Abstract—In this work we present a robust tracking and pose estimation of an end effector of a 7 DoF robotic arm under a binocular stand-alone configuration. The proposed method is composed of sequence of three processes. The first one is a fast segmentation in HSV color space of a planar patch placed at the end effector. This step considers the segmentation using pixels classification and Euclidean distance a similarity measure. The second step is the selection of a region of interest, feature point extraction and tracking over that region. Finally, the feature points can be used to estimate the homography between world reference frame and image frame. The proposed methodology runs in real time and it is applicable for visual servoing and the grasping.

Index Terms—tracking, segmentation, interest points, robotic arm, visual servoing.

I. INTRODUCTION

In recent years, manipulation tasks for service robot have attracted many researches along the world to these topics. The manipulation is a hard problem that has been tackled in different ways. For example, in industry, the manipulation tasks are programmed by a human and only work in a structured environment. In the other hand, service robots use a different approach: compute positions a robotic arm using planning algorithms under unstructured environments where there is uncertainty, illumination conditions, errors on localization, cumulative errors, etc. This ways to solve the same problem, create the main difference between these two fields of robotics.

In order to have a robust service robot, it must have the ability to grasp and manipulate different objects. To solve these tasks, it is required to have an accurate kinematic model of the manipulator. With this model and a motion-planning module, it is possible to move the manipulator from one position to another in a free-collision path. However, as many research institutes have showed it, in order to have a robust module for manipulation task it is necessary to have a verification process that includes a position error feedback. Commonly visual feedback is used in order to do servoing.

Many of the works reported in visual servoing, use a visual tracking system which compute the position of the object based on segmentable features (with low computational processing) most of them in very well structured ambient, making no real difference from the approaches proposes in manufacturing robotics.

In this work, we present a robust and very fast method for detection, tracking and pose estimation of a robotic arm end effector planned to be used on a service robot.

A. Related Work

The use of features like corners, interest points and contours for doing visual servoing is not new, a good work dealing with features can be found in [1]. In this work, it is known the model of the object to be tracked, allowing robustness under partial occlusions of the target. In others developments [2] the extraction of interest regions, template matching and Kalman Filters is applied when the features selection has a preprocessing stage consisting in changing of color space, segmentation and the selection of interest region.

It is important compute the position of a robot (or end effector) in every moment (or every frame) for reaching one end position for an object and calculate the rigid transformation between both planes, end effector and camera [3]. The process of extraction of features not necessarily is by interest points, in [4] this features are statistical moments like area, center of mass and orientation which allow estimate the pose of end effector for ultrasonic probes in medical applications, or micropeg in hole applications [5]. Also this kind of programs assumes that the end effector and the camera are in perpendicular planes and the orientation is easy to compute. Not only the end-effector of robot systems can be tracked, and the concept of shape estimation is important when we need tracking and specific object, like faces [6], the authors extracts features like eyes, nose and mouth for compute the head position using a Kalman Extended Filter.

The transformation matrix (or rigid transformation) for fixed camera systems has to be computed using different reference frames like robot base, camera and end-effector using at least three points or features in the image plane for estimating the transformation between this systems [7]. The pose estimation involves this transformation by corresponding the 2D points in the camera with the 3D points of end effector. The intrinsic parameters of the reference frame (camera) are the rotation and translation.
(taking account the object frame). The object needs to be recognized using edge detection, noise filters and contours approximation, based on white fiducial markers on black background [8]. Some study cases [9] are using four makers that have not be collinear but coplanar for estimate the existing homography between different reference frames using a fast segmentation for different gray values on a white background that corresponds to the features for this computation. Color and shape are treated in [10] using round objects like balls. The segmentation process includes noise reduction and the recognizing process uses Minimum Square fitting with restrictions. Active features like light emitting diode (LED) are used for facilitating the segmentation and feature extraction. The plane is perpendicular to the camera [11]. The number of interest points or features must be between four and six for a good pose estimation [12] the estimation pose problem is solved by Kalman Iterative Filter.

II. PROBLEM DEFINITION AND RESTRICTIONS

We need to calculate the center of mass representative of the end effector (gripper) of a manipulator arm of 7 degrees of freedom because this center is important to create a region of interest related to the gripper position and for visual servoing purposes it will be used for compute the error between one desired position to reach and the gripper position.

