_R$ , introducing some intermediate coordinate systems.

## 3. EXTRACTING THE TARGET POINTS FOR THE INTRINSIC CALIBRATION

To use Zhang's method, we must detect at least 6 points in the image. However, to minimize the error, it is advisable to use more points.

To improve system precision, instead of directly detecting those points, we will look for straight lines, which intersections will give us the needed points. To calculate the straight lines we used the classical Hough algorithm.

The Hough transform is a method to detect a class of geometric known forms, which can be represented as a parametric curve, (like straight lines, polynomials, circles, etc.), in an image (Hough, 1962). It has a high computational cost, but this is not a problem, since this phase does not demand a real time algorithm.

Figure 6 shows the pattern we used: a set of 18 straight lines giving a total of 81 points.

## 4. EXTRACTING THE TARGET POINTS FOR THE EXTRINSIC CALIBRATION

We will use the same strategy we used to extract the target points for the intrinsic calibration, that is, we will use lines which intersection will give us the points. Since the P4P method needs 4 points, we will use a quadrilateral mark (in this case, a rhombus), which was placed on the humanoid robot, as shown in Figure 7. The intersection of its edges will give us its vertices. We could have used the Hough algorithm once again to calculate the rhombus edges, but this phase is executed at each sampling time and requires a faster response. Therefore, we developed a faster method to extract the vertices of a rhombus.

### 4.1 Detecting Rhombus Vertices

Supposing the rhombus has a distinguishable color when compared to the other elements in the image, it is possible to quickly distinguish a pixel belonging to its edges. However, because of several factors (illumination, reflection of the material of the rhombus, noise, etc...) the detected points may not be exactly on its edges. To reduce influence of errors, we use a set of points to calculate the best straight line (in the least squares sense) that represents each edge of the rhombus.

Table 1. Detecting points that belongs to the rhombus edges

```

(1) Search for a initial pixel  $P_0$  in the image that belongs
    to the rhombus
// Right search
(2)  $k \leftarrow 0; M \leftarrow -1$ 
(3) While the pixel  $P_k$  belongs to the rhombus:
    (a) From  $P_k$ , search upward and downward for the
        last pixels belonging to the triangle (points  $U_k$ 
        and  $D_k$ ).
    (b) If  $\|U_k - D_k\| < \text{MIN\_LEN}$  break While
    (c) Calculate  $P_{k+1} \leftarrow \frac{U_k - D_k}{2} + [\Delta x \ 0]^T$ 
    (d)  $M \leftarrow k; k \leftarrow k + 1$ 
// Left search
(4)  $k \leftarrow 0; N \leftarrow +1$ 
(5) While the pixel  $P_k$  belongs to the rhombus:
    (a) From  $P_k$ , search for  $U_k$  and  $D_k$ .
    (b) Se  $\|U_k - D_k\| < \text{MIN\_LEN}$  break While
    (c) Calculate  $P_{k-1} \leftarrow \frac{U_k - D_k}{2} - [\Delta x \ 0]^T$ 
    (d)  $N \leftarrow k; k \leftarrow k - 1$ 

```

This way, initially, starting from a point belonging to the rhombus ( $P_0$ ) we find a set of points belonging to its edges ( $U_i$  e  $D_i$ ). This process is presented in Table 1 and illustrated in Figure 3.

We calculate then two straight lines for the  $U_i$  points:  $L_L^U$ , which goes from the left to the center; and  $L_R^U$ , which goes from the right to the center. Similarly, we calculate the straight lines  $L_L^D$  and  $L_R^D$  from the  $D_i$  points. Such straight lines are calculated so to minimize the quadratic error to the set of points associated with each one of them. The way each point is associated to a straight line is described in Table 2. Once we have the straight lines, their intersection will give us the four points needed.

