A HIERARCHICAL TECHNIQUE FOR IMAGE SEQUENCE ANALYSIS

L. Garrido, F. Marqués, M. Pardàs, P. Salembier, V. Vilaplana

Departament de Teoria del Senyal i Communicacions
Universitat Politècnica de Catalunya
Barcelona, SPAIN

Abstract. This paper presents an image sequence analysis scheme. It combines, in a hierarchical manner, four different homogeneity criteria: gray level, motion, depth and semantic. In order to create each layer in the hierarchy, a specific approach is proposed. Finally, an example of the results obtained with the complete scheme is shown.

1 INTRODUCTION

In the framework of the current standardization processes MPEG-4 and MPEG-7, it is clear that video image analysis is and will be playing a key role. To really exploit all the functionalities provided by MPEG-4 and to help in the description of image sequences in MPEG-7, generic analysis techniques are necessary.

The complexity of the problem leads to combining several homogeneity criteria. Useful partitions should contain regions being homogeneous in gray level, color, motion, depth and/or semantic meaning. However, the correct approach to combine and estimate these types of information is not fixed yet.

The paper discusses in section 2 a general hierarchical scheme for video analysis. Specific implementations of each one of the four levels of analysis is presented in the following sections: texture, motion, depth and semantic levels in sections 3, 4, 5 and 6, respectively. Finally, section 7 shows an example of the complete scheme.

2 GENERAL ANALYSIS SCHEME

In this section, we propose a hierarchical segmentation scheme combining four homogeneity or segmentation criteria: gray level, motion, depth and semantic. The goal of the algorithm is to segment a video sequence in a recursive and causal way. To this end, at each time instant t, a gray level partition $P_g(t)$, a motion partition $P_m(t)$, a depth partition $P_d(t)$ and a semantic partition $P_s(t)$ are defined. The gray level, motion and depth partitions are made of regions that are homogeneous respectively in gray level[2], in motion and in depth (relative

[*] This work has been partially supported by the ACTS projects VIDAS, by France-Telecom/CCETT under the contract 96ME22 as well as by the project TIC 95-1022-C05-05 of the Spanish government

[2] Note that we have actually implemented and tested a gray level segmentation scheme, however the approach can be easily extended to color segmentation.

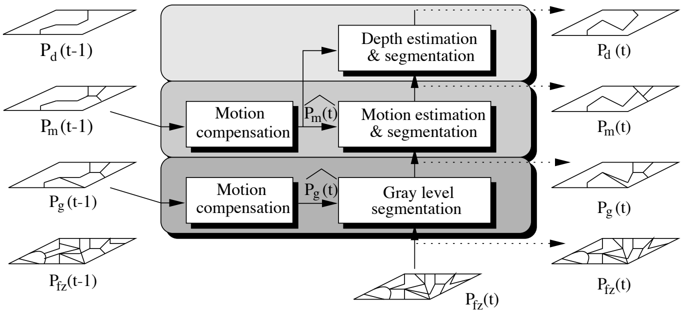


Fig. 1. Hierarchical analysis scheme

distance from the camera). The goal of the semantic partition is to define the presence of a set of regions representing a known object with a semantic meaning. In the sequel, we assume that the object of interest is a face. However, the approach is fairly general and can be extended to other type of objects.

The partitions are organized in a hierarchical way: the gray level partition is created by merging regions belonging to an initial partition which is the partition of flat zones $P_{fz}(t)$ of the original frame[3]. Similarly, the motion partition $P_m(t)$ is created by merging regions of the gray level partition $P_g(t)$ and, finally, regions of the depth partition $P_d(t)$ are built by merging of regions of the motion partition $P_m(t)$. As can be seen, the partitions are structured in a hierarchical way by merging. This structure has been selected for the following reasons:

1. Objects can be structured in a hierarchical way: objects are composed of regions belonging to the same depth plane. These regions are themselves made of a set of regions that are homogeneous in motion. Finally, each region homogeneous in motion may be composed of several regions that are homogeneous in gray level value.
2. Contours at a given level of the hierarchy are constrained by contours of the lower levels. For instance, a contour between two "motion regions", that are regions homogeneous in motion, should coincide with a contour between two "spatial regions". Similarly, contours of "depth regions" should coincide with contours of "motion regions". These restrictions have been used because a precise spatial definition of contours can only be achieved in the gray level partition since no (gray level) estimation has to be performed. Motion and depth contours may be less accurate because of the motion and depth estimation process.

