Trajectory Estimation of a Spinning Flying Object Using a High-Speed Vision

Rafael Herrejon, Shingo Kagami and Koichi Hashimoto

Intelligent Control Systems Laboratory, Tohoku University, Sendai, Japan

KEY WORDS

High-speed vision, visual servoing, target tracking, real time control, robotic manipulation.

ABSTRACT

This paper presents the current progress on a spinning flying object robotic catching system, using general purpose and readily available hardware. The authors propose a recursive least squares (RLS) algorithm to extract and predict the position of a flying object in a 3D environment with the information gathered from only one camera.

1 INTRODUCTION

3D visual tracking of a flying object has been addressed by several researchers and successful approaches to catch the object have been achieved in the recent years.

A hand-eye configuration for ball tracking using a 6 DOF robot manipulator and one CCD camera with Fish-eye lens successfully catching a ball has been achieved [1].

Several 3-D target tracking approaches using high-speed active vision systems have been developed [2].

In many, the trajectory of the manipulator is directly generated using visual information gathered with an active vision system or other specialized hardware created for that specific purpose. Most of the research done in robotic catching also uses a combination of light weight robots [3] with fast grasping actuators to achieve the catching [4][5].

Our system consists of one high speed stationary camera, a personal computer to calculate and predict the trajectory online of the object, a 6 DOF arm to approach the manipulator to the predicted position and a 2 DOF fast actuator in order to achieve the fast grip needed. The system addresses an inherent problem of sensing/actuation systems, the integration of a slow arm robot with a fast actuator derives in composite structure with different constraints requiring an elaborated control theory.

Differences of our system with other previous are the use of non-specialized hardware, the utilization of one single camera to predict the position of the object and the importance of the orientation of the object to be caught.

To successfully catch a target moving fast and irregularly in a 3-D space using direct feedback of visual information with the dynamically controlled hand-arm, several requirements have to be fulfilled: (a) detection of the flying object in the image, (b) algorithms to determine orientation, position and motion of the object, (c) tracking of the object in the image, (d) an iterative object trajectory prediction algorithm that adjusts the predicted path according to new sensory input, (e) online motion trajectory generator that changes the intercept trajectory of the manipulator to intercept and catch the object according to the predicted path.
A black and white camera sends the images to the computer. While the object is not found in the searching area, the detecting algorithm keeps running. When the object is found, an algorithm to calculate the orientation and position of the object in 2D is triggered. The calculations take an average of 15ms. The tracking algorithm moves the search area in the image according to the estimated velocity observed of the object. With RLS, the path of the flying object can be predicted independently of the number of points collected already.

With the object trajectory calculated, a catch position and time can be determined adequately. The furthest point of the object path to the base of the arm is considered as the initial prospective catching point, minimizing the movement of the arm and allowing the arm to reach it if within its constraints.

After an acceptable catch point has been determined, the arm attempts to intercept the object, matching its position.

A Hybrid controller will be used to take advantage of both image based and position based methods. The manipulator and camera have their own coordinate frames as well as units, in millimetres and pixels respectively, and a transformation between the coordinate frames has to be calculated to relate them.

The system is being tested throwing the coin from random locations within 1.50-1.70 meters of the base of the arm. The average time of an object to cover this distance is about 0.7 seconds.

2 VISION SYSTEM

The vision system and the algorithms to locate the object in the image, to determine its shape and to track its position are described in this section.

2.1 General Setup

The vision sensor used is a Point Grey Research Inc, Dragonfly Express Camera. It provides a 640x480 pixels greyscale image at a frame rate of 200 fps. The camera is mounted to a steel frame situated 3 metres above the base of the robot (Figure 1).

![Figure 1: Camera mounted](image1.png) ![Figure 2: Example of image from the camera](image2.png)

The images (Figure 2) are sent through a IEEE-1394 interface to an Intel 3.20 GHz Pentium CPU with 2 GB of RAM with Windows XP.
2.2 Object Location, Orientation and Tracking

Because processing a whole image requires much time and computational power, the search area to locate the object is restricted to the left border of the image, direction where the tossing object will fly from. The algorithm sweeps the image starting from the left top corner of the determined area to the right bottom corner.

The algorithm compares the value of the pixel to a threshold, and given the case that the value is higher, the search continues in the immediate vicinity of the found pixel in a counter clockwise direction to follow the perimeter of the object. If the number of pixels of the perimeter of the object found is out of the set range, the object is discarded and the algorithm continues the search at the right of the discarded object. In case the counted pixels are within the range allowed, the object is considered found and the searching routine is stopped.

This method of searching for the object, comparing the pixel threshold while sweeping the image and finding the perimeter of the object at the same time, avoids the binarization of the image and therefore its processing time.

A spheroidal object model was chosen for simplicity, as the parameters of the model contain centroid, radial length of rotation and direction and length of the rotation axis.

When a coin is tossed, the flipping coin shape is one of an ellipse during the most of the time. The coin used has a 3 cms diameter, which becomes 10 pixels on the image when the coin is the furthest to the camera. When the coin gets the closest to the camera, the diameter in pixels is around 62 pixels.

