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

Alan Y. Chen, Masayoshi Matsuoka, and Surya P. N. Singh

Stanford University

***Abstract -*** *This paper describes an algorithm that tracks and localizes a helicopter using a ground-based trinocular camera array. Using background differencing and a Kalman filter, the helicopter is found in each of the camera images. The location of the moving helicopter in each image is then used to self-calibrate the relative positions and orientations of each of the cameras in the array while simultaneously estimating the 3-D trajectory of the helicopter with respect to the array. Once the camera array's extrinsic parameters have been extracted, simple triangulation can be used in subsequent runs to identify the location of the helicopter in a camera coordinate frame.*

## INTRODUCTION

Position estimation is of critical importance in autonomous robotics research as it is the principal measurement used in machine control and localizing collected data. The approach 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, make it robust to GPS occlusions, and to allow for more aggressive flight maneuvers.

**Figure 1:** Stanford Autonomous Helicopter

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 extrinsic information is usually obtained via calibration of the cameras in the scene utilizing a calibration object, such as a cube with a checkerboard pattern or the cameras are known to be in fixed locations and orientations with known extrinsic parameters. This is not ideal in a field environment because the above methods would require a recalibration of the cameras with the calibration aid every time a camera is jostled or would require a large structure that would fix the cameras in relation to each other while providing enough coverage to view the entire scene. Thus, the process of camera self-calibration is crucial to the tracking problem. Through this the camera array will be able to estimates its geometry while deployed in the field without requiring modifications to the scene or the helicopter. This process also allows us to re-compute the calibration parameters on the fly if a camera has been moved. Without needing an explicit calibration target, self-calibration also serves to make the estimation methods partially invariant to the structure of the vehicle. Our approach uses multiple observations of the same scene motion to recover the extrinsic relationships between the cameras. In particular, this is done using a variant of the structure from motion (SFM) solution. SFM algorithms typically use camera motion to recover static scene structure; however, reversing this approach allows for the computation of the static camera geometry from scene motion

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 described system can 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 Autonomous 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 location to be mapped to a world coordinate frame. Background differencing is used in conjunction with a Kalman filter to locate the helicopter in each image. Essentially, by identifying the background through an average of previous scenes the moving helicopter can be identified as cluster of points in the foreground image. The center of this cluster identifies the approximate center of the helicopter. The addition of the Kalman filter allows the algorithm to focus on a region in the neighborhood of the helicopter while ignoring other moving objects in the scene such as a swaying tree or a moving car.

The self-calibration ability of our system allows us to place the cameras anywhere on a field limited by cable length while having the cameras cover the operating space where the helicopter will fly. Self-calibration to acquire extrinsic parameters has been done by groups in the past [8],[9]. 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 [7]. 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 [14]. Our self-calibration technique utilizes the algorithm developed by Poelman 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 [10].

Once the helicopter in each image has been identified and the cameras calibrated, then helicopter localization is determined through triangulation techniques [13].

## TRACKING AND LOCALIZATION APPROACH

### *Feature Tracking*

The feature tracking algorithm employs the Kalman filter to estimate helicopter location in the image coordinates, both filtering noises in a background differencing method and predicting the helicopter motion based on its stochastic dynamics model [13]. This Kalman filter approach robustly improves the performance of the feature tracking based on background differencing.

### *Background Differencing*

A simple background differencing method is utilized 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:

$$\mathrm{I}_{\mathrm{k}}^{\mathrm{background}}(x,y) = (1-\alpha)\,\mathrm{I}_{\mathrm{k\text{-}1}}^{\mathrm{background}}(x,y) + \alpha\,\mathrm{I}_{\mathrm{k}}^{\mathrm{current}}(x,y)$$

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:

$$\mathrm{I}_{\mathrm{k}}^{\mathrm{difference}}(x,y) = \mathrm{I}_{\mathrm{k}}^{\mathrm{current}}(x,y) - \mathrm{I}_{\mathrm{k}}^{\mathrm{background}}(x,y)$$

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:

$$\bar{x}_k = \sum_i \sum_j x_i\,\mathrm{I}_{\mathrm{k}}^{\mathrm{difference}}(x_i, y_j)$$
$$\bar{y}_k = \sum_i \sum_j y_j\,\mathrm{I}_{\mathrm{k}}^{\mathrm{difference}}(x_i, y_j)$$

Here, the search window $(i, j)$ is a square mask centered at the helicopter location in the previous time step, eliminating unrealistic abrupt jumps in the helicopter location estimate caused by noises and other moving objects.

This simple background differencing method works 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

the target whenever any other fast moving objects are in view, such as swaying trees, airplanes, moving cars, or walking people.

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 [13], the condensation algorithm [1], or the multiple hypothesis tracking [5] (just to name a few). In this research, the Kalman tracking approach is explored to improve robustness in maintaining a lock on the helicopter in this specific helicopter tracking environment.