The end effector has a specific color that is different from other objects in the scene. It is assumed that the position and color of the gripper in the real world is in the field of view of the camera. Once detected the gripper is necessary to calculate local features such as borders and points of interest that will be the basis for more complex calculations such as the homography between the camera and the end effector.

A. System Classification

The system can be classified accordingly to [13]. The figure 1 shows the elements like camera, manipulator, objects and the table that is the common plane for vision and manipulator system. In this way the system has a binocular stand-alone configuration. Is important to mention that the camera has the stereo function, but we only use the left camera.

III. COLOR SEGMENTATION

A. Color Space

Before to the image processing is important to define the color space. One of the main problems in dynamic environments is lighting conditions. In this sense, it is necessary to isolate or control the lighting. Theoretically, one of the color spaces that can control lighting of brightness is HSV (Hue Saturation and Value). In this case, the conversion from RGB to HSV is performed by the equations (1), (2) and (3) [14].

\[
V = \max(R,G,B) \quad (1)
\]

\[
S = \begin{cases} 
V - \min(R,G,B) & \text{if } V \neq 0 \\
0, & \text{if } V = 0 
\end{cases} \quad (2)
\]

\[
H = \begin{cases} 
60(G-B)/S & \text{if } V = R \\
120 + (B-R)/S, & \text{if } V = G \\
240 + (R-G)/S, & \text{if } V = B 
\end{cases} \quad (3)
\]

B. Pixel classification

Segmentation is the process of divides the image into regions [14]. For doing this exist different techniques (based on thresholds, line detection, etc), but here we present an scheme inspired by the classification of pixels in an image given a specific color in HSV and using the Euclidean distance (4) as a measure for similarity.

\[
d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (4)
\]

The variables \(x_i\) and \(y_i\) define two points in the Euclidean Space so they are H and S parameters from HSV (without considering the Value component \(V\), because is the brightness of the scene).

As mentioned before, the problem of image segmentation can be presented as pixel classification, where the pixels are features in the Euclidean Space. In the literature [15] this problem has been solved by using unsupervised learning (K-Means). For the case of supervised learning the KNN-like methods are used.

We used a variant of Euclidean classifiers [20] by considering a fixed radius \(r\). Namely, given a color in equation (5) and a set of pixels, if the Euclidean distance of the pixels to the centroid is less than or equal to \(r\), is considered that these pixels are in that color. For better understanding see the Algorithm 1.

A start criterion for assigning \(r\) is a visual inspection of the space of \(H\) and \(S\) values for a set of colors, considering that the end effector has a specific color. Section VI considers the value of this parameter.

![Figure 1. View of the system elements](image)
Other color spaces like CIEnv are useful for isolating the lightness component or brightness. In this implementation only HSV is considered. For future work it is desirable compare the performance of different color spaces.

\begin{equation}
\text{color} = (H,S)
\end{equation}

IV. CENTROID CALCULATION

The first stage for compute the centroid (center of mass) of an object is the isolation of this object from others and then using the mean of the \( x, y \) positions for get the center. These steps are discussed below.

A. Connected Components

Connectivity between pixels is a basis concept in digital image processing and allows defining limits and regions. Two pixels are connected if they have: common neighbors (using 4- or 8-neighbor test), similar intensity values, and if exists a path between them [16].

In [17] is considered the fact of a connected component in binary images has a topological structure. The regions inside of limits and its correspondence with limits is 1 to 1. This process it also allows know which components are in others.

This process can understand as labeling. Once the segmentation has been performed, the labeling of every isolated region is calculated. In this step we maintain only the larger area in the sense that this area is the more representative structure from gripper.

B. Using mean for centroid calculation

Once that the connected components algorithm and the larger area is maintain, the next step is perform the calculation for get the center of using equation (6), where \( f(x,y) \) is binary image and \( x, y \) are the pixel positions.

\begin{equation}
c(\bar{x}, \bar{y}) = c \left( \frac{\sum_{x=1}^{n} \sum_{y=1}^{n} f(x,y)}{\sum_{x=1}^{n} \sum_{y=1}^{n} f(x,y)} \right)
\end{equation}

V. LOCAL FEATURES

A. Corners

Corners or interest points detection is previous process for homography estimation. This process is only applied in the segmentation image (only on largest area of the segmented color on the end effector). One of the most popular methods for corner detection is the Harris Algorithm [18]. This technique implements invariance to any corner and its response will be positive in region with corners, negative un regions with borders and small in flat regions.

