Motion Estimation for Regions of Reflections Through Layer Separation

Mohamed Abdelaziz Ahmed, Francois Pitite, Anil Kokaram

Sigmedia, Electronics and Electrical Engineering Department, Trinity College Dublin (www.sigmedia.tv)

Abstract

Regions of reflections contain two semi-transparent layers moving over each other. This generates two motion vectors per pel. Current multiple motion estimators either extend the usual brightness consistency assumption to two motions or are based on the Fourier phase shift relationship. Both approaches assume constant motion over at least three frames. As a result they can not handle temporally active motion due to camera shake or acceleration. This paper proposes a new approach for multiple motion estimation by modeling the correct motions as the ones generating the best layer separation of the examined reflection. A Bayesian framework is proposed which then admits a solution using candidate motions generated from KLT trajectories and a layer separation technique [18]. We use novel temporal priors and our results show handling of strong motion inconsistencies and improvements over previous work.

Keywords: KLT, Separation, Intrinsic, Bayesian, Multiple Motions.

1 Introduction

Motion estimation is important in cinema postproduction for many fundamental tools e.g. dust busting, retiming, frame rate conversion and match moving. Most motion estimators [7] assume that there is only one motion at a pixel site and then use the brightness consistency assumption introduced by Horn et al. [6]. Hence image brightness is assumed constant over time given small image displacement. This one motion per site idea is violated in many everyday situations e.g. shadows cast on a moving object or reflections from shiny surfaces. In those situations standard motion estimators fail and cause strange effects when the frames are compensated regardless of the application. This is because there are now at least two semi-transparent layers moving over each other (see Fig.1) generating two motion vectors per pel.

Several motion estimators have been designed to cope with these situations and traditionally there are two stages. The first stage [8, 2, 12, 9, 16, 4, 13, 15] estimates the two motion vectors for each examined site while the second stage (motion-layer labeling) performs motion assignment to two layers [4, 13, 15]. The key issue here though, is that previous work did not explicitly involve a mixing model that explains the transparency or reflection. In Toro et al [15] the *layers* actually were support regions associated with each motion parameter. So for example in Toro et al [15], considering angiogram sequences, if there were four motions applicable to a patch, each model is associated with a separate support map indicating picture areas which were inliers and outliers with respect to the relevant optical flow equation. In Stuke et al [13] the layers referred to a classification that determined whether there were one or two motions at a site but did not actually extract the layers involved.

This paper takes a more complete approach to the problem in a sense. We take the novel approach of exploiting a model for the observed image which is a mixture of two layers that each move according to some motion trajectory over several frames. Thus the motion estimation problem is articulated as a motion and layer separation problem through a Bayesian framework. The computational burden of the joint MAP solution for layers and motion is eased by generating candidates for motion (KLT tracks using [14]) and layers (using [18]) by pre-processing the observed sequence. These candidates are then evaluated within the probabilistic framework to select the best motion inference.

In the next section we review the previous work before going on to propose our Bayesian framework and solution. We will employ the usual model for an observed image M (for mixture) showing a reflection or transparency as a linear combination between the (hidden and underlying) source background and foreground layer images $L1$ and $L2$ as follows.

$$M = L1 + L2 \tag{1}$$

The complete problem is that of estimating both the motions that L1 and L2 exhibit as well as estimating L1 and L2 themselves, even though we are not necessarily interested in L1 and L2.

2 Related Work on Multiple Motion Estimation

2.1 Optical Flow Approaches

For one motion, the optical flow constraint equation (OFCE) [6] is defined as follows.

$$v1_x L1_x + v1_y L1_y + L1_t = 0 \tag{2}$$

Here $(v1_x, v1_y)$ are the x and y motion components for $L1$, and $L1_x, L1_y, L1_t$ are the spatial and temporal gradients of the image layer $L1$. This equation can be rewritten as

Figure 1: From Left; Frames 5,17,25 from a sequence containing reflection of a building superimposed on paper posters. In this sequence the building is moving to the left while the posters are moving to the right. This generates two motion vectors per pel.

$d(v1)L1(x,t) = 0$ where $d(v1) = v1_x\partial x + v1_y\partial y + \partial_t$. Similarly, for another layer $L2$ moving with $(v2_x, v2_y)$, its OFCE is $d(v)L2(x,t) = 0$. As the observed image M containing the reflection is expressed as $M = L1 + L2$, the OFCE for two motions become $d(v1)d(v2)M(x,t) = 0$. This implies

$$v1_x v2_x M_{xx} + v1_y v2_y M_{yy} + M_{tt} + (v1_x v2_y + v2_x v1_y)M_{xy} + (v1_x + v2_x)M_{xt} + (v1_y + v2_y)M_{yt} = 0 \quad (3)$$

Here motion is assumed to be constant over three frames, and the terms M_{xx}, M_{xy}, M_{yy}, M_{tt} refer to the various second order spatial and temporal derivatives. Some approaches [2, 15, 8] were proposed to solve Eq. 3. They assume that motions are constant within an image patch and estimate $(v1, v2)$ by minimizing the left hand side of Eq. 3 within the examined patch. Aach et al. [2] estimates motions by eigensystem analysis of suitably extended tensors, yielding so-called mixed-orientation parameters (MOPs). They show how to decompose the MOP vectors into the individual motions. They show interesting results on observed image mixtures formed by highly textured layers. Mota et al. [8] split Eq. 3 into linear and non-linear parts. Similarly to Aach et al. [2], the linear part is solved by eigenvalues analysis of a suitable structure tensor. A closed form solution is then proposed to solve the non-linear part by expressing motions in form of complex numbers. They show interesting motion estimation on real sequences. Finally, Toro et al. [15] solve for the motions by minimizing the residual of Eq. 3 using the Geman-McClure norm. Optical flow approaches show interesting results on limited number of angiogram sequences. However, they assume constant motion over three frames and hence can not handle non-uniform motion. In addition, the brightness consistency assumption of Horn et al. [6] used here is just valid for small image displacements.

2.2 Fourier Transform Based Approaches

Fourier Transform Based approaches are based on transforming Eq. 1 to the Fourier Domain where motion manifests as phase shifts.

