Spatiotemporal MRF Approach to Video Segmentation: Application to Motion Detection and Lip Segmentation.

F. Luthon, A. Caplier, M. Liévin
Laboratoire des Images et des Signaux
Institut National Polytechnique de Grenoble
LIS, INPG, 46 avenue Félix-Viallet
38031 Grenoble Cedex, France
Tel: +33 (0)4 76 57 43 72 Fax: +33 (0)4 76 57 47 90
Email: Franck.Luthon@inpg.fr

Signal Processing, 76(1):61-80, July 1999

Abstract

In this paper, a spatiotemporal strategy for image sequence analysis is proposed: a video sequence is processed as a 3-D data batch instead of a series of 2-D images.

Applying this approach to motion detection, a 3-D Markovian model associated with a spatiotemporal relaxation is defined. Using a 3-D neighbourhood of pixels for modelling spatiotemporal interactions, robust results are obtained for detecting moving objects in noisy sequences or in the case of overlapping motion.

In order to improve the performance to detect poorly-textured objects or very slow motion, the algorithm is integrated in a spatiotemporal multiresolution scheme. The data pyramid is built by using 3-D low-pass filtering and 3-D subsampling. Robust results for synthetic and real-world outdoor image sequences are reported.

This approach is also applied successfully to speaker's lip segmentation in image sequences, for audiovisual telecommunication.

Key words: motion detection, image sequences, Markov Random Field (MRF), spatiotemporal approach, multiresolution, lip segmentation.

Résumé

Cet article présente une approche spatio-temporelle pour l'analyse de séquences d'images[1] : une séquence est traitée comme un flot de données à trois dimensions au lieu d'une succession d'images à deux dimensions.

L'utilisation de cette approche pour la détection de mouvement conduit à la définition d'un modèle markovien 3-D associé à une relaxation spatio-temporelle. Grâce à une modélisation fine des interactions spatio-temporelles entre les pixels d'un voisinage cubique, des résultats robustes sont obtenus pour la détection d'objets mobiles dans une scène très bruitée et d'objets dont le mouvement s'effectue avec recouvrement d'une image à la suivante.

Dans le but d'améliorer l'aptitude de l'algorithme à détecter des objets très peu texturés et des objets de mouvement très lent, on définit un cadre de multirésolution spatio-temporelle. La pyramide de données est construite par une succession de filtrages et de sous-échantillonnages appliqués dans chacune des trois dimensions. L'intérêt de la multirésolution spatio-temporelle est mis en évidence par divers résultats de détection de mouvement sur des scènes synthétiques et réelles.

Une autre application de cette approche porte sur la segmentation des lèvres d'un locuteur, dans un contexte de télécommunications audio-visuelles.

[1] A paper in French is also available [5].

1 Introduction

Motion detection and region-based segmentation are important issues in image sequence analysis or coding, with applications in video-surveillance and video-communication.

Although three dimensions (x, y, t) are required to describe an image sequence, most of the methods dealing with sequence analysis are time sequential (each image is processed in turn), and work on a pair of consecutive images. This might induce limitations *e.g.* for detecting subpixel motion[2]. A common way to integrate motion information over a larger temporal domain is to use recursive temporal filtering such as Kalman filtering.

In this paper, another strategy is proposed. The point is to consider a video sequence not as an image series, but as a 3-D data batch, taking into account spatial and temporal dimensions within a single process. This approach is coherent with the fact that a moving object covers a volume in the (x, y, t) space.

The scope of the paper is twofold: to give an insight into the pros and cons of the spatiotemporal approach, together with focusing on practical applications. The performance of this approach is indeed illustrated with two applications: robust motion detection and lip segmentation in video sequences.

As for robust motion detection, a 3-D non-separable Markov Random Field (MRF) based algorithm is defined. This method yields better results than the separable version of the same algorithm in the case of noisy sequences or overlapping motion[3]. The same observations (temporal variations of the intensity function) as in the separable case are retained, the enhanced performance of the 3-D algorithm coming from the improvement of the MRF model which is better at taking temporal constraints into account.

To detect subpixel motion and uniform moving objects[4], a larger spatiotemporal domain must be taken into account. This is done by computing observations on a spatiotemporal pyramid.

In section 2, a separable motion detection algorithm is presented. The algorithm is inspired by the work of Bouthémy *et al.* [3]. There are two major differences between the algorithm described in [3] and the one presented here. The first difference is the way temporal information is dealt with. In [3], the processing of each image is done in two steps ("two-pass algorithm"). An initial detection of moving areas at time t is derived when considering images $I(t-1)$ and $I(t)$. This detection is updated when considering images $I(t)$ and $I(t+1)$. Two successive label fields are always simultaneously considered (optimization in two passes), and the decision about uncovered areas is postponed to the next processing pass. In section 2, we propose a "one-pass algorithm": a single label field (the current one) is optimized at each time (and only once). It makes implementation easier, for an equivalent quality of results. This is made possible thanks to another way of doing initialisation: we use a coarse estimate of the future label field, instead of repeting the past as is done in [3]. Uncovered areas are handled by giving more weight to the future than to the past (anisotropy in temporal interactions).

The second difference concerns computational complexity: we use four model parameters (section 2.4), instead of five in [3], since the function expressing the link between observations and labels is simpler in our case. The decision about a temporal clique (past or future) requires only one conditional test to choose among two configurations, while eight different configurations are tested in Bouthémy's algorithm (Table 1 in [3]). The number of 2-D fields required for the relaxation is five in our case (Fig. 1-b), instead of six for Bouthémy's algorithm (Fig. 2 in [3]). Hence, the amount of memory required for data storage is minor in our case. Therefore, the two-step algorithm proposed in [3] is less adequate for real-time implementation (*i.e.* processing at video rate).

Since real-time processing is of major concern for practical video applications, the paper addresses on several occasions the issues of computation cost and hardware implementation, either on general purpose programmable devices (digital signal processors (DSPs) or video processors), parallel machines (SIMD or MIMD) or dedicated circuits (ASICs, VLSI cellular analog networks).

[2] Subpixel motion means displacements of less than one pixel between two images (*i.e.* slow motion).

[3] Overlapping motion means that the intersection of the masks of a moving object at times $t-1$ and t is not empty.

[4] Uniform moving objects means moving objects that are poorly-textured, *i.e.* have uniform intensity.

The algorithm which is presented in section 2 is called *3-D separable motion detection algorithm* in the sense that space and time have distinct roles in the processing (hereafter, the algorithm is referred to as the "separable algorithm").

Its 3-D non separable counterpart is described in section 3. A comparison between the performance of both algorithms is made. In section 4, it is shown how the integration of the 3-D algorithm in a spatiotemporal multiresolution framework allows subpixel motion and poorly-textured moving objects to be detected. In section 5, another application of this approach is presented, for speaker's lip segmentation in a context of audiovisual telecommunication. A discussion in section 6 concludes the paper.

2 Separable MRF Model

MRF modelling is widely used for motion analysis, either for detection, estimation, or segmentation. For a state of the art about image motion analysis and an extensive bibliography, the reader may refer to [13].

2.1 Observations and Labels

The purpose of motion detection is to localize moving and static areas in a dynamic scene. It is a binary labelling problem that consists in attributing to each pixel or *site* $s = (x, y)$ of image S at time t one of the two *labels*: $l_s = a$ if s belongs to a moving area, $l_s = b$ if s belongs to the static background.

With the assumptions of quasi-constant illumination (very small lighting variations between $t-1$ and t) and static camera, motion information is closely related to temporal changes of the intensity function $I_s(t)$. Therefore, *observations* are given by:

$$o_s = |I_s(t) - I_s(t-1)|. \tag{1}$$

The following notation is used: $l = \{l_s, s \in S\}$ and $o = \{o_s, s \in S\}$ represent one particular realisation at time t of the label and observation fields L and O, respectively[5].