As the orientation of the coin is a very important feature necessary to catch it, the method chosen to fit the ellipse was least mean squares because of its robustness. Least squares method is an approach to solving overdetermined or inexactley specified systems of equations in an approximate sense minimizing the sum of the squares of the residuals.

The orientation of the coin can be calculated with the values of minor axis (b) and major axis (a) as shown in Eq. (1)

\[
\cos \theta = b / a
\]  

(1)

Using the previous frame objects’ center of mass and the actual center of mass, the velocity of the coin can be calculated. The tracking algorithm moves the search area in the image for the next frame according to the velocity of the object estimated.

Tracking and orientation of the object can be seen in Figure 3

![Figure 3: Tracking and orientation of the coin](image-url)
2.3 Cross Calibration

The manipulator and camera have their own coordinate frames as well as units, millimetres and pixels respectively, and a transformation between the coordinate frames has to be calculated to relate them. A very intuitive calibration method is used. Considering the axis of the camera and the robot to be parallel but with different sign, a simple relation of multiplication is used to transform the images coordinates.

3 CATCHING ALGORITHM

3.1 Object Trajectory Prediction

The model of a pinhole camera is given by equations 2 and 3

\[ u(t) = f \frac{x(t)}{z(t)} \]  \hspace{1cm} (2)
\[ v(t) = f \frac{y(t)}{z(t)} \]  \hspace{1cm} (3)

If the z axis of the camera is not considered to be parallel to the gravity vector, the trajectory of the coin in the uv plane is given by equations 4, 5 and 6

\[ x(t) = C_1 + C_2 t + C_3 t^2 \]  \hspace{1cm} (4)
\[ y(t) = C_4 + C_5 t + C_6 t^2 \]  \hspace{1cm} (5)
\[ z(t) = C_7 + C_8 t + C_9 t^2 \]  \hspace{1cm} (6)
\[ C_3^2 + C_6^2 + C_9^2 = g^2 / 4 \]  \hspace{1cm} (7)

Equation 7 is a constraint given by the addition of the decomposition of the vector of gravity in its different components on each axis. When the camera is parallel to the gravity vector, the values of \( C_1 \) and \( C_6 \) are 0. In this case, the trajectory of the coin in x and y are linear and in z is parabolic.

The future path of the flying object is predicted using Recursive Least Squares. RLS algorithm computes the best estimate of the state recursively, independently of the number of points already collected [6].

The calculated 3D position of the object is not correct at the moment, nevertheless the 2D trajectory can be accurately estimated.
3.2 Catching Point Selection

When the trajectory of the object has been calculated, a catch position and time can be determined adequately. The furthest point of the object path to the robot's range of movement is considered as the initial prospective catching point, the arm initial condition is stretched.

After an acceptable catch point has been determined, the arm attempts to intercept the object, matching the predicted position. Because of the time the robot takes to reach a position increases accordingly to the extension of the robot, the catch point should not be too close to the robot's workspace boundaries. On the other side, if the point gets too close to the base of the robot may cause to reach its joint limits.

To catching point has to be checked against the workspace constraints and changed if necessary in case it does not to satisfy its constraints.

4 CATCHING EXPERIMENTS

4.1 Experimental Setup

The robot used for this experiment is a 6 DOF PA-10 Robot from Mitsubishi Heavy Industries, Ltd. This robot is one of the most used in the industry for its maneuverability. An Intel Celeron 2.00 GHz CPU with 1 GB of RAM is used to control the Robot.

The basic configuration of the system is shown in Figure 5, the distance from the axes of the camera to the tip of the robot when its fully extended is 170 cms. With the initial conditions applied, the distance increases to 210 cms.

Due to the development of the fast gripper is still at its early stage, the orientation of the coin is not being utilized.
4.2 Results

Using image based servoing, it is possible to move the manipulator on its X, Y axis in order to intercept the flying object.

\[ y(t) = k(v_{\text{desired}} - v_{\text{real}}) \]  \hspace{1cm} (8)
\[ x(t) = k(u_{\text{desired}} - u_{\text{real}}) \]  \hspace{1cm} (9)

The success rate of this approach is very low due to the lack of information on the z axis. Nevertheless, under certain circumstances, the catching of the coin is achieved successfully, e.g. Figure 6.

Figure 6: Sequence of images of the object catching.
5 CURRENT RESEARCH

With the information of only one camera, several trajectories on a 3D space can yield to the same projected path on the image. Current research in order to choose the predicted trajectory correctly is being done, and in this way change the control approach of the robot from image based to position based servoing. The initial position of the object on the z axis might need to be calculated using the relation of pixels on the image/real size of the object.

6 CONCLUSION

This paper presents some of the results using a vision-based servoing to catch a free flying spinning object. Model based prediction methods have been used to predict the path of the object in an image plane. Basic controllers have been used to control the position of the end-effector and catching the object has been achieved successfully.

REFERENCES