Vernon [16] solves for the phase shifts directly in the Fourier domain using the Hough transform to find planes. They show interesting motion estimation and layer separation results on real sequences, however, their approach requires motions to be constant over five frames.

Stuke et al. [13] showed that it is possible to use the Fourier domain expression of the problem to derive a recursive expression for the dependency between more than one frame. For two motions the expression is

$$M_2(\mathbf{X}) + M_0(\mathbf{X} - v1 - v2) - M_1(\mathbf{X} - v1) - M_2(\mathbf{X} - v2) = 0 \quad (4)$$

Stuke et al. [13] and Auvray et al. [4] then solve for $(v1, v2)$ by minimizing the left hand side of Eq.4 using exhaustive search through block matching. Here $(v1, v2)$ are assumed to be spatially constant within a small image patch. They show good motion estimation results on synthetic and real sequences. However, one major drawback of this approach is that it assumes constant motion over three frames and hence leads to error propagation in case of temporally active motion or motion inconsistencies. In addition, it is computationally expensive as N^4 different combinations of $(v1, v2)$ are examined if we search N possible displacements for each motion component $(v1_x, v1_y, v2_x, v2_y)$.

3 Motivation for a new technique

Optical flow approaches cannot handle large image displacements while transform-based approaches can be computationally expensive. In addition, both approaches assume constant motion over at least three frames. This assumption is hardly true for sequences shot by non-professionals using regular hand cameras. Processing such sequences with previous techniques will generate incorrect motion estimation and incorrect motion-layer labeling. This often leads to error propagation through time. In the next section we present a multiple motion estimator that does not assume constant motion over a block of frames, and also explicitly involves the underlying, hidden image layers. We model the correct motion pairs $(v1, v2)$ as the ones generating best layer separation of the examined reflection. Our approach is more computationally efficient than Transform-Based

approaches, handles large image displacements and can handle strong temporal motion inconsistencies.

4 Bayesian Inference for Multiple Motion Estimation through Layer Separation (BIMS)

Expressing the problem of estimating the hidden layers and their motion in a probabilistic fashion, results in the following

$$p(\mathbf{v}, \mathbf{L}|\mathbf{M}) \propto p(\mathbf{M}|\mathbf{v}, \mathbf{L})p(\mathbf{v})p(\mathbf{L}) \quad (5)$$

where $\mathbf{v}$ contains the motion information for both layers $\mathbf{L}$, and $\mathbf{M}$ contains a suitable window of observed frames (in our experiments we use 5 frames). However we are interested in motion and not necessarily in good hidden layer estimation. This would normally imply an approach involving marginalization of the estimated layers. However, in an attempt to target a computationally more attractive scheme, we discuss next the approximate equivalence between motion and the underlying hidden image layers.

4.1 Equivalence between motion and hidden image layers

Weiss [18] was the first to consider separation of images into reflectance (or intrinsic layers) and illumination layers. In his analysis, one layer must be stationary and the gradients of the illumination image follow a Laplacian distribution. Although his problem is more generic in a sense than the problem here, we can consider $L1$ and $L2$ to be reflectance and illumination images, in which case $L2 = M - L1$. Denoting $M_{0:4}$ as the mixtures M for frames t=0:4, the likelihood of M given the layer $L1$ is as follows.

$$P(M|L1) \propto \sum_{i=0}^{4} |\frac{\partial M_i}{\partial x} - \frac{\partial L1}{\partial x}| + \sum_{i=0}^{4} |\frac{\partial M_i}{\partial y} - \frac{\partial L1}{\partial y}| \quad (6)$$

where $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ are the horizontal and vertical spatial derivatives respectively. Hence best estimate (in the ML sense) for the gradients of the layer $L1$ is given by the median of the gradients of the observed image M through the 5 frames in this case. Given that gradient estimate Weiss is then able to reconstruct the underlying image $L1$ itself. This is the situation we have here, except that both layers can be moving and M is a mixture of those moving layers.

Assume now that we motion compensate the frames $M_{0:4}$ with the motion for layer L giving $M^L_{0:4}$. Sarel et al [11] note that, assuming the motion compensated sequence now contains a moving transparent layer against another stationary layer, median filtering of the spatial gradients of this volume will suppress the spatial gradients of the moving layer, leaving behind the spatial gradients of L only. These gradients can then be used to reconstruct the hidden image layers following Weiss' work.

Furthermore we observe that feature point tracking on image patches showing reflections inevitably results in tracks that follow one or the other layer through the 5 frame window in this case. This means that for each examined site, motion candidates for each layer can be derived from nearby KLT trajectories [14] of length more than four frames. $M^L_{0:4}$ for the examined candidate can then be taken as the five image patches centered along the examined trajectory at t=0:4. Thus each motion candidate generates an estimation of either $L1$ or $L2$, and constraints on the appearance of $L1$ and $L2$ can directly be applied to constrain the estimation of $\mathbf{v}$. Furthermore, given that within a patch the motion of a layer is constant, then the trajectory measurements can be used as a constraint on the estimated motion of the layers at each considered site. Thus we do not solve for optimum layer separation, and instead we solve for separation that is good enough to recover the motions of $L1$ and $L2$.

4.2 Approximate inference for motion

Assume that at each site in the image we can propose candidates for motion and layer images. Furthermore assume that we have already estimated a number of feature point trajectories. Then inference for $(v1, v2)$ over the examined patch can proceed as follows

$$\begin{aligned} P(v1, v2, L1, L2|M, \mathcal{T}) &= P(M|v1, v2, \mathcal{T}, L1, L2) \\ &\times P(v1|\mathcal{V}1, \mathcal{T}, L1)P(v2|\mathcal{V}2, \mathcal{T}, L2)P(L1, L2) \end{aligned} \quad (7)$$

Here $\mathcal{T}$ are the estimated feature point trajectories, $P(M|v1, v2, \mathcal{T}, L1, L2)$ is the likelihood of generating the observed image given the motion and associated layers and $p(v1, v2|\cdot)$ imposes spatial and temporal smoothness on the generated motion with $P(L1, L2)$ being priors constraining the appearance of the hidden layers. $\mathcal{V}1, \mathcal{V}2$ are local 8 connected motion neighborhoods in the examined block.

