IMAGE PROCESSING FOR ACQUIRING TARGET COORDINATES WITH A PLATFORM

JĂNEL ARHIP, ADRIAN BORLAN1, OCTAVIAN GRIGORE-MÜLER2

Key words: Image data processing, Image classification, Target acquisition, Data fusion.

In the modern battlefield there are used many methods to hide personnel, equipments and vehicles. Most of them are using natural (vegetation) and artificial (masking, netting) obstacles. Moreover, there are some situations when conditions like night and bad weather may make difficult or impossible to visualize them with visible spectrum sensors. In these cases a stabilized platform with complex sensor (visible and infrared camera) and LASER telemeter are used. In this paper complex data (image) processing and data fusion, used not only to find this kind of targets, but also to identify (classify) them, are described.

1. INTRODUCTION

A multisensor platform is used to discover, identify and establish the position of the targets in adverse conditions (bad weather or low exposure due to vegetation, netting, or masking). In order to achieve this purpose, it is equipped with the following sensors:

1. Electro-optical sensors like CCD-Video and FLIR;
2. Laser telemeter;
3. Global positioning receiver (GPS);
4. Angles detection equipment like inertial inclinometer;
5. North finder receiver for localize the North;
6. Computer process interfaces with digital and analogue high-speed interfaces;

By FOV it means the field of “view” of the acquisition electro-optical CCD camera (Fig. 1). Its main role is to detect the target’s presence by image analysis. For acquiring the target parameters, classification and tracking in adverse condition an infrared camera (FLIR) assisted by a laser telemeter is used.

The field of view of the infrared camera has the same dimensions like FOV, allowing the electronic movement of an image analyzing “window” with smaller

1 SC SELETRON Software & Automatizări SRL, Str. Bucur 4-6, E-mail: seletron@seletron.ro
2 “Politehnica” University of Bucharest/Faculty of Aerospace Engineering, Bucharest, E-mail: octavian.grigore@gmail.com,octavian_grigore@avionics.pub.ro
size (W×H dimensions) inside FOV, in order to be centered on the target. Automatic image analysis is performed in order to obtain the target coordinates relative to the window coordinates. Also using the distances measured by laser telemeter, a target classification is performed. Absolute target coordinates (geographical coordinates) are finally founded in absolute terms based on the information from the image receiver, from GPS, from the system transducers for orientation angles of the multisensor platform and from the North finder receiver.

System schematic is as shown in Fig. 2:

2. PROCESSING IMAGES FROM FLIR TYPE CAMERA

Image obtained through the FLIR sensor, as it seen in Fig. 3, can be represented as a continuous function of spatial objects temperatures \( f(x,y) \), where \( x \) and \( y \) coordinates are in the continuous two-dimensional geometric image space.
Sampling on a rectangular matrix ($N \times M$) through $K$ quantification levels, leads to a representation in the discrete function form: $f_i(x,y)$, where $f_i$ takes discrete values in the range $0...(K-1)$ and $x$ and $y$ represents the discrete values in the fields $0...(N-1)$ on the horizontal dimension, and $0...(M-1)$ respectively on the vertical dimension of the image, separated by intervals $d_x$ and $d_y$. Thus an image can be represented by a matrix of $N \times M$ size, whose elements (called pixels) are vectors expressed as $p_{ij} = (x_i, y_j, f_{ij})$, where $i = 1...(N-1), j = 1...(M-1)$, and $f_{ij}$ can take values between 0 and $(K-1)$.

In our case, for a standard PAL interlaced format image, $N = 720$, $M = 624$, $K = 256$.

![Fig. 3 – Test image on: a) CCD camera; b) FLIR.](image)

Image processing is performed in real time, using a PC/104 standard board, with the schematic as is shown in Fig. 4.

![Fig. 4 – The schema of a PC/104 standard board.](image)

Image analysis is performed inside and outside a “gate” window superimposed on the image, whose coordinates and dimensions are continuously modified depending on the target evolution inside the “gate” as shown in Fig. 5.
For computation of the “movement vector” of the image an analyze in a fixed frame of 512×256 pixels centered on the image is performed.

