Defence Science Journal, Vol. 60, No. 3, May 2010, pp. 238-243

2-D Shape Fitting for Locating Exploding Projectile from Explosion Patch

Kunal Ray

Proof and Experimental Establishment, Chandipur-756 025
E-mail: kunalrayind@gmail.com

ABSTRACT

For test and evaluation of a proximity fuze, it is necessary to know the distance offset of the exploding ammunition round, fitted with the fuze, from a specific target. If the event is recorded using in-line high-speed photography the event of explosion can be resolved in time, and it becomes necessary to ascertain the position of the round, wrt the target, as it is exploding. An estimation of intensity centroid position fails as the flash is non-uniform in nature and is partially occluded by the exploding round. This paper is about an approach to find the location of the round using 2-D shapes fitting of the explosion patch.

Keywords: Proximity fuze, variable-time fuze, explosion patch

1. INTRODUCTION

A proximity fuze or variable-time fuze (VT fuze) is a fuze that is designed to detonate an explosive shell or a missile warhead automatically when the distance to target becomes smaller than a predetermined value or when the target passes through a given plane. These fuzes are used in ammunitions aiming to defeat airborne targets. Since it is impossible to physically hit such targets, except by chance, impact fuzes can not be used to initiate the shell or the warhead. The proximity fuzes initiate the shell when the target is close enough to be defeated by the exploding shell. There are different types of proximity fuzes depending on the sensing principle like radio frequency sensing, optical sensing, acoustic sensing, magnetic sensing, pressure sensing, etc. In this paper, proximity fuze named 'FUZE FB 40' working on the principle of radio frequency sensinghas been used. This type of fuze contains a self-powered radio transmitter-receiver operating on Doppler principle.

The testing of proximity fuzes involves determining whether these can initiate the explosion of a shell within a specified distance from a target. The standard methodology undertaken for testing 'FUZE FB-40' involves erection of a dummy missile target, conforming to the RF signatures exhibited by a sea skimming antiship missile, at a specified distance from the gun. Shells fitted with the fuze are then fired at the erected target. An in-line high-speed camera records the event. The shells are supposed to explode only when these are within a specified zone around the target. The trial requirement was to determine the positional offsets of the exploding shells from the target body from the high-speed video recording of the event. These can be found only if the location of the exploding round can be determined from the recorded data. A standard method to locate exploding round from within an explosion patch is a measure of intensity-weighted centroid. However, because of non-uniform intensity distribution in the explosion patch and occlusion by the disintegrating round, intensity-weighted centroid of the patch can't give a proper estimate of the location of the exploding round. As the explosion always takes place in a spherical wavefront, the shape of the patch can be exploited to locate the shell. This paper discusses use of optimal shape fitting of the patch to locate the shell.

Shape is a fundamental image feature and is known to be used for recognition and description of objects. The shape of objects plays a major role in human recognition and perception[1]. Typically shape features are exploited for automated object detection and classification. Exploitation of shape features is also useful for 3-D reconstruction from sparse data[2]. Among various types of shapes, identification of circles and ellipses play important roles in biomedical image analysis[3], and target detection[4,5]. Often fitting of circular or elliptic shapes to object points are necessary. A very effective method for circle fitting is using the Hough transform for circle detection[6]. This is a very attractive option as it can easily detect even occluded objects. However, a major problem of Hough transform is the computational complexity which rapidly goes up as one moves from simple line detection to circles and other complicated shapes[7].

Typically, most methods for finding predefined shapes from images depend heavily on border points. Recent works have shown that even from random choice of border points, shape context-based descriptors can be evolved and used for shape matching and object recognition[8]. However, if there are strong discontinuity in them along with an unpredictably irregular and possibly re-entrant contour in the object, such methods either will not work or will need highly complex modifications. In such cases, one needs to use methods

Received 8 May 2009, Revised 24 August 2009

that are less dependent on the border. One such method is proposed by Chaudhuri[9] where the object area is used for fitting an optimised circle or ellipse. The method proposed to generate estimate of location of the exploding projectile has been used here.

2. METHODOLOGY

A common approach for finding the position of the shell in an explosion patch involves determination of intensity centroid of the pixels forming the patch. This is essentially the weighted average of locations of the pixels forming the patch with the weight factor being the gray scale intensity of the pixels. The assumption made for this procedure is that the intensity monotonically decreases from the source, i.e., the round, in all directions. This holds true if the explosion is recorded using a normal speed video camera (speed between 25-30 frames/s). Since the process of explosion is very fast compared to the 40 to 33 ms frame time, the recorded data basically contains averaged out information about the event where the above assumption holds. However when high-speed cameras operating at thousands of frames/s are used, the event of explosion is resolved in time and the assumption no longer holds. Further, a major problem of occlusion of the flash patch by the still disintegrating round was faced (Fig. 1, the dark shape with fins is the dummy missile target). The explosion, as it happens in mid-air, should expand with an approximately spherical wavefront, whose cross section in x-y plane, as recorded by the camera, should be roughly circular.