4.2.1 Likelihood

The likelihood $P(M|v1, v2, \mathcal{T}, L1, L2)$ given the layers $L1, L2$ is independent of $v1, v2$ and the trajectory information. Hence we express $P(M|v1, v2, \mathcal{T}, L1, L2)$ using a modified version of Weiss's likelihood term as follows

$$\begin{aligned} &P(M^{Lj}_{n:n+4}|L_j) \propto \\ &\qquad \exp -\frac{\lambda_w}{5}\left(\sum_{i=n}^{n+4} 1 - \text{SSIM}(Lj, M_i^{L_j})\right) \end{aligned} \quad (8)$$

where λ_w is a weight to configure the importance of the Weiss's likelihood. λ_w is fixed to 1 in all experiments. Note that instead of using gradients we use the structure part of the SSIM measure proposed in [17] and the likelihood therefore measures the error in estimating Lj Weiss' technique. We use the SSIM here because it is a dimensionless quantity having a range between 0 and 1. Furthermore using the structural component of the SSIM encourages confidence that this likelihood constrains the appearance of the layer in some useful perceptual way independent of illumination. This reflects the assumption of illumination variation made by Weiss. This is the same illumination variation assumption

made by Weiss. As Weiss' technique estimates layers up to a constant, we turn off the luminance component of the SSIM wherever it is used in this paper.

4.3 Priors

As far as priors on motion are concerned, we can factorize the priors on $\mathbf{v}$ into a spatial component $P_s(\cdot)$ and a temporal component $P_t(\cdot)$. Hence

$$P(v1|\mathcal{V}1,\mathcal{T},L1)P(v2|\mathcal{V}2,\mathcal{T},L2) = P_s(v1|\mathcal{V}1,\mathcal{T})P_t(v1|\mathcal{T})P_s(v2|\mathcal{V}2,\mathcal{T})P_t(v2|\mathcal{T}) \quad (9)$$

where $P_s(\cdot)$ and $P_t(\cdot)$ are independent of $L1$ and $L2$.

Spatial smoothness is then imposed on motion using a Gibbs prior as follows.

$$P_s(v1|\mathcal{V}1) \propto \exp -\lambda_s \left(\sum_{k \in \mathcal{N}} (||v1_e - v1_k||^2) \right) \quad (10)$$

and similarly for $v2$. The neighborhood $\mathcal{N}$ indexes the 8 neighboring blocks to the examined block e, and λ_s is the spatial smoothness strength set to 0.02 in most experiments.

To understand the design of the prior note that trajectories estimated with KLT tracking are clustered using Mean Shift to acquire a crude notion of motion objects. Given a particular trajectory T, we can associate with it a weight W_j calculated during Mean Shift Clustering. Hence the temporal prior is defined as follows

$$P_t(vj|\mathcal{T}^j) \propto \exp -\lambda_t (1 - W_j) \quad (11)$$

This therefore biases motion estimates towards more confident trajectories. λ_t is a weight to configure the importance of this term and set to 1 in all experiments.

The priors for L are designed to contain a distribution P_{dl}, encouraging the layers to be **different**, and P_{tl} encouraging the layers to be temporally smooth along our 5 frame window.

$$P(L1,L2) \propto P_{dl}(L1,L2)P_{tl}(L1)P_{tl}(L2) = \exp -\left[\lambda_{dl} \left(\text{SSIM}(L1,L2) \right) + \lambda_{tl} \left(1 - \text{SSIM}(Lj, Lj_{n-1}) \right) \right] \quad (12)$$

Again we use the SSIM as proxy for measuring image similarity (only structural) and the first term encourages the similarity between the two images to be low. The second term simply measures the similarity of consecutive image frames in each layer and hence encourages that to be high for good temporal smoothness.

Once more $(\lambda_{dl}, \lambda_{tl})$ are various smoothness weights where λ_{dl} is fixed to 10 in all experiments while λ_{dt} is fixed to 10 in most experiments. To avoid assigning the same extracted layer to both $L1$ and $L2$, we impose structural independence between $L1$ and $L2$ through Eq. 12 (first term). The second term of Eq. 12 generates temporally consistent solutions by imposing temporal consistency on the separated layers.

5 Solving for $(v1, v2)$

As stated above we use a pragmatic approach to solve the inference problem by generating plausible candidates for all the variables starting from KLT trajectories.

Motion Candidates: Each frame is examined in non overlapping blocks of 50×50 pels. For each block, all KLT trajectories [14] within 150 pels from the examined block center and of length more than four frames are selected. Denoting the selected trajectories by T, $T(x,y)_n^i$ represents the (x,y) position of the i^{th} trajectory at frame n. The motion candidate (s_x, s_y) proposed by T_n^i is taken as

$$s_x = T(x)_{n+1}^i - T(x)_n^i \quad (13)$$

$$s_y = T(y)_{n+1}^i - T(y)_n^i \quad (14)$$

This candidate describes the forward motion for one of the source layers at the examined block. To group together trajectories corresponding to the same layer, all trajectories T are segmented into coherent clusters by segmenting their corresponding motion candidates (s_x, s_y) (which are calculated using Eq. 13-14). Here Mean Shift Clustering [5] is used with Bandwidth of 2 pels. The four cluster centroids $(\mathcal{S}_x^c, \mathcal{S}_y^c)$, c=1:4, containing most of the examined trajectories serve as a pool of candidates for $(v1, v2)$. We use four motion candidates for simplicity. For each one of those clusters, a new trajectory $\mathcal{T}(x,y)_t^c$ is assigned to it as follows

$$\mathcal{T}(x,y)_n^c = \frac{\sum_{i=1}^K \delta(c - C(i)) T(x,y)_n^i}{\sum_{i=1}^K \delta(c - C(i))} \quad (15)$$

Here K is the number of examined KLT trajectories and $C(i)$ is the cluster assignment label of each examined trajectory resulting from Mean Shift clustering. W_j used in defining the temporal motion prior (see Eq. 11) is then the measured probability that $\mathcal{T}^c$ belongs to the particular cluster c.

