ELSEVIER

Image and Vision Computing 21 (2003) 447–458

image AND vision COMPUTING

www.elsevier.com/locate/imavis

Three dimensional orientation signatures with conic kernel filtering for multiple motion analysis

Weichuan Yu[a,*], Gerald Sommer[b], Kostas Daniilidis[c]

[a]*Department of Diagnostic Radiology, Yale University, BML 332, P.O. Box 208042, New Haven, CT 06520-8042, USA*
[b]*Institute of Computer Science, Christian Albrechts University, Preusserstrasse 1-9, D-24105 Kiel, Germany*
[c]*GRASP Lab, University of Pennsylvania, 3401 Walnut Street, Suite 336C, Philadelphia, PA 19104-6228, USA*

Received 18 June 2001; received in revised form 20 January 2003; accepted 23 January 2003

Abstract

We propose a new 3D kernel for the recovery of 3D orientation signatures. In the Cartesian coordinates, the kernel has a shape of a truncated cone with its axis in the radial direction and very small angular support. In the local spherical coordinates, the angular part of the kernel is a 2D Gaussian function. A set of such kernels is obtained by uniformly sampling the 2D space of azimuth and elevation angles. The projection of a local neighborhood on such a kernel set produces a local 3D orientation signature. In case of spatio-temporal 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. The kernel's local support enables the resulting spatio-temporal signatures to possess higher orientation resolution than 3D steerable filters. Consequently, 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 expectation–maximization based multiple motion detection.

Keywords: Conic kernel; 3D orientation signature; Multiple motion estimation; Hough transform; Expectation–maximization algorithm; Steerable filter

1. Introduction

The motivation of our approach is the local detection and estimation of multiple motions in spatio-temporal 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 spatio-temporal orientation aspect. Adelson and Bergen [1] first pointed out that motion is equivalent to spatio-temporal orientation. They introduced a spatio-temporal energy model for single motion representation. Knutsson proposed to use a 3D structure tensor for orientation recovery and this approach was followed by Bigün [6], Jähne [18,19], and others. To describe multiple motions, Shizawa and co-workers [24,25] 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. Briefly, transparency can be described as two planes of energy concentration in the spectral domain only, while occlusion produces two planes both in the spectral domain and in the spatio-temporal domain accompanied by distortion [5]. The corresponding motion parameters are determined by the normal vectors of these planes. However, determining the precise orientation of two motion planes remains a difficult task, particularly when the angle between two motion planes is small and the energy concentrates at the low frequencies.

Many authors proposed spectral sampling with Gabor or similar filters [10,13,14,29] 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

* Corresponding author. Tel.: +1-203-785-7294; fax: +1-203-737-4273.
E-mail addresses: weichuan@noodle.med.yale.edu (W. Yu), gs@ks.informatik.uni-kiel.de (G. Sommer), kostas@grasp.cis.upenn.edu (K. Daniilidis).

0262-8856/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0262-8856(03)00012-X

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 [8,20,21,26]. Nevertheless, only few approaches dealt with 3D steerability [2,9,27]. These approaches either steer derivatives of Gaussians [9,27] or construct the steerable filter directly in the spectral domain [2]. To achieve high orientation resolution, we need a huge number of basis functions with their angular support covering the entire sphere of orientation. Since detection of multiple motions presumes a high orientation resolution either in the spatio-temporal or in the frequency space, current steerability approaches proved to be impractical.

This motivated us to construct a new 3D kernel with tiny angular support to recover 3D orientation signature. This radial-angular-separable kernel has a conic profile in the 3D Cartesian coordinates and its angular part is a 2D Gaussian with tiny support. In applications, we project the local spatio-temporal imagery onto a huge number of such kernels to form a signature with high orientation resolution.

The rest of the paper is organized as follows: Section 2 describes the details of the new 3D kernel and compares it with current 3D steerable filters. The filter response to 3D planes is also discussed. In Section 3, we explain the algorithm of obtaining 3D orientation signatures in the local derivative space or in the local Fourier domain. In the same section, we also compare this kernel projection to the Hough transform and the expectation–maximization (EM) algorithm in multiple plane estimation. After that, the experiments with both occlusion and transparency sequences are shown in Section 4. Finally, we conclude this paper in Section 5.

2. Conic kernel

2.1. Definition

There are several equivalent coordinates to represent 3D orientation. They differ in the number and form of orientation variables. For example, a 2D polar angle and an implicit elevation angle (between the polar radius and the z-coordinate) are used together to describe 3D orientation in the cylindrical coordinates, while three directional angles (i.e. three angles between three coordinates axes and one vector) are used in the Cartesian coordinates [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 (azimuth and elevation) are needed to represent 3D orientation.

