Information Society Technologies

Project no.:	**IST-FP6-FET-16276-2**
Project full title:	**Learning to emulate perception action cycles in a driving school scenario**
Project Acronym:	**DRIVSCO**
Deliverable no:	**D2.1**
Title of the deliverable:	**Recurrent regularization methods**

Date of Delivery:	31.03. 2007
Organization name of lead contractor for this deliverable:	UGE
Author(s):	M. Chessa, S.P. Sabatini, F. Solari, UGE
Participant(s):	UGE, UGR
Work package contributing to the deliverable:	WP2
Nature:	R
Version:	2.0 (revised on 09/06/08)
Total number of pages:	53
Start date of project:	1 Feb. 2006 **Duration:** 42 months

Project Co-funded by the European Commission		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other program participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Summary: In this deliverable an approach to perform spatial adaptive regularization of the optic flow coming from the front-end is described. The results are quantitatively assessed and compared with the state-of-art.

Contents

1 Revision notes

According to what reported in the review report, the performance of the regularisation methods described in D2.1 was assessed in a merely descriptive and qualitative way. On the basis of this statement and on the experts' comments during the review, we have revised the deliverable D2.1 by including a new Section "Comparative analysis of the results" where we present (1) a statistical evaluation of our technique with respect to other "generalized" spatial/spatio-temporal averaging operations, and (2) a comparison with the state-of-the-art. This Section replaces the Section "Evaluation of recurrent regularization" of the previous version of the deliverable. In addition, we have extended the description of our method and we have included a quantitative characterization of the approximation error of the linear templates obtained by the recurrente generative model. In the new Section "Discussion" a more general comparison of our approach with other related scenarios is presented.

2 General introduction

The main objective of this workpackage is to arrive at a reliable low-level information representation. In this workpackage we address the study of integrative adaptive schemes, mainly along three strategies:

1. investigation of recurrent processing schemes to regularize the single-modality estimations

2. low-level cross modality interaction

3. integration of non-visual signals

Deliverable 2.1 is about recurrent processing for low-level feature regularization. The input to the computational modules developed in Task 2.1 will be the feature maps calculated by the FPGA-based front-end vision system. The improvement of the accuracy/density of these maps is an important step to help high level elaboration, which will occur in WP3 and WP4.

When low level features are noisy an adaptive recurrent filter can be used to perform a spatial regularization. In this deliverable we focus our attention to the regularization of optic flow because it most suffers of problems of sparsity and unreliability (mainly due to the aperture problem). Furthermore, when low level feature are calculated with constrained computation precision (as it is the case in the hardware front-end), the level of noise in the extracted map increases significantly with respect to the software model (with no precision limitations and floating point arithmetic). The deliverable is organized as follows: in Section 3 we present the model (patch-wise linear description of the optic flow and Kalman-based recurrent filtering). In Section 4 we frame our approach with respect to other techniques in the literature and we comparatively assess the results on the basis of standard benchmark sequences. General discussion of the approach is presented in Section 5 and conclusive remarks are summarized in Section 6.

3 Summary

Assuming to have a noise optic flow from the front-end vision module, a context-sensitive filters for regularizing the optic flow is presented. Measured optic flow fields are indeed always somewhat erroneous and/or ambiguous: First, we cannot compute the actual spatial or temporal derivatives, but only their estimates, which are corrupted by image noise. Secondly, optic flow is intrinsically an image-based measurement of the relative motion between the observer and the environment, but we are interested in estimating the actual motion field. (2) Third, optic flow estimates calculated with restricted precision suffers from higher levels of noise (as computation errors). However, real-world motion field patterns contain intrinsic properties that allow to define model structures as groups of pixels sharing the same motion property. By checking the presence of such structures in optic flow fields we can make their interpretation more confident. We propose an optimal recurrent filter capable of evidencing linear motion structures corresponding to 1st-order spatial derivatives or elementary flow components (EFCs). The regularization of the optic flow emerges from a noisy flow as a solution of an iterative process of spatially interacting nodes that correlates the properties of the visual context with that of a structural model of the EFC.

The context-sensitive filter behaves as a template model. Yet, its specificity lies in the fact that the template character is not built by highly specific feed-forward connections, but emerges by stereotyped recurrent interactions (cf. the process equation). Furthermore, the approach can be straightforwardly extended to consider adaptive cross-modal templates (e.g, motion and stereo). By proper specification of the matrix Φ, the process equation can, indeed, potentially model any type of cross-modal spatio-temporal relationships (i.e., cross-modal spatio-temporal context).

The results we obtain are compared with classical post-processing techniques, such as spatial/spatio-temporal filtering, median, Wiener filtering, working on the same spatial neighborhood. Our approach shows, in general, better performances than classic generalized averaging operation. From a more general perspective, the proposed model is conceptually formalized as a recurrent solution of a variational problem that combines local and global constraints. Accordingly, systematic comparison with state-of -the-art techniques based on similar principles are presented. From the comparison, it emerges that the performances of our technique (in general worse than the others) highly depends on the robustness of the (phase-based) initial solution, which (in its current implementation) is dramatically sensitive to noise. It is worth to note that the approach we followed to obtain the initial solution is dictated by the choice made for the FPGA implementation. A fairer comparison of the context regularization approach would require changing the method for obtaining the initial (non regularized) solution, which is yet beyond the scope of WP2 and of this report.

In conclusion we can state that: (1) if the optic flow available from the hardware module is sufficiently reliable it is not convenient to implement the Kalman Filter (KF) regularization algorithm since almost equivalent results can be obtained by spatial averaging at a less computational cost; (2) the patch-wise linear description of the optic flow allows us to obtain a compact representation of the optic flow usable for the extraction of high level visual cues (SVEs). To conclude on the opportunity of a cascade KF regularization we thus need to wait for realist optic flow fields from the FPGA, whereas the linear optic flow description available

by the affine coefficients $c_1, \ldots, c_6$ is a general asset of the approach.

4 A context-sensitive recurrent filter for optic flow regularization

The regularization of optic flow is a complex problem for which many solutions/approaches have been proposed. Among them, most of the techniques include the regularization constraint in the extraction itself of the optic flow [1, 2, 3, 4, 5], whereas the approaches that post-process the optic flow (as the one here proposed) are very seldom [6]. It is worth noting that the choice of operating in cascade to a rough extraction of the optic flow is dictated by the expected necessity of improving the quality of the optic flow provided by the FPGA-based front-end early vision module where a basic version of a phase-based optic flow algorithm will be implemented (cf. precision constraints, such as fixed point arithmetics, warping of images vs. warping of phase maps, ...). From this perspective, here we propose an adaptive recurrent spatial (spatiotemporal) filtering based on Kalman Filter (KF) [7] [8] to remove, or, at least, to reduce the uncertainty associated to a local measure of the optic flow, by making use of *contextual* information that capture coherent properties of velocity vectors over large overlapping image regions (patches). A comparative analysis of our approach with other (classes of) regularization methods are presented in Section 4.

4.1 Kalman-based adaptive filtering

A schematic diagram of the regularization process of the optic flow, represented as an adaptive filter, is shown in Fig. 1: $\boldsymbol{v}^*[k]$ is the real image motion field (the *unknown* stimulus/state) at time step k, $\mathbf{v}[k] = A\boldsymbol{v}^*[k]$ is the oberved optic flow (the *measure*), $\hat{\boldsymbol{v}}[k]$ is the estimated motion field, and $\boldsymbol{v}[k]$ is the reference signal (i.e., what we know about $\boldsymbol{v}^*[k]$). The purpose

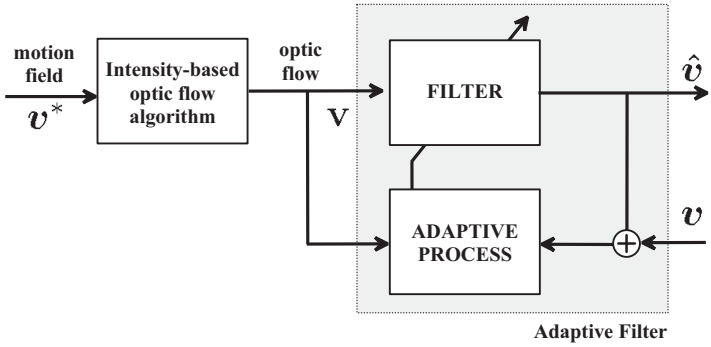


Figure 1: Schematic representation of adaptive early vision filters.

of the adaptive system is to filter the input signal $\mathbf{v}[k]$ (measure) to invert (in some sense) the measure operator and gain an estimation of the solution of $\boldsymbol{v}^*[k]$ by making use of the knowledge $\boldsymbol{v}[k]$.

The filter evolves in time and it takes some time (convergence time) to "learn" to act as the inverse operator A^{-1}, by embedding information about the unknown stimulus in its structure (cf. the learning phase of classical neural networks). After convergence, if the reference signal

is sufficiently representative of the unknown signal the filter could work in an open-loop configuration (without further adaptation) on the basis of what it has already known of the model. In this condition, new measures are not further used to refine the estimation that will be considered as an *a priori* estimate. The closed-loop configuration, on the contrary, guarantees a continuous adaptation leading to *a posteriori* state estimates. In the following, we will see that this distinction between *a priori* and *a posteriori* estimates will be more evident in the formulation of the Kalman filter due to its intrinsic recurrent nature by which the *a priori* estimation (based on the previous experience) is corrected by actual measure to give rise to the *a posteriori* estimate. Kalman Filter indeed does not need to store the entire history of the process, because it only needs the previous estimate. Moreover, by using Kalman Filter we have not only an estimate of the measure (in this case optic flow), but also an associated variance and an uncertainty map that is propagated at each step. For these reasons Kalman Filter can be used to regularize noisy optic flow.

Formally, the KF is characterized by two inputs:

the *process equation*

$$\boldsymbol{x}[k] = \boldsymbol{\Phi}[k, k-1]\ \boldsymbol{x}[k-1] + \boldsymbol{S}[k-1]\ \boldsymbol{s}[k-1] + \boldsymbol{n}_1[k-1] \tag{1}$$

and

the *measurement equation*

$$\boldsymbol{y}[k] = \boldsymbol{C}[k]\ \boldsymbol{x}[k] + \boldsymbol{n}_2[k] \tag{2}$$

The matrix $\boldsymbol{\Phi}[k, k-1]$ is a known state transition matrix that relates the state $\boldsymbol{x}[k-1]$ at the previous time step $k-1$ to the state $\boldsymbol{x}[k]$ at the current step k. The matrix $\boldsymbol{S}[k]$ takes into the account an optional control input to the state. The matrix $\boldsymbol{C}[k]$ is a known measurement matrix. The process and measurement uncertainty are represented by $\boldsymbol{n}_1[k] = N(0, \boldsymbol{\Lambda}_1[k])$ and $\boldsymbol{n}_2[k] = N(0, \boldsymbol{\Lambda}_2[k])$ The space spanned by the observations $\boldsymbol{y}[1], \boldsymbol{y}[2], \cdots, \boldsymbol{y}[k-1]$ is denoted by $\boldsymbol{\mathcal{Y}}_{k-1}$.

Assuming $\boldsymbol{x}$ a vector containing the values of a bunch of visual features over a fixed spatial region, Eq. 1 models the temporal evolution of the relationships among such features, according to specific rules embedded in the transition matrix $\boldsymbol{\Phi}$. For example, if we consider just one feature (e.g., motion velocity), $\boldsymbol{x}[k]$ will represent the "model" optic flow values at time step k, for all the (discrete) locations of the considered spatial regions (the velocity state). If $\boldsymbol{\Phi}$ has a diagonal structure, the process equation will describe the temporal evolution "model" of punctual velocities, independently of the spatial neighborhood values (temporal context). On the other hand, if $\boldsymbol{\Phi}$ shows a non-diagonal structure, the process equation models a temporal evolution "model" of the state that takes into account *also* spatial relationships (spatio-temporal context). More generally, if we build a state vector that collects more multiple features (e.g., motion, stereo, etc.), by proper specification of the transition matrix $\boldsymbol{\Phi}$, the process equation can potentially model any type of *multimodal* spatio-temporal relationships (multimodal context).

4.2 Affine (linear) models of image motion

In order to use Kalman Filter we need to define a process equation that describes the motion field. We can describe motion flow fields in terms of their linear decompositions, on the basis of their first-order (linear) properties. From this perpsective, local spatial features around a given location of flow field, can be of two types: (1) the average flow velocity at that location, and (2) the structure of the local variation in a neighborhood of that locality. The former relates to the smoothness constraint or structural uniformity. The latter relates to linearity constraint or structural gradients [9]. Velocity gradients provide important cues about the 3D layout of the visual scene. Formally, they can be described as linear deformations by a first-order Taylor decomposition (for further details see the enclosed paper in the Appendix):

$$\boldsymbol{v} = \bar{\boldsymbol{v}} + \bar{\mathbf{T}}\boldsymbol{x} \tag{3}$$

where

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = \begin{bmatrix} \partial v_x/\partial x & \partial v_x/\partial y \\ \partial v_y/\partial x & \partial v_y/\partial y \end{bmatrix} . \tag{4}$$

By breaking down the tensor in its dyadic components, the motion field can be locally described through two-dimensional maps ($\boldsymbol{f} : \mathcal{R}^2 \mapsto \mathcal{R}^2$) representing elementary flow components (EFCs):

$$\boldsymbol{v} = \boldsymbol{\alpha}^x \bar{v}_x + \boldsymbol{\alpha}^y \bar{v}_y + \boldsymbol{d}_x^x \left.\frac{\partial v_x}{\partial x}\right|_{\mathrm{x}_0} + \boldsymbol{d}_y^x \left.\frac{\partial v_x}{\partial y}\right|_{\mathrm{x}_0} + \boldsymbol{d}_x^y \left.\frac{\partial v_y}{\partial x}\right|_{\mathrm{x}_0} + \boldsymbol{d}_y^y \left.\frac{\partial v_y}{\partial y}\right|_{\mathrm{x}_0} \tag{5}$$

where $\boldsymbol{\alpha}^i$ are pure translations and $\boldsymbol{d}_j^i$ represent cardinal deformations, basis of the linear deformation space. In Figure 2 elementary flow components are shown. The EFCs can be combined to obtain deformation subspaces representing elementary deformations such as expansion, shear and rotation (see Figures 3, 4 and 5).

It is worthy to note that the components of pure translations could be incorporated in the corresponding deformation components, thus obtaining *generalized deformation components* in which motion boundaries are shifted or totally absent:

$$\begin{aligned} \boldsymbol{v}_x^x &= a_1\boldsymbol{\alpha}^x + a_2\mathbf{d}_x^x \ , \boldsymbol{m}_1 \\ \boldsymbol{v}_y^x &= a_3\boldsymbol{\alpha}^x + a_4\mathbf{d}_y^x \ , \boldsymbol{m}_2 \\ \boldsymbol{v}_x^y &= a_5\boldsymbol{\alpha}^y + a_6\mathbf{d}_x^y \ , \boldsymbol{m}_3 \\ \boldsymbol{v}_y^y &= a_7\boldsymbol{\alpha}^y + a_8\mathbf{d}_y^y \ , \boldsymbol{m}_4 \end{aligned} \tag{6}$$

In this way, we have four classes of deformation gradients: one stretching ($\boldsymbol{v}_i^i$) and one shearing ($\boldsymbol{v}_j^i$) for each cardinal direction. As it will be clear in the following, this choice gives to the model maximum flexibility: every gradient deformation within a single class will be built through the same recurrent network, just by changing its driving inputs on the basis of direct local measures on the input optic flow. Figure 6 shows the four classes of deformation gradients.

It is worthy to note that Eqs. (5) and (6) describe, in fact, an affine model:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} c_1 \\ c_4 \end{bmatrix} + \begin{bmatrix} c_2 & c_3 \\ c_5 & c_6 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{7}$$

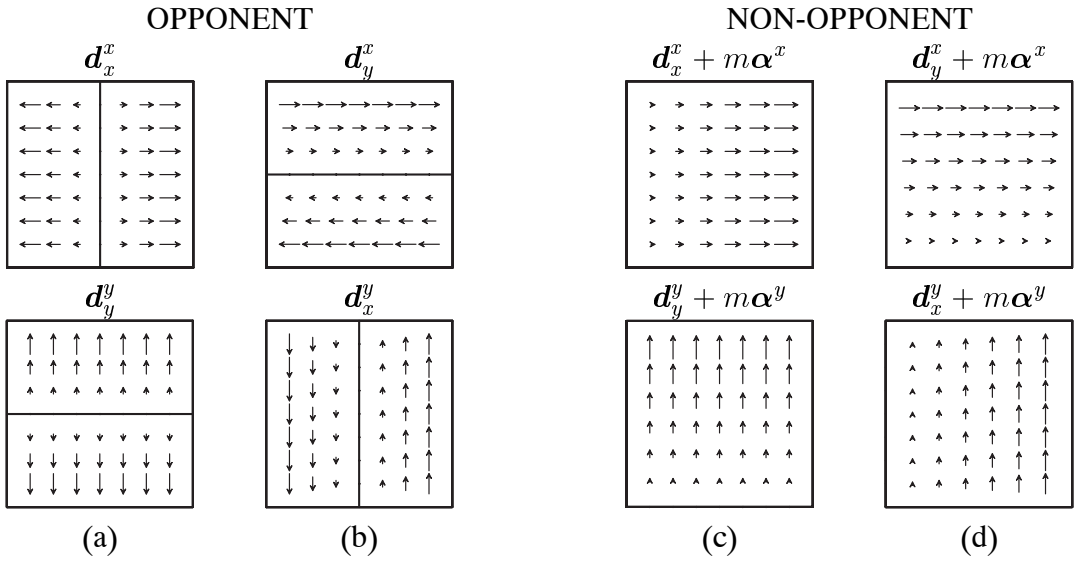


Figure 2: Elementary flow components showing different gradient types. In stretching-type components (a,c) velocity varies *along* the direction of motion; in shearing-type components (b,d) velocity gradient is oriented *perpendicularly* to the direction of motion. Non-opponent patterns are obtained from the opponent ones by a linear combination of pure tranlations and cardinal deformations: $\boldsymbol{d}_j^i + m\boldsymbol{\alpha}^i$, where m is a proper positive scalar constant.

where c_i are constants and v_x and v_y are the horizontal and the vertical components of the flow[1]. The parameter vector $[c_1, c_2, \ldots, c_6]$ describes a specific configuration of optic flow that locally provides a good approximation of 3D rigid moving objects. The six parameter affine model is reasonable to describe the motions of smooth surface in small image regions. The parameters c_i have qualitative interpretations in terms of image motion, for example c_1 and c_4 represent horizontal and vertical translation and we can express divergence (isotropic expansion), curl (rotation about the viewing direction), and the two components of deformations as combination of the c_i's:

$$\begin{aligned} div &= c_2 + c_6 \\ curl &= c_3 - c_5 \\ def_1 &= c_3 + c_5 \\ def_2 &= c_2 - c_6 \end{aligned} \tag{8}$$

4.3 Regularization by adjustable linear templates

4.3.1 Generative models

The templates that approximate the deformation components can be generated recursively by using a lattice network:

$$\boldsymbol{v}[k] = \boldsymbol{\Phi}[k, k-1]\boldsymbol{v}[k-1] + \boldsymbol{n}_2[k-1] + \boldsymbol{s}[k-1]\ , \tag{9}$$

[1] $c_1 = a_1 + a_3, c_2 = a_2, c_3 = a_4, c_4 = a_5 + a_7, c_5 = a_6, c_6 = a_8$.

