OPTIMAL SECOND ORDER DERIVATIVE FILTER FAMILIES FOR TRANSPARENT MOTION ESTIMATION

Hanno Scharr

Institute for Chemistry and Dynamics of the Geosphere, ICG 3
Forschungszentrum Jülich GmbH, 52425 Jülich, Germany
H.Scharr@FZ-Juelich.de

ABSTRACT

Optimal filter families used to discretize a transparent motion model are constructed in this paper. This model is formulated as an optical-flow-like linear constraint equation. Accuracy of the estimation of its parameters heavily depends on the filter families used to implement it. To optimize filter families we derive a *model dependent* optimality criterion based on transfer functions of separable filter families with given fixed size. Using a simple optimization procedure, we demonstrate typical properties of optimal filter sets and state some useful choices. Exemplarily we show their performance on synthetic and real data from a botanical application.

1. INTRODUCTION

Optical-flow-like linear models are very well known and have a wealth of applications. In this paper we focus on a model for two transparent motions introduced by [15] and further refined by [11]. In this model second order derivatives are applied to an image sequence. We show how to design these filters adapting a recent optimization scheme [12].

Formulated via so called mixed-motion-coefficients the transparent motion model is linear and thus can be estimated using schemes designed for optical flow estimation, see e.g. [1, 6, 14] for overviews on different estimation techniques. Similar models have been designed for other complex motion patterns, where the complexity is due to (i) the motions of up to four transparent layers [11], and/or (ii) an additional change of brightness in the layers, which can be due to an additive source term, an exponential decay, or diffusion [13]. Optical flow models with physically motivated brightness changes (see e.g. [5]) are special cases thereof.

In Sec. 2 we derive the transparent motion constraint equation. How to derive an optimization scheme for occuring filter families is shown in Sec. 3. The general optimization approach is reviewed in Sec. 3.1. This optimization uses the ideal parameters estimation outcome in a model formulated in fourier domain as reference function (Sec. 3.2), thus *model dependent filter families* will be derived. The ansatz function (Sec. 3.3) is based on the transfer functions of the filters in a family. Consequently we will briefly recapitulate the transfer functions of consistent spatio-temporal derivatives needed in this paper (Sec. 3.4). A few examples of optimal filter families are discussed in Sec. 3.5, their performance is tested in Sec. 4. The impact of the proposed work is demonstrated using data from a botanical application.

Related work. Due to the well understood properties of derivative filters for numerics of PDE, the need for filter optimization has not been noticed for quite some time. A collection of heuristic filter design approaches can be found e.g. in the first sections of [7]. During the last decade many authors investigated filter optimization e.g. [4, 9, 10, 2, 8, 3, 12]. The optimization strategy used here [12] is capable of optimizing all filters discretizing a linear model as one filter family. Thus all filters optimize the same criterion simultaneously. While filters for single motion optical flow have already been presented in [12], to the best of our knowledge there is no publication focusing on filter optimization for transparent motion estimation.

This work has partly been funded by DFG SPP1114(SCHA 927/1-3).

2. ESTIMATION OF TRANSPARENT MOTION

In this section we recapitulate the derivation of a brightness constancy constraint equation (BCCE) for two transparent motions (first presented in [15]). The BCCE for standard, single motion optical flow is

$$\alpha(\mathbf{v})g = 0 \text{ where } \alpha(\mathbf{v}) := v_x\partial_x + v_y\partial_y + \partial_t \tag{1}$$

with image intensities g, partial derivatives ∂_x, ∂_y, ∂_t in x-, y-, and t-directions, respectively, and displacement vector $\mathbf{v} = [v_x, v_y]^T$. We construct a BCCE for transparent motions by successively applying α to an image sequence g [15]

$$g(\mathbf{x},t) = g_1(\mathbf{x} - \mathbf{u}t) + g_2(\mathbf{x} - \mathbf{v}t) \tag{2}$$