Figure 1. Exploding round fitted with proximity fuze.

To find the possible location of the shell, an optimal circle was tried to fit in the flash patch, whence the centre of the circle gives an estimation of location of the exploding shell. Practically, the explosion wavefront will only approximately be spherical because of non-uniformity in mechanical properties of the shell body. This deviation was assumed to be small compared to the rest of the wavefront. This assumption was found to be holding in general. Further there can be elongation in vertical plane due to gravity. The gravity elongation is negligible in this case as imaging was done at a very high frame rate (at least 1000 frames/s). To perform the shape fitting the steps taken were: (a) segmentation, (b) region identification, (c) fitting the circle, and (d) optimality.

2.1 Segmentation

Before any thing else the flash area was to be segmented from the background. Segmentation can be done using many techniques like thresholding-based, edge-based, region-based, etc. In this case, edge-based techniques won't work properly because of irregular occlusion. Further, the segmentation process was needed to be very fast to have other subsequent processes to get result. So thresholding-based methods were chosen. It was observed that working with intensity value of the pixels (converted from the RGB values) was not suitable. It was further seen that if HSV space was used and only the hue value was looked, the flash could be easily separated from the background (Fig. 2). In threshold-based methods, choice of threshold is very important. Various techniques exist for determining choice of thresholds[10]. To perform actual segmentation, an iterative (optimal) thresholding[11] scheme was used to determine a nearly optimal threshold value for hue channel. This works well even if the image histogram is not bi-modal. The technique can be understood in the following steps:

(a) As first approximation, take four corner pixels in the image to belong to background.
(b) At step t, compute mean object gray level μ_o^t and background gray level μ_b^t based on threshold T^t for step t-1 (Eqn (2)).

Figure 2. Hue channel information of the flash region.

$$\mu_b^t = \frac{\sum_{(i,j)background} f(i,j)}{\#background\_pixels} \quad \text{and}$$

$$\mu_o^t = \frac{\sum_{(i,j)object\_pixels} f(i,j)}{\#object\_pixels} \tag{1}$$

(c) Set $T^{(t+1)} = \frac{\mu_b^t + \mu_o^t}{2}$ (2)

$T^{(t+1)}$ now provides updated threshold value.

(d) Stop when $T^{(t+1)} = T^t$. Otherwise go back to step (b).

An objection to this method may come as it deals in one parameter only when one has three parameter colour image. However it has been seen (as mentioned before) that by moving on to HSV space[12] and using the hue parameter, only suitable segmentation could be got for the purpose.

2.2 Region Identification

After the segmented image was generated, a region finding algorithm was applied which serves a number of purposes. It finds out: (a) the regions based on eight neighbourhood connectivity, (b) the coordinates of pixels belonging to individual regions, and (c) the approximate area of each region represented by the pixel populations of them.

The region finding can be done using either recursive search and modify algorithm or a list-based algorithm. The recursive algorithm can be understood in following steps:

(a) Scan the region of interest for first white pixel.
(b) Modify the pixel to a different value.
(c) Call the search algorithm eight times with locations of eight neighbours of the detected pixel as parameter (for each detected pixel these three steps are repeated).
(d) During the procedure, store the locations of the pixels belonging to the region.
(e) Rescan the region of interest after region finding is over for the next region.

This algorithm is fast, easy to understand and implement, and efficient but the problem is that if the region is too large, there can be stack overflow.

A better solution is a list-based algorithm as follows:

(a) Scan the region of interest and store locations of all white pixels to a list and modify the pixels to a different value.
(b) Move through the list and for each entry, search its eight neighbours.
(c) Any white pixel found, modify that pixel and append the location to the end of the list.
(d) After all neighbours have been checked, move to the next entry in the list.
(e) Continue till the end of the list.

This algorithm is not as efficient as the previous one as it is actually violating locality of reference by appending new entries at the end of the list. However, if relevant entries were tried to be inserted, a major overhead will be incurred in shifting down the rest of the entries of the list. Still that option can be implemented if instead of using iterated arrays for lists, the doubly connected linked lists are used.

