Image Moments-Based Structuring and Tracking of Objects

LOURENA ROCHA, LUIZ VELHO, PAULO CEZAR P. CARVALHO

IMPA–Instituto Nacional de Matemática Pura e Aplicada
Estrada Dona Castorina, 110, 22460
Rio de Janeiro, RJ, Brasil
{lourena, lvelho, pcezar}@visgraf.impa.br

Abstract. This paper presents a new method for structuring and tracking of objects in video sequences. Our approach is based on image moments and the *bsp*-tree data structure. We use invariant properties of these moments to construct a *bsp*-tree and determine an ellipsis that approximates the object's shape. Then, we employ this information to track objects frame by frame through the image sequence. The method works well for segmented images with a single object and we assume that the motion will not change abruptly.

Keywords: BSP-tree, ellipsoidal structure, moment invariants, hierarchical coherence.

1 Introduction

The problem of localizing and tracking moving objects has been an active research topic for the past years. During this time, several approaches for tracking have been developed.

Local approaches, such as optical flow, [6, 1], provide a low-level problem characterization and suffer from some drawbacks [5].

Feature-based approaches are more robust in this respect. Shi and Tomasi [2] for example, propose an optimal feature selection criterion along with feature tracking based on dissimilarity. Their method, however, fails for non rigid surfaces. Other approaches for tracking parameterized patches [3], also have the same limitation.

Techniques based on active contours (i.e. *snakes*) [7], as the one proposed by Isidoro and Sclaroff [4], can track deformable objects, but they need manual initialization.

Approaches based on shape models, using quadrics and superquadrics [9, 8] for example, are very expensive because of the calculation of model parameters.

In this paper we propose a new method that can track several categories of objects, from rigid to deformable or articulated. Our method works for image sequences involving an stationary background and one target object moving through a fixed camera's field of view. These assumptions allow us to segment each image into a binary region representing the moving object by using background subtraction. Moreover, we also assume that the motion is smooth.

Our approach is based on *image moments*. Moment invariants allow us to infer the *equivalent ellipse*, i.e., the best fitting ellipse for a target 2D object. Since an image is the projection of a 3D scene, we can extend this property to get the best fitting ellipsoid for the target 3D object. In [11], ellipsoids are used to approximate the surface of a 3D object from volumetric data. We work only with 2D information and use image moments to perform structuring and tracking of dynamic objects. In this context, moments deal well with noise and uncertainty.

The algorithm is divided in two main stages. In the first stage, we use moments to construct a *bsp*-tree[12] for each frame of the sequence. This hierarchical representation approximates the object by 2^k ellipses at each level k of the tree. In second stage, we use the hierarchical structure of *bsp*-tree to track the object from frame to frame.

This paper is organized in five sections including this introduction. Section 2 defines moments and discusses their properties. Section 3 explains how the method approximates an object using ellipses and how the hierarchical structure is used for tracking. Results are shown in Section 4. Finally, Section 5 concludes the paper.

2 Moments

Image moments and moment invariants play a very important role in object recognition and shape analysis.

The general two-dimensional $(p+q)$th order moments of a grey-level image $f(x, y)$ are defined as:

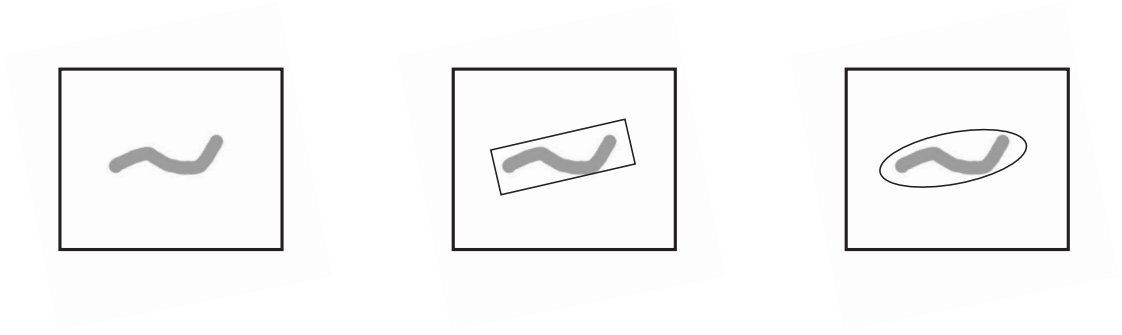
$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \qquad (1)$$