[3] The flat zone partition [5] is the partition made of the largest connected components where the image is constant (a flat zone can be reduced to a single point). It can be computed either on the original frame or after preprocessing by a connected operator.

3. In order to estimate the motion and to perform a reliable motion segmentation, elementary regions should have been previously defined. In this case, the motion estimation can rely on a partition that is related to the image. This approach avoids drawbacks of block-based motion estimation. Similarly, the estimation of the depth requires the knowledge of a reduced set of regions that are homogeneous in motion so that overlapping and uncovered areas can be extracted and studied.

Moreover, we would like to track regions in time, that is to relate regions of partitions $P_g(t)$, $P_m(t)$ and $P_d(t)$ with regions of partitions $P_g(t-1)$, $P_m(t-1)$ and $P_d(t-1)$. Region tracking creates a temporal coherence in the hierarchy of partitions which is useful for the segmentation itself. For example, when estimating the motion of a region of the gray level partition at time t, one can discard pixels not belonging to this region at time $t-1$. Moreover, region tracking is mandatory if one wants to analyze the time evolution of the regions.

The global scheme is depicted in Fig. 1. Assume that the partition hierarchy $P_{fz}(t-1)$, $P_g(t-1)$, $P_m(t-1)$ and $P_d(t-1)$ at time $t-1$ is known. The purpose of the algorithm is to create a similar hierarchy at time t. Note that $P_{fz}(t)$ can be directly extracted from the original frame. The first step of the algorithm is to motion compensate partitions $P_g(t-1)$ and $P_m(t-1)$. This creates two predicted partitions $\widehat{P_g(t)}$ and $\widehat{P_m(t)}$. The depth partition $P_d(t-1)$ is not compensated as a whole because its regions are not homogeneous in motion. However, information about the past can be obtained by using $\widehat{P_m(t)}$. Then, the first segmentation to be performed is the gray level segmentation. It creates $P_g(t)$ based on the knowledge of the current partition of flat zones $P_{fz}(t)$ and the predicted gray level partition $\widehat{P_g(t)}$. Once, $P_g(t)$ has been created, the motion of each region is estimated and a second segmentation step is performed to built the motion segmentation $P_m(t)$. The next step is to estimate the relative depth of the regions of $P_m(t)$ and to create the depth partition $P_d(t)$.

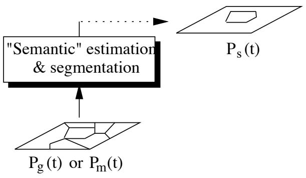


Fig. 2. Semantic analysis

The semantic partition is not directly integrated in the generic analysis scheme of Fig. 1. Indeed, this analysis level is highly related to specific applications. As shown in Fig. 2, it can be considered as a separate parallel hierarchy. The semantic partition $P_s(t)$ is created by merging regions of either the gray level or the motion partitions ($P_g(t)$ or $P_m(t)$).

In the following, each part of the analysis scheme referring to a specific criterion is called a "layer" and is more precisely discussed.

3 TEXTURE LAYER

The first layer to be built is the pure intra frame partition $P_g(0)$. This partition is created starting from $P_{fz}(0)$. Regions, which may be formed by single pixels, will be merged together until a termination criterion such as Peak Signal to Noise Ratio (PSNR) or a certain number of regions is reached. The merging of regions is done in an order defined by a distance computed for every pair of neighboring regions. This distance will be based in a measure of the cost of fusing two regions such as the square error related to the coding of the merged region with a constant model. As a result, the merging process creates regions that are homogeneous in gray level.