The input data for motion analysis can be either the local image derivative space (i.e. a space coordinated by partial derivatives of images with respect to different coordinate axes) or the local Fourier spectrum. They are the same for filtering purpose. For simplicity, we use the same representation $I(x,y,z)$ for both kinds of input data. Here we assume that $I(x,y,z)$ is correctly obtained for every (x,y,z). Thus, the error in obtaining image derivatives or spectrum is not considered. We start orientation analysis by computing a local spherical mapping on the input data: $I(x,y,z) \rightarrow I(r,\theta,\phi)$, where $r=\sqrt{x^2+y^2+z^2}$, $\theta=\arctan(y/x)$, $\phi=\arctan(z/\sqrt{x^2+y^2})$ (Fig. 1). In order to have fine orientation resolution, we use *conic kernels* with small angular support to sample the orientation space locally. These kernels are radial-angular-separable. A conic kernel centered at (θ_i,ϕ_j) reads

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

where $\mathcal{N}_{R_{\min},R_{\max}}^{(\theta_i,\phi_j)}(r)$ is a compensation function along the radial direction, which we will describe in Section 2.2. First we focus on the angular part of the kernel, which is a 2D Gaussian function in the (θ,ϕ)-space:

$$G_0^{(\theta_i,\phi_j)}(\theta,\phi)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{(\mathcal{M}(\theta,\theta_i))^2+(\phi-\phi_j)^2}{2\sigma^2}\right). \tag{2}$$

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

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

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

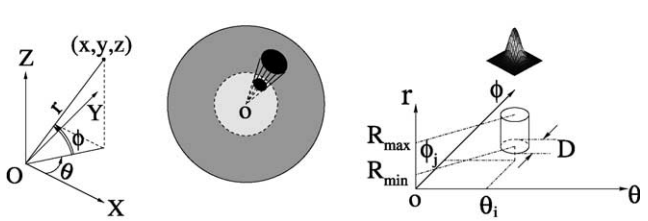

Fig. 1. A conic kernel centered at (θ_i,ϕ_j) with radial boundaries $R_{\min}$ and $R_{\max}$. *Left*: The definition of the spherical coordinate system. *Middle*: The conic kernel in the 3D Cartesian coordinate system. The keypoint is at the center of the sphere. *Right*: The conic kernel with θ, ϕ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

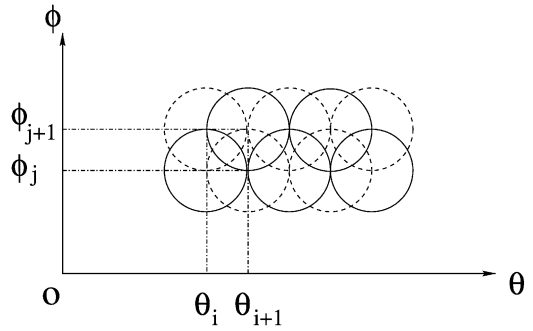

Fig. 2. One example of sampling (θ, ϕ)-plane with a set of conic kernels. The horizontal and vertical distance between two neighboring masks are equal to the mask radius.

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

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

It is easy to find out that a spherical surface containing all possible 3D orientation becomes a rectangular region in the (θ, ϕ)-plane with a range of $-180° \le \theta < 180°$ and $-90° \le \phi \le 90°$, which is periodic along the θ direction and is mirror-symmetric about the boundary along the ϕ direction. Now let us consider the sampling of the (θ, ϕ)-space using a set of conic kernels. In order to cover the entire rectangular region, we overlap neighboring kernels (Fig. 2). We also 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. Eq. (2)) and the sampling interval (i.e. the distance between two neighboring sampling masks, cf. Fig. 2). In this paper, we set the horizontal- and vertical-sampling interval as $\delta\theta = \delta\phi = 3\sigma$. By setting $\sigma = 1/3°$, we need totally $360 \times 181 = 65{,}160$ conic kernels to sample the rectangular (θ, ϕ)-region. In Section 2.3 we will show that the complexity of this sampling is still at a moderate level comparing to current steerable filter approaches.

