Spatiotemporal Video Modelling for Content Summarization

Nuno Vasconcelos Andrew Lippman
MIT Media Laboratory
20 Ames St, E15-320M, Cambridge, MA 02139
{nuno,lip}@media.mit.edu

Abstract

Still have to do this.

1 Introduction

Given the ubiquity of bandwith, connectivity and computational resources associated with modern communications networks, massive repositories of pictorial information start to appear throughout them. The usefulness of such repositories will be, to a significant extent, determined by the availability of systems which can help users navigate through them, and interact with or manipulate their content.

In the case of video databases, the magnitude of stored information is by itself an overwhelming problem as on-line analysis of each frame in the video stream becomes impractical, even if this analysis consists only of very simple operations. There is, therefore, a need to develop procedures for the automatic summarization of video content which can then be used to speed up browsing and retrieval operations. Of particular interest are methods capable of providing *visual* sumarization of the video streams, as these summaries can be directly inspected by human users of the video repository.

Due to this interest and the fact that visual summarization of video sequences has application in a wide range of other domains, a significant body of research has been devoted to this topic in the recent past. The fundamental idea is to compute a single image map which is representative of the pictorial content of the video sequence, by warping all the frames contained in it into a reference coordinate frame and somehow combining their pixel intensities. Because different solutions to the problem have evolved in different research communities, with different applications in mind, the resulting representations have received diverse names. Among these are *salient stills* [19, **?**], *video mosaics* [6, 16, 11, 18, 12], *video sprites* [13, **?**], and *video layers* [20, 5][1].

In spite of this diversity, all these procedures are similar in the sense that they follow the following two fundamental steps.

1. Fitting a global motion model to the motion between each pair of successive frames
2. Computing the summarizing image map by accumulating the information from all the frames after they have been aligned according to the motion estimates computed in the previous step.

Several variations have been proposed with respect to motion estimation. While most rely on simple affine modelling, some have considered more sophisticated options - including planar-perspective models [16, 13, **?**], fully-perspective models [12], and models accounting for motion paralax [15, 11, 16] - or issues such as handling outliers [16, 14] or achieving more accurate estimates by tracking [8, 9, 10]. The second step usually consists of averaging the registered images.

In this paper we show that relying on temporaly localized motion estimates limits the capacity of these representations with ragards to the task of producing a image map capable of providing a visually meaningfull summarization of the video content. This problem is a direct consequence of the fact that representations based on temporally localized motion models cannot capture the global characteristics of the video stream along the temporal dimension. While the intensity map on which they rely contains visual information sumarizing the entire sequence and the parametric motion description is valid over the entire spatial extent of any given frame, the underlying motion models account only for a highly localized temporal neighberhood (usually a frame pair) of the spatiotemporal volume spanned by the sequence. Therefore, they provide no guarantee of coherence along the temporal dimension, allowing motion estimates to oscilate

[1] While, strictly speaking, layering allways includes the construction of multiple image maps, the construction of each of these maps can be implemented, once the scene is segmented into the objects of interest, by procedures similar to that presented in this paper. For this reason we refer to the image map originated by content summarization as layer in the remainder of the paper

between competing scene interpretations which lead to poor image registration.

In order to achieve temporal coherence, we introduce, in this work, a trully global representation in both the spatial and temporal dimensions. For this, we augment the motion model with a generic temporal constraint which guarantees smoothness and avoids behaviour such as that of Figure 2. The resulting motion model is parametric in both space and time and can be fitted to the *entire* sequence at once, with marginal increase in computational complexity. Because the model encompasses both space and time and is fitted to the whole sequence, it locks to the motion which is dominant over the *entire* spatiotemporal volume, guaranteeing temporal coherence and leading to a significantly better summarization of the video content by a summarizing image.

We would also like to note that, while the focus of this paper is on video summarization and the benefits of temporal coherence, the advantages of a parametric spatiotemporal motion representation are not limited to this domain. For example, the fact that a compact description of the dominant motion throughout the entire sequence is available, also makes the representation atractive for purposes of content-based retrieval. Unlike those based on temporally localized motion estimates, our representation originates a single layer and a single spatiotemporal parameter vector which are guaranteed to follow the dominant motion in the sequence, if there is one. This not only implies that retrieval will be more accurate but also that it can be based on either the layer, the motion, or both. Motion based retrieval is dificult when motion is characterized by a large set of temporally localized estimates. We are currently investigating the use of spatiotemporal modeling for this and other applications, and will report on them in the future.

This paper is organised as follows. Section 2 discusses the *dominant* motion assumption inherent to registration-based approaches for content summarization, and how the use of temporally localized motion estimates can undermine the validity of this assumption. Section 3 then presents the spatiotemporal video model which is at the core of our representation. The details of estimating the model parameters from video data are discussed in section 4.1. Finally, section 5 presents summarization results and ilustrates the advantages of relying on spatiotemporal modelling instead of the previous temporally localized approaches.

2 Content summarization by image registration

The main assumption underlying procedures for video summarization through image registration is that there is a *dominant* motion among the motions of the various objects in the scene. If there is a single motion (e.g. a static scene and a moving camera) then (assuming the motion model matches the true scene motion) the sumarization is perfect. If more than one motion is present, the object with the dominant motion is correctly aligned and the remaining objects are "blurred out". The result is a summarizing layer where the dominant object appears crisp and the remaining objects are substituted by ghostly versions that provide a sense for the action in the scene (see for example Figure 7).

One of the main limitations of the dominant motion assumption is that it is not allways straightforward to determine what motion will be dominant. To ilustrate this point, consider a sequence of a bird flying in a region of uniformly blue sky. Because the sky has no texture and, therefore, any motion will be a good fit for the sky region, the dominant motion will be that of the bird. If, however, the sky is textured (e.g. it contains clouds or stars) or there is also a tree in the background, the motion of the bird will no longer dominate. In practice, which motion is dominant depends on the relative sizes of the objects, how they are textured, the relative amplitudes of their velocities, and the oclusion relationships originated as they move.