For the following frames, the gray level layer and the upper ones in the hierarchy handle as well the tracking of regions in time; that is, they relate regions of the partition of one layer $t-1$ to the regions of the partition t of the same layer. The creation of the gray layer partition is done in several steps. First, assuming that partition $P_g(t-1)$ is known, the forward compensation of $P_g(t-1)$ is computed $\widehat{P_g(t)}$. For that purpose, the forward motion of each region of partition $P_g(t-1)$ is estimated. Then, the partition $P_g(t-1)$ is compensated using the technique described in [4].

The compensation defines three set of areas: areas where the prediction is done without conflict, overlapping areas and uncovered areas. Overlapping areas correspond to pixels where more that one region is compensated whereas uncovered areas are zones where no region is compensated. The overlapping areas conflict can be solved using information of the depth layer, i.e. if two or more regions overlap the output at the overlapping area is the region which is closest in the sense of depth.

The compensated partition gives us an approximation of where the regions of $t-1$ are located in the new partition. In order to get reliable temporal links, a second step is performed. Using as initial partition P_{fz}, some regions of this partition are merged relying on a quality criterion (such as PSNR) with the constraint that only regions of P_{fz} contained in the same connected component of the compensated frame can be fused. The merging criterion uses gray level information from neighbor regions of $P_{fz}(t)$ and $\widehat{P_g(t)}$, so that past information is introduced in the process.

Finally, once the temporal link is established, a partition is created by continuing the merging process relaxing the PSNR criterion of the temporal link. Note that, in this process, some regions of P_{fz} may be merged with regions defined during the temporal link. These regions allow the precise definition of the shape of the regions that were present at time $t-1$. By contrast, some regions of P_{fz} are merged together without involving any of the regions defined by the temporal link. These regions are new regions appearing at time t.

4 MOTION ANALYSIS AND SEGMENTATION

As in the gray level layer, the segmentation is achieved in two steps: first, a motion compensation and, then, a merging of regions from the previous layer. The compensation is done as the gray level layer by forward motion estimation of regions of $P_m(t-1)$ and forward compensation. It creates a first approximation of the motion partition called $\widehat{P_m(t)}$.

For the merging itself, a motion estimation is performed to assign a dense motion field to each region of the gray level partition $P_g(t)$. Here also, the technique proposed in [1, 6] is used to assign to each region a polynomial model describing the apparent motion in the horizontal and the vertical directions. The model is estimated using differential methods. From this motion model, two dense motion fields are created by assigning to each pixel the values of its horizontal and vertical displacements. These two dense motion field images are now used as input to the merging algorithm. So, the merging process has to deal with a two component image. As a result, the algorithm defines regions that are homogeneous in the sense of its input images, that is in the sense of motion. During the merging process, the motion field can be re-estimated. However, in our current implementation this re-estimation is not done to limit the computational complexity of the algorithm. In future work, we will investigate fast techniques to re-estimate the motion and we will study the improvement that is obtained.

The merging algorithm itself works as the gray level algorithm: first, a temporal link is created and then the final partition is obtained by merging regions of $P_g(t)$. The merging criterion involves distances between the motion of neighboring regions of $P_g(t)$ and of $\widehat{P_m(t)}$. The main difference with the gray level layer is that the region model deals with the horizontal and vertical displacements. A first order polynomial model is used to take into account a wide range of motions (e.g.: translation, zoom and rotation).

5 DEPTH ANALYSIS AND SEGMENTATION

In this layer the partition $P_d(t)$ is created from the estimation of the relative depth between the regions of $P_m(t)$. Those neighboring regions which are found to be in the same depth level are considered as a unique region in this segmentation level. The relative depth of the regions is estimated by considering the occlusions between regions. This estimation procedure can be described in two steps.

The clues for the depth estimation are obtained from the overlapping zones which appear when the partition $\widehat{P_m(t)}$ is created. When there is relative motion between two neighboring regions (A and B) belonging to different depth levels, an overlapping zone appears at the partition $\widehat{P_m(t)}$. To decide which one of the two regions is in the foreground it has to be checked whether the pixels of the overlapping zone belong to region A or B at $P_m(t)$. The region to which they belong for the most part is assumed to be in the foreground of the other region. When processing sequences of images this overlapping information can be accumulated along the sequence, in order to have more reliable information.