The proposed decomposition of the spherical surface is not uniform since the azimuth angle θ and the elevation angle ϕ are defined differently in the spherical coordinates. All points on a spherical surface with the same θ 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 entire (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 spherical surface [15]. As a result, it produces a rotation-variant signature. Fortunately, this variation is very small due to the huge number of kernels and their tiny support. Practically, the proposed decomposition with 65,160 kernels arranged uniformly in the (θ, ϕ)-space (not uniformly on the spherical surface) has approximately the same effect as sphere tessellation with the same huge number of kernels whose centers lying on a subdivision of the truncated icosahedron. The reason of using our decomposition method mainly lies in the simplicity of indexing and displaying.

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

After applying the conic kernels, the entire set of samples $A_{(\theta_i,\phi_j)}$ forms 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 $G_0^{(\theta_i,\phi_j)}(\theta, \phi)$ as interpolation functions:

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

This constitutes an approximation and not an interpolation of orientation signature. The same method also appears in Radial Basis Functions approaches [22].

2.2. Compensation via radial variation

In this section, we address the design of the radial part of the kernel (i.e. $\mathcal{N}(r)$ in Eq. (1)). In Section 2.1, we have pointed out that there is a distribution variation on the spherical surface. If we like to compensate for this non-uniform distribution, we may construct the term $\mathcal{N}(r)$ as the sum of discrete weights in the conic kernels. This compensation 'strengthens' the outputs of conic kernels with a few points and 'suppresses' those outputs of conic kernels with many points. As a result, we are no more able to know the real point distribution in the (θ, ϕ)-space. However, the distribution information is desired in many applications. For example, in the EM algorithm we purely use statistics to extract parameters from a set of sample points with the belief that there are more *normal* points with similar statistic properties than noise and 'incorrect' sample points with large deviation from the bulk of all data points [23]. The distribution actually works as a weighting factor in the parameter regression procedure. If we lose the distribution information, the estimation result will be much worse. For this reason, we would like to preserve the distribution information by simply setting $\mathcal{N}(r) = 1$.

In Fig. 3, we demonstrate the advantage of keeping distribution information using a range image example. From the 3D plot we observe that the bottom of the cup, the rim

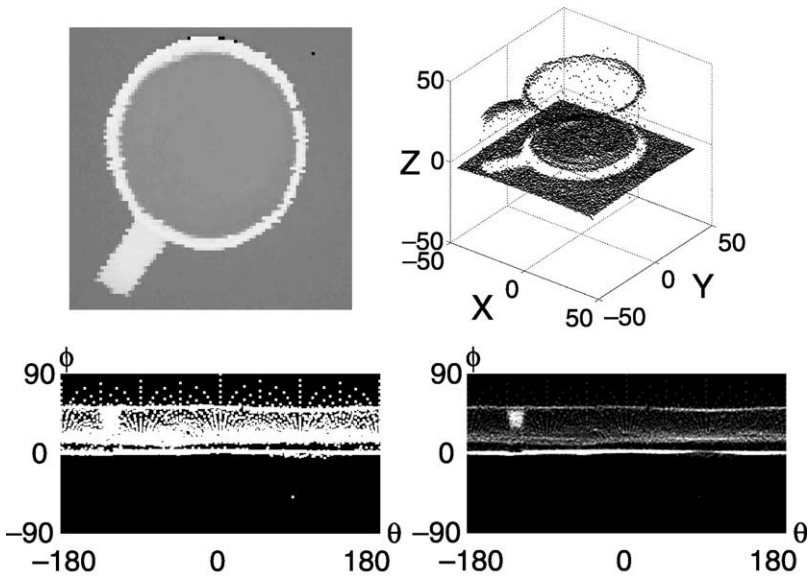

Fig. 3. *Top Left*: Range image of a cup. *Top Right*: 3D plot of the range image. We normalize the z coordinate according to the maximal value of x coordinate. *Bottom Left*: Orientation signature with distribution compensation. We set the center of cup bottom as keypoint. The side of the cup is very much strengthened. We can see the filter response of the handle near $\theta = -120°$, $\phi \in [30°, 50°]$. The white point near $(\theta, \phi) = (80°, -50°)$ is the response of the dark point outside the cup. *Bottom Right*: Orientation signature without distribution compensation. This signature represents the real distribution. We can observe the filter response of the upper part of the handle more clearly. The filter response of the dark point outside the cup is suppressed now.

and the upper part of the handle are well represented in the range image, while there are very few pixels corresponding to the side of the cup. We set the center of the cup bottom as keypoint and normalize the z coordinate according to the maximal value of x coordinate. If we compensate for this non-uniform distribution, the side of the cup is very much 'strengthened' in the corresponding orientation signature (bottom left in Fig. 3). In contrast, if we keep $\mathcal{N}(r) = 1$, the corresponding orientation signature (bottom right in Fig. 3) keeps the distribution information: The bottom and the rim of the cup are much stronger than the side of the cup in the signature. Particularly, the filter response of upper part of the handle is more distinctive near $\theta = -120°$, $\phi \in [30°, 50°]$. Note that the white pixel near $(\theta, \phi) = (80°, -50°)$ (corresponding to the dark point outside the cup in the range image) is also suppressed. This comparison shows that setting $\mathcal{N}(r) = 1$ helps to preserve main structure information and to suppress small disturbance.