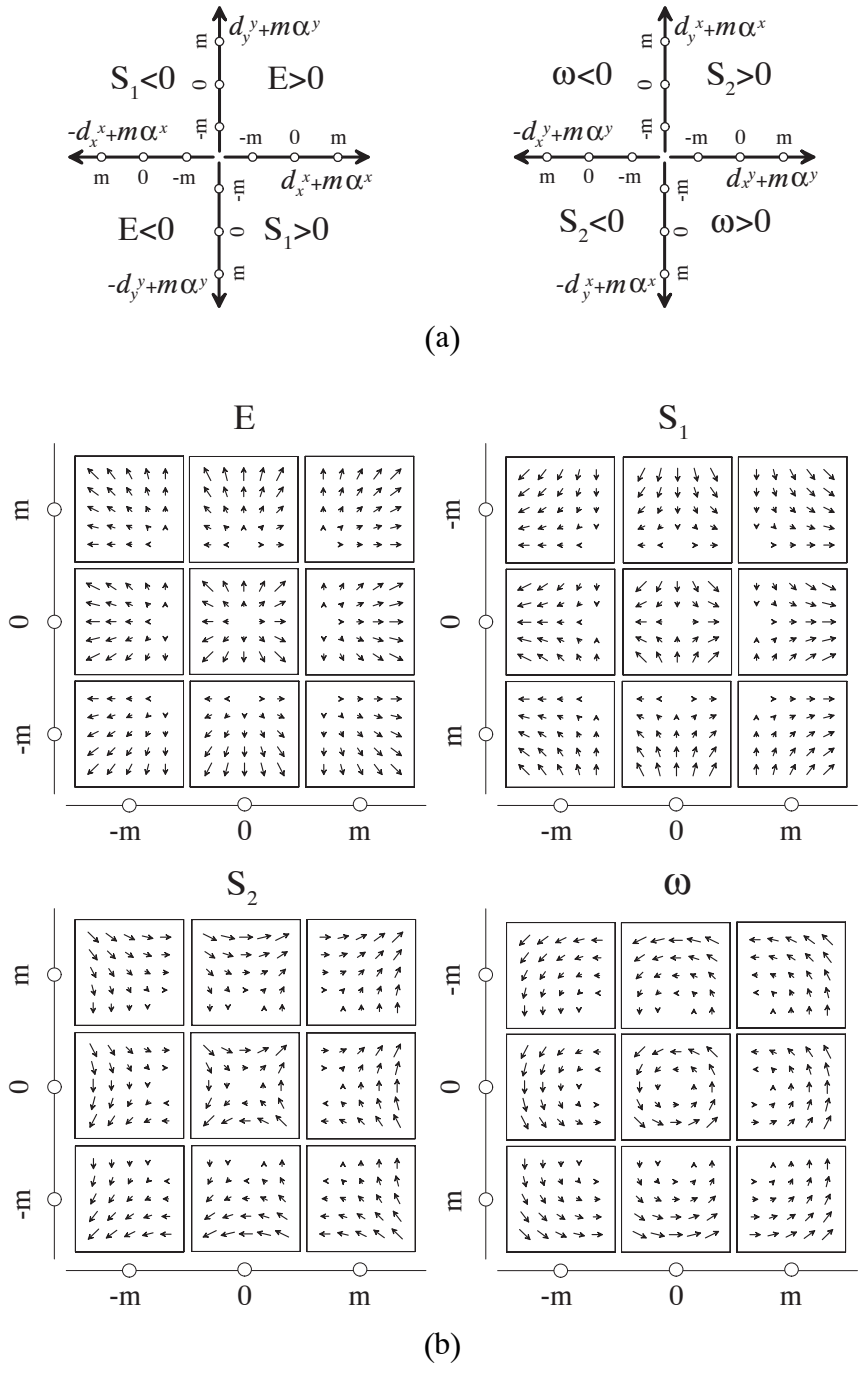


Figure 3: (a) Two deformation subspaces obtained by the set of cardinal EFCs with different values of the parameter m. The quadrants of each subspace characterize an elementary deformation, as evidenced in (b) for expansion ($E > 0$), horizontal positive shear ($S_1 > 0$), oblique positive shear (S_2), and counterclockwise rotation ($\omega > 0$).

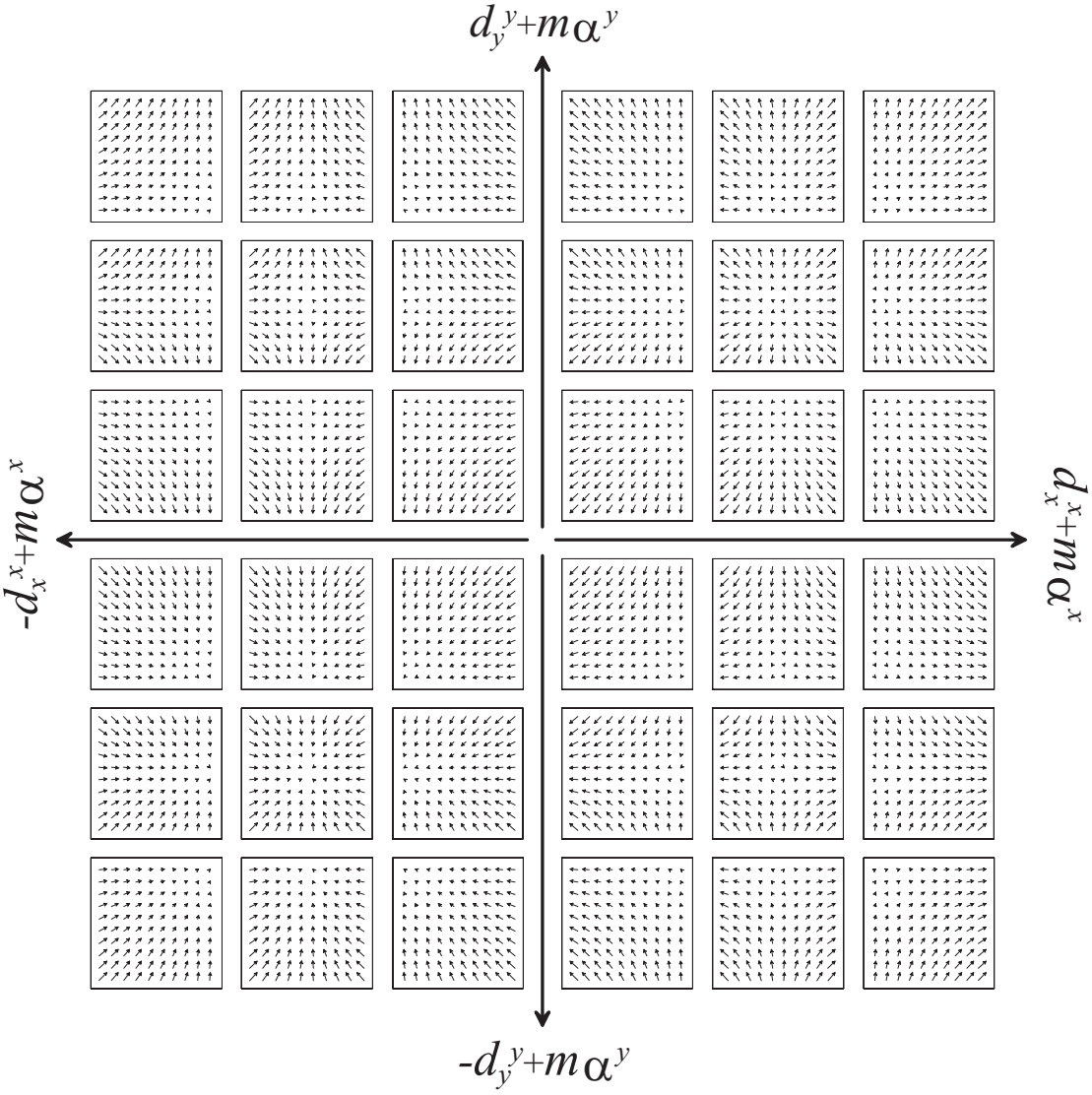


Figure 4: The $E - S_1$ deformation subspace.

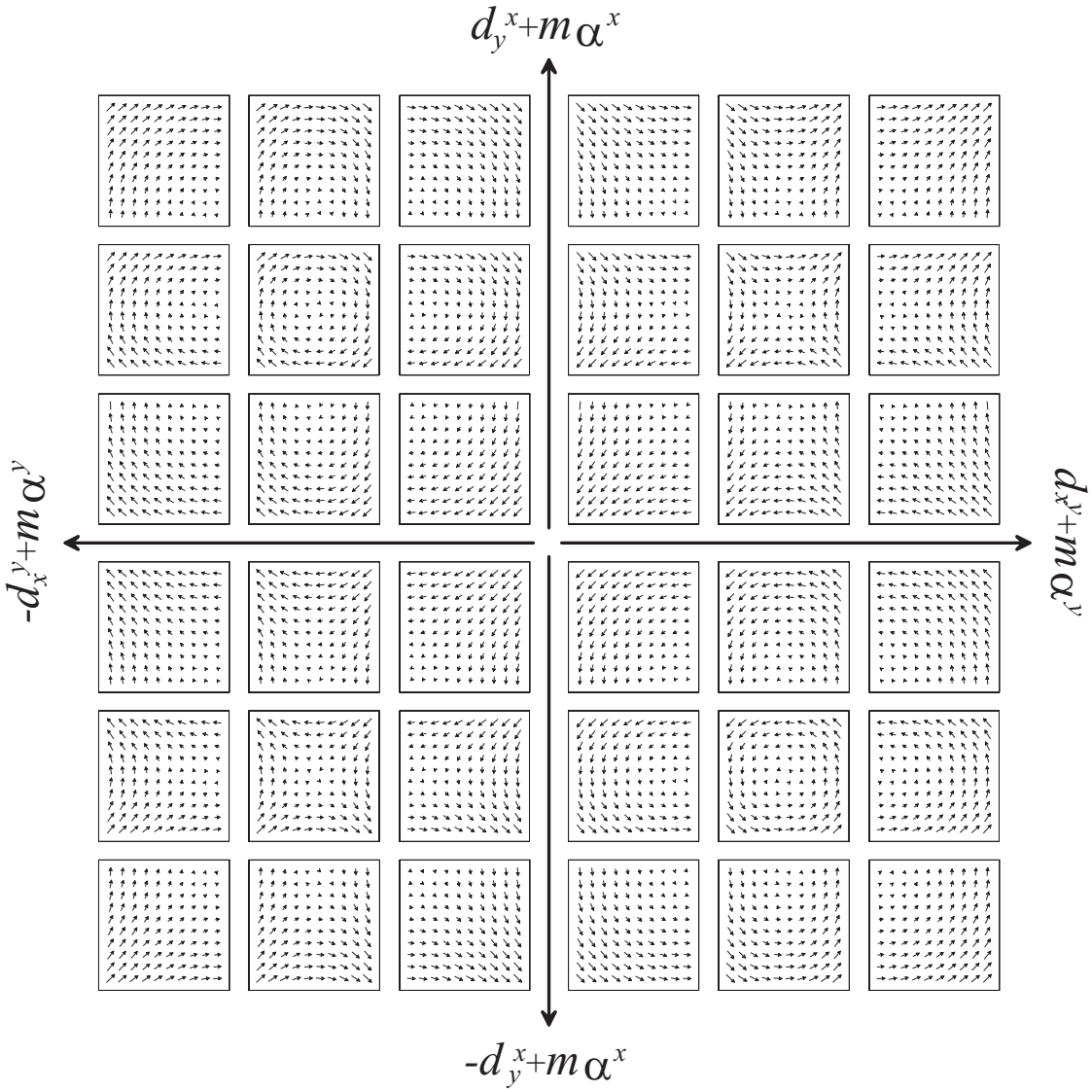


Figure 5: The $\omega - S_2$ deformation subspace.

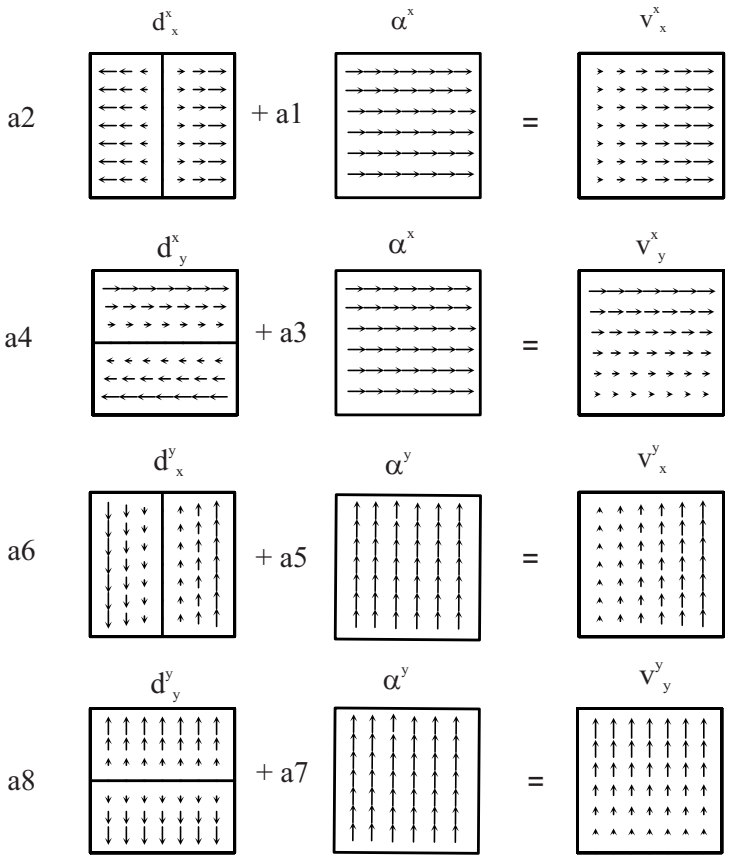


Figure 6: The generalized deformation components ($\boldsymbol{v}_x^x$, $\boldsymbol{v}_y^x$, $\boldsymbol{v}_x^y$, $\boldsymbol{v}_y^y$) are obtained by incorporating the pure translations in the corresponding cardinal deformations.

which describes the temporal evolution, from the previous time step $k-1$ to the current time step k, of the relationships among motion features over a fixed small spatial region $[-L, L] \times [-L, L]$ according to specific rules embedded in the transition matrix $\boldsymbol{\Phi}$. The driving input $\boldsymbol{s}[k]$, evaluated at each time step, by computing the average of the optic flow velocity components at the patch's borders, can be interpreted as the boundary conditions of the lattice network (see Fig. 7), whereas $\boldsymbol{n}_2[k]$ represents the process noise.

It is worth noting that the spatial interactions occur separately for each component of the velocity vectors through 1D nearest neighbor interactions. More precisely, given the difference equation that describes the nearest neighbor cooperation among the spatial nodes n's for the generic velocity component v: $A_{-1}v(n-1) + A_0 v(n) + A_1 v(n+1) = 0$, and solving it with the boundary conditions $v(-L) = \lambda$ and $v(L) = \mu$, we obtain the velocity profiles that approximate the linear templates parameterized by the coefficients a_i and c_i:

$$v(n) = \frac{\mathrm{e}^{-\alpha \mathrm{L}}}{1 - \mathrm{e}^{-4\alpha \mathrm{L}}} \left[(\lambda - \mu \mathrm{e}^{-2\alpha \mathrm{L}}) \mathrm{e}^{-\alpha \mathrm{n}} + (\mu - \lambda \mathrm{e}^{-2\alpha \mathrm{L}}) \mathrm{e}^{\alpha \mathrm{n}} \right] \tag{10}$$

where $\lambda = a_i - Lc_i$ and $\mu = a_i + Lc_i$, and with α depending on the coupling coefficient $A_1 = A_{-1}$ of the 1D lattice network. By a proper choice of the coupling coefficients and of the boundary values λ and μ the velocity profiles result approximately linear (see Fig. 8, Fig. 9

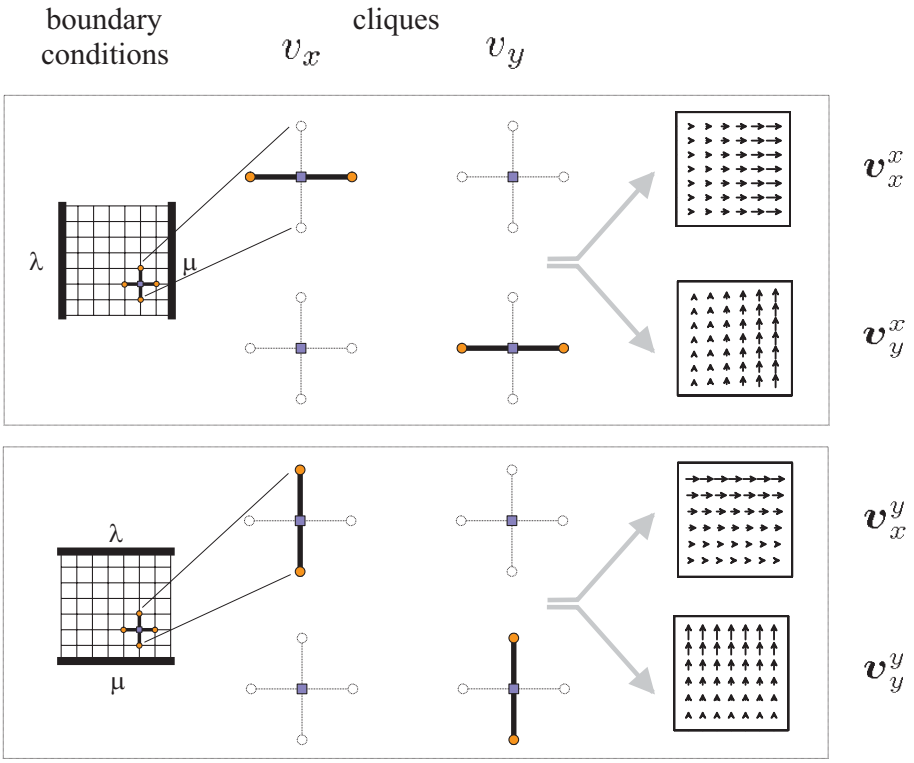


Figure 7: Basic lattice interconnection schemes for the generation of the adjustable linear templates. The lattice networks have a *structuring effect* constrained by the boundary conditions that yields to structural equilibrium configurations, characterized by the specific first-order EFCs. The resulting velocity patterns depend on the directions of the interaction scheme and on the boundary conditions. $\boldsymbol{v}_x^x$ and $\boldsymbol{v}_y^y$ represent the stretching components, whereas $\boldsymbol{v}_y^x$ and $\boldsymbol{v}_x^y$ represent the shearing components. The boundary values λ and μ control the gradient slope and the constant term.

and Fig. 10). To quantify the approximation error, we calculated, as a function of α and L, the average relative integral error between the solution of the lattice network (see Eq. 10) and a straight line that joins the values at the boundaries (λ and μ). In general, for any combination of λ and μ, the larger is the size of the patch, the higher is the approximation error. Though, it is possible to choose the proper value of α to keep the error below an arbitrary tollerance value. Fig. 11(a) shows the curves of constant error ($\epsilon = 0.01$), for different combinations of λ and μ. Fig. 11(b), (c) and (d) show the variability of the approximation error by varying the boundary values λ and μ for a fixed size of the template ($L = 3$, $L = 6$ and $L = 12$, respectively) and for a fixed value of $\alpha = 0.19$. The limited increase of the error over a wide variation of the boundary values in the range of ± 30 pixel/frame demonstrates the validity of the approximation of the linear templates by the generative models.