In the overlapping computation step, an ordering relation between neighboring regions is obtained if there is a relative motion between them. This relation will be more reliable for some regions than for other, depending on the size of the overlapping zones, on the certainty of the overlapping decision, and on the correspondence of the overlapping with the relative motion detected between the two regions. In this step every region is assigned to a depth level considering this information. For this aim a relaxation labeling algorithm is used [3]. The inputs to this algorithm are an initial probability $p_i(\lambda)$ for every region i of being at every depth level λ. These initial probabilities are updated considering the compatibilities between neighboring regions being at different depth levels and the relative certainties of the order relation between regions, obtained in the overlapping computation step. Finally, every region is assigned to the depth level λ which maximizes $p_i(\lambda)$. The regions of $P_d(t)$ are composed of connected regions belonging to the same depth level. In order to achieve a temporal stability the relaxation algorithm is initialized with the information of $P_d(t-1)$.

6 SEMANTIC LAYER: FACE SEGMENTATION

The semantic segmentation $P_s(t)$ is built up as a two step process: the detection of a subimage that contains the object (a face) and a merging step, that specifies which regions of the subimage form the face. As it is a general class detection problem, the underlying probability distribution of the object must be considered. A subspace method and an eigenvector decomposition are used to find a parametric and compact description of faces, taking into account their statistical variability. According to [2], the class membership or likelihood function $P(x/\Omega)$ is modeled as a unimodal gaussian density

$$P(x/\Omega) = \frac{\exp[-\frac{1}{2}(x-\bar{x})^T \Sigma^{-1}(x-\bar{x})]}{(2\pi)^{\frac{N}{2}}|\Sigma|^{\frac{1}{2}}}$$

where the mean and the covariance matrices are estimated using a training data set. In our case, frontal view photographs of different people.

The Mahalanobis distance $d(x) = (x-\bar{x})^T \Sigma^{-1}(x-\bar{x})$ is a sufficient statistic for characterizing the likelihood. Using the eigenvector and eigenvalue decomposition of Σ it is possible to derive a tractable estimate of this distance. The estimate involves just the first M principal coefficients -those of the projection of the input pattern x over the first principal components- and a residual reconstruction error,

$$\hat{d}(x) = \sum_{i=1}^{M} \frac{y_i^2}{\lambda_i} + \frac{1}{\rho}\epsilon^2(x)$$

where λ_i are the M principal eigenvalues, ρ is the average of the unused eigenvalues, and

$$\epsilon^2(x) = \sum_{i=M+1}^{N} y_i^2 = \|x - \bar{x}\|^2 - \sum_{i=1}^{M} y_i^2.$$

With this estimate, the detection problem can be formulated in a maximum likelihood framework. For each point in the input image, the distance between the class and a rectangular subimage centered in that point is calculated. The point with the lowest distance -the maximum likelihood- gives the center of the subimage that contains the object. In order to detect faces of different sizes the search is performed on linearly scaled versions of the input image.

Given this subimage and its partition in homogeneous regions (either $P_g(t)$ or $P_m(t)$), the next step consists in deciding which of these regions belong to a face and which ones are part of the background. The criterion for the merging step is that the union of two regions that belong to the same object should be more similar to the whole object (a face) than only one region.

The subimage and its partition are warped in order to normalize them to the eigenvectors size. The process starts from the region with minimum distance to the face class. The distance between a region and the class is calculated by placing the region on a new subimage with the database background behind and projecting it over the principal components. Next, the distance between the class and the union of this first region with each neighboring region is calculated. The region that produces the greatest decrease in distance is merged to the first one.

