3D-Orientation Signatures with Conic Kernel Filtering for Multiple Motion Analysis

Weichuan Yu
Dept. of Diagnostic Radiology
Yale University
weichuan@noodle.med.yale.edu

Gerald Sommer
Institut für Informatik
Universität Kiel
gs@ks.informatik.uni-kiel.de

Kostas Daniilidis
GRASP Laboratory
University of Pennsylvania
kostas@grasp.cis.upenn.edu

Abstract

In this paper we propose a new 3D kernel for the recovery of 3D-orientation signatures. The kernel is a Gaussian function defined in local spherical coordinates and its Cartesian support has the shape of a truncated cone with axis in radial direction and very small angular support. A set of such kernels is obtained by uniformly sampling the 2D space of polar and azimuth angles. The projection of a local neighborhood on such a kernel set produces a local 3D-orientation signature. In case of spatiotemporal analysis, such a kernel set can be applied either on the derivative space of a local neighborhood or on the local Fourier transform. The well known planes arising from one or multiple motions produce maxima in the orientation signature. Due to the kernel's local support spatiotemporal signatures possess higher orientation resolution than 3D steerable filters and motion maxima can be detected and localized more accurately. We describe and show in experiments the superiority of the proposed kernels compared to Hough transformation or EM-based multiple motion detection.

1 Introduction

The motivation of our approach is the local detection and estimation of multiple motions in spatiotemporal imagery. Optical flow estimation has been extensively studied and the reader is referred to the surveys [4, 12] for an overview of existing methods. While research in single motion estimation is already mature, estimation and analysis of multiple motions (i.e. occlusion and transparency) are still challenging problems.

In this paper we focus on the estimation of multiple motions from the spatiotemporal orientation aspect. Motion estimation was first addressed from the point of view of orientation analysis by Adelson and Bergen [1] who pointed out that motion is equivalent to spatio-temporal orientation. They introduced a spatio-temporal energy model for single motion representation. Knutsson proposed the 3D structure tensor for orientation recovery and this approach was followed by Bigün [6], Jähne [18], and others. To describe multiple motions, Shizawa *et. al.* [23, 22] proposed the superposition principle. Fleet and Langley [7] as well as Beauchemin and Barron [5] analyzed the spectral structure of occlusion and transparency in detail. Transparency can be described as two planes of energy concentration in the spectral domain only, while occlusion produces two planes both in the spectrum as well as in the spatiotemporal domain accompanied by distortion [5]. The corresponding motion parameters are determined by the normal vectors of these planes. Determining the precise orientation of two motion planes, however, remains a difficult task in particularly when the angle between two motion planes is small and energy is concentrated in the low frequencies.

Many authors proposed spectral sampling with Gabor or similar filters [13, 10, 14, 27] in order to detect the motion planes in the frequency domain. One of the main concerns of these approaches is the enormous complexity of computation in sampling the spectrum with fine resolution. To resolve the conflict between performance and complexity, the concept of steerability was introduced [9] and many 2D steerable filters have been applied in image processing and low level computer vision [19, 20, 24, 8]. But only few approaches dealt with 3D steerability. These approaches either steer derivatives of Gaussians [9, 25] or construct the steerable filter directly in the spectral domain [2]. To achieve high orientation resolution, a huge number of basis functions is required whose support is the entire sphere of orientations. Since detection of multiple motions presumes a high orientation resolution either in the spatiotemporal or in the frequency space current steerability approaches proved to be impractical.

This motivated us to construct a new 3D-kernel with conic support in the Cartesian spatiotemporal space. The 3D-kernel is a Gaussian defined in the space of polar and azimuth angles with conic profile in the radial direction. The local image feature space is projected onto a huge number of such kernels with tiny support and a signature with high orientation resolution is obtained. Because of the tiny support of filters the way how these filters decompose the

sphere is of practically minor importance.

We describe how we can obtain such orientation signatures in the image derivative space or in the local Fourier domain. We compare this kernel projection to the Hough transform and to expectation-maximization (EM) multiple plane estimation (section 3). We show experiments with both occlusion and transparency sequences in section 4.

2 Conic Kernels

2.1 Definition

We compute a local spherical mapping on the input data: $I(x,y,z) \rightarrow I(r,\theta,\phi)$, where $r = \sqrt{x^2+y^2+z^2}$, $\theta = \arctan(\frac{y}{x})$, $\phi = \arctan(\frac{z}{\sqrt{x^2+y^2}})$ (figure 1). In order to have fine orientation resolution, we use *conic kernels* with small angular supports to sample the orientation space locally. A *conic kernel* centered at (θ_i, ϕ_j) reads

$$K_{(\theta_i,\phi_j)}(r,\theta,\phi) := \frac{G_0^{(\theta_i,\phi_j)}(\theta,\phi)}{\mathcal{N}_{R_{\min},R_{\max}}^{(\theta_i,\phi_j)}(r)}, \qquad (1)$$

where $\mathcal{N}_{R_{\min},R_{\max}}^{(\theta_i,\phi_j)}(r)$ is a compensation function along the radial direction described later. The angular part of the kernel is a 2D Gaussian function in the (θ,ϕ)-space:

$$G_0^{(\theta_i,\phi_j)}(\theta,\phi) := \frac{1}{2\pi\sigma^2} e^{-\frac{(\mathcal{D}(\theta,\theta_i))^2+(\phi-\phi_j)^2}{2\sigma^2}}. \qquad (2)$$