The problem is that all these factors change as the sequence progresses and the dominant motion may not be dominant at all instants. This is ilustrated by the simple example of Figure 1 which displays three snapshots of a sequence composed by two squares of symilar texture but different sizes, translating at the same speed in opposite directions. When there is overlap, the smaller square (B) occludes the larger one (A).

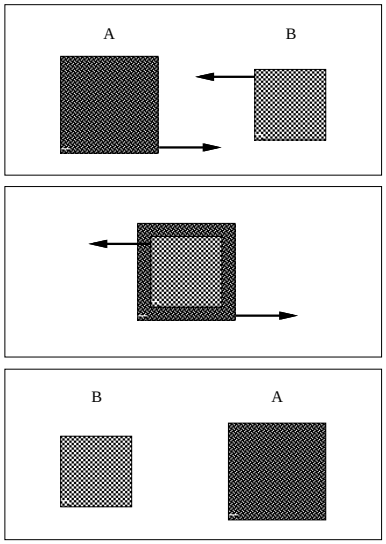

Figure 1. Three snapshots of a sequence where temporaly localized motion estimates fail to identify the dominant motion.

Since all other factors are equal, the dominant motion is

that of the square with the largest number of visible pixels, and A will dominate for most of the sequence. However, in the period where B occludes A (depicted by the center snapshot in the figure), there may be several frame-pairs for which B has the largest number of visible pixels and, therefore, dominates. Hence, as shown in Figure 2, the estimate of the dominant motion will switch between the two possibilities as the sequence progresses.

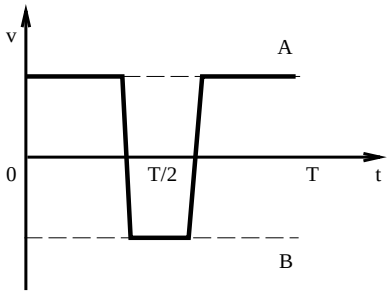

Figure 2. Velocities of each of the objects in the sequence of Figure 1 as a function of time. The dashed lines indicates the paths, in velocity space, of each of the objects in the figure. The solid line indicates the trajectory of the dominant motion, as computed by procedures based on temporaly localized motion models. The occlusion near $t = T/2$ leads to a switch regarding which motion is perceived as dominant.

The consequence is that neither of the two objects will be correctly aligned in the resulting layer, i.e. both will be blurred-out to at least some extent, and it will be much harder to perceive the scene dynamics from this layer than if the registration would have been performed with respect to one of the squares alone.

The importance of integrating motion estimates throughout the sequence has been realized by Irani and his coworkers in [8, 9, 10]. They propose a recursive procedure for building the layer on the fly where, for each frame, they compute the best affine motion estimate between the current layer estimate and that frame. The layer is then registered with the frame and updated by taking a weighted average of the two. The rationale is that, as the sequence progresses, the layer locks onto the object of dominant motion and the other objects are blurred out. This, in turn, reinforces the lock.

There are two reasons why such type of temporal integration will not work for the purpose of content summarization[2]. First, due to the very nature of the procedure, the resulting layer does not provide a characterization of the entire sequence, but simply of the last few frames contained in it. Second, it is not clear, that integration of motion estimates obtained from frame pairs will work for problems such as that of figure 1.

In the case of the figure, such a procedure would start by following A, and B would initially be wiped out of the layer. However, as soon as there were overlap between the two squares, some of B's texture would start to be included as well. By the time of the center snapshot in the figure, depending on the rate at which old information is discarded from the layer and the velocities of the two objects, the layer's texture would either resemble that of A, that of B or something in between. While in the first case everything would go well; in the latter two, B would, with high likelyhood, be tracked throughout the rest of the sequence, leading to a situation even worst than that of Figure 2.

Even though temporal integration is a correct step towards eliminating the uncertainty originated by several competing scene interpretations it does not completly address the dificulties created by the fact that different interpretions may become dominant at different time instants. This issue can only be addressed through representations capable of capturing dominance over the entire spatiotemporal volume spanned by the sequence. We next introduce a spatiotemporal motion model which leads to reprsentations whith such properties.

3 The spatiotemporal motion model

We start by assuming that the motion between consecutive frames in the video sequence can be characterized by an affine transformation, i.e.

$$d^x_{j,j+1} = c^1_j + c^2_j x_j + c^3_j y_j \quad (1)$$

$$d^y_{j,j+1} = c^4_j + c^5_k x_j + c^6_k y_j, \quad (2)$$

where j is the frame number, $\mathbf{x}_j = (x_j, y_j)^T$ are the image coordinates of pixel $\mathbf{x}$, and

$$\mathbf{d}_{j,j+1} = (d^x_{j,j+1}, d^y_{j,j+1})^T = (x_{j+1} - x_j, y_{j+1} - y_j)^T$$

is the displacement applied to the pixel from frame j to frame $j + 1$. However, in order to guarantee consistency of motion estimates across time, we augment the motion model by imposing a generic temporal constraint: *that each pixel follows a path along the sequence according to a smooth trajectory characterized by a (low-order) polynomial, i.e.*

$$\mathbf{x}_j = \mathbf{x}_0 + \sum_{i=0}^{M} \phi_i t^i_j, \quad (3)$$

where t_k is the time-stamp of frame k. The number $M + 1$ of terms of this polinomial provides a trade-off between the degree of smoothness of the approximation, and the capability of following the pixel's trajectory. If $M + 1 = N$, where

[2] We should note that their work was not aimed at recovering a layer for sumarization pursposes, but instead to obtain improved motion estimates and motion-based segmentations.

N is the number of frames in the sequence, the model can follow exactly any possible trajectory, but provides no extra constraint other than those already imposed by the affine model. On the other hand, if $M = 0$ the model forces the pixel to land in the same location at every frame, i.e. allows no motion. In our experience, a low-order polynomial provides a good compromise between these factors - we have used $M = 2$ in all the experiments reported in section 5. The framework is, however, generic and valid for any value of M.