2.3. Comparison with current 3D steerable filters

Current 3D steerable filters are rotated copies of either 3D Gaussians [9,27] or specified basis filters in the frequency space [2]. For example, the n-th derivative of 3D Gaussian along the x-axis reads

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

with $n \in N$ denoting the order of derivative. For clarity we omit normalization constants. The angular terms of 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. As an extension of the 2D filter design technique used by Simoncelli and Farid [26], we could choose eligible components of spherical harmonics to construct 3D steerable filters with arbitrarily narrow angular support. 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 Ref. [31] for a similar argument in 2D space). 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. As a result, we cannot change these combinations to adjust the angular support of filters—the reader may plot the angular support of Gaussian derivatives for an illustrative proof. For comparison of orientation resolution and computation complexity, we choose the first derivative of 3D Gaussian G_1 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

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

where $\bar{u}$ 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.

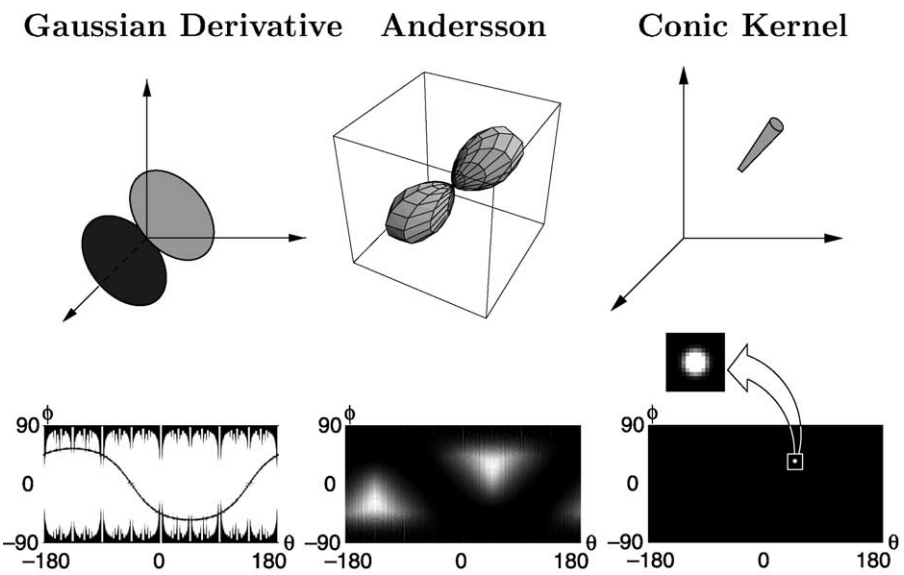

Fig. 4. *Top*: Rendering images of filter kernels. The filter G_1 (left, redrawn from Ref. [17]), B_3 (middle, redrawn from Ref. [2]), and our filter (right) are all centered at $(\theta, \phi) = (45.00°, 35.26°)$. *Bottom*: Corresponding angular support of above filters is shown with white regions in the (θ, ϕ)-space. The smaller the angular support is, the finer the orientation resolution. For clarity we enlarge the angular Gaussian support of our filter in an extra image.

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 10 basis filters evenly on the spherical surface [2]. Consequently, basis filters with order $l \geq 4$ cannot span evenly on the spherical surface since the number of basis filters is equal to $(l + 1)(l + 2)/2$.

In Fig. 4, we show the first derivative of Gaussian 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 kernel, respectively. In the bottom row we also show their angular support 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 G_1 in the (θ, ϕ)-space with $|\phi| > 40°$ 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]. Andersson's filter B_3 has a higher orientation resolution than G_1. But this improvement is limited (cf. Fig. 8 as well). This resolution limitation explains why no steerable filters are applied in the analysis of multiple motions. In contrast, our filter has a much higher orientation resolution, which enables us to analyze multiple orientations 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 voxel in the 3D input data. Concretely,

- The Gaussian derivative G_1 is composed of three basis filters with global support. Each voxel in the input data is therefore involved three times in the scalar product as well as in the interpolation procedure.
- Andersson's B_3 filter has 10 basis filters. Thus, each voxel is involved 10 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. Fig. 2). Roughly speaking, a voxel falling into this area is involved four times in the scalar product. As the interpolation function has the same support as the conic kernel, a voxel is also involved four times in the interpolation.