Image processing has the following steps:
- Histogram computing in “gate” window and outside;
- Image segmentation inside “gate” window based on a cost function;
- Filtering the binary image;
- Computing the target geometric parameters;
- Classification of the target.

The temporal sequence of the algorithm is depicted from the diagram shown in Fig. 6.

As it can be seen from Fig. 6, the image analysis and segmentation are performed in real time, with a delay of only 20 milliseconds (period of a TV field),
because are used probability values computed based on the information contained into the previous TV field.

**Computing the target and background probabilities.** These probabilities are derived from the image histogram analysis, performed inside “gate” window and in a surrounding area, whose surface (number of the pixels included) is equal with that of the “gate” window. The size and dimensions of the “gate” window are received by video processor unit (VP) from main computing unit (MCU) before the start of a new TV frame. In the same time is applied a dynamic optimization based on the “movement vector” computation for every pixel in the gate window. Imposing in the formula

\[ n \times m = 2 \times n \times x + 2 \times m \times y , \text{ where } x = m / 4 , \text{ it will results } y = n / 4 . \]

Fig. 7 – a) Background histogram; b) “gate” window histogram.

By mean of the histogram analysis are generated two vectors: \( b_k(v_0, v_1, ..., v_{255}) \) and \( t_k(t_0, t_1, ..., t_{255}) \), where \( v_k \), respective \( t_k \) are normalized values of the pixels number on every quantization level \( k \). It’s obvious that this values lies inside \((0, 1)\) interval. The probability for a pixel inside “gate” window to be part of a target is computed like: 

\[ p_{i,j} = 1 - b_k(i, j) / t_k(i, j) , \] 

where \( b_k \) and \( t_k \) are the \( b_k \) and \( t_k \) vectors component for the quantization level \( k \) limited to the proper values (in order to maintain \( p_{i,j} \) in the \((0, 1)\) interval), \( i \) and \( j \) are pixel’s relative coordinates to the “gate” left top corner. In the dynamic case (when tangential speed of the target exceeds a threshold) is applied also an optimization operation:

\[ p_{i,j} = p_{i,j} \times (1 + |\text{diff}| / 256) , \] 

where \( |\text{diff}| \) is the modulus of the difference between quantization levels of the pixels in the same position from the \( n-1 \) and \( n \) image fields (the “movement vector”).

**Image segmentation and enhancement.** The probabilities computed using the mentioned algorithm, for every pixel inside “gate” window, are used to obtain a binary image in the gate. This means that we’ll maintain only two levels: “1” for those pixels \((i, j)\) with a \( p_{i,j} > \text{threshold} \) (means that they are considered part of a target) and “0” for background pixels. A second operation for enhancement of the
binary image consists in filtering, in order to eliminate noise and to eliminate also undesired “holes” from the final binary image [6, 7, 8]. Both operations use a matrix of 3×3 pixels centered on the pixel under processing. Noise removal is achieved if \( p_{ij} \) get “0” value for the situation presented in Fig. 9a (low pass filtering). Target “cropping” is obtained if \( p_{ij} \) get the “1” value for the situations presented in Fig. 8 (x doesn’t care).

\[
\begin{array}{ccc}
X & 0 & X \\
0 & Fij & C \\
X & 0 & X
\end{array}
\]

\[
\begin{array}{cccc}
1 & 1 & 1 & 1 \\
1 & Pij & X & X \\
1 & 1 & 1 & 1 \\
X & Fij & 1 & X \\
X & 1 & 1 & 1
\end{array}
\]

Fig. 8 – Image segmentation.

Fig. 9 – a) After low pass filter; b) after target “cropping”.