Layer Candidates: Consider $M_{n-n+4}^{I_c}$ to be the five 50×50 image patches centered along $\mathcal{T}(x,y)_t^c$ at points t=n:n+4. Each one of those images contain a constant layer I_c undergoing varying illumination through time. I_c corresponds to one of the source layers at the examined block and it is estimated using Weiss' [18] technique. Fig 2 (middle row) shows an example of $M_{n:n+4}^{I_c}$ for the green region shown in the first row, left. Here a synthetic sequence is created by mixing lenna with an image of oranges using the additive mixing model previously introduced (see Eq. 1). Lenna undergoes a motion with $(v1x, v1y) = (5, 0)$ while the oranges undergo motion with $(v1x, v1y) = (-5, 0)$. In the second row, I_c is the picture of lenna. Here the picture of lenna is stationary while the oranges are moving to the left through time. Lenna's image (I_c) is extracted by applying Yair's technique on $M_{n-n+4}^{I_c}$ (see third row, first column). Fig 2 (last row) shows the extracted layers for the four motion candidates of the examined block. The best layer estimates in the first and second columns correspond to the correct motions for the examined block.

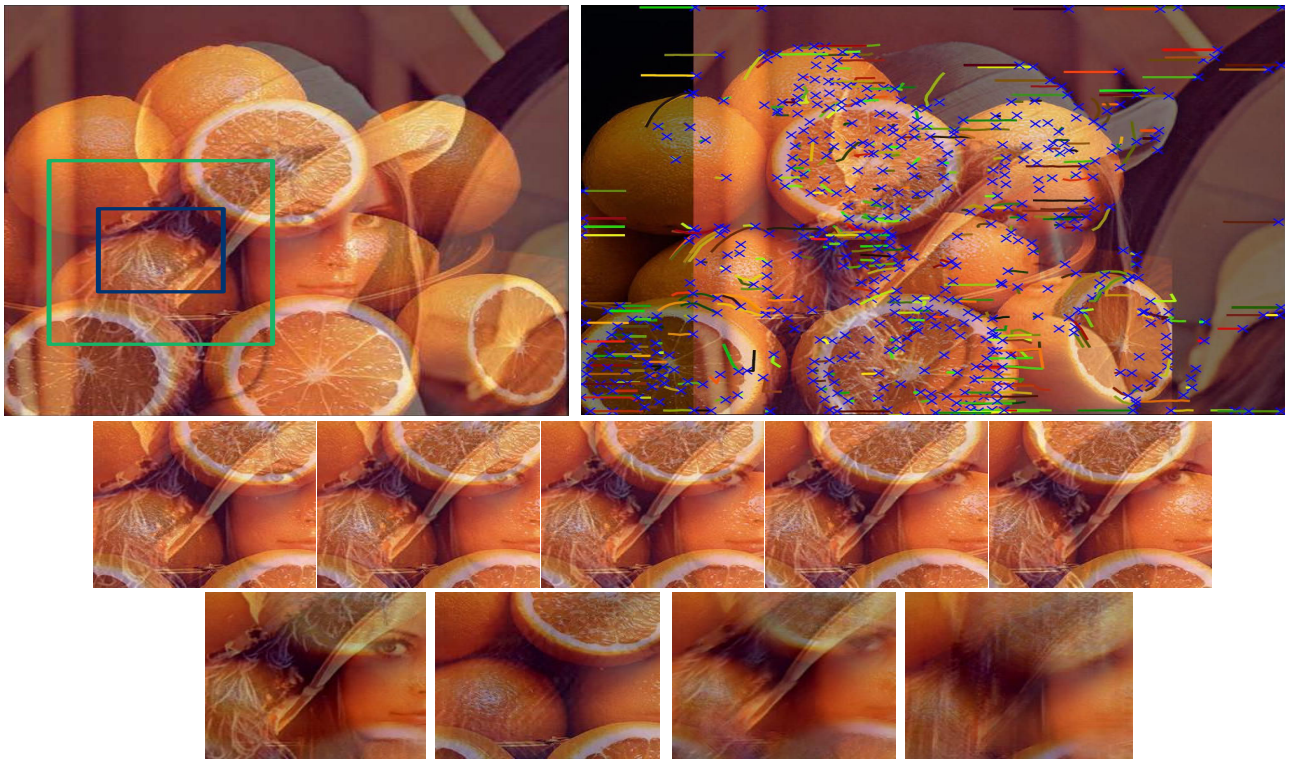

Figure 2: Experiments on a synthetic sequence created by mixing lenna with oranges. Top Row: First frame from the sequence (left) showing a patch to be considered (green) and the associated support patch (blue) from which motion candidates (derived from $\mathcal{T}$) are drawn, the corresponding trajectories $\mathcal{T}$ for the 10^{th} frame (right). Each trajectory starts with a blue cross and has a different color. Lenna and the oranges are moving with constant velocities of $(v1x, v1y) = (5, 0)$ and $(v2x, v2y) = (-5, 0)$ respectively. Middle Row: $M^{I_c}_{n:n+4}$ for the considered patch using the first motion candidate. Bottom Row: Layers extracted using Weiss' technique for each motion candidate. Motion candidates are (from left) $(\mathcal{S}_x, \mathcal{S}_y)$ = $(5, 0)$, $(-5, 0)$, $(3, 3)$ and $(5, -7)$ respectively. As shown, the best layer estimates manifest in the first two images and these correspond to the correct motions of the layers.

Given four motion candidates $(\mathcal{S}^c_x, \mathcal{S}^c_y)$, c=1:4, four layers are calculated for each examined block. As each one of those four layers can be assigned to either $L1$ or $L2$, there are $2^4 = 16$ possible combinations of $(L1, L2)$. Each layer combination corresponds a candidate for the motion pair $(v1, v2)$. The motion pair corresponding to the best estimates of (L_1, L_2) is treated as the motions of the observed mixture.

Final Solution with GraphCuts: Given the candidates for motion and layers above, $(v1, v2)$ are estimated by solving $P(v1, v2, L1, L2|M, \mathcal{T})$ (see Eq. 7) using QPBO Graph Cuts [10]. This is done by choosing between two $(v1, v2)$ candidates at a time until all the 16 possible combinations of $(v1, v2)$ are examined. All blocks for each frame are processed. To process the first frame at $n = 0$ the temporal energy is not considered.

