Maximum Line Segments for Object's Motion Evaluation

A. Fernandez Caballero[1], Modesto Montoya, Miguel A. Fernandez[1] and Juan Moreno[2]

Departamento de Informatica, Escuela Politecnica Superior de Albacete
Universidad de Castilla-La Mancha
Albacete, 02071, Spain
e_mail: 1{caballer, miki}@info-ab.uclm.es
2 jmoreno@sancho.info-ab.uclm.es

Abstract. Motion analysis in sequences of images is a discipline in constant growth due to the great number of applications in which it plays a primordial key function. The presentation of a simple way to obtain, in real time and in a neural form, a fundamental parameter associated to the objects in movement (its size) is the objective of this paper.

1 Introduction

Motion analysis in images is growing in importance in numerous applications. Some of these applications are: (a) television coding by means of compensation, (b) mobile robotics, (c) satellite images, (d) applications, civil as well as military, related to objective pursuit and autonomous guidance, (e) biological and medical images, (f) surveillance and supervision, and, (g) virtual reality [6]. The problem of motion detection is particularly interesting when the objective is to spatially locate the mobile objects in the scene. This motion detection is always strongly bound to the detection of temporary changes in the image. When moving objects exist in a scene, we will always have to handle with changes in the intensity of the images' pixels. This fact has given place to an extensive bibliography, in which we already highlight some classic works with different approaches. For a more extensive study of the topic, we recommend the papers of M.A. Fernandez [2] and A. Fernandez Caballero et al. [5]. To highlight the emergent approach of Fernandez look at [1] [3] [4] in this sense.

The pursuit of elements from one image to another is a common procedure, mainly in applications of surveillance. Some pursuit processes have been defined by means of: (a) a representative model of elements (for example, the image co-ordinates of some characteristic points, the longitude and the orientation of the segments of the contour), (b) a kinetic model of the evolution of the elements (for example, a constant speed, a constant acceleration), (c) a group of relationships between the parameters of the pattern and the measures of the image, and, (d) a temporary filter for the estimate of the parameters of the pattern starting from the image data.

Most research is nowadays dealing with complicate algorithms. Nevertheless, in many applications, it is not necessary to deal with very elaborated data, but real time is the fundamental cue. Often, this is only possible using neural networks associated to very simple parameter extraction.

2 Maximum line segments

One of the principal aspects of the structure of an object is its size. To know the size of an object in absolute terms is useful for the recognition of the object [7] [8]. An object in translation, dilation or rotation can be simplified in terms of the calculation of its size in all instant of time t.

Our proposal is that this variables can be obtained in a simple, well-known way, provided the object's silhouette S is known at every instant t. Therefore we will define the size starting from the longitude of two right lines (or cords) determined by four well-known points of the surface of the object. The points to which we are making reference are (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4), such that:

$$\forall (x,y) \in S(i,j,t), \quad x_1 < x$$
$$\forall (x,y) \in S(i,j,t), \quad x_2 > x$$
$$\forall (x,y) \in S(i,j,t), \quad y_3 < y$$
$$\forall (x,y) \in S(i,j,t), \quad y_4 > y$$

In other words, the four points are:

(x_1, y_1) : point most at the left of the object in the image
(x_2, y_2) : point most at the right of the object in the image
(x_3, y_3) : upper most pixel of the object in the image
(x_4, y_4) : lower most pixel of the object in the image

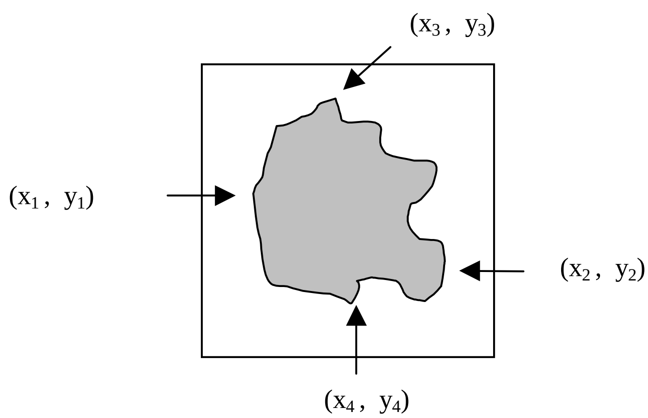

Figure 1. Obtaining of the extreme points of an object

The two cords that we denominate maximum line segments of the object, won't unite the pixels (x_1, y_1) and (x_2, y_2), (x_3, y_3) and (x_4, y_4) to each other, but rather their projections $(X_1, 0)$ and $(X_2, 0)$, $(0, Y_3)$ and $(0, Y_4)$, respectively, as you can appreciate in figure 2.