*Kalman Filter*

The Kalman filter is a well-studied technique, that can be described as an optimal recursive linear estimator, which has been widely used in many computer vision applications, including feature tracking problems. Here, the equations of the Kalman filter algorithm are formulated for this specific tracking problem.

The system model and the measurement model of the Kalman filter are written as:

$$x_{k+1} = Ax_k + w_k$$
$$z_k = Hx_k + v_k$$

where,

- $x_k$: system state
- $A$: transition matrix
- $w_k$: normally distributed process noise
- $z_k$: measurement of the system state
- $H$: measurement matrix
- $v_k$: normally distributed measurement noise

Here, $w_k$ and $v_k$ are zero-mean white, Gaussian random process modeled as:

$$p(w_k) \sim N(0, Q)$$
$$p(v_k) \sim N(0, R)$$

where,

- $Q$: process noise covariance matrix
- $R$: measurement noise covariance matrix

In this feature tracking problem,

$$x_k = [x \quad y \quad \Delta x \quad \Delta y]^T$$
$$z_k = [x_{meas} \quad y_{meas}]^T$$
$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The time update equations project forward in time the current system and error covariance estimates to obtain the a priori estimates for the next time step:

$$x'_{k+1} = Ax_k$$
$$P'_{k+1} = AP_kA^T + Q$$

Then, the measurement update equations incorporate a new measurement in to the a priori estimate to correct a posteriori estimate while updating the Kalman gain matrix $K_k$:

$$K_k = P'_k H^T \left(HP'_k H^T + R\right)^{-1}$$
$$x_k = x'_k + K_k \left(z_k - Hx'_k\right)$$
$$P_k = \left(I - K_k H\right) P'_k$$

*Structure from Motion*

To calibrate the extrinsic parameters of the system, a structure from motion technique defined by Poelman and Kanade in 1997 will be used [10]. As opposed to taking a single camera and taking a stream of images of an object as we move the camera, we will use static cameras and take a stream of images as we move the object in the scene. This will provide the data necessary to utilize the algorithm described below.

The equation below shows the standard conversion from a point in global coordinates ($P$) to a point in local camera coordinates ($p$). $R$ is a rotation matrix, and $t_j$ is the offset of the camera from the global origin, the actual camera position is $-t_j$. ($i$ is the camera and $j$ is the point).

$$p_j = R_i(P_j + t_i)$$
$$R_i = \begin{bmatrix} i_i \\ j_i \\ k_i \end{bmatrix}; t_i = \begin{bmatrix} t_{ix} \\ t_{iy} \\ t_{iz} \end{bmatrix}; p_j = \begin{bmatrix} p_{jx} \\ p_{jy} \\ p_{jz} \end{bmatrix}; P_j = \begin{bmatrix} P_{jx} \\ P_{jy} \\ P_{jz} \end{bmatrix}$$

To convert from 3-D camera frame coordinates to a 2-D image frame coordinate system a scaled orthographic projection, also know as "weak perspective", will be used. This projection technique shown in the equation below approximates perspective projections when the object in the image is near the image center and does not vary a large amount in *z* direction (perpendicular to the camera's image plane). The equations below assume unit focal length and that the world's origin is now fixed at the center of mass of the objects in view.

$$z_i = t_i \cdot k_i$$
$$u_{ij} = \frac{p_{jx}}{z_i},\ v_{ij} = \frac{p_{jy}}{z_i}$$

These equations can be rewritten as:

$$u_{ij} = m_i \cdot P_j + x_i; v_{ij} = n_i \cdot P_j + y_i$$
$$x_i = \frac{t_i \cdot i_i}{z_i},\ y_i = \frac{t_i \cdot j_i}{z_i}$$
$$m_i = \frac{i_i}{z_i},\ n_i = \frac{j_i}{z_i}$$
$$W = R^* P + t^*$$
$$W = \begin{bmatrix} u_{11} & \dots & u_{1N} \\ v_{11} & \dots & v_{1N} \\ \vdots & & \vdots \\ u_{M1} & \dots & u_{MN} \\ v_{M1} & \dots & u_{MN} \end{bmatrix}_{2M \times N}$$
$$R^* = \begin{bmatrix} m_1 \\ n_1 \\ \vdots \\ m_M \\ m_M \end{bmatrix}_{2M \times 3}, t^* = \begin{bmatrix} x_1 \dots x_1 \\ y_1 \dots y_1 \\ \vdots \quad \vdots \\ x_M \dots x_M \\ y_M \dots y_M \end{bmatrix}_{2M \times N}$$

$x_i, y_i$ can be found by subtracting the average image value from the image data leading to the measurement matrix $W^*$. ($M$ is the number of cameras and $N$ is the number of points).

$$x_i = \frac{1}{N}\sum_{j=1}^{N} u_{ij},\ y_i = \frac{1}{N}\sum_{j=1}^{N} v_{ij}$$
$$W^* = W - t^* = R^* P$$