6 Results

We compare our technique (BIMS) against our implementation of two multiple motion estimators, one uses the optical flow approach and the other uses the Fourier Transform Based approach. For the optical flow approach we assume motion is constant within a block of 50×50 pels. We then estimate the motion by solving the OFCE using least square fitting. Note that none of the previous optical flow approaches attempted to solve the motion-layer labeling problem. However in our implementation of the optical flow approach we solve the motion-layer labeling by imposing spatial smoothness on the generated motions as discussed in Sec. 4.3. However here we only have 2 motion candidates per examined block, one for each layer. This generates 4 motion combinations per block. We impose temporal smoothness on the generated motions and we call this technique OPTIC.

For the Transform-Based approach we implement the technique of Stuke et al. [13]. Their technique solves the motion-to-layer labeling problem using MRFs, imposes temporal smoothness on the generated motions, and uses ICM to maximize the maximum-a-posteriori solution. We implement this approach using Graph Cuts. For each block, we estimate $(v1, v2)$ using Block matching. For each motion component we search displacements -8:2:8 (MATLAB notations). This generates $9^4 = 6561$ possible motion combinations of $(v1x, v1y, v2x, v2y)$ for each block. The 16 motion combinations/pairs optimizing the maximum likelihood solution the most are treated as a pool of motion

candidates for the examined block. We call this technique FTRANS. For an examined sequence, the weights of the spatial and temporal energies $(\lambda_s, \lambda_{tl})$ (see Eq.10 and Eq.12) have the same values in BIMS, OPTIC and FTRANS.

Ground-truth Comparison

We test our technique on synthetic and real sequences. Full image sequences results can be found in www.sigmedia.tv/Misc/CVMP2011. Motions as estimated by the examined techniques are compared against ground-truth estimates. For real sequences we generate ground-truth estimates manually. We estimate one dominate motion for each layer. For each layer we manually track one strong feature point through time and set the motion of the examined layer to the motion of examined/tracked feature point. For one frame, the error E between the estimated and ground-truth motions is defined as follows.

$$E = \frac{1}{2}\sum_{j=1}^{2} 1/N \sum_{i \in \mathcal{R}} |(Vj(i) - Uj(i)| \qquad (16)$$

Here $Vj(i)$ is the calculated (2 component) motion for layer j at site i while U is the ground-truth estimate. $\mathcal{R}$ is the examined region representing regions of reflections and N is the number of examined sites/blocks in this region. Ideally one should use automated approaches to detect the regions of reflections/interest $\mathcal{R}$. However reflection detection is a hard problem and so far only one technique is proposed [3] to solve this problem. Hence for the sake of accuracy, we manually define $\mathcal{R}$ for each frame. However, as we are just interested in evaluating our technique in regions where single motion estimators fails, we define regions of reflections as ones having two distinctive semi-transparent layers moving over each other.

6.1 Synthetic Sequences

A synthetic sequence of 53 frames is created by mixing an image of lenna with an image of oranges (both images of size 480×480 pels) using the additive mixing model (see Fig. 4). A repetitive motion model with strong temporal inconsistencies is applied to each layer to examine the robustness of BIMS to camera shake. Lenna is moving with $v1_n = (0, 2)$, $v1_{n+1} = (-2, -3)$ and $v1_{n+2} = (2, 3)$ at $(n, n+1, n+2)$ respectively while Oranges are moving with $v2_n = (0, -2)$, $v2_{n+1} = (2, 3)$ and $v2_{n+2} = (-2, -3)$. This motion pattern repeats itself at $(n+3, n+4, n+5)$ till the end of the sequence. Fig. 3 shows the motion estimation error as defined by Eq. 16 for BIMS, OPTIC and FTRANS. Here $(\lambda_s, \lambda_{tl}) = (0.5, 10)$. As shown, OPTIC and FTRANS generated high motion errors as they assume constant motion over three frames, an assumption that is violated in the examined sequence. However, our technique was able to handle such heavy motion inconsistencies as BIMS does not assume constant motion over any frame window. In addition, unlike previous techniques as in Stuke et al. [13], BIMS imposes temporal smoothness on the separated layers not on the generated motions. This generated motion-layer labeling that is temporally consistent.

Fig. 4 (first two rows) shows the extracted motions for frame 11 (first row) and 12 (second row) respectively for the synthetic sequence. Here motion vectors are scaled by a factor of 5 for illustration clarity. Groundtruth motions for the two frames are $v1_n = (-2, -3)$, $v2_n = (2, 3)$, $v1_{n+1} = (2, 3)$ and $v2_{n+1} = (-2, -3)$ respectively. As shown, the motion-layer labeling generated by BIMS is temporally and spatially consistent. However, both FTRANS (middle column) and OPTIC (Last column) generated large motion errors as the examined motions undergo strong motion discontinuities. In addition, OPTIC generated small motion estimates. Such behavior is expected for optical flow approaches as the OFCE is only valid given small image displacements. Single motion estimators overcame this assumption by iteratively refining the motion estimates. This was done by iteratively shifting the examined frame by the estimated motions and re-estimating motions using OFCE on the shifted frames. However none of the optical flow multiple motion estimators proposed to iteratively refine the motion estimates. One should note that for reflections this is not an easy problem as it requires iteratively shifting each layer with its estimated motion. This will require an intermediate stage that perform layer separation, a problem that requires motions to be known beforehand.

BIMS Parameters Configuration To illustrate the importance of the energy terms in Eq. 8 and in Eq. 10- 12, we change the values of their weights (denoted by λ) and reprocess frame 12 of the synthetic sequence. Fig. 4 (last row, first column) shows the result of switching OFF the temporal consistency term by setting $\lambda_{tl} = 0$. This generated motion-layer labeling that is temporally inconsistent with frame 11. More explicitly, the background motion at frame 11 became the foreground motion at frame 12 and vice versa for the foreground motion. To illustrate the importance of the layer structural independence term, we switch OFF its term by setting $\lambda_{dl} = 0$. The result is the generation of one motion vector per site as shown in Fig. 4 (last row, second column). The reason for this result is that Eq. 8 now assigns the same separated layer to both $L1$ and $L2$. Last, Fig. 4 (last row, third column) shows the result of switching OFF the motion spatial smoothness term by setting $\lambda_s = 0$. This generates motion-layer labeling that is that spatially inconsistent as shown in the red rectangle.