According to above analysis, the conic kernel is more efficient than Andersson's B_3 but slightly less efficient than G_1. We should be aware that a complexity comparison is fair only when the corresponding filters are comparable in orientation resolution. Obviously, the above comparison does not have this basis and should be therefore only regarded as an illustration of the relative implementation complexity of our conic kernel.

The proposed conic filtering is also related to 3D orientation histogram [15] usually obtained in the gradient space. It differs in the sampling of the orientation space: the orientation histogram follows merely the Hough sampling principle [16], whereas the conical kernels here overlap in the angular space. 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 kernel is applied not only for surface

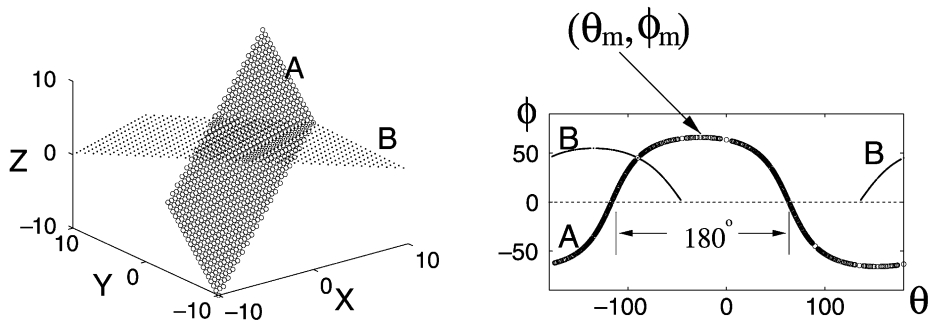

Fig. 5. *Left*: Plane A with normal vector (−2, 1, 1) (drawing with small circles) and plane B with normal vector (1, 1, 1) (drawing with dots) in the Cartesian coordinates. The points on plane B only have positive z coordinates. *Right*: The corresponding curves in the (θ, ϕ)-space. As the points on plane B have only positive z coordinates, curve B only has components with positive ϕ coordinates. See text for details about the extreme point and the distance between two zero-crossing points.

analysis, but also for volume data analysis. It lends itself both to convex and concave objects. Certainly we have to fix the keypoint and the radial boundaries at first.

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 $\mathbf{n} = (n_1, n_2, n_3)^{\mathrm{T}}$ 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 $(x, y, z) \rightarrow (r, \theta, \phi)$ and $(n_1, n_2, n_3) \rightarrow (1, \theta_n, \phi_n)$. After dropping out r we obtain 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 turn into harmonic curves with different amplitudes and phases in the (θ, ϕ)-space (cf. Fig. 5). The normal vector of each plane (i.e. (θ_n, ϕ_n)) is related to the extreme point (θ_m, ϕ_m) on the corresponding curve as follows (see Appendix A for derivation):

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

The motion parameters (u, v) can then be estimated using θ_n and ϕ_n

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

Further, each harmonic curve has two zero-crossing points on the θ axis with a distance of 180° and θ_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 (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. In the following, we will describe the concrete algorithm in detail.

3. Multiple motion estimation using conic kernel

3.1. Algorithm

1. Fix radial parameters $R_{\min}$ and $R_{\max}$ as well as the angular parameter σ ($\sigma = 1/3^\circ$). The parameters D, $\delta\theta$, and $\delta\phi$ are determined by σ. 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 conic kernel 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.

Cluster the non-zero $A_{(\theta_i,\phi_j)}$ near θ axis (i.e. $-\eta \leq \phi \leq \eta$) into the same group if their distance is less than 2η.

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. For each group-pair, search for the non-zero $A_{(\theta_i,\phi_j)}$ along the positive ϕ direction from their middle point and cluster the non-zero $A_{(\theta_i,\phi_j)}$ 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)).