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

Since we deal only with binary objects in this paper, then f is the characteristic function of object G [10], and

$$m_{pq} = \int \int_G x^p y^q dx dy \qquad (2)$$

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



(a) (b) (c)

Figure 1: Approximation based on image moments: (a) source image, (b) enclosing rectangle, (c) enclosing ellipse

In the case of a digital image, the double integral in (1) and (2) must be replaced by a summation. The most common way to do that is to employ the rectangular (i.e., zero-order) method of numeric integration. Then, for binary images we have:

$$m_{pq} = \sum_{A} x^p y^q \tag{3}$$

where the summation extends over all the elements in A, i.e., all the "black" pixels in the image array.

With this framework we can design shape features or measurements that are invariant to certain affine transformations [1].

The *central moments* are defined as:

$$\mu_{pq} = \sum_{A} (x - x_c)^p (y - y_c)^q \tag{4}$$

where x_c and y_c are the coordinates of the center of gravity c, or centroid, of the given object. These moments are invariants to translation.

From the Eq. (3) we see that m_{00} is the area of the pattern, i.e., the number of black pixels. The centroid c can be calculated combining m_{00} with the image moments of the first degree m_{01} and m_{10}. Using the moments of the second degree m_{11}, m_{02} and m_{20}, together, the orientation of the object in the image can be calculated.

The binary image of the *equivalent rectangle* has the same zeroth, first and second moments. The coordinates of the centroid $c = (x_c, y_c)$, the angle θ between the longer edge and the x-axis, and the length of edges, (w, l), are calculated as follows:

[1] We remark that, although we are discussing binary objects, it is possible to generalize all the following relations and results for grey-level objects.

$$\begin{aligned} x_c &= \frac{m_{10}}{m_{00}}, \\ y_c &= \frac{m_{01}}{m_{00}}, \\ \theta &= \frac{tan^{-1}(\frac{b}{a-c})}{2}, \\ w &= \sqrt{6(a + c - \sqrt{b^2 + (a-c)^2})}, \\ l &= \sqrt{6(a + c + \sqrt{b^2 + (a-c)^2})}, \end{aligned} \tag{5}$$

where a, b and c are defined as:

$$\begin{aligned} a &= \frac{m_{20}}{m_{00}} - x_c^2, \\ b &= 2(\frac{m_{11}}{m_{00}} - x_c y_c), \\ c &= \frac{m_{02}}{m_{00}} - y_c^2. \end{aligned}$$

Using these parameters we can infer the *equivalent ellipse*, where c will be its center and w and l, the major and minor axes, respectively. Figure 1 illustrates the above concepts. In the paper, we employ enclosing elements with dimensions $(2w, 2l)$.

3 The Method

The solution that we propose for the tracking problem is divided in two main parts. In the first part we are concerned in constructing a *bsp*-tree representing the object in each frame of the sequence. This structure gives a hierarchical approximation of the object by a set of equivalent ellipses. In the second part, this hierarchical approximation is matched frame by frame to track the target object. The first part is explained in Subsection 3.1, below, and the description of the tracking algorithm is given in Subsection 3.2.

3.1 Construction of the BSP-Tree

For each frame of the image sequence a *bsp*-tree is constructed. In the tree each node represents an equivalent ellipse that approximates a part of the target object. Level k of the tree (0 for the root) has 2^k nodes, corresponding to a set of ellipses that represents the object at that level. The *bsp*-tree is assumed to have the same number of levels for all frames of the image sequence.