As the azimuth angle θ is periodic, we define $\mathcal{D}(\cdot)$ to represent the minimal circular difference between θ and θ_i $(\theta,\theta_i \in [0°, 360°))$

$$\mathcal{D}(\theta,\theta_i) := \min(|\theta-\theta_i|, |\theta-\theta_i-360°|, |\theta-\theta_i+360°|).$$

Theoretically, a Gaussian function is not compactly supported. To form an FIR filter we cut off the angular part of $G_0^{(\theta_i,\phi_j)}(\theta,\phi)$ at the boundary of a circular mask with a fixed diameter D. The diameter D is usually a function of σ and we set $D = 6\sigma$ so that the energy loss of the cut-off area is negligible. Figure 1 shows one example of such a *conic kernel*.

After applying such a *conic kernel* on $I(r,\theta,\phi)$ we obtain a sample at (θ_i,ϕ_j)

$$A_{(\theta_i,\phi_j)} := \sum_{\{(\theta,\phi)|\sqrt{(\theta-\theta_i)^2+(\phi-\phi_j)^2} \le \frac{D}{2}\}} \sum G_0^{(\theta_i,\phi_j)}(\theta,\phi) \sum_{r=R_{\min}}^{R_{\max}} \frac{I(r,\theta,\phi)}{\mathcal{N}_{R_{\min},R_{\max}}^{(\theta_i,\phi_j)}(r)}. \qquad (3)$$

Now let us consider the sampling of the (θ,ϕ) plane using a set of *conic kernels*. A sphere surface forms a rectangular

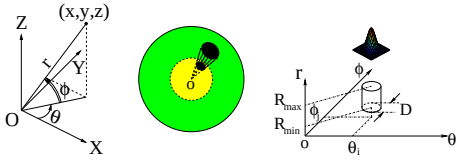

Figure 1: A *conic kernel* centered at (θ_i,ϕ_j) with radial boundaries $R_{\min}$ and $R_{\max}$. **Left:** The definition of the spherical coordinate system. **Middle:** The filter kernel in the 3D Cartesian coordinate system. The keypoint is at the center of the sphere. **Right:** The filter kernel with θ, ϕ and r as coordinates. The *conic kernel* turns into a cylinder with a diameter D. In the (θ,ϕ) plane the circular mask of the cylinder is weighted by a 2D Gaussian function, as shown above the cylinder.

region in the (θ,ϕ) plane, which is periodic along the θ direction and is mirror-symmetric about the boundary along the ϕ direction. We let neighboring kernels overlap in order to cover the entire rectangular region and use the periodicity along the θ direction and mirror-symmetry along the ϕ direction to solve the boundary problem. The number of required *conic kernels* in sampling the entire rectangular region is determined by the scale parameter σ (cf. equation (2)) as well as the sampling step parameter (i.e. the angular distance between the centers of two neighboring sampling masks, cf. figure 2). In this paper we set the horizontal- and vertical- sampling step to be the same as $\delta\theta = \delta\phi = 3\sigma$. As the entire (θ,ϕ) plane has a range of $-180° \le \theta < 180°$ and $-90° \le \phi \le 90°$, by using $\sigma = \frac{1}{3}°$ we need totally $360 \times 181 = 65160$ *conic kernels* to sample the (θ,ϕ) plane with a resolution of 1°. All *conic kernels* have very narrow angular support keeping thus complexity in a moderate level.

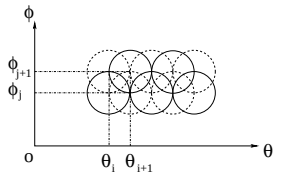

Figure 2: One example of sampling (θ,ϕ) plane with a set of *conic kernels*. The horizontal or vertical distance between two neighboring masks is equal to the radius of one mask.

The proposed decomposition of the sphere is not uniform and theoretically it produces a non-rotation-invariant signature. However, due to the huge numbers of filters and their tiny support the decomposition has practically the same effect with uniformly sampling the sphere with the same huge number of kernels whose centers in that case would be a subdivision of the icosahedron. If we apply, as above, 65160 filters of very small support we obtain approximately the same result. The reason why we prefer regular decom-

position lies mainly in the simplicity of indexing and displaying.

We build a look-up-table (LUT) "off-line" to store the local spherical mapping. The online application of the LUT is of negligible complexity compared with calculating the filter responses. The LUT-based mapping can be applied both in the spatiotemporal as well as in the spectral domain, though the filter support in figure 1 is only displayed in the spatial domain.

