BALLISTIC ORACLE AT MILITARY BORDERS (B.O.M.B)

M. Adnan Al-Alaoui¹, Senior Member IEEE, Rami Farran², Student Member, IEEE, Claudia Lakkis³, Student Member, IEEE, and Ralph Rabbat⁴, Student Member IEEE

¹Dept.of Electrical and Computer Engineering, American University of Beirut, email: adnan@aub.edu.lb
²Electrical and Computer Engineering Department, American University of Beirut, Lebanon
³GIS Department, Khatib and Alami, Beirut, Lebanon
⁴Information Technology, MIT – email: ralph@mit.edu

Abstract - The objective of the paper is to estimate the source and predict the target of a flying object (projectile) using image processing. This is achieved by taking successive 3D images of the object using two cameras and processing their corresponding data. It can be widely used in military applications where it is an intelligent and relatively cheap way to guess the source and target of projectiles, rockets or bombs.

1. INTRODUCTION

A 3D vision deals with images having gray levels that are function of three spatial variables. Two cameras should be used to produce such a result. These cameras are to be placed therefore adjacent to each other to take stereoscopic images. Before processing the images, it should be ensured that the object is in the scope of the cameras. Each time camera 1 takes a shot, the image is subtracted from the previous one, i.e. latest background, and scanned to see if the object has entered the area. If no object is detected, background is updated and the process repeats until an intruder is detected. In this case, a shot from camera 2 is taken and scanned to make sure that it “sees” the object. If not, another shot is taken from camera 2 until it sees the object.

2. OBJECTIVES

The main objective of the paper is to find the equation of motion for a flying object. This equation is supposed to be that of a parabola of the following form:

\[ X = a_x t^2 + b_x t + c_x \]

\[ Y = a_y t^2 + b_y t + c_y \]

\[ Z = a_z t^2 + b_z t + c_z \]

The object should be shrunk to a point, which is its center point; this assumption will give accurate results since the center point of the object is the most likely point to track the parabola. Hence, our processing should be as follows:

- Take 3D shot of the object
- Find pixel value for its center point
- Convert pixel values to coordinates of the object
- Repeat steps 1, 2 and 3 twice to find three different points at \( t_0, t_1 \) and \( t_2 \)
- Calculate \( a_x, b_x, c_x, a_y, b_y, c_y, a_z, b_z \text{ and } c_z \) (the parameters of the equation of motion)
- Extract information from the equation of motion (target and source of object)

3. FIND THE CENTER OF THE OBJECT

Once the object is detected, processing begins to find its center, taken to be the center of the rectangle which constitutes the convex hull of the object.

a. Background is removed (pixel by pixel subtraction) from the image. Note here that after subtraction is complete, the resulting image contains negative values, which may cause problems. Absolute values of the difference should be taken for all the pixel values of the resulting image.

b. Enhancement should be made to the result by taking a mean filter: this filter adds the pixel value to all of its eight neighbors and divides it by nine to get an enhanced image (delete undesired values).

c. Perform “thresholding” on the resulting image to convert it into a binary image as such:

\[ g(x, y) = \begin{cases} 
0 & \text{if } 0 < f(x, y) < 50 \\
1 & \text{if } 50 < f(x, y) < 255 
\end{cases} \]

d. Further enhancement is done to the binary image to remove undesired spots. This is achieved though the erosion tool. Erosion is a tool that removes pixels from the foreground that touches one or more pixel from the background.
e. Finding the Center Point. In this step, a very fast algorithm has to be used to obtain a precise center of the object. This algorithm should be fast to ensure that there is enough time to take six shots (3 – 3D images) and process them.

Traditional algorithms state that a scanning begins from each edge of the image for all the pixels and stops once it hits a pixel different from the background values (in our case 1’s). Such algorithms perform approximately 640 * 480 – 20*640 – 20*480 scans and comparisons (where we suppose that the average size of the object is 20x20 pixels). The algorithm developed in this work decreases this number as follows: Vertical lines are scanned in steps of 5 pixels i.e. each scanned vertical line processed by scanning its pixel values in steps of five. If no white pixel is detected, another adjacent vertical line is scanned where this line is 5 pixels far from the previous one and the process is repeated until a white pixel is detected as shown in figure 1.
Therefore, the left edge of the image is now between the current and previous vertical line. At this step, the horizontal line is scanned at the same y – position as the detected pixel in order to locate the first edge point on that line. However, it is not necessary to start scanning from the edge of the line, a scan can start at the x – position of the previous vertical line since it is sure that no white pixels exist before that line.

