Autonomous Helicopter Tracking and Localization Using a Self-Calibrating Camera Array

Masayoshi Matsuoka, Alan Y. Chen, Surya P. N. Singh
Stanford University

Abstract - This paper describes initial results for a helicopter tracking and localization methods using a ground-based camera array. By tracking the helicopter from the ground, the camera array self-calibrates the relative locations and orientations of each of the cameras in the array. It then simultaneously estimates the 3-D trajectory of the helicopter with respect to the array. Initial experiments of the tracking algorithm have been performed successfully. Future work will focus on the use of structure for motion and related techniques to extend camera self-calibration routines.

INTRODUCTION

Position estimation is of critical importance in autonomous robotics research as it is the principal measurement used in controlling the machine and localizing collected data. The typical approach is to attach (i.e., strap-down) the sensor suite to the robot. The approach under consideration in this project involves using three cameras located on the ground to track and localize a helicopter, such as the Stanford autonomous helicopter (Figure 1), in a fixed coordinate frame. The purpose is to replace an on-board GPS system to lighten the vehicle and to make it robust to GPS occlusions, and to allow for more aggressive flight maneuvers.

The cameras used by the system are located on the ground in positions that will cover a volume of air containing the space the helicopter will operate in. Because the rotation and translation relationship between each camera is unknown, this extrinsic data will need to be extracted through self-calibration of the array. Once the extrinsic data has been determined, then the x-y-z location of the aerial vehicle can be accurately and robustly tracked.

This approach, which has only recently become feasible due to advances in desktop computing and imaging technology, is a novel approach for robotic localization. However, there are several related localization approaches in the field. Approaches like GPS and radar provide high precision localization accuracy, but tend to be expensive, hard to relocate, prone to occlusion, or have to be deployed on the vehicle. Inertial techniques provide high fidelity, but also have quadratic increases in drift error.

We believe the proposed system will be useful as a low-cost portable alternative to radar based positioning systems and be applicable to cases where GPS can not operate (e.g. when the Stanford helicopter is performing upside down and the radio antenna is pointed at the ground).

BACKGROUND

The core problems in this project are the localization of the helicopter in each image frame and the self-calibration of the extrinsic parameters for the three cameras, which allows the image localization to be mapped to a world coordinate frame. To find the helicopter in each image, the standard technique of background subtraction is used. Essentially, by identifying the background through an average of previous scenes the helicopter can be identified as points in the foreground image. By taking the center of the points identified as not being background points, the center of the helicopter is approximated. As it is currently assumed that the only moving object in the foreground scene should be the helicopter, for each time step there should only be one center in each image and therefore correspondence is not an issue. There are several robustness issues with this implementation of
the background subtraction technique, which are explored later in the paper.

The self-calibration ability of our system will allow us to place the cameras anywhere on a field limited by cable length yet having the cameras cover the field of space where the helicopter will fly. Self-calibration to acquire extrinsic parameters has been done by several groups in the past [1],[2]. The main difference is that they move the stereo cameras in order to extract parameters while we will be moving a point in the image to extract the same type of information. For example, Knight and Reid use a stereo head that rotates around an axis to give calibration and head geometry [6]. Zhang shows that you can use four points and several images from a stereo pair which has moved randomly, but is constant with respect to each other, to compute the relative location and orientation of the cameras along with the 3-D structure of the points up to a scale factor [11]. Our self-calibration technique will be very similar to the strategy devised by Tomasi and Kanade. They use one camera tracking several feature points and take a stream of images while moving the camera. With this data, they can determine the motion of the camera and the coordinates of each of the feature points [7].

Once the helicopter in each image has been identified and the cameras calibrated, then helicopter localization is determined through triangulation techniques [10].

**TRACKING AND LOCALIZATION APPROACH**

**Feature Tracking**

The current feature tracking algorithm utilizes a simple background subtraction method to extract the location of a target object in the images coordinates. First, the statistical model of the background is built by updating a running average of the image sequence over time:

\[
I_{\text{background}}(x, y) = (1 - \alpha)I_{\text{background}}(x, y) + \alpha I_{\text{current}}(x, y)
\]

where \(\alpha\) regulates updating speed. Next, the algorithm takes an image difference of the current image and the background image, and then thresholds out the image difference caused by noise:

\[
I_{\text{difference}}(x, y) = I_{\text{current}}(x, y) - I_{\text{background}}(x, y)
\]

Finally, the estimate of a moving object in the image coordinate \((\bar{x}_k, \bar{y}_k)\) is estimated by the population mean of the non-zero pixel distribution of the image difference:

\[
\bar{x}_k = \sum_{i,j} x_j I_{\text{difference}}(x_i, y_j) \\
\bar{y}_k = \sum_{i,j} y_j I_{\text{difference}}(x_i, y_j)
\]

This simple algorithm works only when the target object (the helicopter) is the only actively moving object in the image sequence. Although slowly-moving disturbance like clouds in the sky can be distinguished from the actively moving target object by tuning the \(\alpha\) and the threshold to appropriate values, this algorithm easily fails to track on the target whenever any other fast moving objects, such as swaying trees, airplanes, moving cars, or walking people come into the view.