Given a realisation o of field O, the aim is to find the most probable configuration l of field L. This is done by using the Maximum A Posteriori criterion (MAP). From Bayes theorem and the equivalence between MRF and Gibbs distribution, it is known that the maximisation of the *a posteriori* probability is equivalent to the minimisation of an energy function [9]:

$$\max_l P(L = l \,|\, O = o) \iff \min_l U(l, o). \tag{2}$$

2.2 Energy Functions

The energy function is classically the sum of two terms (corresponding to prior knowledge and data-link, respectively):

$$U(l, o) = U_m(l) + U_a(o, l). \tag{3}$$

The model energy $U_m(l)$ is a regularisation term. It puts a priori constraints (spatiotemporal homogeneity) on the masks of moving objects, erasing isolated points due to noise. Its expression is given by:

$$U_m(l) = \sum_{c \in C} V_c(l_s, l_n) \tag{4}$$

where c denotes any of the binary cliques defined in the neighbourhood of Fig. 1-a. A binary clique

[5] Every time the instant considered is different from the current time t, a temporal index will be added in the notation.

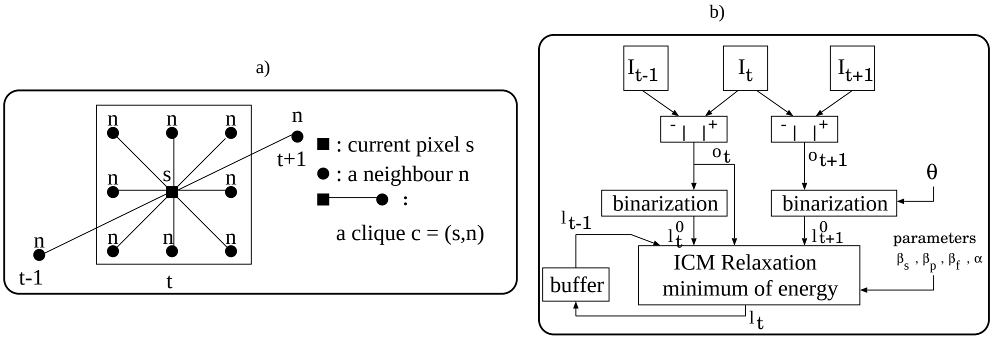

Figure 1: a) Neighbourhood and binary cliques. b) Separable algorithm block diagram (l^0 denotes a coarse estimate or initialisation of label field L).

$c = (s, n)$ is any pair of distinct sites in the neighbourhood, including the current pixel s and any of the neighbours n. C is the set of all cliques. $V_c(l_s, l_n)$ is an elementary potential function associated with each clique $c = (s, n)$. In order to put homogeneity constraints into the model, it is defined as:

$$V_c(l_s, l_n) = \begin{cases} -\beta & \text{if } l_s = l_n \\ +\beta & \text{if } l_s \neq l_n \end{cases} \tag{5}$$

where the positive parameter β depends on the nature of the clique: a parameter β_s is defined for spatial cliques, a parameter β_p for past temporal clique and a parameter β_f for future temporal clique.

The link between labels and observations is expressed by the relationship: $o_s = \Psi(l_s) + g_s$ where g is a Gaussian uncorrelated centered noise with variance σ^2 and:

$$\Psi(l_s) = \begin{cases} 0 & \text{if } l_s = b \\ \alpha > 0 & \text{otherwise.} \end{cases} \tag{6}$$

Ψ models the observations: if a pixel is static, no temporal change occurs in the intensity function and the observation should be zero; if a pixel is mobile, a change occurs and the observation is supposed to take a positive value close to α, which represents the average value of non-zero observations.

The link-to-data energy $U_a(o, l)$ (attachment energy) is derived from the above function:

$$U_a(o, l) = \frac{1}{2\sigma^2} \sum_{s \in S} [o_s - \Psi(l_s)]^2 \tag{7}$$

where the observation variance σ^2 is evaluated on-line for each image.

2.3 Spatial Deterministic Relaxation

Fig. 1-b shows the block diagram of the separable algorithm. The algorithm works on three consecutive frames. Suppose the past label field l_{t-1} has been determined as the result of the previous optimization. The current label field is initialised with a binary map l_t^0 derived from observation field o_t, and a coarse estimate l_{t+1}^0 of the future label field is also derived from binarisation of field o_{t+1}. The binary maps are obtained with the likelihood method proposed in [10], but could also be computed with a simple thresholding method, for computation savings purpose.

To find the minimum of the energy function, the deterministic relaxation algorithm ICM (Iterated Conditional Modes) is used [2]. For each pixel s of the current image, the two labels a and b are tested and the label which induces the minimum local energy in the neighbourhood is kept. The process iterates over the image until convergence, one iteration corresponding to the scanning in x and y dimensions of the image at time t. The stopping criterion for convergence of the relaxation is based on the relative decrease of the global energy function: $\Delta U(l, o) \ / \ U(l, o) = 0.01\%$. Then, the next image of the sequence is processed.

Note that, since the algorithm works with three frames, label fields are obtained with a delay of one frame.

2.4 Parameter Setting

The separable algorithm depends on five parameters: four parameters for MRF modelling $(\beta_s, \beta_f, \beta_p, \alpha)$, plus one threshold parameter θ for binarisation of observations. From various experiments both on real-world and synthetic image sequences, the model parameters are fixed to the following values: $\beta_s = 20$, $\beta_p = 10$, $\beta_f = 30$, $\alpha = 10$. This manual learning phase for parameter tuning was based on empirical observations: contextual homogeneity of detected masks, good agreement between contours of masks and actual moving objects, and insensitivity to acquisition noise. Unsupervised estimation methods, like Expectation-Maximisation [7], could also be used to estimate model parameters $\beta_s, \beta_f, \beta_p$. But they are prohibitive in terms of computation cost. Morevover high precision in the determination of these values is not required (robustness of MRF method insensitive to a slight change of these values). Parameter β_s controls spatial homogeneity and may be decreased in case of very noisy sequences. Parameters β_f and β_p control temporal homogeneity. More weight is given to the future by taking $\beta_f > \beta_p$, so that the background area which has been uncovered during motion is faster eliminated. Indeed, in such a region, the past temporal neighbour is a-labelled while the future one is b-labelled. But the good label is the static one ($l_s = b$), given by the future information. Note that temporal homogeneity constraint can be relaxed in case of fast motion.

Parameter α stands for some kind of average value of non-zero observations. This parameter may either be computed on-line for each image as explained in [3], or fixed to an arbitrary value before processing. From experimental tests, on-line computation of α for each image does not significantly improve motion detection results.

The threshold θ required for binarisation (computation of initial binary maps with a method derived from [10]) is the only parameter which must be adjusted for each sequence. Here, it is determined manually (off-line learning phase at the beginning of video acquisition or before running the automatic processing). One could use likelihood tests such as described in [10, 1] to determine this decision threshold automatically, but at the expense of computation cost. A too low value of θ induces many false detections. A too high value of θ erases moving pixels in overlapping motion areas. For all sequences acquired with the same camera under the same lighting conditions, the same value of θ may be kept (*e.g.* $\theta = 32$ for all street sequences presented in this paper).

2.5 Computational Complexity

The processing rate is evaluated in the case of images of size 128 × 128. When implemented on a Sparc-10 workstation with C programming, the processing of an image takes about $1.8s$ of cpu time ($\approx 0.4s$ per iteration). This corresponds roughly to $N_0 \times N_x \times N_y \times N_i = 2.5\ 10^7$ elementary operations. $N_0 = 400$ is the number of elementary operations (additions, multiplications, conditional tests) involved in the computation of the local energy associated with each pixel (1 multiplication = 10 additions). $N_x = 128$ and $N_y = 128$ represent the image dimensions and $N_i = 4$ is the average number of iterations until convergence.

To achieve real-time processing, various hardware implementations (on parallel SIMD machine, DSP board, or cellular VLSI analog network) have been either developped or simulated [6, 8]. A processing rate of 12 to 25 frames per second is then achieved. Another implementation on a Programmable Video Processor (PVP) for telecommunication applications is now under study. The PVP is an intensive computing unit with a parallel SIMD architecture (8 sub-processors connected to a shared memory of 16 Kbytes) and seven I/O ports for data flow circulation. It offers a high computing power (2 Gops) with a high I/O rate (4 Gbits/s). For images of size 256 × 256 and a clock frequency of 70 MHz, a processing rate of 150 frames/s is obtained when implementing the algorithm on the PVP software simulator.

2.6 Experimental Results

The separable algorithm was tested both on synthetic and real-world image sequences. A typical example for video-surveillance application (traffic control) is shown in Fig. 2. This street sequence, acquired with a standard video camera, contains a single pedestrian walking on the pavement. The image sequence is not very noisy and motion of the pedestrian is large enough between two images, allowing a good detection. The mask of the moving body detected in the image plane is given at four consecutive instants.

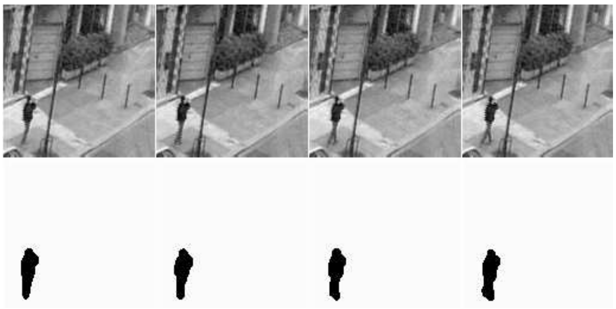

Figure 2: Top) Street sequence with a moving pedestrian; Bottom) Masks of the moving body detected after relaxation (black = moving label, white = static label).

3 3-D Non Separable MRF Model

3.1 Spatiotemporal Relaxation

Although the separable algorithm integrates motion information from three consecutive frames, only the current frame is processed at each time (Fig. 1-b). The 3-D non separable model for motion detection is based on the intuitive idea that, by taking into account more than three consecutive frames of the sequence, the analysis of motion may be improved. Therefore, the video sequence is no longer considered as an image series but as a 3-D data batch. L and O are now 3-D random fields (or volumes).

To find the minimum of the energy function, a spatiotemporal version of ICM is required. The key point is that, at each iteration, the relaxation runs over temporal sections of length N_t (Fig. 3). The scanning is done not only in spatial dimensions (x, y) at a given time t, but in the three dimensions (x, y, t) together. It is performed back-and-forth spatially *and* temporally. One iteration corresponds to the scanning of a whole temporal section. All frames of the temporal section are processed together. After convergence of ICM, labels of all pixels included in that section are available.

All along the paper, we refer to the 3-D non-separable motion detection algorithm as "the 3-D algorithm".

3.2 A Priori Model

The mathematical framework of MRF modelling remains the same. The relationships of section 2 still hold, since there is no restriction about the dimensions of fields L and O. However, fields L and O are now supposed to be spatiotemporal 3-D random fields, bringing about the following changes: in Eq. (7), S represents now a temporal section of N_t images, instead of a single image. The neighbourhood structure associated with L is now a complete spatiotemporal cube (Fig. 4), and

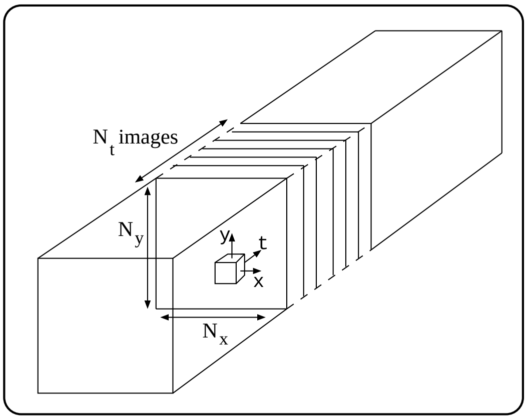

Figure 3: Temporal section of length N_t.

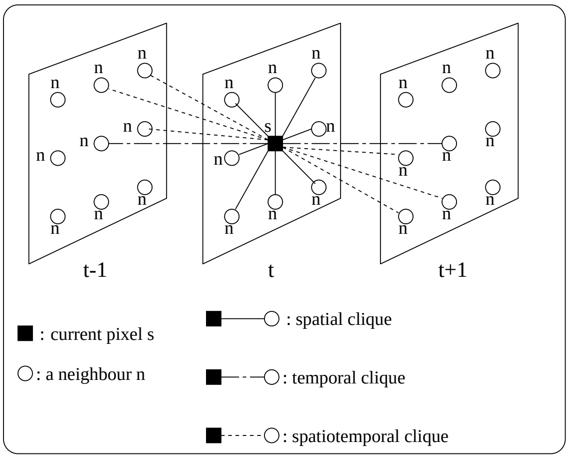

Figure 4: 3-D neighbourhood and binary cliques.

clique parameters (β in Eq. (5)) have to be redefined as functions of sites: $\beta(s,n)$. Moreover, a weight parameter λ is added in the global energy function for balancing $U_m(l)$ and $U_a(o,l)$ influences in this extended neighbourhood:

$$U(l,o) = U_m(l) + \lambda \, U_a(o,l). \tag{8}$$

We suppose that the proposed neighbourhood contains all the dependencies of pixel s. This is the simplest 3-D neighbourhood. One could increase in space and time the size of the neighbourhood, but at the expense of computation cost. In this spatiotemporal neighbourhood, three kinds of binary cliques are defined: spatial, temporal and spatiotemporal (Fig. 5). They differ according to their spatial and temporal extent (along the x, y and t axis, respectively). Let δ_x, δ_y, δ_t represent in the 3-D space (x,y,t) the coordinates of vector $\overrightarrow{(s,n)}$ corresponding to a clique with origin in the current pixel s ($\delta \in \{-1;0;1\}$). Then we get: eight purely spatial cliques (horizontal ($\delta_x = \pm 1$, $\delta_y = 0$, $\delta_t = 0$), or vertical ($\delta_x = 0$, $\delta_y = \pm 1$, $\delta_t = 0$), or diagonal ($\delta_x = \pm 1$, $\delta_y = \pm 1$, $\delta_t = 0$)); two purely temporal cliques ($\delta_x = 0$, $\delta_y = 0$, $\delta_t = \pm 1$); sixteen spatiotemporal cliques (($\delta_x = \pm 1$, $\delta_y = 0$, $\delta_t = \pm 1$) or ($\delta_x = 0$, $\delta_y = \pm 1$, $\delta_t = \pm 1$) or ($\delta_x = \pm 1$, $\delta_y = \pm 1$, $\delta_t = \pm 1$)).

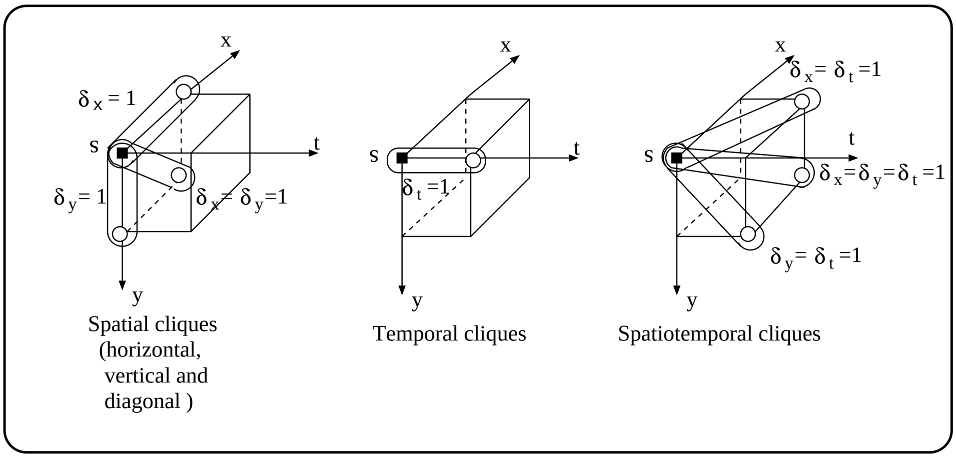

Figure 5: The three types of binary cliques.

For the definition of clique potentials in Eq. (5), a spatial parameter β_s is used to control spatial homogeneity (no distinction is made between x and y) and a temporal parameter β_t for homogeneity in temporal dimension. This is a simple way to take into account the non-homogeneity between space and time. Note that no more distinction is made between past and future, since the 3-D algorithm will propagate information forward and backward in time and allow to change a decision taken in the past, especially as regards uncovered areas, thanks to the *spatiotemporal* nature of iterations (see comments in section 3.5).

All clique potentials are defined with these two parameters, according to the physical principle that interaction with the current pixel gets weaker when the neighbour is far. Here, interaction is assumed to be inversely proportional to the squared distance between sites in the cube. Thus, the actual potential $\beta(s,n)$ associated with a clique $c = (s,n)$ is defined by the following expression:

$$\beta(s,n) = \frac{1}{d^2(s,n) \, \left[\frac{\delta_x^2(s,n)}{\beta_s} + \frac{\delta_y^2(s,n)}{\beta_s} + \frac{\delta_t^2(s,n)}{\beta_t} \right]} \tag{9}$$

where $d(s,n) = \sqrt{\delta_x^2 + \delta_y^2 + \delta_t^2}$ is the Euclidian distance between the current pixel s and the considered neighbour n. This relationship gives:

- $\beta(s,n) = \beta_s$ for spatial horizontal or vertical cliques ($d(s,n) = 1$);

- $\beta(s,n) = \frac{\beta_s}{4}$ for spatial diagonal cliques ($d(s,n) = \sqrt{2}$);
- $\beta(s,n) = \beta_t$ for temporal cliques ($d(s,n) = 1$);
- $\beta(s,n) = \frac{\beta_s \beta_t}{2(\beta_s+\beta_t)}$ for spatiotemporal horizontal or vertical cliques ($d(s,n) = \sqrt{2}$);
- $\beta(s,n) = \frac{\beta_s \beta_t}{3(\beta_s+2\beta_t)}$ for spatiotemporal diagonal cliques ($d(s,n) = \sqrt{3}$).

3.3 Parameter Setting

Four model parameters are required: $\beta_s = 20$, $\beta_t = 5$, $\alpha = 15$ and $\lambda = 5$. These values were determined experimentally (as in the separable model). We choose in practice $\beta_s > \beta_t$ to give more importance to spatial homogeneity which is supposed to be more reliable than temporal homogeneity (especially true in the case of non-deformable objects undergoing arbitrary motion).

Parameter λ controls the influence of both terms of energy. If it is necessary to reinforce a priori constraints (because of bad observations for example), λ should be decreased. If it is necessary to reinforce the link to data, λ should be increased.

The specification of neighbourhood and clique potentials entirely defines the MRF model, so that actual values of N_t, N_x or N_y do not influence the modelling. Different values of N_t were tested. The default value is $N_t = 8$. It may be decreased when spatiotemporal homogeneity constraint is broken (fast motion) and it may be increased for very noisy sequences. Still, at least $N_t = 5$ images per section are required because of temporal boundary effects (first and last images of a section are not processed because of the lack of past and future neighbours, respectively).

3.4 Computational Complexity

At first sight, the computational complexity of the 3-D algorithm may be a bottleneck. In practice, handling a video sequence as a 3-D data batch does not drastically increase the global computation time compared to a serial processing image per image. On a Sparc-10 workstation with C-programming, $4s$ of cpu-time per image of size 128×128 are necessary to detect motion. The increase of computation cost comes primarily from the increased number of iterations required until convergence (10 iterations on average instead of 4), the stopping criterion remaining the same as in section 2.3. The neighbourhood extension (26 neighbours instead of 10) does not induce a major extra computing charge.

On the other hand, the delay required before obtaining motion detection results may be crucial. Since the 3-D algorithm runs on temporal sections of length N_t, all motion masks of a section are available at the same time, when the processing of the whole section is completed. In order to limit both the delay and the required memory for software implementation, N_t should be small (anyway much lower than the actual length of any video sequence).

Therefore, a long sequence should be processed recursively, by cutting it into smaller temporal sections. Fig. 6 illustrates the recursive process with $N_t = 5$. The 3-D algorithm runs in space and time on the first section of five images: images $t-2$, $t-1$ and t are processed together (section 1); then it runs on the second section of five images: images $t-1$, t and $t+1$ are processed, with initial label fields l^0_{t-1}, l^0_t given by results of section 1, etc...

Every time, one new image is stored and only 5 successive frames stay in memory. When image $t+3$ is acquired, the final result for image t may be computed, corresponding to a delay of 120ms ($3 \times 40ms$ for sequences acquired at 25 images per second), which might be acceptable in many applications (*e.g.* video-surveillance).

The recursive process does not increase computational complexity. Of course, each label field is estimated in three consecutive temporal sections. For example, l_t is processed when estimating (l_{t-2}, l_{t-1}, l_t) (section 1), (l_{t-1}, l_t, l_{t+1}) (section 2), and (l_t, l_{t+1}, l_{t+2}) (section 3). But as regards the

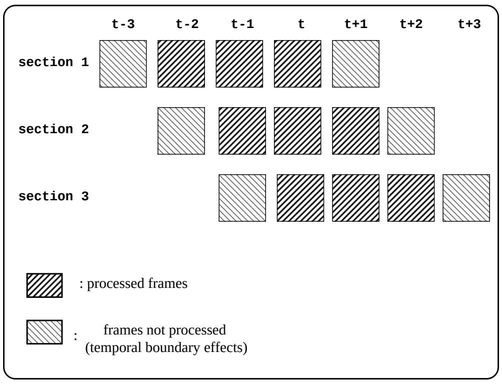

Figure 6: Section-recursive algorithm ($N_t = 5$)

two last estimations (temporal sections 2 and 3), the initial label field l_t^0 is more reliable (close to the final one), so that convergence is faster (fewer iterations are needed).

This recursive version of the algorithm was implemented on a SIMD machine [6]. The parallel machine is a linear network of 256 elementary processors with 4Kbytes of local memory each. It communicates with a host workstation via Ethernet interface. Assembler or C-parallel programming can be used. Local computations are done in parallel. Data are uniformly distributed among processors. The processing rate achieved is around 3 to 4 images/second (images of size 128 × 128).

3.5 Experimental Results

Fig. 7 illustrates the efficiency of the 3-D algorithm to recover moving objects in a noisy sequence. The synthetic sequence contains two moving objects: a clear rectangle which translates rightward (1 pixel/image) and a dark square which translates leftward (1 pixel/image). Shown are the binary masks detected with the separable and the 3-D algorithms, respectively. One can see that spatiotemporal relaxation is useful to eliminate bad detection due to noise (isolated points).

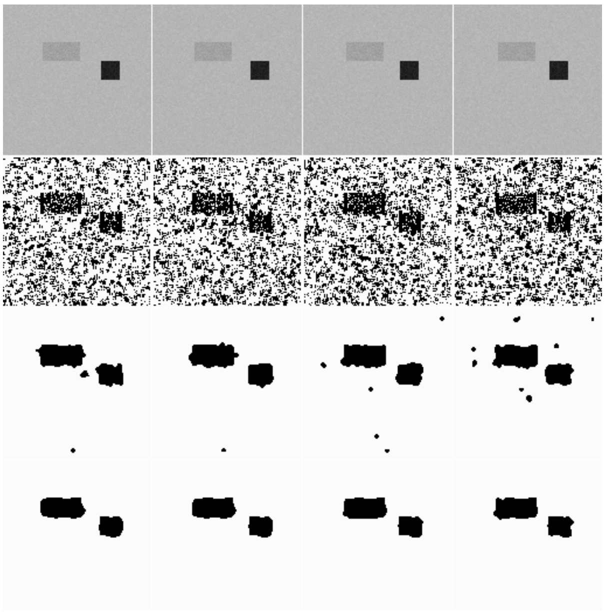

Figure 7: From top to bottom: 1) Synthetic sequence with impulse noise; 2) Initial binary maps ($\theta = 20$); 3) Masks detected after *spatial* relaxation (separable algorithm); 4) Masks detected after *spatiotemporal* relaxation (3-D algorithm, $N_t = 8$).

The 3-D algorithm is also more effective in case of overlapping motion. Indeed, information is propagated both in space and time. The sequence of Fig. 8 contains two moving areas: a group of three pedestrians walking on the pavement and a bicycle riding leftward on the road. With the separable algorithm, the pedestrians mask is only partially recovered because of a lack of information in the overlapping motion area. The separable algorithm implies causal processing and does not allow to back-propagate spatiotemporal constraints in time and to change a decision taken in the past. The 3-D algorithm, in contrary, makes it possible to back-propagate information in time and to fully recover the pedestrians mask for each image of the sequence. In the bottom of Fig. 8, the precision of the masks in terms of contours is shown: the upper mask corresponds to the group of pedestrians, while the lower mask corresponds to the bicycle.

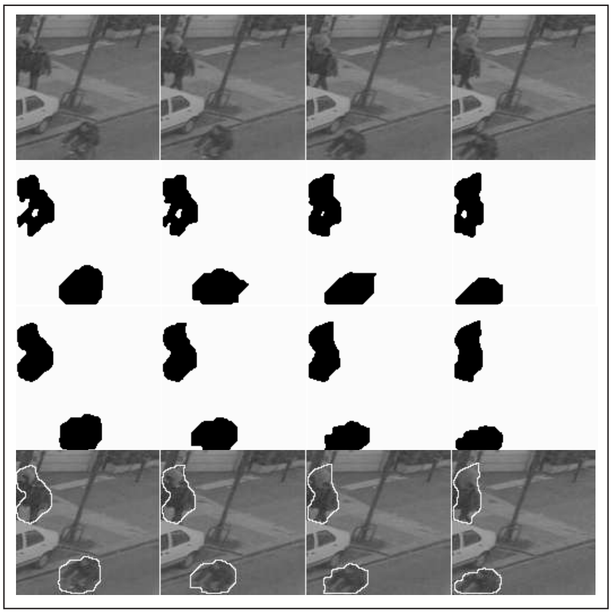

Figure 8: From top to bottom: 1) Street sequence; 2) Masks detected with *separable algorithm*; 3) Masks detected with *3-D algorithm*; 4) Contours of masks obtained with the 3-D algorithm, superimposed on the image sequence.

4 Spatiotemporal Multiresolution Framework

Both versions of the algorithm (separable and 3-D) yield poor results in case of uniform intensity moving areas or subpixel motion. In such cases, although objects are moving, temporal variations of the intensity function are almost zero (observations of poor quality). To solve this problem, the 3-D algorithm is run on a *spatiotemporal pyramid* of data with a coarse-to-fine strategy. Spatial filtering is a common way to deal with large uniform intensity moving areas. Temporal filtering is effective in order to deal with subpixel motion.

Multiresolution may also improve the initialisation step for spatiotemporal relaxation. Indeed it has been conjectured that multiresolution analysis smoothes the energy function [11], making it easier to find the global minimum. This may be crucial when a deterministic relaxation algorithm like ICM is used, since it may get stuck in the first encountered local minimum of the energy function in case of bad initialisation.

4.1 Spatiotemporal Low-Pass Pyramid

The spatiotemporal structure of the 3-D MRF model suggests to build not only a spatial but a *spatiotemporal* pyramid. The basic convolution kernel is the binomial low-pass filter $\frac{1}{4}[1\ 2\ 1]$ which is applied in the three dimensions x, y and t. This gives the 3-D convolution kernel of Fig. 9-a. Inspired by Burt's spatial pyramid [4], this kernel, associated with a spatiotemporal subsampling, is used to build a spatiotemporal pyramid (Fig. 9-b). Note that frames after filtering and before subsampling are less corrupted by noise than frames after filtering and subsampling. An example of spatiotemporal pyramid with three resolution levels is shown in Fig. 10. Spatiotemporal subsampling reduces the size of each image by a factor of 4 and the length of the sequence by a factor of 2 at each resolution level.

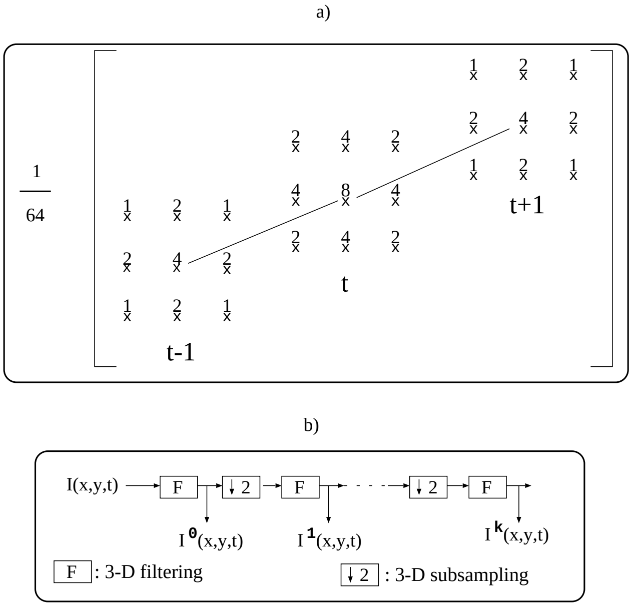

Figure 9: a) Spatiotemporal filter convolution kernel. b) Pyramid building process (the superscript k denotes the resolution level).

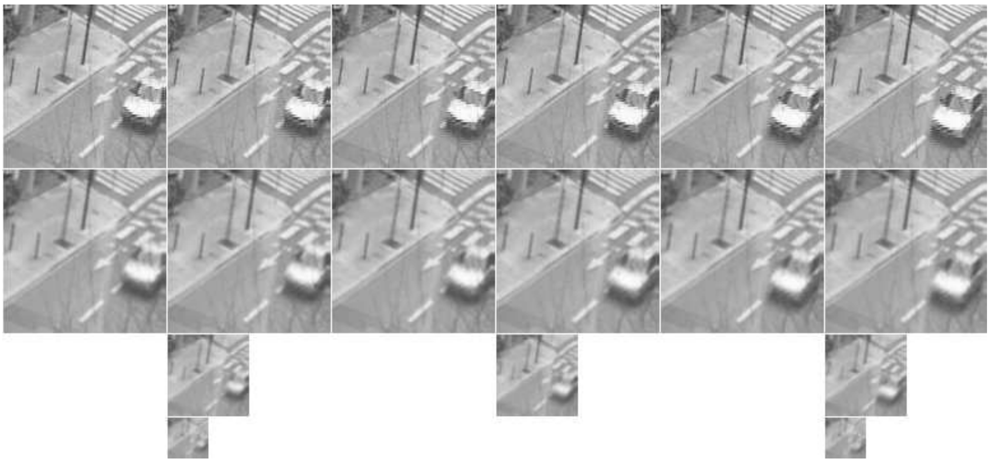

Figure 10: Spatiotemporal pyramid: original sequence in top row, and the three spatiotemporal levels below ($k = 0, 1, 2$).

The 3-D algorithm is run at each level of the spatiotemporal low-pass pyramid. The strategy is coarse-to-fine: the algorithm starts at the lowest resolution level (k_{max}). After spatiotemporal interpolation (Fig. 11), the result of relaxation at level k is used to initialise relaxation at level $k-1$. Running the algorithm on this pyramid gives a multiresolution label field. At each level, the label field is optimised according to observations at the corresponding level in the spatiotemporal pyramid

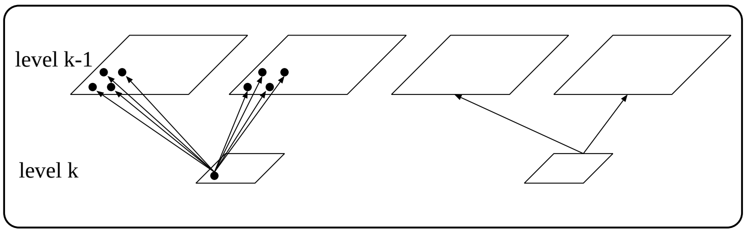

Figure 11: Spatiotemporal interpolation

(Fig. 12).