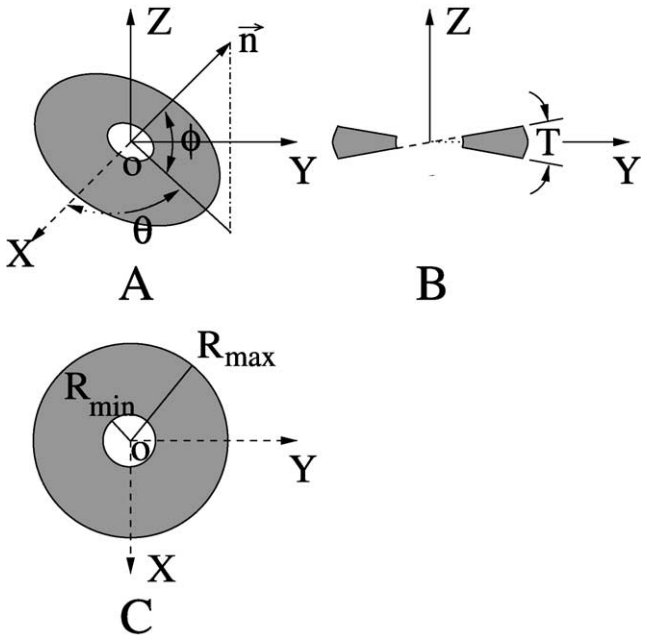

Fig. 6. The 3D Hough transform is equivalent to a filter with a concave disk shape. *A*: A general view of the filter mask. The vector **n** is normal to the filter mask. *B*: Side view of the filter mask. The angular thickness T of the disk is determined by the clustering threshold ε in Eq. (10). *C*: Vertical view of the filter mask.

Apply Eq. (6)-based EM to obtain the final (θ_n, ϕ_n). Then use Eq. (8) for motion estimation.

Since Eq. (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 sampling and searching method for parameter extraction. Concretely, for a set of points coordinated with (x_i, y_i, z_i) $(i = 1, \ldots, N)$ we draw the corresponding vectors in the (n_1, n_2, n_3) space satisfying Eq. (5). Then we search throughout the (n_1, n_2, n_3) space for 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

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

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

$$|x_i u_m + y_i v_m + z_i| \leq \varepsilon. \tag{10}$$

The Eq. (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. Fig. 6). The comparison between our filter shape (Fig. 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 Fig. 7, which is actually the impulse response of the concave disk filter. The Hough image is very similar to our filter response of a 3D plane except that it has no negative ϕ value (the third component of the normal vector is always positive in Eq. (10)). 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 [30], especially the gigantic overlapping of the Hough curves (Fig. 7). As a result, we can extract the parameters of motion planes with much less complexity.

Further, the intersections of different curves in the Hough image are blurred due to the introduction of ε. As a result, the global maximal position is no more a peak, but a smooth *uni-modal* distribution. While searching for the global maximal position is still feasible, searching for the second maximal position is generally problematic. For example, we do not know how to automatically choose the neighborhood of the second maximum, especially when the properties of the uni-modal distribution around the first maximum are unknown. Even if we can choose the neighborhood manually, the second maximum is blurred and its position is *biased* by the distribution around the first maximum. The bias is even

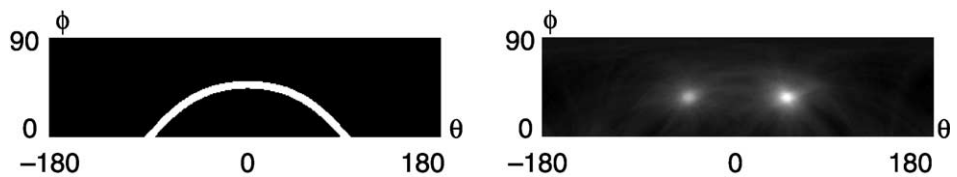

Fig. 7. *Left*: Vectors satisfying Eq. (10) form a curve similar to our filter response to a 3D plane in the (θ, ϕ)-space. The width of the curve is determined by the clustering threshold ε in Eq. (10). *Right*: The Hough image of an occlusion sequence (cf. Fig. 8) with two velocities of (1, 1) [pixel/frame] and (1, −1) [pixel/frame]. The global maximum is blurred due to the introduction of ε. Automatically searching for the second maximum is problematic. See text for details.