Now, the object's location will be determined by a unique characteristic pixel (X_{obj} , Y_{obj}), that is to say, the intersection of the two segments (X_1 , Y_3)(X_2 , Y_4) and (X_2, Y_3)(X_1, Y_4). This pixel will be denominated representative point of the object.

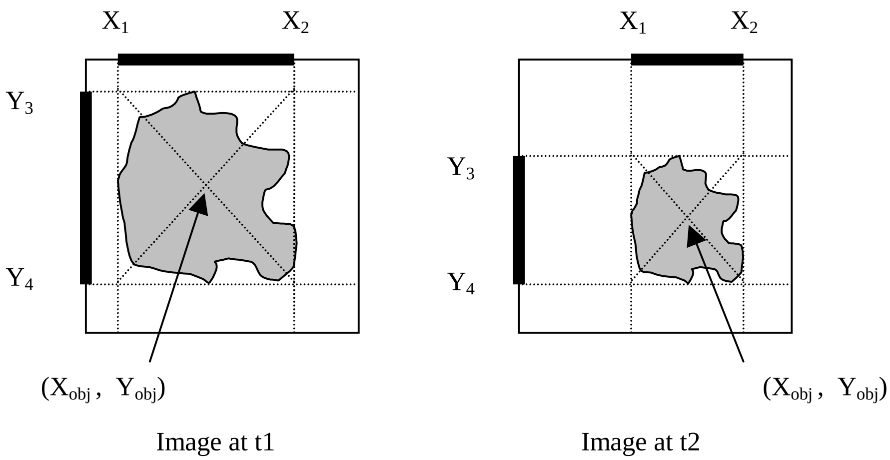

Figure 2. Obtaining of the maximum line segments and of the representative point of the object

3 Motion evaluation

Once the maximum line segments and the representative point of an object have been obtained in a sequence of images, it should be rather simple to detect a lot of motion cases. Anyway, if we consider the following possibilities:

(a) no motion (N),

(b) translation in X or Y-axis (T),

(c) dilation, or translation in Z-axis (D), and,

(d) rotation (R)

we may only obtain by combining them the following possibilities:

(A)	N	no motion detected
(B)	T	pure translation detected
(C)	TD	translation plus dilation detected
(D)	TR	translation plus rotation detected

(E) TDR translation plus dilation plus rotation detected

(F) D pure dilation detected

(G) DR dilation plus rotation detected

(H) R pure rotation detected

We consider the previous states to appear in most cases as shown in graph 1. Graph 1 shows the different possibilities when no change is detected in the representative point's co-ordinates enclosed in brackets. When there is a change in the co-ordinates of the representative point, a T has been added before the result enclosed in parenthesis.

Horizontal segment from image k-1 to image k	Vertical segment from image k-1 to image k	Horizontal versus vertical enlargement scale		
		Similar	[D]	(TD)
	Larger	Different	[DR]	(TDR)
Larger	Equal		[R]	(TR)
	Smaller		[R]	(TR)
	Larger		[R]	(TR)
Equal	Equal		[N]	(T)
	Smaller		[R]	(TR)
	Larger		[R]	(TR)
Smaller	Equal		[R]	(TR)
	Smaller	Similar	[D]	(TD)
		Different	[DR]	(TDR)

Of course, we assume the possibility to offer some erroneous results with an unknown error rate, especially in front of some rotation examples. Nevertheless, if the number of images in a sequence is great enough, this error rate should be very little.

4 A neural implementation

A neural implementation is proposed to obtain values X_1, X_2, Y_3 and Y_4 in real time. That's why we implement an easy to handle neural structure. We start from the basic structure of the multi-functional neuron of figure 3. In this figure we have:

INH_{in} = inhibition signal coming from the preceding neuron
ACT_{in} = input activation signal of the neuron
INH_{out} = inhibition signal toward the following neuron
ACT_{out} = output activation signal of the neuron

This neuron possesses as a primary characteristic the power to be linked in series with other neurons of the same type through the signals INH as shown in figure 4. See that signal INH goes spreading with the initial value 0 (that is to say, don't inhibit) until a certain neuron presents the appropriate condition to transmit the value 1 (do inhibit) starting from that moment.