A basic calculation reveals that $\alpha(u)\alpha(v)g = 0$, thus

$$\begin{array}{l} u_x v_x \partial_{xx} g + (u_x v_y + u_y v_x)\partial_{xy} g + u_y v_y \partial_{yy} g + \\ (u_x + v_x)\partial_{xt} g + (u_y + v_y)\partial_{yt} g + \partial_{tt} g = 0 \end{array} \tag{3}$$

where $\partial_{xx}g$ denotes the second order partial derivative in direction x, etc. Using *mixed motion parameters* [11]

$$\begin{array}{ccc} c_{xx} := u_x v_x & c_{xy} := u_x v_y + u_y v_x & c_{yy} := u_y v_y \\ c_{xt} := u_x + v_x & c_{yt} := u_y + v_y & \end{array} \tag{4}$$

one can rewrite Eq. 3 as

$$\mathbf{d}^T\mathbf{p} = 0 \text{ with } \begin{array}{l} \mathbf{p} := [c_{xx}, c_{xy}, c_{yy}, c_{xt}, c_{yt}, 1]^T \\ \mathbf{d} := [\partial_{xx}g, \partial_{xy}g, \partial_{yy}g, \partial_{xt}g, \partial_{yt}g, \partial_{tt}g]^T \end{array} \tag{5}$$

We observe that Eq. 5 is a linear model with parameter vector $\mathbf{p}$ and data vector $\mathbf{d}$. Parameters in such a linear model can be estimated by a number of standard parameter estimation schemes used for optical flow estimation (see e.g. [1, 14]). We implemented a total least-squares scheme, based on the so called extended or generalised structure tensor [5, 11]. From the parameter vector $\mathbf{p}$ we infer velocities $\mathbf{u}$ and $\mathbf{v}$ by using the following 'trick' [11]. First, we interpret the vectors $\mathbf{u}$ and $\mathbf{v}$ as complex numbers. Then, with the definitions

$$A_0 := c_{xx} - c_{yy} + \mathbf{i}c_{xy} \qquad A_1 := c_{xt} + \mathbf{i}c_{yt} \tag{6}$$

$\mathbf{u}$ and $\mathbf{v}$ are the two solutions of the complex polynomial

$$z^2 - A_1 z + A_0 = 0. \tag{7}$$

3. FILTER FAMILY OPTIMIZATION

3.1 General Approach

The filter optimization scheme we use here operates in the 1. Brillouin zone[1] of the Fourier domain, and allows complex nonlinear optimizations of filter families [12].

We call a discrete filter family *optimal for a model* if results calculated with these filters and results calculated with ideal operators differ as little as possible. To find these filters, we minimize a cost function in a weighted L_2-Norm. The cost is the difference between a reference and an ansatz function. The reference function $f_r(\tilde{k})$ is build using transfer functions of continuous ('ideal') operators; $\tilde{k}$ denotes the wave vector normalized to Nyquist wavenumber. The ansatz function $f_a(\tilde{k},h)$ is very similar to the reference function, but is build using transfer functions of discrete operators – fixed size separable filters. The vector h denotes all free filter coefficients of the sought for filter family. As reference and ansatz functions live in Fourier domain the cost functional reads

$$c(h) = \sqrt{\frac{\int w^2(\tilde{k}) \left(f_r(\tilde{k}) - f_a(\tilde{k},h)\right)^2 \mathrm{d}\tilde{k}}{\int w^2(\tilde{k}) \mathrm{d}\tilde{k}}} \tag{8}$$

Weight function $w(\tilde{k})$ allows to specify statistical importance of different wave vectors $\tilde{k}$. Generally, optimization has to be done in the whole first Brillouin zone. The dimension of this zone depends on the dimension of the data – 3 dimensional in our case –, not on the dimension of the filter family vectors. Using a symmetric weight function $w(\tilde{k})$ and symmetric filter families allows calculation on a fraction of this space. E.g. if the weight function and all filters are symmetric or antisymmetric with respect to the coordinate axes only the positive quadrant (octant etc.) has to be processed. We minimize $e(h)$ using Matlabs `lsqnonlin`-function. In all example calculations we use the weight function

$$w(\tilde{k}) = \prod_{i=1}^{D} \cos^4(\pi \tilde{k}_i/2) \tag{9}$$

which is the transfer function of a 5-tab binomial filter, which is often used for simple preprocessing or to build a Gaussian scale space.