As suggested in related literature, the tracking performance can be greatly improved by taking the probabilities of the predicted target dynamics into consideration, such as using the Kalman tracking [10], the condensation algorithm [3], or the multiple hypothesis tracking [4] (just to name a few). These more advanced robust tracking algorithms will be explored, once we identify actual practical issues in this specific helicopter tracking application in initial field tests.

**Structure from Motion**

To calibrate the extrinsic parameters of the system, a structure for motion technique defined by Tomasi and Kanade in 1992 will be used [9]. The equation below shows the standard conversion from a point in global coordinates \((P)\) to a point in local camera coordinates \((p)\). \(A\) is a rotation matrix, and \(b\) is the offset of the camera from the global origin in the camera frame. \((i\) is the camera and \(j\) is the point).

\[
p_j = AP_j + b
\]

Taking into account all the points and fixing a coordinate system at the mean of all the cameras results in the following equation. (\(M\) is the number of cameras and \(N\) is the number of points)

\[
Q = AD
\]

\[
Q = \begin{bmatrix}
P_{1,1} & \cdots & P_{1,N} \\
P_{2,1} & \cdots & P_{2,N} \\
\vdots & \ddots & \vdots \\
P_{M,1} & \cdots & P_{M,N}
\end{bmatrix}_{2M \times N}
\]

\[
D = [P_1 \ \cdots \ P_N]
\]

\[
A = [A_1 \ \cdots \ A_M]^T
\]
If A and D are full rank, then we know their rank is 3 and therefore Q must also be rank 3. Taking the singular value decomposition of Q and ignoring any right or left singular eigenvectors that correspond with the $4^{th}$ or higher singular values (that appear due to noise) results with:

$$Q \approx U_{2 \times 3} W_{3 \times 3} V_{3 \times N}^T = \tilde{A} \tilde{D}$$

$$\tilde{A} = U$$

$$\tilde{D} = W * V^T$$

$\tilde{A}$ and $\tilde{D}$ represent the affine camera positions and the affine structure of the points in the scene respectively. The last step is to convert this transformation back into Euclidian space and thereby producing the extrinsic properties of the camera array. This will be discussed in the “Next Step” section.

The number of points needed to have a chance at self-calibrating the system is defined by

$$2mn > 8m + 3n - 12$$

with m being the number of cameras and n being the number of points. Given that 3 cameras will be used, a minimum of 4 points will be necessary to self-calibrate. Because our cameras are static, we can move the helicopter to 4 different locations and record images at each location. This will provide the points necessary to self-calibrate [8].

**INITIAL RESULTS**

**Experimental Setup**

The current prototype system consists of a helicopter platform (Figure 1) and a ground-based camera array. The camera array includes three compact digital cameras (Point Grey Research Firefly2 cameras using a Firewire interface) all connected to a single laptop computer (Dell 2.4GHz Pentium 4 Windows XP). The three cameras are deployed on the ground in a triangle array with an approximately 7-meter baseline. The baseline is currently limited by 4.5 meter maximum length of Firewire cables (see also Next Steps section). The camera images are captured at a resolution of 640x480 in an 8-bit grayscale format at a rate of 30 frames per second (fps). The sample image of the helicopter taken by the Firefly2 camera is shown in Figure 2.

**Camera Intrinsic Parameters**

A set of five intrinsic parameters for the Firefly2 cameras $(f_x, f_y, o_x, o_y, k_1)$ is calibrated by Bouguet’s “Camera Calibration Toolbox for MATLAB” [5]. A total of 20 images of the calibration checkerboard (a black and white, 7 x 9 grid of 29 mm x 29 mm squares) are used to calibration each camera (Figure 3). The calibration results are shown in Table 1.

With these calibrated intrinsic parameters, the transformation between the camera-frame coordinates $(X^C, Y^C, Z^C)$ and the pixel coordinates $(x_{img}, y_{img})$ are given via the following equations:
Performance Specifications

An issue with the use of a fixed camera-based approach for tracking the helicopter is that the working volume or flight range is limited by both the stereo requirements and minimum tracker resolution. These two constraints respectively limit how near or how distant the helicopter should be and can be used to define a working region for the helicopter.

Stereopsis requires that the helicopter be clearly seen simultaneously in both images for the range to be estimated. As shown in Figure 4, the proximity of the helicopter to the camera is a function of the baseline and the field of view for the camera. Using a 90° field of view for each camera and a baseline translation of 4.5 meters the lower bound on the location was 1.3 meters from the camera origin (at camera 1).