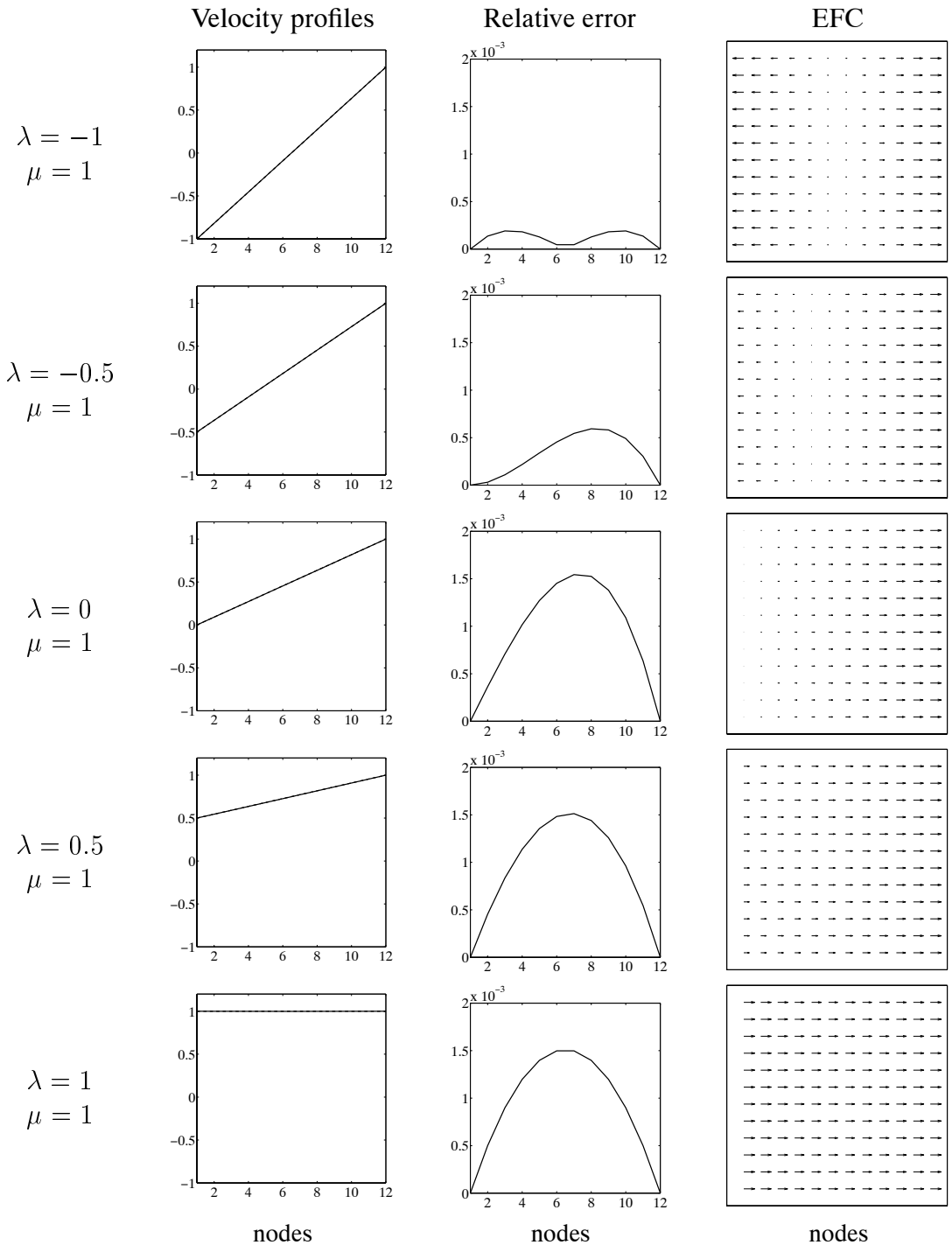


Figure 8: The generative model outputs that approximate the linear templates. Parameters of the 1D lattice network: $\alpha = 0.01$ and $A_1 = A_{-1} = 0.4999$. (Left) The velocity profiles describe the solution of the lattice network (solid line) and the straight line (dashed line) that joins the values at the boundaries (λ and μ). (Middle) The relative error as a function of the spatial support between the solution of the lattice network and the previous straight line. (Right) The template, that locally approximates a generalized deformation component, generated recursively by using the lattice network with the shown parameters.

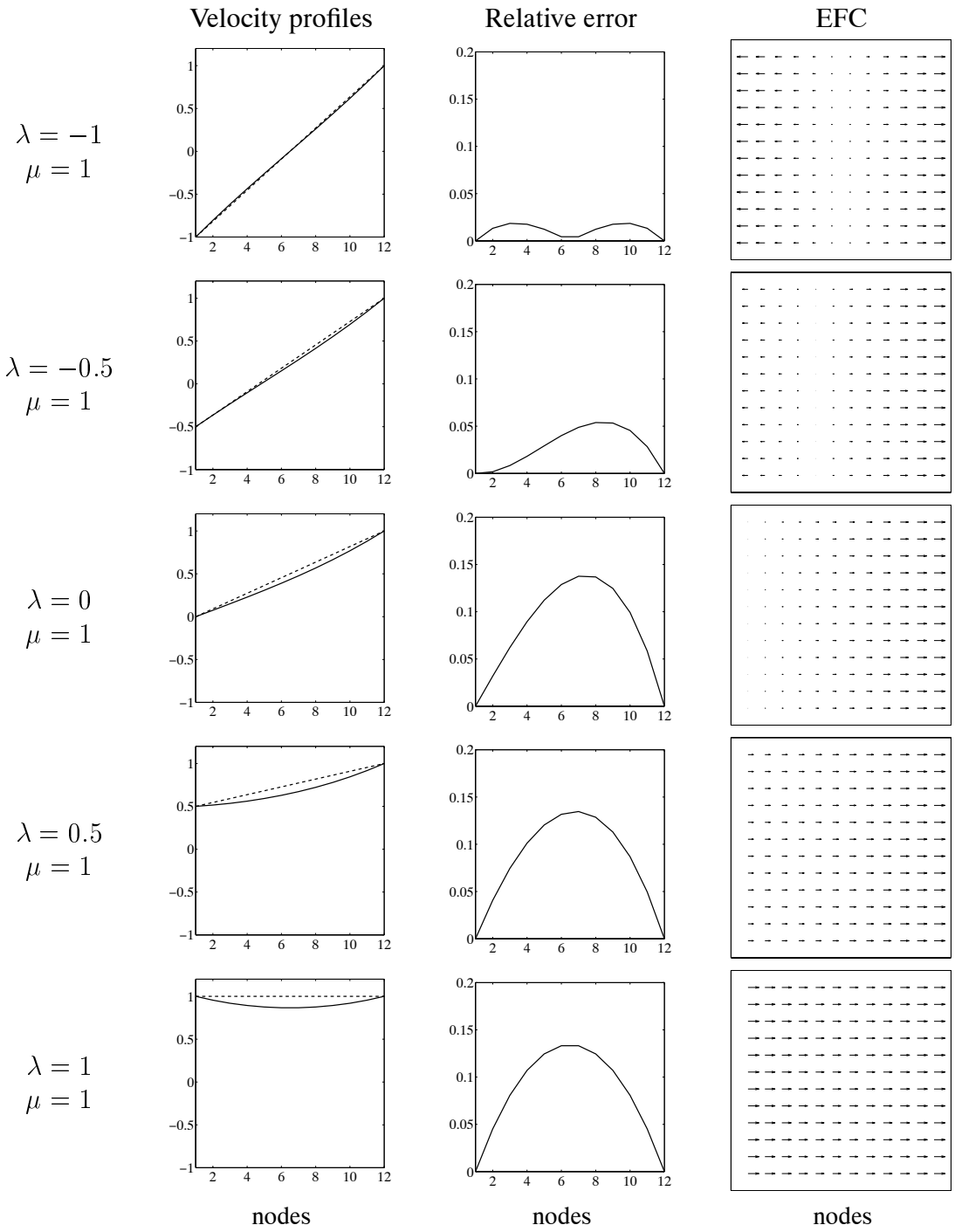


Figure 9: The generative model outputs that approximate the linear templates. Parameters of the 1D lattice network: $\alpha = 0.10$ and $A_1 = A_{-1} = 0.4976$.. (Left) The velocity profiles describe the solution of the lattice network (solid line) and the straight line (dashed line) that joins the values at the boundaries (λ and μ). (Middle) The relative error as a function of the spatial support between the solution of the lattice network and the previous straight line. (Right) The template, that locally approximates a generalized deformation component, generated recursively by using the lattice network with the shown parameters.

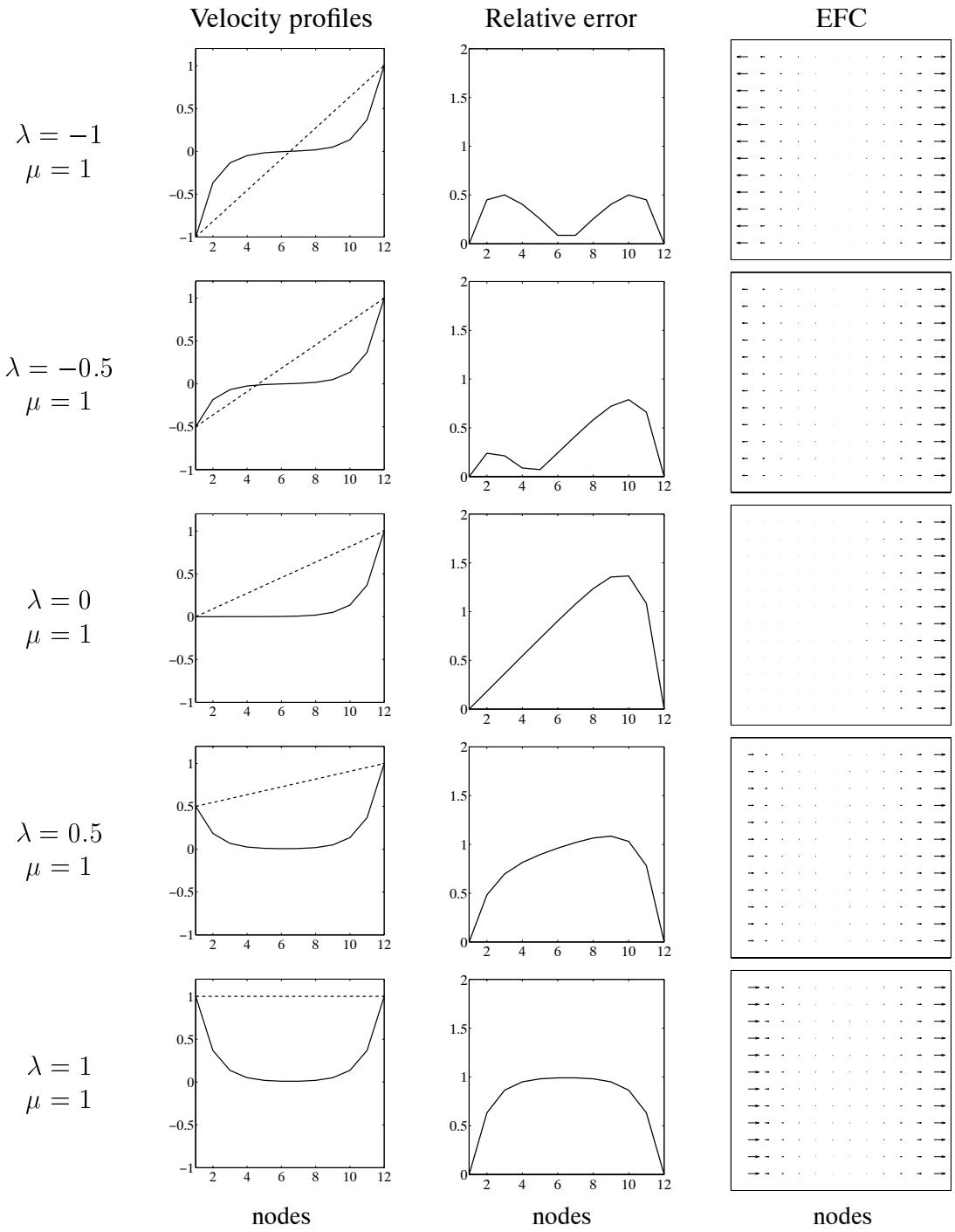


Figure 10: The generative model outputs that approximate the linear templates. Parameters of the 1D lattice network: $\alpha = 1.00$ and $A_1 = A_{-1} = 0.3240$.. (Left) The velocity profiles describe the solution of the lattice network (solid line) and the straight line (dashed line) that joins the values at the boundaries (λ and μ). (Middle) The relative error as a function of the spatial support between the solution of the lattice network and the previous straight line. (Right) The template, that locally approximates a generalized deformation component, generated recursively by using the lattice network with the shown parameters.

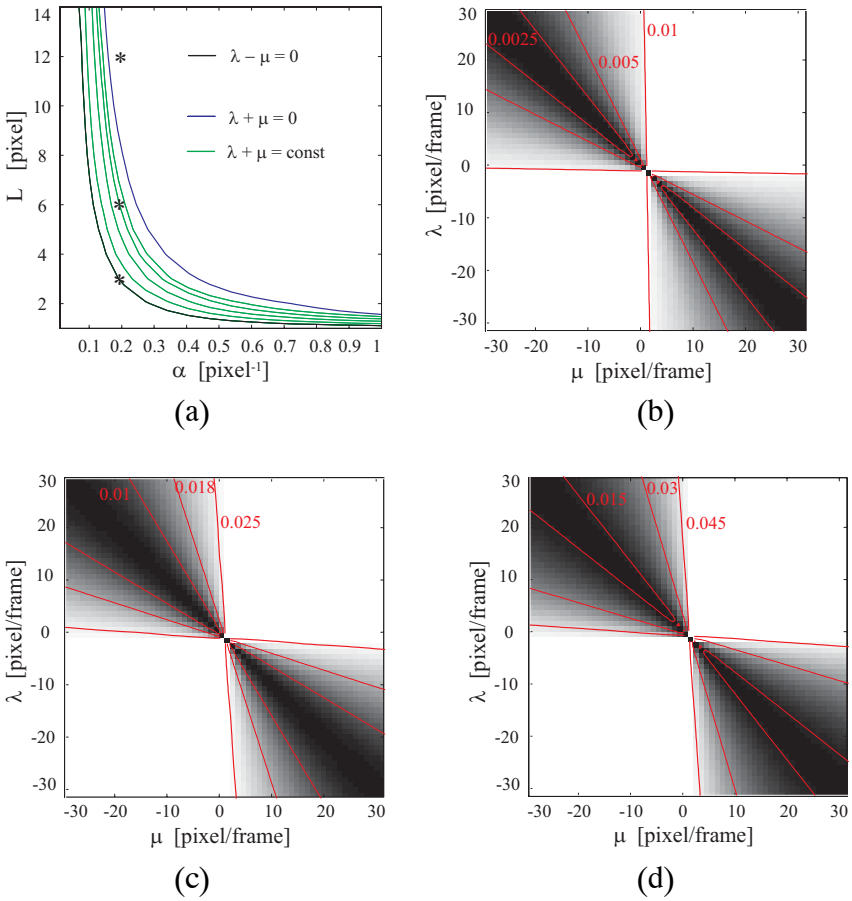


Figure 11: Variations of the average relative integral error for different values of the network parameters. (a) Relationships between the size of the patch L and the diffusive coefficient of the lattice network α for a constant value of the approximation error ($\epsilon = 0.01$) and for different combinations of the boundary values λ and μ. (b)-(d) Variation of the error for the three pairs of L and α evidenced with asterisks (a): $\alpha = 0.19$ and $L = 3$, $L = 6$, $L = 12$, respectively.

4.3.2 Adaptive filtering

The adjustable templates defined in the previous Section can be used as models for a multiple model Kalman filter (KF) to measure the structural properties of the input optic flow. The output of the KF will be the estimate of the motion field on the basis of the spatial contextual information described by the generative models of the EFCs. Since the models are continuously adapted to the measures by changing the boundary conditions for every patch, and the KF interatively integrates the new measures with the knowledge about the motion pattern obtained by the previous measurements, we obtain adaptive estimates of the EFCs. In this way, we perform an adaptive template matching capable of tracking the coefficients of a linear description/approximation of the optic flow.

Formally, given a measurement equation:

$$\mathbf{v}[k] = \mathbf{C}[k]\boldsymbol{v}[k] + \boldsymbol{n}_1[k] \tag{11}$$

where $\mathbf{v}[k]$ is noisy measure, at current time k, of the actual motion field $\boldsymbol{v}[k]$, $\boldsymbol{n}_1[k]$ is the uncertainty of the measure, and $\mathbf{C}$ is a modified unitary operator for discarding the image points where the optic flow is not available or not reliable, the output of the filter will be:

$$\hat{\boldsymbol{v}}[k|\mathbf{V}_k] = \hat{\boldsymbol{v}}[k|\mathbf{V}_{k-1}] + \boldsymbol{G}[k]\boldsymbol{\nu}[k] \tag{12}$$

where: $\hat{\boldsymbol{v}}[k|\mathbf{V}_{k-1}]$ is the *a priori* state estimate, $\hat{\boldsymbol{v}}[k|\mathbf{V_k}]$ is the *a posteriori* state estimate, $\mathbf{V}_k$ represent all the measurements until time steps k, $\boldsymbol{\nu}[k] = \mathbf{v}[k] - \hat{\mathbf{v}}[k|\mathbf{V}_{k-1}]$ is the innovation and $\boldsymbol{G}[k]$ is the Kalman gain. In order to have a statistical measure the discrepancy between predictions and observations, as an indication of the filter's consistency, it is frequently used the Normalized Innovation Squared (NIS) [10]:

$$\mathrm{NIS}_k = \boldsymbol{\nu}^T[k]\boldsymbol{S}^{-1}[k]\boldsymbol{\nu}[k] \tag{13}$$

defined on the basis of the innovation and on its covariance $\boldsymbol{S}$. Since the covariance of the innovation depends on the estimate of the measure noise $\boldsymbol{n}_1$, it is important to have a reliable estimate of the noise in the measure. Thus, the noise covariance matrices are tuned on the basis of the differences (in terms of the mean angular error [11]) of the velocity values measured inside a patch between two consecutive frames. Where the optic flow smoothly changes in time, the measure noise $\boldsymbol{n}_1$ remains low, whereas, where optic flow changes more abruptly, the noise becomes higher and the estimates have a lower confidence. In the multiple model KF the NIS value is used to compute, for each model, the likelihood of the measurements, on which to base the selection among the different models. This choice varies continuously while the filter is operating. In such a case, we cannot make a fixed *a priori* choice of the filter's parameters, but we have to use a continuously varying model-conditioned combination of the candidate state and error covariance estimates. It is worth noting that, in our dynamic multiple model approach, we do not want the probabilities to converge to fixed values, but we want them to be free to change at each new measurement. In the multiple model approach [10] it is assumed that the system obeys one of a finite number of models $\boldsymbol{m}_i$ with $i = 1, 2, \ldots, r$ (with $r = 4$, in our case, corresponding to the four classes of deformation gradients). The likelihood of the measurement $\mathbf{v}$ given a model $\boldsymbol{m}_i$ at time step k is given by:

$$f(\mathbf{v}|\boldsymbol{m_i}) = |\boldsymbol{2\pi S}_{\mathrm{m_i}}|^{-\frac{1}{2}} \mathbf{e}^{-\frac{1}{2}\,{}^{\circ}_{\mathrm{m_i}}\boldsymbol{S}^{-1}_{\mathrm{m_i}}\nu_{\mathrm{m_i}}} \tag{14}$$

where $\boldsymbol{m}_i$ is the considered model. The probability that the candidate model $\boldsymbol{m}_i$ is the correct one is given by the following equation:

$$p_{\mathrm{m}_i}[k] = \frac{f(\mathbf{v}|\boldsymbol{m_i})}{\sum_{j=1}^{r} f(\mathbf{v}|\boldsymbol{m_j})}, \tag{15}$$

with $p_{\mathrm{m}_i}[0] = 1/r, i = 1, 2, \ldots, r$ and $\sum_{i=1}^{r} p_{\mathrm{m}_i}[k] = 1$ at each time step k.