3.2 Reference Function

The model we want to optimize filters for reads $\mathbf{d}^T \mathbf{p} = 0$ (see Eq. 5) with the data vector $\mathbf{d}^T = [\partial_{xx}, \partial_{xy}, \partial_{yy}, \partial_{xt}, \partial_{yt}, \partial_{tt}]g$. Calculating $\mathbf{d}$ for a basis function of Fourier transform

$$g(x,\tilde{k}) = A(\tilde{k}) \exp(\mathrm{i}\pi \tilde{k}(x - x_0)) \tag{10}$$

we get the transfer function of $\mathbf{d}$

$$\hat{\mathbf{d}}(\tilde{k}) = -\pi^2 [\tilde{k}_x^2, \tilde{k}_x\tilde{k}_y, \tilde{k}_y^2, \tilde{k}_x\tilde{k}_t, \tilde{k}_y\tilde{k}_t, \tilde{k}_t^2]^T g(x,\tilde{k}) \tag{11}$$

[1] area limited by Nyquist wavenumber

As only the orientation of this vector is of relevance in the model, not its length (see Eq. 5, $\mathbf{d}$ must be normal to $\mathbf{p}$), we normalize this vector $f_r(\tilde{k}) = \hat{\mathbf{d}}(\tilde{k})/|\hat{\mathbf{d}}(\tilde{k})|$ or

$$f_r(\tilde{k}) = \frac{[\tilde{k}_x^2, \tilde{k}_x\tilde{k}_y, \tilde{k}_y^2, \tilde{k}_x\tilde{k}_t, \tilde{k}_y\tilde{k}_t, \tilde{k}_t^2]^T}{\sqrt{\tilde{k}_x^4 + \tilde{k}_x^2\tilde{k}_y^2 + \tilde{k}_y^4 + \tilde{k}_x^2\tilde{k}_t^2 + \tilde{k}_y^2\tilde{k}_t^2 + \tilde{k}_t^4}} \tag{12}$$

Thus the reference function is independent of input data g.

3.3 Ansatz Function

The Ansatz function is derived in the same way the reference function is calculated, but instead of using continuous operators for the derivatives in $\mathbf{d}$ from Eq. 5, we use fixed size derivative kernels

$$\mathbf{d}_h = [D_{xx}, D_{xy}, D_{yy}, D_{xt}, D_{yt}, D_{tt}]^T g \tag{13}$$

where the lower index h indicates that $\mathbf{d}_h$ depends on the filter coefficients of the second order derivative kernels $D_{..}$. As above, calculating $\mathbf{d}_h$ for a basis function of the Fourier transform (Eq. 10) we get the transfer function of $\mathbf{d}_h$ (where all symbols with a hat depend on $(\tilde{k})$ which we suppressed for ease of notation)

$$\hat{\mathbf{d}}_h(\tilde{k}) = [\hat{D}_{xx}, \hat{D}_{xy}, \hat{D}_{yy}, \hat{D}_{xt}, \hat{D}_{yt}, \hat{D}_{tt}]^T g(x,\tilde{k}) \tag{14}$$

Again, as only the orientation of this vector is of relevance, we normalize it: $f_r(\tilde{k},h) = \hat{d}_h(\tilde{k})/|\hat{d}_h(\tilde{k})|$ or

$$f_r(\tilde{k},h) = \frac{[\hat{D}_{xx}, \hat{D}_{xy}, \hat{D}_{yy}, \hat{D}_{xt}, \hat{D}_{yt}, \hat{D}_{tt}]^T}{\sqrt{\hat{D}_{xx}^2 + \hat{D}_{xy}^2 + \hat{D}_{yy}^2 + \hat{D}_{xt}^2 + \hat{D}_{yt}^2 + \hat{D}_{tt}^2}} \tag{15}$$

Also the ansatz function is independent of the input data g.

3.4 The needed Transfer Functions

All filters optimized in this paper are separable filters composed of 1d odd symmetric or 1d odd antisymmetric filters. Symmetric filters [2] $h(r) = [h_R, \ldots, h_1, h_0, h_1, \ldots, h_R]$ have the transfer function

$$\hat{h}(\tilde{k}) = h_0 + 2\sum_{r=1}^{R} h_r \cos(\pi r \tilde{k}) \tag{16}$$

Consistent identities (i.e. smoothing kernels) in addition have to fulfill the constraint

$$h_0 = 1 - 2\sum_{r=1}^{R} h_r \tag{17}$$

and consistent second order derivatives have to fulfill

$$\sum_{r=1}^{R} h_r = 0, \quad \sum_{r=1}^{R} h_r r = 0 \quad \text{and} \quad \sum_{r=1}^{R} h_r r^2 = 2 \tag{18}$$

First order derivatives are odd antisymmetric filters $h(r) = [h_R, \ldots, h_1, 0, -h_1, \ldots, -h_R]$ with transfer functions

$$\hat{h}(\tilde{k}) = 2\mathrm{i}\sum_{r=1}^{R} h_r \sin(\pi r \tilde{k}) \tag{19}$$

fulfilling the consistency constraints

$$\sum_{r=1}^{R} h_r = 0 \quad \text{and} \quad \sum_{r=1}^{R} h_r r = 1 \tag{20}$$

For details on consistency of filters we refer to [12].