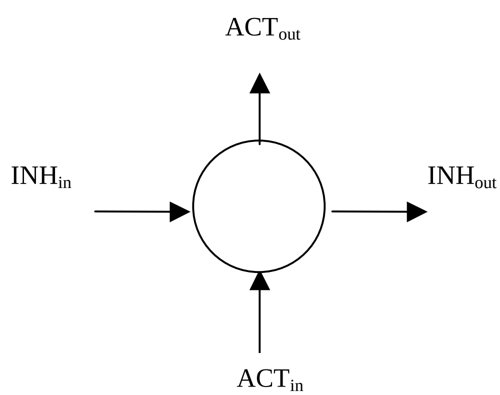

Figure 3. The multi-functional neuron

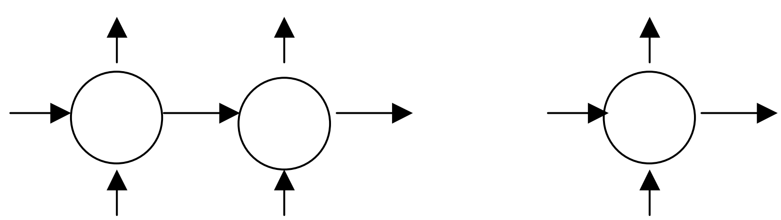

Figure 4. Neural connections

In the concrete case we are interested in, we have to detect the previously described values X_1, X_2, Y_3 and Y_4. To do this, we propose to use four arrays of neurons according to figure 5, where the ACT (ACT_{in} as well as ACT_{out}) signals' purpose is to pass the information to be processed from a lower level (obtaining of the silhouette of the object) to a higher level (calculation of the basic parameters of the object) through the neuron.

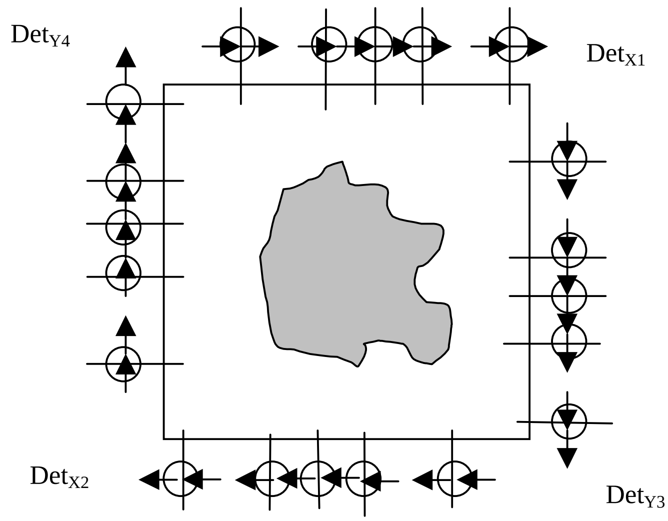

Figure 5. Determination of values X_1, X_2, Y_3 and Y_4.

The algorithm is presented here for the case of the neurons that have to detect lines Det_{X1} or Det_{X2}. For cases Det_{Y3} or Det_{Y4}, change i by j, line by column, and vice versa.

$$ACT_{in}(i, t) = \begin{cases} 1, \text{ if } \Sigma\, S(i, j, t) > 0,\ j=1..\,k \\ 0, \text{ otherwise} \end{cases} \quad (1)$$

Equation (1) tells us that column i neuron has an activation at its input if any image pixel of column i for any row j belongs to the object's silhouette.

$$INH_{out}(i, t) = \begin{cases} 1, \text{ if } INH_{in}(i, t) = 1 \cup ACT_{in}(i, t) = 1 \\ 0, \text{ otherwise} \end{cases} \quad (2)$$

Signal INH_{out} goes spreading through the line of neurons with value 0 until one of two possible events happens (they don't have to be exclusive to each other): (a) an inhibition value of

1 arrives to the neuron, or, (b) the neuron receives an activation signal from the preceding level. In both cases, the signal INH_{out} begins to spread with a value of 1.

$$ACT_{out}(i, t) = i * ACT_{in}(i, t) * [1 - INH_{in}(i, t)] \quad (3)$$

This last equation shows the behaviour of the neuron in its output toward the following level. In this case the elected function allows to elevate to higher instances the value of the position of the neuron inside the line.