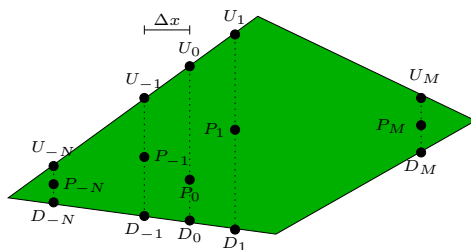


Fig. 3. Detecting points that belongs to the rhombus edges

## 5. HUMANOID MOVEMENTS

The humanoid is controlled by infra-red, and has the following motion commands: walk forward, walk backward, turn left and turn right. It has a static walk, that is, its center of gravity always remains at the middle of the foot of the robot.

Due to movement restriction and the difficulty to precisely control the position of the humanoid, we decided to use a heuristic control technique.

Table 2. Detecting lines of rhombus edges

```

// Up
(1) Initialize  $L_L^U$  with the coefficients determined by the
    points  $U_{-N}$  and  $U_{-(N-1)}$  and  $L_R^U$ , with  $U_M$  and
     $U_{M-1}$ .
(2)  $i \leftarrow -(N-2)$ 
(3) While  $i \leq M-2$ :
    (a) Calculate the distances  $e_L^U$  and  $e_R^U$ , the distances
        between the point  $U_i$  and  $L_L^U$  and  $L_R^U$ , respec-
        tively.
    (b) If  $e_L^U < e_R^U$ , recalculate the coefficients of  $L_L^U$ ,
        considering  $U_i$  belongs to it; else, recalculate the
        coefficients of  $L_R^U$ .
    (c)  $i \leftarrow i + 1$ 
// Down
(4) Do a similar procedure (steps 1 to 3) to determine
     $L_L^D$  and  $L_R^D$  from the  $D_i$  points.

```

The first step is to calculate a trajectory, given the initial point  $P_I$  and a final point  $P_F$  of the humanoid.

The control steps will be: First, calculate the error  $e$  between the humanoid position and the desired robot position and the angle  $\alpha$  between the vector from the robot to  $P_F$  and the  $z$  axis of the humanoid ( $z_R$ ) (Figure 4). If  $\alpha$  is greater than a certain value  $k$ , we command the robot to turn so to make  $\alpha$  close to zero. Next, if  $e$  is greater than a certain value  $l$ , the robot is commanded to walk forward.

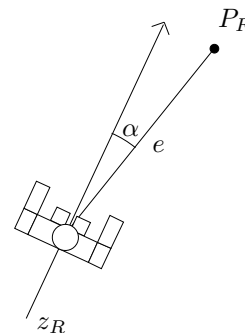


Fig. 4. Humanoid movements

## 6. CAMERA MOVEMENTS

The camera movements are totally dependent of the humanoid movements, because the camera has to follow the humanoid so it will take good pictures of the mark attached to the humanoid. To do this, we have to command the camera to a position that is distant  $d$  from the humanoid and from where the camera aims at the humanoid (the center of the rhombus is at the center of the image). If the  $x$  axis of the humanoid ( $x_R$ ) reaches a critical angle  $\theta$  with the  $z$  axis of the camera ( $z_C$ ) (Figure 5), what would not give a good picture of the rhombus, the humanoid is commanded to stop and the camera is commanded

to the position  $P_D$  in Figure 5, which is distant  $d$  from the humanoid and in a way that  $\theta$  is  $90^\circ$ . This leads to a non-continuous movements, but if the generated trajectory is sufficient smooth, this will not happen frequently.

A especial case that should be observed is when the desired camera position causes a collision with some object. In this case, we must sacrifice temporarily the visibility of the humanoid and place the camera closest to its desired position without causing a collision.

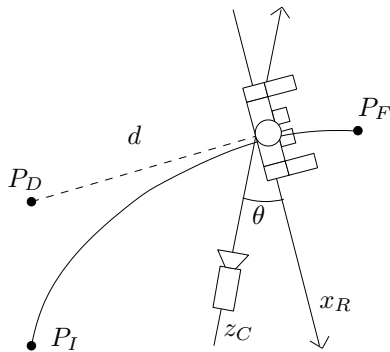


Fig. 5. Critical angle situation

## 7. EXPERIMENTS AND RESULTS

Lets analyze first the camera calibration process. Figure 6 shows the Hough algorithm applied on a pattern to detect the target points needed to calibrate the intrinsic parameters of a camera. Figure 7 shows the result of the developed method to detect rhombus vertices to calibrate the extrinsic parameters of a camera.