[2] Please note that a filter has to be mirrored before it is applied to the data. Thus the usual notation for a 2-tab derivative in numerics is $[-1,1]$ in filter notation it is $[1,-1]$.

3.5 Results: Optimized Filter Sets

In the sections above we defined reference (Eq. 12) and ansatz functions (Eq. 15) using fixed size separable filters, a cost functional (Eq. 8 and Eq. 9) to calculate a distance between them. We minimized this distance for different filter sizes. When filter sizes are the same in all directions, we observe that a filter family consists of only four 1d filters: a first order derivative D^1, a second order derivative D^2 and two smoothing kernels I^1 and I^2. The 3d filters are then

$$\partial_{xy} = D^1_x * D^1_y * I^1_t \quad \text{and} \quad \partial_{xx} = D^2_x * I^2_y * I^2_t \tag{21}$$

where $*$ denotes convolution and lower indices denote the application direction. All filters not introduced above, can be derived by suitably exchanging lower indices.

For $3 \times 3 \times 3$-filters these four filters are ($c = 6.2e-09$, cmp. Eq. 8)

$$\begin{array}{ll} I^1 = [0.12026, 0.75948, 0.12026] & D^1 = [0.5, 0, -0.5] \\ I^2 = [0.21478, 0.57044, 0.21478] & D^2 = [1, -2, 1] \end{array} \tag{22}$$

for $5 \times 5 \times 5$- filters ($c = 1.6e-12$)

$$\begin{array}{ll} I^1 &= [0.01504, 0.23301, 0.50390, 0.23301, 0.01504] \\ I^2 &= [0.01554, 0.23204, 0.50484, 0.23204, 0.01554] \\ D^1 &= [0.06368, 0.37263, 0, -0.37263, -0.063681] \\ D^2 &= [0.20786, 0.16854, -0.75282, 0.16854, 0.20786] \end{array} \tag{23}$$

for $7 \times 7 \times 7$-filters ($c = 2.9e-15$; I^1, I^2, and D^2 are symmetric, D^1 is antisymmetric, please complete coefficients where indicated by "...")

$$\begin{array}{ll} I^1 &= [0.00177, 0.04910, 0.24659, 0.40508, ...] \\ I^2 &= [0.00178, 0.04909, 0.24660, 0.40506, ...] \\ D^1 &= [0.00834, 0.11282, 0.24936, 0, ...] \\ D^2 &= [0.03239, 0.18112, -0.01601, -0.39499, ...] \end{array} \tag{24}$$

and for 9-tab filters ($c = 1.7e-17$)

$$\begin{array}{ll} I^1 &= [0.00023, 0.00943, 0.07744, 0.24047, 0.34485, ...] \\ I^2 &= [0.00023, 0.00943, 0.07744, 0.24047, 0.34485, ...] \\ D^1 &= [0.00117, 0.02575, 0.12138, 0.17531, 0, ...] \\ D^2 &= [0.00502, 0.05634, 0.11698, -0.05537, -0.24594, ...] \end{array} \tag{25}$$

When filters have different sizes in different directions, for each size a set of 4 filters is needed. For $5 \times 5 \times 3$-filters ($c = 1.5e-09$, lower indices indicate the size of the filters)

$$\begin{array}{ll} I^1_3 &= [0.15158, 0.69683, 0.15158] \\ I^2_3 &= [0.14684, 0.70633, 0.14684] \\ D^1_3 &= [0.5, 0, -0.5] \\ D^2_3 &= [1, -2, 1] \\ I^1_5 &= [0.00254, 0.22288, 0.54917, 0.22288, 0.00254] \\ I^2_5 &= [0.00859, 0.21323, 0.55638, 0.21323, 0.00859] \\ D^1_5 &= [0.03885, 0.42230, 0, -0.42230, -0.03885] \\ D^2_5 &= [0.16643, 0.33429, -1.00143, 0.33429, 0.16643] \end{array} \tag{26}$$