2.3 Fitting the Circle

To fit a circle to a region, one needs the centre of the circle and its radius. Let us consider a flash patch A in the region. Let (x_j, y_j), $j = 1, 2, 3\ldots n$ be n pixels belonging to the entire flash patch. Then the centroid (X, Y) of the region is defined by:

$$X = \frac{1}{n}\sum_{i=1}^{n} x_i \; ; \quad Y = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{3}$$

This centroid of the region, given by the averaged positions of its population, can be taken as the first choice for the centre of the circle to be fit. The centroid can also be found using border points only and that centroid can also be used as the first choice[13] (but as the patch can be highly irregular, the centroid based on border points only can be a bad choice). This choice is widely proposed in literature and can serve as a good first choice.

Next, there is need to find the radius. An easy choice is the half of the distance between the most widely spaced points of the region. This will give an upper estimate of the fit. This can be refined by considering the pairs of points placed diametrically opposite of each other and take the mean distance between these or the most common distance between these or maybe the average of highest and lowest of these. That is to say if L_i, $i = 1, 2, 3\ldots k$ be k such distances between the pairs, then the radius R can be

$$\begin{aligned} & R = \text{mean } (L_i) \\ \text{or} \quad & R = \text{mode of values of } L_i \\ \text{or} \quad & R = \text{mean (maximum and minimum of } L_i) \end{aligned} \tag{4}$$

This approach is simple and easy to implement. However this will not work well for highly irregular or re-entrant surfaces. Another way to find the radius can be through fitting of border data to equation. Let (x_j, y_j), $j = 1, 2, 3\ldots N$ be N pixels belonging to the border of the region only. If the centroid from Eqn (3) is taken as centre of the circle, then the border points should satisfy

$$R^2 = (x_k - X)^2 + (y_k - Y)^2 , \; k = 1, 2, 3 \ldots N \tag{5}$$

From this equation, the value of R can be estimated by numerical analysis. However, firstly appropriate border points are needed to be chosen which increase computational load as well as complexity because a measure of appropriateness has to be defined; and secondly after getting the appropriate border points generating R will require considerable computation. Chaudhuri[9] has suggested a very simple but effective mechanism to find the optimal radius by matching the area of the shape to be fitted with area of the region. If n is the population of the flash patch A, then radius will be:

$$R = \sqrt{n/\pi} \tag{6}$$

For this choice, the tricky question of appropriate border points is eliminated and this estimate of radius should be independent of noise signature in border as well as broken border signature due to occlusion.

2.4 Optimality

After the choice of radius the question comes of optimality of the location of the centre of the circle. This is very important, as this centre point will be the result of the exercise namely the estimate of the location of the exploding shell. To optimise the location of the centre from the first choice (Eqn (3)) of centroid, an iterative non-optimality checking algorithm was used. A possible measure of non-optimality of the fitting circle can be the size of the mismatched region between the flash and the circle. The mismatch can be the sum of the number of pixels belonging to the flash patch but outside the fitting circle and the number of pixels inside the circle but not part of the flash patch. That is to say if A is the area of the flash patch and B is the area of the fitted circle then:

$$\begin{aligned} &Non-Optimality = Total\_Non\_Matching\_Area \\ &Non-Optimality = \#[A-(A\cap B)]\cup[B-(A\cap B)] \end{aligned} \tag{7}$$

This may be normalised by dividing with $A\cap B$; however for application here, this will not be necessary. Now the centroid point can be taken as the starting point and this sum can be computed for this choice. The sum can then be computed for its eight neighbours considering these to be possible centres. The neighbour for which the non-optimality sum was lowest can then be taken as the next choice for the centre and the process repeated for its neighbours and so on till the minimum value does not change any more. Two possible problems can be there with this scheme: (a) this can be quite expensive computationally, and (b) there can be run-away iterations.

The first problem is not very important in the present case because the area of interest is small. For bigger targets remembering the direction of move between choices can eliminate six cycles of sum computation for vertical and horizontal movements and four cycles of sum computation for diagonal movements. In Figs 3(a) and 3(b), L is the lowest-valued neighbour. After move, L is the centre pixel. For pixels marked with # as also for the centre pixel, the sum is known from previous step. These can reduce the computation cost considerably. The second problem can be solved by not allowing more than a preset number of iterations to take place. In Fig. 4 two possible cases that can be encountered (black arrows indicate possible non-matching areas) are shown.

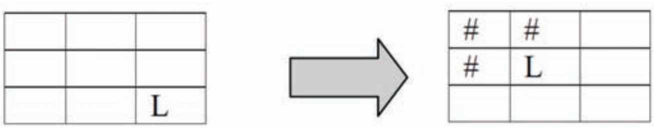

Figure 3. (a) Diagonal movement.