The goal of these processing steps is to obtain a “closed” target object. Besides the target position (centre of gravity) and size, there are also computed the following moments, used for the target classification. It is well known from the moments theory that an image function \( f(x, y) \) can be uniquely described by means of an infinite set of moments: \( \{m_{pq}; p, q = 0, 1, 2, \ldots\} \). In the discrete case, the moments are calculated by: 
\[
m_{pq} = \sum_{x} \sum_{y} x^p y^q f(x, y),
\]
which for a binary image can be separated: 
\[
m_{pq} = \sum_{x} x^p \sum_{y} y^q.
\]
Based on regular moments can be constructed the centered moment of order \( (p+q) \):
\[
\mu_{p,q} = \sum_{x} \sum_{y} (x - \bar{x})^p (y - \bar{y})^q f(x, y).
\]
The normalized centered moment of order \( (p+q) \) is defined as:
\[
\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{[(p+q)/2+1]}}.
\]
We’ll compute the following derived moment \( \Phi_1 = \eta_{1,0} + \eta_{0,2} \) \[3\], which will be used for the classification of the target. It results:

\[
\Phi_1 = \sum_x (x-x_c)^2/n_p^2 + \sum_y (y-y_c)^2/n_p^2 ,
\]

where \( n_p \) is the number of the target pixels. At the end of this processing stage we’ll have the following information relative to the target position and geometric parameters:

- Target dimension \( N_p \) (in pixels);
- Enclosing rectangle (in pixels on X and Y axis) \( N_x \) and \( N_y \);
- Centre of gravity \( C_x \) and \( C_y \) of the detected target (in pixels relative to the “gate” position);
- \( \Phi_1 \) moment.

In the test case, these computed values are as follows: \( N_p = 16\,943 \), \( N_x = 260 \), \( N_y = 101 \), \( C_x = 392.86 \), \( C_y = 192.88 \), \( \Phi_1 = 0.2682 \).

These parameters are transmitted to the MCU in order to allow the classification of the target, to obtain precisely the target position and to support the elaboration of the commands needed for moving the tracking platform in the future position of the tracked target trajectory. In the same time, VP receives from MCU the new position of the “gate” tracking window.

3. CLASSIFICATION OF THE TARGET

This stage of the processing takes place in the MCU and aims to classify segmentation containing a particular target, preferably in two stages. In the first stage, features such as length, width, height, height variation and radius are used to eliminate the segmentation that does not contain targets of interest. Segmentation, which exceeds this step, is then compared to real data stored in the target database. Data of targets located in the database may include: length, width, height, average height turret, etc. The classification is done by using the following algorithm: those segmentation that are beyond the first step of classification are compared with the actual target of interest (prototype vectors), using the formula:

\[
\text{Fit measurement}_j = \frac{\sum_{i=1}^n \left[ 1.0 - \frac{|f_{i,j} - f_{i,k}|}{f_{i,k}} \right]}{n},
\]

where \( f_i \) is feature vector for the segmentation, \( f_j \) is feature vector for the target template, \( j \) and \( n \) are the numbers of the features. In our case was considered a features vector composed only from one feature: the moment \( \Phi_1 \). First we make the
comparison between enclosing rectangles, and after that, if the differences are below 10%, we’ll compute the “fit measurement” value. In order to make possible the comparison between the enclosing rectangles for the template target and the detected one, we use the distance from the laser telemeter and the viewing angle of the FLIR camera – scale factor \( S = \frac{D \times \tan(\alpha/2)}{\text{Dim/2}} \), where: \( D \) is the laser telemeter distance (meters), \( \alpha \) is the viewing angle of the FLIR (degrees) and Dim is horizontal dimension of image (pixels).

In the test case (Fig. 10), \( D = 400 \) m, \( \alpha = 3^\circ \) and Dim = 720. The resulted \( S \) factor is 0.029 m/pixel. We obtain for the detected segmentation \( L = 260 \) pixels and \( l = 94 \) pixels, that means \( L = 7.54 \) m and \( l = 2.73 \) m. For the “truck” template we have

\[
L = 7.5 \text{ m and } l = 2.8 \text{ m, and for “tank” template we have } L = 7.0 \text{ m and } l = 2.6 \text{ m.}
\]

In both cases the test regarding the enclosing rectangle will be positive so, we’ll compute the “fit measurement” value. For the “truck” template we compute a \( \Phi_1 = 0.2033 \) and, for the “tank” template this value is \( \Phi_1 = 0.2620 \). For the detected target, \( \Phi_1 = 0.2682 \). For the “truck” template, the “fit measurement” value is 0.758, meanwhile for the “tank” template the same parameter is 0.9769. The MCU will end this process by classifying the target like a “tank” type target.