To obtain a continuous orientation signature $S(\theta, \phi)$ from the discrete one we use 2D Gaussian functions with local support $G_0^{(\theta_i,\phi_j)}(\theta, \phi)$ as interpolation functions:

$$S(\theta, \phi) := \sum_{\theta_i=-180^\circ}^{179^\circ} \sum_{\phi_j=-90^\circ}^{90^\circ} A_{(\theta_i,\phi_j)} G_0^{(\theta_i,\phi_j)}(\theta, \phi). \quad (4)$$

This constitutes an approximation and not an interpolation of orientation signature and appears also in Radial Basis Functions approaches [21].

2.2 Comparisons with Current 3D Steerable Filters

Current 3D-steerable filters are rotated copies of either 3D-Gaussians [9, 25] or specified basis filters in frequency space [2]. Let us consider first the n-th derivative of 3D Gaussians along the x-axis (we omit normalization constants) $G_n = \frac{\partial^n}{\partial x^n} \exp\{-(x^2 + y^2 + z^2)/2\}$. The angular terms in the first three derivatives in the spherical coordinates are then $-\cos(\theta)\cos(\phi)$, $\cos^2(\theta)\cos^2(\phi)$ (we omit the term -1 from the actual representation $r^2\cos^2(\theta)\cos^2(\phi) - 1$ because it makes no difference to angular variation), and $3\cos(\theta)\cos(\phi) - \cos^3(\theta)\cos^3(\phi)$, respectively. All of them are different combinations of spherical harmonic functions.

Andersson [2] designed an alternative 3D steerable filter directly in the frequency domain. He designed the spectral basis filters as $B_{li}(\overline{u}) = G(\rho)(\hat{n}_{li} \cdot \hat{u})^l$, where $\overline{u}$ and $\hat{u}$ are an arbitrary frequency coordinate vector and its corresponding normalized unit vector, respectively. The vector $\hat{n}_{li}$ denotes the orientation of the i-th basis filter of order l, and $G(\rho)$ represents the radial frequency response.

The main drawback of both approaches is their coarse orientation resolution. In this paper we will not delve into the quantitative definition of orientation resolution due to space limitation. Instead, we provide an illustrative proof. For a filter, its angular support is a natural indicator of the orientation resolution: The smaller the angular support, the finer the orientation resolution. In figure 3 we show the first Gaussian derivative G_1, Andersson's third order filter B_3 in frequency domain [2], and our *conic filter*, respectively. In the bottom row we show their angular supports in the (θ, ϕ) space. Note that the angular support of a filter like Andersson's in the spatial domain is the same as that in the frequency domain since the Fourier transform is an isometric mapping (i.e. it keeps angles). The irregularity of the Gaussian derivative in the (θ, ϕ) space with $|\phi| > 40^\circ$ is caused by the discrete representation of filter kernels. The Gaussian derivative G_1 has such a large angular support that only the gap between its two lobes (shown as the black curve) may be useful. Actually, Huang and Chen used this gap to obtain the orientation of *one* plane in single motion estimation [17]. The orientation resolution of Andersson's filter is better (cf. figure 7 as well) but still lower than in the *conical kernel*. The reason why no steerable filters are applied in multiple motion estimation stems exactly from the resolution limitation of current steerable filters. In contrast, our filter has a much smaller angular support which enables us to analyze multiple orientations more precisely.

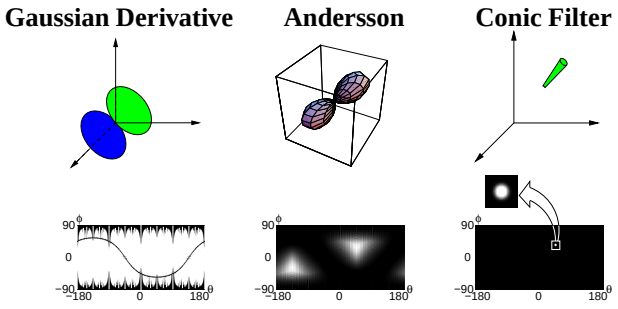

Figure 3: **Top:** The Gaussian derivative G_1 (left, redrawn from [17]), B_3 (middle, redrawn from [2]), and our filter (right) centered at $\theta = 45.00^\circ, \phi = 35.26^\circ$. **Bottom:** The corresponding angular supports of the kernels shown with white regions in the (θ, ϕ) space indicate their orientation resolution.

The computational burden of applying a steerable filter is determined by the number of basis filters and the spatial support of each basis filter. Given the fact that current steerable filters are based on a global decomposition principle and our filter is based on a local decomposition principle, it is more reasonable to compare their complexity by considering the computational burden per pixel in the input data. Concretely,

- The Gaussian derivative G_1 is composed of three basis filters of global support. Each pixel in the input data is therefore involved in the scalar product as well as in the interpolation procedure three times.
- Andersson's B_3 filter has ten basis filters. Thus, each pixel is involved ten times.
- In our filter the quadratic area ($\theta_i \le \theta \le \theta_{i+1}$, $\phi_j \le \phi \le \phi_{j+1}$) is covered by four quadrant masks (cf. figure 2). Roughly speaking, a pixel in this area

is involved in the scalar product four times. As the interpolation function has the same support as the *conic kernel*, a pixel is also involved in the interpolation four times.

