Tracking based on Hue-Saturation Features with a Miniaturized Active Vision System

J. A. Corrales, P. Gil, F. A. Candelas, F. Torres*

**Automatics, Robotics and Computer Vision Group.*
Physics, Systems Engineering and Signal Theory Department, Univesity of Alicante.
03690 San Vicente del Raspeig, Alicante, Spain.
(e-mail:[jcorrales, Pablo.Gil, Francisco.Candelas, Fernando.Torres]@ua.es)

Abstract: This paper presents a miniaturized active vision system for visual tracking. One of the main problems in visual tracking is the autonomy and manageability of the system to be mounted on robotic structures, such as mobile and manipulator robots. The proposed active vision system has been built using a motorized platform characterized by its low price, lightness, small dimensions and wireless control. It is interesting for visual tracking applications where constraints of size and weight must be considered. In our mini active vision system, a tracking method based on CamShift has been implemented. The novelty of our tracker, in comparison with CamShift, is its ability to automatically combine a hue distance component and a saturation component from the HSV colour model in order to track objects in dynamic backgrounds with similar hue values.

1. INTRODUCTION

The tracking of objects from the perception and understanding of an environment is an essential problem in computer vision. Techniques for tracking are useful for a wide range of applications and are currently applied successfully in many areas such as object recognition, autonomous navigation, visual surveillance and camera motion estimation in robotic manipulators.

In general, tracking techniques are based on the search of some features or descriptors in the image which identify the tracked object and differentiate it from the rest of the scene. Some examples of these descriptors are: interest point features (Saunier and Sayed, 2006), colour information (Pérez*, et al.*, 2002), appearance models to define the target and other type of features which are presented in the environment and are projected in the images.

Whenever a computer vision system consists only of one camera without movement, the target to be tracked cannot be detected in many cases due to occlusions or because the target abandons the field of view of the sensor. In these cases, several solutions have been considered in the literature. Thus, there are tracking systems composed of multiple cameras without movements where each camera covers different zones of the same environment, composed of a motorized pan/tilt camera (Mayol*, et al.*, 2002), stereoroscopic-pairs (Bernardino and Santos-Victor, 1999) or fixed cameras mounted at the end-effector of a robotic manipulator for visual servoing (García*, et al.*, 2007).

In the last years, several tracking algorithms based on colour have been proposed. The most commonly known are: CamShift (Bradski, 1998), which was designed for close range face tracking with cameras without movement but it has also been modified for other applications (Malis and Benhimane, 2005); MeanShift (Comaniciu and Meer, 2002); or more recently ABCShift (Stolkin*, et al.*, 2008), which describes an adaptive background model based on the CamShift algorithm. Nowadays, research in tracking has being focused on adaptive methods which are robust to changes not only in the background (Stolkin*, et al.*, 2008) but also in the illumination of the scene (Jacquot*, et al.*, 2005).

This paper describes the design of a novel mini active vision system which is able to track an object with movement in a dynamic scene. This system is able to maintain the tracked object in the centre of the image which is captured by the camera. Furthermore, it can be easily mounted over mini-robots, thanks to its reduced size and weight. The active system does not limit the movements of the robot where it is mounted. Thereby, the sensory information obtained from this vision system can be used to guide the robot in the workspace.

This paper is organized as follows: Section 2 provides an overview of the components of the active vision system and describes the way how they are communicated. Section 3 describes the mathematical model which has been developed to calculate the movements which are needed to keep the tracked object in the centre of the image. This section also presents the improvements that the authors have added to the CamShift algorithm to segment the tracked object: a hue component based on a distance function from a dynamically established hue reference value and the combination of this hue distance with saturation to improve the segmentation with dynamic backgrounds. In section 4, a real experiment of the tracking of an object manipulated by a robot is shown and its results are discussed. Finally, in section 5, the conclusions of this work are exposed.

2. DESIGN OF THE ACTIVE VISION SYSTEM

The design of the active vision system developed in this paper has three advantages over previous similar systems (Kato, *et al.*, 2002; Mayol, *et al.*, 2002): small size, low weight and wireless communications. These requirements make possible the installation of this system on robotic systems without interfering with their movements.