With this approach the probability value approaches 1 when the optic flow has the same structure of the model. None of the models gives a high probability value if none of the EFCs is present in the optic flow. In this way, noisy and unstructured motions are automatically discarded. The final model-conditioned estimate of the state $\boldsymbol{v}$ is computed as a weighted combination of the *a posteriori* states of each candidate filter:

$$\hat{\boldsymbol{v}}[k] = \sum_{i=1}^{r} p_{\mathrm{m}_i}[k]\hat{\boldsymbol{v}}_{\mathrm{m}_i}[k]. \tag{16}$$

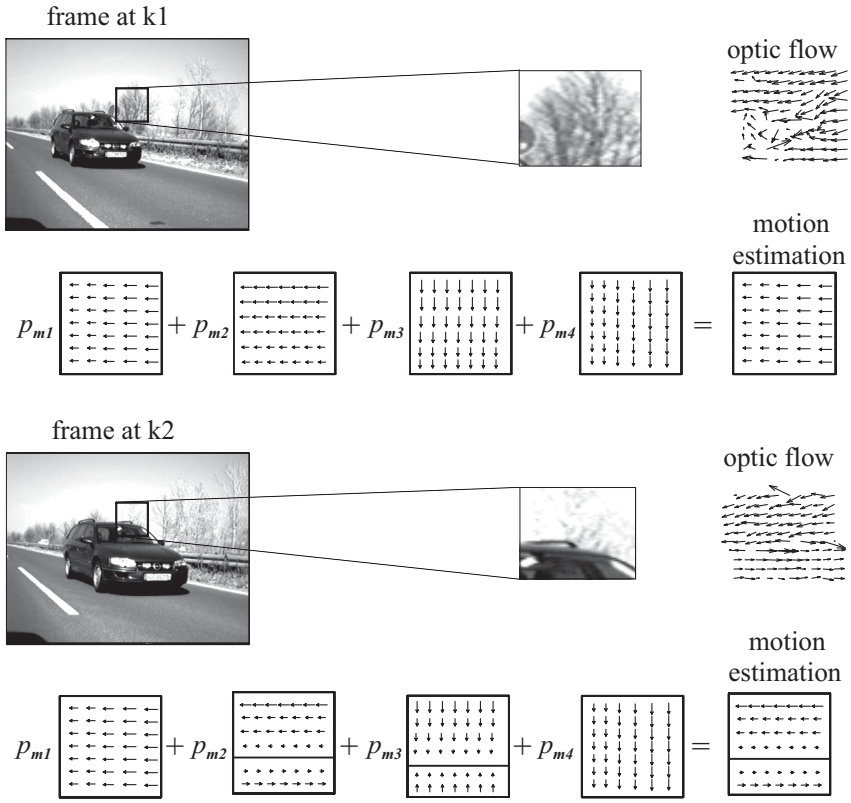


Figure 12: Multiple-model motion estimation in a road scene taken by a rear-view mirror of a moving car under an overtaking situations. The model-based decompositions are evidenced for the same image patch for two different frames at time $k1$ and $k2$. For each optic flow patch the motion is estimated from the actual generalized deformation components weigthed by the corresponding probability values, see Eq.(16).

For the 4 models considered (see Eqs. (6)):

$$\hat{\boldsymbol{v}} = p_{\mathrm{m}_1}\hat{\boldsymbol{v}}_x^x + p_{\mathrm{m}_2}\hat{\boldsymbol{v}}_y^x + p_{\mathrm{m}_3}\hat{\boldsymbol{v}}_x^y + p_{\mathrm{m}_4}\hat{\boldsymbol{v}}_y^y \tag{17}$$

where p_{m_1}, p_{m_2}, p_{m_3}, p_{m_4} are the probabilities related to each model and $\hat{v}_x^x$, $\hat{v}_y^x$, $\hat{v}_x^y$, $\hat{v}_y^y$ are the state estimates for each Kalman filter.

Combining Eqs. (6) and (17) we have:

$$\begin{bmatrix}\hat{v}_x\\ \hat{v}_y\end{bmatrix} = \begin{bmatrix}p_{\mathrm{m}_1}\hat{a}_1 + p_{\mathrm{m}_2}\hat{a}_3\\ p_{\mathrm{m}_3}\hat{a}_5 + p_{\mathrm{m}_4}\hat{a}_7\end{bmatrix} + \begin{bmatrix}p_{\mathrm{m}_1}\hat{a}_2 & p_{\mathrm{m}_2}\hat{a}_4\\ p_{\mathrm{m}_3}\hat{a}_6 & p_{\mathrm{m}_4}\hat{a}_8\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} \tag{18}$$

from which it is possible to derive the estimated coefficients of the affine model:

$$\begin{array}{lll}\hat{c}_1 = p_{\mathrm{m}_1}\hat{a}_1 + p_{\mathrm{m}_2}\hat{a}_3, & \hat{c}_2 = p_{\mathrm{m}_1}\hat{a}_2, & \hat{c}_3 = p_{\mathrm{m}_2}\hat{a}_4\\ \hat{c}_4 = p_{\mathrm{m}_3}\hat{a}_5 + p_{\mathrm{m}_4}\hat{a}_7, & \hat{c}_5 = p_{\mathrm{m}_3}\hat{a}_6, & \hat{c}_6 = p_{\mathrm{m}_4}\hat{a}_8\end{array} \tag{19}$$

Figure 12 shows how the multiple model approach is used to estimate the presence of the different generalized deformation components in the optic flow. First, the deformation

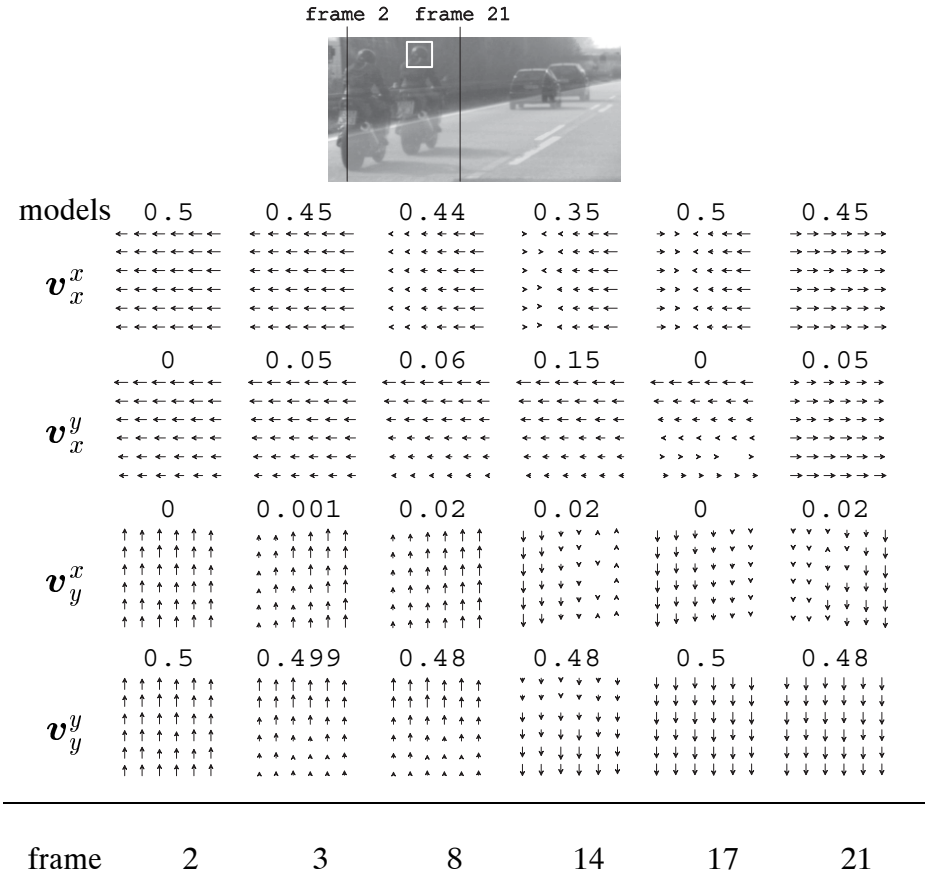


Figure 13: Evolution in time of the four optic flow models in the same image patch. The white squarethat localize the image patch is enlarged for the sake of representation. The sequence is acquired by a car moving on a highway: the independent motion of the motorbike superimposes to the self-motion of the car. The number on the top of each model indicates the associated probability.

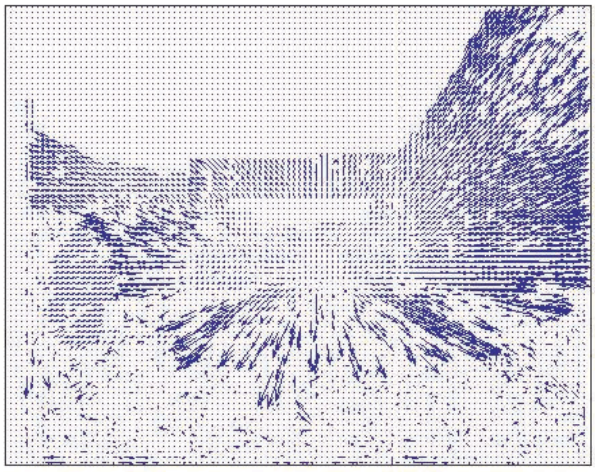
components are adapted accordingly with the optic flow values in input, then a probability value is associated to each component and the final estimate is evaluated by the weighted sum of the single components, see Eq.(16).

Figure 13 shows the evolution in time of the four models related to an optic flow patch in the same position for different frames. The four models are continuosly adjusted on the basis of the input optic flow and a probability value is associated to each model. We can observe through frames the behavior of each model for different motion situations: at frame 2, the patch containsthe motion of the background, only; from frame 8 to frame 17, motion discontinuities appear in the models (e.g., kinetic edges) incorrespondence of the passage of the motorbike; at frame 21, the patch contains the motion of the motorbike, only.

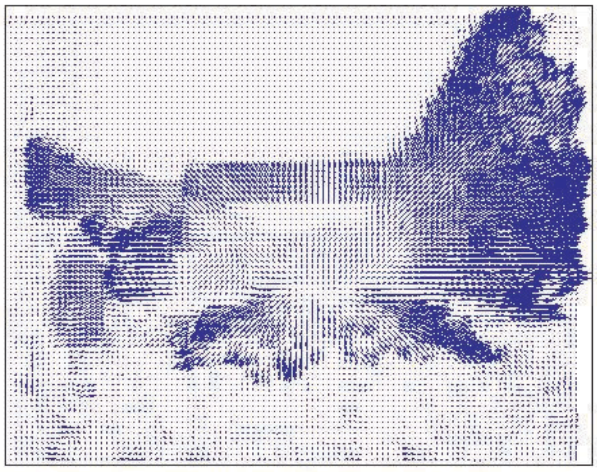
Figure 14 shows the optic flow of a motorway sequence computed from the front-end vision module and its spatial regularization.

This approach involves a high-dimensional state vector since, for each location, it is necessary to consider a spatial neighborhood (patch). The vector state is composed of the two com-

(a)



(b)



(c)

Figure 14: (a)A frame from the motorway sequence. (b) Optic flow computed with the algorithm from the front-end vision module. (b) Spatial regularization using Kalman Filter.

ponents of velocity for each pixel in the considered patch. Considering the high dimensionality of the state vector it is important to verify the efficacy vs. computational costs trade-off of this regularization method, especially when the low level features are already sufficiently accurate.

It is worth to say that the proposed method can be used to obtain a reliable condensation of the optic flow as it will be discussed in Section 5.3.

5 Comparative analysis of the results

5.1 Premise

After the seminal papers of Lucas and Kanade [12] and Horn and Schunck [1] a huge number of methods have been proposed in the attempt of reducing the error while increasing the density of the resulting flow-field maps in different and even more realistic conditions (complex natural scenes, real sensor noise, occlusions, non-rigid motions, etc.). A great impulse to the research has been given by several comparative studies ([11], [13], [14], [3]) , which led to the development and the analysis of systematic "ingredients" for accurate and robust optic flow estimation. Roughly speaking, optic flow methods can be grouped into two main classes: global and local techniques. Global techniques [15] [1] [16] impose a smoothness constraint over the whole image in order to obtain an additional relationship to solve the underconstrained optic flow equation (gradient equation) in which the relationship among the pixel velocity vector (v_x, v_y), the local spatial gradient of luminance (I_x, Iy) and the temporal partial derivative I_t is stated. The global approach, though generally leading to acceptable solutions, imposes a correlation among velocities that does not exist across motion boundaries. Thus, an unlikely flatness of the outcome velocity field appears on the borders of objects having different motions. Local techniques do not require for smoothness to hold all over the image, but only within a restricted neighborhood of each pixel. For example in [17], the gradient equation can be written for each pixel belonging to a small window (all the pixels in the window are assumed to have the same velocity [18]). Sometimes, local techniques lead to poor solutions, especially in regions where luminance is almost uniform, and across object boundaries, where the hypothesis of velocity constancy no longer holds. To improve the estimate, the velocity field has to be regularized, since the local constraints imposed are not sufficient to produce an optic flow smooth enough. Regularization is usually achieved through filtering, with the risk of eliminating important discontinuities. Different filtering techniques, both linear [19] and nonlinear [6], have been used to this purpose. Vector median (VM) filtering is among the most interesting ones, as it permits to reject noisy vectors without degrading motion edges [6], [20]. Parameters indicating the confidence of the estimate can be exploited to obtain an improved vector field [21], [22]. In this Section we will present a systematic comparison between our (recurrent regularization) approach and the other techniques that belong to a consistent class of methods.

5.2 Approach and methodology

The recurrent regularization method here proposed has been conceived to be applied in cascade to a module providing a rough optic flow estimation. Therefore, we first compare the regularization obtained by the recurrent filter with the results obtained by (1) spatial averaging, (2)

spatio-temporal averaging, (3) spatial median, (4) spatio-temporal median, (5) Wiener filtering and (6) bilinear interpolation. More generally, from a conceptual point of view, we can frame our technique as a combination of local and global constraints (cf. CLG methods [3]), where Gabor filters implement the local averaging of the spatial constancy assumption, and the linear model acts as a global regularization constraint. Yet, it is worth noting that, the linear constraint is applied independently for each patch, by minimizing respect the affine parameters related to each patch. As a whole the optic flow is thus regularized by assuming a piecewise linear constraint. From an implementation point of view, the regularization is solved iteratively (cf. Horn and Schunk), and the tentative solution (e.g., provided by the local term) is iteratively refined on the basis of the spatial contextual information. This meets the intrinsic iterative nature of the KF. It worth noting that the KF filter approach allows us to update the estimates continuously as new measures are available, which is a key feature of the KF. From this perspective, it is interesting to analyze the "quality" of the initial (tentative) solution: the better is the initial optic flow measurement, the more reliable will be the final (regularized) solution.

5.2.1 Error metrics

To rate the quality of the regularization methods it is common to use error measures, which compare the estimated (and regularized) to the correct flow field. There are severale possibilities in the literature to compute this error: the *average angular error* (AAE) or the *average squared L^2 norm error* (ASL2E). The first is the most used error measure in the literature, so we decided to follow this approach to compare our regularization method with the other approaches.

The average angular error is defined as follows:

$$AAE = \frac{1}{n_x n_y} \sum_{k=1}^{n_x} \sum_{l=1}^{n_y} \arccos \frac{u_c(k,l)u_e(k,l) + v_c(k,l)v_e(k,l) + 1}{\sqrt{(u_c(k,l)^2 + v_c(k,l)^2 + 1)(u_e(k,l)^2 + v_e(k,l)^2 + 1)}} \quad (20)$$

where (u_c, v_c) is the correct flow field and (u_e, v_e) is the estimated one.

It is worth noting that the AAE has an advantage as well as a disavantage. The advantage is the fact that errors in pixels are not amplified inherent by large displacement (velocity), the disavantage is that velocities are not attached importance and if there are image sequences with different types of translational motion or slightly changing displacements of the same direction, then neglecting the minor occuring motion or adjusting the speed (e.g. by smoothing) may lead to a better average angular error.

The majority of the papers in the literature use AAE measure, so we decided to use this metric instead of others.

5.3 Compression

The patch-based approach allows us to reconstruct the optic flow values within a patch from the 6 affine coefficients. So the method can be also considered as a compression tecnique. In this Section we analyze the compression ratio, by considering the parameters chosen for the analysis presented in the following Sections. We have taken into account patches of 3 different

size: 6×6, 12×12 and 24×24, and the overlapping is a third of the dimension (2, 4 and 8, respectively). In this case the compression ratio is $1.3:1$, $5.3:1$ and $21.3:1$, respectively.

In general, the compression ratio we can obtain is given by:

$$\frac{n \times n}{6 \times (\frac{n}{s})^2} \tag{21}$$

where n is the patch dimension and s is the overlap.

Since the aim of this deliverable is to analyze and compare the regularization issue, we decide to overlap the patches in order to obtain better results for the regularization instead of the compression.

5.4 Contextual combination of partially reliable data

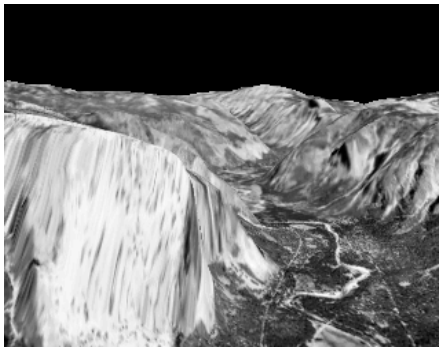
In general, the local extraction of optical flow is relatively inaccurate and non-robust. By non-robust, we mean that the accuracy, in particular parts of the image, is often considerably worse than the general accuracy attainable over much of the rest of the image. The degradation in accuracy is due to a number of factors such as larger noise in that region and/or failure of the underlying image motion model. There are a number of reasons why particular methods of optic flow produce erroneous or inaccurate results. It is useful to categorize these sources according to: (1) failure of the image/motion model [failure of the brightness consistency (weak or strong forms); failure of the motion consistency (weak or strong forms)]; (2) noise (e.g., sensor noise, poor approximation of derivatives in a differential based scheme). When local constraints fail, we can obtain support from neighbor pixels. In particular, (i) when information about motion is missing (and/or is unstable in time) in some points we need the introduction of a spatial (spatio-temporal) coherence, and (ii) when constraints do not hold everywhere (yielding to motion vector outliers) we need methods for combining them robustly. In the following we compared the results of different post-processing techniques: (1) spatial averaging, (2) spatio-temporal averaging, (3) median, (4) spatio-temporal median, (5) Wiener filtering and (6) bilinear interpolation.

For the implementation of these filters we have taken into account both the problem of the border and the non valid values. We have handled the unreliable values of optic flow (the reliability measure comes from the front-end) in the same way of the Kalman Filter technique: if less than half of the values within a patch are under threshold the patch is considered valid and the filter works on the valid values, otherwise the patch is discarded. Different levels of Gaussian noise[2] have been added to the following sequences:

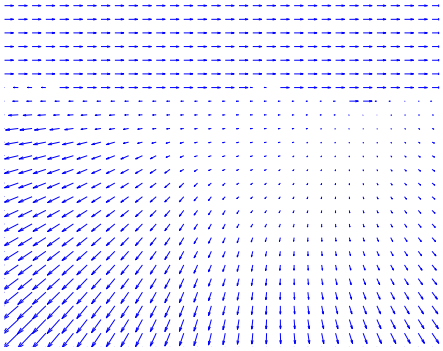
- Otte Sequence (or Marble sequence)
- Yosemite Sequence with clouds
- Diverging tree
- Translating tree.

[2]It is worth noting that, though different noise models can be used, we have decided to use a Gaussian noise model since it allows a more direct comparison of the results available from the literature.

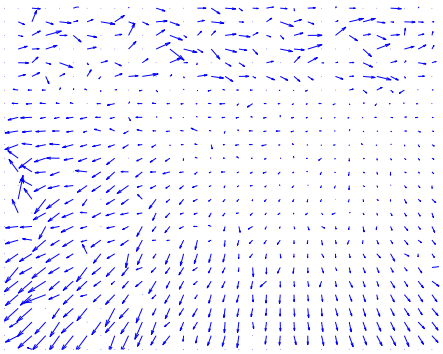
The results of the comparison are shown in Tables 1-4. An example of regularized optic flow is shown in Figure 15.



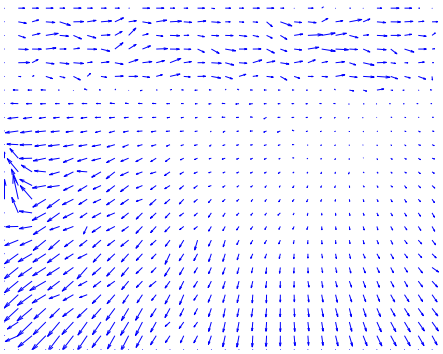
(a) Yosemite sequence



(b) Ground truth



(a) Optic Flow



(b) Regularized Optic Flow

Figure 15: (Yosemite sequence) Comparison between ground truth, optic flow computed with a Gaussian noise added and regularized optic flow with Kalman filter noise $\sigma = 20$.

We further comparatively measure the performance of our technique on the noisy optic flows currently obtained by the mixed software hardware simulations of the FPGA front-end (updated at May 25 and May 30 2008, respectively). The results are shown in Tables 5 and 6. We observe that error on the FPGA-based flow estimates is still quite high and currently justifies a post-processing regularization. The conclusion of the comparison with other techniques are the same we drawn for the artificial noise analysis described above.