We see as, in each array, at most one neuron will pass the value from its position to the next level. All the other ones take a value of 0. The “winner” neuron is the first one that detects where a silhouette’s pixel is found. This way:

Det_{X1}: is able to obtain the position of the column where silhouette appears more to the left, that is to say X_1

Det_{X2}: is able to obtain the position of the column where silhouette appears more to the right, that is to say X_2

Det_{Y3}: is able to obtain the position of the row where silhouette appears more to the top, that is to say Y_3

Det_{Y4}: is able to obtain the position of the row where silhouette appears more to the bottom, that is to say Y_4

4 Tests and results

The algorithms have been applied to the synthetic sequences SOFA 1, 2 and 3 in a software prototype programmed under Visual Microsoft C++. We thank the courtesy of Computer Vision Group, Heriot-Watt University (http://www.cee.hw.ac.uk/~mtc/sofa) for the permission of use of the images.

Figure 6 shows some examples of the 20 images that compose each one of the sequences. In the three sequences we only segment the cube that appears in them, starting from standard techniques, in order to be able to apply our algorithm to the traced silhouettes. We may see how in sequence 1 the cube is rotating, in sequence 2 it goes approaching to the observer, while in sequence 3, the cube goes approaching while it makes a slight inclination.

Our algorithms offer the following direct results: (a) for number 1 sequence, TR, (b) for number 2 sequence, D, and, (c) for number 3 sequence, DR. So, sequences 2 and 3 are correctly classified in their motion possibilities. Sequence 1 throws a raw result of TR, while the correct answer should be R. This is because the algorithm doesn’t differentiate between changes and little changes in the representative point’s co-ordinates. Introducing a lower limit for the detection of this change would throw the correct answer.

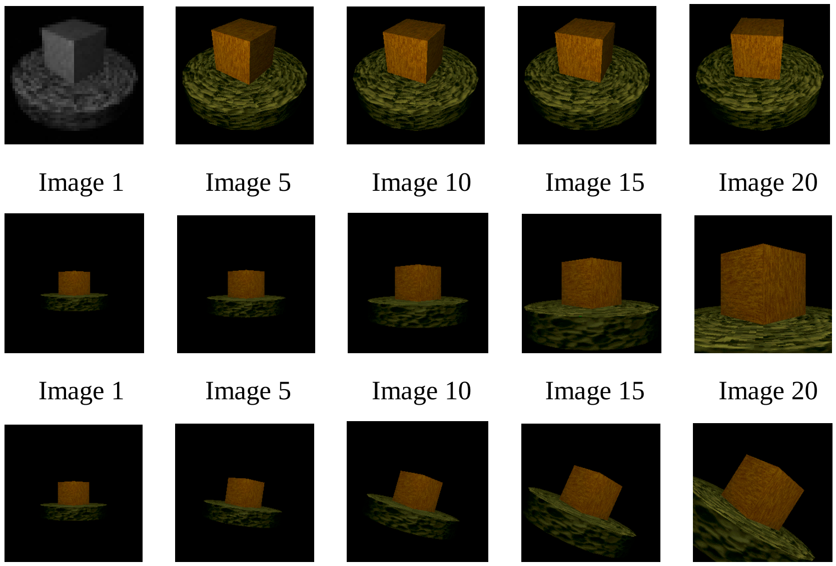

Figure 6. Some images of the SOFA sequences.

Image	X1	X2	Y3	Y4	Xobj	Yobj
S1img1	68	189	23	146	128.5	84.5
S1img2	68	189	23	146	128.5	84.5
S1img3	67	188	23	145	127.5	84
S1img4	67	188	23	145	127.5	84
S1img5	67	187	23	145	127	84
s1img6	67	187	23	145	127	84
s1img7	67	186	23	145	126.5	84
s1img8	67	185	23	145	126	84
s1img9	68	184	23	144	126	83.5
s1img10	68	183	24	144	125.5	84
s1img11	68	182	24	144	125	84
s1img12	69	181	24	143	125	83.5
s1img13	69	180	24	142	124.5	83
s1img14	70	179	24	142	124.5	83
s1img15	71	178	25	141	124.5	83
s1img16	72	177	25	141	124.5	83
s1img17	72	176	25	140	124	82.5
s1img18	73	174	25	139	123.5	82
s1img19	74	173	26	139	123.5	82.5
s1img20	76	172	26	138	124	82

Table 1. Results for the SOFA 1 sequence.

