

An Efficient Object Orientation Detection for Machine Vision



Engineering

KEYWORDS : Binary image, Geometric moments, Morphological operation, OPEN and CLOSE

P. Vijaya kumar

Asst. Pro, Department of ECE, SRM University, Kattankulathur, Tamil Nadu - 603203, India

K. Sathesh kumar

Department of ECE, SRM University, Kattankulathur, Tamil Nadu - 603203, India

ABSTRACT

Object detection and object orientation detection is one of the key important process in machine vision applications. In this paper an efficient algorithm was proposed for calculating the orientation of objects in 2-D space by computing the geometric moments. The real time object detection and orientation detection require less computational complexity methods. The geometric based method provides relatively good performance with a reasonable computational complexity. To increase the efficiency of the algorithm further, Morphological operations such as OPEN and CLOSE are used which eliminates gaps, hole and peninsulas. This moment-based approach is very useful to find the position and orientation of objects in the image plane, which has a wide range of applications from networked embedded robotics to the medical care field.

INTRODUCTION

In the area of image processing a very large amount of research work has been performed on the topic of using geometrical moments for the detection and classification of objects on the images taken by cameras. In this work binary images were taken. Binary images are images that have been quantized to two values, '0' and '1', but in practice the corresponding 8-bit pixel values are 0 and 255. Binary images are usually obtained by thresholding a gray level image. Pixels with a gray level above the threshold are set to '1' and the rest are set to '0'. Choosing the threshold value can be difficult and it has even considered of being a "black art"[7]. Here an efficient algorithm was proposed that finds orientation of the object in the field of camera view by processing binary images and computing the geometric moments of the object. Efficiency of the algorithm[1] was increased by doing morphological operations such as OPEN and CLOSE[3]. The objects may contain holes leads to inaccurate values. In this paper the proposed algorithm will fill the holes prior to find the angle of orientation and increase the accuracy.

The algorithm such as one presented in this paper is suitable to implement for on-chip processing on the cameras such as CMUcam3[5]. It is envisioned that this algorithm will be suitable for a developing research area, networked embedded robotics [4]. This moment-based approach is also utilize to find the posture of a human silhouette for those medical applications where there is a need to provide care to patients who are in a home or office type environment. In these applications one usually wants to know, in near real-time, not only the value of the patients' vital signs, e.g. ECG, blood pressure, etc., but also the location and posture of their bodies. The on-chip processing of binary images also addresses the concern raised in the medical care area about privacy and security of the information being gathered and transmitted. By processing only binary images on the camera chip and then transmitting numbers related to the geometrical moments of those images the security and privacy concerns can be taken care using simple procedures. No high resolution or color images are processed, stored, or transmitted. The relative small number of numerical values transmitted will not require large amounts of overhead if their encryption is desirable.

GEOMETRICAL MOMENTS

Geometric moments have proven to be a very efficient tool for image analysis[2], are used for the detection and classification of objects in images taken by cameras. The general two-dimensional th order moments of a grey-level image are defined as

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy \quad \dots\dots\dots(1)$$

Where p,q= 0,1,2,....

In the case of a binary image, the double integral must be replaced by a summation. Then, we have,

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad \dots\dots\dots (2)$$

Assuming that the binary image f(x, y) has only one object, hereafter noted as blob, the 2-D geometrical moment of order (p+q) of the blob f(x, y) can be calculated by using equation (2). The standard moments from zeroth- to second-order are calculated to determine the center of gravity and other geometrical information for each blob. The first order moments are used to find the centroid (of the given object, by using (3).

$$x_c = \frac{m_{10}}{m_{00}} , y_c = \frac{m_{01}}{m_{00}} , \dots\dots\dots (3)$$

The standard moments are not enough to determine the orientation of the blobs. A common method to determine the orientation of the object is to estimate the orientation of its ellipse mask. The calculation of the central moments is needed to obtain θ, the angle between the major axis of the ellipse and the positive x-axis.

The central moments then can be defined as

$$\mu_{pq} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q \quad \dots\dots\dots(4)$$

The second order central moments allow us to infer the equivalent ellipse mask, and are computed by the following equations.

$$\begin{aligned} \mu_{11} &= m_{11} - \frac{m_{10}m_{01}}{m_{00}} \\ \mu_{02} &= m_{02} - \frac{(m_{01})^2}{m_{00}} \\ \mu_{20} &= m_{20} - \frac{(m_{10})^2}{m_{00}} \end{aligned}$$

The orientation of the object can be measured by calculating angle between major axis of the ellipse mask and positive x- axis. can be calculated by using following equation.

$$\theta = \frac{1}{2} \text{atan} \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad \dots\dots\dots (5)$$

The value of θ lies between ± 90° as illustrated in Figure 1.

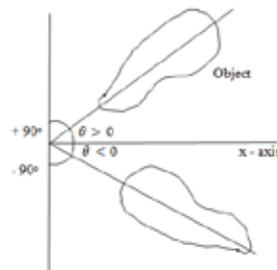


Figure 1: Object orientation with x-axis

MORPHOLOGICAL OPERATIONS

The field of mathematical morphology contributes a wide range of operators to image processing, all based around a few simple mathematical concepts from set theory. Morphology is based on geometry and shape, morphological operations simplify images, and preserve the main shape characteristics of objects. The operators are particularly useful for the analysis of binary images and common usages include edge detection, noise removal, image enhancement and image segmentation. Binary morphological operators implemented with replacement of addition and multiplication by logical OR and AND. These operations take two pieces of data as input: an image to be modify and a structuring element (also known as a kernel).

Structuring element:

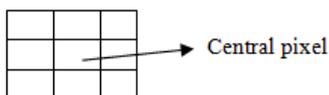


Figure 2 : Structuring element of 3x3 square

A structuring element is a geometric relationship between pixels of different shapes. A structuring element of shape 3x3 square matrix is shown above.

Morphological operations can be done by moving the structuring element over the image I to be modified, in such a way that it is centred over every image pixel at some point. This process is done row-by-row and column-by-column fashion. Structuring element is also often called as moving window B. A window is a geometric relationship that creates a series of miniature images as it is passed over the image row-by-row, column-by-column.

A window B is a way of collecting local image intensities, and a set of coordinate shifts centred around (0, 0). For 3x3 square window the coordinates are

- {(-1,-1), (-1,0), (-1,1),
- {(0,-1), (0,0), (0,1),
- {(1,-1), (1,0), (1,1) }

Windowed set:

Given an image I and D window B Define the windowed set at image coordinate (i, j) as

$$B.I(i, j) = \{I(i-m, j-n) ; (m, n) \in B\}$$

Which is the set of image pixels covered by the window when it is centred at coordinate (i, j).

If B = square(3x3);
Then

$$B.I \quad (i, j) = \{I(i-1, j-1), I(i-1, j), I(i-1, j+1), \\ I(i, j-1), I(i, j), I(i, j+1), \\ I(i+1, j-1), I(i+1, j), I(i+1, j+1)\}$$

When the window is centred at first row and first column pixel, the values in the first row and first column of the window are filled by nearest pixel values. Similar process is applied for all edge pixel of an image I.

Dilation and Erosion:

The two most basic operations in mathematical morphology are erosion and dilation.

Dilation enlarges the boundaries of regions of the object in binary image. Areas of the objects grow in size. Dilation can be performed by taking the logical OR operation over the windowed set B.I. i.e. if any of the pixel in the windowed set is "1", then the central pixel of windowed set is set to "1".

$$Dilation = OR\{ B.I(i, j) \}$$

Dilation can fill the 'gaps' and eliminates or reduce the 'holes' of the object in binary image.

Erosion compresses the boundaries of regions of the object in a binary image. Areas of the objects reduce in size. Erosion can be performed by taking the logical AND operation over the windowed set B.I. i.e. if any of the pixel in windowed set is "0", then the central pixel of windowed set is set to "0".

$$Erosion = AND\{ B.I(i, j) \}$$

Erosion can eliminate 'small regions' and reduce the 'peninsulas'.

OPEN and CLOSE:

Dilation and Erosion can change the size of the object. It leads to change in computed moment values. In order to remain the size of the object we need to perform OPEN and CLOSE operation. OPENing and CLOSEing are nothing but the combination of 'dilation' and 'erosion'. OPENing can be defined as 'erosion' followed by 'dilation', and CLOSEing as 'dilation' followed by 'erosion'.

$$OPEN(I, B) = Dilate[Erode(I, B), B]$$

$$CLOSE(I, B) = Erode[Dilate(I, B), B]$$

OPEN removes too small objects and gaps. CLOSE removes too small holes but not objects and peninsulas. Hence we perform both operations for the algorithm to be effective.

ALGORITHM DESCRIPTION

The algorithm works on a row by row mode from left to right and top to bottom. Camera captures the background image and live image. Difference image can be calculated from background and live images, which leaves only the object in the image plane. Median filter is applied to remove the noise from the difference image, and converted into binary by thresholding the gray level image, then applied the algorithm which finds the orientation of the object.