In appendix A we show that, given the temporal smoothness assumption of equation (3), the motion between sucessive frames will be affine if and only if the polynomial coefficients ϕ_i are themselves the result of an affine transformation of $\mathbf{x}_0$, i.e.

$$\phi_i^x = \rho_i^1 + \rho_i^2 x_0 + \rho_i^3 y_0 \quad (4)$$
$$\phi_i^y = \rho_i^4 + \rho_i^5 x_0 + \rho_i^6 y_0. \quad (5)$$

Substituting these equations in equation (3) and grouping terms we obtain

$$d_{0,j}^x = \sum_{i=0}^{M} (\rho_i^1 t_j^i) + \sum_{i=0}^{M} (\rho_i^2 t_j^i) x_0 + \sum_{i=0}^{M} (\rho_i^3 t_j^i) y_0 \quad (6)$$

$$d_{0,j}^y = \sum_{i=0}^{M} (\rho_i^4 t_j^i) + \sum_{i=0}^{M} (\rho_i^5 t_j^i) x_0 + \sum_{i=0}^{M} (\rho_i^6 t_j^i) y_0 \quad (7)$$

i.e. the displacement of the pixel between frames 0 and j is the sum of $M + 1$ affine transformations with coefficients proportional to the $M + 1$ powers of t_j. Defining

$$\mathbf{\Phi}(\mathbf{x}_0) = \begin{bmatrix} 1 & x_0 & y_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_0 & y_0 \end{bmatrix}, \quad (8)$$

$$\mathcal{T}_j = \begin{bmatrix} t_j^M \mathbf{I}_6 & \dots & t_j \mathbf{I}_6 & \mathbf{I}_6 \end{bmatrix},$$

and

$$\mathbf{p} = (\mathbf{p}_M, \dots, \mathbf{p}_0)^T,$$

where $\mathbf{I}_6$ is the identity matrix of order six, and $\mathbf{p}_i = (\rho_i^1, \dots, \rho_i^6)^T, i = 0, \dots, M$, the spatiotemporal trajectory of the point can be writen in a compact form as

$$\mathbf{x}_j = \mathbf{x}_0 + \mathbf{\Phi}(\mathbf{x}_0)\mathcal{T}_j\mathbf{p} = \mathbf{\Psi}_j(\mathbf{x}_0). \quad (9)$$

4 Estimation of the model components

Given a video sequence $\mathcal{F}_1, \dots, \mathcal{F}_N$, we model each of the frames, $\mathcal{F}_j$, as the outcome of a Gaussian process with mean described by the affine warping of a image $\mathcal{S}$, temporally co-located with $\mathcal{F}_1$. From equation (9), and droping the subscript of $\mathbf{x}_0$,

$$P(\mathcal{F}_j(\mathbf{\Psi}_j(\mathbf{x}))|\mathbf{p}, \mathcal{S}(\mathbf{x})) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\mathcal{F}_j(\mathbf{\Psi}_j(\mathbf{X})) - \mathcal{S}(\mathbf{X}))^2}{2\sigma^2}}.$$

Assuming that each of the Gaussian variables is independent the joint density for all the pixels in the sequence is characterized by

$$P(\mathcal{F}_1, \dots, \mathcal{F}_N|\mathbf{p}, \mathcal{S}) \propto \exp\left\{ \sum_{j,\mathbf{x}} (\mathcal{F}_j(\mathbf{\Psi}_j(\mathbf{x})) - \mathcal{S}(\mathbf{x}))^2 \right\}.$$

In order to determine the parameters of the spatiotemporal motion model and the summarizing layer $\mathcal{S}$ which best explain the observed image data, we rely on a Maximum Likelihood (ML) framework, according to which the optimal motion parameters and summarizing layer are those which minimize the cost function

$$\mathcal{J}(\mathbf{p}, \mathcal{S}(\mathbf{x})) = \sum_{j,\mathbf{x}} (\mathcal{F}_j(\mathbf{x} + \mathbf{\Phi}(\mathbf{x})\mathcal{T}_j\mathbf{p}) - \mathcal{S}(\mathbf{x}))^2. \quad (10)$$

The minimization is performed by iterating between the estimation of the motion parameters given an estimate of the summarizing layer, and the updating of the layer given the new parameter values. Given an estimate for $\mathcal{S}$ the optimal new set of parameters $\mathbf{p}'$ is

$$\mathbf{p}' = \min_{\mathbf{p}} \mathcal{J}(\mathbf{p}, \mathcal{S}(\mathbf{x})) \quad (11)$$

and, given this new set of parameters, the updated estimate of $\mathcal{S}$ is, for each location of the layer,

$$\mathcal{S}'(\mathbf{x}) = \min_{\mathcal{S}(\mathbf{x})} \mathcal{J}(\mathbf{p}', \mathcal{S}(\mathbf{x})). \quad (12)$$

Because it is a quadratic function of $\mathcal{S}(\mathbf{x})$, equation (12) has a simple solution. On the other hand, the first problem (equation (11)) depends non-linearly on the motion parameters. Two diferent paths have been used in the past to solve problems of this type. One class of methods, usually referred to as *indirect*, relies on the computation of optic flow as an intermediate step towards the estimation of the parameters which characterize the parametric motion representation [1, 20]. Given the frames in the sequence, an estimate of the true motion in the scene is first obtained through a standard optical flow estimator [3], and the parametric motion model is then fitted to the estimated optical flow. The second class, usualy known as that of *direct* methods, bypasses this intermediate step, obtaining the motion parameters from the sequence of images itself.

Indirect methods have the advantage that, given the optic flow field, finding the set of parameters which provide the least squares fit to this field is a linear problem with a single global solution, which can therefore be easily solved. However, because they rely on the computation of the optic flow, such approaches are vulnerable to all the problems inherent to optical flow estimation (such as the apperture problem and high ambiguity in non-textured areas). Because they avoid such problems, direct methods have become more popular in the recent past [2]. Our solution to the problem of equation (11) falls in this category.