and for $7 \times 7 \times 5$-filters ($c = 5.0e-13$)

$$\begin{array}{ll} I^1_5 &= [0.01534, 0.23312, 0.50306, 0.23312, 0.01534] \\ I^2_5 &= [0.01533, 0.23314, 0.50306, 0.23314, 0.01533] \\ D^1_5 &= [0.06433, 0.37134, 0, -0.37134, -0.06433] \\ D^2_5 &= [0.20875, 0.16500, -0.74749, 0.16500, 0.20875] \\ I^1_7 &= [0.00149, 0.04651, 0.24630, 0.41140, ...] \\ I^2_7 &= [0.00154, 0.04643, 0.24639, 0.41129, ...] \\ D^1_7 &= [0.00731, 0.11035, 0.25737, 0, ...] \\ D^2_7 &= [0.02945, 0.18576, -0.00811, -0.41419, ...] \end{array} \tag{27}$$

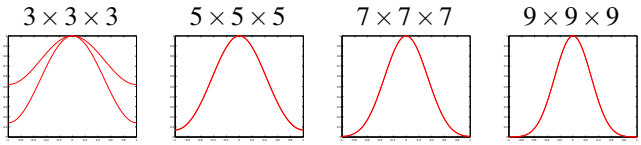

Figure 1: Transfer functions of 1d smoothing filters of the indicated filter families.

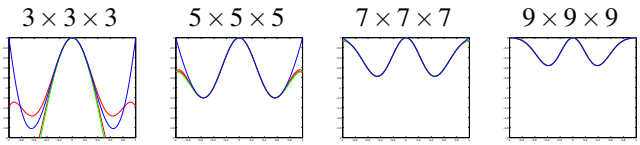

Figure 2: Transfer functions of second order derivatives calculated from the 1d filters of the indicated filter families. Red: $-\pi^2\tilde{k}^2\hat{I}^1$, and $-\pi^2\tilde{k}^2\hat{I}^2$, blue: $\mathrm{i}\pi\tilde{k}\hat{D}^1$, green: $\hat{D}^2$.

We observe that the distance measure c (cmp. Eq. 8) drops about 2 to 3 orders of magnitude when all filter sizes are increased by 2. Further, smoothing kernels I^1 and I^2 become more and more similar, the larger the filters are.

In Fig. 1 the transfer functions of the 1d smoothing kernels $\hat{I}^1$, and $\hat{I}^2$ are plotted. We observe that (inherent) smoothing increases rapidly with filter size. In Fig. 2 the transfer functions $\hat{D}^2$, $\mathrm{i}\pi\tilde{k}\hat{D}^1$, $-\pi^2\tilde{k}^2\hat{I}^1$, and $-\pi^2\tilde{k}^2\hat{I}^2$ are plotted. They are all second order derivatives. This plot is inspired by [3]. We observe that the larger the filters are, the better the derivative kernels approximate derivatives of the smoothing kernels in the same sense like Derivative-of-Gaussian filters are constructed. While [3] directly optimizes this behavior based on sampling theory reasoning, we get it without any assumptions on the filters, but via the purpose they are used for. For large filters where all filters have the same size[3], results from [3] and the ones we get are comparable. Then (and only then) the method [3] is preferable due to its much simpler, 1d optimization concept. But especially when filter sizes are not the same in each direction, e.g. when the temporal reach of the method has to be limited, results are quite different. This can be seen when comparing the filters in the $5 \times 5 \times 3$-filter family with the filters in the $3 \times 3 \times 3$- and the $5 \times 5 \times 5$-filter family. In the $5 \times 5 \times 3$-filter family the 5-tab filters try to correct the errors introduced by the 3-tab filters. Consequently there is *model specific* cross talk between filters applied in different directions. This effect gets less dramatic, when filter sizes increase, compare $7 \times 7 \times 5$-filter family with $5 \times 5 \times 5$ and $7 \times 7 \times 7$ versions.

4. EXPERIMENTS

Experimental results have been obtained by using a synthetic image sequence with ground truth and a sequence from a botanical application (see Fig. 4).