Pixel by pixel scanning is made on this horizontal line to detect the first edge point. X1, X2, Y1, Y2 are set to this point where X1 and X2 are the vertical edges of the object while Y1 and Y2 are the horizontal edges of the object. Processing of the pixels of the horizontal line continues to find the position of the right edge on this line. At this stage, X2 is updated. Iterations have to continue downwards and each time X1, X2 and Y2 are updated to get the minimum X1 and the maximum X2 and Y2. Figure 2 describes the process.

Moreover, iterations have to be made upwards to cover the whole object. Here, X1, X2 and Y1 are updated to get the minimum X1, Y1 and maximum X2.

Finally, as shown in figure 3, the center is found by taking the center of the rectangle formed by X1, X2, Y1 and Y2.

As we can see from the above discussion, this algorithm reduces the number of calculations dramatically. The average number of pixel calculations can be found as follows: (320/5)*(240/5) + 20*20 = 3472. A comparison to the value obtained from the traditional algorithm (640*480 – 20*640 – 20*480 = 284800) shows that the number of calculations has been reduced by a factor of about 82.

This procedure returns four values for X1, X2, Y1 and Y2. The center is taken to be at: Xc = (X1 + X2)/2 and Yc = (Y1 + Y2)/2. To each 3D image are associated two pairs of coordinates for the center from the two cameras: (X1c, Y1c) from camera1, and (X2c, Y2c) from camera2.

4. CONVERT IMAGE COORDINATES TO SPATIAL COORDINATES

The only spatial coordinates available are the coordinates of the two cameras. Below is a brief discussion of how the pixel coordinates of the center point are converted to spatial coordinates.

Each camera is taken as the center of a spherical system and a line is drawn from the camera to the center point. The equation of this line is easy to find in spherical coordinates since only the angles have to be determined. Hence two equations of straight lines are found in different coordinate systems. These straight lines are transformed to another common rectangular system and their intersection is located. This intersection point is the center point of the object. However, and due to the approximation in finding the center point using the two cameras and due to the delay between the shots of the two cameras, the above-mentioned straight lines may not intersect. Consequently, some approximation has to be made to find the desired point. Here, the midpoint of the segment perpendicular to both straight lines is found. This midpoint is a good approximation of the center point. Theoretical calculations define the following:

A - The equation of the first line is determined using the equation of a straight line is given by: X = a t + x0, Y = b t + y0, and Z = c t + z0.

Given α1, α2 (the vision angles of the camera), and a1 and b1 (the pixel coordinates of the center) such that α1 > α2. X, Y and Z have to be found as shown in figure 4.
First we have to transform the centers of coordinates from the lower left to the middle of the picture:
\[ a_1' = a_1 - 320, b_1' = b_1 - 240. \]
If we project the straight line joining the camera lens and the center of the object on the XY plane we obtain \( D_1' \). Similarly, \( D_1'' \) is obtained by projecting the same line on the YZ plane. Let \( \Theta_1 \) be the angle between \( D_1' \) and the Y axis, and \( \Phi_1 \) be the angle between \( D_1'' \) and the Y axis.

Transform this equation into main coordinates:
\[
\begin{align*}
D_1' & : X = (\cos \phi_1 \tan \theta_1) t + x_{01} \\
& \quad Y = (\cos \phi_1) t + y_{01} \\
& \quad Z = (\sin \phi_1) t + z_{01}
\end{align*}
\]
Where \( x_{01}, y_{01} \) and \( z_{01} \) are the coordinates of camera 1.

B - The equation of the second line is determined in a similar fashion.

C - In order to determine the segment that is perpendicular to the two lines, the reasoning shown in figure 5 is used:

Figure 5. Finding the segment perpendicular to the two lines

The equation of the two straight lines is given by:
\[
\begin{align*}
D_1' & : X = a_1 t + x_{01} \\
& \quad Y = b_1 t + y_{01} \\
& \quad Z = c_1 t + z_{01} \\
D_2' & : X = a_2 m + x_{02} \\
& \quad Y = b_2 m + y_{02} \\
& \quad Z = c_2 m + z_{02}
\end{align*}
\]

Use the determinant of the two vectors parallel to the two lines in order to calculate the coordinates of the vector perpendicular to both \( D_1 \) and \( D_2 \).

Let \( K_1 = b_1 c_2 - b_2 c_1, K_2 = c_1 a_2 - a_1 c_2, K_3 = a_1 b_2 - a_2 b_1 \). Taking one point on \( D_1 \) and one point on \( D_2 \), they verify their respective equations. By forcing the line passing through these two points to have a direction parallel to the vector \( V \), \( (K_1, K_2, K_3) \); which is perpendicular to \( D_1 \) and \( D_2 \); the coordinates of these two specific points can be determined. Thus, the center point of the flying object can be approximated to be the midpoint of the segment joining these two points.

