Extraction of Layers of Similar Motion through Combinatorial Techniques

Romain Dupont[1,2], Nikos Paragios[1], Renaud Keriven[1], and Phillipe Fuchs[2]

[1] Atlantis Research Group, CERTIS
ENPC, Marne-La-Vallee, France
fax: (+33) 1 64 15 21 99
tel: (+33) 1 64 15 21 72
{dupont,paragios,keriven}@certis.enpc.fr

[2] Centre de Robotique, CAOR
ENSMP, Paris, France
fuchs@ensmp.fr

Abstract. In this paper we present a new technique to extract layers in a video sequence. To this end, we assume that the observed scene is composed of several transparent layers, that their motion in the 2D plane can be approximated with an affine model. The objective of our approach is the estimation of these motion models as well as the estimation of their support in the image domain. Our technique is based on an iterative process that integrates robust motion estimation, MRF-based formulation, combinatorial optimization and the use of visual as well as motion features to recover the parameters of the motion models as well as their support layers. Special handling of occlusions as well as adaptive techniques to detect new objects in the scene are also considered. Promising results demonstrate the potentials of our approach.

1 Introduction

Motion perception is an important characteristic of biological vision used as input in various tasks like to determine the focus of attention, etc. Therefore, motion analysis has been a long time objective of computational vision, a low to mid-level task.

The segmentation of an image sequence into regions with homogeneous motion is a challenging task in video processing [1,2,3,4,5,6] that can be used for various purposes such as video-based surveillance and action recognition. In addition, it can be considered for video compression [1] since the motion model and the corresponding supporting layers provide a compact representation of the scene.

Motion/displacement is a well-defined measurement in the real world. On the other hand, one can claim that recovering the corresponding quantity in the image plane is a tedious task. Optical flow calculation [7,8,9,10,5] is equivalent with the estimation of a motion displacement vector for each pixel of the image

plane that satisfies the visual constancy constraint. Such a task refers to an ill-posed problem where the number of unknown variables exceeds the number of constraints. The use of smoothness constraints [11] and other sophisticated techniques were consider to address such an issue.

Parametric motion models are an alternative to dense optical flow estimation [12,10,13,5]. The basic assumption of such a technique is that for an image block, the 2D motion in the image plane can be modeled using a parametric transformation. Such assumption is valid when the block refers to a projection of 3D planar patch.

The objective of this work is to recover different planar surfaces, or motion layers, and the motion parameters describing their apparent displacements. In the literature, a K-mean clustering algorithm [1] on the motion estimates, or a minimum description length (MDL) [3] were considered to determine the number of motion planes. In the latter case, the extraction is done according to a maximum likelihood criterion, followed by optimization by the Expectation-Maximization algorithm [14,3]. More recent approaches [15] refer to region growing techniques within combinatorial optimization [16].

In this paper, we present an iterative technique to estimate the motion parameters, the support of each layer as well as its visual properties. The latter is used to overcome cases where motion information is not enough to estimate the support. Our approach addresses in a very efficient fashion motion estimation through a robust incremental technique, accounts for occlusions through a forward/backward transformation and recover the layer support through a MRF-based formulation that is optimized with the graph cut approach and the α-expansion algorithm. To this end, motion residuals, visual appearance as well as spatial and temporal smoothness constraints are considered.

The reminder of this paper is organized according to the following fashion. In section 2, we briefly introduce the problem under consideration while in section 3, an iterative approach to recover the motion parameters is presented. The extraction of the support regions of the different layers is part of section 4, while in section 5 we discuss the implementation details of our approach and provide experimental results and future directions.

2 Decomposition of Scenes in Motion Layers

Let us consider a static scene that consists of several planes, a moving observer on the scene, and a sequence of 2D images acquired by the observer. Due to the camera's ego motion from one image to the next, one will be able to observe motion on the static parts of the scene. Such motion (2D projection) depends on the camera projection model and the depth level of the different planes in the 3D scene. Here, we consider the projective camera model. The concept of motion decomposition in layers [1] consists of separating the image domain into n support regions $\mathcal{S}_i, i \in [1, n]$ with their corresponding motion models $(\mathcal{A}_i, \sigma_i)$. The i-th layer $\mathcal{L}_i, i \in [1, n]$ is defined as the couple $(\mathcal{S}_i, \mathcal{A}_i)$.

The image domain Ω is partioned into n disjoint sets $\mathcal{S}_i$ such that $\cup_{i=1}^{n}\mathcal{S}_i = \Omega, \quad \mathcal{S}_i \cap \mathcal{S}_j = \emptyset, i \neq j$ and neither the number of layers, not their support regions, nor their motion parameters are known. In the reminder of this paper, we will propose efficient methods to address the estimation of these unknown variables.