4.1 Synthetic Sequence

The image sequence used in the experiments consists of two motion layers. Each layer is generated by moving given basic patterns. The first motion layer g_1 (cmp Eq. 2) is gen-

[3] and when the weight function in the error norm is symmetric wrt. all coordinate axes

erated by moving such a pattern with known 'actual' velocity $\mathbf{u}_a = [0,-1]^T$, the second layer g_2 via $\mathbf{v}_a = [1,1]^T$. We use integer shifts in order to avoid interpolation errors. In the experiments analyzing systematic errors the optimized filters produce using the estimation scheme from [5], we use smooth noise patterns as depicted in Fig. 3. They are static i.i.d. noise patterns smoothed by a 5-tab binomial filter $[1,4,6,4,1]/16$ applied in x- and y-direction. No time-varying noise has been added to the sequences, and therefore, errors presented in Sec. 4.3 are *systematic errors* only, coming from the filter discretization and estimation process.

4.2 Error Measures

For the estimation of optical flow the most popular error measure is the *angular error* $E_{\mathbf{v}}$ (see [1], eq. 3.38) defined by

$$E_{\mathbf{v}} = \arccos(\mathbf{r}_a^T \mathbf{r}_e) \tag{28}$$

where the lower index of E indicates, which velocity is used to obtain this error, $\mathbf{r}_a = [v_{x,a}, v_{y,a}, 1]/(v_{x,a}^2 + v_{y,a}^2 + 1)^{1/2}$ is the known ('actual') ground truth spatio-temporal velocity vector of length 1 and $\mathbf{r}_e = [v_{x,e}, v_{y,e}, 1]/(v_{x,e}^2 + v_{y,e}^2 + 1)^{1/2}$ is the estimated velocity vector. A similar definition holds for velocities $\mathbf{u}$.

4.3 Results

We tested how well the optical flow fields are estimated by using a total-least-squares scheme [5]. We used several filter sets: central differences ($3\times1\times1$), 3-tab optimized ($3\times3\times3$), 5-tab optimized ($5\times5\times5$), 7-tab optimized ($7\times7\times7$), and 9-tab optimized ($9\times9\times9$) filters (cmp. Sec. 3.5).

We observe that errors are rather high for central differences and 3-tab optimized filters. Using 5-tab optimized filters, the errors drop about 1-2 orders of magnitude and another 1-2 orders of magnitude for 7-tab and 9-tab filters, respectively. The systematic errors are then much smaller than usual errors introduced by noise or model errors.

4.4 Real Sequence

The sheath region of a pine needle (see Fig. 4) consists of a semitransparent layer, becoming more and more opaque the closer we are to the base of the needle. Motion underneath this layer shall be measured in order to quantify local growth. The size of the growth zone depends on environmental parameters like temperature, humidity etc. In the current data set only rigid motion is visible, indicating that the growth zone is completely hidden in the opaque region at the base of the needle. In future data sets the growth zone is expected to be more elongated and thus be visible in the transparent region. In Fig. 4 motion fields estimated using $3\times1\times1$, $5\times5\times5$, $7\times7\times7$, and $9\times9\times9$-filters are shown. The motion field estimated using $3\times3\times3$-filters looks identical to the one where $3\times1\times1$-filters are used. Motion vectors longer than 4 pixels/frame are cut off for better visualization as a pine needle can only grow about half that fast. We observe that using simple central differences ($3\times1\times1$-filters) or $3\times3\times3$-filters, reliable results can be achieved nowhere in the transparent region. Only in the small region on the lower left of the images acceptable motion vectors are obtained, even though they appear to be too short. No covering layer (semitransparently) occludes the moving needle there. All other estimated motion vectors are too long and

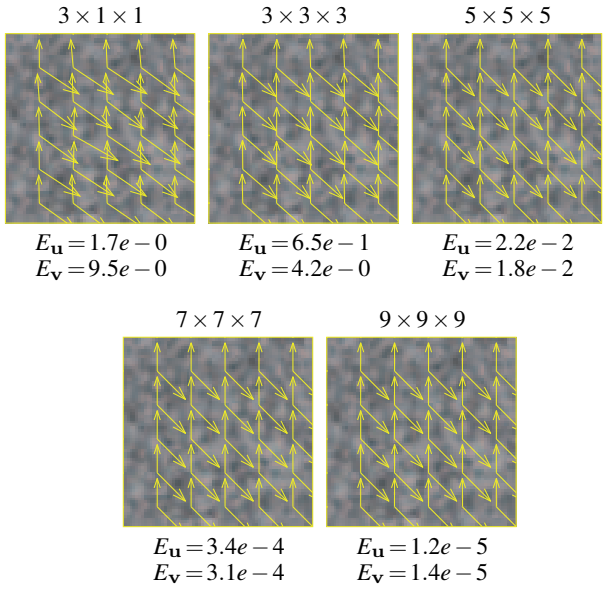

Figure 3: Flow fields using different filter sets. Filter sizes as indicated. Motion vectors are scaled by a factor 10. Errors $E_{\mathbf{u}}$ and $E_{\mathbf{v}}$ are angular errors (Eq. 28) in degree.