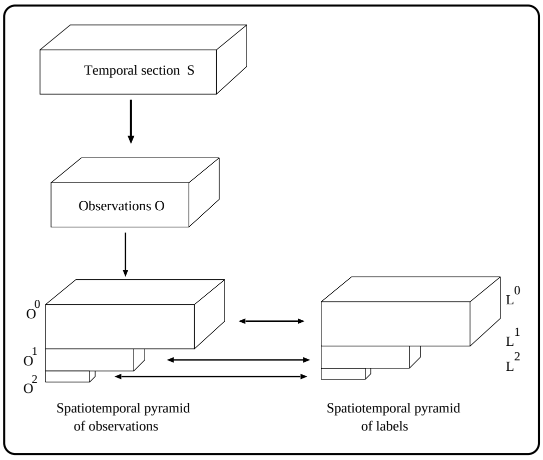
Temporal section S
Observations O
O^0
O^1
O^2
L^0
L^1
L^2
Spatiotemporal pyramid of observations
Spatiotemporal pyramid of labels

Figure 12: Spatiotemporal multiresolution scheme.

The optimal number of levels for the pyramid (k_{max}) depends on the size and speed of moving objects: to detect very slow motion, k_{max} should be increased. If the scene contains very small moving objects, k_{max} should be decreased. The default value is $k_{max} = 2$ (three resolution levels).

4.2 Pyramid of Observations

The spatiotemporal filtering integrates motion information over a larger spatial and temporal domain, so that observations at low resolution levels are more relevant in case of subpixel motion and uniform moving areas. Spatial filtering improves observations for poorly-textured areas, while temporal filtering on many consecutive frames improves observations in case of subpixel motion.

Fig. 13 exhibits the quality of observations, computed both with mono- and multiresolution schemes, for the well-known *Trevor* sequence. This sequence represents the motion of a TV speaker. Motion between two consecutive frames is very slow (subpixel motion) and many areas of the shirt, head and hands are poorly-textured. The figure presents the multiresolution observations at three resolution levels. The darker the pixel, the larger the corresponding observation. Multiresolution observations are clearly more consistant than monoresolution observations in that case.

4.3 Parameter Adaptation

Parameters of the algorithm are adapted along the pyramid as explained below.

First, the evolution of observations along the pyramid has been investigated in order to adapt parameter α at each resolution level. Of course, low-pass filtering reduces the amplitude of observations. Let us focus on a pixel s at a motion transition, as shown in Fig. 14 (vertical edge moving rightward at a speed of 1 pixel/frame). After spatiotemporal filtering, the amplitude of observation is divided by a factor $\frac{8}{3} \simeq 3$ (obvious computation with the 3-D convolution kernel of Fig. 9-a). Therefore

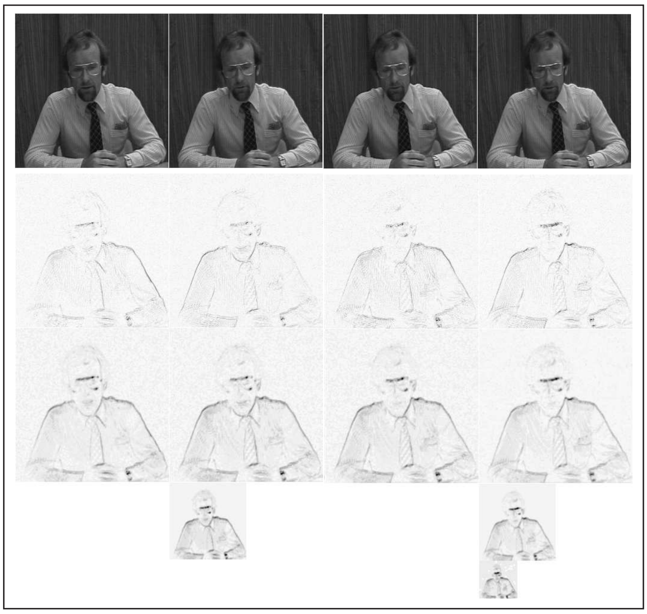

Figure 13: From top to bottom: 1) Four consecutive images of *Trevor* Sequence; 2) Monoresolution observations; 3) Multiresolution observations, 3 resolution levels ($k = 0, 1, 2$). All displays are normalised in order to span over the full available dynamic range $[0; 255]$.

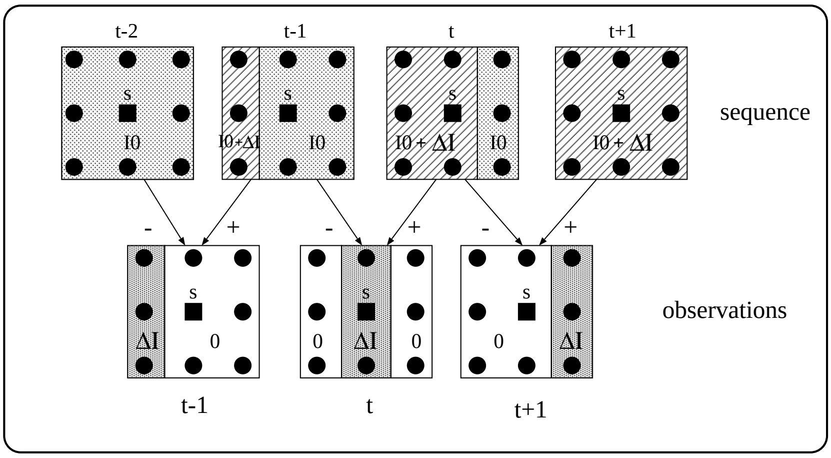

Figure 14: Evolution of observation after spatiotemporal filtering: typical case of a vertical edge moving rightward. $I_s(t-1) = I_0$, $I_s(t) = I_0 + \Delta I$ and $I_s(t+1) = I_0 + \Delta I$. ΔI represents the amplitude of step observed at pixel s. *Before 3-D spatiotemporal filtering*: $o_s = |I_s(t) - I_s(t-1)| = |\Delta I|$. *After spatiotemporal filtering*: $o_s = \frac{1}{64}(4|\Delta I| + 16|\Delta I| + 4|\Delta I|) = \frac{3}{8}|\Delta I|$.

parameter α, which stands for the average value of non-zero observations, has to be reduced in the same proportion along the pyramid, *i.e.* $\alpha_k = \alpha_0/3^k$. Since observations decrease by a factor of about 3, the observation variance σ^2 decreases by a factor of about 9, so that the ratio in Eq. (7) remains constant.

Secondly, since spatial and temporal information are integrated in the same way along the pyramid, the parameter ratio β_s/β_t is kept constant for all resolution levels.

Thirdly, from a qualitative point of view, spatiotemporal interactions should get weaker at low resolution levels, since two neighbouring pixels are actually far away in the full-resolution image sequence. The evolution of potentials $\beta(s,n)$ should be related to the physical distance between pixels in a square grid. This can also be stated from a quantitative point of view: at each resolution level, the physical distance $d(s,n)$ between two pixels is actually doubled because of subsampling. This leads to a decrease of 4 for clique potentials $\beta(s,n)$ in Eq. (9). For computational simplicity, this evolution law is simply implemented by adapting the weight factor λ_k as follows: $\lambda_k = 4^k\ \lambda_0$. The global energy is then: $U(l,o) = U_m(l) + \lambda_k\ U_a(o,l)$.

Finally, parameter θ does not need to be adapted along the pyramid, since the binarisation method derived from [10] (and hence parameter θ) is only used for label initialisation at the lowest resolution level k_{max}. At finer resolution levels, initialisation is simply performed by interpolating the results of lower resolution levels, with no need of θ. But compared to the monoresolution scheme, θ must be increased when multiresolution is used (experimental observation). The theoretical explanation of the necessary increase of θ with multiresolution level is the influence of data low-pass filtering on the method given in [10] for setting the threshold value.

4.4 Computational Complexity

The building of the pyramid is not computationally expensive: since the 3-D convolution kernel of Fig. 9-a is separable, the implementation of the spatiotemporal filtering is equivalent to the implementation of three 1-D binomial filters in x, y and t dimensions, respectively.