4.1 Estimating the motion parameters

In order to obtain the motion parameter vector which minimizes equation (11) we rely on the *Gauss-Newton* method [4] which, as shown in appendix B, leads to an iterative procedure of the form

$$\mathbf{p}^{k+1} = \mathbf{p}^k + \gamma^k \mathbf{d}^k, \tag{13}$$

where

$$\mathbf{d}^k = \left[\sum_j \mathcal{T}_j^T \alpha_j^k \mathcal{T}_j\right]^{-1} \sum_j \mathcal{T}_j \beta_j^k, \tag{14}$$

$$\alpha_j^k = \sum_{\mathbf{x}} \mathbf{\Phi}(\mathbf{x})^T \nabla_{\mathbf{x}} \mathcal{F}_j(\mathbf{\Psi}_j^k(\mathbf{x})) \nabla_{\mathbf{x}}^T \mathcal{F}_j(\mathbf{\Psi}_j^k(\mathbf{x})) \mathbf{\Phi}(\mathbf{x}), \tag{15}$$

$$\beta_j^k = \sum_{\mathbf{x}} [\mathcal{F}_j(\mathbf{\Psi}_j^k(\mathbf{x})) - \mathcal{S}(\mathbf{x})] \mathbf{\Phi}(\mathbf{x})^T \nabla_{\mathbf{x}} \mathcal{F}_j(\mathbf{\Psi}_j^k(\mathbf{x})), \tag{16}$$

$\mathcal{F}_j(\mathbf{\Psi}_j^k(\mathbf{x}))$ is the result of warping the j^{th} frame with the current estimate of the transformation associated with it ($\mathbf{\Psi}_j^k$), $\nabla_{\mathbf{x}}$ the gradient with respect to the image coordinates, and γ_k a scalar determined by a line-search.

The procedure for the estimation of the spatiotemporal motion parameters can therefore be summarized as follows.

1. Set $k = 0$. Compute an initial parameter estimate $\mathbf{p}^0$. Our initialization strategy is presented in section 4.3.
2. For each frame in the sequence, $\mathcal{F}_j, j = 1 \ldots N$:
   - warp the frame according to the current estimate of the motion parameters $\mathbf{p}^k$ and equation (9);
   - compute the spatial gradient of the warped frame, $\nabla_{\mathbf{x}} \mathcal{F}_j(\mathbf{\Psi}_j^k(\mathbf{x}))$;
   - compute α_j^k and β_j^k according to equations (15) and (16).
3. Compute $\mathbf{d}^k$.
4. Find γ^k by a line search. In our implementation, this is done by considering $\gamma_l^k = 2^{-l}, l = 0, \ldots, 4$, computing $\mathbf{p}_l^{k+1} = \mathbf{p}^k + \gamma_l^k \mathbf{d}^k$ for every l, and choosing the one which minimizes the cost function of equation (11).
5. If $||\mathbf{p}^{k+1} - \mathbf{p}^k|| < T$, where T is a pre-defined threshold, stop. Otherwise, set $k = k + 1$ and go to 2.

4.2 Updating the summarizing layer

Once the optimal motion parameters are determined, the estimate of the lyer $\mathcal{S}$ can be updated through the minimization of equation (12). It is straightforward to show that setting to zero the derivative, with respect to $\mathcal{S}(\mathbf{x})$, of this equation leads to

$$\mathcal{S}'(\mathbf{x}) = \frac{1}{N} \sum_j \mathcal{F}_j(\mathbf{x} + \mathbf{\Phi}(\mathbf{x}) \mathcal{T}_j \mathbf{p}'). \tag{17}$$

This has the intuitive appeal that once the optimal motion parameters are found, the optimal layer is simply the mean of all the images in the sequence after they are warped to the layer's coordinate frame. Equation 17 has, in fact, been used in one form or another on the majority of previous proposals for the the construction of image layers [20] and mosaics [16].

Given the new summarizing layer, a new set of motion parameters can be computed, leading to the iterative minimization of equations (11) and (12). Notice, that since each step in the iteration is guaranteed to decrease the cost function or leave it unchanged, and the cost function is bounded below by zero, the procedure is guaranteed to converge to a (possibly local) minimum.

4.3 Parameter initialization

As was noted in section 4.1, the Gauss-Newton method requires an initial estimate for the motion parameters. In fact, because at each iteration this method linearizes the cost function around the current parameter estimate, this initial guess should be in the basin of atraction of the desired optimum. It is therefore important to obtain an initial estimate which is close to the true optimum. In order to obtain such an estimate estimate, we start by computing the set of affine transformations between all pairs of successive frames in the sequence, and then find the spatiotemporal motion parameters by a simple least squares fit.

The estimation of the best affine fits to the motion between successive frames is based on a variation of the method proposed in [2]. For each $j = 1, \ldots, N-1$, we compute

$$\mathbf{q}_j = \min_{\mathbf{q}} \sum_{\mathbf{x}} (\mathcal{F}_{j+1}(\mathbf{x} + \mathbf{\Phi}(\mathbf{x})\mathbf{q}) - \mathcal{F}_j(\mathbf{x}))^2,$$

where $\mathbf{\Phi}(\mathbf{x})$ is given by equation (8) and $\mathbf{q}_j$ the 6-dimensional vector of parameters which characterizes the affine transformation between $\mathcal{F}_j$ and $\mathcal{F}_{j+1}$. This minimization is itself performed with the Gauss-Newton method, leading to an iterative procedure of the form

$$\mathbf{q}_j^{k+1} = \mathbf{q}_j^k + \gamma^k (\hat{\alpha}_j^k)^{-1} \hat{\beta}_j^k, \tag{18}$$

where

$$\hat{\alpha}_j^k = \sum_{\mathbf{x}} \mathbf{\Phi}(\mathbf{x})^T \nabla_{\mathbf{x}} \mathcal{F}_{j+1}(\psi_j^k(\mathbf{x})) \nabla_{\mathbf{x}}^T \mathcal{F}_{j+1}(\psi_j^k(\mathbf{x})) \mathbf{\Phi}(\mathbf{x}), \tag{19}$$

$$\hat{\beta}_j^k = \sum_{\mathbf{x}} [\mathcal{F}_{j+1}(\psi_j^k(\mathbf{x})) - \mathcal{F}_j(\mathbf{x})] \Phi(\mathbf{x})^T \nabla_{\mathbf{x}} \mathcal{F}_{j+1}(\psi_j^k(\mathbf{x})), \tag{20}$$

and $\psi_j^k(\mathbf{x}) = \mathbf{x} + \Phi(\mathbf{x})\mathbf{q}_j^k$. This iteration is initialized with $\mathbf{q}_j$ equal to zero, and embedded in a multiresolution framework to improve the convergence efficiency [2].