We should be aware of the point that a complexity comparison is fair only when the corresponding filters are comparable in orientation resolution. The *conic kernel* is more efficient than Andersson's B_3 but slightly less efficient than the first Gaussian derivative which however lacks in orientation resolution.

Another possibility to achieve such a fine orientation resolution with a global decomposition method is to generalize the filter design technique used by Simoncelli and Farid [24] from 2D to 3D space and use flexible combinations of prechosen spherical harmonic components to form a 3D filter with narrow angular shape. However, as we already pointed out in [29], we would also have to face the considerably higher computation effort in order to build a 3D filter mask with narrow angular shape (In order to achieve the same fine orientation resolution in 2D space, a steerable filter using global decomposition method needs generally about ten times more computation than the 2D version of our *conic filter* [29]. In 3D space this difference will be even larger due to the increase of dimension). One might think that higher order derivatives would increase orientation resolution. This can hardly be achieved because Gaussian derivatives are fixed combinations of spherical harmonics – the reader may plot the angular supports of Gaussian derivatives for an illustrative proof.

The proposed conic filtering is also related to 3D orientation histograms obtained usually in gradient space. It differs in the sampling of the orientation space: the conical supports in the angle space here overlap whereas the orientation histogram follows merely the Hough sampling principle [16].

2.3 Compensation via Radial Variation

In this section, we address the design of the weighting function $\mathcal{N}(r)$ (cf. equation (1)). In the spherical coordinates the azimuth angle θ and the polar angle ϕ are defined differently. All points with the same θ on a sphere surface lie on a great circle of this sphere, whereas all points with the same ϕ (except $\phi = 0°$) lie on a small circle. If we divide the whole (θ, ϕ) space with a homogeneous grid, it is easy to see that the higher the latitude value is, the denser the grid points are on the sphere surface [15]. We may establish the weighting function $\mathcal{N}(r)$ as the sum of discrete weights in the filter kernels to compensate the non-uniform distribution on the sphere surface. The consequence is that we are no more able to know the real distribution density information on the sphere surface. However, the density information is desirable in many motion estimation approaches. Thus, we would like to preserve the distribution density information by simply setting $\mathcal{N}(r) = 1$.

2.4 Conic Kernel Response to a 3D-Plane

In the 3D Cartesian coordinate system, a plane passing through the origin $(0, 0, 0)$ with a unit normal vector $\mathbf{n} = (n_1, n_2, n_3)^T$ reads

$$xn_1 + yn_2 + zn_3 = 0. \tag{5}$$

In order to represent a plane with parameters θ and ϕ, we convert the Cartesian coordinates into spherical coordinates $(x, y, z) \to (r, \theta, \phi)$ and $(n_1, n_2, n_3) \to (1, \theta_n, \phi_n)$. Elimination of r yields an equation with variables θ and ϕ

$$\cos(\phi)\cos(\phi_n)\cos(\theta - \theta_n) + \sin(\phi)\sin(\phi_n) = 0. \tag{6}$$

For horizontal and vertical planes with normal vectors parallel to the coordinate axes, their corresponding representations in the (θ, ϕ) space are straight lines. In motion analysis we usually encounter tilted planes which, in the (θ, ϕ) space, turn into harmonic curves with different amplitudes and phases (cf. figure 4). For each curve, the normal vector of the corresponding plane is determined by the coordinates ϕ_m and θ_m of maximal response as follows:

$$\begin{cases} \theta_n &= \theta_m + 180° \\ \phi_n &= 90° - \phi_m. \end{cases} \tag{7}$$

The θ_n and ϕ_n are then used in motion estimation

$$\begin{cases} u &= \cos(\theta_n)\cot(\phi_n) \\ v &= \sin(\theta_n)\cot(\phi_n) \end{cases}. \tag{8}$$

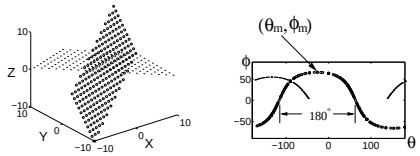

Figure 4: **Left:** A plane with normal vector $(-2, 1, 1)$ (plotted with small circles) and a plane with normal vector $(1, 1, 1)$ (plotted with dots) in the Cartesian coordinates. **Right:** The corresponding curves in the (θ, ϕ) space. See text for details about the extreme point. The curve corresponding to the second plane has only positive ϕ coordinates.

On the the θ axis, θ_m lies in the middle of two zero-crossing points which have a distance of $180°$. This extra geometry constraint is very useful in determining the number of motions automatically as well as in obtaining reasonable initial values of motion parameters. In practice, we obtain a set of points in the (θ, ϕ) space. Extracting the parameters (θ_n, ϕ_n) from these points is then a standard regression problem. For a single curve least square estimation is applicable; for multiple curves we may apply the EM algorithm as described below.