The portable part of the system is composed of a wireless CMOS colour mini-camera (dimensions: 21x21x21mm; weight: 21g) and three micro-servos (dimensions: 22x11x20 mm; weight: 8g). These three servos have been assembled with small pieces of aluminium so that their rotation axes are perpendicular. Therefore, the camera can be rotated around three different axes: pan (angle around the *Y* axis), tilt (angle around the *X* axis) and roll (angle around the *Z* axis). Each servo has three wires: two for power (0V and +6V) and one for the PWM (Pulse-Width Modulation) signal which controls the rotation angle.

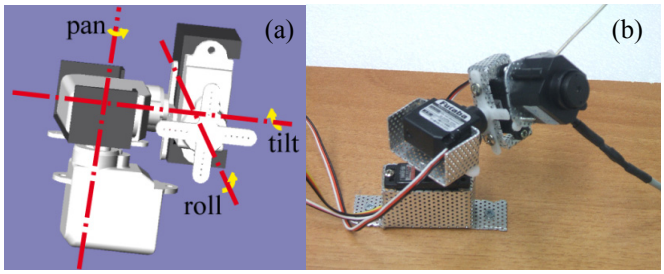

Fig. 1. Assembly of servos: (a) diagram, (b) real prototype.

The servos are connected to a controller board (*Arduino Diecimila*) which sends the PWM control pulses to each servo through three output pins. This board has a ZigBee module which establishes a wireless serial connection with a controller PC.

The controller PC is the fixed part of the system and deals with the processing of the images captured by the camera and the calculation of the rotation angles of the servos. The overall system works as follows:

1. The mini-camera registers a colour image of the tracked object and sends it to the controller PC through a 1.2GHz wireless link.

2. The controller PC receives the images from the mini-camera through a wireless receiver which is connected to a frame-grabber (*Matrox Morphis*). This frame-grabber digitalizes the analogic PAL signal of the wireless receiver in order to obtain 628x582 colour digital images. These images are processed to determine the rotation angles of the servos which are needed to see the tracked object in the middle of the image.

3. The estimated rotation angles of the servos are sent to the controller board by a ZigBee modem which is installed in the controller PC. Each rotation angle is codified as a two-byte command: the first byte identifies the servo which will be moved (pan servo, tilt servo or roll servo) and the second byte identifies the value of the rotation angle in degrees.

4. The controller board receives the wireless servo commands through the ZigBee module and process them in order to generate the corresponding PWM pulses in the digital output where the servo to be moved is connected. Finally the selected servo rotates until it reaches the angle specified by these PWM pulses.