The center point is approximated to be the midpoint of the segment perpendicular to the two straight lines. Therefore, the coordinates of the center point are given by the following equations:

\[
\begin{align*}
\text{x mid} & = \frac{a_1 + a_2}{2} \\
\text{y mid} & = \frac{b_1 + b_2}{2} \\
\text{z mid} & = \frac{c_1 + c_2}{2}
\end{align*}
\]
\[ X_c = \frac{a_x t_0 + x_{01} + a_2 m_0 + x_{02}}{2} \]
\[ Y_c = \frac{b_y t_0 + y_{01} + b_2 m_0 + y_{02}}{2} \]
\[ Z_c = \frac{c_z t_0 + z_{01} + c_2 m_0 + z_{02}}{2} \]

5. FIND THREE DIFFERENT POINTS AT \( t_0, t_1 \) AND \( t_2 \)

No information is obtained from finding the center point only once. Three different positions must be calculated at three different times \( t_0, t_1 \) and \( t_2 \) respectively to get three points and therefore deduce the equation of a parabola. Once the processing of the first 3D image is finished, another pair of images (3D image) is taken immediately. \( t_1 \) is considered to be the time between the first image and the second (\( t_0 = 0 \)). When the processing of the second image finishes, the current time is saved and set as \( t_2 \) then a third image is taken and processed. Here three points are obtained: \( C_0 (X_0, Y_0, Z_0) \) at \( t_0 = 0 \), \( C_1 (X_1, Y_1, Z_1) \) at \( t_1 \), \( C_3 (X_3, Y_3, Z_3) \) at \( t_3 \). Three points are needed to find the equation of motion of the object because the path followed by that object is a parabola. The equation of the parabola passing through these three points at \( t_0, t_1 \) and \( t_2 \) respectively is defined in the following paragraph.

6. FIND EQUATION OF MOTION

The equation of motion is supposed to be a parabola (projectile motion), where the corresponding general equation is given by:
\[ X = a_x t^2 + b_x t + c_x \]
\[ Y = a_y t^2 + b_y t + c_y \]
\[ Z = a_z t^2 + b_z t + c_z \]
Where \( a_x, b_x, c_x, a_y, b_y, c_y, a_z, b_z \) and \( c_z \) are to be calculated.
\[ a_x = \frac{t_1 (y_2 - y_0) - t_2 (y_1 - y_0)}{\Delta} \]
\[ a_y = \frac{t_1^2 (y_2 - y_0) - t_2^2 (y_1 - y_0)}{\Delta} \]
\[ c_z = x_0; \quad \text{and,} \quad \Delta = (t_1^2 t_2 - t_1^2 t_1) \]

Hence, all the unknown parameters of the parabola has been calculated, and the equation of motion is well defined.

7. ESTIMATE THE SOURCE AND TARGET PROJECTILE

Once the equation of motion of the flying object is found, much information can be extracted from it. The source and target of the flying object can be easily calculated once the map is given. The origin with respect to which the camera coordinates are measured lies on the ground. This implies that the equation of the ground is \( Z = 0 \). Eventually, the source and the target of the flying object are the intersection of the equation of motion and the plane \( Z = 0 \).
\[ Z = a_z t^2 + b_z t + c_z = 0 \]
This gives two values for \( t \). One is negative and the other is positive. Replacing the negative value of \( t \) in the equation of motion will give the coordinates of the source. Replacing the positive value of \( t \) in the equation of motion will give the coordinates of the target.

8. RESULTS

By using only two digital cameras, source and target estimation were made possible just by using the digital image processing tools and some knowledge of math. Our results give a closed form solution for target and source position estimation of a flying object given the pixel coordinates of its center at three different time instants. This closed form solution is very useful since it can be implemented using almost any software and it contains pure mathematical calculations once the pixel values of the center of the object are known. Practical experiments were carried out using a ball as the projectile. The target and source of the ball were estimated with an error not exceeding 7%. The error due mainly to the following reasons: a) The center of gravity is an approximate one. B) the delay between the shots taken by the two cameras.

9. ACKNOWLEDGMENTS

Thanks are due to Mr. Kamal Mikati, supervisor of computer laboratories, and Professor Fadl Moukalled, Chairman of the Mechanical Engineering Department at AUB, for providing us with needed hardware. This work was supported, in part, by the University Research Board of American University of Beirut.

REFERENCES