The relaxation at low resolution levels is quick due to the smaller number of sites and therefore Markovian constraints are propagated faster. Compared with the full-resolution level $k = 0$, the data flow to be processed at level $k = 1$ is reduced by a factor of 8 (N_x, N_y, N_t decrease each by a factor of 2). Thus, one iteration at resolution level k is equivalent (in terms of computation cost) to $1/2^{3k}$ iterations at the finest resolution level ($k = 0$).

Then, at higher (finer) resolution levels, fewer iterations are needed compared to a monoresolution scheme, because of a better initialisation propagated from lower resolution levels. So, multiresolution usually reduces the overall number of iterations.

The computation time has been recorded experimentally for many sequences. The same stopping criterion as in section 2.3 was used. In fact, the multiresolution spatiotemporal algorithm does not drastically speed up the processing rate. So the main interest of the multiresolution framework here is the improved performance for detecting subpixel motion and poorly-textured moving areas as shown in next section, but not computation savings.

4.5 Experimental Results

Fig. 15 presents the masks detected in a case of subpixel motion with both versions (mono- and multiresolution) of the 3-D algorithm. The synthetic scene contains three mobile objects: a clear rectangle moving rightward (1 pixel/frame), a dark square moving leftward (1 pixel/frame) and another square on the left moving slowly upwards (0.35 pixel/frame). With the 3-D *monoresolution* algorithm, the slowest square is badly detected. With the *multiresolution* version of the algorithm, this square is well detected, starting from the second resolution level.

Fig. 16 presents the masks detected for the *Trevor* sequence with both versions (mono- and multiresolution) of the 3-D algorithm. Monoresolution masks are very fragmented, since speaker's

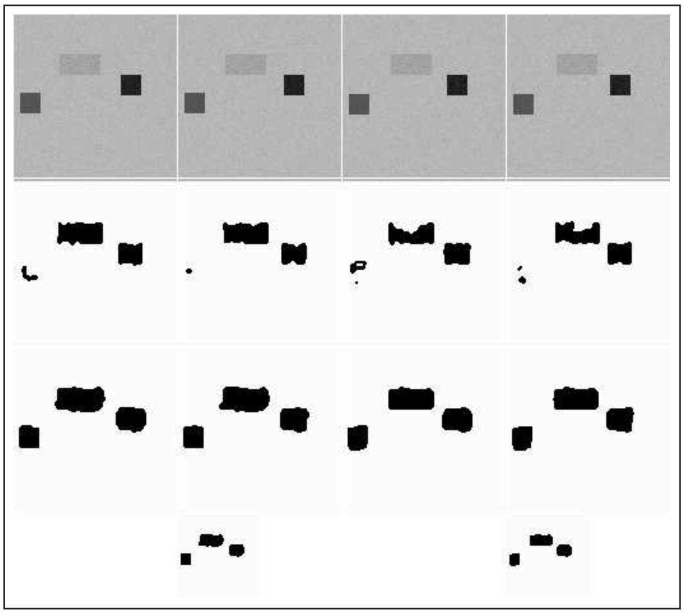

Figure 15: From top to bottom: 1) Synthetic sequence with the lower-left dark square undergoing subpixel motion; 2) Monoresolution masks; 3) Multiresolution masks (2 levels : $k = 0, 1$).

motion is very slow and the scene contains many poorly-textured areas (hands, shirt, head). On the contrary, multiresolution masks are spatially and temporally homogeneous. The whole body is fully detected, starting from the third resolution level ($k = 2$).

5 Lip Segmentation

The proposed approach was also applied to lip segmentation in color image sequences, for audiovisual communication between two speakers. Fig. 17 shows the context of application for a high quality and low bit rate videophone. It can also be used for man-machine communication (automatic speech recognition) or videoconferencing.

The speaker wears a light helmet equipped with a micro-camera and a microphone, so that the camera is fixed with respect to the head. The segmentation is based on the assumption that lips are areas in the face were red hue and motion predominate.

The main steps of the processing are as follows (details may be found in [12]). First, a color video sequence of speaker's face is acquired under natural lighting conditions and without any particular make-up. A logarithmic color transform is performed from RGB (red, green, blue) to HIS (hue, intensity, saturation) color space, in order to gain independence from illumination brightness and noise.

Then, two observations are derived. The first observation is computed from the hue value at each pixel: it gives information about areas where red hue is most prominent. The second observation is the same as in Eq. (1): frame differences between two consecutive images. It gives information about motion areas.

From these two thresholded observations, four initial labels (a_0, a_1, b_0, b_1) are derived, for coding four pixel classes: pixels with ($_1$), respectively without ($_0$) motion, belonging (a), respectively not belonging (b), to red hue areas.

The spatiotemporal MRF approach is then used for regularizing the solution. Some changes were introduced in the model presented in section 3.2, in order to take into account better the *a priori* knowledge available for this specific application (lip shape and motion). Namely, the spatiotemporal potential function $\beta(s, n)$ is now inversely proportional to the Euclidian distance (and not the squared

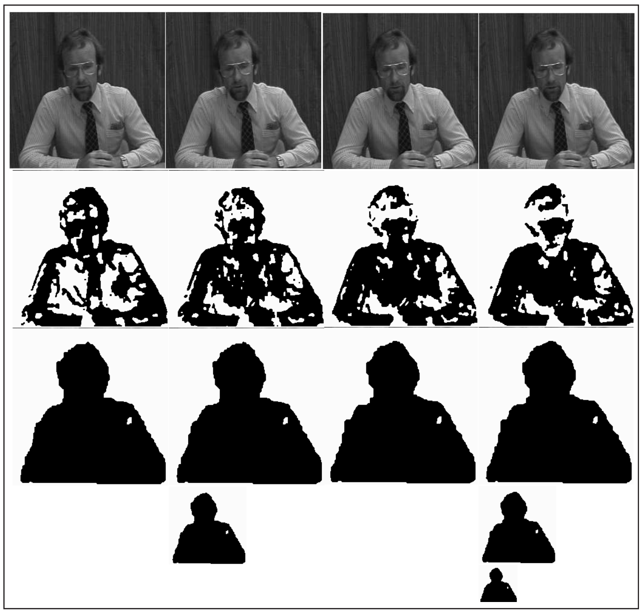

Figure 16: From top to bottom: 1) Four images of *Trevor* sequence; 2) Monoresolution masks; 3) Multiresolution masks (3 levels : $k = 0, 1, 2$).

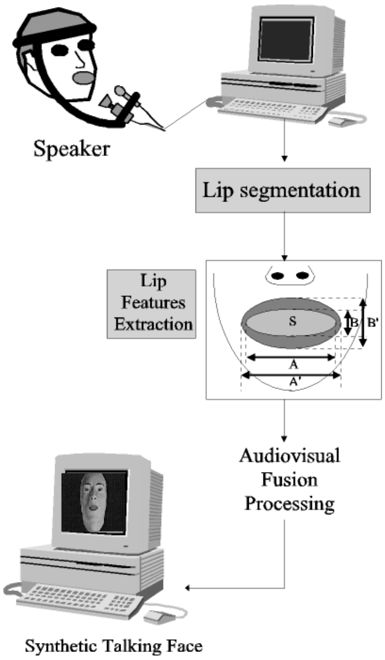

Figure 17: Context of audiovisual communication: from the image sequence of speaker's face, geometrical features of lips are extracted and provide modelling parameters for talking face synthesis and animation.

distance) between two neighbours. As in section 3.2, the distance integrates two elementary potentials β_s and β_t as scaling factors. But for this application, we force some spatial anisotropy: $\beta_x = 2.\beta_y = \beta_s$ in order to put emphasize on horizontal configurations (geometrical constraints on lip shape). This yields:

$$\beta(s,n) = \frac{1}{\sqrt{\left(\frac{\delta_x}{\beta_x}\right)^2 + \left(\frac{\delta_y}{\beta_y}\right)^2 + \left(\frac{\delta_t}{\beta_t}\right)^2}} = \frac{\beta_s \beta_t}{\sqrt{\beta_t^2 \left(\delta_x^2 + 4\delta_y^2\right) + \beta_s^2 \delta_t^2}}. \qquad (10)$$

Moreover, in contrary to section 3.2, parameters β_s and β_t are not constant, but depend on the labels taken by sites s and n. They are defined to constrain the model to, respectively, spatial homogeneity of labels, and temporal homogeneity of hue when no motion is detected. For example, $\beta_s(l_s, l_n)$ is proportional to: $|r(s) - r(n)| + |m(s) - m(n)|$, where $r(s)$ and $m(s)$ are binary digits (0 or 1) coding the presence at pixel s of red hue and motion, respectively. For the definition of $\beta_t(l_s, l_n)$, see Table 3 in [12].