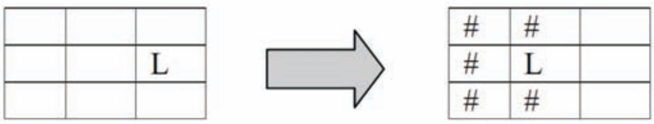

Figure 3. (b) Horizontal movement.

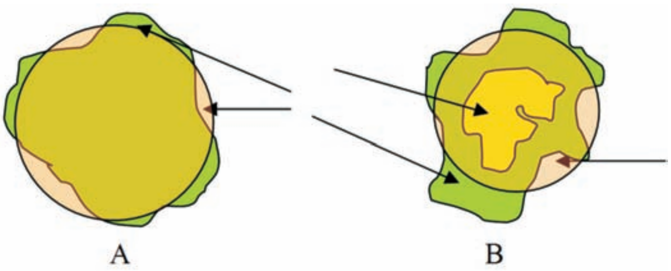

Figure 4. Two possible shape-fitting situations.

For the shape A, there are no problems and minimising the sum of non-matching pixels will almost certainly give rational result. However for shape B, the non-matching pixels due to the irregular hole inside the object may actually move the shape away from a proper solution, particularly if there are multiple holes of differing size and non-uniform distribution. In the present case, post-segmentation shapes similar to shape 'B' may be encountered (compare Fig. 2). To avoid this possible error, non-matching pixels inside the circle have not been considered as valid and only the pixels that are outside the circle but belonging to the patch in the sum of non-matching pixels have been considered. It has been found experimentally that in most cases, iterations of no more than four steps are necessary to reach the optimal centre. However, for grossly irregular shapes higher number of iterations may be required.

Figures 5 and 6 show, respectively, the segmented patch and the fitted circle in green with centre indicated in red. The shape of Fig. 4 indicates that if the pixels from the inner hole had been considered for optimality, the fitted circle would have shifted more towards left, i.e., towards the target, which would have been an improper result. Figure 7 shows the result of fitting for the image shown in Figs 1 and 2 (in an enlarged view).

Figure 5. Segmented patch.

Figure 6. Fitted circle in green and centre in red.

Figure 7. Enlarged view of the fitted circle in green and centre in red for the image in Fig. 1.

3. DISCUSSION AND RESULTS

The fitting technique is shown to provide a stable mechanism for estimating the location of an exploding shell. The success of this technique depends not only on the optimised fitting parameters but also on the quality of segmentation and region finding. In some cases, the segmentation output can give disjoint regions, which together form the patch. In those cases, it may be useful to pre-process the segmentation output using morphological operators to merge the regions.

However, use of morphological operators in the present case requires caution as they may cause major modification in the overall shape of the explosion patch. A possible solution can be the use of morphological operators for regions well inside the signature. However, the possible computation cost for identifying such eligible regions should not be overlooked. Otherwise use of such operators may result in improper shift of the centre of the fitting circle. Various types of margining techniques can also be used depending on the situation at hand and after due consideration about the possible effect on centre of fitting circle.

In the developed procedure, user is supposed to select out an area covering flash patch with the mouse and over that area, the procedure is to run. The computation cost of the entire procedure comprises three parts. The cost for segmentation must include the cost for the procedure for conversion from RGB space to HSV space. This is high because the generation of hue information requires a number of floating-point calculations including square root. However this is completely dependent upon the selected area. The cost for segmentation procedure is dominated by the cost for identifying the optimal threshold value. This can be quite large as it is an iterative process. However in the present case the flash region is of the order of hundred pixels squared only. So it is hoped for not too large a value. Experimentally it was seen that the time taken for the threshold selection procedure does not go up if a larger area is selected. This is to be expected because the region of interest remains of the same size. The procedure for region identification is also costly, however, the use of a recursive algorithm should bring the cost down, and for the present case of small regions, this may not be very high. Finally, the cost for circle identification and optimisation can be high if the starting choice of the circle centre is poor.

One possible approach that has not been explored is the use of edge detectors for the initial steps. This is because the highly irregular and re-entrant character of the flash patch can give rise to a lot of inner edge points, which should be discarded. Further, presence of smoke and fog can also introduce spurious edge points. Removal of these spurious outer and inner edge points is not a trivial task and can increase complexity to a great extent. For this reason, edge detection-based methods were not used.