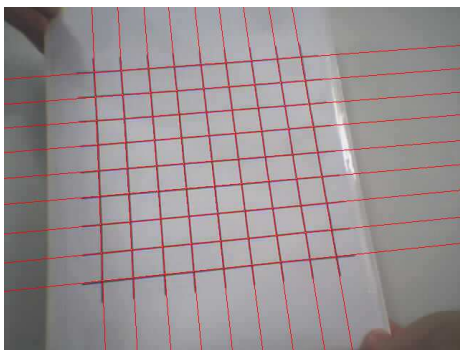


Fig. 6. Extracting target points for intrinsic camera calibration

In the intrinsic camera calibration using 4 different images of the pattern shown in figure 6, we found the following intrinsic parameters matrix  $A$ :

$$A = \begin{bmatrix} 757,25 & 2,06 & 389,67 \\ 0 & 754,24 & 260,5 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

To validate the intrinsic parameters found and, at the same time, the extrinsic calibration process,

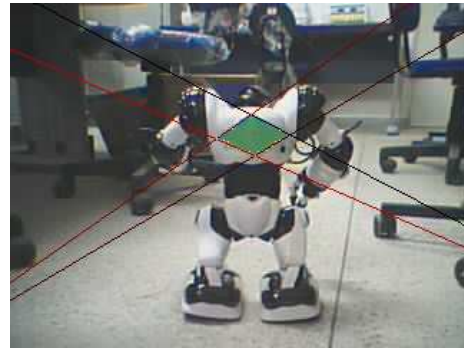


Fig. 7. Extracting target points for extrinsic camera calibration

we used the calculated matrix  $A$  to find the position of the camera using 4 rhombuses showed in Figure 8 which position are known relative to the world. The result for each rhombus should be all the same. The result for each rhombus (in centimeters and degrees) and the mean error between the observations (in centimeters) are shown in Table 3. Notice that the errors in the  $x$  and  $y$  axes and Roll and Yaw angle are small. The errors in the  $z$  axe and in the Pitch angle are higher, but still acceptable given the nature of the problem.

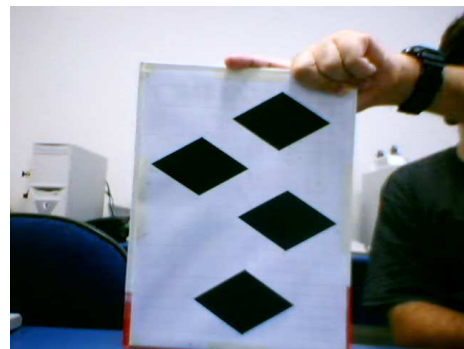


Fig. 8. Rhombuses used for validating intrinsic and extrinsic calibration

	Rho. 1	Rho. 2	Rho 3	Rho. 4	Error
$x$	-9,808	-9,838	-10,308	-9,93	0,265
$y$	18,059	17,428	17,235	16,962	0,580
$z$	36,87	38,114	37,532	39,515	1,419
Roll	-1,682	3,583	-1,433	-1,5	2,643
Pitch	-5,253	-2,782	7,878	-4,524	6,855
Yaw	0,91	2,536	2,261	2,525	0,857

Table 3. Calibration results for figure 8

To analyze the humanoid heuristic control method we commanded it to a final position  $(x, y, z) = (0, 4.7, 60)$  from the start position  $(0, 4.7, 29)$  and Pitch=0. The  $z$  axis position of the humanoid during the experiment can be seen in figure 9. The final position achieved by the humanoid was  $(3.3, 4.8, 58.2)$ . In this experiment, the wheeled robot did not move, since this functionality will be implemented later.

The bigger the rhombus appears in the image, the better are the results of the pose estimation.

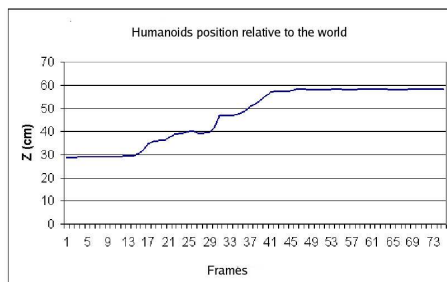


Fig. 9.  $z$  position of the humanoid relative to the world

This tells us that the camera should be as close as possible to the humanoid so to fit the rhombus image with its vertex reaching the boundaries of the whole image. However, if the camera is too close, one or more vertices of the rhombus can be momentarily out of the image. Besides, the camera should keep a certain distance from the humanoid so it can freely move in the environment. Therefore, we have to calculate the optimal distance  $d$  that the camera should keep from the humanoid.

Figure 10 shows the standard deviation between several estimations of the  $z$  position of the camera in function of  $r$ , which is the ratio between rhombus area and total figure area. Notice that when  $r$  is bigger than 5%, the deviation varies just a little, but when  $r$  becomes smaller than 5% the deviation increases rapidly. This shows us that we should keep  $r$  less than 5% so the measurements are coherent. This ratio corresponds to the distance  $d = 25\text{cm}$ .

To determine the critical angle we did a similar process used to calculate the distance  $d$ . Positioning the camera 25cm from the humanoid and calculating the standard deviation for several measures (pose estimation) against several different pitch angles, we can determine the critical angle  $\theta$  which keeps the standard deviation less than 5%, which we found to be 50 degrees.

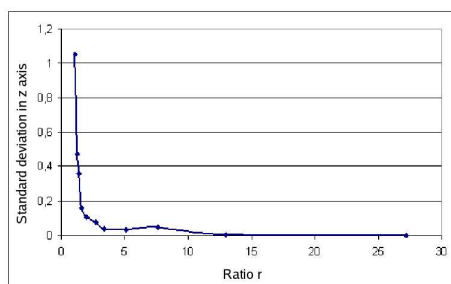


Fig. 10. Standard deviation in humanoid estimated  $z$  position against  $r$

## 8. CONCLUSIONS

Zhang's intrinsic calibration method showed to be appropriate to the system, giving a flexible, fast

and robust calibration. The Hough algorithm was very precise detecting the straight lines

Since it has an analytic solution, the P4P proposed by Kamata gives us a constant time response, ideal for real time applications. The proposed method to extract the vertices of the rhombus showed to be very fast and robust, even with low cost web cameras and illumination variation.

The heuristic method to control the humanoid was appropriate to its limited kind of motion, and we made an analysis of the position that the camera should stay to minimize the error in the pose estimation process.

The presented system can be used in applications where, besides a humanoid robot, is also available a second mobile robot with a precise locating system and a camera. It allows experiments with low cost and simple embedded electronics humanoid robots.

## REFERENCES

- Albert, M. E. and J.H. Connell (2004). Visual rotation detection and estimation for mobile robot navigation. In: *IEEE International Conference on Robotics and Automation*.
- Faugeras, O. D., Q.-T. Luong and S.J. Maybank (1992). Camera self-calibration: Theory and experiments. In: *European Conference on Computer Vision*. pp. 321–334.
- Haralick, Robert M., Chung-Nan Lee, Karsten Ottenberg and Michael Nölle (1991). Analysis and solutions of the three point perspective pose estimation problem. In: *CVPR - IEEE Conference on Computer Vision and Pattern Recognition*. Maui, HA, USA.
- Hough, P. V. C. (1962). Methods and means for recognizing complex patterns.
- Kamata, Sei-ichiro, Richard O. Eason, Masafumi Tsuji and Eiji Kawaguchi (1992). A camera calibration using 4 point-targets. In: *11th IAPR - International Conference on Pattern Recognition*. Hague, Netherlands.
- Longuet-Higgins, H., C (1981). A computer algorithm for reconstructing a scene from two projections. *Nature* pp. 133–135.
- Nogueira, Marcelo B. (2005). Posicionamento e movimentação de um robô humanóide utilizando imagens de uma câmera móvel externa. Master's thesis. UFRN, Natal- RN.
- Wee, Wow (2004). [www.robosapienonline.com](http://www.robosapienonline.com).
- Zhang, Z (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 1330–1334.