6.2 Real Sequences

Eight real sequences containing 329 frames of size 576×720 are processed with OPTIC, FTRANS and BIMS. $(\lambda_s, \lambda_{tl})$ for the 8 examined sequences are set to $(0.05, 10)$, $(0.01, 10)$, $(0, 05, 10)$, $(0.05, 10)$, $(0.01, 10)$, $(0.02, 10)$, $(0.2, 10)$ and $(0.02, 10)$. All examined sequences (see Fig. 5) contain a foreground reflection layer superimposed on a background layer. Sequences are shot with handheld cameras at a distance close to the background layer, leading to motion unsteadiness and acceleration. The result is that the background layer is moving faster than the foreground layer in most sequences.

Tab. 1 shows the motion estimation errors (as defined in Eq. 16) for the examined sequences. As shown in Tab. 1, BIMS generates the least motion errors in all the examined

Figure 4: Multiple motions for frame 11 (top row) and frame 12 (second row) as estimated by, From left; BIMS, FTRANS and OPTIC. Green and blue vectors represent the estimated motions of Lenna and Oranges respectively. Motion estimations of BIMS for the two frames are $v1_n = (-2, -3)$, $v2_n = (2, 3)$, $v1_{n+1} = (2, 3)$ ***and*** $v2_{n+1} = (-2, -3)$ ***respectively. Motions shown in the figure are scaled by a factor of*** 5 ***for illustration clarity. BIMS generated motion-layer labeling that is temporally consistent despite the strong motion discontinuities in the examined sequence. However, FTRANS and OPTIC failed to handle the strong motion discontinuities. Last Row: Examining effect of terms (on frame 12). From Left: Turning OFF the temporal consistency term from BIMS generates motion-layer labeling that is temporally inconsistent with frame 11; Turning OFF the layer structural independence term from BIMS generates one motion per examined site; Turning OFF the motion spatial smoothness term from BIMS generates motion-layer labeling that is spatially inconsistent (see the red box).***

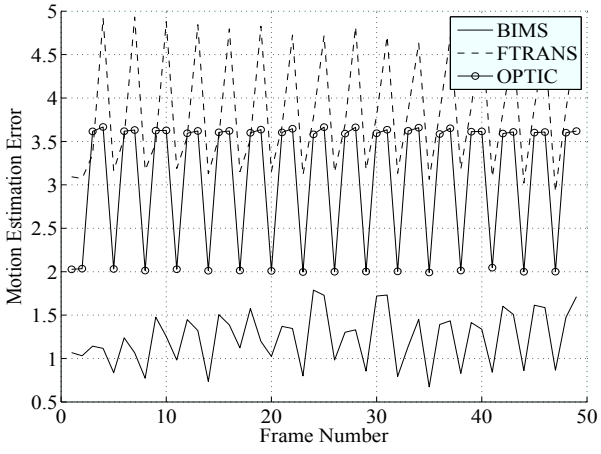

Figure 3: Motion Estimation Error for BIMS (-), FTRANS (- -) and OPTIC (-o) on an artificially created sequence. FTRANS and OPTIC generated large motion errors as the examined sequence undergoes strong motion inconsistencies.

sequences. For **Bulb** and **WindRef2** we create strong motion inconsistencies by dropping every third and fourth frame from the original sequences. By doing so we strongly violate the 3-frames motion consistency assumption used by optical flow and Transform-Based approaches. The result is generation of large motion errors by OPTIC and FRANS as shown in Tab. 1 and in Fig 5 (third and fourth rows). However, our technique BIMS handles these strong motion inconsistencies much better than OPTIC and FTRANS (see table and Fig 5 last column). Similar results are generated for **SelimK2** and **PicRef** (see Table and Fig 5 (first and fifth rows respectively)). **SelimK2** undergoes motion acceleration while **PicRef** undergoes slight camera shake. OPTIC and FTARNS were not able to handle this slight camera shake. In **PortraitA2** (see Fig 5 (second row)) FTRANS generates high motion estimation errors especially for the purple region. In this region the vertical white bar is moving to the left and hence generates an aperture effect. FTRANS failed to handle this effect as it does not incorporate enough global information. However BIMS handles this effect successfully as KLT trajectories near this region capture global motion information that are good enough to recover the motion