Each node contains the following parameters:

- *c* = (x_c, y_c) - the object centroid and equivalent ellipse's center;
- *d* - the direction of the ellipse's minor axis ;
- m_{00} - moment of the zeroth degree ;
- m_{01}, m_{10} - moments of first degree;
- m_{11}, m_{02}, m_{20} - moments of second degree,

The above parameters are computed according to Eq. (5), with exception of direction d, whose definition is:

$$d = (\cos(\theta + \frac{\pi}{2}), \sin(\theta + \frac{\pi}{2})),$$

with θ as in (5).

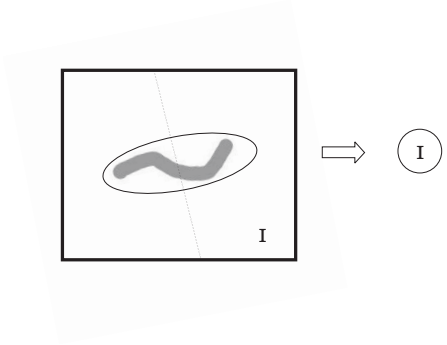
Given a frame i of the sequence and the number k of levels of the tree, the procedure that constructs the approximating *bsp*-tree consists of the following steps:

1 - Compute m_{00}, m_{01}, m_{10}, m_{11}, m_{02}, m_{20}, c, d and get the ellipse of center c and minor axis with direction d. Place this information in the root node of the tree. (Figure 2(a))

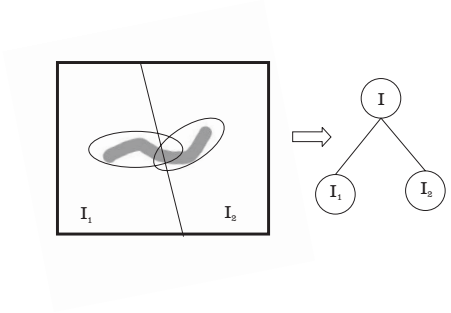
2 - Subdivide the image in direction d and add two children nodes to the root, each one containing the information corresponding to one sub-image.(Figure 2(b))

3 - For each children node it recursively applies the algorithm (steps 1 and 2) until reaching level k.

After applying this procedure for each image we will get a new sequence of frames where the object structure is represented by a set of ellipses. (Figure 3)



(a)



(b)

Figure 2: Construction of the *bsp*-tree: (a) level 0, (b) level 1

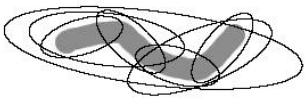


Figure 3: Object fitting by 2^k ellipses at each level. Structure with 3 levels (0, 1 and 2).

3.2 Tracking

Define bsp_j and $bsp_{(j+1)}$ as the trees of frames j and $(j+1)$ respectively. For each pair of consecutive frames apply the algorithm below:

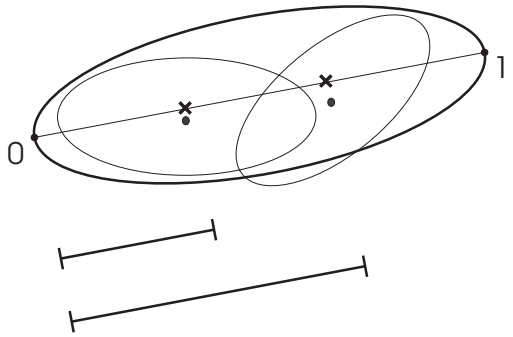
1 - Parameterize the major axis of the ellipse in the root node of bsp_j and $bsp_{(j+1)}$ between 0 and 1.

2 - In each *bsp*-tree orthogonally project the center of the ellipses in children nodes on the parameterized axis of the ellipse in the root node. (Figure 4)

3 - For each tree, calculate the distance of the two projections to origin of the parameterized axis. Then, make the correspondence according to the distance of the projection to the origin.