5.5 Combining "local" and "global" constraints

One of today's most widely used techniques for the computation of the optic flow are differential methods. Together with phase-based counterparts such as [23] they belong to the techniques with the best performance [11] [13]. Generally, differential techniques can be classified into two broad classes. On one hand there are local methods, which often lead to sparse

	patch		$\sigma_n = 0$	$\sigma_n = 10$	$\sigma_n = 20$	$\sigma_n = 40$
phase-based		AAE	5.74	12.06	16.58	23.25
		STD	14.56	17.03	18.49	20.79
		dens	99	95	93	89
spatial averaging	6 × 6	AAE	5.97	10.70	15.22	22.30
		STD	15.86	18.01	21.32	24.51
		dens	100	100	100	100
	12 × 12	AAE	5.45	8.92	19.51	18.71
		STD	14.10	16.39	10.50	21.95
		dens	100	100	100	100
	24 × 24	AAE	5.31	7.73	10.77	15.22
		STD	10.82	13.79	16.86	17.54
		dens	100	100	100	100
spatio-temporal averaging	6 × 6	AAE	5.79	9.56	13.13	19.72
		STD	15.66	17.21	19.49	22.43
		dens	100	100	100	100
	12 × 12	AAE	5.44	8.24	11.31	17.05
		STD	14.14	15.84	18.16	20.25
		dens	100	100	100	100
	24 × 24	AAE	5.50	7.44	10.04	14.48
		STD	11.22	13.28	15.83	16.40
		dens	100	100	100	100
median	6 × 6	AAE	5.96	10.85	15.32	22.55
		STD	16.53	18.46	21.72	24.88
		dens	100	100	100	100
	12 × 12	AAE	5.37	8.96	12.38	18.41
		STD	15.77	17.29	19.98	22.54
		dens	100	100	100	100
	24 × 24	AAE	4.90	7.84	10.35	14.42
		STD	14.85	16.60	18.06	18.36
		dens	100	100	100	100
spatio-temporal median	6 × 6	AAE	5.70	9.67	13.07	19.74
		STD	16.34	17.73	19.92	23.02
		dens	100	100	100	100
	12 × 12	AAE	5.24	8.24	10.88	16.59
		STD	15.65	16.90	18.69	20.92
		dens	100	100	100	100
	24 × 24	AAE	4.90	7.58	9.63	13.74
		STD	14.94	16.64	17.55	17.69
		dens	100	100	100	100
Wiener	6 × 6	AAE	5.85	10.83	15.16	22.18
		STD	16.01	9.49	21.29	24.38
		dens	99	95	93	89
	12 × 12	AAE	5.50	9.49	13.22	19.41
		STD	15.56	17.46	20.37	23.19
		dens	99	95	93	89
	24 × 24	AAE	5.45	8.55	11.72	16.97
		STD	14.94	16.26	19.15	21.45
		dens	99	95	93	89
interpolation	6 × 6	AAE	6.15	11.50	16.36	23.77
		STD	16.01	18.37	21.71	25.12
		dens	99	100	100	100
	12 × 12	AAE	6.23	11.39	16.06	23.27
		STD	15.81	18.04	21.27	24.57
		dens	100	100	100	100
	24 × 24	AAE	6.96	12.15	16.92	23.85
		STD	15.97	17.99	21.13	24.16
		dens	100	100	100	100
Kalman	6 × 6	AAE	4.62	9.17	12.86	18.87
		STD	12.96	15.19	16.11	17.94
		dens	100	97	96	96
	12 × 12	AAE	4.08	7.47	9.80	15.34
		STD	13.61	14.77	13.89	15.91
		dens	100	100	96	95
	24 × 24	AAE	3.66	6.32	8.47	13.17
		STD	10.36	12.32	13.54	14.17
		dens	96	96	96	96

Table 1: Results for the Yosemite sequence with clouds. Gaussian noise with varying standard deviations σ_n has been added.

	patch		$\sigma_n = 0$	$\sigma_n = 10$
phase-based		AAE	7.66	9.98
		STD	12.35	14.11
		dens	88	71
spatial averaging	6 × 6	AAE	6.68	8.24
		STD	13.55	14.12
		dens	92	84
	12 × 12	AAE	6.42	6.87
		STD	13.33	12.79
		dens	94	86
	24 × 24	AAE	6.55	6.24
		STD	13.16	11.96
		dens	94	88
spatio-temporal averaging	6 × 6	AAE	6.13	7.26
		STD	13.36	13.34
		dens	93	84
	12 × 12	AAE	5.84	6.07
		STD	13.05	12.07
		dens	93	86
	24 × 24	AAE	6.04	5.92
		STD	13.02	12.21
		dens	94	88
median	6 × 6	AAE	6.59	8.20
		STD	13.70	14.41
		dens	93	83
	12 × 12	AAE	6.21	6.63
		STD	13.42	12.83
		dens	94	86
	24 × 24	AAE	5.99	5.62
		STD	13.25	11.91
		dens	94	88
spatio-temporal median	6 × 6	AAE	5.94	6.97
		STD	13.40	13.25
		dens	93	84
	12 × 12	AAE	5.55	5.56
		STD	13.08	11.83
		dens	93	86
	24 × 24	AAE	5.44	5.13
		STD	13.08	11.99
		dens	94	88
Wiener	6 × 6	AAE	5.95	7.49
		STD	12.50	12.74
		dens	88	71
	12 × 12	AAE	5.69	7.12
		STD	12.13	12.02
		dens	88	71
	24 × 24	AAE	5.68	7.25
		STD	12.13	11.47
		dens	88	71
interpolation	6 × 6	AAE	7.68	9.88
		STD	14.85	15.64
		dens	94	88
	12 × 12	AAE	9.30	12.20
		STD	16.46	17.65
		dens	99	98
	24 × 24	AAE	10.36	13.06
		STD	17.02	17.72
		dens	100	100
Kalman	6 × 6	AAE	5.64	8.66
		STD	9.20	13.74
		dens	87	72
	12 × 12	AAE	5.34	7.84
		STD	8.80	12.53
		dens	88	73
	24 × 24	AAE	5.30	7.36
		STD	8.62	14.41
		dens	88	75

Table 2: Results for the Otte sequence. Gaussian noise with varying standard deviations σ_n has been added.

	patch		$\sigma_n = 0$	$\sigma_n = 10$
phase-based		AAE	2.54	9.03
		STD	2.67	10.80
		dens	100	75
spatial averaging	6×6	AAE	1.84	6.54
		STD	2.24	7.73
		dens	100	90
	12×12	AAE	1.81	5.07
		STD	2.60	5.97
		dens	100	97
	24×24	AAE	2.16	4.66
		STD	3.26	5.35
		dens	100	99
median	6×6	AAE	1.88	6.11
		STD	2.10	7.95
		dens	100	90
	12×12	AAE	1.71	4.31
		STD	1.86	5.69
		dens	100	97
	24×24	AAE	1.74	4.08
		STD	1.90	5.69
		dens	100	99
Wiener	6×6	AAE	1.73	6.57
		STD	2.01	8.00
		dens	100	75
	12×12	AAE	1.54	5.90
		STD	2.02	6.52
		dens	99	75
	24×24	AAE	1.62	6.09
		STD	2.13	5.73
		dens	99	75
interpolation	6×6	AAE	1.85	8.31
		STD	1.87	9.50
		dens	100	94
	12×12	AAE	2.37	8.78
		STD	3.27	9.37
		dens	100	100
	24×24	AAE	3.57	9.95
		STD	5.54	10.34
		dens	100	100
Kalman	6×6	AAE	2.12	5.63
		STD	4.00	6.39
		dens	100	91
	12×12	AAE	2.83	5.10
		STD	5.51	6.32
		dens	100	97
	24×24	AAE	4.87	5.06
		STD	7.61	4.53
		dens	100	100

Table 3: Results for the Diverging tree sequence. Gaussian noise with varying standard deviations σ_n has been added.

	patch		$\sigma_n = 0$	$\sigma_n = 10$
phase-based		AAE	0.91	4.52
		STD	2.01	6.65
		dens	100	77
spatial averaging	6 × 6	AAE	0.64	3.04
		STD	1.44	4.47
		dens	100	91
	12 × 12	AAE	0.58	2.07
		STD	1.19	2.94
		dens	100	97
	24 × 24	AAE	0.85	1.95
		STD	1.79	2.99
		dens	100	99
median	6 × 6	AAE	0.59	2.79
		STD	1.53	4.84
		dens	100	90
	12 × 12	AAE	0.41	1.48
		STD	0.66	3.05
		dens	100	97
	24 × 24	AAE	0.34	1.13
		STD	0.49	3.7
		dens	100	99
Wiener	6 × 6	AAE	0.66	4.19
		STD	1.84	7.20
		dens	100	77
	12 × 12	AAE	0.62	4.58
		STD	1.82	6.43
		dens	99	77
	24 × 24	AAE	0.69	5.34
		STD	1.83	5.67
		dens	99	77
interpolation	6 × 6	AAE	0.67	4.00
		STD	1.32	5.70
		dens	100	95
	12 × 12	AAE	1.00	4.55
		STD	2.20	5.88
		dens	100	100
	24 × 24	AAE	1.61	5.00
		STD	3.61	6.35
		dens	100	100
Kalman	6 × 6	AAE	0.58	3.01
		STD	2.87	5.14
		dens	100	92
	12 × 12	AAE	0.79	2.61
		STD	4.06	4.65
		dens	100	98
	24 × 24	AAE	1.17	1.94
		STD	4.39	2.53
		dens	100	100

Table 4: Results for the Translating tree sequence. Gaussian noise with varying standard deviations σ_n has been added.

patch 6 × 6			
	AAE	STD	dens
FPGA simulation	20.33	19.13	72
Kalman	15.91	15.19	97
spatial averaging	15.60	15.57	97
median	15.53	16.27	97
Wiener	18.44	16.45	72
interpolation	16.91	17.09	99

patch 12 × 12			
	AAE	STD	dens
FPGA simulation	20.33	19.13	72
Kalman	14.17	12.59	99
spatial averaging	13.62	13.10	99
median	13.32	13.93	99
Wiener	18.00	15.03	72
interpolation	16.49	16.48	99

patch 24 × 24			
	AAE	STD	dens
FPGA simulation	20.33	19.13	72
Kalman	14.12	11.57	99
spatial averaging	13.29	12.14	100
median	13.07	13.80	100
Wiener	18.08	14.10	72
interpolation	17.11	16.80	100

Table 5: Results for the Yosemite sequence with clouds. The optic flow is obtained by a multiscale software simulation of FPGA.

patch 6 × 6			
	AAE	STD	dens
FPGA simulation	22.96	20.83	98
Kalman	19.73	17.94	1007
spatial averaging	19.82	18.42	100
median	19.53	18.60	100
Wiener	20.05	19.35	98
interpolation	20.61	19.26	100

patch 12 × 12			
	AAE	STD	dens
FPGA simulation	22.96	20.83	98
Kalman	18.62	15.75	100
spatial averaging	18.63	16.40	100
median	17.70	16.28	100
Wiener	18.89	18.01	98
interpolation	20.58	18.93	100

patch 24 × 24			
	AAE	STD	dens
FPGA simulation	22.96	20.83	98
Kalman	18.40	15.46	83
spatial averaging	18.21	14.13	100
median	16.86	14.18	100
Wiener	18.26	16.24	98
interpolation	21.20	18.85	100

Table 6: Results for the Yosemite sequence with clouds. The optic flow is obtained by a mixed software hardware simulations of the FPGA: the optic flow part is performed directly using the FPGA and the multiscale part (expansion, merge procedure and warping) is performed by the software model.

flow fields by neglecting the image areas where normal flow cannot be estimated. Using spatial/spatiotemporal constancy assumptions to cope with the aperture problem their flow field is often computed by optimizing some local energy expression. A typical representative of this strategy is the Lucas-Kanade technique [12] and the structure tensor approach of Bigün et al. [24] [25]. On the other hand there are global methods, that lead to 100 % dense flow fields, but are not that robust under influences of noise. Supplementing the optic flow constraint with a regularizing smoothness term global energy based functionals are obtained, that have to be minimized. The classic Horn and Schunck technique [1] and its numerous discontinuity-preserving variants [26, 27, 28, 29, 30, 31, 2, 32, 33, 34, 35, 36] can be counted to this strategy type. Recently, Brhun et al. have systematically discussed the role of the different smoothing strategies and their effects. On that basis, they developed and analyzed in detail a novel "hybrid" approach that combines the advantages offered by local and global techniques [3] and that is formulated as a general variational problem:

$$E(\boldsymbol{v}) = \int_{\Omega} [\text{data term}] d\boldsymbol{x} + \lambda \int_{\Omega} [\text{regularization term}] d\boldsymbol{x} \tag{22}$$

where Ω is the image domain and the parameter λ controls the relative importance of the two terms.

Since the prototypical approach of Horn and Schunck [1] in 1981, variational methods are among the best performing and best understood techniques for computing the optic flow. Such methods determine the desired displacement field as the minimiser of a suitable energy functional, where deviations from model assumptions are penalised. In general, this energy functional consists of two terms: a data term that imposes temporal constancy on certain image features (e.g. brightness or its phase), and a smoothness term that regularises the often non-unique (local) solution of the data term by an additional smoothness constraint. While the data term represents the assumption that certain (characteristic) image features do not change over time and thus allow for a retrieval of corresponding objects in subsequent frames, the smoothness term stands for the assumption that neighbouring points most probably belong to the same object and thus undergo a similar type of motion. Variational optic flow methods are global methods. If there is not sufficient local information, the data term is os small that it is dominated by the regularization term, which fills in information from more informative surrounding locations. Thus, in contrast to local methods, the *filling-in effect* of global variational approaches always yields dense flow fields such that no subsequent interpolation steps become necessary. Everything is automatically accomplished within a single variational framework. The price to pay for such advantages is the high number of iterations that are usually required to solve (at each frame!) large linear (or non linear) systems of equations with the desired accuracy. Since only neighboring pixels are coupled in iterative relaxation schemes, it may take thousand of iterations to propagate information to large distances, and usually multigrid approaches are adopted to guarantee a much faster convergence (e.g., see [37]. It is worth noting that the recursive solution of our Kalman-based regularization filter overcomes this problem, since the filter uses statistical models to properly weight each new measurement relative to past information, and the algorithm iteratively repeats itself for each new measurement vector, using values stored from the previous cycle. (Yet, it pays the price of the memory required to store the whole status vector of the previous step!).

From this general perspective, the optic flow regularization technique we proposed can be framed within a combined local/global approach. Indeed, considering that:

1. the phase-based front-end optic flow algorithm implements a "local" data conservation constraint based on phase constancy, where (spatial) locality is related to the size of the spatial quadrature pair of Gabor-like bandpass kernels by which the image sequence is pre-filtered, and
2. the affine optic flow models implement patch-wise linearity constraints (cf the linear models $\boldsymbol{v}_M(\boldsymbol{c})$ defined in Section 3 of this report),

we can interpret, at a conceptual level, our approach as a recursive solution of the following general variational problem:

$$E(\boldsymbol{v}, \boldsymbol{c}) = \sum_{\theta} \int_{\Omega} \left(\nabla\phi(\boldsymbol{x}, t) \cdot (\boldsymbol{v}, 1)\right)^2 d\boldsymbol{x} + \lambda \sum_{p} \int_{\Omega_p} |\boldsymbol{v} - \boldsymbol{v}_M(\boldsymbol{c})|^2 d\boldsymbol{x} \tag{23}$$

where θ is the orientation channel associated to the bandpass Gabor filter used in WP1, Ω_p is the p-th image patch, and $\boldsymbol{c} = [c_1, c_2, c_3, c_4, c_5, c_6]$ are the affine coefficients. The iterative solution of 23 can be interpreted as the update equation of KF (cf. Eq. 12).

In the following we will compare the Kalman-based recurrent regularization with the several formulations of combined "local" and "global" methods, by considering the effect of the single components as well as that of their combinations:

Local vs. local: comparative analysis of the phase-based local solution vs. Lucas and Kanade by varying: (1) the filter size (and number of scales) and (2) the noise variance.

(Local+global) vs. (local+global): comparative analysis of the Kalman-based regularization of the phase-based local solution respect with (1) Horn and Schunck (pure "global"), (2) Combined Local-Global (CLG) methods ([3]), by varying: (1) the scale (pyramid level) and (2) the noise variance.

Global linear constraints: specific comparative analysis on different methods using linear parameterization constraints of the optic flow [4] [5].

5.5.1 Lucas and Kanade

The idea behind this technique is the assumption of a constant optic flow field within a certain neighborhood of size ρ. In addition the gradient constraint equation is used ($I_x v_x + I_y v_y + I_t = 0$). Embedded in a weighted least square fit, those two assumptions make it possible to overcome the aperture problem and thus to determine the unknown constants v_x and v_y in each location (x, y, t) minimizing the following function:

$$E_{LK}(\boldsymbol{v}) = \int_{\Omega} G_\rho * (I_x v_x + I_y v_y + I_t)^2 d\boldsymbol{x} \tag{24}$$

or, equivalently, in a more compact form:

$$E_{LK}(\boldsymbol{v}) = \int_{\Omega} \boldsymbol{u}^T (G_\rho * J)\, \boldsymbol{u} d\boldsymbol{x} = \int_{\Omega} \boldsymbol{u}^T J_\rho\, \boldsymbol{u} d\boldsymbol{x} \tag{25}$$

where

$$J = \begin{bmatrix} I_x^2 & I_x I_y & I_x I_t \\ I_y I_x & I_y^2 & I_y I_t \\ I_t I_x & I_t I_y & I_t^2 \end{bmatrix}$$

is the *motion tensor*, $\boldsymbol{u} = (v_x, v_y, 1)$, and G_ρ is a Gaussian convolution kernel, whose standard deviation ρ serves as an *integration scale* over which the main contribution of the least square fit is computed. This technique results in non-dense flow fields, maybe one of the major drawbacks of local flow estimation strategies since many computer vision applications require dense flow estimates. Therefore subsequent interpolation steps are required. But this method has also a lot of advantages: (i) no need for an iterative algorithm making the implementation easy and effective, (ii) fast performance, (iii) robustness under noise. Especially when dealing with very noisy images Lucas and Kanade's leads to good results using a sufficiently large value for ρ .

The comparisons between the DRIVSCO phase-based optic flow algorithm and Lucas and Kanade [3]are shown in Table 7 for the following standard sequences:

- Diverging Trees
- Translating Trees
- Yosemite sequence with/without clouds

In all situations we also added a Gaussian noise with STDs $\sigma_n \in \{0, 10, 20\}$. Following the approach in [38] the noise for each frame within a sequence is created separately and therefore not identical. The results of the Lucas and Kanade method are obtained from [38]. The results in this Section have been computed by optimizing the parameters of the phase-based algorithm in order to obtain the lowest average angular error and the highest density. We used constant parameters for all the sequences and for each σ_n except for the number of scales. A proper choice of the parameters would give lower average angular error especially when there is no noise added. Since the optic flow from the hardware front end might be noisy and unreliable we decided to test our approach even in presence of high noise values.

5.5.2 Horn and Schunck

In contrast to the local Lucas and Kanade method, Horn and Schunck's technique [1] uses a global strategy that satisfies the demand for a dense flow estimate. Thus, it is our representative of the class of global differential methods. They embed the gradient constraint into a regularization framework to end up with the desired dense flow field. That leads to a global energy functional:

$$E_{HS}(\boldsymbol{v}) = \int_\Omega \boldsymbol{u}^T J \boldsymbol{u} d\boldsymbol{x} + \lambda \int_\Omega |\nabla v_x|^2 + |\nabla v_y|^2 d\boldsymbol{x} = \int_\Omega \boldsymbol{u}^T J \boldsymbol{u} d\boldsymbol{x} + \lambda \int_\Omega c_2^2 + c_3^2 + c_5^2 + c_6^2 d\boldsymbol{x} \tag{26}$$

where the $c_i = c_i(\boldsymbol{x})$ are the linear (affine) coefficients defined in Eq. (7). The energy functional itself consists of two different terms: the *data term*, that penalizes deviations from the grey

[3]Here we used a single scale implementation of the Lucas and Kanade algorithm [38], where the integration scale ρ is adapted to the kind sequence.

