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

Weichuan Yu\textsuperscript{a,}\textsuperscript{*}

\textsuperscript{a}Department of Diagnostic Radiology, Yale University
BML 332, P.O.Box 208042, New Haven, CT 06520-8042, USA
Tel: +1 203 785-7294, Fax: +1 203 737-4273

Gerald Sommer\textsuperscript{b}

\textsuperscript{b}Institute of Computer Science, Christian Albrechts University
Preusserstrasse 1-9, D-24105 Kiel, Germany

Kostas Daniilidis\textsuperscript{c}

\textsuperscript{c}GRASP Lab, University of Pennsylvania
3401 Walnut Street, Suite 336C, Philadelphia, PA 19104-6228, USA

Abstract

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.

Key words: Conic Filter, 3D Orientation Signature, Multiple Motion Estimation, Hough Transform, EM Algorithm, Steerable Filter

* Corresponding author.

Email addresses: weichuan@noodle.med.yale.edu (Weichuan Yu),

Preprint submitted to Image and Vision Computing 18th June 2001
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 spatiotemporal 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 particular 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 orientational 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 orientational 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 orientational resolution is obtained. Because of the tiny support of filters the way how these filters decompose the sphere is of practically minor importance.

gs@ks.informatik.uni-kiel.de (Gerald Sommer), kostas@grasp.cis.upenn.edu (Kostas Daniilidis).
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). The experiments with both occlusion and transparency sequences are shown in section 4. Finally, this paper is concluded in section 5.

2 Conic Filter

2.1 Definition

There are several ways to represent 3D orientation such as cylinder coordinates and spherical coordinates. All these representations are equivalent and can be converted from each other if required. But the number and the form of orientation variables in different representations are different. For example, in the cylinder coordinates the 3D orientation is described by a 2D polar angle and an implicit angle between the polar radius and the z-coordinate. Another example is the directional angles (i.e. three angles between a vector and three coordinates axes) which are used in [9]. For orientation analysis we believe that the orientation variables should be explicit and the number of variables should be as small as possible to alleviate the complexity of indexing and visualization. Thus, we choose the spherical coordinates in which only two angles (longitude and latitude) are needed to represent 3D orientation.

We compute a local spherical mapping on the input data: \( I(x, y, z) \rightarrow I(r, \theta, \phi) \), where \( r = \sqrt{x^2 + y^2 + z^2} \), \( \theta = \arctan\left(\frac{y}{x}\right) \), \( \phi = \arctan\left(\frac{z}{\sqrt{x^2 + y^2}}\right) \) (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 \((\theta_i, \phi_j)\) reads

\[
K(\theta, \phi) := G_0^{(\theta, \phi)}(\theta, \phi),
\]

where \( N_{r_{\text{min}}, r_{\text{max}}}^\theta (r) \) is a compensation function along the radial direction described later. The angular part of the kernel is a 2D Gaussian function in the \((\theta, \phi)\)-space:

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

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

\[
\mathcal{D}(\theta, \theta_i) := \min(|\theta - \theta_i|, |\theta - \theta_i - 360^\circ|, |\theta - \theta_i + 360^\circ|).
\]

Theoretically, a Gaussian function is not compactly supported. To form an FIR filter we cut off the angular part of \( G_0^{(\theta, \phi)}(\theta, \phi) \) at the boundary of a circular mask with a fixed diameter
Figure 1. A conic kernel centered at \((\theta_i, \phi_j)\) with radial boundaries \(R_{\text{min}}\) and \(R_{\text{max}}\). **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 \(\theta\), \(\phi\) and \(r\) as coordinates. The conic kernel turns into a cylinder with a diameter \(D\). In the \((\theta, \phi)\) plane the circular mask of the cylinder is weighted by a 2D Gaussian function, as shown above the cylinder. \(D\). The diameter \(D\) is usually a function of \(\sigma\) 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 \((\theta_i, \phi_j)\)

\[
A(\theta_i, \phi_j) := \sum_{\{(\theta, \phi)\} \sqrt{(\theta - \theta_i)^2 + (\phi - \phi_j)^2 \leq \sigma^2}} G^{\theta_i, \phi_j}(\theta, \phi) \sum_{r = R_{\text{min}}}^{R_{\text{max}}} \frac{I(r, \theta, \phi)}{N^{\theta_i, \phi_j}}.
\]  

(3)

Now let us consider the sampling of the \((\theta, \phi)\) plane using a set of conic kernels. A sphere surface forms a rectangular region in the \((\theta, \phi)\) plane, which is periodic along the \(\theta\) direction and is mirror-symmetric about the boundary along the \(\phi\) direction. We let neighboring kernels overlap in order to cover the entire rectangular region and use the periodicity along the \(\theta\) direction and mirror-symmetry along the \(\phi\) direction to solve the boundary problem. The number of required conic kernels in sampling the entire rectangular region is determined by the scale parameter \(\sigma\) (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 \((\theta, \phi)\) plane has a range of \(-180^\circ \leq \theta < 180^\circ\) and \(-90^\circ \leq \phi \leq 90^\circ\), by using \(\sigma = 1^\circ\) we need totally \(360 \times 181 = 65160\) conic kernels to sample the \((\theta, \phi)\) plane with a resolution of \(1^\circ\). All conic kernels have very narrow angular support keeping thus complexity in a moderate level.

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.

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 re-
Figure 2. One example of sampling $(\theta, \phi)$ plane with a set of conic kernels. The horizontal or vertical distance between two neighboring masks is equal to the radius of one mask.

After applying the conic kernels, the entire set of samples $A(\theta_j, \phi_j)$ form a discrete orientation signature $A(\theta, \phi)$ in the orientation space. To obtain a continuous orientation signature $S(\theta, \phi)$ from the discrete one we use 2D Gaussian functions with local support $G_0^{(\theta, \phi)}(\theta, \phi)$ as interpolation functions:

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

(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

$$G_n = \frac{\partial^n}{\partial x^n} \exp\{-(x^2 + y^2 + z^2)/2\}$$

with $n \in N$ denoting the order of derivative. For clarity we omit normalization constants. 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.
Similar to the 2D filter design technique used by Simoncelli and Farid [24] we could choose eligible components of spherical harmonics to construct 3D steerable filters with arbitrarily narrow angular support. However, a large number of basis filters is needed to guarantee high orientational resolution. One might think that higher order derivatives would increase orientational resolution. This can hardly be achieved because Gaussian derivatives are fixed combinations of spherical harmonics and we cannot change these combinations to adjust the angular supports of the filters — the reader may plot the angular supports of Gaussian derivatives for an illustrative proof. Based on this reason we take the first derivative of 3D Gaussian $G_1$ for orientation comparison because the number of required basis filters is minimal and we can make a fair comparison.

Andersson [2] designed an alternative 3D steerable filter directly in the frequency domain. He designed the spectral basis filters as

$$B_{l_i}(\bar{\omega}) = G(\rho)(\hat{n}_{l_i} \cdot \hat{u})^l,$$

where $\bar{\omega}$ and $\hat{u}$ are an arbitrary frequency coordinate vector and its corresponding normalized unit vector, respectively. The vector $\hat{n}_{l_i}$ denotes the orientation of the $i$-th basis filter of order $l$, and $G(\rho)$ represents the radial frequency response.

While Andersson succeeded in improving the orientation resolution by using higher order filters, this improvement is very limited. After studying the regular polyhedra in detail, Andersson held that it is impossible to distribute more than ten basis filters evenly on the sphere surface [2]. Consequently, basis filters with order $l \geq 4$ cannot span evenly on the sphere surface, as the number of basis filters is equal to $(l + 1)(l + 2)/2$.

In figure 3 we show the first Gaussian derivative $G_1$, Andersson’s third order filter $B_3$ [2] ($B_3$ has the finest angular support among all Andersson’s filters), and our conic filter, respectively. In the bottom row we show their angular supports in the $(\theta, \phi)$ 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 $(\theta, \phi)$ 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 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 higher orientation resolution which enables us to analyze multiple orientations more precisely.

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,
Figure 3. **Top:** The rendering image of filter kernels. The filter $G_1$ (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 above filters are shown with white regions in the $(\theta, \phi)$ space. Those supports are measurements of the orientation resolution. For clarity we enlarge the angular support of our filter in an extra image.

- 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 ($\theta_i \leq \theta \leq \theta_{i+1}, \phi_j \leq \phi \leq \phi_{j+1}$) 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 orientational 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 orientational resolution.

Another possibility to achieve such a fine orientation resolution with a global decomposition method is to generalize the filter design technique in [24] from 2D to 3D space and use flexible combinations of pre-chosen spherical harmonic components to form a 3D filter with narrow angular shape. However, we would also have to face the considerably higher computation effort in order to build a 3D filter mask with narrow angular shape (see [29] for a similar argument in 2D space).
The proposed conic filtering is also related to 3D orientation histogram [15] 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]. Besides, the 3D histogram is applied for 3D surface analysis. If the object is convex, the corresponding 3D orientation histogram is shift- and scale-invariant. In contrast, our 3D filter is applied not only for surface analysis, but also for volume data analysis. It lends itself both to convex and to concave objects. But we must fix the keypoint and the radial boundaries at first.

2.3 Compensation via Radial Variation

In this section, we address the design of the weighting function \( N(r) \) (cf. equation (1)). In the spherical coordinates the azimuth angle \( \theta \) and the polar angle \( \phi \) are defined differently. All points with the same \( \theta \) on a sphere surface lie on a great circle of this sphere, whereas all points with the same \( \phi \) (except \( \phi = 0^\circ \)) lie on a small circle. If we divide the whole \((\theta, \phi)\) 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 \( N(r) \) 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 \( N(r) = 1 \).

2.4 Conic Kernel Response to 3D-Planes

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

\[
x_1 + y_2 + z_3 = 0. 
\]

(5)

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

\[
\cos(\phi) \cos(\phi_n) \cos(\theta - \theta_n) + \sin(\phi) \sin(\phi_n) = 0. 
\]

(6)

For horizontal and vertical planes with normal vectors parallel to the coordinate axes, their corresponding representations in the \((\theta, \phi)\) space are straight lines. In motion analysis we usually encounter tilted planes which, in the \((\theta, \phi)\) space, turn into harmonic curves with different amplitudes and phases (cf. figure 4). The normal vector of each plane (i.e. \((\theta_n, \phi_n)\)) is determined by the extreme point \((\theta_m, \phi_m)\) on the corresponding curve as follows (see
Appendix for derivation):

\[
\begin{align*}
\theta_n &= \theta_m + 180^\circ \\
\phi_n &= 90^\circ - \phi_m
\end{align*}
\]  \hfill (7)

The \( \theta_n \) and \( \phi_n \) are then used in estimation of motion parameters \((u, v)\)

\[
\begin{align*}
u &= \cos(\theta_n) \cot(\phi_n) \\
v &= \sin(\theta_n) \cot(\phi_n)
\end{align*}
\]  \hfill (8)

Besides, each curve has two zero-crossing points on the \( \theta \) axis with a distance of 180° and \( \theta_m \) lies exactly in the middle of these two zero-crossing points. 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 \((\theta, \phi)\) space. Extracting the parameters \((\theta_n, \phi_n)\) 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. In the following we will describe the concrete algorithm in detail.

Figure 4. **Left**: A plane with normal vector \((-2, 1, 1)\) (drawing with small circles) and a plane with normal vector \((1, 1, 1)\) (drawing with dots) in the Cartesian coordinates. **Right**: The corresponding curves in the \((\theta, \phi)\) space. See text for details about the extreme point and the distance between two zero-crossing points. Since the components on the plane with normal vector \((1, 1, 1)\) have only positive \(z\) coordinates, we observe only positive \(\phi\) coordinates on the corresponding curve.

3 Multiple Motion Estimation Using Conic Filter

3.1 Algorithm

1. Fix radial parameters \(R_{\min}\) and \(R_{\max}\) as well as the angular parameter \(\sigma\) which determines \(D, \delta \theta,\) and \(\delta \phi\) automatically. Also fix another threshold parameter \(\eta\) \((\eta = 2^\circ)\).

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 $(\theta_i, \phi_j)$ on the local derivative space or the
   local energy spectrum by using the LUT (cf. $A(\theta_i, \phi_j)$ in equation (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 $\theta$ axis (i.e. $-\eta \leq \phi \leq \eta$) into the same group if their
distance is less than $2\eta$.

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 $\phi$ 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 $(\theta_m, \phi_m)$ and consequently an initialization of $(\theta_n, \phi_n)$
(cf. equation (7)).

7. Apply equation (6) based EM to get final $(\theta_n, \phi_n)$ for motion estimation (cf. equation (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 $(I_{ix}, I_{iy}, I_d)(i = 1, \cdots, N)$ 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

\[
\begin{align*}
\begin{cases}
u_m = \frac{n_{1m}}{n_{3m}}, \\
v_m = \frac{n_{2m}}{n_{3m}}
\end{cases}
\end{align*}
\]

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 $\varepsilon$ yielding

\[
|I_{ix}u_m + I_{iy}v_m + I_d| \leq \varepsilon.
\]

(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 $\phi$ 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. A: General projection drawing of the filter mask. The vector $n$ is normal to the filter mask. B: Side view of the filter mask. The angular thickness $T$ of the disk is determined by the clustering threshold $\epsilon$ in equation (10). C: Vertical view of the filter mask.

Since the intersections of different curves in the Hough image are blurred due to the introduction of $\epsilon$, the global maximal position is no more a peak, but a smooth unimodal 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 unimodal 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], our 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 (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 of applying Gaussian derivative $G_1$ and Andersson’s third order filter clearly fail to provide the correct structure of multiple planes.

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 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.

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 $B_3$ fail to characterize multiple orientations. Our filter provides a reasonable signature. Its two extreme points lie near $(0^\circ, 45^\circ)$ and $(180^\circ, 45^\circ)$ and
<table>
<thead>
<tr>
<th>model</th>
<th>initial values</th>
<th>compensation</th>
<th>iteration</th>
<th>occluding</th>
<th>occluded</th>
</tr>
</thead>
<tbody>
<tr>
<td>spatial</td>
<td>arbitrarily</td>
<td>yes</td>
<td>3</td>
<td>(0.927, 0.998)</td>
<td>(0.949, -0.971)</td>
</tr>
<tr>
<td></td>
<td>set</td>
<td>no</td>
<td>3</td>
<td>(0.986, 0.999)</td>
<td>(0.986, -0.988)</td>
</tr>
<tr>
<td>spectral</td>
<td>properly</td>
<td>yes</td>
<td>1</td>
<td>(0.938, 1.005)</td>
<td>(0.923, -0.960)</td>
</tr>
<tr>
<td></td>
<td>set</td>
<td>no</td>
<td>1</td>
<td>(0.980, 0.997)</td>
<td>(0.963, -0.974)</td>
</tr>
<tr>
<td>model</td>
<td>arbitrarily</td>
<td>yes</td>
<td>7</td>
<td>(1.187, 1.194)</td>
<td>(1.112, -1.147)</td>
</tr>
<tr>
<td></td>
<td>set</td>
<td>no</td>
<td>4</td>
<td>(0.898, 0.948)</td>
<td>(1.106, -1.099)</td>
</tr>
<tr>
<td></td>
<td>properly</td>
<td>yes</td>
<td>2</td>
<td>(1.182, 1.191)</td>
<td>(1.110, -1.145)</td>
</tr>
<tr>
<td></td>
<td>set</td>
<td>no</td>
<td>2</td>
<td>(0.966, 1.002)</td>
<td>(1.007, -1.026)</td>
</tr>
</tbody>
</table>

Table 1
Estimation results of the occlusion sequence shown in figure 7. We use $(u_{10}, v_{10}) = (0.8, 0.3)$ and $(u_{20}, v_{20}) = (1.2, -0.1)$ to simulate arbitrarily set initial values. 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 $\sigma_r = 0.1$.

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 indicate 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 equation (8) based EM algorithm to estimate motions in the spatial domain (row 4).
Figure 9. **Row 1:** The 17-th, 32-th and 48-th frames of the flower garden sequence. Each frame has $240 \times 352$ pixels. Here we consider the 32-th frame as the central frame. **Row 2:** The amplitudes of orientation signatures applying $G_1$ (left), $B_3$ (middle), and our conic filter (right). **Row 3 Left:** Estimation results using the single motion model. At motion boundaries the results are not correct. **Row 3 Middle:** Two motion candidate regions according to the eigenvalue analysis. **Row 3 Right:** Regions with the aperture problem. **Row 4:** Optical flow of the occluding signal (left) and of the occluded signal (right) using the EM algorithm on our spatial orientation signatures.

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 10. **Row 1**: The first, 16-th and 32-th frames of the image sequence. Each frame has $288 \times 384$ 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.

**Appendix: the relation between** $(\theta_n, \phi_n)$ **and** $(\theta_m, \phi_m)$

In figure 11 we represent all possible unit vectors on the 3D plane with a circle. The normal vector $\mathbf{n}$ is perpendicular to all vectors on this plane, including the vector $\mathbf{m}_1$ (pointing to
the extreme coordinates \((\theta_m, \phi_m)\) and \(\mathbf{m}_2\) (pointing to the point \((\theta_m - 90^\circ, 0)\)). As \(\mathbf{m}_2\) is also perpendicular to \(\mathbf{m}_1\), \(\mathbf{m}_2\) is then the normal vector of the plane containing \(\mathbf{n}\) and \(\mathbf{m}_1\). Since \(\mathbf{m}_2\) lies in the horizontal XY plane, the dotted plane containing \(\mathbf{n}\) and \(\mathbf{m}_1\) is then perpendicular to the XY plane. In this vertical plane we have

\[
\phi_n + 90^\circ + \phi_m = 180^\circ. \tag{11}
\]

This vertical plane always divides the circle equally as it passes through the origin. Taking into account that angles in the \(\theta\) direction are periodic we have

\[
|\theta_n - \theta_m| = 180^\circ.
\]

Without affecting the calculation of the velocity we simply take

\[
\theta_n - \theta_m = 180^\circ. \tag{12}
\]

Then we obtain equation (7).

**Acknowledgment**

The financial support of the first author by German Academic Exchange Service (DAAD) and of the second author by Deutsche Forschungsgemeinschaft (DFG) grant 320/1-3 is gratefully acknowledged. The third author appreciates the financial support by NSF CDS-97-03220, NSF IIS-0083209, ARO/MURI DAAH04-96-1-0007, Advanced Network and Services, and Penn Research Foundation. We thank G. Birkelbach and H. Farid for their helpful suggestions and discussions.
References