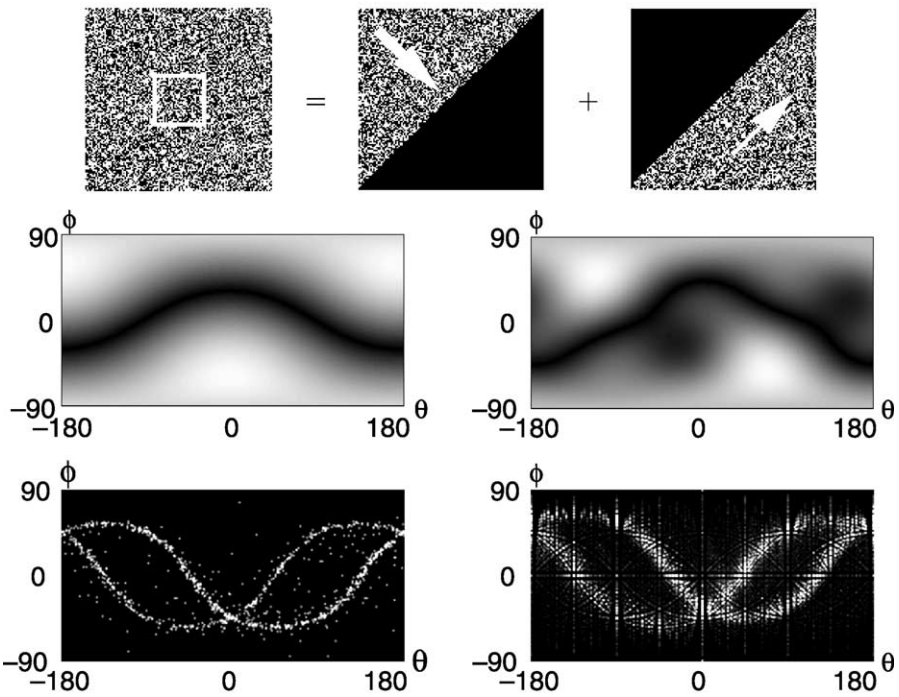

Fig. 8. *Top*: One frame of an occlusion sequence can be separated as overlapping of two motions. The white window indicates the multiple motion region in which we compare orientation signatures of different filters. The white arrows denote the moving directions and the black regions denote static background. *Middle*: The amplitudes of orientation signatures using G_1 (left) and B_3 (right). The structure of multiple planes is hardly to see. *Bottom*: The orientation signatures using our filter in the derivative space (left) and in the spectral domain (right). The curves in the spectral orientation signature are blurred.

worse, when these two maxima locate near each other. Both will result in an inaccurate estimation.

The EM algorithm consists of subsequent iterations of 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 [3,11,28], 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. Experiment

We begin with an artificial occlusion sequence (Fig. 8). The occluding signal has a constant flow of (1, 1) [pixel/frame] and the occluded signal has a flow of (1, −1). We use the Gaussian derivative G_1 with a support of $5 \times 5 \times 5$ pixels inside 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

Table 1
Estimation results of the occlusion sequence shown in Fig. 8. We use $(u_{10}, v_{10}) = (0.8, 0.3)$ and $(u_{20}, v_{20}) = (1.2, -0.1)$ to simulate arbitrarily 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 parameters

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)

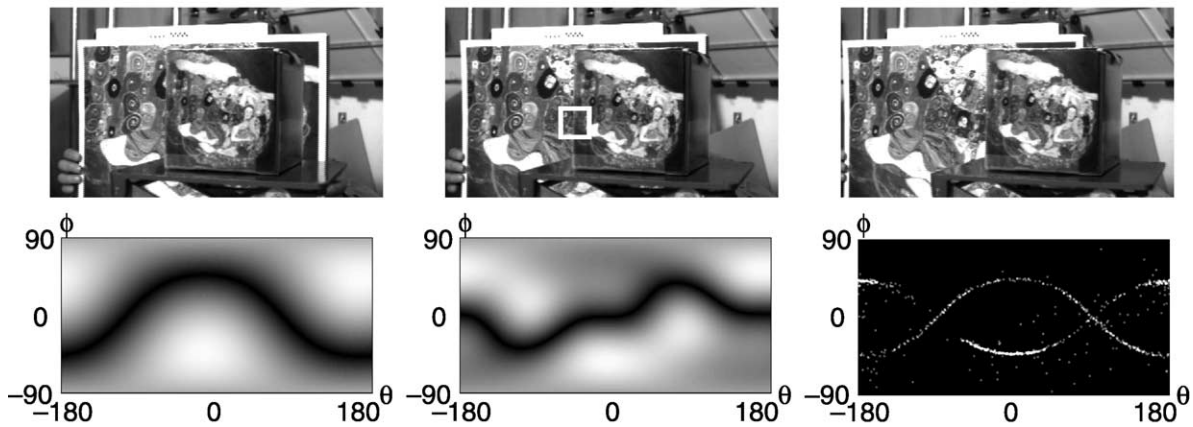

Fig. 9. *Top*: The first, 16-th and 32-th frame of an occlusion sequence. The white window in the 16-th frame indicates an occlusion region. *Bottom*: The amplitudes of orientation signatures applying G_1 (left), B_3 (middle), and our conic kernel (right) in the derivative space of the white window in frame 16.

space because otherwise the spectral resolution will be too coarse. The orientation signatures of applying G_1 and B_3 obviously fail to provide the correct structure of multiple planes. In contrast, the orientation signatures using our conic kernels recover the two planes both in the derivative space and in the spectral space.