After the set of affine parameters between consecutive frames is estimated, we can obtain the intial estimate of the spatiotemporal parameter vector through two simple steps. In the first step, we find the transformation between the layer and each of the frames in the sequence. For this, we note that the composition of two affine transformations is still an affine transformation and, therefore, the transformation bewteen the layer and frame j in the sequence can be obtained by the composition of all the affine transformations between the layer and $\mathcal{F}_j$

$$\hat{\mathbf{\Psi}}_j(\mathbf{x}) = \psi_1(\mathbf{x}) \circ \ldots \circ \psi_{j-1}(\mathbf{x}),$$

where $\hat{\mathbf{\Psi}}_j$ is the transformation from the layer to frame j, the ψ_i are the pairwise transformations, $\circ$ the operator for the composition of affine tranformations, and $\hat{\mathbf{\Psi}}_1(\mathbf{x})$ the identity transformation. Denoting by $\hat{\mathbf{u}}_j$ the vector of affine parameters associated with $\hat{\mathbf{\Psi}}_j$,

$$\hat{\mathbf{\Psi}}_j(\mathbf{x}) = \mathbf{x} + \Phi(\mathbf{x})\hat{\mathbf{u}}_j$$

and comparing with equation (9) we obtain the relationship

$$\hat{\mathbf{u}}_j = \mathcal{T}_j \mathbf{p},$$

where $\mathbf{p}$ is the vector of spatiotemporal motion parameters. In the second step we find the initial estimate for this vector by solving this equation in the least squares sense [17]

$$\mathbf{p}^0 = \left[\mathcal{T}^T \mathcal{T}\right]^{-1} \mathcal{T}^T \hat{\mathbf{u}}, \tag{21}$$

where $\hat{\mathbf{u}} = (\hat{\mathbf{u}}_1^T, \ldots, \hat{\mathbf{u}}_N^T)^T$, and $\mathcal{T} = (\mathcal{T}_1^T, \ldots, \mathcal{T}_N^T)^T$.

4.4 Implementation complexity

In this section, we show that the temporal coeherence achieved through the introduction of the spatiotemporal motion model does not imply a significant increase in computational complexity over that already required for the computation of the optimal pairwise affine transformations.

For this we start by analyzing equation (13), and noticing that the bulk of the work resides on the computation of the matrices α_j^k and β_j^k as, for a given j, they involve cycling through all the pixels in the frame and for each pixel computing an affine transformation, the spatial image gradient, and the product of all the matrices involved in equations (15) and (16). Compared to all this, the complexity of evaluating each of the terms in equation (14) is, given α_j^k and β_j^k, negligible. Hence, the cost per iteration of equation (13), is basically the cost of evaluating all the N α_j, and β_j.

We now turn to equation (18) to see that, also here, for each j, the cost per iteration is basically that of evaluating $\hat{\alpha}_j$ and $\hat{\beta}_j$. Because, by comparing equations (15) and (16) with equations (19) and (20), it is clear that the cost of evaluating α_j is equal to that of evaluating $\hat{\alpha}_j$, and the cost of evaluating β_j is the same as that of evaluating $\hat{\beta}_j$, we conclude that the cost per iteration of equation (13) is approximatly the same as the cost of one iteration of all the $\mathbf{q}_j$ in equation (18). I.e. in terms of complexity the two procedures are similar, the only difference being that while in the case of the spatiotemporal model all the α_j and β_j need to be computed before the parameter vector can be updated, in the case of the pairwise model only $\hat{\alpha}_j$ and $\hat{\beta}_j$ are required for the computation of $\mathbf{q}_j$. This is ilustrated by Figure 3.

Figure 3. Computational sequence associated with the spatiotemporal (top) and the pairwise (bottom) models.

Since the spatiotemporal fit requires initialization through the pairwise model, and both processes have the same cost per iteration, the cost of introducing the temporal constraint is to, at most, double the computational requirements. In practice, this cost is much smaller, because the spatiotemporal fit requires a reduced number of iterations (tipically one or two) to converge. Obviously, this cost must be multiplied by the number of times that the minimization of equation (11) is performed. Our experiments reveal, however, that a single iteration of the procedure in section 4 is sufficient for most sequences and, hence, only one minimization is tipically required. This is in fact the case of all the experiments reported on section 5.

The major burden imposed by the temporal constraint on the motion is the requirement to store all the frames in the sequence during the computation of the motion parameter vector. It should be noticed, however, that this increase occurs only for secondary storage. Because the computation of α_j and β_j depends only on the summarizing layer and frame $\mathcal{F}_j$ the spatiotemporal fit only requires two frame buffers. Since this is also the number required by the pairwise model, the increase in complexity is negligible for applications where the sequences are available on secondary storage anyway (such as retrieval from video databases, generation of salient stills, or interactive applications in general).

5 Summarization results

In this section we report on the results of several experiments comparing the performance of video summarization based on temporally localized motion models against that achieved with the spatiotemporal representation now proposed. In all experiments, the localized motion estimates were computed through the procedure of section 4.3, but without performing the least squares fit of equation (21). I.e. the summarization layers were created directly from the affine transformations $\hat{\Psi}_j$. With respect to the spatiotemporal model, we used a second-order temporal smoothness constraint ($M = 2$ in equation (3), and performed a single iteration of the iterative minimization of equations (11) and (12). Under these settings, the computational cost of the two methods is comparable.

Figure 4 presents a sequence for which the assumption of a single global motion is realistic. This is the least interesting case as both the temporally localized and the global representation have no dificulty in creating a meaningful mosaic that captures a description of the entire scene.

A more dificult example is presented in Figure 5. The scene depicted in the figure contains two distinct motions: the camera is "zooming out", while the people are moving forward. In this case, because the background is rich in texture and occupies the bulk of the image, the camera motion is dominant for all the frames in the sequence and, once again, the two approaches produce similar results, i.e. all the frames are aligned with respect to the background.