are thus not displayed. The larger the optimized filters, the larger and visibly more accurate the motion fields become. More vectors appear in the correct length range. At the right and towards top and bottom of the needle the motion fields become less accurate, because the semitransparent layer becomes thicker and thus more opaque there; signal to noise ratio drops severely.

Part of the positive effect of larger filters comes from the additional inherent smoothing. In order to test this, we presmoothed the data in all coordinate directions using the 9-tab filter I^1 from Eq. 25. We then estimated the motion using the $3\times1\times1$-filter set (cmp. Fig. 5). If the better performance came from the smoothing only, the result now should resemble the result using $9\times9\times9$-filters. But it remains below the performance depicted in Fig. 4.

5. CONCLUSION AND SUMMARY

In this paper we demonstrated how to optimize filter families used to discretize a transparent motion model. The ansatz and reference functions are derived from the orientation of the data vector of the linear motion model. The resulting separable, fixed size filter families are similar to the ones introduced in [3], if (and only if) all sizes of 1d filters in the family are the same and the weight function used in the cost functional is symmetric. Then smoothing kernels and inherent smoothing of derivative kernels must match across all filters applied *in the same direction*. The optimization scheme from [3] optimizes this behavior, thus – in this special case – it is preferable due to its simpler, 1d implementation. In all other cases, the optimization procedure put forward here yields better filter kernels, best adapted to the model they are used in. E.g. when filter sizes vary, larger filters try to compensate errors introduced by smaller ones. 1d optimization independent from the model to discretize is then no longer applicable. In addition to the filter coefficients, this paper