4. DATA FUSION AND TARGET COORDINATES ACQUIRING

This final processing stage is to combine the information obtained through sensors presented before, in order to determine the analyzed target absolute coordinates. As inputs in the process of merging we have:

1. Geographical position and elevation of the multisensor platform, obtained from a GPS receptor;
2. Values of the orientation angles: compass, roll and pitch obtained from a gyrocompass sensor type;
3. The value of elevation angle of the optical equipment (video camera, FLIR);
4. The LOS (line of sight) for the detected target.
The calculation results presented here allow to obtain the absolute geographic coordinates of the target. We consider the angles of elevation of the target: AZi and ELi in the azimuth, respectively in the elevation plan, obtained by summing angles obtained from the transducers of angular platform to the angles determined by the optical system (AZo, ELo) and by FLIR system (AZI, ELI), respectively. These angles are relative to the tracking platform. Tracking system can be represented as a system of three-dimensional axes $O'XYZ'$ angular moved to the Cartesian axes $OXYZ$ focused on East, West and the axis of gravity land as the angle between the projection axis $O'Z'$ on $XOY$ plane and the axis $OZ'$ measured effect anticlockwise A1 (measured = "pitch"), the angle between the projection axis $OY'XOZ$ and its focus $OY'$ measured clockwise A2 (measured = "roll"), the angle between the projection axis $O'X'YOZ$ and its focus $O'X'$ measured clockwise is A3 (measured = "compass").

Fig. 11 – The platform angle definition.

By means of the following computing formulas it was obtained the target coordinates $(X_o, Y_o, Z_o)$ in the Cartesian reference own tracking platform, where $R_{md}$ distance is determined by means of the laser telemeter.

\[
X_o = R_{md} \cos(EL_i) \cos(AZ_i),
Y_o = R_{md} \cos(EL_i) \sin(AZ_i),
Z_o = R_{md} \sin(EL_i). \tag{5}
\]

Applying successively corrections of compass, roll and pitch angle, it get the corrected coordinates $(X, Y, Z)$:

\[
X_c = X_o \cos(A3) - Y_o \sin(A3),
Y_c = Y_o \cos(A3) + X_o \sin(A3),
Z_c = Z_o, \tag{6}
\]

\[
Y_r = Y_c \cos(A2) - Z_c \sin(A2),
Z_r = Z_c \cos(A2) + Y_c \sin(A2),
X_r = X_c,
\]

\[
X = X_r \cos(A1) - Z_r \sin(A1),
Z = Z_r \cos(A1) + X_r \sin(A1), Y = Y_r.
\]

Next reading from the GPS receiver the coordinates of the multisensor platform: Long0, longitude reference and Lat0, respectively, the latitude of
Image processing for acquiring target coordinates

reference, then it was converted those coordinates in UTM coordinate system \((X_{utm}, Y_{utm})\). The geographical coordinates (longitude, latitude, altitude) of the target results from the application of the following relations \((X,Y,Z)\):

\[
X = X_{utm} + X', \quad Y = Y_{utm} + Y'.
\]

The \(X\) and \(Y\) values thus corrected according to the platform position in multisensor UTM reference system will be converted to the latitude and longitude (WGS84 ellipsoid), using conversion formulas [2].

5. CONCLUSIONS

The procedures and equations described above allow for the definition and implementation of a computing algorithm that solve complex problems of the detection and classification of the target in adverse conditions, masked partially by natural (vegetation) or by specific countermeasures (netting or masking).

The results obtained allow the computing power necessary information in the implementation of integrated computing platform. It was found that such a modern industrial computer, based on a Pentium Dual Core class processor and a DSP co-processor has enough computing power to implement both the stabilization of the platform and the recognition and tracking of the target.

Received on 10 June 2009

REFERENCES