If $R^*$ and $P$ are full rank, then we know their rank is at least 3 and therefore $W^*$ must be also be at least rank 3. Taking the singular value decomposition of $W^*$ and ignoring any right or left singular eigenvectors that correspond with the 4th or higher singular values (that appear due to noise) results with:

$$W^* \approx U_{2M \times 3} \Sigma_{3 \times 3} V^T{}_{3 \times N} = \tilde{R}\tilde{P}$$
$$\tilde{R} = U$$
$$\tilde{P} = \Sigma V^T$$

$\tilde{R}$ and $\tilde{P}$ represent the affine camera positions and the affine structure of the points in the scene respectively which can then be transferred back to Euclidian space with a matrix $Q$. To determine $Q$ we will use the $2M+1$ linear constraints defined below. The last constraint will avoid the trivial solution satisfied by everything being 0.

$$W^* = \tilde{R} Q Q^{-1} \tilde{P}$$
$$|m_i|^2 = |n_i|^2 = \frac{1}{z_i^2} \Rightarrow |m_i| - |n_i| = 0$$
$$m_i \cdot n_i = 0$$
$$|m_1| = 1$$

With these constraints and the Jacobi Transformation of $Q$ the affine system can then be converted back into Euclidian space. If the resulting $Q$ is non-positive definite, then distortions, possibly due to noise, perspective effects, not enough rotation in the system, or a planar object in the view, has overcome the third singular value of $W$ [10].

Multiply all the rotation matricies and the newly found matrix of points by $(R_1)^{-1}$ to convert everything into a coordinate frame based on camera 1.

After this process, the only remaining extrinsic parameters still unknown is $t_i$. To find $t_i$, least squares can be used by expanding the equation below to encompass all the points in each camera.

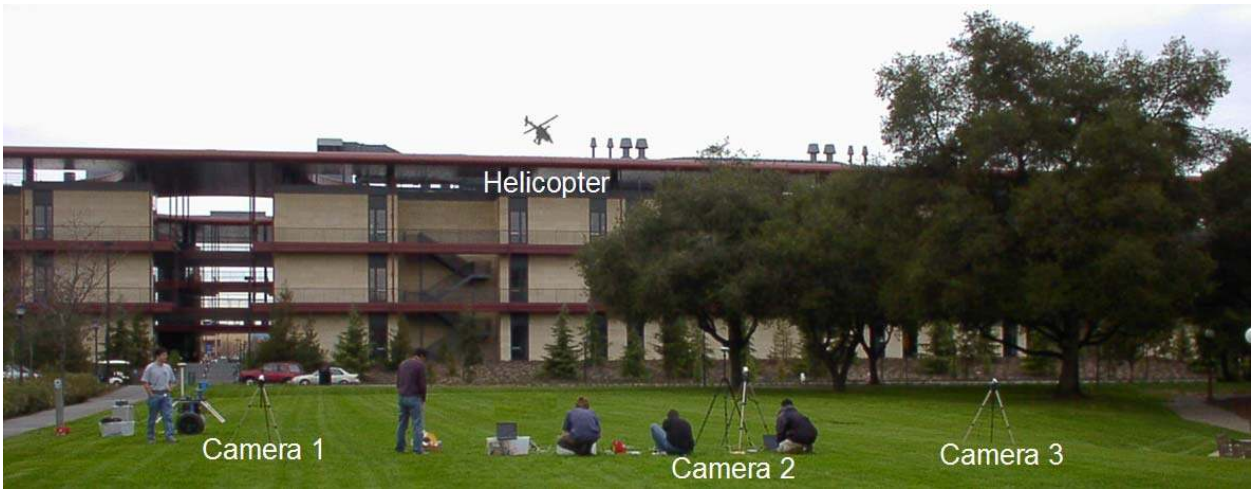

**Figure 2:** Experimental setup (a helicopter and three cameras)

$$\begin{bmatrix} u_{ij} \\ v_{ij} \\ z_i \end{bmatrix} - \begin{bmatrix} i_i \cdot p_j \\ j_i \cdot p_j \\ 0 \end{bmatrix} = R_i t_i$$

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

$$2MN > 8M + 3N - 12$$

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 [11].

## RESULTS

*Experimental Setup*

The current prototype system consists of a helicopter platform and a ground-based camera array (Figure 2). 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 camera images are captured at a resolution of $640 \times 480$ 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 3a, 3b, and 3c.

*Tracking*

The tracking algorithm based on the background differencing method with the Kalman filter (described above) was implemented in the field on each camera to track a common helicopter. Figure 3a, 3b, and 3c are the snap shots of the tracking results by the camera 1, 2, and 3, respectively. The black box is the tracking marker centered at the estimated helicopter location and the thin white larger box is the $150 \times 150$ search window of the background differencing method.

This particular flight test was conducted in an open field on Stanford campus next to a heavy traffic road, where moving cars and walking people constantly came in and out of the scene. While the background differencing-only method frequently failed to track the helicopter in such a busy environment, the Kalman filter was able to maintain the lock on the helicopter during the flight.