The computation time taken by each of the procedures has been experimentally studied and shown in Table 1. The application software has been written using Microsoft Visual C++ 6.0. The time estimates were found by the program running in debug mode in a 2.63 GHz Pentium 4 machine, running Windows XP service pack 2. During this estimate, antivirus software was disabled and no other program was running in the foreground. It is seen that even in the worst case, the entire procedure should not take more than 200 ms. It was observed during the estimate that for images with good contrast and relatively regular shapes, the time taken for circle fitting and optimisation came down to be below 20 ms, and in case of hazy images with poor contrast, the corresponding time taken went up

Table 1. Experimental estimates for time taken in procedures

Procedure	Average time taken (ms)
Generating hue information	50-60 (varies with area selected)
Optimal thresholding	30-35
Counting regions	15-20
Circle fitting and optimisation	35-45

to as high as 90 ms. The time for the other procedures did not show such variations.

As mentioned above, possible problem may come if the acquired image is found to be hazy due to the presence of fog or smoke from the gun. In such cases, the simple segmentation procedure used may be inadequate. So need will be for sophisticated algorithms or to enhance the image as a pre-processing step. An advantage in such cases can be the fact that the object of interest is self-luminous, so simple histogram-based enhancement techniques or adaptive neighbourhood processing techniques should be sufficient[14,15].

4. CONCLUSIONS

An approach to estimate the possible location of an exploding round from the image of explosion patch by fitting optimal circle has been proposed. The steps for the process, namely segmentation using hue-based optimal thresholding, region identification through recursive (or list-based search), and fitting of the circle through area matching and shape centroid estimation are very simple. The optimality criteria used for estimating the radius of the fitting circle and for estimating the location of the centre of the fitting circle are simple but effective and are easy to implement. The procedure is seen to give good result. The computational cost of the entire process is also not high.

ACKNOWLEDGEMENTS

The author wishes to acknowledge Maj Gen Anoop Malhotra, Director; Mr T.K. Biswal, Sc F; and Mr A. Bose, Sc E, Proof & Experimental Establishment, for rendering support and encouragement and Dr Debashish Chaudhuri, Sc E, Defence Electronics Application Laboratory, for rendering technical support.

REFERENCES

1. Hoffman, D.D. & Richards, W.A. Parts of recognition. *Cognition*, 1984, **18**(1-3), 65-96.
2. Lempitsky, V. & Boykov, Y. Global optimization for shape fitting. *In* Proceedings of IEEE Conference on CVPR, Minneapolis, June 2007.
3. Paton, K.A. Conic sections in chromosome analysis. *Pattern Recognition*, 1970, **2**, 39-51.
4. Zhu, C. & Wang, R. A fast automated extraction algorithm of elliptic object groups from remote sensing images. *Pattern Recognition Lett.*, 2004, **25**(13), 1471-478.
5. Zhang, S.C. & Liu, Z.Q. A robust real-time ellipse detector. *Pattern Recognition*, 2005, **38**(2), 273-87.
6. Duda, R.O. & Hart, P.E. Using the hough transforms to detect lines and curves in pictures. *Commun. ACM*, 1972, **15**(1).
7. Evans, F. Survey and comparison of the hough transform. *In* IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management, New York, 1985. pp. 378-80.
8. Belongie, S.; Malik, J. & Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, April 2002, **24**(24).
9. Chaudhuri, D. Object area-based method for elliptic and circular fit of a two-tone image. *Def. Sci. J.*, 2008, **58**(6), 710-14.
10. Sahoo, P.K.; Sultani, S.; Wong, A.K.C. & Chen, Y.C. Survey of thresholding techniques. *CVGIP*, 1988, **41**(2), 233-60.
11. Image processing, analysis and machine vision. edited by Sonka, Milan; Hlavac, Vaclav & Boyle, Roger. Ed. 2. pp. 129-30.
12. Digital image processing. edited by Gonzalez & Woods. Ed. 2. pp. 299.
13. Haralick, R.M. A measure of circularity of digital figures, *IEEE Transactions*, **SMC 4**, 1974.
14. Gordon & Rangayyan. Feature enhancement of film mammograms using fixed and adaptive neighbourhoods. *Applied Optics*, 1984, **23**, 560-64.
15. Pizar, *et al*. Adaptive histogram equalisation and its variations. *CVGIP*, 1987, **39**, 355-68.

Contributor

Mr Kunal Ray obtained his BSc (Physics) from Serampore College, West Bengal, in 2001, and MSc (Physics) from University of Calcutta, West Bengal, in 2003. He is presently working as a Scientist at the Proof and Experimental Establishment, Chandipur, Orissa, India.