With this modelling, one obtains robust label fields after relaxation, exhibiting areas in the face where red hue and motion are predominant (Fig. 18).

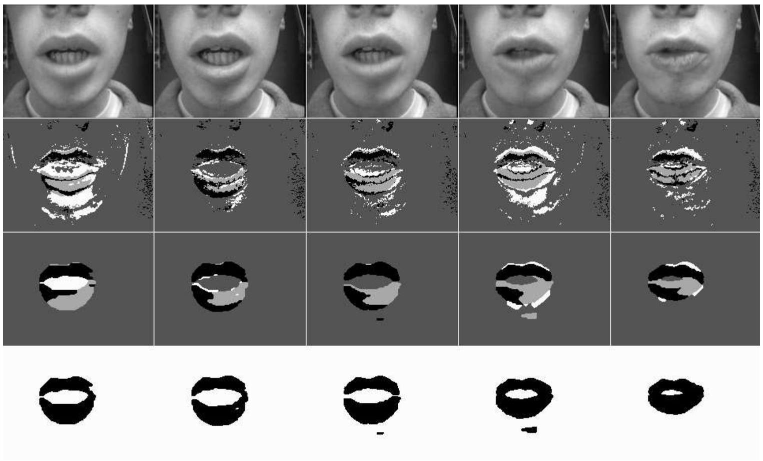

Figure 18: *From top to bottom*: 1) Sequence of luminance images: male face without make-up; 2) Initial label fields; 3) Final label fields after relaxation: *the four labels are shown in gray levels (from white to black:* b_1, a_1, b_0, a_0*)*; 4) Sequence of lip masks (combination of a_0 and a_1).

From the final label field, a region of interest is determined automatically (mouth bounding box in Fig. 19). Measurements of geometrical features are performed on lip masks (height and width, surface), and used for face synthesis at the receiver's end.

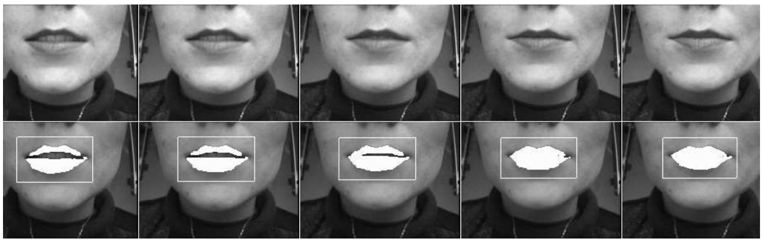

Figure 19: *Top)* Sequence of luminance images: female face with soft red make-up; *Bottom)* Sequence of lip masks with bounding box superimposed on the luminance.

The proposed method for lip segmentation solves two crucial problems that usually arise in such a context: indeed, the processing gains independence both from lighting conditions and make-up of lips. This is due both to the use of the logarithmic color transform, and to the robust spatiotemporal

MRF model which is effective for detecting the elusive contours of lips and recovering homogeneous lip areas.

A parallel implementation of this algorithm on a Programmable Video Processor is under study. The achievable processing rate is estimated to be 13 images/s for images of size 256 × 256.

6 Discussion

A spatiotemporal strategy for image sequence analysis was presented, and applied successfully to motion detection and lip segmentation in a Markovian framework. It primarily consists in processing a video sequence as a 3-D data batch.

With such an approach, improved performance is reported for motion detection in case of noisy sequences and in case of overlapping motion.

A 3-D spatiotemporal multiresolution scheme coherent with the 3-D MRF is also proposed. This multiresolution approach is efficient to handle two difficult cases: subpixel motion and poorly-textured moving areas. But in case of very fast motion, the multiresolution algorithm yields worse results than the monoresolution version. This is due to the fact that temporal filtering induces an averaging of motion information over many images, so that it is no longer possible to precisely detect motion boundaries. As a result, motion masks are bigger than actual moving objects. Spatial multiresolution without temporal multiresolution would be beneficial in that case, since it allows to spatially linearize intensity without temporal blurring. Mono- and multiresolution algorithms being complementary, it would be interesting to develop a strategy for switching automatically between both versions of the algorithm according to the analysed sequence. Moreover, the multiresolution pyramid involves 3-D low-pass filtering. In order to limit the blurring effect, the use of 3-D wavelets (3-D orthogonal high-pass and low-pass filter banks) could be considered.

The second application reported here concerns speaker's lip segmentation in a color video sequence. The interest of the spatiotemporal method, together with a logarithmic color transform, is supported by the good quality of results obtained in this challenging situation (natural images of speaker's face without any particular make-up or lighting).

The spatiotemporal approach has also been used to compute spatiotemporal gradients with spline functions (results not reported here). The implementation involves 3-D recursive filterings. Thus, we do believe it could also be applied with success to optical flow estimation.

References

[1] T. Aach, A. Kaup, R. Mester, "Statistical model-based change detection in moving video", Signal Processing, Vol. 31, No. 2, March 1993, pp. 165-180.

[2] J. Besag, "On the Statistical Analysis of Dirty Pictures", Journal of Royal Statistical Society, Vol. B-48, No. 3, 1986, pp. 259-302.

[3] P. Bouthémy, P. Lalande, "Recovery of moving object masks in an image sequence using local spatiotemporal contextual information", Optical Engineering, Vol. 32, No. 6, June 1993, pp. 1205-1212.

[4] P.J. Burt, E.H. Adelson, "The Laplacian Pyramid as Compact Image Code", IEEE Trans. on Communications, Vol. 31, No. 4, 1984, pp. 532-540.

[5] A. Caplier, F. Luthon, "Approche spatio-temporelle pour l'analyse de séquences d'images. Application en détection de mouvement", Traitement du Signal, Vol. 14, No. 2, 1997, pp. 195-208.

[6] A. Caplier, F. Luthon, C. Dumontier, "Real-Time Implementations of an MRF-based Motion Detection Algorithm", Real-Time Imaging, Vol. 4, No. 1, February 1998, pp. 41-54.

[7] M. Dempster, N. Laird, D. Rubin, "Maximum Likelihood from incomplete data via the EM algorithm", Journal of the Royal Statistical Society, Series B, Vol. 39, 1977, pp. 1-38.

[8] C. Dumontier, F. Luthon, J.P. Charras, "Real-Time Implementation of an MRF-based Motion Detection Algorithm on a DSP Board", Proc. 7^{th} IEEE Digital Signal Processing Workshop, Loen, Norway, September 1996, pp. 183-186.

[9] S. Geman, D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", IEEE Trans. Pattern Anal. and Machine Intel., Vol. 6, No. 6, November 1984, pp. 721-741.

[10] Y. Z. Hsu, H. H. Nagel, G. Reckers, "New likelihood test methods for change detection in image sequences", Comput. Vision Graph. Image Process., Vol. 26, 1984, pp. 73-106.

[11] J. Konrad, E. Dubois, "Multigrid Estimation of Image Motion Fields using Stochastic Relaxation", Proc. 2^{nd} IEEE Int. Conf. on Computer Vision, Tarpon Springs, Florida, December 1988, pp. 354-362.

[12] M. Liévin, F. Luthon, "Lip features automatic extraction", Proc. 5^{th} IEEE Int. Conf. on Image Processing, Chicago, Illinois, October 1998, Vol. 3, pp. 168-172.

[13] A. Mitiche, P. Bouthémy, "Computation and analysis of Image Motion: A Synopsis of Current Problems and Methods", International Journal of Computer Vision, Vol. 19, No. 1, 1996, pp. 29-55.