Fuzzy logic to track objects from MPEG video sequences

L. Rodriguez-Benitez
E.U.P. de Almaden
Univ. de Castilla-la Mancha
luis.rodriguez@uclm.es

J. Moreno-Garcia
E.U.I.T.I. de Toledo
Univ. de Castilla-la Mancha
juan.moreno@uclm.es

J.J. Castro-Sanchez, L. Jimenez
E.S.I. de Ciudad Real
Univ. de Castilla-la Mancha
josejesus.castro,luis.jimenez@uclm.es

Abstract

This work presents a tracking algorithm that takes as input a MPEG compressed video. Our algorithm makes use of fuzzy logic during the video data extraction, the segmentation of regions with similar motion parameters and finally, the tracking of real objects. The presented method is tested on tracking of vehicles in traffic scenes obtaining good results in complex situations.

Keywords: Tracking of objects, MPEG video, fuzzy logic, linguistic labels.

1 Introduction.

In Computer Vision, the low level processes extract information of the pixel intensity of an image. Using this information those regions with similar characteristics are searched to detect objects that can be relevant to a concrete application. Object tracking throughout a video sequence can be seen like a process divided into two phases: (1) an object segmentation in each frame is done, (2) the correspondences between objects are established frame to frame.

This work is different in two aspects with classical techniques of computer vision. On the one hand, we work directly over the compressed video signal by using only the information related to the motion. On the other hand, our segmentation process is based in approximate reasoning techniques applied to the motion data of a MPEG video sequence. The movement and the position of the pixels are represented by using the *Linguistic Motion Vectors*. These are grouped using as criteria the spatial proximity and a similarity measurement with respect to a conceptual representation of a region (blob) denoted as *Linguistic Blob*. The Linguistic Blobs constitute the input of the algorithm presented in this work, that is, the input data are the linguistic blobs obtained from the segmentation of every frame in the video sequence [3]. We present an algorithm that determines the correspondence between image regions that represent the same real object in different frames of a video scene. The algorithm is based in a fuzzy similarity measurement and in a set of spatiotemporal constraints. The output of the algorithm is a list with all objects positions along the scene grouped in a conceptual representation denoted as *Linguistic Object*. We apply our method to track vehicles in traffic scenes with multiple objects in motion, vehicle occlusions, variable conditions of illumination, etc.

Section 2 describes the elements that compose the compressed video signal and that are used in this work. After that, a review of some papers that use the obtained data by analyzing the compressed video signal is presented. In Section 3 we show our works related to object segmentation to facilitate the understanding of this paper. Section 4 explains the tracking method presented in this work. Section 5 presents the obtained results in the different experiments. Finally, conclusions and future

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 675–681
Torremolinos (Málaga), June 22–27, 2008

works are given in Section 6.

2 Motion Data on MPEG compressed domain.

In this section, we present some information on MPEG standard and introduce basic notation that will be needed to understand the suggested technique. Concretely, we describe the kind of frames stored in the MPEG stream and how the motion information is represented through the motion vectors.

2.1 MPEG Stream Structure

There are three kinds of frames [1]: 1. Intra(I): Encodes the image by 8x8 blockwise Discrete Cosine Transform and Variable Length Coding. 2. Predicted(P): These kind of frames are coded using motion-compensated prediction from a previous P or I picture. This is known as forward prediction. 3. Bi-directionally predicted (B): These frames are coded by using both, past and future frames, as reference. This is known as bi-directional prediction.

The macroblock is the basic unit in the MPEG stream and it is an area of 16 by 16 pixels and within this the motion vectors are stored. Each pixel has a luminance (Y) component and two chrominance components associated (Cb and Cr) with it. In this paper we are interested in how motion vectors carry the displacement of the current macroblock with respect to a previous or next reference frame.

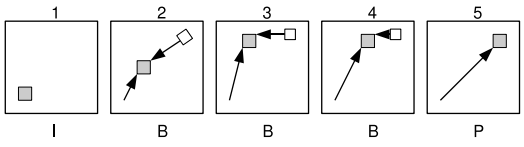

Figure 1: Motion vectors associated with one macroblock

The motion information in an MPEG stream video is stored in the Motion Vectors (MVs). There are usually only small movements from one frame to another one in a video sequence. For this reason, the macroblocks can be compared between frames, as it is shown in Figure 1 instead of encoding the whole macroblock, the difference between the two macroblocks is encoded. The displacement between two macroblocks in different frames gives the motion vector associated with some macroblock. A vector defines a distance and a direction and has two components: right_x and down_x.