In terms of motion, one can find in the literature parametric models of various degrees of freedom like rigid, similarity, homographic, quadratic, etc. Affine model is a reasonable compromise between low complexity and fairly good approximation of the non-complex motions of objects at about the same depth. It consists of 6 degrees of freedom,

$$\mathcal{A}(x,y) = \begin{pmatrix} a_0 \cdot x + a_1 \cdot y + a_2 \\ a_3 \cdot x + a_4 \cdot y + a_5 \end{pmatrix}$$

Such a model describes accurately (as detailed in [2]) the motion induced by a planar object viewed from a moving camera. Furthermore, this model describes well the 2D motion of the projection of an arbitrary 3D scene undergoing camera rotations, zoom and small camera translations [17]. Likewise, when the overall depth of the object is greater than the depth within the object, the model describes the image motion with a sub-pixel accuracy.

Motion estimation consists of recovering the parameters of this model such that a correspondence between the projections of the same 3D patch within two consecutive images is established. In principle, motion estimation refer to an ill-posed problem since neither the projection model, neither the internal parameters of the cameras are known and therefore constraints are to be deduced from the images toward its estimation.

3 Recovering the Parameters of the Motion Models

The intensity preservation constraint (equivalent to the brightness constancy assumption) [7] is often used to address motion estimation. The essence of this constraint is that under the assumption of planar, Lambertian surfaces and without global illumination changes, the appearance of the 2D projection of the same 3D patch will not change over time. Therefore if a motion vector $\mathbf{dx} = (dx, dy)$ is assumed for the pixel $\mathbf{x} = (x, y)$, then the following condition is to be satisfied:

$$I(\mathbf{x};t) \approx I(\mathbf{x} + \mathbf{dx}; t+1), \ (i)$$
$$I(\mathbf{x};t) \approx I(\mathbf{x} + \mathcal{A}(\mathbf{x}); t+1), \ (ii)$$

for the case of dense motion (i) and for the case of affine motion (ii). Given such a condition, one can define the total motion residual according to:

$$E(\mathcal{A}) = \int_{\Omega} \left| I(\mathbf{x};t) - I(\mathbf{x} + \mathcal{A}(\mathbf{x}); t+1) \right|^2 d\mathbf{x} \tag{1}$$

Solving the inference problem, that is recovering the parameters of the affine model through the lowest potential of the above function is a common practice in

computational vision. One can consider an iterative process using a well adopted first order linear form of optical flow constraint:

$$\mathcal{A}(\mathbf{x}) \cdot \nabla I(\mathbf{x};t) + \nabla_t I(\mathbf{x}) = 0 \tag{2}$$

where ∇I to the spatial gradient and $\nabla_t I$ to the temporal gradient. One can consider minimizing the corresponding cost function

$$E(\mathcal{A}) = \int_\Omega |\mathcal{A}(\mathbf{x}) \cdot \nabla I(\mathbf{x};t) + I(\mathbf{x};t+1) - I(\mathbf{x};t)|^2 \, d\mathbf{x} \tag{3}$$

with standard linear methods that will fail though to capture large displacements between two successive frames. To overcome this limitation, we consider an iterative process as prescribed in [12]. To this end, one can consider an incremental update of the motion parameters where at each step, given the current estimates $\mathcal{A}$, we seek to recover an improvement of the estimation $\Delta\mathcal{A}$ such that the accumulation of existing parameters and the improvement minimizes the following residual error:

$$E(\Delta\mathcal{A}) = \int_\Omega \left[I(\mathbf{x};t) - I(\mathbf{x}+\mathcal{A}(\mathbf{x});t+1) - \Delta\mathcal{A}\nabla I(\mathbf{x}+\mathcal{A}(\mathbf{x});t+1)\right]^2 d\mathbf{x} \tag{4}$$

that has a closed form solution. While one can claim that such an incremental method will improve the estimation process, it will still suffer from the presence of outliers resulting an estimation bias. Robust estimation process like an M-estimator can be used to overcome this limitation. Such a method assigns weights $we(\mathbf{x})$ to the constraints at the pixel level that are disproportional to their residual error, thus rejecting the motion outliers. To this end, one should define the influence function, $\psi(x)$ like for example the Tukey's estimator [Fig. 1]:

$$\psi(x) = \begin{cases} x({K_\sigma}^2 - x^2)^2 & \text{if } |x| < K_\sigma \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where K_σ characterizes the shape of the robust function. The weights $we(\mathbf{x})$ are then computed as following: $we(\mathbf{x}) = \frac{\psi(r(\mathbf{x}))}{r(\mathbf{x})}$ ([12]).

One can now consider such a process for each layer in an independent fashion, that consists of minimizing the following cost function

$$E(\Delta\mathcal{A}_1, ..., \Delta\mathcal{A}_n) = \tag{6}$$
$$\sum_{k=1}^{n} \int_\Omega \chi_{\mathcal{S}_i}(\mathbf{x}) \rho \left[I(\mathbf{x};t) - I(\mathbf{x}+\mathcal{A}_i(\mathbf{x});t+1) - \Delta\mathcal{A}_i \nabla \; I(\mathbf{x}+\mathcal{A}_i(\mathbf{x});t+1)\right] d\mathbf{x}$$

where $\chi_{\mathcal{S}_i}$ is the characteristic function of the region $\mathcal{S}_i$. Once the support layers are known, one can proceed to a straightforward estimation of the motion models. Occlusions due to motion of the observer and the scene often arise in motion and stereo reconstruction and must be taken into account. Such a case can be accounted for through the joint estimation of the backward/forward motion;

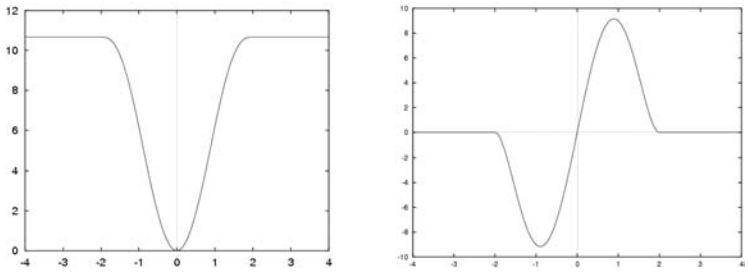

Fig. 1. Tukey function ρ (on the left) and its derivative ψ (on the right)

Let (i) $\mathcal{A}_1, ..., \mathcal{A}_n$ be the motion models that create visual correspondences between the images t and $t+1$ (such that $I(\mathbf{x};t) = I((\mathbf{x}+\mathcal{A}(\mathbf{x});t+1))$ and (ii) $\mathcal{A}'_1, ..., \mathcal{A}'_n$ the ones that create visual correspondences between the images $t+1$ and t (such that $I(\mathbf{x};t+1) = I((\mathcal{A}'(\mathbf{x});t))$. Then, we seek for a simultaneous estimate of the improvements of both models forward/backward according to:

$$E(\Delta\mathcal{A}_1, ..., \Delta\mathcal{A}_n, \Delta\mathcal{A}'_1, ..., \Delta\mathcal{A}'_n) = \tag{7}$$
$$\sum_{k=1}^{n} \int_{\Omega} \chi_{\mathcal{S}_i}(\mathbf{x}) \rho \left[I(\mathbf{x};t+1) - I(\mathbf{x}+\mathcal{A}'_i(\mathbf{x});t) - \Delta\mathcal{A}'_i(\mathbf{x}) \nabla\ I(\mathbf{x}+\mathcal{A}'_i(\mathbf{x});t) \right]$$
$$+ \sum_{k=1}^{n} \int_{\Omega} \chi_{\mathcal{S}_i}(\mathbf{x}) \rho \left[I(\mathbf{x};t) - I(\mathbf{x}+\mathcal{A}_i(\mathbf{x});t+1) - \Delta\mathcal{A}_i(\mathbf{x}) \nabla\ I(\mathbf{x}+\mathcal{A}_i(\mathbf{x});t+1) \right]$$

Under the assumption on the absence of occlusion, one can consider that for a given pixel, both transformations capture its real motion and therefore, posing $\mathbf{x}' = \mathbf{x} + \mathcal{A}(\mathbf{x})$, the following condition will be satisfied:

$$\mathbf{x}' + \mathcal{A}'(\mathbf{x}') = \mathbf{x} \tag{8}$$

Such a concept is presented in [Fig. (2)]. The distance between the origins of the pixel $\mathbf{x}$ and its position upon the application of forward/backward motion models;

$$\mathcal{D}(\mathbf{x}) = \|\mathbf{x}' + \mathcal{A}'(\mathbf{x}') - \mathbf{x}\|^2 \tag{9}$$

can be considered as an indicator on the presence of occlusions and used to ponderate the influence function ψ defined in equation 5:

$$\psi(x) = \psi(x) \cdot \frac{1}{1+\mathcal{D}(\mathbf{x})} \tag{10}$$

Hence, occlusions will have low influence on the estimation process. However, given the robust estimation process that was considered for any given partition of the image, we will be able to recover affine motion models that, to some extent, describe the observed motion. Therefore, we refer to the egg and the chicken problem where it is crucial to have a consistent estimation of the support layers.

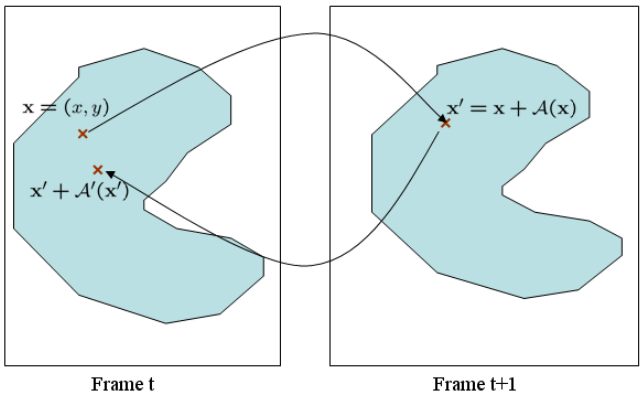

Fig. 2. Occlusion Detection: the Euclidean distance between the pixel origin and the corresponding one after being transformed through the forward/backward motion is used to detect occlusions.

4 Extraction of Support Layers

Let us consider a partition of the image into n segments

$$\{\mathcal{S}_1, ..., \mathcal{S}_n\} : \cup_{i=1}^{n} \mathcal{S}_i = \Omega, \quad \mathcal{S}_i \cap \mathcal{S}_j = \emptyset, i \neq j$$

The problem of extracting support within the layer decomposition process consists of selecting for each pixel of Ω, the label among these n that dictates the most appropriate motion model for this image patch. One can see such a task in the form of a labeling problem, where one should assign to the pixel $\mathbf{x}$ a label $\omega(\mathbf{x}) \in [1, n]$ according to a certain criterion. Within our approach we adopt motion and appearance terms to address such a labeling process while imposing certain spatial and temporal smoothness constraints on the label space.

4.1 Motion Criterion

Let us consider the distribution of the residual errors within a layer. Under the assumption of proper motion estimation as well as correct classification, one can consider that residual errors are due to the presence of noise that in the most general case white.

Therefore, the motion residual $r_i(\mathbf{x}) = |I(\mathbf{x}, t) - I(\mathbf{x} + \mathcal{A}_i(\mathbf{x}), t + 1)|$ for the layer $\mathcal{L}_i$ obeys a normal law $G(\mu_i, \sigma_i)$. Consequently, the probability for given a pixel within the region $\mathcal{L}_i$ to actually being part of this region according to the observed residual is:

$$p\left(r_i(\mathbf{x})|\mathcal{A}_i, \sigma_i\right) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(\frac{-{r_i}^2(\mathbf{x})}{2{\sigma_i}^2}\right) \tag{11}$$

where σ_i is the standard deviation computed during the motion estimation for each layer support. We consider the following robust estimator which tolerates

50% of outliers efficiently:

$$\sigma_i = 1.4826 \left\{ \underset{\mathbf{x} \in \mathcal{S}_i}{\text{median}} |r_i(\mathbf{x})| \right\} \tag{12}$$

One can assume independence on the distribution of the residual errors within the pixels of a support layer $\mathcal{S}_i$ and given the expected distribution, would like to maximize the conditional density,

$$\begin{aligned} p_i(r_i(\mathbf{x})|\mathcal{A}_i, \sigma_i) &= p\left(\bigcap_{\mathbf{x} \in \mathcal{S}_i} \{r_i(\mathbf{x})|\mathcal{A}_i, \sigma_i\} \right) \\ &= \prod_{\mathbf{x} \in \mathcal{S}_i} p(r_i(\mathbf{x})|\mathcal{A}_i, \sigma_i) \end{aligned} \tag{13}$$

Furthermore, independence on the residual errors is assumed between the different support layers. Then, using the Bayes rule, one can consider the posterior for the labeling process ω according to the motion characteristics in the following fashion:

$$p(r_i(\mathbf{x})|\mathcal{A}_i, \sigma_i, \omega) = \prod_{\mathbf{x} \in \Omega} p_{\omega(\mathbf{x})}(r_{\omega(\mathbf{x})}(\mathbf{x})|\mathcal{A}_{\omega(\mathbf{x})}, \sigma_{\omega(\mathbf{x})}) \tag{14}$$

where the assumption that all labelings are equal probable was made. Maximizing the posterior is equivalent with minimizing the negative log-likelihood of such a density:

$$\begin{aligned} E_{motion}(\omega) &= -\int_{\Omega} \log \left[p_{\omega(\mathbf{x})} \left(r_{\omega(\mathbf{x})}(\mathbf{x})|\mathcal{A}_{\omega(\mathbf{x})}, \sigma_{\omega(\mathbf{x})} \right) \right] d\mathbf{x} \\ &= \int_{\Omega} \left(\log \left[\sigma_{\omega(\mathbf{x})} \right] + \frac{{r_{\omega(\mathbf{x})}}^2(\mathbf{x})}{2{\sigma_{\omega(\mathbf{x})}}^2} \right) d\mathbf{x} \end{aligned} \tag{15}$$

The lowest potential of this objective function will classify image pixels according to their residual errors. Such classification will reflect the maximum posterior according to the expected distribution of the residual error for each layer. However, motion estimates are reliable when image structure is present and consequently motion-based classification may be ambiguous in some cases, like in the lack of texture.

4.2 Visual Appearance Criterion

We overcome this limitation through the introduction of a visual grouping constraint, where a classification according to the observed intensities is to be considered. To this end, we consider a flexible parametric density function - Gaussian mixture - to describe the visual properties of each layer;

$$p_i(I) = \sum_{k=1}^{m_i} \pi_{\{i,k\}} \; p_{\{i,k\}} \left(I|\mu_{\{i,k\}}, \Sigma_{\{i,k\}} \right) \tag{16}$$

where $p_i()$ is the colour distribution of the i-th layer that consists of m_i Gaussian components with $\pi_{\{i,k\}} \in [0,1]$ being the prior of the component k (or its proportion in the mixture) and $(\mu_{\{i,k\}}, \Sigma_{\{i,k\}})$ the mean and the covariance matrix of this component. These parameters are estimated from an observed distribution through an EM algorithm [18]. To efficiently determine the number of Gaussian components per mixture, a Minimum Description Length (MDL) criterion is considered. Such a colour distribution has been chosen because it provides a simple and efficient way to learn the visual characteristics of each layer while not being constraint to be a unimodal. Therefore even regions with very different colour characteristics that belong to the same plane will be accounted for. Then, the posterior segmentation probability can be considered as the most efficient metric to recover the separation of the image domain into regions of support for the different layers according to their expected appearance properties. Similar to the case of motion, we consider that layers as well as pixels within regions are independent and all possible labelings are equally probable, leading to the following objective function

$$\begin{aligned} E_{visual}(\omega) &= -\int_{\Omega} \log\left[p_{\omega(\mathbf{x})}\left(I(\mathbf{x})\right)\right] d\mathbf{x} \\ &= \int_{\Omega} \log\left(\sum_{k=1}^{m_{\omega()}} \pi_{\{\omega(),k\}}\ p_{\{\omega(),k\}}\left(I()|\mu_{\{\omega(),k\}}, \Sigma_{\{\omega(),k\}}\right)\right) d\mathbf{x} \end{aligned} \tag{17}$$

where $\mathbf{x}$ were omitted from the notation due to the lack of space. One can seek the lowest potential of these two terms weighted according to some constant to recover the most appropriate image partition in terms of support layers. Such a method will be able to determine support through an independent decision process according to the similarity between the observed image and the expected properties in terms of appearance and residual error. Such an independent process will form several discontinuities that will be quite disrupting to the human eye and will violate the condition that images are assumed to be consistent at a local scale.

4.3 Spatial Smoothness

Such a limitation is often addressed using local smoothness constraints on the label domain, that consists of saying that neighborhood pixels should belong to the same layer;

$$E_{smooth}(\omega) = \int_{\Omega} \left[\int_{\mathcal{N}(\mathbf{x})} \mathcal{V}(\omega(\mathbf{x}), \omega(\mathbf{u}))d\mathbf{u}\right] d\mathbf{x} \tag{18}$$

where $\mathcal{N}(\mathbf{x})$ is the local neighborhood of $\mathbf{x}$. Here, the function $\mathcal{V}$ has the following form (named Pott's model):

$$\mathcal{V}(\omega(\mathbf{x}), \omega(\mathbf{u})) = \begin{cases} +\alpha_{diff} & , \omega(\mathbf{x}) \neq \omega(\mathbf{u}) \\ 0 & , \omega(\mathbf{x}) = \omega(\mathbf{u}) \end{cases} \tag{19}$$

with $\alpha_{diff} > 0$ and the local neighborhood consists of pixels that are 4- or 8-connected. Such a term will penalize discontinuities in the support space that are also discontinuities in the motion space. While such an assumption seems natural, it is not valid when considering pixels that refer to real discontinuities of the observed scene. In that case, we should tolerate label discontinuities, which is satisfied through a multiplicative factor applied to the smoothness potential that is inversely proportional to the image gradient [19], or:

$$\mathcal{V}^{\mathbf{g}}(\omega(\mathbf{x}), \omega(\mathbf{u})) = \mathcal{V}(\omega(\mathbf{x}), \omega(\mathbf{u})) \exp\left(-\frac{\|I(\mathbf{x}) - I(\mathbf{u})\|^2}{2\sigma^2}\right) \tag{20}$$

Such a term will produce smoothness on the label space in rather uniform regions while it will relax the constraint in areas where physical discontinuities are present.

One can further explore smoothness in the temporal domain. Given that we are treating sequences of images observing the same scene, the assumption of smoothness within the labeling in the temporal space is valid.

4.4 Temporal Smoothness

Let us consider a sequence of images $I(;1), I(;2), ..., I(;\tau)$, as well as a sequence of labelings $\omega(;1), \omega(;2), ..., \omega(;\tau)$. We assume that we are currently treating the image $t \in [1, \tau]$ and the motion models $\mathcal{A}_1, ..., \mathcal{A}_{t-1}$ have correctly been estimated. Then, we define a smoothness function on the temporal space that takes into account the motion models and the support layers of the previous frame:

$$\mathcal{V}^{\mathbf{t}}(\omega(\mathbf{x};t)) = \begin{cases} +\alpha_{diff} & , \omega(\mathbf{x};t) \neq \omega(\mathcal{A}_{t-1}^{-1}(\mathbf{u});t-1) \\ 0 & , \omega(\mathbf{x};t) = \omega(\mathcal{A}_{t-1}^{-1}(\mathbf{u});t-1) \end{cases} \tag{21}$$

where $\mathcal{A}_{t-1}^{-1}$ is the inverse motion model that establishes correspondences between the frames $I(;t)$ and $I(;t-1)$. One can now introduce an additional temporal smoothness term:

$$E_{tsmooth}(\omega) = \int_{\Omega} \mathcal{V}^{\mathbf{t}}(\omega(\mathbf{x}))d\mathbf{x} \tag{22}$$

where particular attention is to be paid to address the presence of new objects in the scene. Motion residual errors, visual consistency and spatial and temporal smoothness can now be considered to recover the optimal partition of the image given the expected characteristics of each layer, or:

$$E(\omega) = E_{motion}(\omega) + \alpha E_{visual}(\omega) + \beta E_{smooth}(\omega) + \gamma E_{tsmooth}(\omega) \tag{23}$$

The lowest -sub-optimal- potential of the discrete form of the above function can be determined using several techniques of various complexity like the iterated conditional modes [20], the highest confidence first [21], the mean field and simulated annealing [22] and the min-cut max flow approach [23]. Because of its efficiency, the graph-cut framework is retained to recover the optimal solution on the label assignement problem [23].

5 Graph-Cuts and Implementation

The graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is a set of nodes $\mathcal{V}$ and directed edges $\mathcal{E}$ connecting them. Two special *terminal* nodes are present: the *source* s and the *sink* t. Each edge connecting nodes p and q is assigned a weight $w(p, q)$. We break all edges in two groups: n-links and t-links. A n-link is an edge connecting two non-terminal nodes. A t-link connects a non-terminal node with a terminal node, s or t. The cut C is a partitioning of the nodes of the graph into two disjoint subsets S et T such that the source $s \in S$ and the sink $t \in T$. Its cost $c(S, T)$ is the sum of the weights of all edges (p, q) such that $p \in S$ and $q \in T$. The minimum cut is the cut with minimal cost and can be determined in polynomial time with a max-flow extraction algorithm.

For the energy (23), finding directly the optimal solution is not feasible in practical. Indeed, the problem of multi-labeling is NP-hard and no polynomial method is available to obtain the optimal solution. However, the α-expansion algorithm [23] gives fastly a good approximation of this energy which is guaranteed to be within a factor of 2 from the optimal one. To minimize the energy (23), we proceed as follows: we start with an initial layer assignments, obtained at the previous iteration. Then, for each $\alpha \in [1, n]$, we improve this energy by modifying some labelings to the label α via an *α-expansion move* which we describe here: considering a binary graph $\mathcal{G}$, each pixel $\mathbf{x} \in \Omega$ is representd by a non-terminal node p connected to the source with a weight $t_{s,p}$ and to the sink with a weight $t_{p,t}$. Each pair of neighbouring nodes (p, q) - if their layer assignments are different - are linked through an intermediate node a with weights $t_{s,a}$, $t_{p,a}$ and $t_{a,q}$ respectively. For t-link weights, we define the data cost function $D_p(\omega)$ as $D_p(\omega) = E_{motion}(\omega) + \alpha E_{visual}(\omega) + \gamma E_{tsmooth}(\omega)$. The table 1 summarizes all the weights associated to the n- and t-links. The minimal cut gives the new layer assignments which is optimal considering label α against all the others.

link	weight	for
$t_{s,p}$	$D_p(\omega(\mathbf{x}))$	$\omega(\mathbf{x}) \neq \alpha$
$t_{s,p}$	∞	$\omega(\mathbf{x}) = \alpha$
$t_{p,t}$	$D_p(\alpha)$	$\forall\ \omega(\mathbf{x})$
$t_{s,a}$	$\mathcal{V}(\omega(\mathbf{x}), \omega(\mathbf{u}))$	$\omega(\mathbf{x}) \neq \omega(\mathbf{u})$
$t_{p,a}$	$\mathcal{V}(\omega(\mathbf{x}), \alpha)$	$\omega(\mathbf{x}) \neq \omega(\mathbf{u})$
$t_{a,q}$	$\mathcal{V}(\alpha, \omega(\mathbf{u}))$	$\omega(\mathbf{x}) \neq \omega(\mathbf{u})$
$t_{p,q}$	$\mathcal{V}(\omega(\mathbf{x}), \alpha)$	$\omega(\mathbf{x}) = \omega(\mathbf{u})$

Table 1. Weights associated to each nodes of the α-expansion graph.

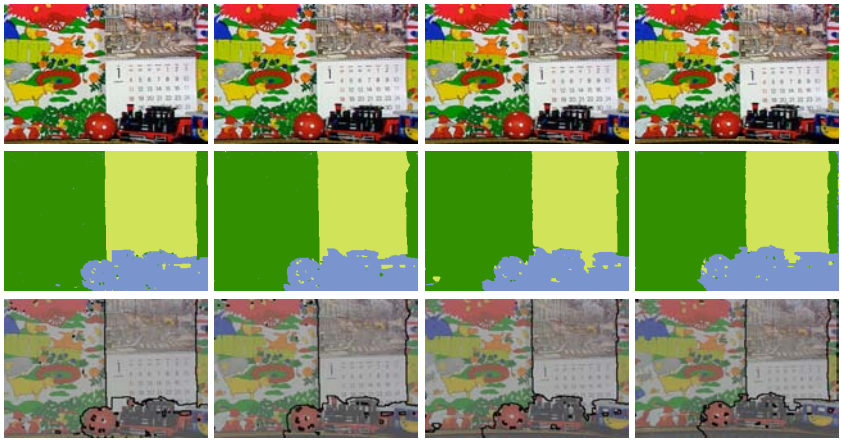

Fig. 3. Results on calendar sequence (one frame on four is considered). Each column represent a frame: frames 0,12,24,36 are represented here. First row, original sequence; second row, the layers extracted ; third row, superposition of layers boundaries with original sequence.

5.1 Implementation details

Once appropriate modules have been presented to address each sub-task, now we can proceed to the definition of the overall concept. In the first image, a random sampling into $\mathcal{N}$ segments is considered. These segments are used as support layers, and the motion as well as visual properties are obtained. Such measures are then introduced to the α-expansion algorithm that will provide a new image partition with different visual and motion properties. The process is repeated until convergence. Initialization of the layer support from one image to the next is done through the motion models and the same process as the one of the first frame is considered. One critical step is the estimation of the number of layers. Toward this end, we use two techniques sequentially. The first one reduces the number of layers and the second one detects new layers (or new objects) which appear in the video.

5.2 On the number of layers

We merge two layers if their motions are similar. As motion parameters does not define uniquely the motion over all the layer support, rather than considering them directly, we consider the optical flow generated by the two motion models. Hence, using the notation introduced in section 3, the motion similarity criterion r_{ij} between two layers i and j is computed as follows:

$$r_{ij} = \frac{1}{|\mathcal{S}_i|} \int_\Omega (\mathcal{A}_i(\mathbf{x}) - \mathcal{A}_j(\mathbf{x}))^2 \chi_{\mathcal{S}_i}(\mathbf{x}) d\mathbf{x} + \frac{1}{|\mathcal{S}_j|} \int_\Omega (\mathcal{A}_i(\mathbf{x}) - \mathcal{A}_j(\mathbf{x}))^2 \chi_{\mathcal{S}_j}(\mathbf{x}) d\mathbf{x}$$

where $|\mathcal{S}_i| = \int_\Omega \chi_{\mathcal{S}_i}(\mathbf{x}) d\mathbf{x}$ (similarly for $|\mathcal{S}_j|$). If r_{ij} drops down under a certain threshold, the two layers i and j are merged together. Furthermore, layers with too small support (and so giving bad motion estimation) or with too important variance (due to too many outliers in the support) are deleted.

New objects which appear must be detected and classified in new layers. Toward this end, we proceed as follows: first, warp residual is computed for the whole frame, giving a residual map. We apply a binary threshold $\mathcal{T}$ to this map. All pixels whose residual is higher than $\mathcal{T}$ are extracted and pixels which do not belong to a large connected region (the minimal size $\mathcal{T}_{min}$ of the region is defined empirically) are ignored. These connected regions which contain the remaining pixels are considered as new layer support.

6 Discussion and conclusion

6.1 Experimental results

The validation was also done on two classical sequences (calendar sequence FIG. 3 and flowers sequence FIG. 4) to permit comparisons with previous methods. One can see that the algorithm extracts well the different layers for both sequences. In FIG. 3, if the calendar and the train are well segmented, the ball is over-segmented (due to lack of texture and similar colors with the background) and is classified in the same layer than the train.

For the flowers sequence, the first frame is over-segmented but the number of layers is then well determined. The background is well distinguished from the middle-plane (the house and the flowers). Colors criterion permits to overcome ambiguities in the sky due to lack of texture. However, branches are not well classified with the good layer. Indeed, due to the small distance between the tree and the camera, as branches do not belong to the same 3D plane, their motion can not be represented with the same affine model than the one of the tree.

6.2 Conclusion

In this paper, a method to robust motion estimation and layered reconstruction of scene according to parametric motion models is presented. Our method performs robust motion estimation while being able to account for occlusions through a forward/backward iterative estimation process. Furthermore, within an forward/backward schema, our approach groups the image domain into layers according to motion, appearance and spatial and temporal smoothness constraints. Promising experimental results demonstrate the potential of the proposed method as shown in [FIG. 3 and 4].

Computational complexity is the most important limitation of the proposed approach. In particular the motion estimation step is time consuming and can lead to sub-optimal results. Hardware implementation of the method in Graphics Processing Units is under consideration. The sequential nature of the proposed approach is also a limitation. To this end, one can consider a combinatorial approach where the parameters of the affine models are also recovered through an α-expansion algorithm.

Acknowledgments

The authors would like to thank Olivier Juan, for providing the EM algorithm used for the estimation of the Gaussian mixture model parameters and Yuri Boykov from the University of Western Ontarion for fruitful discussions.

References

1. Wang, J., Adelson, E.: Representing Moving Images with Layers. IEEE Transactions on Image Processing **3** (1994) 625–638
2. Bergen, J.R., Anandan, P., Hanna, K.J., Hingorani, R.: Hierarchical model-based motion estimation. In: ECCV '92: Proceedings of the Second European Conference on Computer Vision, Springer-Verlag (1992) 237–252
3. Ayer, S., Sawhney, H.: Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding. In: IEEE International Conference in Computer Vision, Caibridge, USA (1995) 777–784
4. Weiss, Y.: Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In: CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), Washington, DC, USA, IEEE Computer Society (1997) 520
5. Shanon X. Ju, Michael J. Black, A.D.J.: Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In: CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96), Washington, DC, USA, IEEE Computer Society (1996) 307
6. Cremers, D., Soatto, S.: Variational space-time motion segmentation. In: International Conference on Computer Vision (ICCV). (2003) 886–893
7. Horn, B., Schunck, B.: Determinating Optical Flow. Artificial Intelligence **17** (1981) 185–203
8. Barron, J., Fleet, D., Beauchemin, S., Burkitt, T.: Performance of optical flow techniques. Computer Vision and Pattern Recognition (CVPR) (1992) 236–242
9. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: 8th European Conference on Computer Vision (ECCV). (2004) 25–36
10. Black, M.J., Anandan, P.: The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. Comput. Vis. Image Underst. **63** (1996) 75–104
11. Tikhonov, A.: Ill-Posed Problems in Natural Sciences. Coronet (1992)
12. Odobez, J.M., Bouthemy, P.: Robust multiresolution estimation of parametric motion models. Journal of Visual Communication and Image Representation **6** (1995) 348–365

13. Black, M.J., Jepson, A.D.: Estimating optical flow in segmented images using variable-order parametric models with local deformations. IEEE Trans. Pattern Anal. Mach. Intell. **18** (1996) 972–986
14. Darrell, T., Pentland, A.P.: Cooperative robust estimation using layers of support. IEEE Trans. Pattern Anal. Mach. Intell. **17** (1995) 474–487
15. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cut. In: CVPR (2). (2004) 972–979
16. Zabih, R., Kolmogorov, V.: Spatially coherent clustering using graph cuts. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). (2004) 437–444
17. Irani, M., Anandan, P.: A unified approach to moving object detection in 2d and 3d scenes. IEEE Trans. Pattern Anal. Mach. Intell. **20** (1998) 577–589
18. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. John Wiley & Sons (1973)
19. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: ICCV. (2001) 105–112
20. Besag, J.: On the statistical analysis of dirty images. Journal of Royal Statistics Society **48** (1986) 259–302
21. Chou, P., Brown, C.: The theory and practice of bayesian image labeling. International Journal of Computer Vision **4** (1990) 185–210
22. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence **6** (1984) 721–741
23. Boykov, Y., Veksler, O., Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 1222–1239

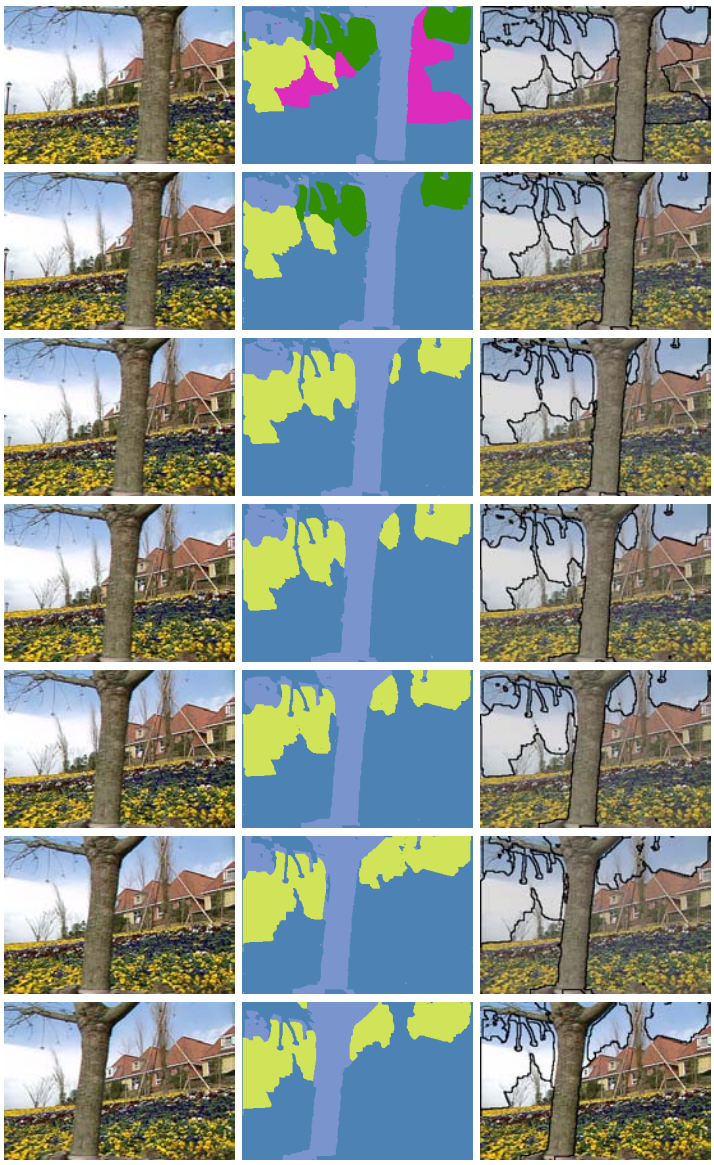

Fig. 4. Results on flowers sequence. Each column represent a frame: frames 0,3,6,...,18 are represented here. First row, original sequence; second row, the layers extracted ; third row, superposition of layers boundaries with original sequence. One can note that layers become more accurate and stay constant throughout the sequence. Main parameters: $\alpha = 0.25$, $\beta = 10$, $\gamma = 1$, $\mathcal{T}_{min} = 0.2$.