Figures 6 and 7, present more challenging sequences, for which there is uncertainty with regards to the dominant motion. In both sequences it is hard to determine, a priori, which of the motions will dominate. In Figure 6 the bacgkround (a waterfall) presents a highly nonrigid motion which cannot be well aproximated by an affine field. The person in the foreground is itself a non-rigid object, but can be seen as a composition of several (aproximatly) rigid parts. The body is subject to a right to left motion which can be reasonably approximated by an affine transformation. On the other hand, the arms are subject to complex motion, the head rotates independently of the body and is partially ocluded by the heads, there are variations due to shading and facial expressions, and some of the parts (e.g. arms) are not visible throughout the entire sequence. In summary, this is an extremely hard sequence to summarize.

Due to all this complexity, the method based on temporally localized motion estimates oscilates between the several possible candidates for dominant motion. The motion trajectories are, therefore, very erratic, and the obtained layer is very fuzy. On the other hand, the approach based on spatiotemporal modelling seems to lock onto the body motion, leading to a much clearer summary of the scene content. Notice, in particular, how it would be easier to identify the person in the scene from the layer originated by the spatiotemporal procedure.

Figure 7 presents a final example, for which the relevance of the temporal consistency provided by spatiotemporal modeling is most clear. In this case, the background is static but weakly textured, the body of the person in the foreground is subject to an approximatly affine motion field, but the arm moves in a non-rigid fashion. Once again, it is unclear which motion is dominant and the temporally localized motion model leads to an erratic estimate and significant uncertainty in the recovered layer. On the other hand, the spatiotemporal model locks onto the body motion, leading to a layer that summarizes the scene content in a much more meaningfull way.

Notice, that when all the frames are aligned with respect to the same object (in this case the body), it is not only easier to recognize this object (the person), but also to understand the scene dynamics. In the case of the figure, the spatiotemporal layer provides a significantly better description for the motion of the arm throughout the sequence (even though the arm serves as a reference for some of the frames when the temporally localized model is used).

A Constraints on the temporal smoothness coefficients

Assuming that the trajectories of points in the image plane satisfy the smoothness constraint of equation (3), we now determine how the coefficients of that equation must, themselves, be constrained in order to guarantee affine motion between consecutive frames (equations (1) and (2)). For this we prove the following theorem.

Theorem 1 *Consider a motion trajectory satisfying the smoothness constraint of equation (3). Then* $\mathbf{x}_j$ *is an affine transformation of* $\mathbf{x}_0$ *if and only if each of the coefficients in the equation is itself an affine transformation of* $\mathbf{x}_0$*. I.e. for a motion trajectory satisfying equation (3),* $\mathbf{x}_j$ *is an affine transformation of* $\mathbf{x}_0$ *if and only if equations (4) and (5) are satisfied.*

Proof:

i) Assume equations (4) and (5) hold. Then by simple substitution in equation (3) we obtain equations (6) and (7). Comparing these equations with (1) and (2), it is clear that the former define an affine transformation between $\mathbf{x}_0$ and $\mathbf{x}_j$.

ii) In order to prove the reverse direction, we start by considering an *homogeneous* coordinate system [**?**], where $\mathbf{X}_j = (1, x_j, y_j)^T$ and noting that, in such a coordinate system, affine transformations are obtained by matrix multiplication. I.e. if $\mathbf{X}_j$ is an affine transformation of $\mathbf{X}_0$, then

$$\mathbf{X}_j = \mathbf{Q}_j \mathbf{X}_0, \tag{22}$$

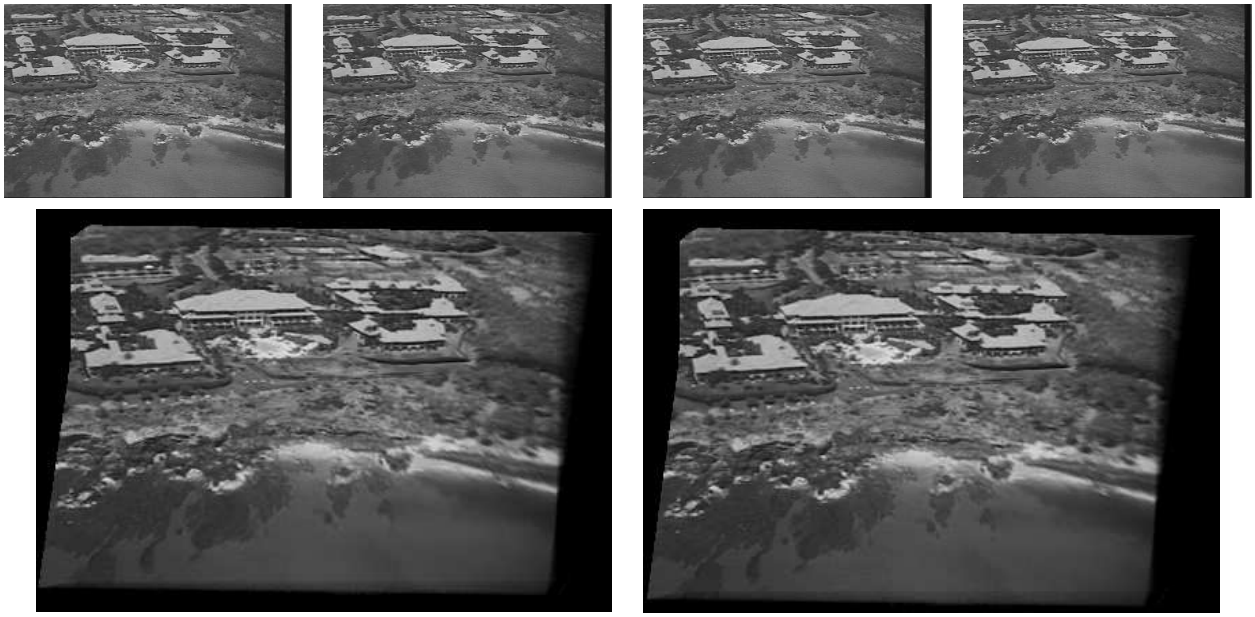

Figure 4. Four frames from a sequence containing a single global motion (top) and layers obtained with temporally localized (bottom-left) and with the spatiotemporal model (bottom-right). The layers are identical.