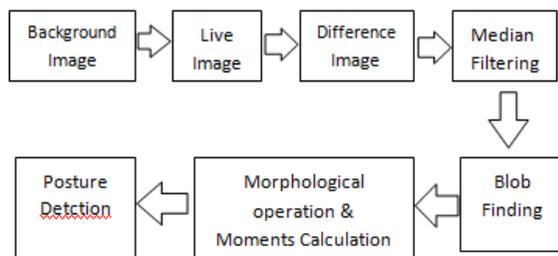


Figure 3: System Block diagram

- Algorithm process the binary image row-by-row from top to bottom.
- Each blob is identified by labelling.
- There is a data structure with information about each blob.
- When a new blob is identified, it is labelled with new number and new data structure is initialised and blob data will update in new data structure.
- After processing all the rows of the image there is a data structure with each blob.
- OPENing and CLOSEing operations are applied to each blob, which eliminates the small objects, peninsulas and fill the holes and gaps.
- After OPEN and CLOSE operations each blob is processed again for finding the hole. If the holes still present they are filled to make the blob perfect.
- Compute the geometrical moments of zeroth order to second order. i.e. , , i.e. $m_{-00}, m_{-01}, m_{-10}, m_{-11}, m_{-02}, m_{-20}$.
- Compute the second order central moments. i.e. μ_{-11}, μ_{-02} and μ_{-20} .
- Finally compute the centroid (x_c, y_c) and angle of orientation of the each blob.

RESULTS AND DISCUSSION

The proposed efficient algorithm was tested by performing simulation in MATLAB on various binary images, containing gaps,

holes and peninsulas. Morphological operations will eliminate gaps, holes and peninsulas. Here centroids and angles of orientation of objects in the images are calculated with and without morphological operations. For morphological operations we taken the structuring element of shape 11x11 square matrix.

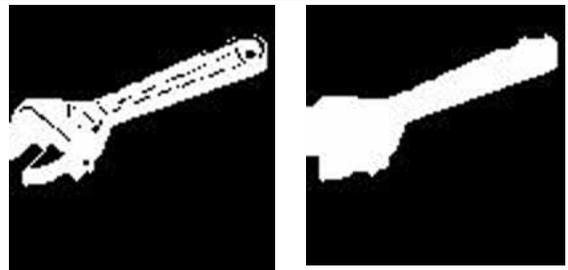
In the following figures figure (a) shows the binary image before OPEN and CLOSE process and figure (b) shows the image after OPEN and CLOSE process.



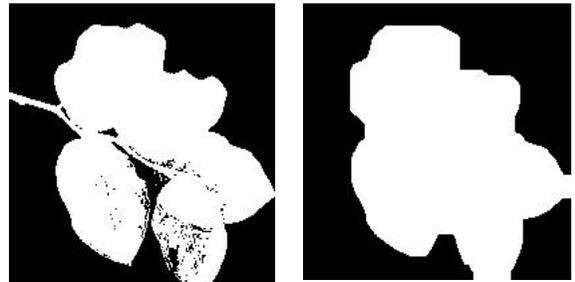
(a) (b)
Figure 4: Random shape 1



(a) (b)
Figure 5: Random shape 2



(a) (b)
Figure 6: Spanner image



(a) (b)
Figure 7: Leaf image

Performance of the proposed algorithm is tabulated below

Table 1: Centroid and angle calculation results

Image	Centroid		Orientation Angle
	(a)	(b)	
Random shape 1	(a)	80.7	76.1 - 79.3
	(b)	62.9	76.5 - 49.8
Random shape 2	(a)	128.1	82.9 49.2
	(b)	122.2	82.9 43.8
Spanner image	(a)	79.1	83.6 22.7
	(b)	79.8	82.1 23.2
Leaf image	(a)	105.9	105.7 - 54.4
	(b)	108.2	106.9 - 57.0

REFERENCE

[1]. Yanfei Liu and Carlos Pomalaza-Raez - "A Low-Complexity Algorithm for the On-Chip Moment Computation of Binary Images" 2009, IEEE.
 [2]. M. K. Hu, "Visual Pattern Recognition by Moment Invariants", IRE Trans. Inform. Theory, vol. IT-8, no. 2, pp. 179-187, 1962. [3]. Dr. Shah- "Binary Image Processing", YouTube [4]. M. Bordignon, E. Pagello, and A. Saffiotti- "An inexpensive, Off- The Shelf Platform for Networked Embedded Robotics," in Proc. Of the 1st International. Conference. on Robot Communication and Coordination (Robo Comm), Article No. 45, Athens, Greece, October 15-17,2007 [5]. A. Rowe, A. G. Goode, D. Goel, and I. Nourbakhsh, CMUcam3-"An Open Programmable Embedded vision Sensor" Robotics Institute, Carnegie Mellon university, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-0713, May 2007. [6]. S. Dey, B. Bhattacharya, M. Kundu and T. Acharya- "A Simple Architecture for Computing Moments and Orientation of an Image", Fundamental Informatic, 52(4), pp. 285-295, April 2002. [7]. O. Faugeras- "Three-Dimensional Computer Vision: A Geometric Viewpoint", MIT Press, 1993.