The tracking resolution requires that the helicopter is large enough in the image sequence to be reliably found and tracked. Assuming the helicopter is distant (i.e., its range >> focal length), the camera image can be modeled using the perspective pinhole camera model [7]. This gives the following constraint on the range of helicopter from the cameras:

\[
\frac{X^C}{Z^C} = \frac{1}{f_x} x_d(1+k,r^2) \\
\frac{Y^C}{Z^C} = \frac{1}{f_y} y_d(1+k,r^2) \\
x_d = (x_{neg} - o_x) \\
y_d = (y_{neg} - o_y) \\
r^2 = x_d^2 + y_d^2
\]

The depth constraint can also be used to calculate the localization error for a unit pixels tracking error. For a given depth the translational error can be found as follows:

\[
X_{estimate} = \frac{X_{error \ pixels} \times depth}{f_x} \\
X_{unit \ pixel \ error} = \frac{depth}{f_x}
\]

With the specific values for our camera, the unit error in translation at the maximum range (35m) was found to be 5 cm.

Feature Tracking

The tracking algorithm based on image difference (described above) was implemented and tested using a single camera to track a moving mouse-pointer on the screen of a second laptop. Figures 5a through 5d show, respectively, example images of a current grabbed image, a running averaged background image, the image difference between the current image and the background image, and the image with the tracking marker on the estimated mouse location.
Further, this algorithm was also implemented using a multiple camera configuration as shown in Figure 6. Currently processing three images captured using the three different cameras takes 200 milliseconds, resulting in an estimation speed of 5 Hz.

Verification of Tracking Results

Figures 7a through 7d show the results of the mouse-pointer tracking experiment. Each $x$-$y$ is undistorted using the intrinsic parameters calibrated previously (see intrinsic parameters section). These $x$-$y$ outputs in the image coordinates are used as the inputs for the aforementioned structure from motion problem to resolve the camera extrinsic parameters and the target object motion.
There are several tasks that remain to be done before we can successfully track and localize a helicopter with self-calibrating cameras. Some immediate short-term goals are to explore the possibility of using Firewire repeaters and to build camera mounts for our cameras. Having Firewire repeaters would allow us to extend the baselines of our cameras as this is currently limited by the Firewire single cable length specification (4.5 meters). Another possibility is to network two or three computers together over a wireless network which could potentially give a significantly larger distance between the cameras, but may introduce measurement delay. Currently, the cameras are deployed by placing them in a location at the end of the Firewire cable. Because of the stiffness of the cable and the shape of the camera, the extrinsic parameters of the camera array are not always static. To eliminate this issue, we will need to build mounts for each of the cameras to hold them in place once we have deployed them.

The last step in getting a preliminary system functioning is to convert back from the affine geometry to Euclidian geometry. Forsythe suggests two possibilities: solving these equations with a non-linear least square method or by utilizing the Cholesky decomposition after assuming the geometry constraints are linear. Both methods need to be explored further before determining which strategy best suits our purposes. Once that is finalized, the auto-calibration strategy will be implemented into current C++ framework. To obtain the four points necessary for auto-calibration, the helicopter will be flown to 4 non-planar points. If this strategy proves too difficult then a large calibration cube or a long pole with a ball at the end can be used to produce four points for the system to utilize.

Once the initial system is implemented, we will test our system and compare the results to a GPS solution. We will also have to do a full sensitivity analysis on our solution to determine how errors would affect the system. For example, if there were a percent error in the baseline calibration, what effect would this have on the accuracy of the resulting position estimate?

While the above approach could produce a working system, it may not be robust or fast enough for control algorithms installed in the helicopter. One concern is the background differencing technique to find the helicopter. If objects enter into the view of the cameras or if the helicopter enters a field which makes it hard to distinguish the helicopter from the background, then the system could break down. One way to deal with the situation is to avoid the problem. If we deploy the cameras in such a way that nothing can enter the view, straight up from under the helicopter, we can eliminate some of those issues. However, a fast moving cloud or a plane would still affect the algorithm. Another option is to implement a window of where the helicopter could be based on its last known position and only differencing that section. That could eliminate a large number of spurious objects entering the image scene if they are far away from the helicopter, and it could increase the speed of the system. If image subtraction still seems as a suspect strategy for identifying the helicopter in an image, then a different color helicopter canopy or a bright LED placed on the helicopter may be used. This would help identify the helicopter more robustly by making the image subtraction method more effective or by eliminating its need together. These are options on top of what was mentioned at the end of the “Feature Tracking” section.

The last step would be to implement a feature finder onto the system that would allow the orientation of the helicopter to be found along with its position in 3D global coordinate frame. This feature finder will need at least 3 points on the helicopter in order to define a...
plane which can then be used to extract the orientation of the vehicle.

**SUMMARY AND CONCLUSION**

Most of the components needed to produce a working tracking and localizing system based on cameras that can self-calibrate have been completed. The cameras have been individually calibrated, and a system is in place to locate the helicopter in each image. Based on the motion of the helicopter, structure for motion can be used to extract extrinsic parameters of the camera array up to an affine coordinate system. The biggest hurdle remaining is to now convert that affine coordinate system to Euclidian coordinate frame where the x-y-z position outputs can be directly inputted into the helicopter control algorithm. Once the self-calibration system has been implemented and tested, then various feature tracking methods can be tested to improve speed and robustness.

**REFERENCES**