Figure 5. Four frames from a sequence containing two motions but reduced dominance ambiguity (top) and layers obtained with temporally localized (bottom-left) and with the spatiotemporal model (bottom-right). The layers are identical.

Figure 6. Four frames from a sequence containing several motions and high dominance ambiguity (top) and layers obtained with temporally localized (bottom-left) and with the spatiotemporal model (bottom-right). The spatiotemporal model leads to better image registration, and perceptually more meaningfull content-summarization.

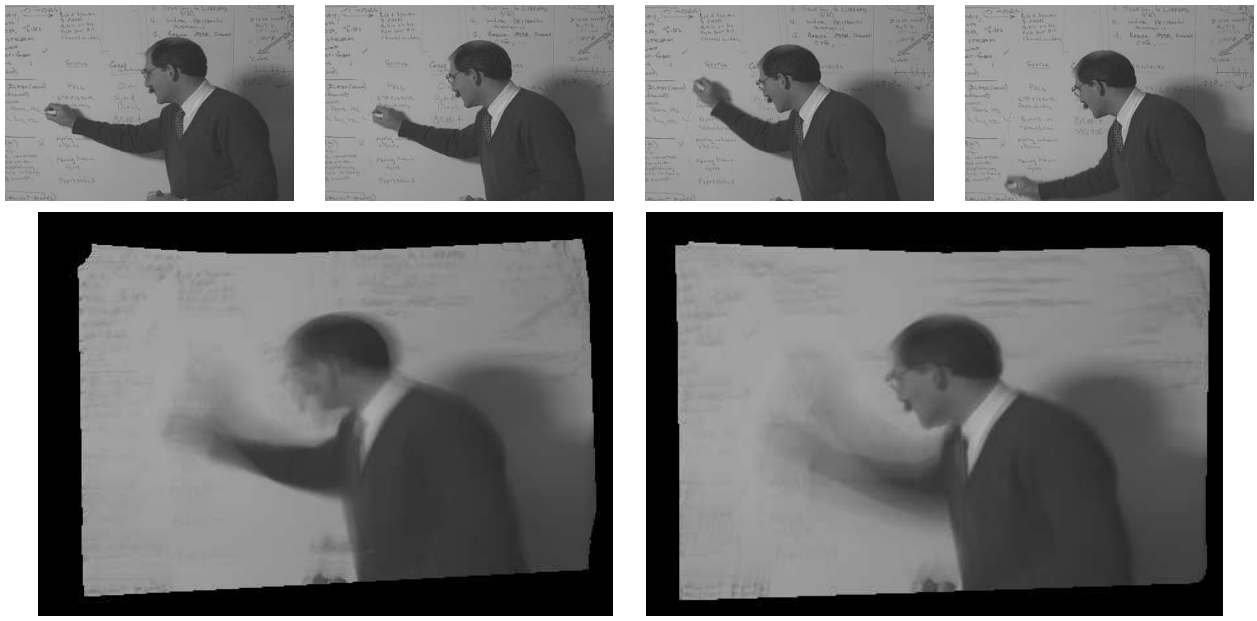

Figure 7. Four frames from a sequence containing three motions and high dominance ambiguity (top) and layers obtained with temporally localized (bottom-left) and with the spatiotemporal model (bottom-right). When localized estimates are used, registration is sometimes performed with respect to the body and other times with respect to the moving arm, leading to a summarization which is harder to understand. Notice that the layer on the left provides a better description of both the person and the action in the scene (arm motion).

where

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ - & - & - \\ - & - & - \end{bmatrix}, \tag{23}$$

and $-$ can be any real number. In the new coordinate system, equation (3) becomes

$$\mathbf{X}_j = \mathbf{X}_0 + \sum_{i=0}^{M} \mathbf{\Phi}_i t_j^i, \tag{24}$$

with $\mathbf{\Phi}_i = (0, \phi_i^T)^T$.

Assume that $\mathbf{x}_j$ is an affine transformation of $\mathbf{x}_0$. Then, combining equations (22) and (24)

$$\sum_{i=0}^{M} \mathbf{\Phi}_i t_j^i = (\mathbf{Q}_j - \mathbf{I})\mathbf{X}_0, j = 1, \ldots, N,$$

where N is the number of frames in the sequence. Next, pick any M distinct j (for example the first M) and construct the following system of equations

$$\begin{bmatrix} \vdots \\ \mathbf{X}_0^T(\mathbf{Q}_j^T - \mathbf{I}) \\ \vdots \end{bmatrix} = \begin{bmatrix} & \vdots & & \\ 1 & t_j & \ldots & t_j^{M-1} \\ & \vdots & & \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_0^T \\ \mathbf{\Phi}_1^T \\ \vdots \\ \mathbf{\Phi}_{M-1}^T \end{bmatrix}. \tag{25}$$

Calling $\mathbf{T}$ the matrix which is a function of the t_j^i, and noticing that it is a Vandermonde matrix, it is clear that (because all the j are different) it has full rank [7]. The system can thus be inverted, leading to

$$\mathbf{\Phi}_i^T = \left(\mathbf{T}^{-1}\right)_i \mathbf{V}, i = 1, \ldots, M,$$

where $\left(\mathbf{T}^{-1}\right)_i$ is the i^{th} row of $\mathbf{T}^{-1}$, and $\mathbf{V}$ the matrix on the left-hand side of equation (25). Hence, each $\mathbf{\Phi}_i$ is a linear combination of all the $(\mathbf{Q}_j - \mathbf{I})\mathbf{X}_0$ vectors, i.e.

$$\mathbf{\Phi}_i = \left(\sum_{j=0}^{M-1} \mu_j (\mathbf{Q}_j - \mathbf{I}) \right) \mathbf{X}_0, \; i = 1, \ldots, M.$$

Because the $\mathbf{Q}_j$ matrices are of the form given in equation (23), the matrices $(\mathbf{Q}_j - \mathbf{I})$ have zeros in all the positions of their first rows and the equation becomes

$$\begin{bmatrix} 0 \\ \phi_i \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ - & - & - \\ - & - & - \end{bmatrix} \mathbf{X}_0, i = 1, \ldots, M$$

i.e. ϕ satisfies equations (4) and (5) for all i. □ It follows from the properties of affine transformations that if, for all j, $\mathbf{x}_j$ is an affine transformation of $\mathbf{x}_0$, then it is also an affine transformation of $\mathbf{x}_{j-1}$, i.e. the motion between consecutive frames is affine.