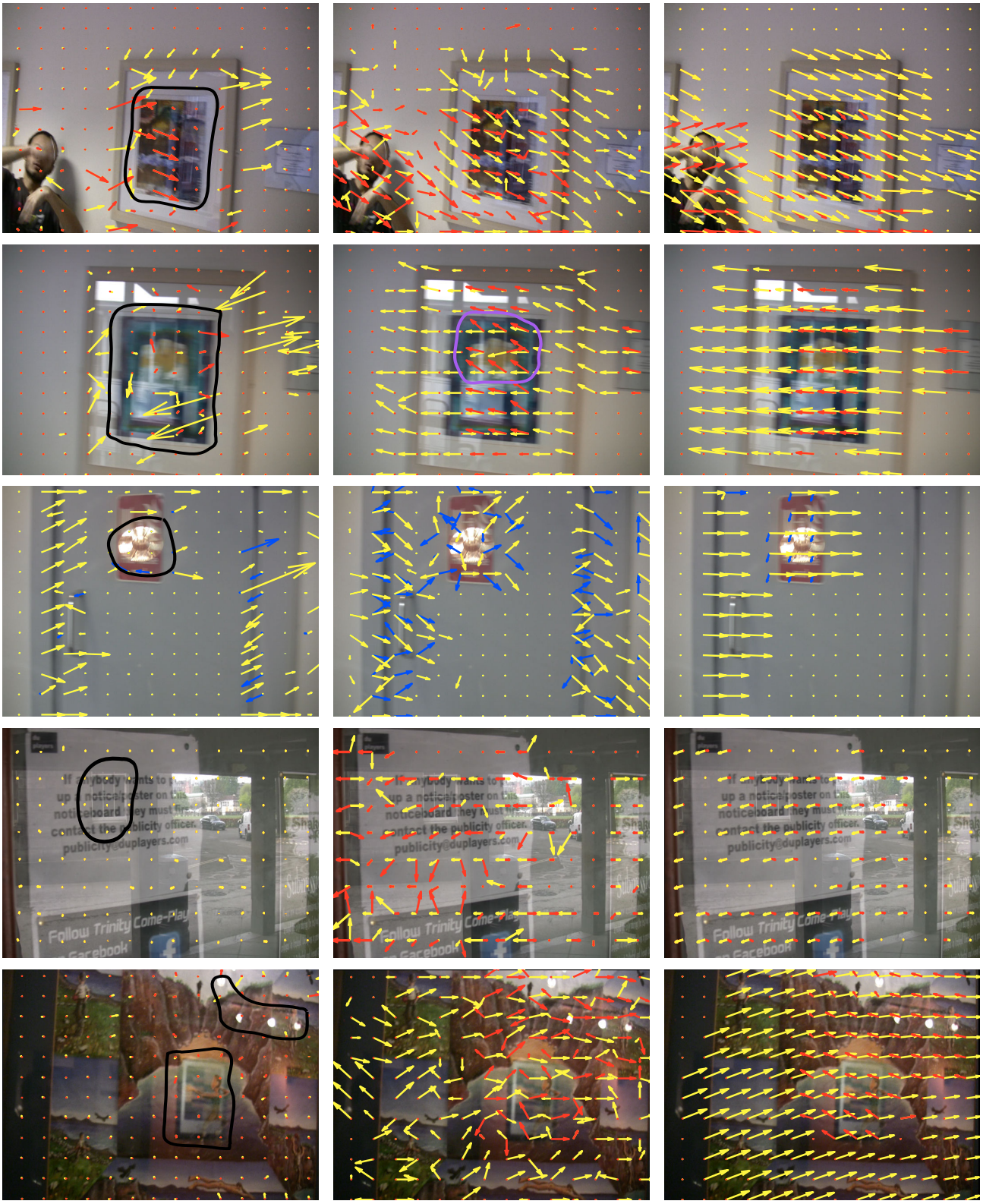

Figure 5: Motion estimation on real sequences with reflections (regions of interest $\mathcal{R}$ are shown in black in the first column), using from left to right: OPTIC, FTRANS and BIMS. Yellow vectors represent the motion of the background layer while the foreground layer is represented by either red or blue vectors (for viewing contrast). In all sequences the background layer is moving with a higher speed than the foreground. OPTIC generates erroneous small motion estimates while FTRANS cannot handle temporal motion inconsistencies. BIMS generates the best motion estimates in all sequences. In the second row, the foreground and background motions of BIMS are in the same direction however the background motion has a stronger magnitude. This matches the ground-truth estimate and handles the aperture effect (shown in purple) better than FTRANS.

	SelimK2	Bulb	RedBack	PortraitA2	PicRef	WindRef1	WindRef2	BuildOnWind3
OPTIC	7.1±1.9 2.8 11.4	14.7±9 4.9 27.8	5.4±1.8 2.1 9.1	10.8±2 6.8 15.1	12.5±4.6 4.3 25.6	3.1±2.0 2.0 10.6	5.9±1.2 3.5 9.6	5.5±3.1 2.0 10.1
FTRANS	4.6±1.0 2.8 11.3	15.1±7.3 6.4 26.2	2.6±1.2 0.7 5.2	6.1±1.7 2.5 9.4	10.2±4.6 4.7 23.0	2.7±4 4.0 12.5	3.7±0.8 1.8 5.6	5.2±2.0 2.6 8.1
BIMS	1.8±0.5 1.1 3.1	2.2±0.8 0.8 4.2	1.1±0.4 0.5 2.2	2.9±0.9 1.5 4.7	2.15±3.5 0.7 5.7	1.4±0.3 0.32 6.7	1.3±0.4 0.4 2.3	1.0±0.36 0.32 1.89

Table 1: Motion estimation errors for OPTIC, FTRANS and BIMS on 8 real sequences. For each technique we show the mean, minimum and maximum error over the frames of the examined sequence. As shown our technique BIMS generates the least error in all examined sequences.

of the white vertical bar.

6.3 Reflection Detection for Multiple Motion Estimation

Processing an entire sequence with BIMS may generate false estimations in regions where there is no reflection. Fig. 5 (first row, last column) shows an example of this scenario. Here the actor is moving his hands quickly generating pathological motion. Processing the actor's hands with BIMS generates erroneous measurements. Hence it is required to mask out this region from the BIMS motion estimates. An approach to do so is by manually selecting the regions of interest/reflections $\mathcal{R}$ as shown in black in Fig. 5. This however requires large manual intervention. Instead it is possible to use Ahmed et al. [3] work on Automated Reflection Detection to estimate the regions of interest. Fig. 6 (in green) shows reflection masks of Ahmed et al. [3] generated for three of the eight examined sequences. These masks are used to weight the motion estimates in regions not containing reflection. For Fig. 6 (first column) the hand motion of the actor is successfully discarded.

6.4 Computational Complexity

Our technique consists of three main stages. KLT trajectories extraction, Layer Separation using Weiss [18] and MAP optimization using QPBO Graph-cuts. A C++ implementation of KLT trajectories extraction is used [1]. This takes ≈ 1 second to process one frame of size 576×720 pels. A C++ implementation of QPB0 Graph-Cuts is also available. This stage is computationally efficient as it is applied on block basis not on pel-basis. For a frame in standard definition and blocks of size 50×50 pels, there is 11×14 blocks/sites examined by QPBO. The most computationally expensive stage in BIMS is Layer Extraction using Weiss approach. To calculate the layer for one motion candidate, it requires 4×5 spatial derivatives, 2×5 Fast Fourier Transforms Operators and 2×5 Inverse Fast Fourier Transforms Operators. For one examined site and 4 motion candidates, Weiss approach requires 80 spatial derivatives, 40 FFTs and 40 IFFTs.