In addition to bring robustness to the current method, Shi-Tomasi in [19] also consider the pyramidal approach to ensure good corners for tracking. The input parameters are in the Table 1. The feature tracking is other important issue related to homography estimation, because for getting a good pose estimation the same points must be the same on every frame taken from the camera. This process internally uses a matching points calculation for getting similar points in every frame taken from camera.

VI. POSE ESTIMATION

This chapter only presents theoretically results using the concepts discussed below. For getting the pose estimation of the gripper one previous calculus is the homography.

The homography or projective transformation is any transformation, which is linear in projective coordinates and invertible [21]. A general procedure for estimate the homography between two images is described in [22]. The main stages are:

a) Compute the interest points in each image.

b) Calculate a set of interest points matches based on proximity and similarity.

c) The homography is estimated using this matching.

However, we need to find the homography \( H \) between a set of points in image plane and a set of points in world frame, this relationship is given in equation (7).

\begin{equation}
\begin{bmatrix}
x_1 \\
y_1 \\
1
\end{bmatrix}
= H
\begin{bmatrix}
x_2 \\
y_2 \\
1
\end{bmatrix}
\end{equation}

Where \( x_1, y_1 \) are the coordinates related to the image frame \( IFr \) and \( x_2, y_2 \) are for the world frame \( WFr \). In figure 2a, we can see the largest area taken from the gripper (marked with a green rectangle). This area has a rectangular shape with strong corners.

The shape in the gripper has two representations. The first one is in \( IFr \) and the second one is in \( WFr \). For the first representation in figure 2b is plotted the set of corners obtained in the previous stage (Section V, subsection B). In figure 2c is drawn the same set of points related to the \( WFr \). The units are pixels for figure 2b and meters for figure 2c.
Both planes are labeled in clockwise order starting from the lower left point.

a. The largest area of gripper is marked with a green rectangle

b. Labeled points on the largest area of the gripper (pixels)

c. Labeled points in the world frame (meters)

Figure 2. Different sets of points for image frame and world frame

VII. IMPLEMENTATION

The results reported in this document were performed using the OpenCV libraries [14] in C language running on a Linux (Ubuntu 8.04) platform on a PC Intel Centrino with 512 MB of RAM. We obtain only the left image from a stereo camera with a resolution of 320x240 pixels and the average image processing for each image was 0.05 seconds. Figure 3 shows the real system and we can see the camera, manipulator arm, end effector (gripper), the objects and the table (common plane for the system).

A. Color Set

The color set is shown in Figure 4 and is delimited by the elements in the equation (8).

\[ UT = \{ \text{yellow, blue, brown, purple, red, green, orange} \} \]  

In this set the end effector color is orange. This color can be changed but is important that has enough linear separability (see figure 4a) from other colors.

We pick up 100 samples from every element of the color (using the mouse and the clicking in the object). For every color we get the mean (or color centroid) using only HS components. Equation (9) shows this.

\[ \text{color}_i = \frac{\sum_{k=1}^{n} H_k S_k}{n} \]

Where \( \text{color}_i \) is the i-th representative color (or color mean), \( H_k S_k \) are the samples and \( n \) is the total of samples. The parametric space for the samples is shown in figure 5a and figure 5b shows only the mean of these samples.

A. Segmentation and local features

Although the color space defined in Figure 5a is linearly separable and the colors in (8) do no interfere with the end effector color segmentation there is some noise that comes in the form of misclassified pixels or bad segmentation.
This noise increment as the radius $r$ increases in the algorithm 1. We are not consider the V value because theoretically is the brightness of the scene, but depending on the hour of the day and lighting conditions is necessary adjust $r$ to reject or accept more colors for better segmentation. Figure 5b shows the set of colors to be classified as orange by changing $r$. Figure 6 shows other view of color space with different values of $r$.

One way to not consider the noise for the calculus of the end effector centroid is eliminate all small areas (areas that do not exceed a threshold) in the contour extraction step.

The specific centroid for end effector color is shown in equation 10.

$$gripper_{-color} = (8,105)$$

Considering different values in $r$ the noise increments. For example, in $r=50$ we can see in figure 5 that some pixels related to yellow color are bad classified like orange pixels. For avoiding this bad calculus on feature extraction step, we only calculate features on the larger area segmented from the gripper.