3 Authors' previous work

Before describing the operation of the system, we report the obtention of the object segmentation by analyzing the compressed video signal.

[2] is the first reference in the literature to the linguistic motion vector (LMV). This concept is a linguistic description of the velocity and the motion direction of a motion vector. To obtain this conceptual representation, the magnitudes of the vector and the position of the macroblock in the picture are translated from the crisp domain to the fuzzy domain. To do this, we use a linguistic variable [6] associated with each one of the input variables. These variables are the vertical and horizontal velocities (magnitudes) of the vector, and the vertical and horizontal positions of the macroblock (number of column and row respectively). The values of the linguistic variables are consisted of a sets of linguistic labels. Figures 2 to 5 show the ordered sets used in this work, where the vertical and horizontal velocities are the variables X_{vv} and X_{hv} respectively, and the vertical and horizontal positions are the variables X_{vp} and X_{hp} respectively.

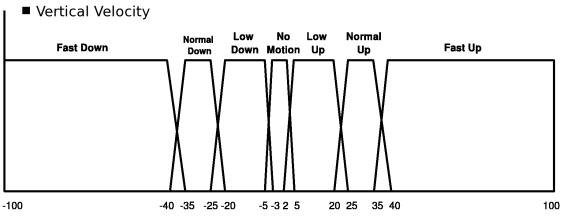

Figure 2: Variable X_{vv}

A motion objects segmentation algorithm is presented in [3]. Its output is a set of groups of macroblocks with similar motion vectors situated in closed positions. These groups of

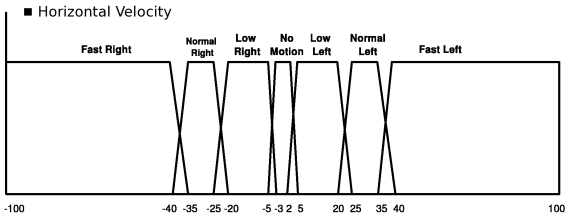

Figure 3: Variable X_{hv}

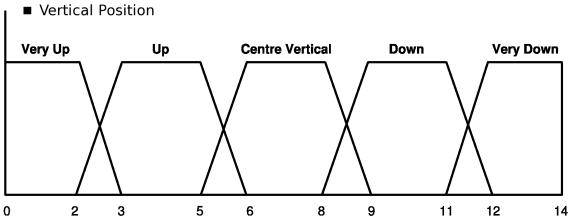

Figure 4: Variable X_{vp}

macroblocks are named as *Blobs* [4], and its corresponding fuzzy representation is called *Linguistic Blob*. To obtain an object motion description from a video scene, we must establish correspondences between blobs that are present in different frames of the sequence. The aim of this work is to establish these correspondences, that is, the object tracking in the video scene.

A Linguistic Blob (LB) is the sextuple:

$$< FN, Size, I_{hv}(r), I_{vv}(d), I_{vp}(ro), I_{hp}(c) >$$

where FN is the number of frame where the LB is situated, the $Size$ indicates the number of LMVs associated to it, i.e. its size, and the last four components $(I_{hv}(r), I_{vv}(d), I_{vp}(ro), I_{hp}(c))$ are the linguistic intervals that represent the velocity and the position of the Blob, where r is $right\_x$, d is $down\_x$, ro is the image row where the macroblock is located and c is the column.

Now, we must define the concept of linguistic interval: a linguistic interval is an ordered set of linguistic labels taken from a linguistic variable and its membership grades. The linguistic labels are the labels whose membership grade is greater than 0 after the fuzzification of an input value x over the domain of that linguistic variable. A linguistic interval is represented as $I_j^{p,q}(x)$ where p and q are its

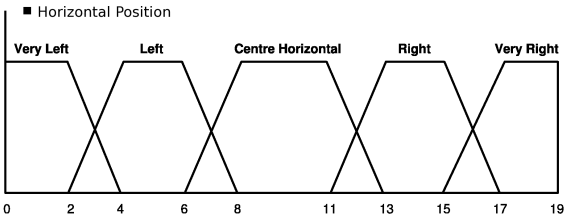

Figure 5: Variable X_{hp}

first and last labels respectively, and j identifies the input variable.

4 Tracking algorithm

This section presents an algorithm to establish the correspondence between LBs of different frames to track the objects throughout the video sequence. First, we present the definition of Linguistic Object. It allows to represent linguistically the position and direction of the object motion.