Optical flow approaches are computationally fast as closed-form solutions exist. However they generate poor motion estimates. Transform-Based approaches mainly use Block matching to solve for motions. For an examined site, K^4 block matching operations are required if one searches K possible displacements for each motion component. Hence there is 6561 and 83521 block matching operations for $K = 9$ and $K = 17$ respectively. Note in this technique there is an explosion in computational complexity in return for motion estimation accuracy. Searching a small range of motions reduces computational load but increases the chance of motion estimation inaccuracy. This problem does not exist in BIMS as motion candidates are proposed by the KLT trajectories.

The average time for processing one standard definition frame with BIMS and FTRANS is ≈ 1.3 and ≈ 3.2 minutes respectively. Here we use $K = 9$ for FTRANS and the average time is taken over a sequence of 50 frames. Both approaches are implemented using MATLAB and on the same Processor. By setting $K = 17$ the average processing time for one frame with FTRANS is ≈ 28 minutes. Note the explosion in computational load.

7 Conclusion

We have presented a Bayesian framework for multiple motion estimation in regions of reflection. By articulating the problem as the joint solution of motion and layers we are able to impose new constraints on the estimated motion through the layers that have been extracted. A key advance is to encourage motion to be estimated which improves the plausibility of the extracted layers. This allows our technique to be robust to temporal motion activity and to give improved motion fields as compared to existing approaches. We also propose a novel motion candidate selection scheme based on KLT trajectories which allows the joint problem to admit a tractable solution. We show examples where automated reflection detection can be used to weight out erroneous motion estimates in regions not containing reflection. However, the technique does require distinctive foreground and background layers. The development of an interactive scheme to circumvent this issue would help the tool to be useful in the post-production industry.

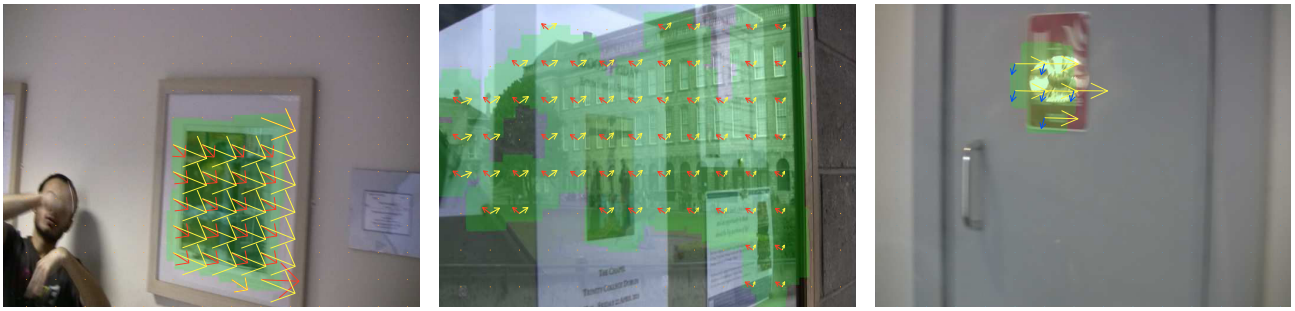

Figure 6: Motion estimation of BIMS with automated generated reflection detection masks (shown in green). Yellow and red/blue vectors show the background and foreground motions respectively. Reflection detection masks are generated from Ahmed et al. [3]. Detection masks are used to weight out erroneous measurements in regions not containing reflections.

Acknowledgements

This work is funded by the Irish Research Council for Science Engineering and Technology (IRCSET) and Science Foundation Ireland (SFI) PI CAMP Award 08/IN.1/I2112.

References

[1] http://www.ces.clemson.edu/ stb/klt/.

[2] T. Aach, C. Mota, I. Stuke, M. Muhlich, and E. Barth. Analysis of Superimposed Oriented Patterns. *TIPS*, 15(12):3690 –3700, 2006.

[3] A. Ahmed, Francois Pitie, and Anil Kokaram. Reflection Detection in Image Sequences. In *CVPR*, pages 705–712, 2011.

[4] V. Auvray, P. Bouthemy, and J. Linard. Motion-based segmentation of transparent layers in video sequences. In *Multimedia Content Representation, Classification and Security*, pages 298–305, 2006.

[5] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *ICCV*, pages 1197–1203, 1999.

[6] Berthold K. P. Horn and Brian G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.

[7] Anil Kokaram. Motion Picture Restoration. Springer Verlag.

[8] Cicero Mota, Ingo Stuke, Til Aach, and Erhardt Barth. Divide-and-Conquer Strategies for Estimating Multiple Transparent Motions. In *International Workshop on Complex Motion, Schloss Reisensburg, Germany. Lecture Notes on Computer Science, LNCS*, pages 66–77, 2005.

[9] M. Pingault and D. Pellerin. Motion estimation of transparent objects in the frequency domain. *Signal Processing*, 84(4):709–719, 2004.

[10] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via Extended Roof Duality. In *CVPR*, pages 1–8, 2007.

[11] Bernard Sarel and Michal Irani. Separating transparent layers of repetitive dynamic behaviors. In *ICCV*, pages 26–32, 2005.

[12] M. Shizawa and K. Maze. Unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *CVPR*, pages 289 –295, 1991.

[13] Ingo Stuke, Til Aach, Erhardt Barth, and Cicero Mota. Multiple-Motion-Estimation by Block-matching using MRF. *International Journal of Computer and Information Science*, 5(1), 2004.

[14] Carlo Tomasi Takeo and Takeo Kanade. Detection and tracking of point features. Carnegie Mellon University Technical Report CMU-CS-91-132, 1991.

[15] J.G. Toro, F.J. Owens, and R. Medina. Multiple motion estimation and segmentation in transparency. In *ICASSP*, pages 2087–2090, 2000.

[16] D. Vernon. Decoupling fourier components of dynamic image sequences: a theory of signal separation, image segmentation and optical flow estimation. In *ECCV*, pages 68–85, 1998.

[17] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIPS*, 13(4):600–612, April 2004.

[18] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, pages 68–75, 2001.