The corners calculation is performed in the binary image related to larger area and which has been filtered from noise. The thresholds for the detectors are shown in Table 1.

Finally, the algorithm 2 shows the complete procedures for getting the gripper centroid, segmentation and corners features from an RGB image with $r=35$ and color in equation (10). Further in algorithm 2 is calculated a region of interest from the farthest point from the centroid of the largest area. This region captures only the largest area segmented from the gripper.

![Figure 6. Different radius for extending or contracting the segmentation](image)

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>PARAMETERS FOR CORNER AND EDGE DETECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>Parameters</td>
</tr>
<tr>
<td>Harris</td>
<td>$K = 0.04$</td>
</tr>
<tr>
<td>Shi-Tomasi</td>
<td>Quality level $= 0.01$, Minimum distance: 5 Block size = 3</td>
</tr>
</tbody>
</table>

VIII. RESULTS

The theoretical analysis and implementation results involve the detection and monitoring of the gripper of a manipulator arm. This section demonstrates the application of these concepts using real time images. The complete procedure is in Algorithm 2.

The figure 7 shows pictures from the principal stages of Algorithm 2 for every frame from the camera.

In line 24 of the Algorithm 2 the $r^2$ computation can be performed using the Euclidean distance from the centroid $cen$ for every point on $contour_2$ and maintain only the larger distance and finally apply this to the $roi$ calculation, because the largest distance is related to get the circular mask.

The Figure 7 shows 4 images related to:

- Figure 7a shows one frame taken from the camera.
- Figure 7b shows the binary image after the segmentation.
- The corners extraction is plotted with yellow points in figure 7c.
- Figure 7d shows the circular region of interest that remarks the largest area segmented on the gripper.
For a better validation of results, we can obtain the ground truth mask. Typically, for creating a ground truth mask, it is necessary to have a visual inspection of the object in the image and then a human selects a set of vertices that represents a specific form. These vertices are used to build a polygonal region of interest that filters out those pixels outside from the vertices and create a binary image. For example, in figure 8a we can see one image related to the gripper.

Using this figure, we can select the representative vertices of the largest area of the gripper, for make the polygon that builds the ground truth mask in figure 8b. Figure 8c shows the ground truth mask over the image for a better visual understanding. For visual comparison, in figure 8d we put the segmentation results using algorithm 1.

The next paragraphs are related with numerical data taken from those binary images.

One way to compare the ground truth mask and the segmentation result is using the equations (11) and (12).

\[
\text{error} = \frac{\sum \sum (gt \oplus sg)}{ag} \times 100
\]  

(12)

Where \(ag\) is the area in pixels in the ground truth image \(gt\). This term can be seen as a logical matrix. The \(sg\) term is the segmentation image and \(error\) is the percent of misclassified pixels.

The main reason for use the logical operation \(\text{xor} (\oplus)\) is because returns 1 only when its arguments are different. In this sense, if there is a bad segmentation inside or outside of ground truth, we will count every pixel misclassified in the segmentation.

A good example for better understanding the \(\text{xor}\) operation on images (segmentation and ground truth) is shown in figure 8e. The operation was performed between image 8b and 8d. The quantitative results are in equation (13), (14) and (15).

\[
\text{ag} = 654 \text{ pix}
\]  

(13)

\[
\text{as} = 623 \text{ pix}
\]  

(14)

\[
\text{er} = 5.65 \% 
\]  

(15)

Where \(ag\) is the total of white pixels in the ground truth image, \(as\) is the total of white pixels in the segmented image, and \(er\) is the error using equation (12). The term \(as\) is not used for computer error.

IX. CONCLUSIONS

This work presented a fast schema for compute the tracking and feature extraction for an end effector of a manipulator arm. The current vision system is not susceptible to errors, for example, in cases when the gripper is to far from camera or there is bright light in the scene, the segmentation fails.

After segmentation the process continues to calculate the interest points in the gripper because these features are intrinsic properties of the gripper that will perform more complex calculations, such as the homography and determining a rotation of the gripper with respect to the camera.

In future work, it is desirable to compare the detection process using a scheme based on pattern matching, object classification and even other color spaces in order to make it more robust to noise.
Figure 7. Detection process of the end effector

Figure 8. Ground truth and segmentation results
REFERENCES