Figures 4a through 4c show the resulting helicopter trajectory in the image coordinate. The blue solid lines show the helicopter trajectory tracked by the Kalman tracker. The red dashed lines show the helicopter trajectory manually post-traced in the logged images as true reference. Although the Kalman tracker was able to keep tracking the helicopter, the tracking markers were sometimes lagging in following the helicopter when the helicopter accelerated faster than the pre-defined dynamic model in the Kalman filter equations; fine tuning of the process noise covariance matrix would be necessary for better performance. The mean errors for the Kalman tracking from the true references were roughly 6~8 pixels, as shown in Table 1.

| | mean [pixel] | std [pixel] |
|---|---|---|
| Camera A | 6.9 | 5.0 |
| Camera B | 7.2 | 5.2 |
| Camera C | 8.3 | 6.2 |

**Table 1:** Tracking errors in pixels

**Figure 3a:** Helicopter tracking by Camera 1

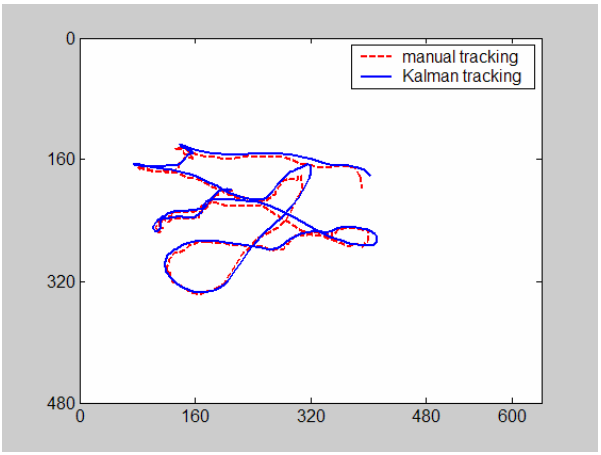


**Figure 4a:** Helicopter trajectory in image coordinate (Camera 1)

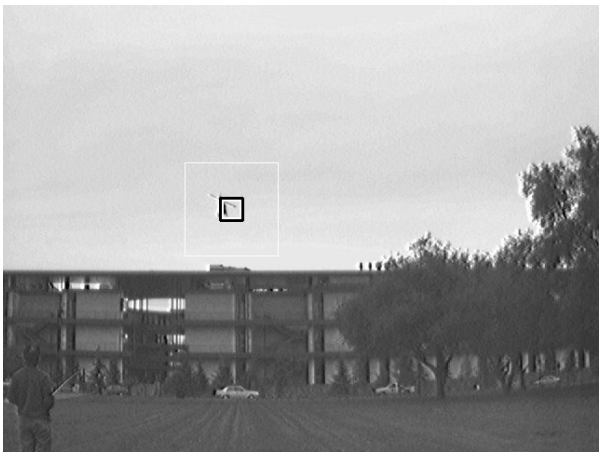

**Figure 3b:** Helicopter tracking by Camera 2

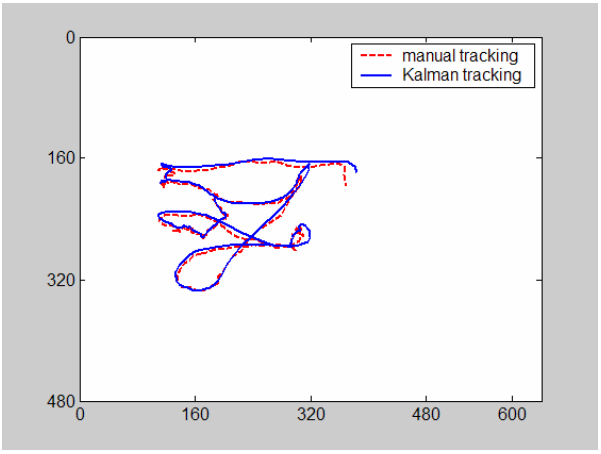


**Figure 4b:** Helicopter trajectory in image coordinate (Camera 2)

**Figure 3c:** Helicopter tracking by Camera 3

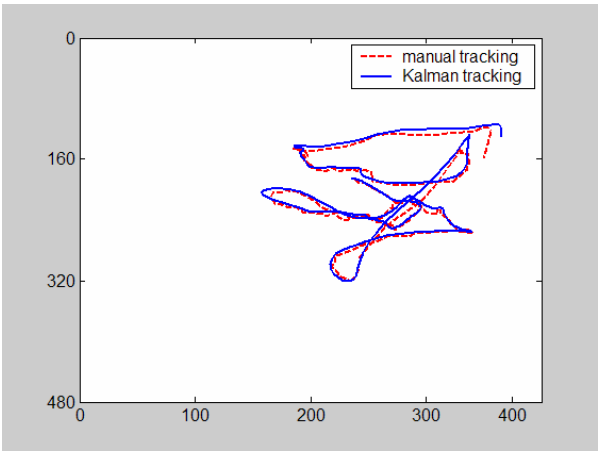


**Figure 4c:** Helicopter trajectory in image coordinate (Camera 3)

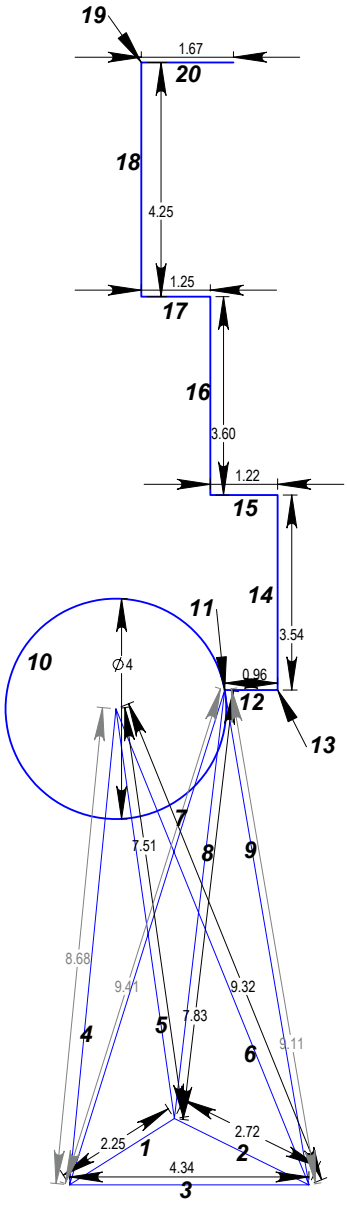


**Figure 5:** Man walking a path (gray distance in meters)

*Structure from Motion*

SFM was used to calibrate the cameras tracking a person walking a specified route (Figure 5). The three cameras were located at the vertices of the triangle at the bottom. The object person started at the right side of the circle, walked around the circle in a clock wise direction, stepped up on the bench on the right side, and then walked along the bench to the end of the path.

**Figure 6:** Image of a man walking along a path (camera 1)

| Segment ID | Scale factor |
|---|---|
| 1 Camera 1 – Camera 2 | 472.4 |
| 2 Camera 2 – Camera 3 | 722.3 |
| 3 Camera 1 – Camera 3 | 515.8 |
| 4 Camera 1 - Center of Circle | 755.4 |
| 5 Camera 1 - Side of Circle | 590.8 |
| 6 Camera 2 - Center of Circle | 641.5 |
| 7 Camera 2 - Side of Circle | 755.5 |
| 8 Camera 3 - Center of Circle | 568.2 |
| 9 Camera 3 - Side of Circle | 584.9 |
| 10 Left -Right Side of Circle | 382.0 |
| 11 Top - Bottom of Circle | 389.9 |
| 12 Small Segment Right of Circle | 1523.9 |
| 13 Step to Top of Bench | 326.9 |
| 14 Walking Left and Walking | 944.5 |
| 15 Turning Left and Walking | 709.5 |
| 16 Turning Right and Walking | 1219.8 |
| 17 Turning Left and Walking | 497.1 |
| 18 Turning Right and Walking | 541.8 |
| Mean | 674.6 |
| Scale Factor (1,2,3) | 570.2 |

**Table 2:** Man Walking a Path

An image of the scene is shown in Figure 6 from the perspective of camera 1. The extrinsic parameters extracted from the person walking along the designated route twice were then used in triangulation for real-time tracking. The final image of the scene is displayed in Figure 7.

Taking each segment along the path, the actual measured distance and the distance reported through triangulation was compared to determine the scale factors in Table 2. In an actual run, the scale factor