4 - Compute the geometric transformation between the ellipse in bsp_j and its correspondent in $bsp_{(j+1)}$.

5 -Repeat the procedures above for each children node in bsp_j and $bsp_{(j+1)}$.



(a)

Figure 4: Projection of the centers of the ellipses contained in children nodes on the parameterized axis of the ellipse contained in the root node

Then, given the sequence of frames, along with the number of frames and levels of *bsp*-tree, specified by n and k, respectively, we finally get the tracking algorithm:

```
Tracking(frames, n, k)
{
  Compute_all_bsp(frames, n, k);
  Compute_corresp_transf(frames, k);
}
```

4 Results

In order to demonstrate our algorithm we give three examples of shapes with different topologies. The first example, shown in Figure 5, is a simple "S" shape. The second example, shown in Figure 6, is a more complicated "X" shape with branches. The third example, shown in Figure 7, is an "O" shape which has a hole. Note that the method captures the object shape very well in all cases.

The tracking procedure for matching the hierarchical structure frame-by-frame still is currently being implemented. Figure 8 shows preliminary results that indicate the accomplishment of our goals.

We intend in the future to apply the algorithm to track real objects in video sequences. More specifically, we plan to take advantage of grey-scale image sequences where the camera remains stationary and the first frame has just the background (Figures 9 (a) and (b)). In this case, we can segment each frame into a binary region representing the moving object by background subtraction (Figure 9 (c)) and applying a threshold. This pre-process provides an sequence suitable for input into our algorithm (Figure 9 (d)).

5 Conclusion

In this paper we have proposed a method for structuring and tracking objects in an image sequence. The algorithm uses image moments to build a hierarchical structure which approximates the object by a set of ellipses at different levels. Tracking is performed by exploiting structural and temporal coherence of the representation.

Some problems that we did not address in this paper are occlusion, tracking of multiple objects and motion discontinuities. Future work will go in these directions.

Acknowledgements

The first author would like to thank Vinícius Moreira Mello for discussions, contributing to the ideas in this paper and for help in the implementation.

This research has been developed in the VISGRAF Laboratory at IMPA. VISGRAF Laboratory is sponsored by CNPq, FAPERJ, FINEP and IBM Brasil.

References

[1] R. Cutler and M. Turk, *View-based Interpretation of Real-Time Optical Flow for Gesture Recognition*, Proc. Third IEEE Conference on Face and Gesture Recognition, Nara, Japan, April 1998.

[2] J. Shi and C. Tomasi, *Good Features to Track*, IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, 1994.

[3] G. D. Hager and P.N. Belhumer, *Real-Time Tracking of Image Regions With Changes in Geometry and Illumination*, Proceedings IEEE Conference on Computer Vision and Pattern Recognition, pp. 403-410, 1996.

[4] S. Sclaroff and J. Isidoro,*Active Blobs*, Proc. of ICCV, 1998.

[5] F. Dellaert and R. Collins, *Fast Image-Based Tracking by Selective Pixel Integration*, ICCV99 Workshop on Frame- Rate Applications, September 1999.

[6] J.L. Barron and D.J. Fleet and S.S. Beauchemin and T.A. Burkit, *Performance of Optical Flow Techniques*, CVPR, vol. 92, pp. 236-242.

[7] M. Kass and A. Witkin and D. Terzopoulos, *Snakes: Active Contour Models*, International Journal of Computer Vision, 1(4), pp. 321-331, 1987. Marr Prize Special Issue.