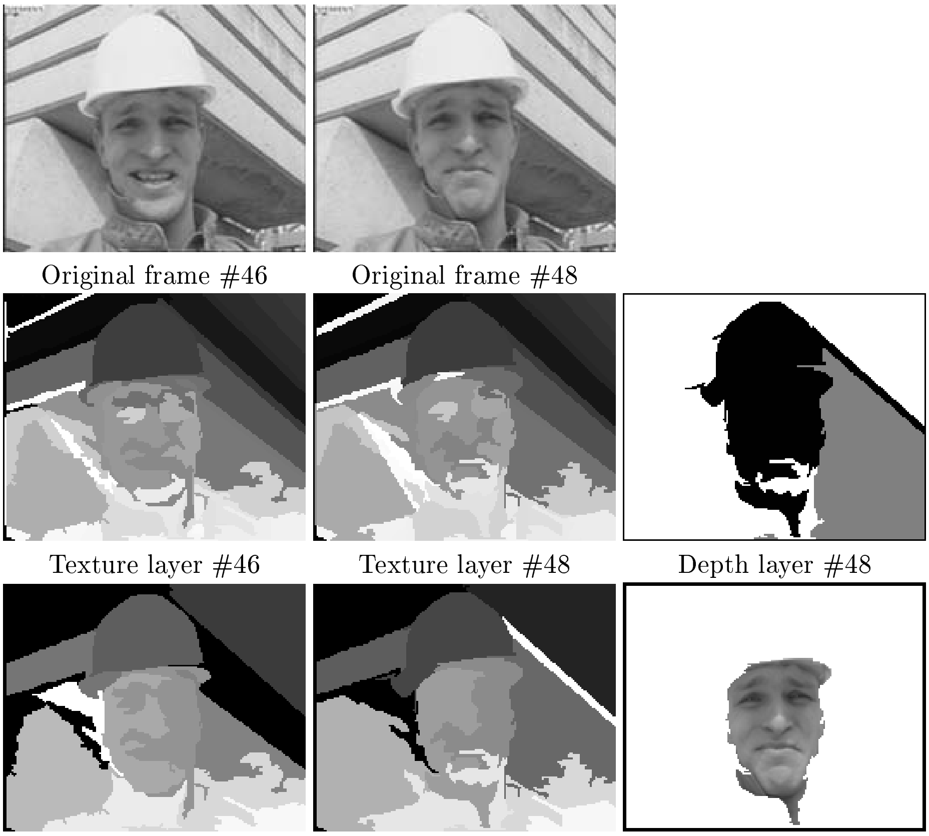
The process continues, in an iterative way, merging a new region if it is adjacent to any of the previous merged ones and if it produces the greatest decrease in distance, until no more regions can be merged; that is, no further union decreases the distance.

7 RESULTS

In Figure 3, frame #48 of the *Foreman* sequence is analyzed. This analysis is based on the results obtained for frame #46 and, therefore, $P_g(46)$ and $P_m(46)$ are presented. For frame #48, the four analysis layers are shown: $P_g(48)$ and $P_m(48)$ contain 50 and 12 regions, respectively. On turn, $P_d(48)$ has detected three different regions, where the region represented with a gray level value of 128 has not been assigned to any depth level.

References

1. J. L. Dugelay and H. Sanson. Differential methods for the identification of 2D and 3D motion models in image sequences. *Signal Processing, Image Communication*, 7:105–127, 1995.
2. B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *IEEE Proceedings of the Fifth International Conference on Computer Vsion*, Cambridge, U.S.A., June 1995.



Original frame #46 Original frame #48

Texture layer #46 Texture layer #48 Depth layer #48

Motion layer #46 Motion layer #48 Semantic layer #48

Fig. 3. Results of the complete hierarchical scheme

3. A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on systems, man and cybernetics*, 6:420–433, 1976.
4. P. Salembier, F. Marqués, and A. Gasull. Coding of partition sequences. In L. Torres and M. Kunt, editors, *Video Coding: The second generation approach*, pages 125–170. Kluwer Academic Publishers, 1996.
5. P. Salembier and J. Serra. Flat zones filtering, connected operators and filters by reconstruction. *IEEE Transactions on Image Processing*, 3(8):1153–1159, August 1995.
6. H. Sanson. Toward a robust parametric identification of motion on regions of arbitrary shape by non-linear optimization. In *Proceedings of IEEE Internatioanl Conference on Image Processing, ICIP'95*, volume I, pages 203–206, October 1995.