Image	X1	X2	Y3	Y4	Xobj	Yobj
S2img1	100	157	106	151	128.5	128.5
S2img2	99	158	105	152	128.5	128.5
S2img3	98	159	104	153	128.5	128.5
S2img4	97	160	103	154	128.5	128.5
S2img5	95	162	102	155	128.5	128.5
S2img6	94	163	101	156	128.5	128.5
S2img7	93	164	100	157	128.5	128.5
S2img8	91	166	98	159	128.5	128.5
S2img9	89	168	97	160	128.5	128.5
S2img10	87	170	95	162	128.5	128.5
S2img11	85	172	93	164	128.5	128.5
S2img12	83	174	91	166	128.5	128.5
S2img13	80	177	88	169	128.5	128.5
s2img14	78	179	85	172	128.5	128.5
s2img15	74	183	82	175	128.5	128.5
s2img16	71	186	78	179	128.5	128.5
s2img17	67	190	74	183	128.5	128.5
s2img18	62	195	68	189	128.5	128.5
s2img19	56	201	62	195	128.5	128.5
s2img20	50	207	53	204	128.5	128.5

Table 2. Results for the SOFA 2 sequence.

Image	X1	X2	Y3	Y4	Xobj	Yobj
S3img1	100	157	106	151	128.5	128.5
s3img2	98	159	106	152	128.5	129
s3img3	96	161	105	152	128.5	128.5
s3img4	95	162	103	154	128.5	128.5
s3img5	92	165	101	156	128.5	128.5
s3img6	90	167	99	158	128.5	128.5
s3img7	88	169	96	161	128.5	128.5
s3img8	86	171	94	163	128.5	128.5
s3img9	83	174	91	166	128.5	128.5
s3img10	81	176	88	169	128.5	128.5
s3img11	78	179	85	172	128.5	128.5
s3img12	74	183	82	175	128.5	128.5
s3img13	71	186	78	179	128.5	128.5
s3img14	67	190	74	183	128.5	128.5
s3img15	63	194	69	188	128.5	128.5
s3img16	58	199	64	193	128.5	128.5
s3img17	53	204	58	199	128.5	128.5
s3img18	47	210	52	205	128.5	128.5
s3img19	40	217	45	212	128.5	128.5
s3img20	32	225	36	221	128.5	128.5

Table 3. Results for the SOFA 3 sequence.

5 Conclusions

A simple but effective method for the detection of an important parameter of an object in movement (its size), has been presented in this paper. The algorithm is likely to be implemented in hardware, using neural mechanisms, pursuing the objective of obtaining the searched data in real time.

Our research team is specially interested in extracting simple feature characteristics of the moving objects in image sequences. Therefore, the image segmentation phase doesn't fit too much in our recent contributions. We are rather paying special attention on the analysis of parameters of motion.

6 References

1. M.A. Fernandez & J. Mira. (1993). Permanence memory: A system for real time motion analysis in image sequences. *MVA'92 IAPR Workshop on Machine Vision Applications*, 249-252. Tokyo.
2. M.A. Fernandez. (1993). Análisis de movimiento en secuencias de imagen. In: *Notas de Visión y Apuntes sobre la Ingeniería del Software*, III Curso de Verano de Informática, Colección Estudios, 24, Universidad de Castilla-La Mancha, 99-110.
3. M.A. Fernandez, J. Mira, M.T. López, J.R. Alvarez, A. Manjarrés & S. Barro. (1995). Local accumulation of persistent activity at synaptic level: Application to motion analysis. *From Natural to Artificial Neural Computation*, IWANN'95, 137-143. Springer-Verlag, Germany.
4. M.A. Fernandez. (1997). Una arquitectura modular de inspiración biológica con capacidad de aprendizaje para el análisis de movimiento en secuencias de imagen en tiempo real. Tesis Doctorales, 48. Universidad de Castilla-La Mancha. Ph. D. thesis.
5. A. Fernandez Caballero, M.D. Lozano & A. Pons. (1998). La inspiración biológica de los modelos computacionales de análisis de movimiento de imágenes. *Ensayos*, 12, December 1997, Universidad de Castilla-La Mancha.
6. A. Mitiche & P. Bouthemy. (1996). Computation and analysis of image motion: A synopsis of current problems and methods. *International Journal of Computer Vision*, vol. 19, issue 1, 29-55.
7. K. Mutch. (1986). Determining object translation information using stereoscopic motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, not. 6, 750-755.
8. K. Mutch & L.C. Heiny. (1986). Calculating object size from stereoscopic motion. Proceedings, CVPR86. IEEE Publ. 86CH2290-5, 183-187.