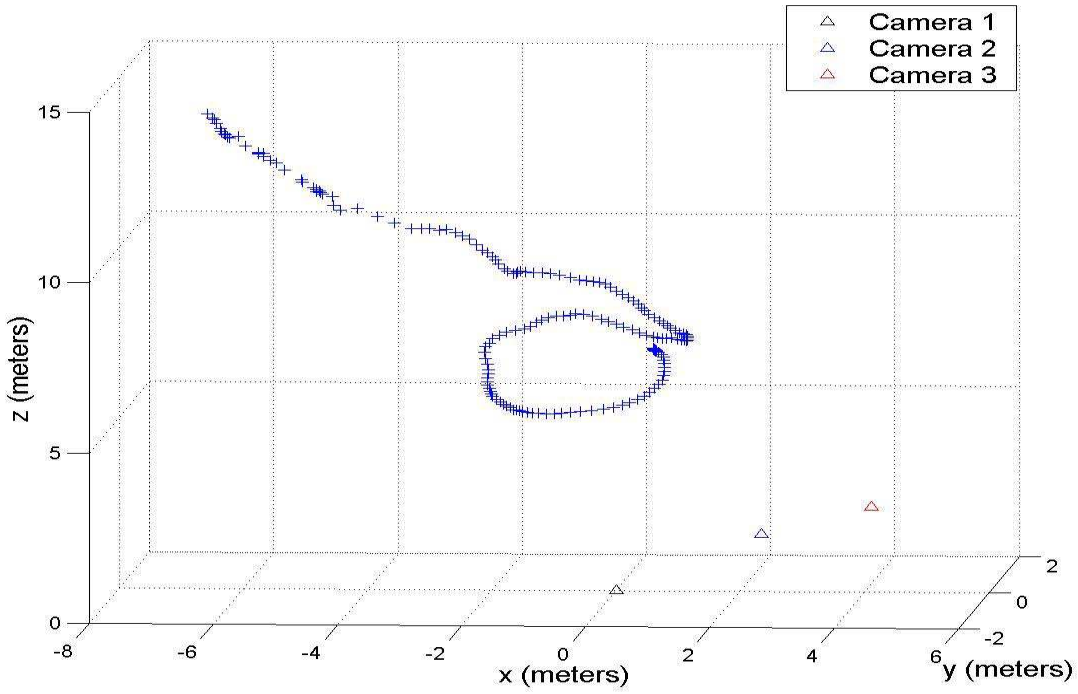


**Figure 7:** Final plot of real time triangulation of a person walking a path

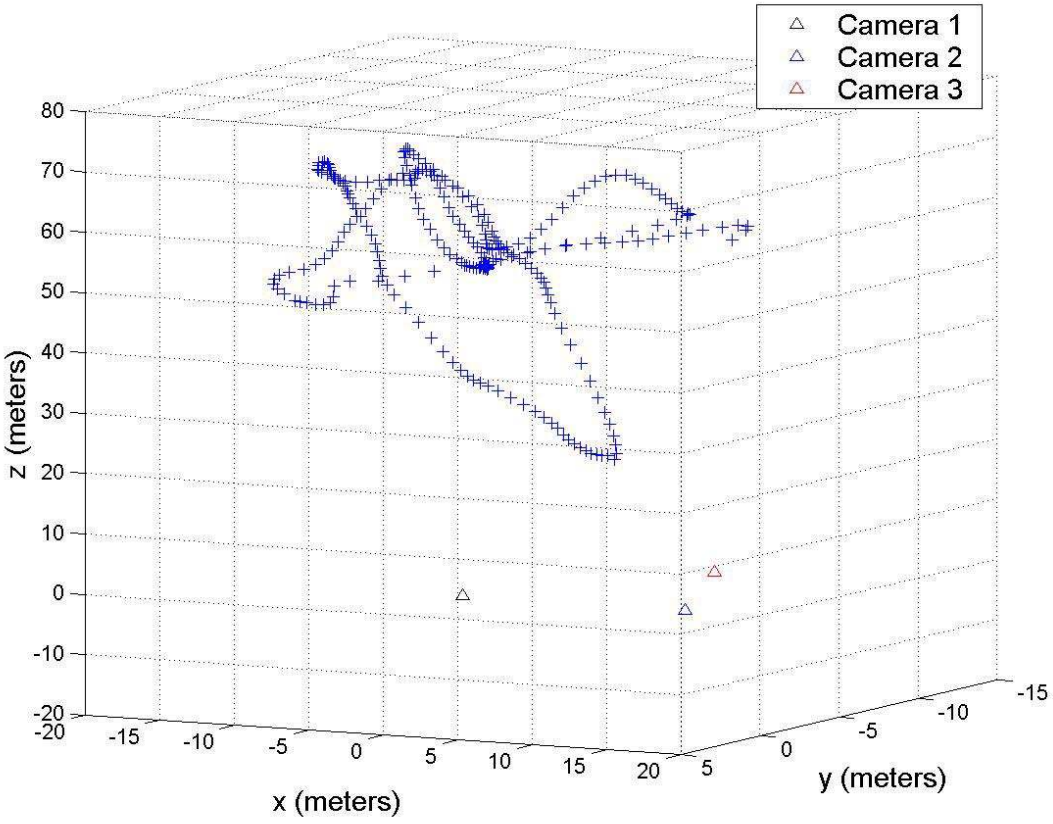


**Figure 8:** Final plot of a real time triangulation of a helicopter run

would be determined with only the baseline distances between the three cameras, as those are the only known parameters. By including the scale factor in triangulation, the location of the object can be determined in meters with respect to camera 1.

Looking at the results of this data, the scale factor does not appear to be uniform. Part of this can be explained by the near-perspective projection camera assumption because the calibration data varied a large amount in the *z* direction and did not stay in the center of the images. This can be seen in that some of the larger errors seem to occur in segments where a large traverse in the *z* direction was performed and at the edge of the camera images.

Another run was performed while tracking a helicopter. The scale factors of the baseline between the cameras are shown below.

| Segment | Scale factor |
|---|---|
| Camera 1-Camera 2 | 11.80 |
| Camera 2-Camera 3 | 58.41 |
| Camera 1-Camera 3 | 31.96 |
| Mean | 34.06 |

**Table 3:** Helicopter scale factors

While the scale factor is not uniform here either, the resulting real-time plot qualitatively matches the trajectory of the helicopter during flight. The final plot resulting from this run is shown in Figure 8. Although there is no GPS data to use as ground truth, the path shown in the plot does appear to follow the path shown in the images and video.