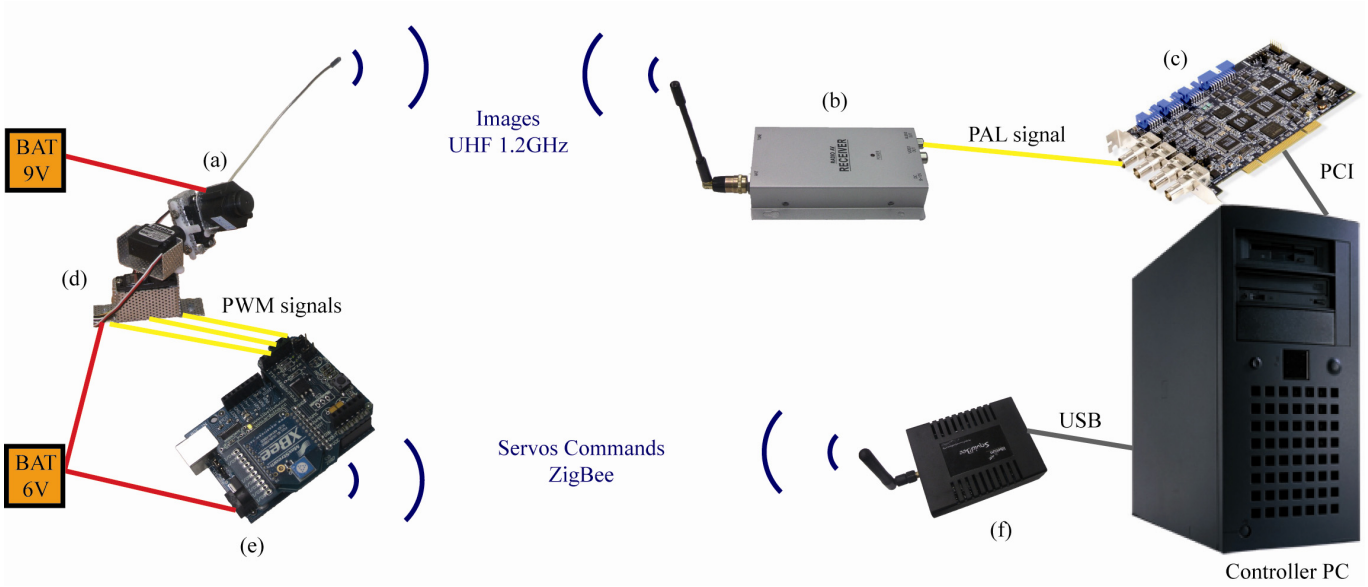

Fig. 2. Components of the active vision system: (a) mini-camera, (b) 1.2GHz wireless receptor, (c) frame-grabber (*Matrox Morphis*), (d) servos, (e) controller board (*Arduino Diecimila*) with ZigBee module and (f) ZigBee modem.

3. VISION-BASED CONTROL OF THE SYSTEM

3.1 Pan/Tilt Estimation for the Control of the Servos

This section introduces a suitable geometric model of the image formation process from which the equations for the pan/tilt control in the tracking process are derived.

A model of a pinhole camera has been assumed to reduce the process of image formation to tracing rays from points on objects to pixels on the image plane (Ma*, et al.*, 2004). This model can be represented by (1), which is used to determine a precise correspondence between points P_C (*X, Y, Z*) in the 3D space and their projected images *p*(*u, v*) in the 2D image plane. The first matrix K_s in (1) contains the scale factors (s_x, s_y), which relate image coordinates (in metric units) to pixel coordinates, and the coordinates (o_x,o_y) of the principal point. The second matrix K_f (perspective matrix) contains the focal length *f* of the camera. Finally, the last matrix Π_0 (projection matrix) enables the product between the homogenous coordinates of the 3D point and the previous matrices.

$$\lambda\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}s_x & 0 & o_x\\0 & s_y & o_y\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}f & 0 & 0\\0 & f & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix} \quad (1)$$

The first two matrices in (1) are grouped into the calibration matrix *K*. This matrix contains all the intrinsic parameters of the camera, which are estimated by means of a calibration process (Bouguet, 2007). Thereby, the image formation process can be modelled by the following equation:

$$p = K\cdot\Pi_0\cdot P_C = K\cdot\Pi_0\cdot {}^{C}T_M\cdot P \quad (2)$$

where the matrix ${}^{C}T_M$ represents the transformation between the camera frame *C* and the world frame *M* when 3D points are not expressed in the camera frame. In the system developed in this article, this matrix is a rotation matrix *R* because the camera is only rotated by pan and tilt angles. The tracking algorithm calculates the rotation angles of this matrix which make the centre of the object situated at pixel *p*(*u, v*) match with the central pixel *p'*(*u', v'*) of the image. Applying the projection model of the camera, the following equation is obtained:

$$p' = K\cdot\Pi_0\cdot R\cdot P \quad (3)$$

Substituting the 3D point P by its projection model in (3), the current position *p* of the tracked object in the image can be related with the desired position *p'* in the centre of the image through the following equation:

$$p' = K\cdot R\cdot K^{-1}\cdot p \quad (4)$$

The rotation matrix *R* can be decomposed into two matrices: $R_{pan}(y_C,\alpha)$ for the pan angle and $R_{tilt}(x_C,\beta)$ for the tilt angle:

$$R = R_{pan}(y_c,\alpha)\cdot R_{tilt}(x_c,\beta) \quad (5)$$

$$R_{pan}(y_c,\alpha)=\begin{bmatrix}\cos(\alpha) & 0 & \sin(\alpha)\\0 & 1 & 0\\-\sin(\alpha) & 0 & \cos(\alpha)\end{bmatrix} \quad (6)$$

$$R_{tilt}(x_c,\beta)=\begin{bmatrix}1 & 0 & 0\\0 & \cos(\beta) & -\sin(\beta)\\0 & \sin(\beta) & \cos(\beta)\end{bmatrix} \quad (7)$$

For the sake of simplicity, the pan and tilt rotation axis are supposed to coincide with the y_C and x_C axis of the camera coordinate system. Therefore, pan rotations only involve movements in the image along the x_C axis while tilt rotations involve movements in the image along the y_C axis. The rotation matrix *R* in (4) can be split into two equations:

$$\begin{bmatrix}u'\\v\\1\end{bmatrix} = K\cdot R_{pan}(y_C,\alpha)\cdot K^{-1}\cdot\begin{bmatrix}u\\v\\1\end{bmatrix} \quad (8)$$

$$\begin{bmatrix}u\\v'\\1\end{bmatrix} = K\cdot R_{tilt}(x_C,\beta)\cdot K^{-1}\cdot\begin{bmatrix}u\\v\\1\end{bmatrix} \quad (9)$$

cos(α) *is* isolated from (8) while *cos*(β) is isolated from (9). In both cases, a second-degree equation is obtained:

$$\cos(\alpha)=\frac{-b_\alpha\pm\sqrt{b_\alpha^2-4a_\alpha c_\alpha}}{2a_\alpha}\ \text{with}$$
$$a_\alpha=(o_x-u)^2+(fs_x)^2$$
$$b_\alpha=-2(o_x-u)(o_x-u') \quad (10)$$
$$c_\alpha=(o_x-u')^2-(fs_x)^2$$

$$\cos(\beta)=\frac{-b_\beta\pm\sqrt{b_\beta^2-4a_\beta c_\beta}}{2a_\beta}\ \text{with}$$
$$a_\beta=(o_y-v)^2+(fs_y)^2$$
$$b_\beta=-2(o_y-v)(o_y-v') \quad (11)$$
$$c_\beta=(o_y-v')^2-(fs_y)^2$$

A positive solution and a negative solution are obtained from each equation. The negative solution is not considered

because it implies rotation angles in the second and third quadrants which are not possible for the camera. When the arc-cosine is applied to the positive solution, two angles with the same absolute value but with opposite signs are obtained. Finally, the sign of the rotation angle is determined according to the direction towards which the servo has to move in order to make the object be in the centre of the image.

3.2 Tracking Algorithm by Combining Hue and Saturation.

The CamShift (Continuously Adaptive Mean Shift) algorithm (Allen*, et al.*, 2004; Bradski, 1998) has been used to develop the tracking of a selected object in a sequence of images which are captured by the mini-camera. This algorithm is an variation of the Mean Shift algorithm (Comaniciu and Meer, 2002) which can deal with dynamic probability distributions (with changes in their size and their position) which represent moving objects. All the images captured by the mini-camera are transformed from the RGB model to the HSV model because the hue and saturation components are used to segment the tracked object.

The CamShift algorithm (Bradski, 1998) can be summarized in the following steps:

1. The user selects the object to be tracked in the first image registered by the mini-camera. This selected region is the initial search window for the Mean Shift algorithm.
2. A target distribution (represented by a histogram $h(x)$) is obtained from the selected region. This distribution represents the probability that a specific colour belongs to the tracked object.
3. For each new image, a probability distribution image is obtained by back-projecting the histogram calculated at step 2. Each pixel value of this image indicates the probability that the corresponding pixel belongs to the target.
4. The Mean Shift algorithm is applied to the probability distribution image of step 3 in order to obtain the centroid of the region which best matches the target distribution.
5. The location of the search window for the next image is situated at the centroid calculated at step 4 and its size is also re-computed. Go back to step 3.

The original CamShift algorithm uses the hue component of the standard HSV model to calculate the histograms and back-projections of steps 2 and 3. In this colour model, the hue component is a counter-clockwise angle around an inverted cone in the range [0º, 360º]. However, this order is not natural because visually similar colours may have very different hue values (e.g. 1º and 359º are similar red hues). Thereby, similar hues may involve the creation of very different histograms at step 2.

In this paper, a first contribution is the use of a hue distance (Hambury and Serra, 2001) instead of the original hue component to solve this problem. The hue distance is a function which represents each hue value H as a distance from a reference hue value H_{ref}. The following distance function is used instead of the hue component:

$$d\left(H, H_{ref}\right) = \begin{cases} \left|H - H_{ref}\right| & \text{if } \left|H - H_{ref}\right| \le 180^{\circ} \\ 360^{\circ} - \left|H - H_{ref}\right| & \text{if } \left|H - H_{ref}\right| > 180^{\circ} \end{cases} \quad (12)$$

The hue reference H_{ref} is the hue value which has the highest frequency in the histogram $h(x)$ obtained from the region selected at step 1:

$$H_{ref} = x \quad \text{where } x \in \left[0^{\circ}, 360^{\circ}\right] \left| \max\left\{h\left(x\right)\right\} \right. \quad (13)$$

Firstly, a histogram of the hue component of the standard HSV model is calculated in order to obtain the hue reference H_{ref} using (13). Afterwards, the histogram is re-computed using the hue distance in (12) and it is used as target distribution at step 2. All the following images are transformed to the HSV model but using the hue distance instead of the hue component.

The use of the hue component to obtain the probability distribution image at step 3 is not sufficient when there are elements in the background which have similar hue values to the target object. In this case, the Mean Shift algorithm may include wrongly in the search window elements from the background. An approach based on differences between consecutive frames (Chu*, et al.*, 2007) is not useful in this case because the background also changes when the camera moves.

For these reasons, a second contribution of this paper is the combination of the hue distance and the saturation component from the HSV model. In the step 2 of the algorithm, two histograms are calculated: one for the hue distance $h_{d(H,Href)}(x)$ and another one for the saturation component $h_S(x)$. In the step 3 of the algorithm, the histogram $h_{d(H,Href)}(x)$ is used to obtain the back-projection $B_{d(H,Href)}(x, y)$ of the hue distance channel, and the histogram $h_S(x)$ is used to obtain the back-projection $B_S(x, y)$ of the saturation channel. These two back-projections are combined according to the following equation in order to create a final probability distribution image $B(x, y)$ which is used by the Mean Shift algorithm at step 4:

$$B\left(x, y\right) = B_{d\left(H, Href\right)}\left(x, y\right) - \left(\neg\, B_S\left(x, y\right)\right) \quad (14)$$

$$\neg\, B_S\left(x, y\right) = 255 - B_S\left(x, y\right) \quad (15)$$

Equation (14) removes from the hue distance back-projection those pixels whose saturation channel does not match the saturation values of the tracked object. Most of the background pixels, whose hue values are similar to the

tracked object, can be removed because their saturation values are different (see Fig. 3). Therefore, only the pixels with similar hue and saturation to the object are considered.

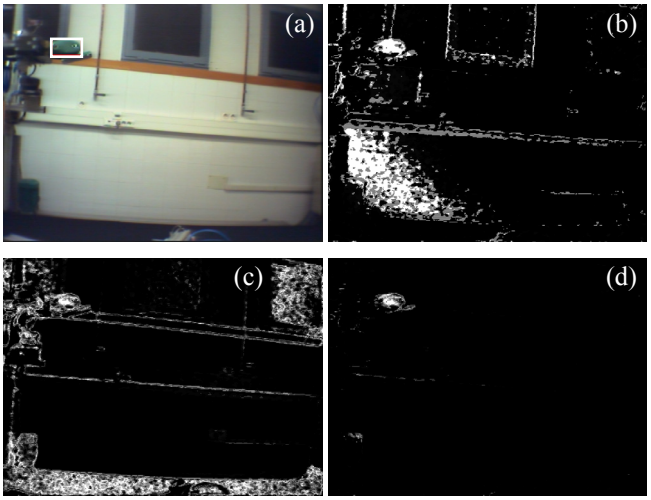

Fig. 3. Comparison of probability distribution images: (a) original image with the tracked object inside a rectangle; (b) hue distance back-projection; (c) saturation back-projection and (d) combination of hue distance and saturation back-projections.

4. EXPERIMENTAL RESULTS

An experiment has been developed in order to verify that the active vision system is able to track an object. In particular, a metallic piece has been grasped by a robotic manipulator which is moving at 70mm/s. Figure 4 shows some frames of the trajectory of the tracked object (which is enclosed in a rectangle) when the active vision system does not move.

Figure 5.a shows the trajectory of the centroid of the object in the image space. This trajectory implies movements of the object in the x and y axis of the image and thus, the active vision system has to perform combined pan/tilt rotations in order to maintain the object in the centre of the image.

Figure 6 shows some frames registered by the mini-camera while the active vision system is performing the tracking of the object. Figure 5.b depicts the location of the centroid of the object. It shows how the tracking process is successful because the system is able to maintain the centroid of the object near the centre of the image with a small error (30±16 px in the x axis and 22±13 px in the y axis).

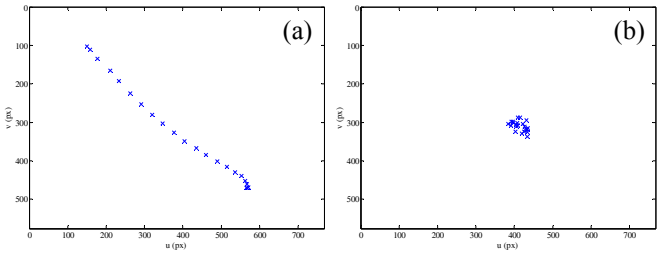

Fig. 5. Trajectory of the centroid of the tracked object: (a) without movements of the servos and (b) with movements of the servos.

Figure 7.a depicts the evolution of the rotation angle of the pan servo and Figure 7.b depicts the evolution of the rotation angle of the tilt servo. Both figures show a continuous evolution of the angles which implies that the system performs the tracking process without abrupt movements.

Fig. 4. Trajectory of the object (frames 1, 7, 14 and 21) while the active vision system does not move.

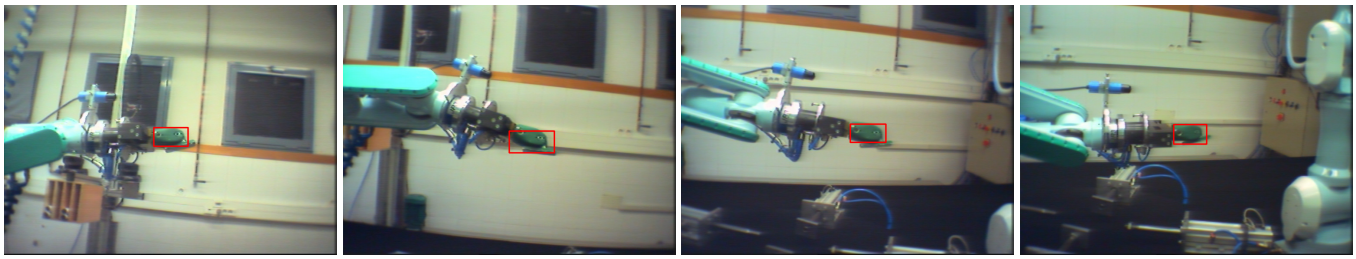

Fig. 6. Trajectory of the object (frames 1, 7, 14 and 21) while the active vision system is performing the tracking.

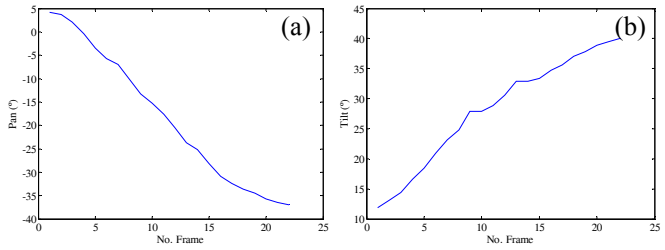

Fig. 7. Evolution of rotation angles: (a) angle of pan servo and (b) angle of tilt servo.

5. CONCLUSIONS

In this paper, a miniaturized active vision system has been designed and implemented in order to track a moving object. Its small dimensions and wireless control makes it possible to install it on robotic systems without interfering in their tasks. The image formation process is modelled to obtain the correspondence between the current position of the object in the image and the rotation angles which are needed to see the object in the centre of the image.

The Camshift algorithm has been used to determine the centroid of the object in the sequence of images captured by the mini-camera. Two contributions have been added to the original algorithm: the use of the hue distance and the combination of the back-projection of the hue distance and the saturation of the HSV model. The hue distance represents hue values as distances to a reference hue value, which is dynamically established from the object histogram. Therefore, the problem of the circular ordering of the original hue channel is solved because this distance function guarantees that different hue distance values imply visually different hues. However, the use of the hue distance to track the object is not sufficient when there are regions with similar hues in the background. For this reason, a combination of the back-projection of the hue distance component and the saturation component has been developed. This combination removes those pixels whose hue and saturation do not match the object.

This algorithm has been successfully verified in the tracking of different objects carried by a robotic manipulator along different paths. One of those has been shown in the Figures 4 and 6. In future work, it would be interesting to add a technique which predicts the movement of the object (e.g. Kalman filter) to reduce the tracking errors. This system will also be used to supervise human-robot interaction tasks.

ACKNOWLEDGMENTS

This work is funded by the Spanish Ministry of Education and Science (MEC) through the research project DPI2005-0622 and the pre-doctoral grant AP2005-1458.

REFERENCES

Allen, J. G., R. Y. D. Xu and J. S. Jin (2004). Object Tracking using CamShift Algorithm and Multiple Quantized Feature Spaces. In: *Pan-Sydney Area Workshop on Visual information processing*, pp. 3-7, Sydney, Australia.

Bernardino, A. and J. Santos-Victor (1999). Binocular tracking: integrating perception and control. *IEEE Transactions on Robotics and Automation,* **15**(6), 1080-1094.

Bouguet, J. Y. (2007). Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/

Bradski, G. R. (1998). Computer Vision Face Tracking for Use in a Perceptual User Interface. *Intel Technology Journal,* **2**(2), 13-27.

Comaniciu, D. and P. Meer (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **24**(5), 603-619.

Chu, H., S. Ye, Q. Guo and X. Liu (2007). Object Tracking Algorithm Based on Camshift Algorithm Combinating with Difference in Frame. In: *IEEE International Conference on Automation and Logistics*, pp. 51-55, Jinan, China.

García, G. J., J. Pomares and F. Torres (2007). A New Time-Independent Image Path Tracker to Guide Robots using Visual Servoing. In: *12th IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 975-964, Patras, Greece.

Hambury, A. G. and J. Serra (2001). Morphological Operators on the Unit Circle *IEEE Transactions on Image Processing,* **10**(12), 1842-1850.

Jacquot, A., P. Sturm and O. Ruch (2005). Adaptive Tracking of Non-Rigid Objects Based on Color Histograms and Automatic Parameter Selection. In: *IEEE Workshop on Motion and Video Computing*, pp. 103-109, Breckenridge, Colorado, USA.

Kato, T., T. Kurata and K. Sakaue (2002). Face Registration Using Wearable Active Vision Systems for Augmented Memory. In: *Digital Image Computing: Techniques and Applications*, pp. 1-6, Melbourne, Australia.

Ma, Y., S. Soatto, J. Kosecka and S. S. Sastry (2004). *An Invitation to 3D Vision: From Images to Geometric Models*, Springer Verlag, New York, USA.

Malis, E. and S. Benhimane (2005). A Unified Approach to Visual Tracking and Servoing. *Robotics and Autonomous Systems,* **52**(1), 39-52.

Mayol, W. W., B. J. Tordoff and D. W. Murray (2002). Wearable Visual Robots. *Personal Ubiquitous Comput.,* **6**(1), 37-48.

Pérez, P., C. Hue, J. Vermaak and M. Gangnet (2002). Color-Based Probabilistic Tracking. In: *7th European Conference on Computer Vision-Part I*, pp. 661-675, Copenhagen, Denmark.

Saunier, N. and T. Sayed (2006). A Feature-Based Tracking Algorithm for Vehicles in Intersections. In: *3rd Canadian Conference on Computer and Robot Vision*, pp. 59-66, Quebec, Canada.

Stolkin, R., I. Florescu, M. Baron, C. Harrier and B. Kocherov (2008). Efficient Visual Servoing with the ABCshift Tracking Algorithm. In: *IEEE International Conference on Robotics and Automation*, pp. 3219-3224, Pasadena, California, USA.