Diverging Tree				
		$\sigma_n = 0$	$\sigma_n = 10$	$\sigma_n = 20$
Lucas/Kanade	AAE	6.38	6.84	7.53
	STD	5.78	4.46	4.98
phase-based	AAE	2.86	9.08	13.60
	STD	2.73	10.60	13.39
Transating Tree				
		$\sigma_n = 0$	$\sigma_n = 10$	$\sigma_n = 20$
Lucas/Kanade	AAE	0.97	1.25	1.28
	STD	0.50	0.61	0.73
phase-based	AAE	0.97	4.48	6.95
	STD	2.69	6.19	8.82
Yosemite with clouds				
		$\sigma_n = 0$	$\sigma_n = 10$	$\sigma_n = 20$
Lucas/Kanade	AAE	8.79	10.53	12.58
	STD	12.83	12.58	11.88
phase-based	AAE	5.74	12.06	16.58
	STD	14.56	17.02	18.49
Yosemite without clouds				
		$\sigma_n = 0$	$\sigma_n = 10$	$\sigma_n = 20$
phase-based	AAE	2.15	5.47	8.68
	STD	3.12	6.58	9.18

Table 7: Results for the diverging tree, translating tree and Yosemite sequences.

value constance assumption (gradient constraint) and the *smoothness term*, where the square of the flow gradient is used to penalize nonsmooth flow fields. This leads to smooth minimizing functions $(v_x(x, y, t), v_y(x, y, t))$ guaranteeing at the same time the best compliance possible with the optic flow constraint. The weighting of the smoothness term can be adjusted by the weight $\lambda > 0$ serving as a regularization parameter: larger values for λ result in a stronger penalization of large flow gradients thus leading to smoother flow fields.

5.5.3 Least Squares and Total Least Squares methods

The methods described in [39] assume some motion consistency within a patch (Ω_p), after having computed the spatio-temporal derivatives of the image brightness function. Then an estimate of the motion is obtained with an Ordinary Least Squares technique (WLS) or with the proposed Total Least Squares method (WTLS).

5.5.4 Combined Local-Global (CLG)

The aim of the Combined Local-Global (CLG) method described in [40] [37] [3] is to combined the advantages of both local and global approaches for the computation of optic flow. The

authors combined Lucas and Kanade algorithm (local) with Horn and Schunk method (global):

$$E_{CLG}(\boldsymbol{v}) = \int_{\Omega} \boldsymbol{u}^T J_\rho \boldsymbol{u} d\boldsymbol{x} + \lambda \int_{\Omega} c_2^2 + c_3^2 + c_5^2 + c_6^2 d\boldsymbol{x} \tag{27}$$

More precisely, different types of spatial and spatiotemporal regularisers can be implemented (see [3]). Indeed, with respect to the fact that the motion of objects often varies only slowly over time, it seems desirable to impose some amount of temporal or piecewise temporal smoothness as well. In this context, one may also think of extending the Gaussian smoothing to the temporal domain. Althought in principle introducing spatiotemporal models is not very difficult, in practice spatiotemporal models have not been used too often so far. An early suggestion for spatiotemporal anisotropic image-driven regularisers goes back to Nagel [41], followed by spatiotemporal flow-driven approaches such as [28] [42]. The main reason why such techniques have hardly been studied in the literature is the large amount of memory that is required to process multiple frames simultaneously. In the meantime, however, the fast development of standard desktop PCs allows even the computation of whole image sequences of reasonable size and spatiotemporal methods became increasingly appealing in the last years. For the sake of completeness, we will consider in the comparison both spatial (2D-CLG) and spatiotemporal (3D-CLG) implementations, as well as non-linear (i.e., nonquadratic optimization) and multiresolution implementations of the CLG method. We compare the result obtained with our approach using the Yosemite sequence, for which the comparative results are available in the considered papers.

Tables 8 and 9 are organized as follows:

- In the first column there is the name of the algorithm.
- In the second column there is a summary of the parameters that have been chosen for the comparison.
- The last 3 columns show the average angular error (AAE), the standard deviation (STD) and the density.

5.5.5 Noise sensitivity

A specific analysis on the effects of input noise on the results of the methods presented in Section 4.5 is shown in Table 10 and Table 11.

5.6 Model-based regularization

The local constancy assumption over the interaction scale ρ adopted by Lucas and Kanade, can be extended to include a first-order, non constant, term in the flow constraint equation [22]. More generally, for a small image region Ω_p, an affine (linear) transformation is assumed to well approximate the image motion of a smooth surface (cf. Eq. 4), and the best optic flow estimation is obtained by minimizing the energy functional:

	parameters	AAE	STD	Dens.
Horn/Schunck	(Barron et. al 1994).	31.69	-	100
WLS2	$\sigma = 2.0, 15 \times 15, m = 30$, no check	3.17	6.46	100
	$\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.99$	3.13	7.07	76.2
WLS6	$\sigma = 2.0, 15 \times 15, m = 30$, no check	2.86	6.76	100
WTLS2	$\sigma = 2.0, 15 \times 15, m = 30$, no check	2.49	3.08	100
	$\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.99$	2.14	2.58	81.6
WTLS6	$\sigma = 2.0, 15 \times 15, m = 30$, no check	2.05	2.92	100
	$\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.99$	2.01	2.81	96.3
	$\sigma = 2.0, 15 \times 15, m = 30, R^2 = 0.999$	1.74	2.37	72.0
2D-CLG	linear	7.09	-	100
3D-CLG	linear	6.24	-	100
2D-CLG	non linear	6.03	-	100
3D-CLG	non linear	5.18	-	100
2D-CLG	non linear multires.	4.86	-	100
3D-CLG	non linear multires.	4.17	-	100
Kalman	6×6	4.62	12.96	100
	12×12	4.68	13.61	100
	24×24	3.66	10.36	92

Table 8: Results for the Yosemite sequence with clouds.

	parameters	AAE	STD	Dens.
WLS2	$\sigma = 2.0, 15 \times 15, m = 30$, no check	2.51	2.57	100
WLS6	$\sigma = 2.0, 15 \times 15, m = 30$, no check	2.02	2.05	100
WTLS2	$\sigma = 2.0, 15 \times 15, m = 30$, no check	2.56	2.34	100
WTLS6	$\sigma = 2.0, 15 \times 15, m = 30$, no check	1.97	1.96	100
2D-CLG	linear	2.64	-	100
3D-CLG	linear	2.31	-	100
2D-CLG	non linear	1.79	-	100
3D-CLG	non linear	1.46	-	100
2D-CLG	non linear multires.	1.62	-	100
3D-CLG	non linear multires.	1.02	-	100
Kalman	6×6	2.97	5.67	100
	12×12	2.95	6.21	100
	24×24	2.55	4.62	100

Table 9: Results for the Yosemite sequence without clouds

		$\sigma_n = 0$	$\sigma_n = 10$	$\sigma_n = 20$	$\sigma_n = 40$
2D-CLG	AAE	2.64	4.45	6.93	11.30
	STD	2.27	2.94	4.31	7.41
3D-CLG	AAE	1.79	2.53	3.47	5.34
	STD	2.34	2.75	3.37	3.81
Kalman 6×6	AAE	2.97	5.55	9.02	15.59
	STD	5.67	8.22	10.94	14.54
Kalman 12×12	AAE	2.95	4.51	6.64	12.21
	STD	6.21	7.33	8.49	121.14
Kalman 24×24	AAE	2.55	3.87	5.85	10.58
	STD	4.62	5.97	8.64	10.76

Table 10: Results for the Yosemite sequence without clouds. Gaussian noise with varying standard deviations σ_n has been added.

		$\sigma_n = 0$	$\sigma_n = 10$	$\sigma_n = 20$	$\sigma_n = 40$
2D-CLG	AAE	7.14	9.19	10.17	15.82
	STD	9.28	9.62	10.50	11.53
3D-CLG	AAE	6.18	7.25	8.62	11.21
	STD	9.19	9.39	9.97	11.19
Kalman 6×6	AAE	4.62	9.17	12.86	18.87
	STD	12.96	15.19	16.11	17.94
Kalman 12×12	AAE	4.08	7.47	9.80	15.34
	STD	13.61	14.77	13.89	15.91
Kalman 24×24	AAE	3.66	6.32	8.47	13.17
	STD	10.36	12.32	13.54	14.17

Table 11: Results for the Yosemite sequence with clouds. Gaussian noise with varying standard deviations σ_n has been added.

$$E_A(\boldsymbol{c}) = \int_{\Omega_p} (\nabla I \cdot \boldsymbol{v}(\boldsymbol{c}) + I_t)^2 d\boldsymbol{x} \tag{28}$$

where $\boldsymbol{c}$ is the affine coefficient vector. This approach constitutes the "skeleton" of a general class of methods that rely upon a linear parameterization of the optic flow (or image motion) within a small spatial window. Additional "global" constraints are usually introduced to add a spatial coherence constraint between the parameters of the neighboring affine patches. For the comparison, we will refer to the original "Skin-and-bones" method [4] and to a more recent, over-parameterized, variant [5].

5.6.1 Skin and Bones

In [4] the authors describe a method for estimating optical flow that strikes a balance between the flexibility of local dense computations and the robustness and accuracy of global paramterized flow models. The approach tiles the image with a fixed set of patches and assumes that the motion within the regions can be represented by a small number of affine motions that can be thought of as "layers". A spatial coherence constraints that favors solution which are "smooth" is also considered:

$$E_A(\boldsymbol{c}) = \sum_p \int_{\Omega_p} f(\nabla I \cdot \boldsymbol{v}(\boldsymbol{c} + I_t) d\boldsymbol{x} + \lambda \int_{\Omega} f(||\boldsymbol{c} - \boldsymbol{c}(\boldsymbol{\xi})||) d\boldsymbol{\xi} \tag{29}$$

where $f(\cdot)$ is a robust error norm and $\boldsymbol{c}(\boldsymbol{\xi})$ describes the neighboring affine motion.

5.6.2 Over-Parameterized Variational optical Flow

The Over-Parameterized Variational model presented in [5] represents optical flow vector at each pixel by different coefficients of the same motion model in a variational framework. The authors describe optical flow with a set of basis functions of the flow model (fixed and selected a priori) and they recover the space and time varying coefficients of the model. They take into account different models of motion within a patch: affine over-parameterized model, rigid motion model, pure translation motion model and constant motion model.

In Table 12 only the results for the affine model with 3D smoothness constraints are reported, because the approach is the most similar to Kalman-based regularization.

6 Discussion on related scenarios

Perception can be viewed as an inference process to gather properties of real-world, or *distal*, stimuli (e.g., an object in space) given the observations of *proximal* stimuli (e.g., the object's retinal image). The distinction between proximal stimulus and distal stimulus touches on something fundamental to sensory processes and perception. The proximal stimulus, not the distal stimulus, actually sets the receptors' responses in motion. Considering the ill posedness of such (inverse) problem, one should include (*a priori*) constraints to reduce the dimension of

	AAE	STD
Bones	2.77	3.40
Skin&Bones	2.16	2.00
Over-Param. var affine 2D	1.18	1.31
Over-Param. var constant 3D	1.07	1.21
Over-Param. var rigid motion 3D	0.96	1.25
Over-Param. var affine 3D	0.91	1.18
Over-Param. var translation 3D	0.85	1.18
Kalman 6×6	2.97	5.67
Kalman 12×12	2.95	6.21
Kalman 24×24	2.55	4.62

Table 12: Results for the Yosemite sequence without clouds.

the allowable solutions, or, in other terms, to reduce the uncertainty on visual measures. These considerations apply both if one tackles the problem of interpretation (understanding) as a whole, and if one considers the confidence on single feature measurements. In general, KF represents a recursive solution to an inverse problem of determing the distal stimulus based on the proximal stimulus, in case

1. we adopt a stochastic version of the regularization theory involving Bayes' rule
2. we assume Markovianity
3. we consider linear Gaussian models (linearity and Gaussian normal densities).

- The first condition can be motivated by the fact that the a priori contraints necessary to regularize the solution can be described in probabilistic terms. Bayes' rule allows the computation of the *a posteriory* probability $p(\boldsymbol{x}|\boldsymbol{y})$ as follows:

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}$$

  where $p(\boldsymbol{x})$ is the *a priori* probability densities for the distal stimulus and represents *a priori* knowledge about the visual scene; $p(\boldsymbol{y}|\boldsymbol{x})$ is the likelihood function for $\boldsymbol{x}$. This function represents the transformation from the distal to proximal stimulus and includes information about noise in the proximal stimulus. Finally, $p(\boldsymbol{y})$ is the probability of obtaining the proximal stimulus. The inverse problem of determining the distal stimulus can be solved by finding $\hat{\boldsymbol{x}}$ that maximizes the *a posteriori* probability, $p(\boldsymbol{x}|\boldsymbol{y})$. Such $\hat{\boldsymbol{x}}$ is called a maximum a posteriori (MAP) estimator. Although the Bayesian framework is more general than the standard regularization, there exist a relationship between the deterministic and stochastic methods of solving inverse problems. Under some assumptions about the probability densities, maximizing the *a posteriory* probability $p(\boldsymbol{x}|\boldsymbol{y})$ is, indeed, equivalent to minimizing the Tikhonov functional[4].

[4]The regularization method of solving ill-posed inverse problems was formulated by Tikhonov in the early 60s

- The second concept, the Markovianity, captures the step-by-step local nature of the interactions in a cooperative system, and makes possible Kalman recursion, by allowing to express *global* properties of the state in terms of its *local* properties. Under these hypotheses the conditional probability that the system is in a particular state at any time is determined by the distribution of states at its immediately preceding time. That is, the conditional distribution of that states of a system given the present and past distributions depends only upon the present. Specifically, considering the visual signal as a Random Field, the Markovianity hypothesis implies that the joint probability distribution of that random field has associated positive-definite, translational invariant conditional probabilities that are spatially Markovian (Markov Random Fields (MRFs))[5].

- The third assumption represents the necessary conditions to achieve the exact, analytical solution of the KF.

Relationships with MRFs. MRF theory is a branch of probality theory for analyzing the spatial or contextual dependencies of physical phenomena. It is used in visual/image processing to model context dependent entities such as image pixels and correlated features probalistic distributions of interacting labels, i.e., to describe in statistical terms the structure and correlations present in natural images. Formally, MRF theory tells us how to model the *a priori* probability of contextual dependent patterns. Contextual constraints may be expressed locally in terms of conditional probabilities $p(f_i|\{f'_i\})$, where $\{f'_i\}$ denotes the set of values at the other sites $i' \neq i$, or globally as the joint probability $p(f)$. Because local information is more directly observed, it is normal that a global inference is made on local properties. How to make a global inference using local information becomes a non-trivial task. MRF theory provides a mathematical foundation for solving this problem, by relating global and local properties of a cooperative system. Information on the nearest neighborhood is used to calculate conditional probabilities. In their pioneering work, Geman and Geman [44] expressed the statistical prop-

[43]. In this method, the solution is obtained by finding $\hat{\boldsymbol{x}}$, which minimizes a functional:

$$E = ||A\boldsymbol{x} - \boldsymbol{y}||^2 + \lambda||P\boldsymbol{x}||^2$$

where λ is a regularization parameter. The first norm evaluates how close the distal stimulus is to the proximal stimulus, and the second norm evaluates how well the *a priori* constraints are satisfied. If the proximal stimulus is reliable, λ should be small, otherwise λ should be large. In Tikhonov's theory, A is assumed to be a linear operator, $P\boldsymbol{x}$ a linear combination of the first p derivatives of the distal stimulus $\boldsymbol{x}$, and the norm are quadratic.

[5]A system is temporally Markovian if its state at a particular time depends on its state at the immediate preceding time but not on any of its states at earlier times. Similarly, a system is spatially Markovian if the states of its constituent elements depend on those of their neighbors, but not on the states of units that are spatially remote. These local temporal and spatial properties can be described mathematically using the probalistic language of Markov chains and processes, and Markov Random Fields (MRFs). Formally, by considering a random field $F = \{F_1, \ldots, F_m\}$ as a family of random variables defined on the set $\mathcal{S}$, in which each random variable F_i takes a value f_i in $\mathcal{L}$, F is said to be a MRF on $\mathcal{S}$ with respect to a neighborhood system $\mathcal{N}$ iff the following two conditions are satisfied:

$$p(f) > 0, \forall f$$

$$p(f_i|f_{\mathcal{S}-\{i\}}) = p(f_i|f_{\mathcal{N}_i})$$

where $f_{\mathcal{N}_i} = \{f_{i'}|i' \in \mathcal{N}_i\}$ at the site neighboring i.

erties of the natural images in terms of cooperative interactions among pixel elements. The images are endowed with an artificial equilibrium dynamics that evolves the lattice system through a series of configurations to a near-optimal low energy state. Depending on the task being addressed, the optimal states obtained by MRF image processing methods, are those for which noise, blur, and other artifacts have been removed (image reconstruction) and/or where pixels belonging to the same entity have been identified (segregation and segmentation).

Remark 1: MRF methods for image processing usually assume to have the direct accessibility to the "system", whereas in Kalman filter theory only system's measures are observable. More generally we can refer to dynamic (discrete time) *state space models* [45] [46] [47] (cf. also Hidden MRF) given by

$$\boldsymbol{x}[k] \sim p(\boldsymbol{x}[k] \mid \boldsymbol{x}[0], \ldots, \boldsymbol{x}[k-1]) \ , \quad \text{system}$$

$$\boldsymbol{y}[k] \sim p(\boldsymbol{y}[k] \mid \boldsymbol{x}[k]) \ , \quad \text{observations}$$

where $\boldsymbol{y}[k]$ contain the observations at time step k, while $\{\boldsymbol{x}[k]\}$ is an underlying stochastic process which in some cases may have a physical meaning while in other cases it is merely included in order to describe the distribution of the observation process properly. Typically, some prior distribution is placed on $\boldsymbol{x}[0]$. An important task when analyzing data by state space models is estimation of the underlying state process, based on measurements from the observation process. The interest might be on $\boldsymbol{x}[k]$ itself, or merely is a tool for making prediction on $\boldsymbol{y}[k]$. In this perspective, the process (state) equation can be a MRF. The presence of the measurement equation (observations) makes more evident the distinction between the feed-forward and feed-back components of the filter.

Remark 2: Although it is straightforward to derive, in the case of dynamic state-space models (MRF models in time series) for linear Gaussian models, the KF, as an efficient and exact algorithm for computing inference, *spatial* MRFs should be reformulated to be mathematically identical to dynamic models and make the KF work.

Remark 3: The process equation, thought as a state space model describes the statistical properties of the system (visual signal). In this sense, it can be used to model statistical Gestalt rules (good continuation, common fate, etc.) with typical constraint priors, such as "smoothness", "continuity", etc. Yet, optic flow patterns generated by ego- or rigid-body motion, show specific features that cannot be described only in statistical terms, since the velocity vectors in different spatial locations are subject to topological and geometric constraints. It is worthy to note that the process equation adopted in our study, models a structural property of the state space. In that sense, it is possible to describe *specific* vector configurations over (large) spatial regions (i.e., "that radial pattern outflowing from P" *vs* "radiality"). Accordingly, the filter behaves as a template-matching model. To look for Gestalts on the basis of statistical properties a different approach should be followed, requiring the definition of a process equation on a statistical basis.

7 Conclusions

The recurrent adaptive regularization technique proposed in this deliverable always yields better results than those of other non-adaptive averaging operations, for the increasing values of noise level added on the input visual signal. Yet, if the optic flow available from the hardware front-end is sufficiently reliable, it is not convenient to implement the adaptive technique since almost equivalent results can be obtained with less computationally expensive techniques. However, the approach proposed could be useful to provide an high-order description of motion, focusing for example on kynetic boundaries, rotation and expansions of optic flow. These high-order descriptors can be used to provide a compact representation of optic flow that allows us to describe a patch of optic flow by the 6 coefficients of the affine model, only. Therefore, the adaptive patch-based method can be used as a regularization method if the representation maps obtained from the hardware front-end are very noisy. Though, even if not used for regularization purposes, in the framework of DRIVSCO this patch-based optic flow representation can be seen as high-level features (symbolic level) on which basic semantics can be applied. This represents a straightforward link to the work in WP3 (feedback loops as signal-to-symbol loops). This will be explored in the next period.

References

[1] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[2] H.-H. Nagel. Constraints for the estimation of displacement vector fields from image sequences. *In Proc. Eighth International Joint Conference on Artificial Intelligence*, 2:945–951, 1983.