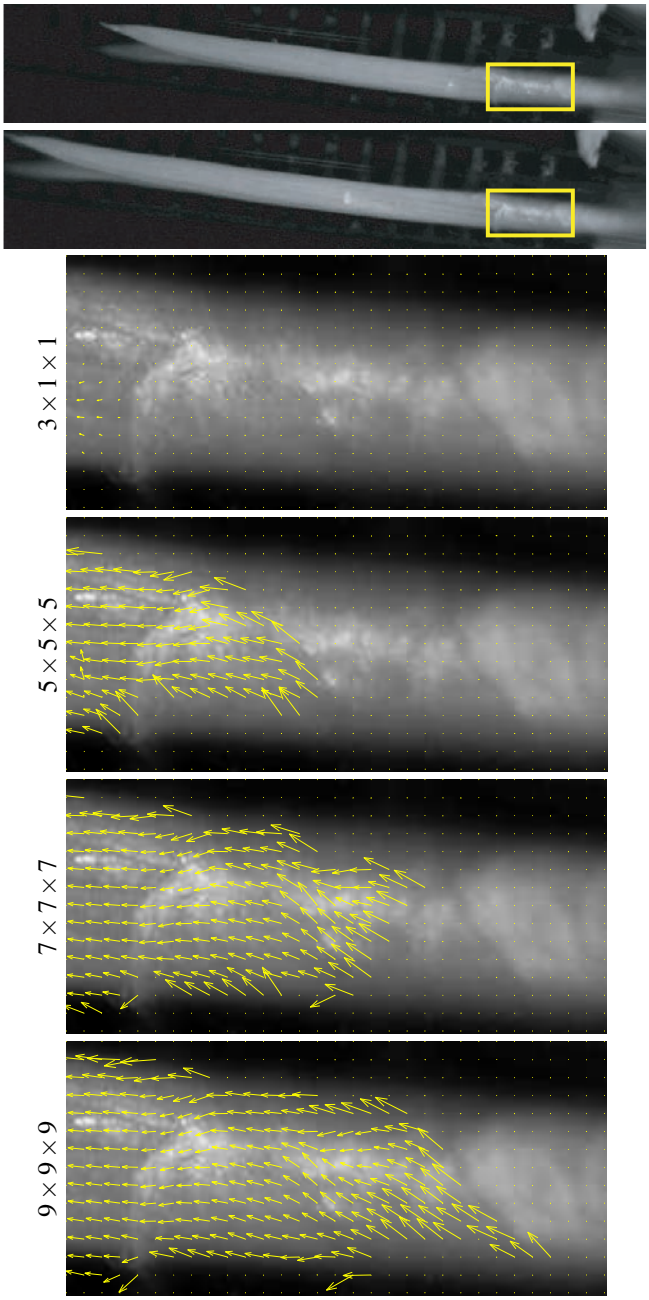

Figure 4: Growing pine needle. Top two images: 1^{st} and 100^{th} image of the sequence, the frames indicate the 256×128 sheath region. Then from top to bottom: motion estimation results on the sheath region using $3 \times 1 \times 1$, $5 \times 5 \times 5$, $7 \times 7 \times 7$, and $9 \times 9 \times 9$-filters. Vectors longer than 4 pixels/frame are cut off. Vectors are scaled by a factor of 4.

demonstrates the reduction of systematical errors in transparent motion estimation using these filters. The error drops about 1-2 orders of magnitude when increasing filter size by 2 tabs. But the larger the filters are, the more smoothing they introduce, reducing spatio-temporal resolution of the estimated motion field. If regularization has to be small in a given application a trade off between small filter size and small systematical error has to be found.

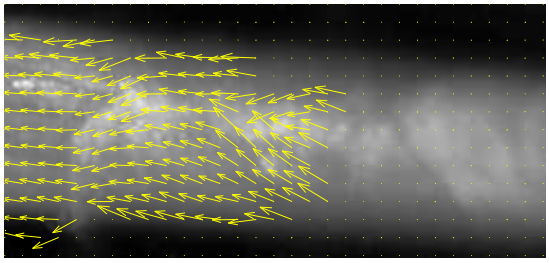

Figure 5: Motion field from growing pine needle using data presmoothed by 9-tab I^1 (Eq. 25) and $3 \times 1 \times 1$-filters. Vectors longer than 4 pixels/frame are cut off. Vectors are scaled by a factor of 4.

REFERENCES

[1] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[2] M. Elad, P. Teo, and Y. Hel-Or. Optimal filters for gradient-based motion estimation. In *ICCV'99*, 1999.

[3] H. Farid and E. P. Simoncelli. Differentiation of discrete multi-dimensional signals. *IEEE Trans Image Processing*, 13(4):496–508, April 2004.

[4] H. Farid and E.P. Simoncelli. Optimally rotation-equivariant directional derivative kernels. In *7th Int'l Conf Computer Analysis of Images and Patterns*, Kiel, 1997.

[5] H. Haußecker and D. J. Fleet. Computing optical flow with physical models of brightness variation. *PAMI*, 23(6):661–673, June 2001.

[6] H. Haußecker and H. Spies. Motion. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook of Computer Vision and Applications*. Academic Press, 1999.

[7] B. Jähne, H. Scharr, and S. Körkel. Principles of filter design. In *Handbook of Computer Vision and Applications*. Academic Press, 1999.

[8] Behz. Kamgar-Parsi, Behr. Kamgar-Parsi, and A. Rosenfeld. Optimally isotropic laplacian operator. *IEEE Trans. Img. Proc.*, 8(10), Oct. 1999.

[9] H. Knutsson, M. Andersson, and J. Wiklund. Multiple space filter design. In *Proc. SSAB Symposium on Image Analysis*, Göteborg, Schweden, 1998.

[10] H. Knutsson, M. Andersson, and J. Wiklund. Advanced filter design. In *Proc. SCIA*, 1999.

[11] Cicero Mota, Ingo Stuke, and Erhardt Barth. Analytic solutions for multiple motions. In *Proceedings of the International Conference on Image Processing*, pages 917–920, 2001.

[12] H. Scharr. Optimal filters for extended optical flow. In *International Workshop on Complex Motion 2004, LNCS 3417*.

[13] H. Scharr, I. Stuke, C. Mota, and E. Barth. Estimation of transparent motions with physical models for additional brightness variation. In *13th European Signal Processing Conference, EUSIPCO*, 2005.

[14] C. Schnörr and J. Weickert. Variational image motion computation: Theoretical framework, problems and perspectives. In *DAGM'2000*, pages 476–487, 2000.

[15] M. Shizawa and K. Mase. Simultaneous multiple optical flow estimation. In *ICPR'90*, pages 274–278, 1990.