*Working Volume Optimization*

The use of vision as compared to GPS has the disadvantage of requiring visual line of sight (for at least two cameras) and bounding the operating space for which the vehicle position can be reliably estimated. This space can be described as the common working volume for all three cameras such that the helicopter remains sufficiently large in the image plane in order to be reliably tracked [2].

Assuming perspective geometry, the working volume for each camera can be modeled as a right cone with its apex at the camera frame origin. The slant angle of the cone is half the beam width of the camera (40 degrees). The height of the cone is defined by the maximum depth for which the helicopter will project an image of at least 10 pixels (30m for the Stanford Autonomous Helicopter). As the arbitrary translations and rotations between the cameras complicate the bounded geometry, the working volume was solved using numerical techniques.

In addition to providing operating constraints, the working volume calculation provides insight on the optimal placement of the cameras. Assuming that the cameras are placed in a manner such that they are symmetrically placed on the circumference of a circle whose center is near the vehicle, the working volume can be solved for as a function of both the arc angle and the radius (see Figure 9).

An outcome of this calculation is that the maximum working volume is found at the degenerate case of zero arc angle as this causes perfect intersection of the projection cones for each camera. An alternative optimizing constraint is to minimize the sensitivity of the depth estimate to pixel error. This is found by projecting a point in the working volume to the camera image planes, adding a unit error to one of the camera images, and then triangulating using these erroneous points. For the center of the circle the depth error is given by:

$$s = \delta \cot(\theta)$$

where δ is the pixel error, θ is the arc angle and s is the corresponding depth error (in pixels)

The working volume analysis provides design guidance on the relative placement of the cameras. As shown in Figure 9, the depth estimation error for a unit pixel image error is exceptionally large for small arc angles between the cameras. This error decreases rapidly as the angle increases and is unity for an arc angle of 45 degrees. This result is also intuitive as it is a well know consequence in computer vision that an increased baseline provides greater robustness in depth estimation. Greater arc angles also have the effect of reducing the range of depths present. This is advantageous for near-perspective analysis, but requires more careful placement of the helicopter to ensure that its motions remain within the working volume.

However, as the arc angle is increased the operating space available decreases, albeit at a lower rate. Thus, a moderate arc angle (such as 30 degrees) maintains much of the volume while having low sensitivity to pixel errors.

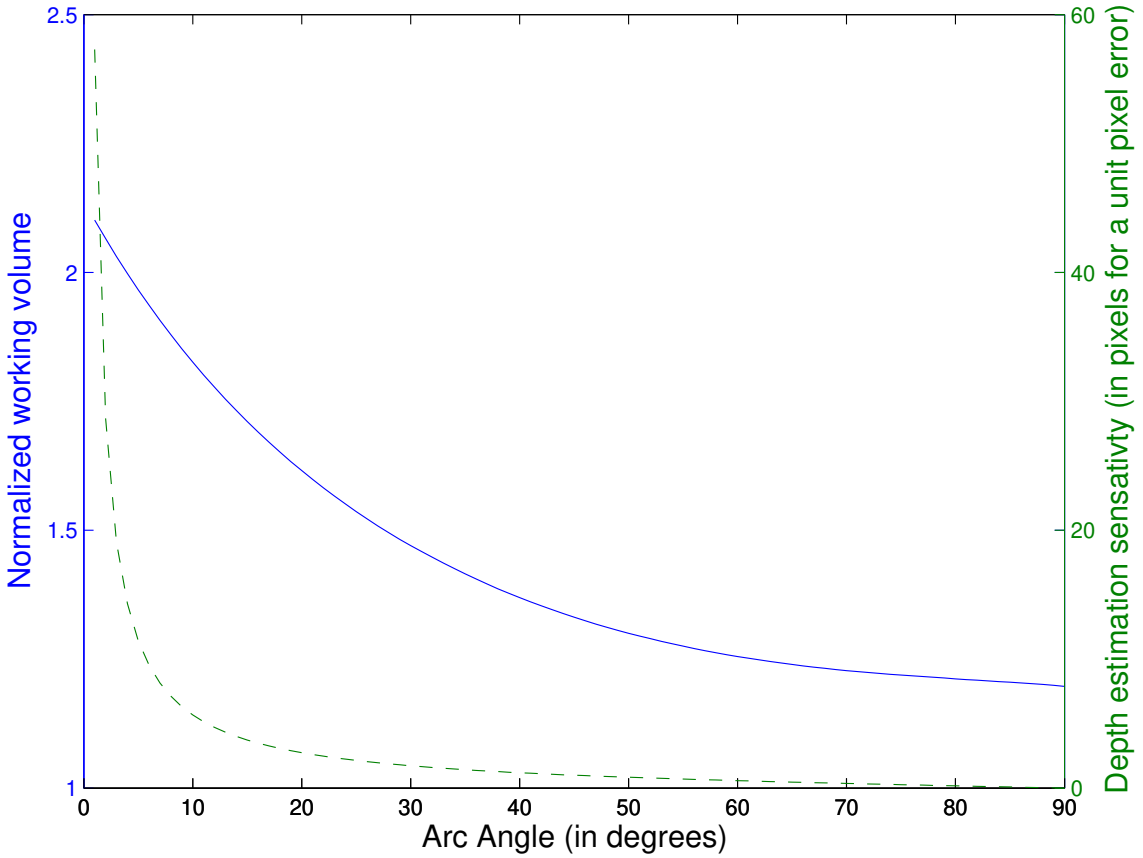


**Figure 9:** Normalized volume and sensitivity for changes in arc angle. The plot shows that depth error decreases faster than the reduction in volume.

## CONCLUDING REMARKS

A method for self-calibration of a camera array has be developed and fielded. In addition, this work has extended and developed methods for tracking and localizing a helicopter. The system takes three intrinsically calibrated cameras and places them in the field at unknown locations and orientations forming an arc angle of approximately 30 degrees with the only constraint that the helicopter to be tracked will always be in view of all the cameras. A tracking system utilizing background differencing and a Kalman filter accurately and robustly tracks the helicopter in motion despite other moving objects in the scene. With at least four non-coplanar data points taken from each camera simultaneously as the helicopter moves, SFM and least squares can be used to extract the extrinsic parameters between each of the cameras assuming near-perspective projection. Once the extrinsic parameters have been found, the helicopter can be tracked and localized up to a scale factor with the origin at camera 1. If the baseline between each of the cameras is known, an estimate of the scale factor can be found to re-project all the points onto a Euclidian coordinate system based on a known metric (meters, feet, yards).

### *Future Work*

The near-perspective assumption is not very accurate in the case of a moving helicopter. The helicopter will often move great distances in the z direction and will not stay near the center of each of the camera's image plane. Han and Kanade have developed a method based on iterating the near-perspective SFM algorithm until the perspective SFM solution can be solved [5]. This will also hopefully help find the scale factor more accurately between what SFM returns and a world metric.

Several helicopter runs still must be performed in order to compare the results from this method to on-board GPS data. That will provide the final ground truth needed to validate this technique.

## ACKNOWLEDGEMENTS

The authors wish to acknowledge the suggestions and contributions of Professors Sebastian Thrun and Gary Bradski for their guidance, comments, and equipment support. The authors also wish to thank Jamie Schulte,

Ben Tse, Justin Tansuwan, Nathan Marz, Professor Andrew Ng, and others of the Stanford Autonomous Helicopter project for their assistance. This research effort and paper were conducted as part of the CS 223b, Introduction to Computer Vision course at Stanford University.

## REFERENCES

[1] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine,* vol. 19:1, Jan. 2004.

[2] A. Blake, D. McCowen, H. R. Lo, and P. J. Lindsey, "Trinocular Active Range Sensing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 15:5, 477-483, May 1993.

[3] Jean-Yves Bouguet, "Camera Calibration Toolbox for MATLAB," MRL, Intel Corp.

[4] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2003.

[5] M. Han and T. Kanade, "Perspective Factorization Methods for Euclidean Reconstruction," Carnegie Mellon CMU-RI-TR-99-22, Aug 1999.

[6] M. Isard and A. Blake, "Condensation: conditional density propagation for vision tracking," *Int. J. Computer Vision*, vol. 29:1, 5-28, 1998.

[7] J. Knight and I. Reid, "Self-Calibration of a Stereo Rig in a Planar Scene by Data Combination," In *Proc. of the International Conference on Pattern Recognition*, Sep 3-8, 2000, 1411-1414.

[8] P. Liang, P., Y. Chang and S. Hackwood, "Adaptive self-calibration of vision-based robot systems, IEEE Transactions on Systems, Man and Cybernetics, vol 19:4, July-Aug, 1989, 811-824.

[9] G. Mayer, H. Utz and G. Kraetzschmar, "Towards autonomous vision self-calibration for soccer robots," Proc. of the Intelligent Robots and Systems (IROS) Conference, vol. 1, 2002, 214-219

[10] C. J. Poelman and T. Kanade, "A Paraperspective Factorization Method Shape and Motion Recovery," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 19. No 3. March 1997.

[11] S. Thrun and G. Bradski, D Russakoff, CS223b Lecture "Structure from Motion," Stanford University, Feb 2004.

[12] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: a Factorization Method: Full Report on the Orthographic Case," Technical Report Cornell 92-1270 and Carnegie Mellon CMU-CS-104.

[13] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, Prentice Hall, 1998.

[14] Z. Zhang, "Motion and Structure of Four Points from One Motion of a Stereo Rig with Unknown Extrinsic Parameters," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol 17. 12, Dec 95.