[3] A. Bruhn. *Variational Optic Flow Computation - Accurate Modelling and Efficient Numerics*. PhD thesis, 2006.

[4] S.X. Ju, M. Black, and A. Jepson. Oskin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. *Computer Vision and Pattern Recognition (CVPR'96)*, 1996.

[5] T. Nir, A. M. Bruckstein, and R. Kimmel. Over-parameterized variational optical flow. *International Journal of Computer Vision*, 76(2):205–216, 2008.

[6] T. Koivunen and A. Nieminen. Motion field restoration using vector median filtering on high definition television sequences. *In Proc. Visual Communications and Image Processing'90*, 1360:736–742, 1990.

[7] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall International Editions, 1991.

[8] A. Gelb. *Applied Optimal Estimation*. Artech House, MIT Press, 1974.

[9] J.J. Koenderink. Optic flow. *Vision Res.*, 26(1):161–179, 1986.

[10] Y. Bar-Shalom and X.R. Li. *Estimation and Tracking, Principles, Techniques, and Software*. Artech House, 1993.

[11] J.L. Barron, D.J. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. of Comp. Vision*, 12:43–77, 1994.

[12] B. Lucas and T. Kanade. An iterative image registration technique withan application to stereo vision. *Proc. DARPA Image Understanding Workshop*, pages 121–130, 1981.

[13] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: an analysis of eight optical flow algorithms. *In Proc. 1998 British Machine Vision Conference, Southampton, England*, 1998.

[14] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *In Proc. Eleventh IEEE International Conference on Computer Vision*, 2007.

[15] E. C. Hildreth. Computaion undelying the measurement of visual motion. *Artificial Intelligence*, 23:185–203, 1984.

[16] H.-H. Nagel. On the estimation of the optical flow relations between different approaches and some new results. *Artificial Intelligence*, 33:299–324, 1987.

[17] T. S. Huang. Image sequence analysis. *Berlin, Germany, SpringerVerlag*, 1981.

[18] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. *Proc. Sixth European Conference on Mathematics in Industry*, pages 237–252, 1992.

[19] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60:79–87, 1988.

[20] J. Astola, P. Haavisto, and Y. Neuvo. Vector median filters. *Proc. IEEE*, 78:678–689, 1990.

[21] F. Bartolini, V. Cappellini, C. Colombo, and A. Mecocci. Multi-window least'squares approach to the estimationof optical flow with discontinuities. *IEEE Transaction Pattern Analysis Machine Intelligence*, 32:1250–1256, 1993.

[22] M. Tistarelli. Multiple constraints for optical flow. *flow. In J.-O. Eklundh, editor, Computer Vision ECCV 94, of Lecture Notes in Computer Science*, 801:61–70, 1994.

[23] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *Int. J. of Comp. Vision*, 1:77–104, 1990.

[24] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.

[25] J. Bigün and G. H. Granlund. Optical flow based on the intertiamatrix in the frequency domain. *Proc. SSAB Symposium on Picture Processing*, 1988.

[26] L. Alvarez, M. Esclarin, J.and Lefebure, and J. Sanchez. A pde model for computing the optical flow. *Proc. XVI Congreso de Ecuaciones Diferenciales y Aplicaciones*, pages 1349–1356, 1999.

[27] G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM Journal on Applied Mathematics*, 60(1):156–182, 1999.

[28] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 292–302, 1991.

[29] I. Cohen. Nonlinear variational method for optical flow computation. *In Proc. Eighth Scandinavian Conference on Image Analysis*, 1:523–530, 1993.

[30] F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1217–1232, 1993.

[31] A. Kumar, A. R. Tannenbaum, and G. J. Balas. Optic flow: a curve evolution approach. *IEEE Transactions on Image Processing*, 5(4):1218–1231, 1996.

[32] P. Nesi. Variational approach to optical flow estimation managing discontinuities. *Image and Vision Computing*, 11(7):419–439, 1993.

[33] M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. *In J.-O. Eklundh, editor, Computer Vision ECCV 94, of Lecture Notes in Computer Science*, 801:295–304, 1994.

[34] C. Schnörr. Segmentation of visual motion by minimizing convex non-quadratic functionals. *In Proc. Twelfth International Conference on Pattern Recognition*, A:661–663, 1994.

[35] D. Shulman and J. Herve. Regularization of discontinuous flow fields. *In Proc. Workshop on Visual Motion*, pages 81–90, 1989.

[36] J. Weickert and C. Schnörr. A theoretical framework for convex regularizers in pde-based computation of image motion. *International Journal of Computer Vision*, 45(3):245–264, 2001.

[37] A. Bruhn and J. Weickert. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[38] A. Bruhn. Regularization in motion estimation. Master's thesis, 2001.

[39] A. Bab-Hadiashar and D. Suter. Robust optic flow computation. *International Journal of Computer Vision*, 29(1):59–77, 1998.

[40] A. Bruhn, J. Weickert, and C. Schnorr. Combining the advantages of local and global optic flow methods. *DAGM*, pages 454–462, 2002.

[41] H.-H. Nagel. Extending the oriented smoothness constraint into the temporal domain and the estimation of derivatives of optical flow. *In O. Faugeras, editor, Computer Vision ECCV 90, of Lecture Notes in Computer Science*, 427:139–148, 1990.

[42] J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3):245–255, 2001.

[43] A.N. Tikhnov and V.Y. Arsenin. *Solutions of ill-posed problems*. Winston, Washington, DC, 1977.

[44] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741, 1984.

[45] A.C. Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press, Cambridge, 1989.

[46] M. West and J. Harrison. *Bayesian forecasting and dynamic models*. Springer-Verlag, New York, 1997.

[47] H.R. Künsch. State space and hidden Markov models. In *Complex Stochastic Systems, no. 87 in Monographs on Statistics and Applied Probability*, pages 109–173. Chapman and Hall, London, 2001.

Paper

The following section consists of this paper:

M. Chessa, S.P. Sabatini, F. Solari, G.M. Bisio. *A recursive approach to the design of adjustable linear models for complex motion analysis*. IASTED Conference on Signal Processing, Pattern Recognition and Application. 14-16 February 2007, Innsbruck, Austria.

A RECURSIVE APPROACH TO THE DESIGN OF ADJUSTABLE LINEAR MODELS FOR COMPLEX MOTION ANALYSIS

Manuela Chessa, Silvio P. Sabatini, Fabio Solari, Giacomo M. Bisio
Department of Biophysical and Electronic Engineering
University of Genoa
Via all'Opera Pia 11A
16145 Genova - ITALY
email: manuela@dibe.unige.it

ABSTRACT
Parametric models are widely used in motion analysis. Traditionally, affine or learned models are adopted. Here, we propose the use of a set of linear models that dynamically adjust their properties to approximate first-order structures in noisy optic flow fields. Each model is generated by the evolution of a recursive network that can be used as a process equation of a multiple model Kalman Filter. The presence of a model is checked by computing the consistence between the observations (data) and the predictions (model). In each image region, for each model, a probability value can be computed, on which to base motion analysis. Experimental results on multiple motion detection problems and facial expressions analysis validate the approach. The algebraic transformations relating our linear descriptors with the traditional affine models are discussed.
KEY WORDS
Recursive Filtering, Motion Detection, Kalman Filter, Optic Flow.

1 Introduction

Reliable complex (e.g. multiple or non rigid) motion analysis is a challenging problem in computer vision, with several impacts in different application domains. Indeed, distinguishing on visual basis, different motion causes may help the recognition of actions and events [1] [2] [3], such as gestures [4] [5] and facial expressions [6], the location of objects whose trajectory could intersect observer's path, or to coordinate movements to interact with other moving objects (separating ego-motion from independent object motion), as well the reconstruction of the 3-D structure of the observed scene.

The multiple motion detection problem can be addressed as a segmentation problem relying on local descriptors of the optic flow. A popular class of local flow descriptors is based on parameterized models of optic flow [7]. Such models, learned from examples [8] [9], or specified a priori as constant and affine (linear) models, are characterized by a small number of parameters, which provide a concise description of the optic flow structure that can be used to recognize motion patterns from image sequences. In general, linear models can be used both for estimating optic flow directly from the spatio-temporal image derivatives and for filtering a dense optic flow field. In the recent years, the former approach greatly affermated [10] since the recovering of the model coefficients directly from the spatiotemporal variations of image intensity improves the accuracy and stability of the motion estimates. These methods work very well when the model is a good approximation to the image motion, but they fall short when large image regions are not well modeled by a single parametric motion. This could happen because of the complexity of motion or because of the presence of multiple motions.

In this paper, we propose a method to design adjustable linear models for the analysis of complex dense optic flow fields. The models are specified as discrete space-time dynamical systems, in the velocity space, that are characterized by an unforced or "free" response, given by the structure of network interconnections, and a forced response related to the contingent local optic flow information in input. In this way, given a motion information represented by an optic flow field extracted by a "classical" algorithm, we recognize if a group of velocity vectors relates to a specific motion pattern, on the basis of their spatial relationships in a local neighborhood. More precisely, the analysis/detection occurs through a spatial recurrent filter that checks the consistency between the spatial structural properties of the input flow field pattern and a set of linear models representing (first-order) elementary components of the optic flow [11]. In order to design a filter that checks this consistency, in an adaptive way, the linear models can be considered the process equations of a multiple model Kalman Filter (KF). Motion segments emerge from the noisy flows as the output of the KF that compares its prediction to the actual observations of the local properties of the optic flow.

Many works in the literature make use of the Kalman Filter for motion estimation. It has been used to estimate kinematic parameters (rotational and translational velocities and acceleration) of three-dimensional features [12] or to track 2D features through a sequence [13]. In [14] affine motion models are used to perform a region-based tracking in long image sequences and a standard Kalman Filter generates recursive estimation of each motion parameter. The novelty of the approach presented in this paper is in the def-

inition of models, which describe the optic flow and not the motion in the 3D space.

2 Linear models

Motion flow fields usually consist of large patches of flow-patterns, which result from a common cause (e.g., from ego-motion or object motion). These flow-patterns can be characterized on the basis of their first-order (linear) differential properties. From this perspective, local spatial features around a given location of a flow field can be of two types [11]: (1) the average flow velocity at that location, and (2) the structure of local variation in the neighborhood. The former relates to the *smoothness constraint* or *structural uniformity*, the latter refers to the *linearity constraint* or *structural gradients*. Velocity gradients provide important information about the 3-D layout of the visual scene.

Formally, the velocity gradient tensor can be written as follows:

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = \begin{bmatrix} \partial v_x/\partial x & \partial v_x/\partial y \\ \partial v_y/\partial x & \partial v_y/\partial y \end{bmatrix}. \quad (1)$$

If we consider a point $\boldsymbol{x} = (x, y)$ in the spatial image domain, the linear properties of a motion field $\boldsymbol{v}(x, y) = (v_x, v_y)$ around the point $\boldsymbol{x}_0 = (x_0, y_0)$ can be characterized by a first-order Taylor expansion:

$$\boldsymbol{v} = \bar{\boldsymbol{v}} + \bar{\mathbf{T}}\boldsymbol{x} = \bar{\boldsymbol{v}} + \begin{bmatrix} \bar{T}_{11} & \bar{T}_{12} \\ \bar{T}_{21} & \bar{T}_{22} \end{bmatrix} \boldsymbol{x} \quad (2)$$

where $\bar{\boldsymbol{v}} = \boldsymbol{v}(x_0, y_0) = (\bar{v}_x, \bar{v}_y)$ and $\bar{\mathbf{T}} = \mathbf{T}|_{\mathbf{x_0}}$. By breaking down the tensor in its dyadic components, the motion field can be locally described through two-dimensional maps representing elementary flow components (EFCs) and Eq. (2) can be written as:

$$\boldsymbol{v} = \boldsymbol{\alpha}^x \bar{v}_x + \boldsymbol{\alpha}^y \bar{v}_y + \mathbf{d}_x^x \bar{T}_{11} + \mathbf{d}_y^x \bar{T}_{12} + \mathbf{d}_x^y \bar{T}_{21} + \mathbf{d}_y^y \bar{T}_{22} \quad (3)$$

where $\boldsymbol{\alpha}^i$ are pure translations:

$$\boldsymbol{\alpha}^x : \begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{\alpha}^y : \begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and $\mathbf{d}_j^i$ are cardinal deformations:

$$\mathbf{d}_x^x : \begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} x \\ 0 \end{bmatrix}, \quad \mathbf{d}_y^x : \begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} y \\ 0 \end{bmatrix}$$

$$\mathbf{d}_x^y : \begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 0 \\ x \end{bmatrix}, \quad \mathbf{d}_y^y : \begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} 0 \\ y \end{bmatrix}.$$

The components of pure translations $\boldsymbol{\alpha}^i$ can be incorporated in the corresponding deformations components, thus obtaining *generalized deformation components*:

$$\begin{aligned} \boldsymbol{v}_x^x &= a_1\boldsymbol{\alpha}^x + a_2\mathbf{d}_x^x \triangleq \boldsymbol{m}_1 \\ \boldsymbol{v}_y^x &= a_3\boldsymbol{\alpha}^x + a_4\mathbf{d}_y^x \triangleq \boldsymbol{m}_2 \\ \boldsymbol{v}_x^y &= a_5\boldsymbol{\alpha}^y + a_6\mathbf{d}_x^y \triangleq \boldsymbol{m}_3 \\ \boldsymbol{v}_y^y &= a_7\boldsymbol{\alpha}^y + a_8\mathbf{d}_y^y \triangleq \boldsymbol{m}_4 \end{aligned} \quad (4)$$

In this way, we have four classes of deformation gradients: one stretching ($\boldsymbol{v}_i^i$) and one shearing ($\boldsymbol{v}_j^i$) for each cardinal direction. As it will be clear in the following, this choice gives to the model maximum flexibility: every gradient deformation within a single class will be built through the same recurrent network, just by changing its driving inputs on the basis of direct local measures on the input optic flow. Figure 1 shows the four classes of deformation gradients.

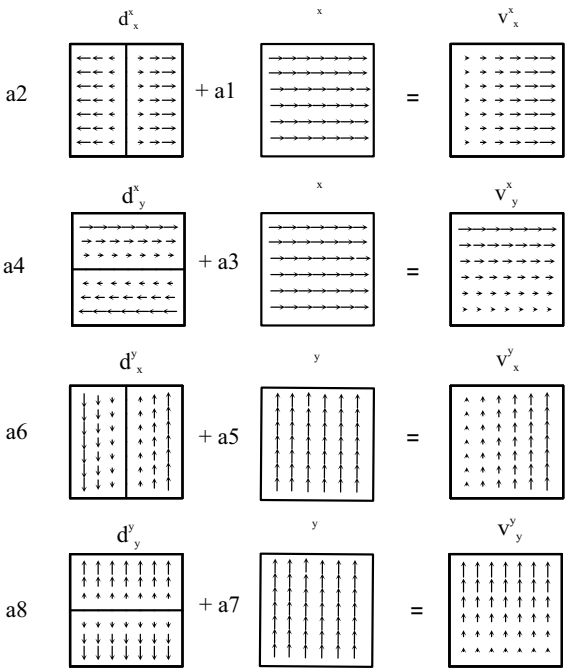


Figure 1. The generalized deformation components ($\boldsymbol{v}_x^x$, $\boldsymbol{v}_y^x$, $\boldsymbol{v}_x^y$, $\boldsymbol{v}_y^y$) are obtained by incorporating the pure translations in the corresponding cardinal deformations.

It is worthy to note that Eqs. (3) and (4) describe, in fact, an affine model:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} c_1 \\ c_4 \end{bmatrix} + \begin{bmatrix} c_2 & c_3 \\ c_5 & c_6 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \quad (5)$$

where c_i are constants and v_x and v_y are the horizontal and the vertical components of the flow. The parameter vector $[c_1, c_2, \ldots, c_6]$ describes a specific configuration of optic flow that locally provides a good approximation of 3D rigid moving objects. The six parameter affine model is reasonable to describe the motions of smooth surface in small image regions. The affine model is not sufficient to describe non rigid motion, like the motion of a human face [6]. However the motion within small patches can be still approximated by a first-order model. The relationships between these patches will describe the global motion of the face. The parameters c_i have qualitative interpretations in terms of image motion, for example c_1 and c_4 represent horizontal and vertical translation and we can express divergence (isotropic expansion), curl (rotation about the viewing direction), and the two components of shear defor-

mation (squashing, def_1, or stretching, def_2) as combination of the c_i's:

$$\begin{aligned} div &= c_2 + c_6 \\ curl &= c_3 - c_5 \\ def_1 &= c_3 + c_5 \\ def_2 &= c_2 - c_6 \end{aligned} \tag{6}$$

3 Kalman filtering

The problem of evidencing the presence of a certain complex pattern in the optic flow is posed as an adaptive filtering problem. The Kalman Filter is an optimal recursive adaptive filter [15], in the sense that it can iteratively process new measures as they arrive, on the basis of the knowledge about the system obtained by previous measurements. Kalman filtering is an optimal estimator if noise is independent, zero-mean and normally distributed. The output of the filter will be the *a posteriori* estimate of motion field improved by the additional (contextual) information provided by Kalman innovation.

Kalman filtering needs a *measurement equation* and a *process equation*. Formally we can write the following *measurement equation*:

$$\mathbf{v}[k] = \mathbf{C}[k]\boldsymbol{v}[k] + \boldsymbol{n}_1[k] \tag{7}$$

where $\mathbf{v}[k]$ is the optic flow at current time k, an intensity-based measure of the actual velocity field $\boldsymbol{v}[k]$ and $\boldsymbol{n}_1[k]$ models the uncertainty of the algorithm. The linear operator $\mathbf{C}$ represents a general "early-vision filter" providing a noisy measure of an observable property of the visual stimulus.

The *process equation* models the temporal evolution, from the previous step $k-1$ to the current time k, of the relationships among visual features over a fixed spatial region, according to specific rules embedded in the transition matrix $\mathbf{\Phi}$:

$$\boldsymbol{v}[k] = \mathbf{\Phi}[k, k-1]\boldsymbol{v}[k-1] + \boldsymbol{n}_2[k-1] + \boldsymbol{s}[k-1] \tag{8}$$

where $\boldsymbol{s}[k]$ is a driving input that can be interpreted as the boundary conditions of a lattice network (see Figure 2) and $\boldsymbol{n}_2[k]$ represents the process noise. Matrix $\mathbf{\Phi}$ together with driving inputs $\boldsymbol{s}[k]$ implements a specific linear deformation component (see Eq. (4)). More precisely, this matrix models space-invariant nearest neighbor interactions within a finite region Ω in the image plane.

The driving input $\boldsymbol{s}[k]$ is evaluated at each step, by computing the average of optic flow velocity components at the boundary. So, the four models are adapted to the measures continuously. The spatial interactions occur separately for each component of the velocity vectors through anisotropic nearest neighbor interconnection schemes. Specifically, for the x component we have:

$$\begin{aligned} v_x(i,j)[k] = \; & w_N^x v_x(i,j-1)[k-1] \; + \\ & + \; w_S^x v_x(i,j+1)[k-1] \; + \\ & + \; w_W^x v_x(i-1,j)[k-1] \; + \\ & + \; w_E^x v_x(i+1,j)[k-1] \; + \\ & + \; w_T^x v_x(i,j)[k-1] \; + \\ & + \; n_2^x(i,j)[k-1] \; + \\ & + \; s_x(i,j) \end{aligned} \tag{9}$$

and the same equation applies for v_y. The resulting pattern depends on the anisotropy of the interaction scheme and on the boundary conditions. By example, considering, for the sake of simplicity, a rectangular domain $\Omega = [-L, L] \times [-L, L]$, the EFC $\boldsymbol{m}_1$ can be obtained through:

$$\begin{aligned} & w_T^x = 0.1 \\ & w_N^x = w_S^x = 0 \qquad\qquad w_N^y = w_S^y = 0 \\ & w_W^x = w_E^x = 0.45 \qquad\qquad w_W^y = w_E^y = 0 \end{aligned}$$

$$s_x(i,j) = \begin{cases} \lambda & \text{if } i = -L \\ \mu & \text{if } i = L \\ 0 & \text{otherwise} \end{cases} \qquad s_y(i,j) = 0$$

where the boundary values λ and μ are related to the coefficients c_1 and c_2, and control the gradient slope and the constant term. In a similar way we can obtain the other components (see Figure 2). In this way, all the structural constraints necessary to model the continuum of linear deformations are embedded in the lattice interconnection scheme of the process equation. The resulting lattice network has a *structuring effect* constrained by the boundary conditions that yields to structural equilibrium configurations, characterized by the specific first-order EFCs that properly describe the input flow.

To describe the Kalman filtering processing, we define $\hat{\boldsymbol{v}}[k|\mathbf{V}_{k-1}]$ as the *a priori* state estimate at step k, given the knowledge of the process at step $k-1$, and $\hat{\boldsymbol{v}}[k|\mathbf{V_k}]$ as the *a posteriori* state estimate at step k given the measurement at step k. $\mathbf{V}_{k-1}$ and $\mathbf{V}_k$ represent all the measurements until step $k-1$ and k respectively, the aim of the filter is to compute an *a posteriori* estimate starting from the *a priori* estimate and from the weighted difference between the current and the predicted measurement:

$$\hat{\boldsymbol{v}}[k|\mathbf{V}_k] = \hat{\boldsymbol{v}}[k|\mathbf{V}_{k-1}] + \boldsymbol{G}[k](\mathbf{v}[k] - \hat{\mathbf{v}}[k|\mathbf{V}_{k-1}]). \tag{10}$$

The difference term $\mathbf{v}[k] - \hat{\mathbf{v}}[k|\mathbf{V}_{k-1}]$ is the *innovation* $\boldsymbol{\nu}[k]$, while the matrix $\boldsymbol{G}[k]$ is the Kalman gain that minimizes the *a posteriori* error covariance:

$$\boldsymbol{K}[k] = E\{(\boldsymbol{v}[k] - \hat{\boldsymbol{v}}[k|\mathbf{V}_k])(\boldsymbol{v}[k] - \hat{\boldsymbol{v}}[k|\mathbf{V}_k])^T\}. \tag{11}$$

The covariance matrix $\boldsymbol{K}[k]$ provides us only information about the properties of convergence of the Kalman Filter and not whether it converges to the correct values. Hence, we have to measure the discrepancy between predictions and observations in statistical terms, as an indication of the filter's consistency. A frequently used quantitative measure

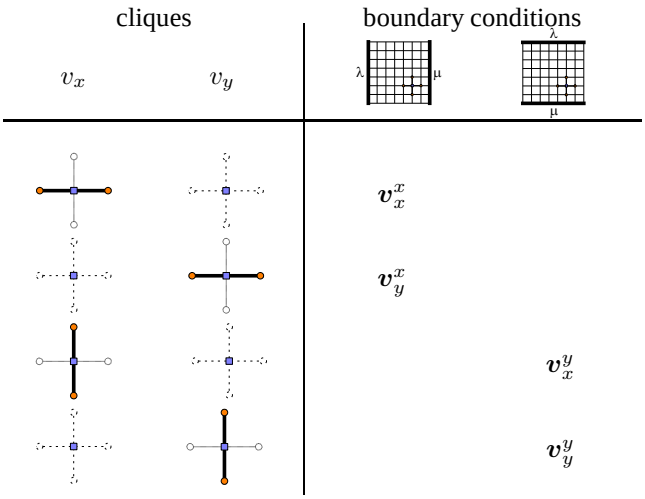


Figure 2. Basic lattice interconnection schemes for the elementary flow components considered. By a proper choice of the interconnection weights and of the boundary values λ and μ the velocity profiles result approximately linear.

of consistency is the Normalized Innovation Squared (NIS) [16]:

$$NIS_k = \boldsymbol{\nu}^T[k]\boldsymbol{S}^{-1}[k]\boldsymbol{\nu}[k] \tag{12}$$

where $\boldsymbol{S}$ is the covariance of the innovation. In our model the NIS value is used to compute the likelihood of the measurement.

3.1 Multiple model approach

The structure of the optic flow approximation leads to a multiple model adaptive estimator: we can use a bank of parallel Kalman Filters, each with a different model (a different process equation embedding a generalized deformation component). We need a dynamic multiple model approach because the choice between the four possible models varies continuously while the filter is operating. In such a case, we cannot make a fixed *a priori* choice of the filter parameters, but we need to use a continuously varying model-conditioned combination of the candidate state and error covariance estimates. It is worthy to note that, in the dynamic multiple model approach, we do not want the probabilities to converge to fixed values, but we want them free to change at each new measurement. In the multiple model approach [16] [17] it is assumed that the system obeys one of a finite number of models. Thus, we must assume that the correct model $\boldsymbol{m}$ is one among all the possible models $\boldsymbol{m}_i$ with $i = 1, 2, \ldots, r$.

The likelihood of the measurement $\mathbf{v}$ given a particular model $\boldsymbol{m}_i$ at time step k is given by:

$$f(\mathbf{v}|\boldsymbol{m}_\mathbf{i}) = |\mathbf{2}\pi\boldsymbol{S}_{\boldsymbol{m}_\mathbf{i}}|^{-\frac{1}{2}}\mathbf{e}^{-\frac{1}{2}\boldsymbol{\nu}_{\boldsymbol{m}_\mathbf{i}}^{\mathbf{T}}\boldsymbol{S}_{\boldsymbol{m}_\mathbf{i}}^{-1}\boldsymbol{\nu}_{\boldsymbol{m}_\mathbf{i}}} \tag{13}$$

where $\boldsymbol{m}_i$ is the considered model. The probability that the candidate model $\boldsymbol{m}_i$ is the correct one is given by the following equation:

$$p_{\boldsymbol{m}_i}[k] = \frac{f(\mathbf{v}|\boldsymbol{m}_\mathbf{i})}{\sum_{j=1}^{r} f(\mathbf{v}|\boldsymbol{m}_\mathbf{j})} \tag{14}$$

with $p_{\boldsymbol{m}_i}[0] = 1/r, i = 1, 2, \ldots, r$ and $\sum_{i=1}^{r} p_{\boldsymbol{m}_i}[k] = 1$ at each time step k. The final model-conditioned estimate of the state $\boldsymbol{v}$ is computed as a weighted combination of the *a posteriori* states of each candidate filter:

$$\hat{\boldsymbol{v}}[k] = \sum_{i=1}^{r} p_{\boldsymbol{m}_i}[k]\hat{\boldsymbol{v}}_{\boldsymbol{m}_i}[k]. \tag{15}$$

For the 4 models considered (see Eqs. (4)):

$$\hat{\boldsymbol{v}} = p_{\boldsymbol{m}_1}\hat{\boldsymbol{v}}_x^x + p_{\boldsymbol{m}_2}\hat{\boldsymbol{v}}_y^x + p_{\boldsymbol{m}_3}\hat{\boldsymbol{v}}_x^y + p_{\boldsymbol{m}_4}\hat{\boldsymbol{v}}_y^y \tag{16}$$

where $p_{\boldsymbol{m}_1}, p_{\boldsymbol{m}_2}, p_{\boldsymbol{m}_3}, p_{\boldsymbol{m}_4}$ are the probabilities related to each model and $\hat{v}_x^x$, $\hat{v}_y^x$, $\hat{v}_x^y$, $\hat{v}_y^y$ are the state estimates for each Kalman filter.

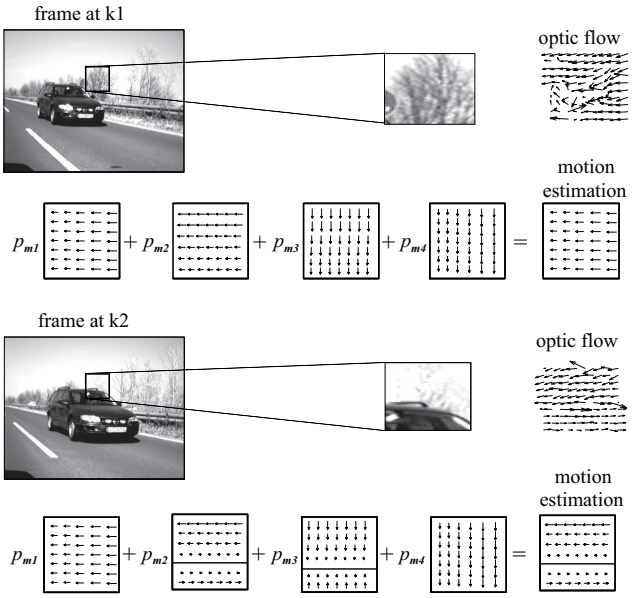


Figure 3. Multiple-model motion estimation in a road scene taken by a rear-view mirror of a moving car under an overtaking situations. The model-based decompositions are evidenced for the same image patch for two different frames at time $k1$ and $k2$. For each optic flow patch the motion is estimated from the actual generalized deformation components weigthed by the corresponding probability values, see Eq.(15).

Combining Eqs. (4) and (16) we have:

$$\begin{bmatrix}\hat{v}_x\\ \hat{v}_y\end{bmatrix} = \begin{bmatrix}p_{\boldsymbol{m}_1}\hat{a}_1 + p_{\boldsymbol{m}_2}\hat{a}_3\\ p_{\boldsymbol{m}_3}\hat{a}_5 + p_{\boldsymbol{m}_4}\hat{a}_7\end{bmatrix} + \begin{bmatrix}p_{\boldsymbol{m}_1}\hat{a}_2 & p_{\boldsymbol{m}_2}\hat{a}_4\\ p_{\boldsymbol{m}_3}\hat{a}_6 & p_{\boldsymbol{m}_4}\hat{a}_8\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} \tag{17}$$

from which it is possible to derive the estimated coefficients of the affine model:

$$\hat{c}_1 = p_{\boldsymbol{m}_1}\hat{a}_1 + p_{\boldsymbol{m}_2}\hat{a}_3, \quad \hat{c}_2 = p_{\boldsymbol{m}_1}\hat{a}_2, \quad \hat{c}_3 = p_{\boldsymbol{m}_2}\hat{a}_4$$
$$\hat{c}_4 = p_{\boldsymbol{m}_3}\hat{a}_5 + p_{\boldsymbol{m}_4}\hat{a}_7, \quad \hat{c}_5 = p_{\boldsymbol{m}_3}\hat{a}_6, \quad \hat{c}_6 = p_{\boldsymbol{m}_4}\hat{a}_8 \tag{18}$$

Figure 3 shows how the multiple model approach is used to estimate the presence of the different generalized deformation components in the optic flow. First, the deformation components are adapted accordingly with the optic flow values in input, then a probability value is associated to each component and the final estimate is evaluated by the weighted sum of the single components, see Eq.(15).

4 Results

To assess the performances of the approach, we applied recursive Kalman filtering to optic flows related to both real-world driving sequences and facial expressions. A "classical" algorithm [18] has been used to extract the optic flows.

If we analyze a multiple motion sequence we expect that objects in the background have the same divergence values, whereas other objects moving in the scene will have a different divergence. Therefore, by mapping the sum of c_2 and c_6, we are able to obtain a good segmentation of the objects in the scene. Figures 4 and 5 show examples of multiple motion segmentation using divergence information for different real-world traffic scenes.

Figure 6 shows how this approach can be used to analyse different areas of optic flow in a complex motion sequence like a facial expression. If we consider the values of the affine model coefficients we are able to describe the motions of the different areas of the face. In the figure five different areas of the face have been chosen and the coefficients of the affine models have been computed and plotted as a function of time. The relationships between the temporal behaviour of these values and their spatial positions could describe quite well the face motion.

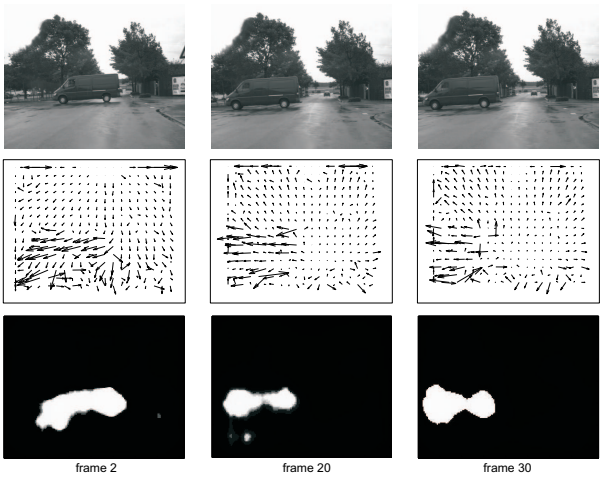


Figure 4. Example of multiple motion segmentation. The camera is moving towards the van that is crossing the street.

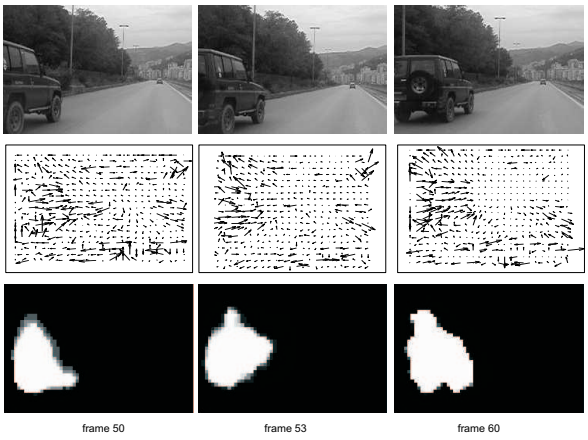


Figure 5. Example of multiple motion segmentation. The camera and the car are moving along the same direction.

5 Conclusions

The problem of evidencing the presence of a certain complex feature in the optic flow is an important step towards motion segmentation. We have shown that it is possible to solve this problem on the basis of both direct input and contextual information, by recurrent adaptive filtering of the optic flow. Direct information comes from the input measures and the context from reference signals, represented as a set of specific linear models. Kalman-filter based techniques to switch between models have been known for some time in the control literature [16]. Here, we propose a similar approach to permit multiple linear models as multiple competiting hypotheses. Accordingly, the multi-model Kalman Filter yields the optimal estimates of the weights of the adjustable linear models. A great potential advantage of the multiple-model approach is that recognition and feature extraction can be performed jointly, and so the form of the expected linear component can be used to guide feature search, potentially making it more efficient and robust. The use of linear models to analyze image motion has been previously investigated in [7], where the authors proposed the use of parameterized motion models to represent complex motions. In that paper, they adopted both linear and learned basis flow fields to describe the motion of large portions of the face. Here, by considering small areas of facial expression, we are able to approximate image flows with linear models. A systematic comparison between the two approaches will be tackled in a future work.

Acknowledgements

This work has been partially supported by the EU Project NEST-2003-12963 "Multi-channel cooperativity in visual processing (MCCOOP)" and by the EU Project IST-2003-016276-2 "Learning to emulate perception-action cycles in

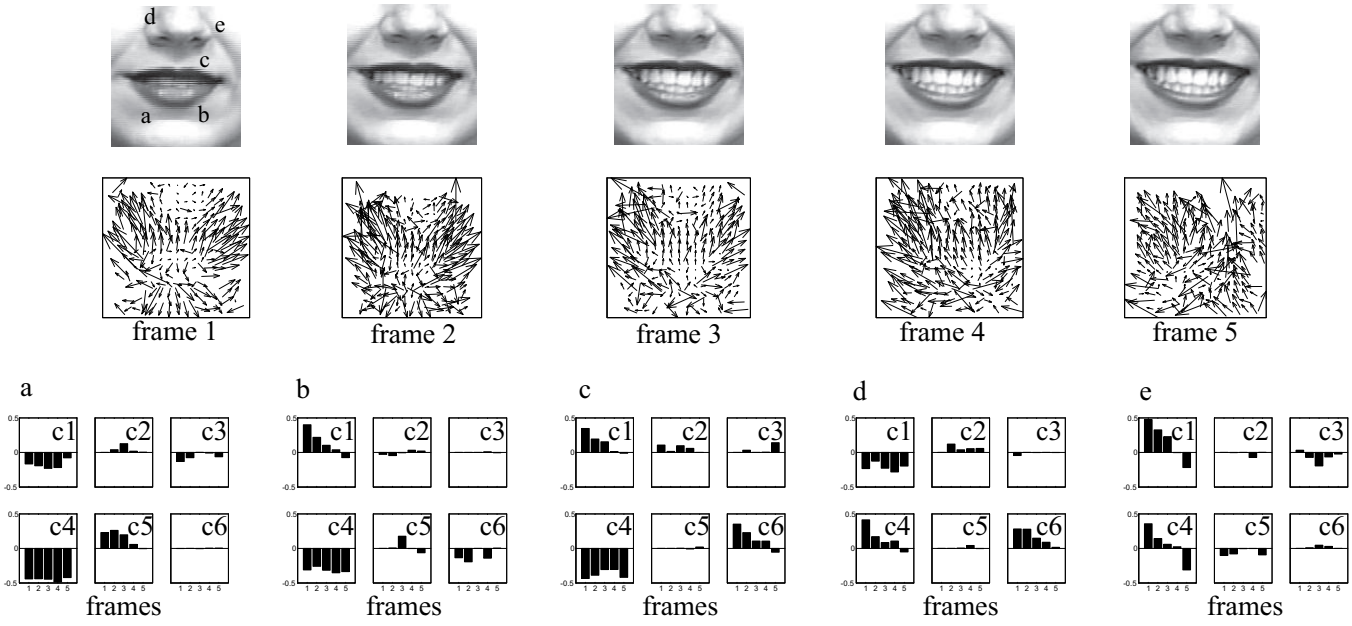


Figure 6. Top: five frames from a facial expression sequence. On the first frame the letters indicate the positions of the analysed areas. Middle: Optic flows computed from the sequence. Bottom: temporal evolutions of the six affine model coefficients for the five selected positions (a-e).

a driving-school scenario (DRIVSCO)".

References

[1] M.J. Black, "Explaining optical flow events with parameterized spatio-temporal models", *CVPR*, 1999.

[2] Y. Rui and P. Anandan, "Segmenting visual actions based on spatio-temporal motion patterns", *CVPR*, 2000.

[3] J. Davis and A. Bobick, "The representation and recognition of action using temporal templates.", *CVPR*, 1997.

[4] Y. Yacoob and M.J. Black, "Parameterized modeling and recognition of activities", *Computer Vision and Image Understanding*, vol. 73, pp. 232–247, 1999.

[5] T. Darrell and A. Pentland, "Space-time gestures", *CVPR*, 1993.

[6] M.J. Black and Y. Yacoob, "Recognizing facial expression in image sequences using local parameterized models of image motion", *International Journal of Computer Vision*, vol. 25, pp. 23–48, 1997.

[7] D.J. Fleet, MJ. Black, Y.Yacoob, and A.D. Jepson, "Design and use of linear models for image motion analysis", *International Journal of Computer Vision*, vol. 36, pp. 171–193, 2000.

[8] M.J. Black, Y. Yacoob, A.D. Jepson, and D.J. Fleet, "Learning parameterized models of image motion", *CVPR*, 1997.

[9] Y. Yacoob and L. Davis, "Learned temporal models of image motions.", *International Journal of Computer Vision*, pp. 446–453, 1998.

[10] M.J. Black and P. Anandan, "The robust estimation of multiple motion: Parametric and piecewise-smooth flow fields", *Computer Vision and Image Understanding*, vol. 63, pp. 75–104, 1996.

[11] J.J. Koenderink, "Optic flow", *Vision Res.*, vol. 26, pp. 161–179, 1986.

[12] Z. Zhang and O. D. Faugeras, "Three-dimensional motion computation and object segmentation in a long sequence of stereo frames.", *International Journal of Computer Vision*, vol. 7, pp. 211–241, 1992.

[13] S. M. Smith and J. M. Brady, "Asset-2: Real-time motion segmentation and shape tracking.", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 814–820, 1995.

[14] F. G. Meyer and P. Bouthemy, "Region-based tracking using affine motion models in long image sequences.", *CVGIP: Image Understanding*, vol. 60, pp. 119–140, 1994.

[15] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall International Editions, 1991.

[16] Y. Bar-Shalom and X.R. Li, *Estimation and Tracking, Principles, Techniques, and Software*, Artech House, 1993.

[17] G. Welch and G. Bishop, "An introduction to the kalman filter", *in SIGGRAPH 2001*, Los Angeles, USA, 2001.

[18] B. Lucas and T. Kanade, "An iterative image registration technique withan application to stereo vision", *Proc. DARPA Image Understanding Workshop*, pp. 121–130, 1981.