Abstract – MPEG-4 based video coding applications require the segmentation of each video image in its principal moving objects to be coded independently from each other.

Several techniques of video objects segmentation for coding purposes have been presented in literature; all such segmentation techniques are based on the smart soft-thresholding of the motion fields, the best ones dealing with dense motion fields.

Anyway, MPEG-4 based coding structures require a block based (sparse) motion field estimation. The use of block based coding structures, don’t allow fair video objects segmentation for the intrinsic inaccuracy of motion estimate the block based structure of the motion field, specially on moving object border blocks.

In this context the segmentation obtained basing only on motion information is inaccurate, but it can be enhanced by the joint use of several information at hand, like color, motion, frame difference, prediction error, texture and so on.

In this work a locally connected unsupervised neural network approach is presented, to obtain the segmentation of a moving video object (VO) on a fixed or slow–translating background.

The purpose of background-foreground segmentation is here addressed by a split and merge like criterion: the neural network is applied to avail on correlation between several image components such as color, motion vectors and frame difference to simplify each image of the sequence into disjoint regions, thus reducing the dimensionality of the problem. The segmentation is obtained working on sub-spaces the whole subspace spanned by the several image components used to define the image, in a partition tree scheme.

The second step consists on a Principal Component Analysis (PCA) to simplify the several clusters obtained in the first phase and obtain a two sets partition of the whole image. As the main information used in this phase deal principally on motion estimates and frame difference the PCA simplification step is likely to produce an image partition close to the searched BF segmentation.

Preliminary results of the proposed method applied on “Foreman” video sequence seem promising. Network parameters tuning to make the proposed algorithm feasible is still needed; applications to a MPEG-4 based video coding structure is a future goal we will pursue. A complete neural network structure designed to produce the searched image segmentation, based on results of this preliminary study, is under construction.

I. INTRODUCTION

A fundamental task for MPEG-4 based video coding applications is the possibility of segmenting each image in its principal moving objects to be coded independently from the others.

In MPEG-4 standard, each object originates a bit stream, and the several bit streams obtained, one for each defined object in the scene, are multiplexed, together with side information, to the receiving link end. This way the coded frame can adapt both coding parameters and bit budget of a given link in a very flexible and bandwidth adaptive video coding structure.

Though studio coding of a multimedia application can be a relatively simple task, real time coding of video sequences acquired by portable handsets (as for personal video communication over wireless channels, UMTS) requires simple, easily reliable and efficient object segmentation procedures.

To properly deal with video segmentation into objects, motion is a fundamental source of information, but its use regardless of other available information can produce poor results, specially when applied on block based motion estimation algorithms. A very good review paper dealing with the challenging problem of motion segmentation is [1].

To produce good results in the segmentation of the video sequence, both intra and inter-frame information should be exploited; color, shape, texture and contours, as intra frame information, and motion, object temporal shape, color and motion coherence should be used to properly define a cost function whose minimization should give the searched Background Foreground video segmentation.

Instead of using heuristic considerations in defining the functional, in this work, a locally connected unsupervised neural network is used to simplify the huge amount of vector information available at the coder hand, in order to obtain an image partition based on both pictorial information (mainly color) and motion information (block motion description). Once a proper image partition is obtained the goal is to assign each detected cluster to one between two possible image sub-sets corresponding to foreground and background.

We pursue a blind (e.g. non model-based) Background-Foreground (BF) segmentation of “head and shoulders” class of video sequences.

Several authors have proposed the joint use of information for video sequences motion segmentation, and a few (at our knowledge) the use of neural networks.

Our work differs with other multidimensional approaches presented in literature because we make use of a arbitrarily shaped block based motion estimation technique, thus being able to adapt to MPEG-4 like object oriented coding frameworks, instead of dense motion fields. A simple and efficient object oriented block based motion estimation procedure is used at this aim ([5]).

To obtain the image segmentation the proposed approach performs two steps in a split and merge goal oriented framework: the first step is used to split the image in subsets on the basis of the color and motion information obtained besides the motion estimation procedure. The image is then described by its clusters, so that the problem of image segmentation results in a highly simplified one; each cluster is considered as a candidate for the inclusion either in the background or in the foreground image subsets.

The proposed technique to obtain the desired video Background-Foreground (BF) segmentation uses a direct principal components analysis (PCA) approach.

The PCA step consists in decomposing the correlation matrix of the centroids of the obtained image segmentation into its eigen-structure; image re-composition starting from the two principal components of the correlation matrix produces the searched BF image partition.

Even if a hard image decision on the several obtained clusters is applied, results on “Foreman” video sequence show that the accuracy in object detection and contour definition is fine enough to allow clusters re-aggregation into two maximally coherent in colors and motion separate subsets result in the searched BF segmentation.

Preliminary results are presented for Foreman sequence in QCIF (144 × 176) YUV color format.
II. MOTION ESTIMATION

Video coding requires motion estimation to exploit the time redundancies inherently present in video sequences. The basic principle of motion compensated predictive coding is that successive images in a sequence are similar and locally motion in the video sequence can be described as a pure translation.

In standard Block Matching (BMA) algorithms a sparse motion field is computed by means of a block based technique. The image to be predicted is split in squared blocks. For each block of the frame to be predicted, a search of the most similar block on the previous frame is carried out to find the best approximation: the block at hand is compared with all the displaced blocks of the previous image in a given search domain around its position. The displaced block whose prediction error is the least is chosen as the prediction, while its displacement from the actual block defines the motion vector.

The simplicity and good performances of Block Matching based motion estimation algorithms have favored its use in most coding standards as MPEG v.1–4 and H.261–3 (9–17)). Usual choices for the block sizes are 8x8 (block) or 16x16 (macro-block) pixels, while the search domain can vary depending on the application.

All BMA based techniques assume pure translational motion of blocks. The blocky structure of the obtained motion estimation may severely affect the predicted image (blocky artifacts). Wrong estimates of the motion vector for object border blocks often occur because of the presence, inside the block, of pixels belonging to the foreground moving object together with pixel from the background. In [20] the authors introduced the Sliced Block Matching Algorithm (S–BMA), a simple modification of the standard BMA technique, able to deal with the object shape and reduce the prediction error on the moving object border blocks reducing the blocking artifacts effect.

Basically the motion estimation technique splits blocks on the basis of the sign of the difference of two consecutive frames: stated $I_d(x,y)$ and $I_{xy}(x,y)$ two consecutive frames of the video sequence, the frame difference is defined by:

$$FD(x,y) = I_d(x,y) - I_{xy}(x,y)$$ (1)

where $(x,y)$ represents the (row, column) pixel location on the image.

The block segmentation can be obtained by the aid of a threshold, separating positive frame FD high energy zones from negative ones and isolating regions with low FD. Assuming $Th$ as a selected threshold value, three types of regions can hence be defined, as:

$$P = \{(x,y) | FD(x,y) > Th, \ 0 \leq x \leq N, \ 0 \leq y \leq M\}$$

$$N = \{(x,y) | FD(x,y) < Th, \ 0 \leq x \leq N, \ 0 \leq y \leq M\}$$

$$Z = \{(x,y) | |FD(x,y)| < \epsilon, \ 0 \leq x \leq N, \ 0 \leq y \leq M\}$$

with $N$ and $M$ the horizontal and vertical frame sizes, respectively. For each block, each detected zone is separately and independently estimated from each other thus avoiding background pixel misleading contribution to the estimation of the motion for a boundary block of the moving object. This requires the same total computational burden used by BMA, as the three areas in each block do not overlap.

III. LOCALLY CONNECTED COMPETITIVE UNSUPERVISED NEURAL NETWORK

Similarly to standard competitive neural networks, the proposed network is composed by $M$ processing elements, where each unit receives an input signal $x = (x_1, x_2, ..., x_N)$ from an external data-base and is characterized by the weight vector $\mathbf{w} = (w_1, w_2, ..., w_N)$. When the input is received, the unit computes the Euclidean distance $d(x, w)$ between the input and the weight. Then, for a single element of the network, the following competitive learning algorithm is proposed in table1.

Table 1.

<table>
<thead>
<tr>
<th>Step</th>
<th>$y(k) = f(\sigma(k) - d(x(k), w(k)))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>$\Delta w(k) = (\alpha(k) + \beta(k)) y(k)(x - w(k))$</td>
</tr>
<tr>
<td></td>
<td>$w(k + 1) = w(k) + \Delta w(k)$</td>
</tr>
<tr>
<td>Step 3</td>
<td>$\Delta \sigma(k) = (\alpha(k) + \beta(k)) y(k)\epsilon(x, w(k)) - \sigma(k))$</td>
</tr>
<tr>
<td></td>
<td>$\sigma(k + 1) = \sigma(k) + \Delta \sigma(k)$</td>
</tr>
<tr>
<td>Step 4</td>
<td>$\Delta \alpha(k) = \alpha(k) y(k)\delta$</td>
</tr>
<tr>
<td></td>
<td>$\alpha(k + 1) = \alpha(k) + \Delta \alpha(k)$</td>
</tr>
<tr>
<td></td>
<td>$\Delta \beta(k) = -\alpha(k)$</td>
</tr>
<tr>
<td></td>
<td>$\beta(k + 1) = \beta(k) + \Delta \beta(k)$</td>
</tr>
</tbody>
</table>

Step 5. Iterate to Step 1 until $\Delta \sigma(k) < \epsilon$.

In table 1 $y$ is the neuron output, is the logistic function with $f(k) = 0$ if $u \leq 0$, $\sigma$ if $u > 0$, $\alpha(k)$ is the learning coefficient, $\beta(k)$ is a coefficient that changes with the same rate of $\alpha$, $\delta$ is a constant value whereas $\sigma(k)$ is an adaptive threshold, the behavior of which will be clarified later on.

The behavior of the algorithm can be described as follows. From first equation of table 1 it is clear that the activation of the neuron $(y > 0)$ occurs only if the extreme of the input vector $x$ is within the hyper-sphere with radius given by $\sigma$ and center given by the extreme of $w$.

When $y = 0$, equation (2) is driven by the value of $\alpha$. On the other hand, when $\alpha$ goes to zero, the value of $y$ predominates and becomes a ‘winner’ indicator. In this way, it is possible to implement a learning law that generates a transition from a phase of ‘weak’ undifferentiated learning.
contribute to the learning. Simulation results are carried out to show the capability of the unit to self-tune towards a single centroid.

Network with Local Connections: Differently from WTA (Winner Take All) paradigms, the proposed network herein is an array of $N$ units, which are characterized by local inhibitory connections. In particular, the proposed network architecture is described by the following equations:

$$\Delta w_j(k) = (\alpha(k) + \beta(k)y_j(k))x - w_j(k)$$

$$\Delta w_j(k) = [(\alpha(k) + \beta(k)y_j(k))(x - w_j(k))] \theta(y_j)$$

$$\Delta w_k(k) = [(\alpha(k) + \beta(k)y_k(k))(x - w_k(k))] \theta(y_k)$$

$$\Delta w_n(k) = [(\alpha(k) + \beta(k)y_n(k))(x - w_n(k))] \theta(y_n)$$

where $\theta(\cdot)$ describes the action of the inhibitory links, with $\theta(y_j) = 0$ if $y_j > 0.5$ and $\theta(y_j) = 1$ if $y_j < 0.5$. The behavior of the network can be described as follows. At the beginning each unit receives samples from the input space. However, only the first unit learns, since it has no input inhibitory links, whereas all the other units are inhibited, due to the presence of input inhibitory links. Mathematically, this is because $\theta(y_j) = 0$ when the first unit has $y_j > 0.5$. As a consequence, from (6) it follows that the units from 2 to $N$ are inhibited. Successively, when the first unit has found a cluster center, its threshold value decreases, $y_1$ decreases and $\theta(y_1) = 1$. Therefore, the first inhibitory link is turned off and the second unit can start its learning phase. This means that the input samples, which make active the first unit, continue to produce learning only for the first unit, whereas the remaining samples produce learning for the second unit. This process is iterated by applying the following strategy:

- if $\theta(y_n) = 1$, unit $k+1$ is added to the network;
- the samples that make active the units from 1 to $k$ will produce learning only for the first $k$ units;

On the other hand, the learning stage is stopped when one of the following condition is satisfied:

- all the $N$ units have been utilized;
- the last unit added to the network satisfies equation (5).

Notice that the proposed network architecture combines the advantages of both sequential cluster search and unsupervised competitive networks (UCNN). This feature leads to a major advantage over the performances of unsupervised competitive neural networks. Namely, due to the presence of inhibitory links, lower priority units cannot affect higher priority units. As a consequence, if two networks are composed of $N$ and $(N+M)$ units, the common $N$ units will behave in the same way in both the networks and will found exactly the same centroids. This means that it is always possible to add some extra units (in order to check if some cluster in the data structure has been missed) without modifying the behavior of the first unit. Notice that UCNN’s lack of this property, that is, they suffer of the problem of the dependence of the final partitions of the data on the number of the network elements [5]. In order to show the behavior of the proposed architecture and the ability to select the proper number of units, a network composed of 8 elements in a data set composed of 4 clusters (bi-dimensional gaussian clusters with unitary variance and centered respectively in $(2,2)$, $(-2,2)$, $(-2,-2)$ and $(-2,-2)$) is considered in the following experiment. Figure 3 shows that only 5 elements have been involved in the learning process. The remaining 3 elements have their thresholds unchanged to their initial values. Furthermore, notice that only 4 elements have their threshold values lower than the steady-state value. This means that only 4 elements converge towards a dense region of the input space, that is, they have found the corresponding cluster center.

In order to comply with the possible data structure of the proposed problem, a first step of data handling has been applied. The neural network approach. In fact, uses Euclidean distance to decide about cluster regions. This is a good choice for isotropic cluster densities, but don’t fit well to anisotropic cases. The Singular Value Decomposition (SVD) space analysis has been applied, as a first step, for a proper space transformation of a possible anisotropic cluster densities to isotropic ones and then the neural network has been applied on modified input data.

The inverse transformation has been applied on the clusterized data to obtain the results presented in figure 3.

IV. PCA FOR BF IMAGE PARTITION ALGORITHM

The proposed technique to obtain the desired video Background-Foreground (BF) segmentation uses a direct approach based on the principal component analysis (PCA). Once a first image partition has been created by the proposed neural network, the PCA step consists in decomposing the correlation matrix of the centroids of the obtained image segmentation into its eigensstructure; image recomposition starting from the two principal components of the correlation matrix produces the searched BF image partition. This produces a hard image decision on the several obtained clusters, anyway its application presents the advantage of being a linear method, of easy hardware implementation and above all one step, requiring no iterations.

This way a compact object definition results very often.

V. EXPERIMENTAL RESULTS

The image partition has been applied in a layered structure, first using colors, then motion and at last frame difference. The input vector used has been obtained grouping color information $(y, u, v)$ motion information $(v_x, v_y)$ at each image pixel, and the gray level frame difference, thus expanding the image to a 6-D vector space

![Fig. 3: partition tree scheme](image-url)
application of only color information and, at the last step, for each cluster of the newly defined object segmentation, three clusters on the frame difference.

The clustering phase has been conducted on the sub-spaces of the whole 6-D space, in order to stress at best each information. The image clusters obtained refer to a 3-steps procedure, each applied to each cluster of the preceding step, as pictorially described in fig. 3. 3x3x3 cluster result on the final image, to be sent to the PCA BF image segmentation.

The image sharp contours and sub-block motion description allow the definition of areas of coherent motion and color is very close to the true image in the scene. The noisy appearance of the obtained segmentation can be easily enhanced by majority filtering ([5]). Fuzzy cluster merging originate the segmentation shown in fig. 5. Even if the obtained results don’t comprise the whole foreground moving objects parts (the hat is missing), the segmentation obtained can be considered satisfactory for several reasons:

- when motion estimates fail, the resulting segmentation cannot take care of semantic information, as the proposed approach is blind (no side information is assumed available on the object to be segmented);
- motion fields failures often happen on smooth translating surfaces (the hat); motion estimate behavior (for the segmentation process only) can be enhanced by skipping frames when motion is too slow: a variable frame rate may allow sufficient frame differences in all the zones of the moving object to be segmented. Anyway variable frame rate, at least for segmentation purposes ([7]), should be required.

VI. CONCLUSIONS AND DISCUSSION

The proposed algorithm seems promising in performances, but several aspect of the problem have still to be faced.

The main goal of this application lie in its intrinsic arbitrarily shaped block motion estimation structure, easing both segmentation and clustering.

The proposed network structure, as it is described here, anyway, reveals sensible to variations in its input parameters. This is an highly unwanted feature, as the searched algorithm should result robust and video independent.