B Parameter estimation

The optimal set of spatiotemporal motion parameters is, for a given layer, the one which minimizes equation (11). As pointed out in section 4.1, this minimization is carried out through the Gauss-Newton method. For a least squares cost function

$$\mathcal{J}(\mathbf{p}) = \sum_i J_i(\mathbf{p})^2,$$

this method consists of the iteration described by equation (13) with

$$\mathbf{d}^k = \left[\sum_i \nabla_{\mathbf{p}} J_i(\mathbf{p})^T \nabla_{\mathbf{p}} J_i(\mathbf{p}) \right] \sum_i J_i(\mathbf{p}) \nabla_{\mathbf{p}} J_i(\mathbf{p}).$$

For the cost function of equation (10)

$$J_i(\mathbf{p}) = \mathcal{F}_j(\mathbf{\Phi}(\mathbf{x})\mathcal{T}_j\mathbf{p}) - \mathcal{S}(\mathbf{x})$$

and

$$\nabla_{\mathbf{p}} J_i(\mathbf{p}) = \mathcal{T}_j^T \mathbf{\Phi}(\mathbf{x})^T \nabla_{\mathbf{x}} \mathcal{F}_j(\mathbf{\Psi}_j(\mathbf{x})),$$

where $\mathbf{\Psi}_j(\mathbf{x})$ is defined by equation 9, leading to

$$\mathbf{d}^k = \left[\sum_{j,\mathbf{x}} \mathcal{T}_j^T \mathbf{\Phi}(\mathbf{x})^T \nabla_{\mathbf{x}} \mathcal{F}_j(\mathbf{\Psi}_j(\mathbf{x})) \nabla_{\mathbf{x}} \mathcal{F}_j(\mathbf{\Psi}_j(\mathbf{x}))^T \mathbf{\Phi}(\mathbf{x}) \mathcal{T}_j \right]^{-1} \sum_{j,\mathbf{x}} (\mathcal{F}_j(\mathbf{\Psi}(\mathbf{x})) - \mathcal{S}(\mathbf{x})) \, \mathcal{T}_j^T \mathbf{\Phi}(\mathbf{x})^T \nabla_{\mathbf{x}} \mathcal{F}_j(\mathbf{\Psi}_j(\mathbf{x})). \tag{26}$$

Since the $\mathcal{T}_j$ do not depend on $\mathbf{x}$, they can be taken out of the summation with respect to the spatial coordinates, leading to equations (14) to (16). Because the matrix within the square brackets is positive semidefinite, $\mathbf{d}^k$ is a descent direction if it is also invertible [4]. In our implementation the matrix inversion is based on a singular value decomposition [**?**],to minimize the sensitivity to noise when the matrix does not have full rank.

References

[1] G. Adiv. Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, Vol. PAMI-7, July 1985.

[2] P. Anandan, J. Bergen, K. Hanna, and R. Hingorani. Hierarchical Model-Based Motion Estimation. In M. Sezan and R. Lagendijk, editors, *Motion Analysis and Image Sequence Processing*, chapter 1. Kluwer Academic Press, 1993.

[3] J. Barron, D. Fleet, and S. Beauchemin. Performance of Optical Flow Techniques. *International Journal of Computer Vision*, vol. 12, 1994.

[4] D. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1995.

[5] T. Darrel and A. Pentland. Cooperative Robust Estimation Using Layers of Support. Technical Report 163, MIT Media Laboratory Perceptual Computing Group, June 1993.

[6] M. Hansen, P. Anandan, K. Dana, G. Wal, and P. Burt. Real-Time Scene Stabilization and Mosaic Construction. In *Proc. ARPA Image Understanding Workshop*, 1994.

[7] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[8] M. Irani and S. Peleg. Image Sequence Enhacement Using Multiple Motions Analysis. *Technical Report 91-15, Hebrew Unviersity of Jerusalem*, December 1991.

[9] M. Irani, B. Rousso, and S. Peleg. Detecting and Tracking Multiple Moving Objects Using Temporal Integration. In *Proc. ECCV, Santa Margherita, Italy*, 1992.

[10] M. Irani, B. Rousso, and S. Peleg. Computing Occluding and Transparent Motions. *International Journal of Computer Vision*, 12:1, 1994.

[11] R. Kumar, P. Anandan, M. Irani, J. Bergen, and K. Hanna. Representation of Scenes from Collections of Images. In *Proc. IEEE Workshop on Representation of Visual Scenes*, 1995.

[12] S. Mann and R. Picard. Virtual Belows: Constructing High Quality Stills from Video. In *Proc. ICIP*, 1994.

[13] M.Lee, W. Chen, C. Lin, C. Gu, T. Markoc, S. Zabinsky, and R. Szeliski. A Layered Video Object Coding System Using Sprite and Affine Motion Model. Vol. 7, February 1997.

[14] J. Odobez and P. Bouthemy. Robust Multiresolution Estimation of Parametric Motion Models in Complex Image Sequences. In *Proc. seventh EUSIPCO European Conf. Sig. Proc.*, 1994.

[15] H. Sawhney. Simplifying Motion and Structure Analysis Using Planar Paralax and Image Warping. In *Proc. ICPR*, 1994.

[16] H. Sawhney and S. Ayer. Compact Representations of Videos Through Dominant and Multiple Motion Estimation. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, Vol. 18, August 1996.

[17] G. Strang. *Linear Algebra and its Applications*. Harcourt Brace Jovanovich, Inc., 1985.

[18] R. Szeliski. Image Mosaicing for Tele-Reality Applications. In *Proc. IEEE Workshop Applications of Computer Vision*, 1994.

[19] L. Teodosio. Salient Stills. Master's thesis, MIT Media Lab, 1992.

[20] J. Wang and E. Adelson. Representing Moving Images with Layers. *IEEE Trans. on Image Processing*, Vol. 3, September 1994.