Table 1: Example of Linguistic Object

IF = 2
FF = 9
NF = 6
Size = 2.82
LB_2 Size=3:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,1\}\}$
LB_3 Size=4:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,1\}\}$
LB_5 Size=1:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,1\}\}$
LB_6 Size=3:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,1\}\}$
LB_8 Size=2:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,0.75,R,0.25\}\}$
LB_9 Size=4:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,0.75,R,0.25\}\}$

A Linguistic Object is the quintuple:

$$< IF, FF, NF, Size, \{ListBlobs\} >$$

where IF and FF are the initial and final frames that defines the time interval during the object is present in the scene, NF is the number of frames with object motion information, $Size$ is the average size of the LBs

and *ListBlobs* is a list of all the LBs that compound the object ($\{LB_{IF}, \ldots, LB_{FF}\}$).

Table 1 shows an example of a linguistic object where its initial and final frame are the frames 2 and 9 respectively, its size is 2.82, it is compounded by 6 frames (LB_2, LB_3, LB_5, LB_6, LB_8, and LB_9). So, *ListBlobs* represents that the object is moving Low Right, its velocity is null (No Motion), its vertical position is maintained constant (Down) and its horizontal position changes from "Center Horizontal" to "Center Horizontal and Right".

The tracking process is done by using the following Algorithm:

```
1.Filtering based on MinimumSizeLBs
2.Creating a LO for each LB in 1st frame
3.for j = 2 to |FRAMES|
     3.1 Comparing LBs and LOs
     3.2 Aggregating or creating LOs
4.Deleting LOs present in few frames
5.Deleting LOs with small size
```

The first action consists in eliminating those LBs with a little size (sentence 1 and Section 4.1) and initializing each LO by using the LBs in the first frame (sentence 2 and Section 4.1). Now the main loop is executed (sentence 3 and Section 4.2). This loop is used for comparing the LOs with the LBs (sentence 3.1 and Section 4.3) and for aggregating or creating a LO (sentence 3.2 and Section 4.4). Finally, when the main loop is finished, those LOs detected in a few number of frames (sentence 4 and Section 4.5) or with small size are deleted (sentence 5 and Section 4.5). The following subsections explain in detail the actions of the algorithm.

4.1 Initializing and filtering

Input data are those LBs obtained from the segmentation process with a size greater or equal than the system parameter called *MinimumSizeLBs* (Sentence 1). For example, if this parameter takes the value 4 only LB_3 and LB_9 fulfill this condition (Table 1). This process eliminates LBs caused by shadows, brightness, noise, etc. These LBs are characterized by a small size.

Each LB of the first frame with motion information is considered a new *Linguistic Object* in the initialization process (Sentence 2). If the two first LBs are $\{15, 3, \{SM, 1\}, \{SM, 0.33, AR, 0.66\}, \{AB, 1\}, \{CH, 1\}\}$ *and* $\{15, 4, \{LAR, 1\}, \{SM, 0\}, \{CV, 1\}, \{I, 1\}\}$, then there are two LOs that take the values $<$ 15, 15, 1, 3 $>$ and $<$ 15, 15, 1, 4 $>$ respectively (the ListBlobs of each object are not written in this example).

4.2 Comparing between LBs and LOs

The algorithm's main loop is used to do the comparison process between the LBs of each frame and all the LOs created previously. As it can be observed, it is a comparison between different concepts, but the comparison is really made between a LB and the last LB of *ListBlobs* in a LO. That is because it represents the last known position of the object. When the results of the comparison determines whether a LB and a LO are similar the LB is added to the LO. There are two restrictions which limits the LBs and LOs that could be associated:

1. $LO(FF) - LB(FN) > DistanceFrames$: we assume continuity of the object motion to avoid confusions with objects that could appear later in the same area of the image. So, the distance between the final frame of LO and the current frame of LB number cannot be greater than the parameter $DistanceFrames$. For example, if the parameter $DistanceFrames$ is 10, the LO shown in Table 1 is not compared with LB_{33} since the constraint is not fulfilled (LO_{FF} is LB_9).
2. Spatio-temporal restrictions: The motion direction information of a LO in a frame t (I_{hv}, I_{vv}) could be used to predict possible correspondences with a LB in a future frame (t'). It will not be considered the unions between LBs that not fulfill these intuitive constraints. For example, if I_{hv} and I_{hp} in LB_{FF} take the values *"Low Right"* and *"Left"* respectively, and the LB in the frame t' takes a value for I_{hp} equal to *"Very Left"* then this situation cannot occur because if a LO is situated to the "Left" and the displacement of

the object is to the "right", it is not possible that its following position is "Very Left".

4.3 Parameters related to the aggregation

When the constraints detailed in Section 4.2 are applied, a subset of LBs (candidates to represent a new object position in the sequence) are not considered in the selection process. This Section explains the determination of the parameters that allows the selection of a LB with respect to other options. We would like to highlight that for each new studied frame, a single LB must be associated to a LO, so a single LB can be aggregated to the LO.

We select the LB of the current frame LB_{CF} that fulfills the Equation 1 (LB_{CF} and LO must be sufficiently close), and that obtains the minimum value of the function TD.

$$TD(LO,\ LB_{CF}) < \delta \tag{1}$$

where Total Distance (TD) is defined in Equation 2.

$$TD(LO, LB_y) = max(D_{hv}, D_{vv}, D_{vp}, D_{hp}) \tag{2}$$

As it is shown, TD is the maximum of the distance of the input variables (hv, vv, vp and hp). Equation 2 uses a distance measurement D between each one of the intervals of LB_{CF} and the intervals of $LB_{FF} \in LO$ ($I_j^{p_a,q_a}(x_a)$ and $I_j^{p_b,q_b}(x_b)$ in Equation 3). This distance is calculated using Equation 3.

$$D(I, I') = \left| \frac{C(I_j^{p_a,q_a}(x_a)) - C(I_j^{p_b,q_b}(x_b))}{Max_j - Min_j} \right| \tag{3}$$

where $C(I_j^{p_j,q_j}(x_j))$ is the central value of the interval, and Max_j and Min_j are the maximum and minimum values of the support of X_j respectively.

We observe empirically that this distance is not sufficient to obtain optimal results. We can calculate a weighted value of distance between LO and LB_{CF} (WD) using another information related to LO and LB_{CF} (Equation 4). In Equation 4, $TD(LO,\ LB_{CF})$ is written as TD.

$$WD = A{*}TD{+}B{*}LB_{CF}(Size){+}C{*}LO(NF) \tag{4}$$

The parameters of Equation 4 (A, B and C) are tuned empirically taken the values $A = -0.5$, $B = 0.1$ and $C = 0.4$.

Equation 4 benefits the union with LOs with a "stabilized trajectory" (the LOs with greater number of frames) respect to new LOs (parameter C). The parameter B is used to indicate that the LBs with greater size are better candidates than LBs with small size since there is a greater certainty that they correspond with objects and not with noise. In brief, Equation 4 takes into account the total distance (TD), the size of LB_{CF} and the size of LO.

The final union is done between the LO and the LB that satisfies Equation 1 and maximizes Equation 4. δ is tuned empirically too.

4.4 Aggregating linguistic blobs

An aggregation is done when a LB_{CF} is selected. The items of the modified LO takes the following values:

1. LO(FF)=CF
2. LB_{CF} is added to ListBlobs of LO.
3. LO(Size) is modified using Equation 5.

$$\frac{LB(Size) * LO(NF)}{LO(NF)+1} + \frac{LB(Size)}{LO(NF)+1} \tag{5}$$

For example, Table 2 shows the aggregation of LB_{11}:
$\{11, 3, \{\{LD, 1], \{NM, 1], \{AB, 1], \{CH, 0.75, D, 0.25\}\}\}$ to the LO shown in Table 1. The new values of FF (final frame) and NF (Number of Frames) are 11 and 7 respectively. The new size is $LO(Size) = \frac{2.82*6}{7} + \frac{3}{7}$=2.84.

4.5 Eliminating noisy LOs

When the tracking process is finished, the obtained LOs caused by the noise are elim-

Table 2: Aggregating a LB into a LO

IF: 2
FF: 11
FN: 7
Size: 2.84
LB_2 Size=3:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,1\}\}$
LB_3 Size=4:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,1\}\}$
LB_5 Size=1:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,1\}\}$
LB_6 Size=3:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,1\}\}$
LB_8 Size=2:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,0.75,R,0.25\}\}$
LB_9 Size=4:
$\{\{LR,1\},\{NM,1\},\{D,1\},\{CH,0.75,R,0.25\}\}$
LB_{11} Size=3:
$\{\{LR,1],\{NM,1],\{D,1\},\{CH,0.75,R,0.25\}\}$

inated. We use two parameters for this purpose: *MinSizeObject* and *MinFrames*. Those LOs that fulfills Equations 6 and 7 are deleted.

$$LO(FF) - LO(IF) < MinFrames \quad (6)$$

$$LO(Size) < MinSizeObject \quad (7)$$

In brief, the LOs with a reduce number of frames or with a small size are deleted from the set of real objects in the scene.

5 Experimental results

Our method is tested to track vehicles in traffic scenes in a highway of three tracks. The central track is used to turns and incorporations to the route, with a flow approximated of 2.200 vehicles to the hour in rush hour. We use a webcam located inside a car to record the traffic scenes. The position of the camera is different if we compare with the applications of traffic monitoring (they usually use aerial scenes). The sampling frequency is 30 frames per second and a resolution of 320X240 pixels.

The results of 8 experiments are shown in Tables from 3 to 5. The values of the system parameters used in the experiments are presented in Table 3. In Table 4, each row is a sequence detailing the size of MPEG video (kilobytes), the duration (seconds), the total number of frames and the total number of motion vectors. The last row is the sum of each one of the shown values.

Table 3: System Parameters

Parameter	Value
MinimumSizeLBs	3
DistanceFrames	15
δ	0.3
MinFrames	8
MinSizeObject	1.3

Table 4: Input data

Size	Time	Frames	MVs
3891	27	828	4661
3174	23	680	4008
2458	17	519	2430
3788	27	819	4914
2300	31	936	6831
2530	34	1012	7812
1683	22	673	5059
2533	34	1014	7786
22357	215	6481	43501

Table 5 shows the analysis of the results. Each row corresponds with an experiment, except the last two rows that represent the total results expressed in absolute values and percentage respectively. The information shown is the total number of vehicles in the scene, the vehicles detected (Correct) and the vehicles not detected (Incorrect).

As it can be observed, 118 vehicles are correctly detected of a total of 156 vehicles in the scene. It means that more than a 75% are well tracked. The tracking errors are mainly caused by big vehicles (especially trucks and buses). That is because these vehicles cause that the information of the motion vectors are very dispersed. Another problem is caused by the lack of motion vectors in those situations when the average speed is low. As there is

not enough input data the system does not respond as expected.

Table 5: Final Results

Vehicle	Correct	Incorrect
20	14	6
16	10	6
13	10	3
17	17	0
19	14	5
27	20	7
18	13	5
26	20	6
156	118	38
100%	75,6%	24,4%

6 Conclusions

A method to obtain a linguistic description of motion objects by using a MPEG video is presented in this work. Some conceptual linguistic representations based on linguistic labels are used. These labels allow to describe in a comprehensible way the object motion in the scene. Fuzzy logic [5] is a good method to manage the imprecision, and it helps to manipulate the inherent noise present in compressed video. The obtained vehicle trajectories are easily interpretable.

An advantage of our method is that the size of the input data is very small if it is compared with other computer vision systems. Our method does not need to decompress the MPEG video, so we obtain a good execution time. The proposed method does not limit the number of objects to detect. The obtained results are good (75% of the vehicles correctly detected) considering the low quality of the input video.

As future works, we are going to improve our method to execute our system in parallel with different values of configuration, different definitions of linguistic variables and different parameters. These values depend on the class of object to track, and we believe that this must avoid the problems detected in the experiments (big vehicles).

Acknowledgements

This work has been funded by the Spanish Ministry of Education and Science (TIN2007-62568) and the Regional Government of Castilla-la Mancha (PAC06-0141 and PBC06-0064).

References

[1] Moving Picture Experts Group. The MPEG home page. Available in: *http://www.chiariglione.org/mpeg/*.

[2] L. Rodriguez-Benitez and J. Moreno-Garcia and J.J. Castro-Schez and L. Jimenez, Linguistic Motion Description for an Object on MPEG Compressed Domain, In *Proceedings of Eleventh International Fuzzy Systems Association World Congress*, International Fuzzy Systems Association, 2005.

[3] L. Rodriguez-Benitez and J. Moreno-Garcia and J.J. Castro-Schez and L. Jimenez, An approximate reasoning technique for segmentation on compressed MPEG video, In *Proceedings of VISAPP, 2nd International Conference on Computer Vision Theory and Applications*, 2007.

[4] Christopher Richard Wren and Ali Azarbayejani and Trevor Darrell and Alex Pentland, Pfinder: Real-Time Tracking of the Human Body, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19 (7) (1997) 780–785.

[5] L.A. Zadeh, *Fuzzy Set*, Information and Control, 1960.

[6] L.A. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning, *Information Science*, 1975.