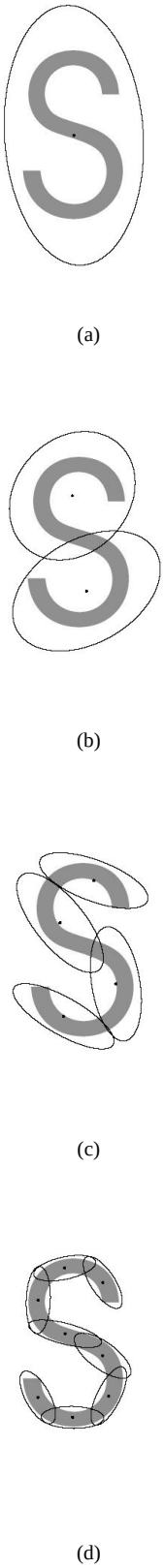
[8] D. Metaxas, *Shape Representation and Nonrigid Motion Tracking Using Deformable Superquadrics*, Geometric Methods in Computer Vision, B.C. Vemuri(ed.), Proc. SPIE 1570, San Diego, CA, July 1991, 12-20.

[9] Pentland, Alex, *Modal Descriptions for Recognition and Tracking*, Proceedings of IAPR Workshop on Machine Vision Applications, pp. 435-444, December 1992.

[10] Horn, Paul, *Robot Vision*, MIT electrical engineering and computer science series, 1987.

[11] F. Banégas, M. Jaeger, D. Michelucci, M. Roelens, *The Ellipsoidal Skeleton in Medical Applications*, Sixth Solid Modeling 2001, Ann Arbor, Michigan, June 4-8, 2001.

[12] Henry Fuchs, Zvi M. Kedem, Bruce F. Naylor, *On Visible Surface Generation by a Priori Tree Structures*, Computer Graphics" 14(3), p. 124-133, 1980.



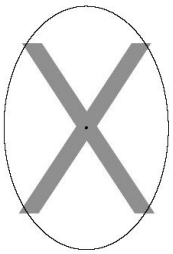
(a)

(b)

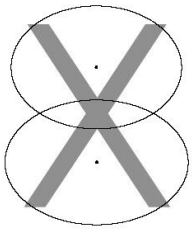
(c)

(d)

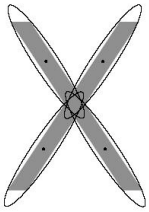
Figure 5:



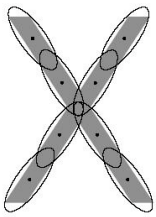
(a)



(b)

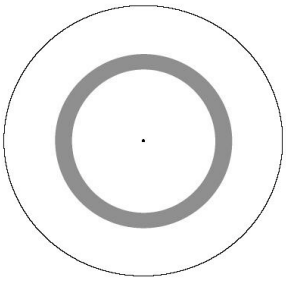


(c)

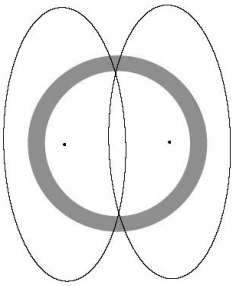


(d)

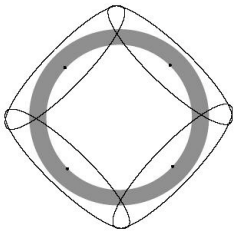
Figure 6:



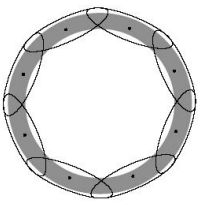
(a)



(b)

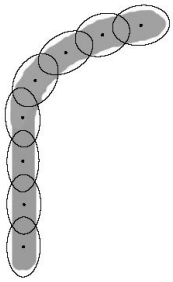


(c)

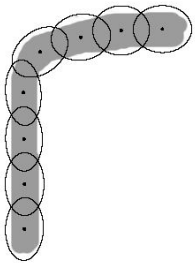


(d)

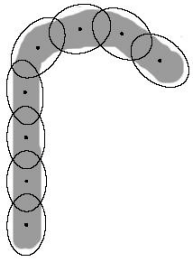
Figure 7:



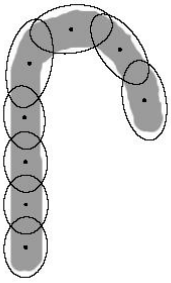
(a)



(b)



(c)



(d)

Figure 8:

(a)

(b)

(c)

(d)

Figure 9: