



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

*Joint motion estimation and layer segmentation in
transparent image sequences - Application to
transparent-motion-compensated noise reduction in
X-ray image sequences*

Vincent Auvray — Patrick Bouthemy — Jean Liénard

N° 6762

Novembre 2008

_____ Thèmes NUM et BIO _____

Joint motion estimation and layer segmentation in transparent image sequences - Application to transparent-motion-compensated noise reduction in X-ray image sequences

Vincent Auvray*[†], Patrick Bouthemy*, Jean Liénard[†]

Thèmes NUM et BIO — Systèmes numériques et Systèmes biologiques
Équipe-Projet Vista

Rapport de recherche n° 6762 — Novembre 2008 — 41 pages

Abstract: This paper is concerned with the estimation of the motions and the segmentation of the spatial supports of the different layers involved in transparent X-ray image sequences. Classical motion estimation methods fail on sequences involving transparent effects since they do not explicitly model this phenomenon. We propose a method that comprises three main steps: initial block-matching for two-layer transparent motion estimation, motion clustering with 3D Hough transform, and joint transparent layer segmentation and parametric motion estimation. It is validated on synthetic and real clinical X-ray image sequences.

Secondly, we derive an original transparent motion compensation method compatible with any spatio-time filtering technique. A direct transparent motion compensation method is proposed. To overcome its limitations, a novel hybrid filter is introduced which locally selects which type of motion compensation is to be carried out for optimal denoising. Convincing experiments on synthetic and real clinical images are also reported.

Key-words: X-ray, Motion estimation, Motion compensation, Noise reduction, Transparent motion estimation.

* IRISA/INRIA, Vista Team, 35042 Rennes, France

[†] General Electric Healthcare, Advanced Medical Application Team, 78533 Buc, France

Estimation et segmentation jointe des mouvements transparents - Application au débruitage compensé en mouvement transparents de séquences d'images en rayons X

Résumé : Ce document porte sur l'estimation des mouvements et la segmentation des supports spatiaux des différentes couches impliquées dans les séquences d'images transparentes par rayons X. Les méthodes d'estimation de mouvement classiques échouent sur les séquences contenant des effets de transparence puisqu'elles ne modélisent pas explicitement ce phénomène. Nous proposons une méthode qui comporte trois étapes majeures: une mise en correspondance par blocs initiale pour l'estimation locale de deux mouvements transparents, une extraction des mouvements avec une transformée de Hough 3D, et une estimation de mouvement paramétrique jointe à une segmentation en couches. La méthode est validée sur des séquences cliniques par rayons X synthétiques et réelles.

D'autre part, nous introduisons une méthode originale de compensation des mouvements qui est compatible avec toute méthode de filtrage spatio-temporel. Une méthode de compensation directe des mouvements transparents est proposée. Pour dépasser ses limitations, nous introduisons un filtre hybride qui choisit localement quel type de compensation de mouvement doit être effectuée pour aboutir à un débruitage optimal. Nous présentons des résultats convaincants obtenus sur images synthétiques et réelles.

Mots-clés : Rayons X, estimation de mouvement, compensation de mouvement, réduction de bruit, estimation de mouvement transparent

1 Introduction

Most image sequence processing and analysis tasks require an accurate computation of image motion. However, classical motion estimation methods fail in the case of image sequences involving transparent layers. Situations of transparency arise in videos for instance when an object is reflected in a surface, or when an object lies behind a translucent one. Transparency may also be involved in special effects in movies such as the representation of phantoms as transparent beings. Finally, let us mention progressive transition effects such as *dissolve*, often used in video editing. Some of these situations are illustrated on Fig.1.

In this paper, we are particularly concerned with the transparency phenomenon occurring in X-ray image sequences (even if the developed techniques can also be successfully applied to video sequences [1]). Since the radiation is successively attenuated by different organs, the resulting image is ruled by a multiplicative transparency law (that is turned into an additive one by a log operator). For instance, the heart can be seen over the spine, the ribs and the lungs on Fig.3.

When additive transparency is involved, the grayvalues of the different ob-

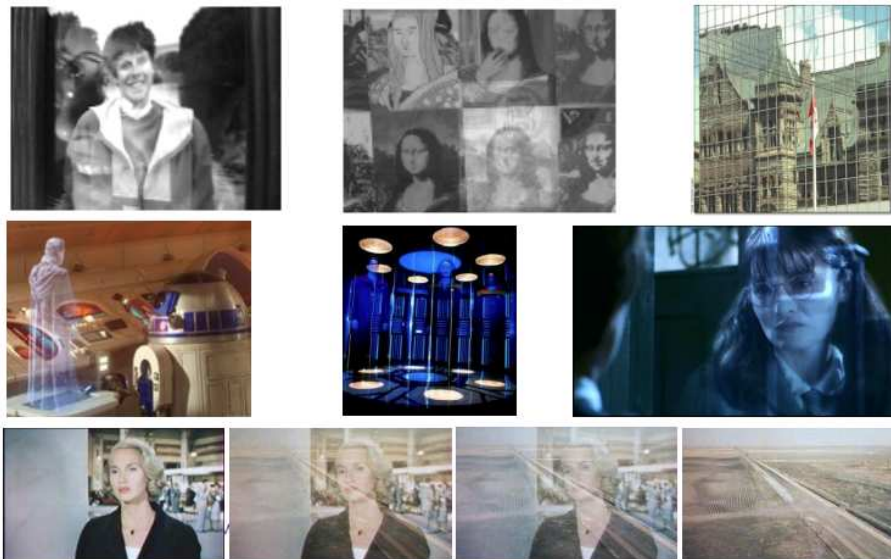


Figure 1: Examples of transparency configuration in videos. Different reflections are shown in the top row, three examples of *phantom* effects in the middle row, and one example of a dissolve effect for a gradual shot change in the bottom row.

jects superimpose and the brightness constancy of points along their image trajectories, exploited for motion estimation [2], is no longer valid. Moreover,

two different motion vectors may exist at the same spatial position. Therefore, motion estimation methods that explicitly tackle the transparency issue have to be developed.

In this paper, we deal both with transparent motion estimation and spatial segmentation of the transparent layers in the images. We mean that we aim at recovering both the motion and the spatial support of each transparent layer. Transparent layer segmentation is an original topic to be distinguished from the transparent layer separation task: a spatial segmentation aims at directly delimiting the spatial support of the different transparent objects based on their motions, whereas a separation framework [3, 4, 5] leads to recover the grayvalue images of the different transparent objects. The latter can be handled so far in restricted situations only (e.g., specific motion must be assumed for at least one layer, the image globally includes only two layers), while we consider any type of motions and any number of layers. We aim at defining a general and robust method since we will apply it to noisy and low-contrasted X-ray image sequences.

We do not assume that the number of transparent layers in the image is known or limited. In contrast, we will determine it. We only assume a local two-layer configuration, i.e., the image can be divided into regions where at most two transparent layers are simultaneously present. We will call such a situation *bi-distributed transparency*. This is not a strong assumption since this is the most commonly encountered configuration in real image sequences.

Finally, we derive from the proposed transparent motion estimation method a general transparent motion *compensation* method compatible with any spatio-time filtering technique. In particular, we propose a novel method for the temporal filtering of X-ray image sequences that avoids the appearance of severe artifacts (such as blurring), while taking advantage of the large temporal redundancy involved by the high acquisition frame rate.

The remainder of the paper is organized as follows. Section 2 includes a state-of-the art on transparent motion estimation and introduces the fundamental transparent motion constraint. In Section 3, we recall the specificity of X-ray image sequences and we formulate a simple but realistic image formation model. It will be exploited to generate image sequences with ground truth.

In Section 4, we present and discuss the different assumptions involved in the motion estimation problem statement. Section 5 details the MRF-based framework that we propose, while Section 6 deals with the practical development of our joint transparent motion estimation and spatial layer segmentation method. In Section 7, we present the proposed filtering method, involving a novel transparent motion compensation procedure. We report in Section 8 experimental results for transparent motion *estimation* on realistic test images as well as on numerous real clinical image sequences. Section 9 presents *denoising* results on realistic test images and real clinical image

sequences. Finally, Section 10 contains concluding remarks and possible extensions.

2 Related work on transparent motion estimation

A first category of transparent motion estimation method attempts to directly extend usual motion estimation strategies to the transparency case [6, 7]. Approaches that are particularly robust to deviations from the brightness assumption are adopted, but the weak point is that transparency is not explicitly taken into account. The method [8] focuses on the problem of transparent motion estimation in angiograms to improve stenosis quantification accuracy. The motion fields are iteratively estimated by maximizing a phase correlation metric after removing the (estimated) contribution of the previously processed layer. However, it leads to interesting results only when one layer dominates the other one (which is not necessarily the case in interventional X-ray images).

Among the methods which explicitly tackle the transparency issue in the motion estimation process, we can distinguish two main classes of approaches. The first one works in the frequency domain [9, 10, 11], but it must be assumed that the motions are constant over a large time interval (dozen of frames). These methods are therefore unapplicable to image sequences involving time-varying movements, such as cardiac motions.

The second class of methods formulates the problem in the spatial image domain using the fundamental Transparent Motion Constraint (TMC) introduced by Shizawa and Mase [12], or its discrete version developed in [13]. The latter states that, if one considers the image sequence I as the addition of two layers I_1 and I_2 ($I = I_1 + I_2$), respectively moving with velocity fields $\mathbf{w}_1 = (u_1, v_1)$ and $\mathbf{w}_2 = (u_2, v_2)$, the following holds:

$$\begin{aligned} r(x, y, \mathbf{w}_1, \mathbf{w}_2) &= I(x + u_1 + u_2, y + v_1 + v_2, t - 1) + I(x, y, t + 1) \\ &\quad - I(x + u_1, y + v_1, t) - I(x + u_2, y + v_2, t) = 0 \end{aligned} \quad (1)$$

where (x, y) are the coordinates of point \mathbf{p} in the image. For sake of clarity, we do not make explicit that \mathbf{w}_1 and \mathbf{w}_2 may depend on the image position. Expression (1) implicitly assumes that \mathbf{w}_1 and \mathbf{w}_2 are constant over time interval $[t - 1, t + 1]$. Even if the hypothesis of constant velocity can be problematic at a few specific time instants of the heart cycle, Eq.1 offers us with a reasonable and effective Transparent Motion Constraint (TMC) since the temporal velocity variations are usually smooth. This constraint can be extended to n layers by considering $n + 1$ images while extending the motion invariance assumption accordingly [13].

To compute the velocity fields using the TMC given by (1), we a global function J is usually minimized:

$$J(\mathbf{w}_1, \mathbf{w}_2) = \sum_{(x,y) \in \mathfrak{S}} r(x, y, \mathbf{w}_1(x, y), \mathbf{w}_2(x, y))^2 \quad (2)$$

where $r(x, y, \mathbf{w}_1(x, y), \mathbf{w}_2(x, y))$ is given by Eq.1 and \mathfrak{S} denotes the image grid.

Several methods have been proposed to minimize expression (2), making different assumptions on the motions. The more flexible the hypothesis, the more accurate the estimation, but also the more complex the algorithm. A compromise must be reached between measurement accuracy on one hand and robustness to noise, computational load and sensitivity to parameter tuning on the other hand.

Dense velocity fields are computed in [14] by adding a regularization term to (2), and in [15] by resorting to a Markovian formalism. It enables to estimate non-translational motions at the cost of higher sensitivity to noise and of high algorithm complexity. In contrast, stronger assumptions on the velocity fields are introduced in [16] and [17] by considering that \mathbf{w}_1 and \mathbf{w}_2 are constant on blocks of the image, which allows fast but less accurate motion estimation. In [13], the velocity fields are decomposed on a B-spline basis, so that this method can account for complex motions, while remaining relatively tractable. However, the structure of the basis has to be carefully adapted to particular situations and the computational load becomes high if fine measurement accuracy is needed.

3 X-ray imaging properties and simulation scheme

In this section, we first address the transparency issue in the X-ray image formation process. The properties of X-ray imaging are explored with a view to realistic simulation of X-ray images, which will provide us with ground truth data for performance evaluation.

3.1 Transparency

X-rays are attenuated in various ways depending on the materials they are going through. An attenuation coefficient μ_{mat} , typical of each material, intervenes in the Φ_{out} flow of monochromatic X photons coming out of an object of thickness d radiated with the Φ_{in} input flow:

$$\Phi_{out} = \Phi_{in} e^{-\mu_{mat} d} \quad (3)$$

Assuming that tissues have a constant attenuation coefficients, the radiation flow out of n tissues of thicknesses d_i and attenuation coefficients μ_i is given

by:

$$\Phi_{out} = \Phi_{in} \prod_{i=1}^n e^{-\mu_i d_i} \propto \prod_{i=1}^n e^{-\mu_i d_i} \quad (4)$$

The global attenuation being the product of the attenuations of each organ, we have to consider a *multiplicative transparency*. It turns into an additive one by applying a log operator. As a result, the measured image I is the superimposition of n sub-images I_i (the *layers*) undergoing their own motion. At pixel \mathbf{p} and time t , we have:

$$I(\mathbf{p}, t) = \sum_{i=1}^n I_i(\mathbf{p}, t) \quad (5)$$

It is actually difficult to give an exact definition of the concept of *layer*. It is tempting to assign a layer to each organ (one layer for the heart, one for the diaphragm, one for the spine, etc). It is however more appropriate for our purpose to consider two organs undergoing the same motion or coupled motions as forming one single layer (for instance, the heart and the tissues of the lungs that it carries along). Conversely, we will divide an organ into two layers if necessary (for instance, the walls of the heart when they have two different apparent motions due to the acquisition angulation). Formally, we will define a layer as any physical unit having a coherent motion under the imaging angulation. Let us point out that the layer specification is dependent on how we define *coherent motion*. As explained in Section 4, it will result from the choice of the 2D parametric motion model.

3.2 Image Formation Model

3.2.1 Image formation process

In order to generate realistic test images, we need to investigate the X-ray formation process [18, 19]. The X photons produced by the generator do not follow an exact straight line from the tube focal spot to the detector, they respect statistical laws implying possibilities of deviation. This can be modeled with a Poisson *quantum noise* corrupting the image, that can finally be considered as of constant standard deviation after applying a square-root operator [20].

Moreover, part of the radiation absorbed by the anatomy is *scattered* in an isotropic way. Even if this effect is limited by anti-scatter grids, it causes a “haze” on the images that can be modeled by the addition of a low-pass version of the original image. Finally, the detector has a Modulation Transfer Function (MTF), due the scintillator that slightly blurs the signal, and due to the finite dimension of the photoreceptor cells. It has been measured for the considered imaging system. The detector also produces a low electronic noise.

3.2.2 Image simulation

To quantitatively assess the performance of our motion estimation and compensation method, we aim at generating realistic (short) image sequences supplying known ground-truth in terms of layers and motions. The simulation proceeds in two steps: we first generate attenuation maps from real high-dose clinical images, and then we combine them under known simulated conditions.

To achieve the first step, we use *images from real exams* (Fig.2) in order to ensure a content representative for X-ray images. We select them among high-dose exams to be able to consider them as noise-free. We inverse the encoding and the MTF transforms, convert the resulting graylevel images into the input radiation on the detector and roughly compensate for the scatter. The procedure to realize the latter step is to subtract 20% of a 64×64 boxcar low-passed version of the radiation image. The resulting radiation image is proportional to the anatomy attenuation map corresponding to the initial image content.

Once two such attenuation maps have been built from two different initial images, we move them by known displacements to generate (multiplicatively) a realistic two-layer anatomy configuration in motion. We finally simulate short (three-image) sequences under known conditions, including layer motion, quantum noise, scatter, electronic noise and MTF (subsection 3.2.1).

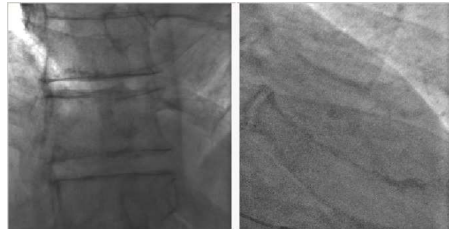


Figure 2: The two images from high-dose exams used to form the two moving layers in the realistic sequence generation. On the left, an image from an abdominal exam. On the right, an image from a cardiac exam.

4 Transparent motion estimation problem statement

We consider the general problem of motion estimation in *bi-distributed transparency*. It refers to transparent configurations including any number of layers globally, but at most two locally. This new concept, which suffices to

handle any transparent image sequence in practice, is discussed in subsection 4.1.

To handle this problem, we resort to a joint segmentation and motion estimation framework. Because of transparency, we need to introduce a specific segmentation mechanism that allows distinct regions to superimpose, and to derive an original transparent joint segmentation and motion estimation framework.

Finally, to allow for a reasonably fast and robust method (able to handle X-Ray images), we consider transparencies involving parametric motion models as explained in subsection 4.2.

4.1 Bi-distributed transparency

We do not impose any limitation on the number of transparent layers *globally* involved in the image. Nevertheless, we assume that the images contain at most two layers at every spatial position \mathbf{p} , which is acceptable since three layers rarely superimpose in real transparent image sequences. We will refer to this configuration as the *bi-distributed transparency*.

Even in the full transparency case encountered in X-ray exams, where acquired images result from cumulative absorption by X-ray tissues, the image can be nearly always divided into regions including at most two moving transparent layers, as illustrated on Fig.3. The only region involving three layers in this example is insignificant since the three corresponding organs are homogeneous in this area.

Unlike existing methods, we aim at *explicitly* extracting the segmentation of the image in its transparent layers, which is an interesting and exploitable output *per se* and is also required for the motion-estimation stage based on the two-layer TMC.

4.2 Transparent motion constraint with parametric models

We decide to represent the velocity field of each layer by a 2D polynomial model. Such a parametric motion model accounts for a large range of motions, while involving few parameters for each layer. We believe that affine motion models offer a proper compromise since they can describe a large category of motions (translation, rotation, divergence, shear), while keeping the model simple enough to handle the transparency issue in a fast and tractable way. Our method could consider higher-order polynomial models as well, such as quadratic ones, if needed. Let us point out that in case of a more complex motion, the method is able to over-segment the corresponding layer in regions having a motion compatible with the considered type of parametric model. Complex transparent motions can still be accurately estimated at the cost of oversegmentation.

The velocity vector at point (x, y) for layer k is now defined by $\mathbf{w}_{\theta_k}(x, y) =$

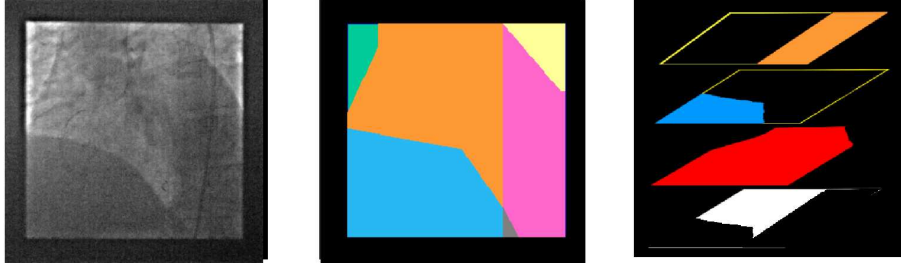


Figure 3: One image of X-ray exam yielding a situation of bi-distributed transparency (on the left). In the middle, the segmentation of the image in its different regions: three two-layer regions (the orange region corresponds to the “heart and lungs”, the light blue one to “heart and diaphragm” and the pink one to “heart and spine”), two single-layer regions (the lungs in green and spine in yellow), and a small three-layer region (“heart and diaphragm and spine” in grey). On the right, its spatial segmentation into four transparent layers (i.e., their spatial supports, spine in orange, diaphragm in blue, heart in red and lungs in white). By definition, the spatial supports of the transparent layers overlap. The colors have been independently chosen in these two maps.

$(u_{\theta_k}(x, y), v_{\theta_k}(x, y))$:

$$u_{\theta_k}(x, y) = a_{1,k} + a_{2,k}x + a_{3,k}y \quad \text{and} \quad v_{\theta_k}(x, y) = a_{4,k} + a_{5,k}x + a_{6,k}y. \quad (6)$$

We then introduce a new version of the TMC (Eq.1) that we call the Parametric Transparent Motion Constraint (PTMC):

$$r(x, y, \mathbf{w}_{\theta_1}, \mathbf{w}_{\theta_2}) = I(x + u_{\theta_1} + u_{\theta_2}, y + v_{\theta_1} + v_{\theta_2}, t - 1) + I(x, y, t + 1) - I(x + u_{\theta_1}, y + v_{\theta_1}, t) - I(x + u_{\theta_2}, y + v_{\theta_2}, t) = 0 \quad (7)$$

with \mathbf{w}_{θ_1} and \mathbf{w}_{θ_2} given in (6).

The next section introduces the MRF-based framework that concludes the problem statement.

5 MRF-based framework

5.1 Observations and remarks

We have to segment the image into regions including at most two layers to estimate the motion models associated to the layers from the PTMC (7). Conversely, the image segmentation will rely on the estimation of the different transparent motions. Therefore, we have designed a joint segmentation

and estimation framework based on a Markov Random Field (MRF) modeling. A *joint* approach is more reliable than a sequential one (as in [21]) since estimated motions can be improved using a proper segmentation and vice versa.

Joint motion estimation and segmentation frameworks have been developed for “classical” image sequences [22][23][24][25][26][27], but have never been studied in the case of transparent images. In particular, we have to introduce a novel segmentation allowing regions to superimpose. Moreover, the bi-distributed assumption implies to control the number of layers simultaneously present at a given spatial location.

The proposed method will result in an alternate minimization scheme between segmentation and estimation stages. To maintain a reasonable computational load, the segmentation is carried out at the level of blocks. Typically, the 256×256 images are divided in 32×32 blocks (for a total number of blocks $S = 64$). We will see in subsection 6.2 that this block structure will also be exploited in the initialization step. According to the targeted application, the block size could be fixed smaller in a second phase of the algorithm. The pixel-level could even be progressively reached, if needed.

The blocks are taken as the sites s of the MRF model (Fig.4). We aim at labeling the blocks s according to the pair of transparent layers they are belonging to. Let $e = \{e(s), s = 1, \dots, S\}$ denote the label field with $e(s) = (e_1(s), e_2(s))$, where $e_1(s)$ and $e_2(s)$ designate the two layers present at site s . $e_1(s)$ and $e_2(s)$ are given the same value when the site s involves one layer only. The spatial supports of the transparent layers can be straightforwardly inferred from the labeling of the two-layer regions (i.e., from the elements of each pair that forms the label).

Let us assume that the image comprises a total of K transparent layers, where K is to be determined. To each layer is attached an affine motion model of parameters θ_k (six parameters). Let $\Theta = \{\theta_k, k = 1, \dots, K\}$.

5.2 Global energy functional

We need to estimate the segmentation defined by the labels $e(s)$, and the corresponding transparent motions defined by the parameters Θ . The estimates will minimize the following global energy functional:

$$\begin{aligned}
 F(e, \Theta) &= \sum_{s \in S} \sum_{(x,y) \in s} \rho[r(x, y, \theta_{e_1(s)}, \theta_{e_2(s)})] - \mu \eta[s, e(s)] \\
 &+ \mu \sum_{\langle s,t \rangle} (1 - \delta(e_1(s), e_1(t)))(1 - \delta(e_1(s), e_2(t))) \\
 &+ (1 - \delta(e_2(s), e_1(t)))(1 - \delta(e_2(s), e_2(t))). \quad (8)
 \end{aligned}$$

The first term of Eq.8 is the data-driven term based on the PTMC defined in Eq.7. Instead of a quadratic constraint, we resort to a robust function

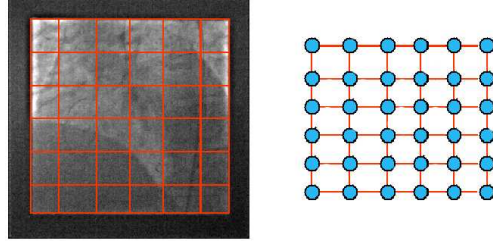


Figure 4: MRF framework. *On the left*, a processed image divided in blocks (36 blocks for the sake of clarity of the figure. In fact, this 1024×1024 image would be divided in 1024 blocks of 32×32 pixels, which would allow for an accurate segmentation). *On the right*, the graph associated with the introduced Markov model. The sites are plotted in blue and their neighbouring relations are drawn in orange.

$\rho(\cdot)$ in order to discard outliers, i.e., points where the PTMC is not valid [28]. We consider the Tukey function as robust estimator. It is defined by:

$$\rho(r, C) = \begin{cases} \frac{r^6}{6} - \frac{C^2 r^4}{2} + \frac{C^4 r^2}{2} & \text{if } |r| < C \\ \frac{C^6}{6} & \text{else} \end{cases} \quad (9)$$

It depends on a scale parameter C which defines the threshold above which the corresponding point will be considered as an outlier. To be able to handle any kind of images, we will determine C on-line as explained in Section 6.3.

The additional functional $\eta(\cdot)$ is introduced in Eq.8 to detect single layer configurations. It is a binary function which is 1 when s is likely to be a single layer site. It will be discussed in subsection 5.3.

The last term of the global energy functional $F(e, \Theta)$ enforces the segmentation map to be reasonably smooth. We have to consider the four possible label transitions between two sites (involving two labels each). $\delta(\cdot, \cdot)$ is equal to 1 if the two considered elements are the same and equals 0 otherwise.

The μ parameter weights the relative influence of the two terms. In other words, a penalty μ is added when introducing between two sites a region border involving a change in one layer only, and a penalty 2μ when both layers are different. A transition between a mono-layer site s_1 and a bi-layer site s_2 will also imply a penalty μ (as long as the layer present in s_1 is also present in s_2). μ is determined in a content-adaptive way, as explained in Section 6.3.

5.3 Detection of a single layer configuration

Over single layer regions, Eq.1 is satisfied provided one of the two estimated velocities (for instance $\mathbf{w}_{\theta_{\mathbf{e}_1(s)}}$) is close to the real motion of this single

layer whatever the value of the other velocity ($\mathbf{w}_{\theta_{e_2(s)}}$). The minimization of Eq.8 without the $\eta(\cdot)$ term would therefore not allow to detect single layer regions because a “imaginary” second layer would be introduced over these sites. Thus, we propose an original criterion to detect these areas.

We define the residual value:

$$\nu(\hat{\theta}_{e_1(s)}, \theta_{e_2(s)}, s) = \sum_{(x,y) \in s} r(x, y, \hat{\theta}_{e_1(s)}, \theta_{e_2(s)})^2 \quad (10)$$

If it varies only slightly for different values of $\theta_{e_2(s)}$ (while keeping $\theta_{e_1(s)}$ constant and equal to its estimate $\hat{\theta}_{e_1(s)}$), it is likely that the block s contains one single layer only, corresponding to $e_1(s)$. $\eta(\cdot)$ would be set to 1 in this case to favour the label $(e_1(s), e_1(s))$ over this site (and to 0 in the other cases).

Formally, to detect a single layer corresponding to $\theta_{e_1(s)}$, we compute the mean value $\bar{\nu}(\hat{\theta}_{e_1(s)}, s)$ of the residual $\nu(\theta_{e_1(s)}, \cdot, s)$ by applying n motions (defined by θ_j , $j = 1, \dots, n$.) to the second layer. We want to decide if $\bar{\nu}(\hat{\theta}_{e_1(s)}, s)$ is significantly different from the minimal residual on this block, $\nu(\hat{\theta}_{e_1^*(s)}, \hat{\theta}_{e_2^*(s)}, s)$, where $(e_1^*(s), e_2^*(s))$ are the current labels at site s . This minimal residual is in practice coming from the iterative minimization of Eq.8 presented in Section 6.1.

To meet this decision independently of the image texture, we first compute a representative value for the residual of the image, given by:

$$\nu_{med} = \mathit{med}_{s \in S} \nu(\hat{\theta}_{e_1^*(s)}, \hat{\theta}_{e_2^*(s)}, s), \quad (11)$$

and its median deviation

$$\Delta \nu_{med} = \mathit{med}_{s \in S} \left| \nu(\hat{\theta}_{e_1^*(s)}, \hat{\theta}_{e_2^*(s)}, s) - \nu_{med} \right|. \quad (12)$$

(This assumes that the motion models have been well estimated and the current labeling is correct on at least half the sites). Then, we set

$$\eta(s, e_1(s), e_2(s)) = 1 \quad \text{if} \quad \left| \bar{\nu}(\theta_{e_1^*(s)}, s) - \nu(\theta_{e_1^*(s)}, \theta_{e_2^*(s)}, s) \right| < \alpha \Delta \nu_{med} \quad (13)$$

$$\text{and } e_1(s) = e_2(s), \quad (13)$$

$$\eta(s, e_1(s), e_2(s)) = 0 \quad \text{otherwise}, \quad (14)$$

where η is the functional introduced in Eq.8. This way, we favour the single layer label $(e_1(s), e_1(s))$ at site s when the condition (13) is satisfied. The same process is repeated to test for $\theta_{e_2(s)}$ as the motion parameters of a (possible) single layer. In practice, we fix $\alpha = 2$.

6 Joint parametric motion estimation and segmentation of transparent layers

This section describes the minimization of the energy functional (8) along with its initialization. We also explain how the parameters are set on-line, and how the number of layers globally present is estimated.

The overall joint transparent motion estimation and layer segmentation algorithm is summarized in Algorithm 1.

6.1 Minimization of the energy functional F

The energy functional F defined in (8) is minimized iteratively. When the motion parameters are fixed, we use the ICM technique to update the labels of the blocks: the sites are visited randomly, and for each site the label that minimizes the energy functional (8) is selected.

Once the labels are fixed, we have to minimize the first term of Eq.8, which involves a robust estimation. It can be solved using an IRLS (Iteratively Reweighted Least Square) technique which leads to minimize the equivalent functional [29]:

$$F_1(\Theta) = \sum_{s \in S} \sum_{(x,y) \in s} \alpha(x,y) r(x,y, \theta_{e_1(s)}, \theta_{e_2(s)})^2 \quad (15)$$

where $\alpha(x,y)$ denotes the weights. Their expression at the iteration j of the minimization is given by:

$$\alpha^j(x,y) = \frac{\rho'(r(x,y, \hat{\theta}_{e_1(s)}^{j-1}, \hat{\theta}_{e_2(s)}^{j-1}))}{2r(x,y, \hat{\theta}_{e_1(s)}^{j-1}, \hat{\theta}_{e_2(s)}^{j-1})} \quad (16)$$

with $\hat{\theta}^{j-1}$ the estimate of θ . computed at iteration $j-1$, and ρ' the derivative of ρ .

Even if each PTMC involves two models only, their addition over the entire image allows us to simultaneously estimate the K motion models globally present in the image by minimizing the functional $F_1(\Theta)$ of Eq.15 (which is defined in a space of dimension $6K$). If the velocity magnitudes were small, we could consider a linearized version of expression (15) (i.e., by relying on a linearized version of the expression r). Since large motions can occur in practice, we introduce a multiresolution incremental scheme exploiting Gaussian pyramids of the three consecutive images. At its coarsest level L , motions are small enough to resort to a linearized version of functional $F_1(\Theta)$ (Eq.15). The minimization is then achieved using the conjugate gradient algorithm. Hence, first estimates of the motions parameters are provided, they are denoted $\hat{\theta}_k^L$, $k = 1, \dots, K$.

At the level $L-1$, we initialize θ_i^{L-1} with $\tilde{\theta}_i^{L-1}$, where $\tilde{a}_{i,k}^{L-1} = 2\hat{a}_{i,k}^L$ ($i = 1, 4$)

Algorithm 1 Joint transparent motion estimation and layer segmentation algorithm.

- **Initialization**

- Transparent two-layer block-matching.
- 3D Hough-transform applied to the computed pairs of displacements (simplified affine models). Each vote is assigned a confidence value related to the texture of the block and the reliability of the computed displacements.
- First determination of the global number of transparent layers and initialization of the affine motion models by extraction of the relevant peaks of the accumulation matrix.
- Layer segmentation initialization (using the maximum likelihood criterion).

Iteratively,

- **Robust affine motion model estimation when the labels are fixed**

Energy minimization using the IRLS technique. Multi-resolution incremental Gauss-Newton scheme.

- **Label field determination (segmentation) once the affine motion parameter are fixed**

Energy minimization using the ICM technique. Criterion (13) is evaluated to detect single layer configurations in each block S .

- Update of the number of layers (merge process).

Finally,

- Introduction of a new layer if a given number of blocks verify relation (24). The overall algorithm is reiterated in this case.

and $\tilde{a}_{i,l}^{L-1} = \hat{a}_{i,l}^L$ ($l = 2, 3, 5, 6$). We then write $\theta_k^{L-1} = \tilde{\theta}_k^{L-1} + \Delta\theta_k^{L-1}$, and we minimize $F_1(\Theta)$ with respect to the $\Delta\theta_k^{L-1}$, $k = 1, \dots, K$, once r is linearized around the $\tilde{\theta}_k^{L-1}$, using the IRLS technique. This Gauss-Newton method, iterated through the successive resolution levels until the finest one, allows us to simultaneously estimate the affine motion models of the K transparent layers.

6.2 Initialization of the overall scheme

Such an alternate iterative minimization scheme converges if properly initialized. To initialize the motion estimation stage, we resort to a transparent block-matching technique that tests every possible pair of displacements in a given range [17]. More specifically, for each block s , we compute

$$\zeta(\mathbf{w}_1, \mathbf{w}_2, s) = \sum_{(x,y) \in s} r(x, y, \mathbf{w}_1, \mathbf{w}_2)^2 \quad (17)$$

for a set of possible displacements $\mathbf{w}_1 \times \mathbf{w}_2$, where r is given by Eq.1. The pair of displacements $(\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2)$ is the one which minimizes Eq.17. This scheme is applied on a multiresolution representation of the images to reduce the computation time (it would be higher than in the case of non-transparent motions since the explored space is of dimension 4).

To extract the underlying layer motion models from the set of computed pairs of displacements, we apply the Hough transform on a three-dimension parameter space (i.e., a simplified affine motion model):

$$\begin{cases} u = a_1 + a_2x \\ v = a_4 + a_2y \end{cases} \quad (18)$$

Indeed, restricting the Hough space to a 3D space obviously limits the computational complexity and improves the transform efficiency, while being sufficient to determine the number of layers and to initialize their motion models. Each displacement $\mathbf{w} = (u, v)$ votes for the parameters:

$$\begin{cases} a_1 = a_2x - u \\ a_4 = a_2y - v \end{cases} \quad (19)$$

, defining a straight line. The Hough space has to be discretized in its three directions. Practically, we have chosen a one pixel step for the translational dimensions a_1 and a_4 , and for the divergence term a_2 a step corresponding to a one pixel displacement in the center of the image. An example of computed Hough accumulation matrix is given on Fig.5.

If the layers include large homogeneous areas (which is the case in X-ray images), the initial block-matching is likely to produce a relatively large number of erroneous displacement estimates. To improve further the initialization stage, we adopt a continuous increment mechanism of the accumulation matrix based on a confidence value depending on the block texture.

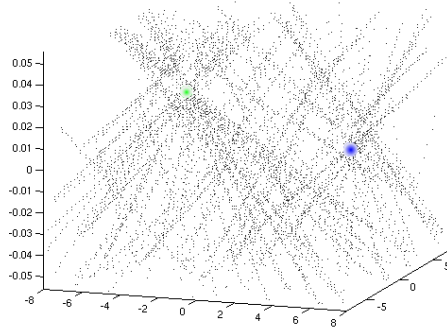


Figure 5: Accumulation matrix in the space (a_1, a_4, a_2) , built from the displacements computed by a transparent block-matching technique. These displacements are presented on the left of Fig.6. The ground truth of the two motion models present in the image sequences are plotted in green and blue.

To compute the confidence value associated to a block s and a displacement \mathbf{w}_1 (the other displacement being fixed to $\hat{\mathbf{w}}_2$), we analysis the behavior of $\zeta(\cdot, \hat{\mathbf{w}}_2, s)$. If it remains close to its minimal value $\zeta(\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, s)$, then the layer associated to \mathbf{w}_1 is homogeneous and $\hat{\mathbf{w}}_1$ should be assigned a low confidence value. Conversely, if $\zeta(\cdot, \hat{\mathbf{w}}_2, s)$ has a clear minimum in $\hat{\mathbf{w}}_1$, the corresponding layer must be textured, and $\hat{\mathbf{w}}_1$ can be considered as reliable. More precisely, we compute in each block s :

$$c_1(s) = \left| \frac{1}{n} \sum_{\Delta \mathbf{w}} \zeta(\hat{\mathbf{w}}_1 + \Delta \mathbf{w}, \hat{\mathbf{w}}_2, s) - \zeta(\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, s) \right|, \quad (20)$$

$$c_2(s) = \left| \frac{1}{n} \sum_{\Delta \mathbf{w}} \zeta(\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2 + \Delta \mathbf{w}, s) - \zeta(\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, s) \right| \quad (21)$$

where n is the number of tested displacements $\Delta \mathbf{w}$. To normalize these coefficients, we compute their first quartile \tilde{c} over the image, and then assign to each block s and computed displacement $\hat{\mathbf{w}}_i$ ($i = 1, 2$) the value $c_i(s)/\tilde{c}$ (or 1 if $c_i(s)/\tilde{c} > 1$). Then, the 25% more reliable computed displacements are assigned the value 1, whereas those that are less informative, or which are not correctly computed, are given a small confidence value.

The Hough transform allows us to cluster the reliable displacement vectors. We successively look for the dominant peaks in the accumulation matrix, and we decide that the corresponding motion models are relevant if they “originate” from at least five computed displacements that have not been considered so far. Conversely, a displacement computed by the transparent block-match technique is considered as “explained” by a given motion model

if it is close enough from the mean velocity induced by this motion model over the considered block (in practice, distant by less than two pixels). This method yields a first evaluation of the number of layers K and an initialization of the affine motion models. Then, the label field is initialized by minimizing the first term of Eq.8 only (i.e., we consider a maximum likelihood criterion). Figure 6 illustrates the initialization stage.

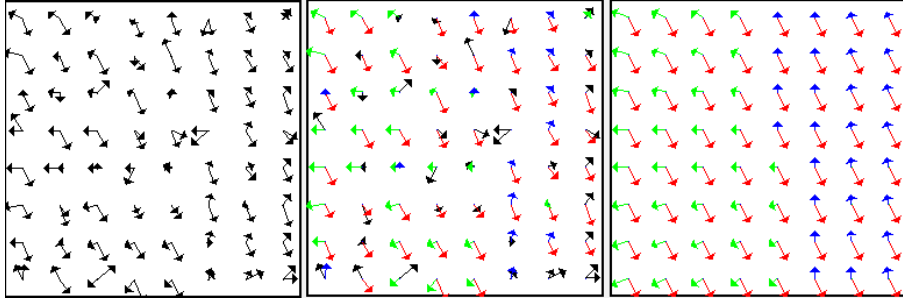


Figure 6: Example of the initialization stage for a symbolic example. *On the left*, the displacements computed by the transparent block-matching. *In the middle*, the velocity fields corresponding to the affine models extracted by the Hough transform. Three layers are detected; they are plotted in red, green and blue. The erroneous displacements are plotted in black. *On the right*, the true displacements.

6.3 Content adaptive parameter setting

Two parameters have to be set for the functional F (Eq.8) to be defined: the scale parameter C of the robust functional, and the parameter μ weighting the relative influence of the data-driven and the smoothing term. C is determined as follows:

$$\begin{cases} \bar{r} &= \operatorname{med}_{\mathbf{p} \in \mathfrak{S}} r(\mathbf{p}, \theta_1, \theta_2) \\ \overline{\Delta r} &= 1.48 \times \operatorname{med}_{\mathbf{p} \in \mathfrak{S}} |r(\mathbf{p}, \theta_1, \theta_2) - \bar{r}| \\ C &= 2.795 \times \overline{\Delta r} \end{cases} \quad (22)$$

when \mathbf{p} is a pixel position, and \mathfrak{S} refers to the image grid.

The use of the medians allows to evaluate representative values \bar{r} and $\overline{\Delta r}$ of the “mean” and “deviation” residual values without being disturbed by the outliers. The factor 1.48 enables to unbiased the estimator of $\overline{\Delta r}$, and the factor 2.795 has been proposed by Tukey to correctly estimate C [30].

The μ parameter is determined in a content-adaptive way:

$$\mu = \lambda \operatorname{med}_{s \in S} \sum_{(x,y) \in s} \rho(r(x,y, \hat{\theta}_{e_1(s)}, \hat{\theta}_{e_2(s)})). \quad (23)$$

where $\hat{\theta}_i$ is the estimate of θ_i from the previous iteration of the minimization. According to the targeted application, λ can be set to favour the data-driven velocity estimates (small λ), or to favour smooth segmentation (higher λ). In practice, the value $\lambda = 0.5$ has proven to be a good trade-off between regularization and over-segmentation.

6.4 Update of the number of transparent layers

To update the number K of transparent layers, we have designed two criteria. On one hand, two layers whose motion models are too close (typically, difference of one pixel on average over the corresponding velocity fields) are merged. Furthermore, a layer attributed to less than five blocks is discarded, and the corresponding blocks relabeled. On the other hand, we propose means to add a new layer if required, based on the maps of weights generated by the robust affine motion estimation stage.

The blocks where the current labels and/or the associated estimated motion models are not coherent with every pixels they contain should include low weight values delivered by the robust estimation stage for the outlier pixels. It then becomes necessary to add a new layer if a sufficient number of blocks containing a large number of pixels with low weights are detected. More formally, we use as indicator the number of weights smaller than a given threshold. The corresponding points will be referred to as *outliers*. To learn which number of outliers per block is significant, we compute the median value of outliers N_o over the blocks, along with its median deviation ΔN_o . A block s is considered as mis-labeled if its number $N_o(s)$ of outliers verifies:

$$N_o(s) > N_o + \gamma \cdot \Delta N_o \quad (24)$$

$$\text{with } N_o = \underset{s \in S}{\text{med}} N_o(s) \quad (25)$$

$$\text{and } \Delta N_o = \underset{s \in S}{\text{med}} |N_o(s) - N_o| \quad (26)$$

In practice, we set $\gamma = 2.5$. If more than five blocks are considered as mis-labeled, we add a new layer. We estimate its motion model by estimating an affine model from the displacement vectors supplied by the initial block-matching step in these blocks (using a least-square estimation), and we run the joint segmentation and estimation scheme on the whole image again.

7 Motion-Compensated Denoising Filter for Transparent Image Sequences

In this section, we exploit the estimated transparent motions for a denoising application. To do so, we propose a way to compensate for the transparent motions, without having to *separate* the transparent layers.

7.1 Transparent Motion Compensation

7.1.1 Principle

A first way of tackling the problem of transparent motion compensation is to separate the transparent layers and compensate the individual motion of each layer, layer per layer. However, the transparent layer separation problem has been solved in very restricted conditions only [8, 5]. As a result, this cannot be applied in general situations as those encountered in medical image sequences.

Instead, we propose to globally compensate the transparent motions in the image sequence without prior layer separation. To do so, we propose to rearrange the PTMC (7) to form a *prediction* of the image \tilde{I} at time $t + 1$, based on the images at time instants $t - 1$ and t and exploiting the two estimated affine motion models $\hat{\theta}_1$ and $\hat{\theta}_2$:

$$\begin{aligned} \tilde{I}(\mathbf{p}, t + 1) &= I(\mathbf{p} + \mathbf{w}_{\hat{\theta}_1}(\mathbf{p}), t) + I(\mathbf{p} + \mathbf{w}_{\hat{\theta}_2}(\mathbf{p}), t) \\ &- I(\mathbf{p} + \mathbf{w}_{\hat{\theta}_1}(\mathbf{p}) + \mathbf{w}_{\hat{\theta}_2}(\mathbf{p}), t - 1) \end{aligned} \quad (27)$$

Eq.27 allows us to globally compensate for the transparent image motions. It enables to handle X-ray images that satisfy the bi-distributed transparency hypothesis i.e., involving *locally* two layers, without limiting the total number of layers *globally* present in the image.

Any denoising temporal filter can be made transparent-motion-compensated by considering, instead of past images, transparent-motion-compensated images \tilde{I} given by Eq.27. As a consequence, details can be preserved in the images, and no blurring introduced if the transparent motions are correctly estimated.

However, relation (27) implies an increase of the noise level of the predicted image since three previous images are added. The variance of the noise corrupting \tilde{I} is the sum of the noise variances of the three considered images. This has adverse effects as demonstrated in the next subsection, if a simple temporal filter is considered.

7.1.2 Limitation

Transparent motion compensation can be added to any spatio-temporal filter. We will illustrate its limitation in the case of a pure temporal filter. More precisely, we consider the following temporal recursive filter [31]:

$$\hat{I}(\mathbf{p}, t + 1) = (1 - c(\mathbf{p}, t + 1))I(\mathbf{p}, t + 1) + c(\mathbf{p}, t + 1)\tilde{I}(\mathbf{p}, t + 1) \quad (28)$$

where $\hat{I}(\mathbf{p}, t + 1)$ is the output of the filter, i.e. the denoised image, $\tilde{I}(\mathbf{p}, t + 1)$ is the predicted image and $c(\mathbf{p}, t + 1)$ the filter weight. This simple temporal filter is frequently used since its implementation is straightforward and its behavior well-known. Spatial filtering tends to introduce correlated effects

that are quite disturbing for the observer (especially when medical image sequences are played at high frame rates). This filter is usually applied in an adaptative way to account for incorrect prediction, which can be evaluated by the expression $|I(\mathbf{p}, t + 1) - \tilde{I}(\mathbf{p}, t + 1)|$. More specifically, the gain is defined as a decreasing function of the prediction error.

To illustrate the intrinsic limitation of such a transparent-motion compensated filter, we study its behavior under ideal conditions: the transparent motions are known as well as the level of noise in the different images. Furthermore, we ignore the low-pass effect of interpolations. The noise variances $\sigma_{\hat{I}}^2(t + 1)$, $\sigma_{\tilde{I}}^2(t + 1)$ and σ^2 (constant in time) of the images $\hat{I}(t + 1)$, $\tilde{I}(t + 1)$ and $I(t)$ respectively, are related as follows (from Eq.28):

$$\sigma_{\tilde{I}}^2(t + 1) = (1 - c(t + 1))^2 \sigma^2 + c(t + 1)^2 \sigma_{\hat{I}}^2(t + 1) \quad (29)$$

under the assumption that the different noises are independent. On the other hand, Eq.27 implies (for a recursive implementation of this filter):

$$\sigma_{\tilde{I}}^2(t + 1) = 2\sigma_{\tilde{I}}^2(t) + \sigma_{\hat{I}}^2(t - 1) \quad (30)$$

For an optimal noise filtering, one should choose $c(t + 1)$ so that $\hat{\sigma}^2(t + 1)$ is minimized:

$$c(t + 1) = \frac{2\sigma_{\hat{I}}^2(t) + \sigma_{\tilde{I}}^2(t - 1)}{2\sigma_{\hat{I}}^2(t) + \sigma_{\tilde{I}}^2(t - 1) + \sigma^2} \quad (31)$$

Eq.29 and 31 define a sequence $(\sigma_{\tilde{I}}^2(t))_{t \in N}$. We show in appendix A that it asymptotically reaches a limit:

$$\lim_{t \rightarrow \infty} \sigma_{\tilde{I}}(t) = \sqrt{\frac{2}{3}} \sigma \simeq 0.816 \sigma. \quad (32)$$

Even if we assume that the motions were known, *transparent motion-compensated recursive temporal filter can not allow for a significative denoising rate*. Similarly, even if transparent motion-compensated *spatio-temporal* filters do not exhibit the exact same behavior, they denoise less efficiently than their non-compensated counterparts.

7.2 Hybrid Filter

7.2.1 Problem statement

Transparent motion compensation allows for a better contrast preservation since it avoids blurring. However, it affects the noise reduction efficiency by increasing the noise of the predicted image. We therefore propose to exploit the transparent motion compensation when appropriate only, to offer a better tradeoff between denoising power and information preservation. We distinguish four local configurations:

- **C₀**. *Both layers are textured* around pixel \mathbf{p} . The global transparent motion compensation is needed to preserve details. The filter output will rely on $\tilde{I}(\mathbf{p}, t + 1)$ and $I(\mathbf{p}, t + 1)$ only (instead of $I(\mathbf{p}, t)$ and $I(\mathbf{p}, t + 1)$ for the case without motion compensation).
- **C₁**. *The first layer only is textured* around pixel \mathbf{p} . We will just perform the motion compensation of this layer but still applied to the compound intensity. The filter will then exploit $I(\mathbf{p}, t + 1)$, $\hat{I}(\mathbf{p} + \mathbf{w}_{\hat{\theta}_1}(\mathbf{p}), t)$ and $\tilde{I}(\mathbf{p}, t + 1)$ (which still carries pertinent information here, but will be assigned a small weight because of its noise level):

$$\begin{aligned} \hat{I}(\mathbf{p}, t + 1) &= \alpha(\mathbf{p}, t + 1)I(\mathbf{p}, t + 1) + \beta(\mathbf{p}, t + 1)\hat{I}(\mathbf{p} + \mathbf{w}_{\hat{\theta}_1}(\mathbf{p}), t) \\ &+ (1 - \alpha(\mathbf{p}, t + 1) - \beta(\mathbf{p}, t + 1))\tilde{I}(\mathbf{p}, t + 1) \end{aligned} \quad (33)$$

As in 7.1.2, explicit expressions can be computed for the optimal weights (see Tab.1 for their expression in the case of a temporal hybrid filter).

- **C₂**. *The second layer only is textured* around pixel \mathbf{p} . We use a combination of $I(\mathbf{p}, t + 1)$, $\hat{I}(\mathbf{p} + \mathbf{w}_{\hat{\theta}_2}(\mathbf{p}), t)$ and $\tilde{I}(\mathbf{p}, t + 1)$.
- **C₃**. *Both layers are homogeneous* around pixel \mathbf{p} . The four intensities can be used: $I(\mathbf{p}, t + 1)$, $\hat{I}(\mathbf{p} + \mathbf{w}_{\hat{\theta}_1}(\mathbf{p}), t)$, $\hat{I}(\mathbf{p} + \mathbf{w}_{\hat{\theta}_2}(\mathbf{p}), t)$ and $\tilde{I}(\mathbf{p}, t + 1)$.

A fifth configuration is added w.r.t. the motion estimation output.

- **C₄**. *The motion estimates are erroneous*. In this case, we duplicate $I(\mathbf{p}, t + 1)$ only. This fifth configuration makes the hybrid filter adaptive, in the sense that it will keep displaying coherent images even if erroneous motion estimates are supplied.

7.2.2 Configuration selection and designed hybrid filtering

This subsection deals with the detection of the local configuration among the five listed above. Let us assume that I_1 only is textured around pixel \mathbf{p} . Then, we can write (for convenience, we will write \mathbf{w}_1 and \mathbf{w}_2 instead of $\mathbf{w}_{\hat{\theta}_1}(\mathbf{p})$ and $\mathbf{w}_{\hat{\theta}_2}(\mathbf{p})$):

$$\begin{aligned} I(\mathbf{p}, t + 1) &= I_1(\mathbf{p}, t + 1) + I_2(\mathbf{p}, t + 1) = I_1(\mathbf{p} + \mathbf{w}_1, t) + I_2(\mathbf{p} + \mathbf{w}_2, t) \\ &\simeq I_1(\mathbf{p} + \mathbf{w}_1, t) + I_2(\mathbf{p} + \mathbf{w}_1, t) \simeq I(\mathbf{p} + \mathbf{w}_1, t) \end{aligned} \quad (35)$$

We have exploited in (35) the local contrastless of the layer I_2 . As a result, we can compare $I(\mathbf{p}, t + 1)$ and $I(\mathbf{p} + \mathbf{w}_1, t)$ to decide whether I_2 is uniform around \mathbf{p} . To do so, we have to establish if these two values differ only because of the presence of noise, or if they actually correspond to different physical points. This is precisely the problem handled by adaptive filters.

We resort to the same mechanism. Rather than adopting a binary decision to select one given configuration \mathbf{C}_i , that would be visually disastrous since neighboring pixels would be processed differently,

- we first compute for each pixel \mathbf{p} two factors: $f_1(\mathbf{p})$ associated to “the layer 1 is uniform” and $f_2(\mathbf{p})$ associated to “the layer 2 is uniform”. They are defined as decreasing functions of $|I(\mathbf{p}, t+1) - I(\mathbf{p} + \mathbf{w}_2, t)|$ (resp. $|I(\mathbf{p}, t+1) - I(\mathbf{p} + \mathbf{w}_1, t)|$). A third factor $f_{12}(\mathbf{p})$ is associated to “ $\tilde{I}(\mathbf{p}, t+1)$ is a good prediction of $I(\mathbf{p}, t+1)$ ”. It is a decreasing function of $|I(\mathbf{p}, t+1) - \tilde{I}(\mathbf{p}, t)|$. This enables to associate each configuration (\mathbf{C}_i) , $i = 0..4$, an appropriate weighting factor, as shown in Eq.36.
- we filter the image using relation (37) by considering in turn each configuration \mathbf{C}_i , $i = 0..4$, and we get the output images $\hat{I}_{(\mathbf{C}_i)}(\mathbf{p}, t)$.
- we combine linearly these five output images as follows to yield the final denoised image:

$$\begin{aligned}
\hat{I}(\mathbf{p}, t) &= f_{12}(\mathbf{p})(1 - f_1(\mathbf{p}))(1 - f_2(\mathbf{p}))\hat{I}_{(\mathbf{C}_0)}(\mathbf{p}, t) \\
&+ f_{12}(\mathbf{p})(1 - f_1(\mathbf{p}))f_2(\mathbf{p})\hat{I}_{(\mathbf{C}_1)}(\mathbf{p}, t) \\
&+ f_{12}(\mathbf{p})f_1(\mathbf{p})(1 - f_2(\mathbf{p}))\hat{I}_{(\mathbf{C}_2)}(\mathbf{p}, t) \\
&+ f_{12}(\mathbf{p})f_1(\mathbf{p})f_2(\mathbf{p})\hat{I}_{(\mathbf{C}_3)}(\mathbf{p}, t) \\
&+ (1 - f_{12}(\mathbf{p}))\hat{I}_{(\mathbf{C}_4)}(\mathbf{p}, t)
\end{aligned} \tag{36}$$

To summarize, the overall scheme comprises two modules:

- The first one filters the images based on different (transparent or non-transparent) motion compensation schemes (subsection 7.2.1).
- The second module locally weights the five intermediate images according to the probability of the considered configuration (subsection 7.2.2).

7.2.3 Temporal hybrid filter

In the case of a purely temporal hybrid filter, the expression for a given configuration is defined by:

$$\begin{aligned}
\hat{I}(\mathbf{p}, t+1) &= \alpha(\mathbf{p}, t)I(\mathbf{p}, t+1) + \beta(\mathbf{p}, t)\hat{I}(\mathbf{p} + \mathbf{w}_1, t) \\
&+ \delta(\mathbf{p}, t)\hat{I}(\mathbf{p} + \mathbf{w}_2, t) + \gamma(\mathbf{p}, t)\tilde{I}(\mathbf{p}, t+1),
\end{aligned} \tag{37}$$

where α , β , δ and γ are filter weights locally specified. $\beta = 0$ and $\delta = 0$ for \mathbf{C}_0 ; $\delta = 0$ for \mathbf{C}_1 ; $\beta = 0$ for \mathbf{C}_2 ; $\beta = 0$, $\delta = 0$ and $\gamma = 0$ for \mathbf{C}_4 . When the noise level of the input images involved in (37) is known or estimated, one can analytically set the other weights for an optimal filtering (Tab.1).

Table 1: Optimal filter weights for the five possible configurations. The noise standard deviation noise of the acquired image is denoted σ , the one of the previous denoised image $\sigma_{\hat{I}}$ and the one of the predicted image $\sigma_{\tilde{I}}$.

Configuration	α	β
(C ₀)	$\frac{\sigma_{\tilde{I}}^2}{\sigma^2 + \sigma_{\tilde{I}}^2}$	0
(C ₁)	$\frac{\sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2}{\sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2 + \sigma^2 \sigma_{\tilde{I}}^2 + \sigma_{\tilde{I}}^2 \sigma^2}$	$\frac{\sigma^2 \sigma_{\tilde{I}}^2}{\sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2 + \sigma^2 \sigma_{\tilde{I}}^2 + \sigma_{\tilde{I}}^2 \sigma^2}$
(C ₂)	$\frac{\sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2}{\sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2 + \sigma^2 \sigma_{\tilde{I}}^2 + \sigma_{\tilde{I}}^2 \sigma^2}$	0
(C ₃)	$\frac{\sigma_{\tilde{I}}^4 \sigma_{\hat{I}}^2}{\sigma_{\tilde{I}}^4 \sigma_{\hat{I}}^2 + \sigma^2 \sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2 + \sigma_{\tilde{I}}^2 \sigma^2 \sigma_{\hat{I}}^2 + \sigma_{\tilde{I}}^4 \sigma^2}$	$\frac{\sigma^2 \sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2}{\sigma_{\tilde{I}}^4 \sigma_{\hat{I}}^2 + \sigma^2 \sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2 + \sigma_{\tilde{I}}^2 \sigma^2 \sigma_{\hat{I}}^2 + \sigma_{\tilde{I}}^4 \sigma^2}$
(C ₄)	1	0

Configuration	δ	γ
(C ₀)	0	$\frac{\sigma^2}{\sigma^2 + \sigma_{\tilde{I}}^2}$
(C ₁)	0	$\frac{\sigma_{\tilde{I}}^2 \sigma^2}{\sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2 + \sigma^2 \sigma_{\tilde{I}}^2 + \sigma_{\tilde{I}}^2 \sigma^2}$
(C ₂)	$\frac{\sigma^2 \sigma_{\tilde{I}}^2}{\sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2 + \sigma^2 \sigma_{\tilde{I}}^2 + \sigma_{\tilde{I}}^2 \sigma^2}$	$\frac{\sigma_{\tilde{I}}^2 \sigma^2}{\sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2 + \sigma^2 \sigma_{\tilde{I}}^2 + \sigma_{\tilde{I}}^2 \sigma^2}$
(C ₃)	$\frac{\sigma_{\tilde{I}}^2 \sigma^2 \sigma_{\hat{I}}^2}{\sigma_{\tilde{I}}^4 \sigma_{\hat{I}}^2 + \sigma^2 \sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2 + \sigma_{\tilde{I}}^2 \sigma^2 \sigma_{\hat{I}}^2 + \sigma_{\tilde{I}}^4 \sigma^2}$	$\frac{\sigma_{\tilde{I}}^4 \sigma^2}{\sigma_{\tilde{I}}^4 \sigma_{\hat{I}}^2 + \sigma^2 \sigma_{\tilde{I}}^2 \sigma_{\hat{I}}^2 + \sigma_{\tilde{I}}^2 \sigma^2 \sigma_{\hat{I}}^2 + \sigma_{\tilde{I}}^4 \sigma^2}$
(C ₄)	0	0

8 Transparent motion estimation results

8.1 Transparent Motion Estimation Results on Realistic Generated Image Sequences

We have tested our transparent motion estimation scheme on realistic image sequences generated as described in sub-section 3.2.2. It supplies a meaningful quantitative assessment of the performance of our method under realistic conditions. It also allows us to compare the performance of different settings of our algorithm in order to choose the optimal one¹.

In this subsection, we focus on images containing two layers only, each one spread over the full image. It is indeed difficult to simultaneously assess the quality of the motion segmentation and of the layer segmentation². The overall performance of the global method is discussed over real experiments in subsection 8.2.

More specifically, we have applied our method on 250 three-frame sequences, the first layer (abdomen image) undergoing a translation and the second layer (heart image) an affine motion. To generate the affine motion of the second layer, we proceed in two steps. First, we randomly choose the two translational and the scaling (denoted h) parameters so that the resulting displacement magnitude lies in the range of -8 to 8 pixels. Then, we convert the obtained transformation into a set of affine motion models by allowing the two pairs of affine parameters a_2, a_6 on one hand, and a_3, a_5 on the other hand, to vary from their reference value (respectively h and 0), in a range of respectively $h \pm 0.2h$ and $\pm 0.2h$. Consequently, the generated motions are similar to anatomic motions, while not perfectly following the model assumed by the Hough transform in the initialization step. The two generated motions are also required to sufficiently differ from each other, i.e., from 2 pixels in average over the image grid (An observer would not perceive two distinct transparent layers otherwise!)

We have considered image sequences representative of diagnostic (high dose) and fluoroscopic (low dose) exams (with a noise of standard deviation $\sigma = 10$ (SNR: 34 dB) and $\sigma = 20$ (SNR: 28 dB) respectively), at different scatter rates (a real typical value being 20%). The images are coded on 12 bits, and their mean value is typically 500. Running the overall framework takes about 30 seconds for 288×288 images on a Pentium IV (2.4 GHz and 1 Go). The global estimation error is formally estimated below (Eq.38).

¹Each parameter is either computed online (in particular the crucial ones like the C parameter of the Tukey function (9), or the μ parameter of the MRF function (8)), or set once for all based on tests on the realistic generated image sequence (preprocessing, interpolation type, Block-Matching search range, accumulation matrix structure...). The method is therefore fully automatic.

²An erroneous segmentation that mislabels one block will dramatically impact the global estimation error (38), even if the considered block is low textured and little informative. The residual error would be a better error metric, yet it is much less intuitive.

Table 2: Performance evaluation of the proposed method for different noise levels and scatter rates: average, standard deviation and median value (in pixels) of the global error computed over 250 generated image sequences

Metric on the global error	Noise level	Scatter rate		
		0%	20%	50%
Mean	$\sigma = 10$	0.22	0.27	0.36
	$\sigma = 20$	0.53	0.82	1.67
Standard deviation	$\sigma = 10$	0.63	0.66	0.70
	$\sigma = 20$	0.78	0.93	1.65
Median	$\sigma = 10$	0.08	0.10	0.14
	$\sigma = 20$	0.25	0.41	1.00

Table 2 contains the the mean value (in pixels) of the global estimation error ϵ computed from 250 tests, as well as its standard deviation and its median value:

$$\epsilon = \frac{1}{|\mathfrak{S}|} \sum_{\mathbf{p} \in \mathfrak{S}} \sqrt{\|\mathbf{w}_{\theta_1^{true}}(\mathbf{p}) - \mathbf{w}_{\hat{\theta}_1}(\mathbf{p})\|^2} + \sqrt{\|\mathbf{w}_{\theta_2^{true}}(\mathbf{p}) - \mathbf{w}_{\hat{\theta}_2}(\mathbf{p})\|^2} \quad (38)$$

where $\mathbf{w}_{\theta_i^{true}}$ (resp. $\mathbf{w}_{\hat{\theta}_i}$) refers to the velocity vectors (given by the true (resp. estimated) models). We can observe that very satisfactory results are obtained. The average error raises to 0.36 pixels only for the most difficult diagnostic case. For comparison, the best method from the state of the art [8] reached a precision of about 2 pixels on similar data (involving quadratic motion models though). The estimation accuracy remains very good on the difficult fluoroscopic image sequences ($\sigma = 20$), where subpixel precision is maintained if the scatter rate is not too high. In this last case (50% scatter rate), the motion estimation remains interesting but is less accurate. The other indicators demonstrate the repeatability of the method over the different experiments.

As for every method based on equation (1), our framework assumes temporal motion constancy over two successive time intervals. This hypothesis may be critical for clinical image sequences at some time instants of the heart cycle. To test the violation of this assumption, we have carried out the following experiment.

We have randomly chosen two affine models (θ_1^1, θ_2^1) as explained above, and applied them between time instants $t - 1$ and t . We have then computed a second transparent motion field (θ_1^2, θ_2^2) , allowing each coefficient to vary from 10, 20 or 30% around (θ_1^1, θ_2^1) , and applied it between time instants t and $t + 1$. This way, a sequence of three images with temporal motion

Table 3: Average of the global estimation error for different noise levels and different temporal motion variations (with 20% scatter rate).

Noise level	Temporal motion variation			
	0%	10%	20%	30%
$\sigma = 10$	0.27	0.32	0.45	0.59
$\sigma = 20$	0.81	1.00	0.92	1.07

variation is generated. We have evaluated the global errors between the estimated motion field and (θ_1^1, θ_2^1) on one hand, and (θ_1^2, θ_2^2) on the other hand. We report its mean value computed over 250 generated sequences in Tab.3.

We can note a *progressive* degradation of the estimation accuracy with the amount of temporal motion change. Then, it is not that critical that the temporal motion constancy over two successive time intervals is not strictly verified. The transparent motion estimation for fluoroscopic images remains accurate, even if the two successive motions vary in a range of 30%.

8.2 Transparent Motion Estimation Results on Real Clinical Image Sequences

The previous experiments are useful to study the behaviour of the proposed method, to fix the options and the parameters, and to quantitatively compare it to other motion estimation methods. However, it does not validate the relevance of the two-layer model per se, since the generated images themselves rely on this model. In this section, we present results obtained on real image sequences that demonstrate the bi-transparency model validity.

We present motion estimation results out of three real clinical image sequences and one video. We display several frames along with the estimated motion fields in Fig.7-9, at some interesting time instants of the sequences. The velocity fields are plotted with colored vectors, whose length is twice the real displacement magnitude for sake of visibility.

The motion estimation quality is evaluated by visual inspection since no ground truth is available, and since the resulting displaced image differences are difficult to interpret due to the lack of contrast. Anyway, the reliability of the estimated motions is objectively demonstrated by the convincing results of transparent-motion-compensated denoising given in Section 9.

The image of Fig.7 corresponds to an area of about $5cm \times 5cm$ size, located between the heart (dark mass on the right) and the lungs (bright tissues). It also contains a static background where ribs are visible. In the considered region, the heart carries the lungs tissues along, so that they have the same apparent motion. The motions of the two layers are correctly estimated: the

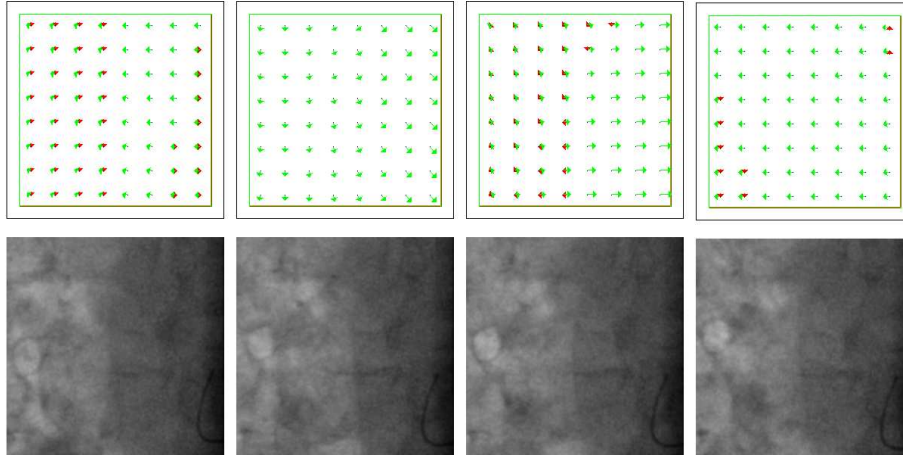


Figure 7: Four estimated two-layer motion fields, along with the corresponding fluoroscopic image at four different time instants. One layer corresponds to the static background (ribs) and its estimated affine model is plotted in red. The other layer involves the heart and the lung tissues and its estimated affine model is plotted in green. First and third instants are within in the diastole phase, second and fourth ones in the systole phase.

red arrow field corresponds to the static background (it is not plotted when it is exactly equal to 0), and the green one to the estimated affine model for the layer formed by the pair “heart and lungs”. Its motion is coherent with the observation, both during the diastole (first and third presented images) and the systole (second and last images).

The sequence shown on Fig.8 a-c) is a cardiac interventional image sequence. It globally involves three distinct transparent layers:

- The static background, which includes the contrasted surgical clips.
- The set “diaphragm and lungs”. The diaphragm is the dark mass in the bottom left corner of the image, and the lungs form the bright tissues in the other half of the image. Their motions are close, so that they can be considered as forming a single moving layer.
- The heart is also visible, even if its layer is less textured: it is the convex light-grey area on the right of the image. It can be easily seen on the first displayed image. A catheter (interventional tube) is inserted in one of its coronary, which has an independent motion.

We first report results obtained at a time instant where the three layers are static (Fig.8a-Fig.8d). Only one region is detected, which is correct: our method is still effective when no transparent motion is involved.

At a second time instant, the group “diaphragm and lungs” is still static.

The velocity field supplied by the corresponding estimated motion model is plotted in red and the estimated motion of the heart in green (Fig.8e). Both motion models are correctly estimated. Interestingly, the movement of the catheter in the upper part of the image is properly segmented as well, even if it forms a thin structure in a noisy image sequence.

The image content of the third considered time instant (Fig.8f) is also of interest, since the three layers are now undergoing independent motions. In this borderline situation (since we have three moving layers in some blocks), the method again proves to perform well: it manages to focus on the two dominant layers in the different regions. As a result, the red velocity field corresponds to the static background layer, the green one to the lungs layer, and the blue one to the heart motion layer.

The sequence presented on Fig.9 a-c) is a cardiac interventional image

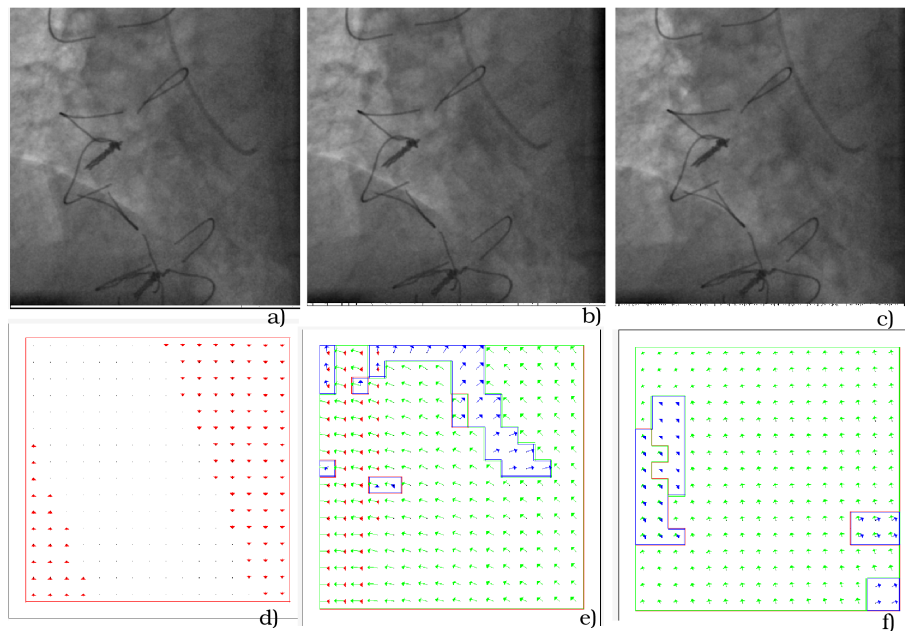


Figure 8: Second example of a X-ray interventional cardiac image sequence, a-c: Images acquired at three different time instants, d,e,f: the corresponding velocity fields supplied by the estimated affine motion models, plotted in different colours according to the transparent layer they are belonging to. On the left, illustration of the method ability to detect single layer situations; in the middle, correct segmentation and estimation of the motions of small objects, even included in noisy images; on the right, handling of a transparency situation with three simultaneous transparent layers in some areas (see main text).

sequence. It depicts an about $5\text{cm} \times 5\text{cm}$ area of the anatomy, where the

heart (dark mass filling three quarters of the image, nearly static under the considered acquisition angulation) superimposes on the lungs (bright tissues in the upper right of the image). We give results for three distant instants of this sequence. The velocity fields plotted on Fig.9 d-f) are supplied by the affine motion models estimated at the three considered time instants.

A *global* two-layer transparency correctly explains the observed motions at the first time instant (Fig.9d). The green velocity vectors correspond to the group “lungs and diaphragm”, animated by the breathing, and the red field refers to the transparent layer of the heart (it is present all over the image but is not plotted when it is perfectly null). Let us also point out that the static background is merged with the heart layer.

It is necessary to introduce a bi-distributed transparency configuration to explain the motions observed at the second considered time instant (Fig.9e). The red velocity field still refers to the (almost) static background, which now includes the mass of the heart and the diaphragm (motionless at this time). The blue velocity field corresponds to the upward motion of breathing carrying along the lungs. The green velocity field accounts for a supplementary layer corresponding to the set of coronary arteries taken as a whole, whose motion becomes perceptible. It is properly handled and correctly estimated. This demonstrates the ability of the method to focus on the two dominating motions even in situations of three-layer transparency (here, static layer, lungs layer and coronary arteries layer).

The last reported result (Fig.9f) highlights the performance of the method when situations of high complexity are encountered. All the different motions are indeed correctly estimated (by observing the sequence) even if over-segmentation is noticeable. Let us mention that a less fragmented spatial segmentation could be obtained by increasing the value of the regularization factor λ (Eq.23), but at the cost of a less accurate match between estimated motion models and observed motions. The trade-off has to be met according to the targeted application.

Finally, Fig.10 reports experiments conducted on a sequence extracted from a movie, picturing a couple reflected on an apartment window. To our knowledge, it is the first time a real transparent video sequence is processed (we mean a sequence which has not been constructed for that purpose). The reflection superimposes to a panning view of the city behind. The camera is undergoing a smooth rotation, making the reflected faces and the city undergo two apparent translations with different velocities in the image. At some time instant, the real face of a character appears in the foreground but does not affect our method because of its robustness. The obtained segmentation and motion estimation are satisfying.

More results on video sequences can be found in [1].

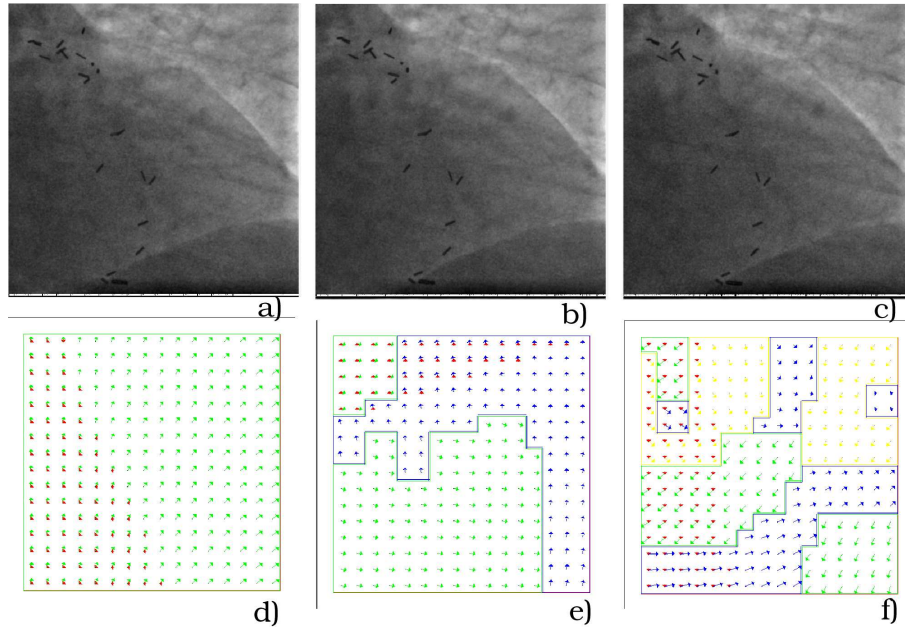


Figure 9: Example of a X-ray interventional cardiac image sequence, a-c: image acquired at three different time instants, d,e,f: the corresponding velocity fields supplied by the estimated affine motion models, plotted in different colours according to the segmented layer they are belonging to. The image on the left presents an example of global bi-transparency; the second one illustrates the ability of the method to handle dominant motions in case of three simultaneous transparent layers in some areas; the third one refers to a complex configuration in which a trade-off has to be met between accurate motion estimates and clean segmentation maps.

9 Denoising results

We have tested the proposed denoising method in the case of purely *temporal* filters because of their practical interest (explained in subsection 7.1.2). Three denoising filters are compared: the adaptive recursive filter [31] without motion compensation (ANMCR) acting as a reference, the transparent-motion-compensated recursive filter (MCR) described in subsection 7.1, and the proposed hybrid recursive filter (HR) developed in subsection 7.2. The MCR and HR filters exploit transparent motions estimated by the method of Section 6.

The adaptive function of the ANMCR and MCR filters, taking into account the relation between filter gain and prediction error, is pictured on Fig.11. It has been designed heuristically to provide efficient noise reduction without introducing artifacts. It has three parts, defined by two thresholds ($s_1 = \sigma$



Figure 10: Example of a movie depicting two people reflected on an apartment window. From left to right and top to bottom: the first frame of the sequence; one of the three images corresponding to the reported results later in the sequence; the obtained segmentation into the transparent layer supports (the green polygonal line in the middle roughly encloses the reflected people); the velocity fields supplied by the estimated affine motion models; displaced frame differences computed by compensating the motion of one of the two layers.

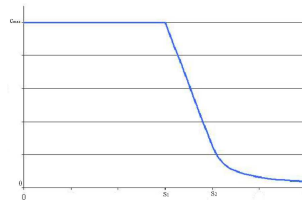


Figure 11: Decreasing function used as adaptive function in the different filters. It has three parts: a constant one for small prediction errors, a linear one in a transition area, and a vanishing one for large prediction errors.

and $s_2 = 2\sigma$ in practice): a constant part for the low prediction errors (where the coefficient is set to the optimal value for noise reduction c_{max}), a linear decreasing one in a transition area, and a vanishing one for large prediction errors. We have specified the three factors f_1 , f_2 and f_{12} of the hybrid filter in a similar way. c_{max} is set to 1, s_1 to 1.5σ and s_2 to 2σ for that filter.

9.1 Transparent-Motion-Compensated Denoising Results on Realistic Generated Image Sequences

We have tested the proposed denoising method on realistic synthetic image sequences simulating the X-ray imaging process and the transparency phenomenon (subsection 8.1). The obtained image sequence is corrupted by a

Table 4: Normalized residual noise evolution given by the rate $\hat{\sigma}(t)/\sigma$ for a realistic synthetic image sequence typical of X-ray exams, processed by the adaptive temporal filter without motion compensation (ANMC), with transparent motion compensation (MC) and by the proposed hybrid filter (HR).

t	2	3	4	5	6	7	8
ANMCR	0.71	0.69	0.66	0.63	0.60	0.59	0.58
MCR	0.87	0.82	0.79	0.79	0.78	0.78	0.78
HR	0.76	0.66	0.60	0.57	0.56	0.55	0.54

strong noise typical of fluoroscopic exams ($\sigma = 20$).

Tab.4 contains the evolution of the residual noise level of the filtered images. The transparent motion compensated filter soon reaches a denoising limit, as predicted by the theory. The hybrid filter performs slightly better than the ANMCR filter, as far as residual noise level is concerned.

The residual noise maps are given in Fig.12. They show that the hybrid filter preserves better the image details, and that the MCR filter also outperforms the ANMCR filter. However, the residual noise is much more perceptible in the case of MCR filter than for the other two filters.

Combining the different performance criteria, we can claim that the HR filter appears as the best choice among the three filters for that set of experiments.

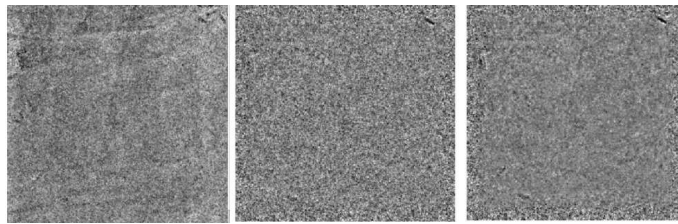


Figure 12: Residual noise of the eighth image of the generated sequence respectively obtained with the ANMCR filter, the MCR filter and the proposed HR filter (see main text).

9.2 Transparent-Motion-Compensated Denoising Results on Real Clinical Images

It is difficult to illustrate denoising results by means of static printed images, when the considered images are meant to be observed dynamically on a specific screen in a dark cath-lab. However, the major interest of our

framework being its ability to improve the quality of real interventional images, we present three typical denoising results in this subsection.

Since the MCR performs noticeably worse than the two other filters, we will compare the performance of the ANMCR and HR filters only. The images processed with the former will be presented on the right of the figures, and those with the latter on the left, at different time instants. Both are heuristically parameterized to provide a visually equivalent global denoising effect, so that the difference of performance will be mainly assessed based on the quality of contrast preservation and on the presence of artifacts. We have drawn arrows on the figures to highlight the regions of interest (that appear immediately on a dynamic display).

Results on a cardiac fluoroscopic exam are reported on Fig.13 at different time instants. The dark mass of the heart (on the right) can be distinguished from the bright tissues of the lungs (on the left). These two organs are superimposed to the background, where spine disks can be seen. The comparison of the output images obtained with the HR filter (on the left) and the ANMCR filter (on the right) reveals a much better contrast preservation of the heart with the HR filter (even if the printed figures do not give the immediate improvement impression that an observer has in ideal observation conditions). This is confirmed by the observation of the lungs.

The second image sequence (Fig.14) corresponds to a cardiac exam where

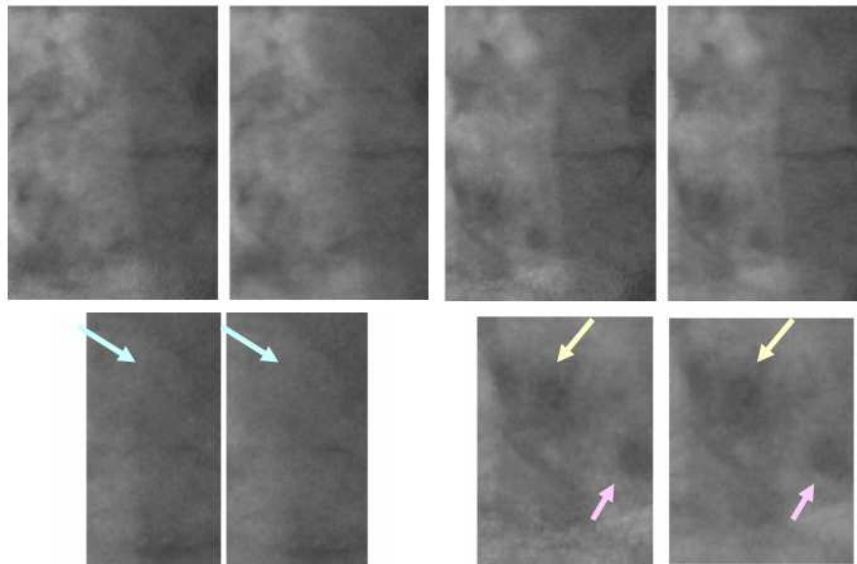


Figure 13: Top: Two time instants of a fluoroscopic sequence processed with the HR (on the left part) and the ANMCR filter (on the right), Bottom: one detail of each image. The first one highlights the better cardiac border contrast, and the second the better lungs detail preservation.

the catheter motion has been correctly handled by the transparent motion estimation module. We indeed observe that the catheter is more contrasted on the images processed by the HR filter than the ANMCR filter.

The last experiment exhibits the “noise tail” artifact induced by the AN-

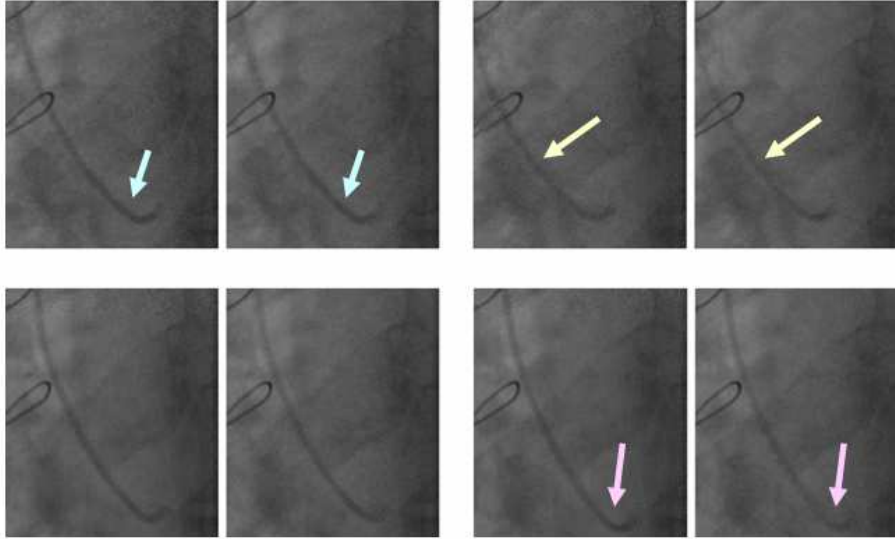


Figure 14: Four time instants of a fluoroscopic sequence processed with the HR (on the left part) and the ANMCR filter (on the right). We observe a better contrast preservation of the catheter with the hybrid filter.

MCR filter. When a moving textured object is detected by this filter, the corresponding area is kept without filtering in the output image. As a result, a region of the output image is more noisy than its neighborhood, which can be disturbing. In this situation, the hybrid filter is able to denoise the whole image, and thus does not introduce such artifacts. This phenomenon is pictured on Fig.15. We have added on the right of the figure a zoom on the region of interest. We observe that the curve corresponding to the moving border of the heart remains corrupted on the image denoised with the ANMCR filter. This artifact disappears on the image processed by the HR filter.

10 Conclusion

We have defined an original and efficient method for estimating transparent motions in video sequences, which also delivers an explicit segmentation of the image into the spatial supports of the transparent layers. It has proven to be robust to noise, and to be computationally acceptable. We assume that the images can be divided into regions containing at most two moving

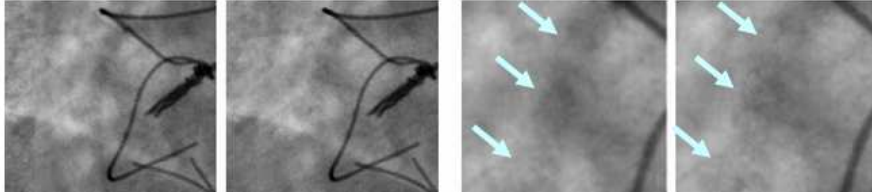


Figure 15: Fluoroscopic sequence processed with the HR (on the left part) and the ANMCR filter (on the right). The two images on the right correspond to a zoom on the region of interest of the two images on the left.

transparent layers (we call this configuration *bi-distributed transparency*), and that the layer motions can be adequately represented by $2D$ parametric models (in practice, affine models). The proposed method involves three main steps: initial block-matching for two-layer transparent motion estimation, motion clustering with a 3D Hough transform, and joint transparent layer segmentation and parametric motion estimation. The number of transparent layers is also determined on-line. The last step is solved by the iterative minimization of a MRF-based energy functional. The segmentation process is completed by a mechanism detecting regions containing one single layer.

Experiments on realistic generated image sequences have allowed us to fix the optimal settings of this framework, and to qualitatively evaluate its performance. It turns out to be excellent on diagnostic images, and satisfactory on fluoroscopic images (with normal scattering). We have also demonstrated the quality of the transparent motion estimation on various real clinical images, as well as on one video example. Satisfactory results have been reported both for motion estimation and layer segmentation. The method is general enough to successfully handle different types of image sequences with the very same parametrization.

To the best of our knowledge, our contribution forms the first transparent motion estimation scheme that has been widely applied on X-ray image sequences. Let us note that it could be used in applications other than noise reduction. For instance, it could be exploited to compensate for the patient motion in order to provide the clinician artifact-free subtracted angiography [32]. Other medical applications include the extraction of clinically relevant measurements, the tracking of features of interest (such as interventional tools) [33] or the compression of image sequences [34].

The second main contribution is the design of an original motion compensation method which can be associated to *any* spatio-temporal noise filtering technique. A direct transparent motion estimation method based on the TMC is first presented and its limitations were studied. To bypass them, an *hybrid* motion compensation method is proposed which locally combines

or selects different options, leading to an optimal noise reduction/contrast preservation trade-off. Convincing results on realistic synthetic image sequences and on real noisy and low-contrasted X-ray image sequences have been reported

Possible extensions include the improvement of the energy minimization method (for instance, by exploiting a graph-cut technique [35]). Further speeding-up the processing can also be investigated. A temporal smoothing term could also be added to the global energy functional. Finally, the other applications that could benefit from this processing should be studied [36].

A Denoising limit of the temporal transparent motion compensated filter

Fixed points From Eq.29 and 31 it comes:

$$\sigma_i^2(t+1) = \frac{(2\sigma_i^2(t) + \sigma_i^2(t-1))^2 \sigma^2 + \sigma^4(2\sigma_i^2(t) + \sigma_i^2(t-1))}{(2\sigma_i^2(t) + \sigma_i^2(t-1) + \sigma^2)^2} \quad (39)$$

The possible limits of this series are given by the fixed points. Let us denote $\bar{\sigma}^2 = \bar{V}$. We have:

$$\bar{V} = f(\bar{V}) = \frac{9\bar{V}^2\sigma^2 + 3\sigma^4\bar{V}}{(3\bar{V} + \sigma^2)^2} \quad (40)$$

$$\bar{V}(9\bar{V}^2 - 3\sigma^2\bar{V} - 2\sigma^4) = 9\bar{V}(\bar{V} + \frac{1}{3}\sigma^2)(\bar{V} - \frac{2}{3}\sigma^2) = 0 \quad (41)$$

As a result, the three fixed points are $-\frac{1}{3}\sigma^2$, 0 and $\frac{2}{3}\sigma^2$.

The first one corresponds to a negative variance and thus makes no sense here. To decide whether the other two are attractive or repulsive, we form the function:

$$g(V) = \frac{9\sigma^2\bar{V}^2 + 3\sigma^4\bar{V}}{(3\bar{V} + \sigma^2)^2} - \bar{V} = \frac{-9\bar{V}^3 + 3\sigma^2\bar{V}^2 + 2\sigma^4\bar{V}}{(3\bar{V} + \sigma^2)^2} \quad (42)$$

More precisely, if the derivative for a fixed point is greater than 1, the corresponding point is repulsive. Otherwise, it is attractive.

$$g'(V) = \frac{dg}{dV}(V) = \frac{27V^3 - 27\sigma^2V^2 + 2\sigma^6}{(3V + \sigma^2)^3} \quad (43)$$

For the two considered fixed points:

$$g'(0) = 2 \quad \text{and} \quad g'(\frac{2}{3}\sigma^2) = 0 \quad (44)$$

Finally, even if the sequence has two fixed points, $\frac{2}{3}\sigma^2$ is the only possible finite limit.

Convergence Nevertheless, the series could diverge. We have to study how its distance to the attractive fixed point evolves. Let us consider:

$$\hat{V}_2(t) = \sigma_f^2(t) - \frac{2}{3}\sigma^2 \quad (45)$$

Two consecutive elements of this series are related as follows:

$$\hat{V}_2(t+1) = \sigma_f^2(t+1) - \frac{2}{3} = f(\sigma_f^2(t)) - \frac{2}{3}\sigma^2 \quad (46)$$

$$= f(\hat{V}_2(t) + \frac{2}{3}\sigma^2) - \frac{2}{3}\sigma^2 \quad (47)$$

$$= \frac{9\sigma^2\hat{V}_2(t)^2 + 15\sigma^4\hat{V}_2(t) + 6\sigma^6}{(3\hat{V}_2(t) + 3\sigma^2)^2} - \frac{2}{3}\sigma^2 \quad (48)$$

$$= \frac{3\sigma^2\hat{V}_2(t)^2 + 3\sigma^4\hat{V}_2(t)}{(3\hat{V}_2(t) + 3\sigma^2)^2} = \frac{\sigma^2}{3} \frac{\hat{V}_2(t)}{\hat{V}_2(t) + \sigma^2} \quad (49)$$

$$= \frac{\sigma^2}{3(\hat{V}_2(t) + \sigma^2)} \hat{V}_2(t) \quad (50)$$

The first element of the last expression is strictly smaller than 1 for $\hat{V}_2(t) > -\frac{2}{3}\sigma^2$ (that is to say for every value of the variance but the repulsive fixed point 0).

The series of variances then converges monotonically to the attractive fixed point $\frac{2}{3}\sigma^2$.

References

- [1] V. Auvray, P. Bouthemy, J. Liénard, Motion-based segmentation of transparent layers in video sequences, in: Proc. of International Workshop of Multimedia Content Representation, Classification and Security, Istanbul, Turkey, 2006.
- [2] B. Horn, B. Schunk, Determining optical flow, Artificial Intelligence 17 (1981) 185–203.
- [3] Y. Weiss, Deriving intrinsic images from image sequences, in: Proceedings International Conference on Computer Vision ICCV 2001, Vancouver, 2001, pp. 68–75.
- [4] B. Sarel, M. Irani, Separating transparent layers through layer information exchange, in: European Conference on Computer Vision (ECCV'04), Vol. 4, Prague, 2004, pp. 328–341.
- [5] B. Sarel, M. Irani, Separating transparent layers of repetitive dynamic behaviors, in: IEEE International Conference on Computer Vision (ICCV'05), Vol. 1, Beijing, 2005, pp. 26–32.

- [6] M. Black, P. Anandan, The robust estimation of multiple motions: parametric and piecewise-smooth flow field, *Computer Vision and Image Understanding* 19 (1) (1996) 57–91.
- [7] M. Irani, B. Rousso, S. Peleg, Computing occluding and transparent motions, *International Journal of Computer Vision* 12 (1) (1994) 5–16.
- [8] R. Close, C. Abbey, C. Morioka, J. Whiting, Accuracy assessment of layers decomposition using simulated angiographic image sequences, *IEEE Trans. on Medical Imaging* 20 (10) (2001) 990–998.
- [9] W. Yu, G. Sommer, K. Daniilidis, Multiple motion analysis in spatial domain or in spectral domain?, *Computer Vision and Image Understanding* 90 (2) (2003) 129–152.
- [10] P. Milanfar, Two-dimensional matched filtering for motion estimation, *IEEE Transactions on Image Processing* 8 (3) (1999) 438–444.
- [11] M. Pingault, D. Pellerin, Motion estimation of transparent objects in the frequency domain, *Signal Processing* 84 (4) (2004) 709–719.
- [12] M. Shizawa, K. Mase, Principle of superposition: A common computational framework for analysis of multiple motions, in: *IEEE Workshop on Visual Motion*, Princeton, 1991, pp. 164–172.
- [13] M. Pingault, E. Bruno, D. Pellerin, A robust multiscale B-spline function decomposition for estimating motion transparency, *IEEE Trans. on Image Processing* 12 (11) (2003) 1416–1426.
- [14] I. Stuke, T. Aach, C. Mota, E. Barth, Estimation of multiple motions: regularization and performance evaluation, in: *Image and Video Communications and Processing 2003*, SPIE, Vol. 5022, Santa Clara, 2003, pp. 75–86.
- [15] I. Stuke, T. Aach, C. Mota, E. Barth, Estimation of multiple motions using block-matching and Markov random fields, in: *SPIE Conf. Visual Communications and Image Processing, VCIP'2004*, San Jose, 2004, pp. 274–278.
- [16] M. Pingault, D. Pellerin, Optical flow constraint equation extended to transparency, in: *XI European Signal Processing Conference, EU-SIPCO'2002*, Toulouse, 2002.
- [17] I. Stuke, T. Aach, C. Mota, E. Barth, Estimation of multiple motions by block matching, in: *4th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2003)*, Luebeck, 2003, pp. 358–362.

-
- [18] A. Macovski, *Medical Imaging Systems*, 2nd Edition, Prentice-Hall, 2004.
 - [19] T. Aach, U. Schiebel, G. Spekowius, Digital image acquisition and processing in x-ray imaging, *Journal of Electronic Imaging* 8 (1999) 7–22.
 - [20] F. Anscombe, The transformation of Poisson, binomial and negative-binomial data, *Biometrika* 35 (1948) 246–254.
 - [21] J. Toro, F. Owens, R. Medina, Multiple motion estimation and segmentation in transparency, in: *Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing*, Istanbul, 2000, pp. 2087–2090.
 - [22] D. Murray, H. Buxton, Scene segmentation from visual motion using global optimization, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 9 (2) (1987) 220–228.
 - [23] P. Bouthemy, E. F. cois, Motion segmentation and qualitative dynamic scene analysis from an image sequence, *International Journal of Computer Vision* 10 (2) (1993) 157–182.
 - [24] Y. Weiss, E. Adelson, A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models, in: *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, San Francisco, 1996, pp. 321–326.
 - [25] H. Sawhney, S. Ayer, Compact representations of videos through dominant and multiple motion estimation, *IEEE Trans. Pattern Analysis and Machine Intelligence* 18 (8) (1996) 814–830.
 - [26] J.-M. Odobez, P. Bouthemy, Direct incremental model-based image motion segmentation for video analysis, *Signal Processing* 6 (2) (1998) 143–155.
 - [27] N. Paragios, R. Deriche, Geodesic active regions and level set methods for motion estimation and tracking, *Computer Vision and Image Understanding* 97 (2005) 259–282.
 - [28] P. Hubert, *Robust Statistics*, John Wiley and Sons, 1981.
 - [29] J.-M. Odobez, P. Bouthemy, Robust multiresolution estimation of parametric motion models, *Journal of Visual Communication and Image Representation* 6 (4) (1995) 348–365.
 - [30] P. Holland, R. Welsch, Robust regression using iteratively reweighted least-squares, *Communication Statistic-Theory Method A* 6 (9) (1977) 813–827.

-
- [31] J. Brailean, R. Kleihorst, S. Efstratiadis, A. Katsaggelos, R. Legendijk, Noise reduction filters for dynamic image sequences: a review, *Proceedings of the IEEE* 83 (9) (1995) 1272–1292.
 - [32] E. Meijering, K. Zuiderveld, M. Viergever, Image registration for digital subtraction angiography, *International Journal of Computer Vision* 31 (1999) 227–246.
 - [33] S. Baert, M. Viergever, W. Niessen, Guide wire tracking during endovascular interventions, *IEEE Trans. Med. Imaging* 22 (8) (2003) 965–972.
 - [34] V. Nzomigni, C. Labit, J. Liénard, Motion-compensated lossless compression schemes for biomedical sequence storage, in: *Picture Coding Symposium, Lausanne, 1993*.
 - [35] Y. Boykov, G. Funka-Lea, Graph cuts and efficient n-d image segmentation, *International Journal of Computer Vision (IJCV)* 70 (2) (2006) 109–131.
 - [36] V. Auvray, J. Liénard, P. Bouthemy, Multiresolution parametric estimation of transparent motions and denoising of fluoroscopic images, in: *MICCAI'05, Vol. 2, Palm Springs, USA, 2005*, pp. 352–359.



Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399