3 Motion Estimation Using Conic Filtering

3.1 Algorithm

1. Fix radial parameters $R_{\min}$ and $R_{\max}$ as well as the angular parameter σ which determines D, $\delta\theta$, and $\delta\phi$ automatically. Also fix another threshold parameter η ($\eta = 2^\circ$).
2. Set $\theta_i = -180^\circ$, $\phi_j = -90^\circ$;
3. If $\theta_i < 180^\circ$
   if $\phi_j \leq 90^\circ$
   apply the filter centered at (θ_i, ϕ_j) on the local derivative space or the local energy spectrum by using the LUT (cf. $A_{(\theta_i,\phi_j)}$ in eq. (3));
   $\phi_j = \phi_j + \delta\phi$;
   end
   $\theta_i = \theta_i + \delta\theta$;
   end
4. Cluster the nonzero $A_{(\theta_i,\phi_j)}$ near θ axis (i.e. $-\eta \leq \phi \leq \eta$) into the same group if their distance is less than 2η.
5. If the centroids of two groups have a distance $\in [180^\circ - \eta, 180^\circ + \eta]$, these two groups form a group-pair. The number of group-pairs indicates the number of motions.
6. For each group-pair, search along the positive ϕ direction from their middle point and cluster the nonzero $A_{(\theta_i,\phi_j)}$ into different polar groups like in step 4. The weight-center of the vertical group gives us a guess of (θ_m, ϕ_m) and consequently an initialization of (θ_n, ϕ_n) (cf. eq. (7)).
6. Apply eq. (6) based EM to get final (θ_n, ϕ_n) for motion estimation (cf. eq. (8)).

Since the equation (5) based 3D Hough transform as well as the planar EM algorithm can extract the orientation parameters of planes **directly**, the readers may ask why we project the 3D data onto the 2D feature space before parameter extraction. The answer lies in the following analysis of the 3D Hough transform and the EM algorithm.

3.2 Comparison with Hough Transform and EM estimation

The Hough transform [16] is a sample and search method for parameter extraction. Concretely, for a set of points coordinated with $(I_{ix}, I_{iy}, I_{it})(i = 1, \cdots, N)$ we draw the corresponding vectors in the (n_1, n_2, n_3) space satisfying the equation (5). Then we search in the (n_1, n_2, n_3) space the position with the maximal number of vector intersections to obtain the desired normal vector (n_{1m}, n_{2m}, n_{3m}). This vector is used for motion estimation

$$\begin{cases} u_m &= \frac{n_{1m}}{n_{3m}} \\ v_m &= \frac{n_{2m}}{n_{3m}} \end{cases} . \tag{9}$$

Practically, we sample the speed space (i.e. (u_m, v_m)-space) with a finite interval and relax the orthogonality criterion with a positive threshold ε yielding

$$\mid I_{ix} u_m + I_{iy} v_m + I_{it} \mid \leq \varepsilon . \tag{10}$$

The equation (10) based 3D Hough transform is equivalent to a 3D filter with a concave disk shape centered at the origin of the 3D space (cf. figure 5). The comparison between our filter shape (figure 1) and the shape of the disk leads to the conclusion that our filter samples the orientation space more efficiently than the 3D Hough transform. The conclusion is also confirmed by the Hough image of a point in figure 6. The Hough image is actually the impulse response of the concave disk filter. It is very similar to our filter response of a 3D plane except that the Hough image has no negative ϕ value (we only use normal vectors with $n_3 > 0$). Taking into account that the filter response of a 3D plane consists of plenty of filter responses of points we justify the above conclusion easily. The aforementioned superiority enables our filter to reduce the enormous memory requirement in Hough based approaches [28], especially the gigantic overlapping of the Hough curves (figure 6). As a result, we can extract the parameters of motion planes with much less complexity.

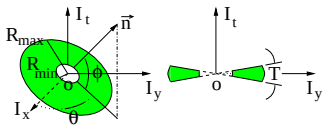

Figure 5: The 3D Hough transform is equivalent to a filter with a concave disk shape. **Left:** General projection plot of the filter mask. The vector **n** is normal to the filter mask. **Right:** Side view of the filter mask. The angular thickness T of the disk is determined by the clustering threshold ε in equation (10).

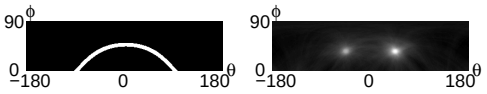

Figure 6: **Left:** Vectors satisfying eq. (10) form a curve similar to our filter response of a 3D plane. The width of the curve is determined by the clustering threshold ε in eq. (10). **Right:** The Hough image of an occlusion sequence (cf. figure 7).

Since the intersections of different curves in the Hough image are blurred due to the introduction of ε, the global maximal position is no more a peak, but a smooth *uni-modal* distribution. While the search of the global maximal position is still feasible, the search of the second maximal position is generally problematic because the properties of the *uni-modal* distribution are unknown and we do not know how to get rid of the neighbors of the global maximum automatically in seeking the second maximum. Besides, even by choosing the neighborhood manually, the second maximum is blurred and its position is *biased* by the distribution around the first maximum. Both will result in an inaccurate estimation.

The EM algorithm consists of subsequent iterations of the expectation and maximization step until there is no significant difference in the parameter estimates. In the expectation step, the membership weights of points are updated by the new results of parameter estimation; in the maximization step, we use the usual maximum likelihood method to estimate parameters with the updated assignment of points to groups.

Since the EM algorithm is an iterative method, it has no closed-form solution. Generally, we do not know the number of motions exactly. Unlike other implicit constraints [3, 11, 26], other filter helps to determine the number of motions explicitly. Moreover, convergence and robustness of the EM algorithm are very much dependent on the initial values. Using the orientation signature of our filter we can facilitate a good initial value close to the correct solution.

4 Experiments

We begin with an artificial occlusion sequence in figure 7. The occluding signal has constant flow of $(1, 1)$ [pixel/frame] and the occluded signal has flow of $(1, -1)$. We use the impulse response of the first Gaussian derivative with a support of $5 \times 5 \times 5$ pixels and a $33 \times 33 \times 1$ window for orientation analysis in the derivative space. For spectral orientation analysis we choose a $32 \times 32 \times 32$ window and adapt all spectral components in this window. Here we cannot take a narrower mask like in the derivative space because otherwise the spectral resolution will be too coarse. The orientation signatures using our filter show two distinct curves whose extreme points locating near $(-135°, 54°)$ and $(135°, 54°)$, respectively. In contrast, the orientation signatures of applying G_1 and B_3 clearly fail to provide the correct structure of multiple planes: Though we observe some blurred peaks in both signatures, these peaks either are at the wrong position (G_1) or only correspond to the dominant curve (B_3).

Table 1 lists the EM estimation results using our spatial orientation signature and spectral orientation signature. Here we take the orientation signatures both with and without averaging compensation to confirm the analysis in section 2.4. In the first test we set initial values arbitrarily. In the second test we take proper initialization according to the extreme point analysis introduced in section 2.5. The estimation results without averaging compensation are better than the results with compensation and proper initialization reduce the number of iterations in the EM algorithm greatly. Besides, the data quality in the spatial orientation signature is good enough so that the estimation results with arbitrarily initial values are as precise as the results with properly initial values. But the curves in the spectral orientation signature are blurred and the EM algorithm is susceptible to be blocked

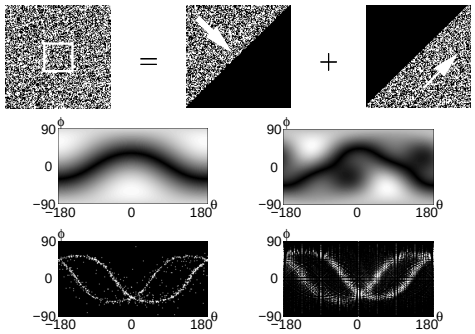

Figure 7: **Top:** One frame of an occlusion sequence can be separated as overlapping of two motions. The white window indicates the multiple motion region in which we compare orientation signatures of different filters. The white arrows denote the moving directions and the black regions denote static background. **Middle:** The amplitudes of orientation signatures using G_1 (left) and B_3 (right). **Bottom:** The orientation signatures using our filter in the derivative space (left) and in the spectral domain (right).

by a local minimum. Properly initial values help the EM algorithm converge to the the desired global minimum. The spectral estimation results are not comparable to the spatial estimation results because the data quality are not in the same level, as already shown by two orientation signatures.

model	initial values	compensation	iteration	occluding	occluded
spatial model	arbitrarily set	yes	3	$(0.927, 0.998)$	$(0.949, -0.971)$
		no	3	$(0.986, 0.999)$	$(0.986, -0.988)$
	properly set	yes	1	$(0.938, 1.005)$	$(0.923, -0.960)$
		no	1	$(0.980, 0.997)$	$(0.963, -0.974)$
spectral model	arbitrarily set	yes	7	$(1.187, 1.194)$	$(1.112, -1.147)$
		no	4	$(0.898, 0.948)$	$(1.106, -1.099)$
	properly set	yes	2	$(1.182, 1.191)$	$(1.110, -1.145)$
		no	2	$(0.966, 1.002)$	$(1.007, -1.026)$

Table 1: Estimation results of the occlusion sequence shown in figure 7. We use $(u_{10}, v_{10}) = (0.8, 0.3)$ and $(u_{20}, v_{20}) = (1.2, -0.1)$ as arbitrary initialization and the properly initial values are set as $(u_{10}, v_{10}) = (0.9, 1.1)$ and $(u_{20}, v_{20}) = (0.9, -1.1)$. For both approaches we use the same tolerance parameter $\sigma_r = 0.1$.

In order to test the performance of the EM algorithm on determining the number of motions, we propose an example of a single moving signal with a velocity $(1, -1)$. Both spatial and spectral EM algorithms should converge to one speed even with arbitrarily initial values if they are able to determine the number of motions automatically. With the initial values $(1.2, -0.1)$ and $(0.8, 0.3)$ the spatial EM algorithm converges to $(0.995, -1.001)$ after 2 iterations and the spectral EM algorithm converges to $(1.057, -1.045)$ and $(0.951, -1.011)$ after 2 iterations. Taking into account that the spectrum of the sequence is blurred, the result is not surprising. To confirm if the spectral EM algorithm converges with the properly initial values, we run the program again by setting both initial values as $(0.9, -1.1)$. This time

the spectral EM algorithm converges to $(1.004, -1.029)$ after 2 iterations. Thus, we verify that the EM algorithm cannot the number of motions exactly and the properly initial values play a critical role for data with "bad" quality.

In figure 8 is an occlusion sequence consisting of an occluding signal moving right with a velocity of about $(1, 0)$ and an occluded signal moving left at about $(-1, 0)$. Using this knowledge we compare the orientation resolutions of different filters. Inside the white window in the 16-th frame we apply G_1, B_3, and our *conic filter* in the derivative space to obtain orientation signatures. Both G_1 and B_3 can only roughly indicate the curve with the extreme point $(180°, 45°)$ with blurred peaks. Our filter provides a reasonable signature. Its two extreme points lie near $(0°, 45°)$ and $(180°, 45°)$ and are ideally consistent with the motions.

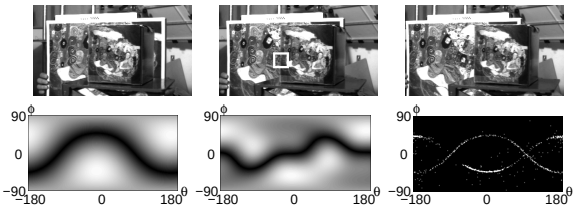

Figure 8: **Top:** The first, 16-th and 32-th frames of an occlusion sequence. The white window in the 16-th frame indicates an occlusion region. **Bottom:** The amplitudes of orientation signatures applying G_1 (left), B_3 (middle), and our *conic filter* (right) in the derivative space of the white window in frame 16.

Figure 9 shows the well known "flower garden" occlusion sequence. In one multiple motion region (white window) we calculate the partial derivatives and apply G_1, B_3 and our *conic filter* to obtain orientation signatures in the derivative space (cf. row 2) for resolution comparison. To demonstrate the entire procedure of multiple motion estimation, we first estimate motions with the single motion model. At the occlusion boundaries the results are not correct. After the eigenvalue analysis [18] we detect two motion candidate regions and the regions with the aperture problem. Only in the multiple motion candidate regions apply we the spatial EM algorithm to estimate motions (row 4).

In figures 10 we demonstrate a real example of transparency sequence. It contains a right moving portrait and a mirrored left moving muesli package. We use the eigenvalue analysis to determine the multiple motion candidates and apply the EM algorithm on the spectral signatures for motion estimation. Note that the spatial estimation algorithms cannot treat transparency sequences. The optical flow in the spectral EM approach is sparse. It is due to the fact that in some regions of the package we do not have adequate texture information. For a robust performance we ignore these regions in estimation after the eigenvalue analysis [18].

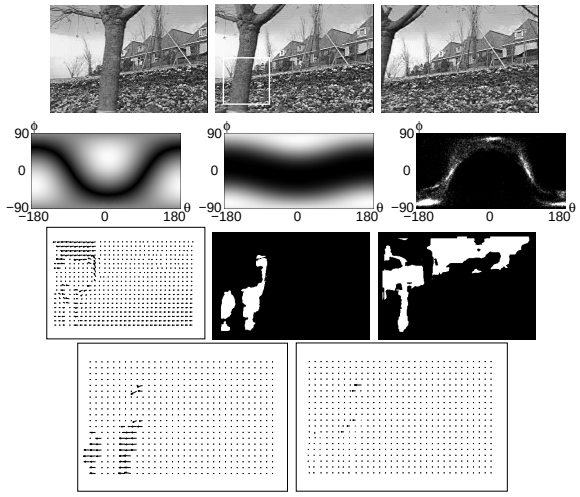

Figure 9: **Row 1:** The 17-th, 32-th and 48-th frames of the "flower garden" sequence. **Row 2:** The orientation signatures applying G_1 (left), B_3 (middle), and our *conic filter* (right) in the white window. **Row 3 Left:** Estimation results using the single motion model. **Row 3 Middle:** Two motion candidate regions. **Row 3 Right:** Regions with the aperture problem. **Row 4:** Optical flow of the occluding (left) and of the occluded signal (right) using the EM algorithm on our spatial orientation signatures.

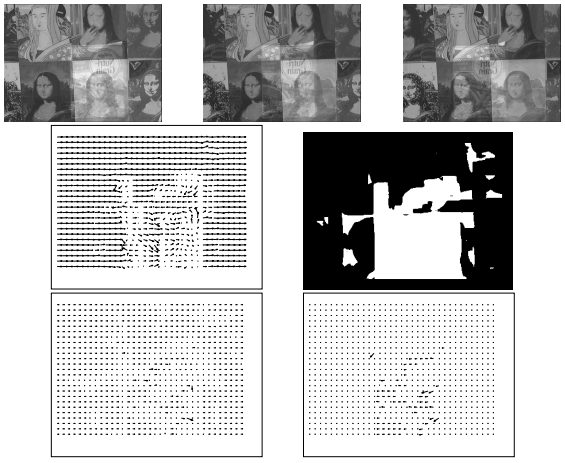

Figure 10: **Row 1:** The first, 16-th and 32-th frames of the image sequence. Each frame has 288×384 pixels. **Row 2 Left:** Estimation results using the single motion model in the 16-th frame. **Row 2 Right:** Marked two motion candidate regions according to the eigenvalue analysis. **Row 3:** Optical flow of two signals using the EM algorithm on our spectral orientation signatures.

5 Conclusion

In this paper we studied the recovery of multiple motions from the standpoint of orientation analysis. We proposed a new 3D *conic kernel* for motion estimation. This method is

superior to current 3D steerability approaches in achieving higher orientation resolution with lower complexity. Comparisons showed that this new method is similar to the 3D Hough transform, but more efficient and robust. Besides, it facilitates the convergence of EM estimation when results are used as EM start values.

References

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America*, 1(2):284–299, 1985.

[2] M. T. Andersson. *Controllable Multidimensional Filters and Models in Low Level Computer Vision*. PhD thesis, Department of Electrical Engineering, Linkoeping University, Linkoeping, Sweden, 1992.

[3] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *Proc. Int. Conf. on Computer Vision*, pages 777–784, Boston, MA, June 20-23, 1995.

[4] S.S. Beauchemin and J.L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27:433–467, 1995.

[5] S.S. Beauchemin and J.L. Barron. The frequency structure of 1d occluding image signals. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:200–206, 2000.

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

[7] D.J. Fleet and K. Langley. Computational analysis of non-Fourier motion. *Vision Research*, 34:3057–3079, 1994.

[8] T.C. Folsom and R.B. Pinter. Primitive features by steering, quadrature, and scale. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(11):1161–1173, 1998.

[9] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[10] N.M. Grzywacz and A.L. Yuille. A model for the estimate of local image velocity by cells in the visual cortex. *Proc. Royal Society of London.*, B 239:129–161, 1990.

[11] H. Gu, Y. Shirai, and M. Asada. MDL-based segmentation and motion modeling in a long image sequence of scene with multiple independently moving objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(1):58–64, 1996.

[12] H. Haußecker and H. Spies. Motion. In B. Jähne, H. Haußecker, and P. Geißer, editors, *Handbook of Computer Vision and Applications*, volume 2, chapter 13, pages 309–396. Academic Press, 1999.

[13] D. J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1(4):279–302, 1987.

[14] F. Heitger, L. Rosenthaler, R. Von der Heydt, E. Peterhans, and O. Kuebler. Simulation of neural contour mechanisms: from simple to end-stopped cells. *Vision Research*, 32(5):963–981, 1992.

[15] B. K. P. Horn. *Robot Vision*. MIT Press, 1986.

[16] P.V.C. Hough. A method and means for recognising complex patterns. U.S. Patent 3,069,654, 1962.

[17] C.L. Huang and Y.T. Chen. Motion estimation method using a 3d steerable filter. *Image and Vision Computing*, 13:21–32, 1995.

[18] B. Jähne. *Spatio-Temporal Image Processing*. Springer-Verlag, 1993.

[19] M. Michaelis and G. Sommer. Junction classification by multiple orientation detection. In *Proc. Third European Conference on Computer Vision*, volume I, pages 101–108, Stockholm, Sweden, May 2-6, J.O. Eklundh (Ed.), Springer LNCS 800, 1994.

[20] P. Perona. Deformable kernels for early vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(5):488–499, 1995.

[21] T. Poggio and F. Girosi. Networks for approximatation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[22] M. Shizawa and T. Iso. Direct representation and detection of multi-scale, multi-orientation fields using local differentiation filters. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 508–514, New York, NY, June 15-17, 1993.

[23] M. Shizawa and K. Mase. A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 289–295, Maui, Hawaii, June 3-6, 1991.

[24] E. P. Simoncelli and H. Farid. Steerable wedge filters for local orientation analysis. *IEEE Trans. Image Processing*, 5(9):1377–1382, 1996.

[25] E. P. Simoncelli and D. J. Heeger. A Model of Neuronal Responses in Visual Area MT. *Vision Research*, 38(5):743–761, 1998.

[26] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 520–526, Puerto Rico, June 17-19, 1997.

[27] Y. Xiong and S. A. Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo and optical flow. *International Journal of Computer Vision*, 24(1):25–59, 1997.

[28] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5):331–338, 1990.

[29] W. Yu, K. Daniilidis, and G. Sommer. Approximate orientation steerability based on angular Gaussians. *IEEE Trans. Image Processing*, 10(2):193–205, 2001.