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 distribution compensation to confirm the analysis in Section 2.2. In the first test, we set initial values arbitrarily. In the second test, we take proper initialization according to

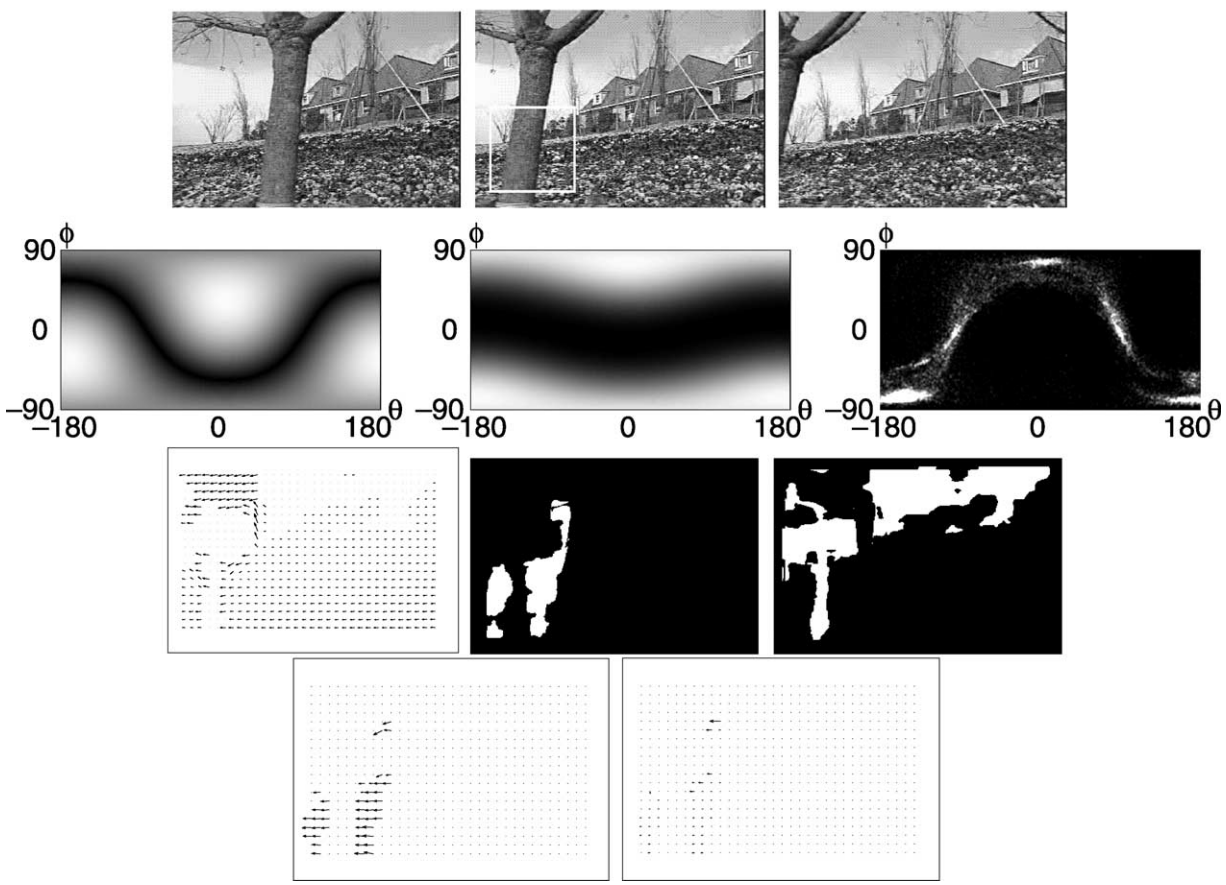

Fig. 10. *Row 1*: The 17-th, 32-th and 48-th frames of the flower garden sequence. Each frame has 240 × 352 pixels. Here we consider the 32-th frame as the central frame. *Row 2*: The amplitudes of orientation signatures applying G_1 (left), B_3 (middle), and our conic kernel (right). *Row 3 Left*: Estimation 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.

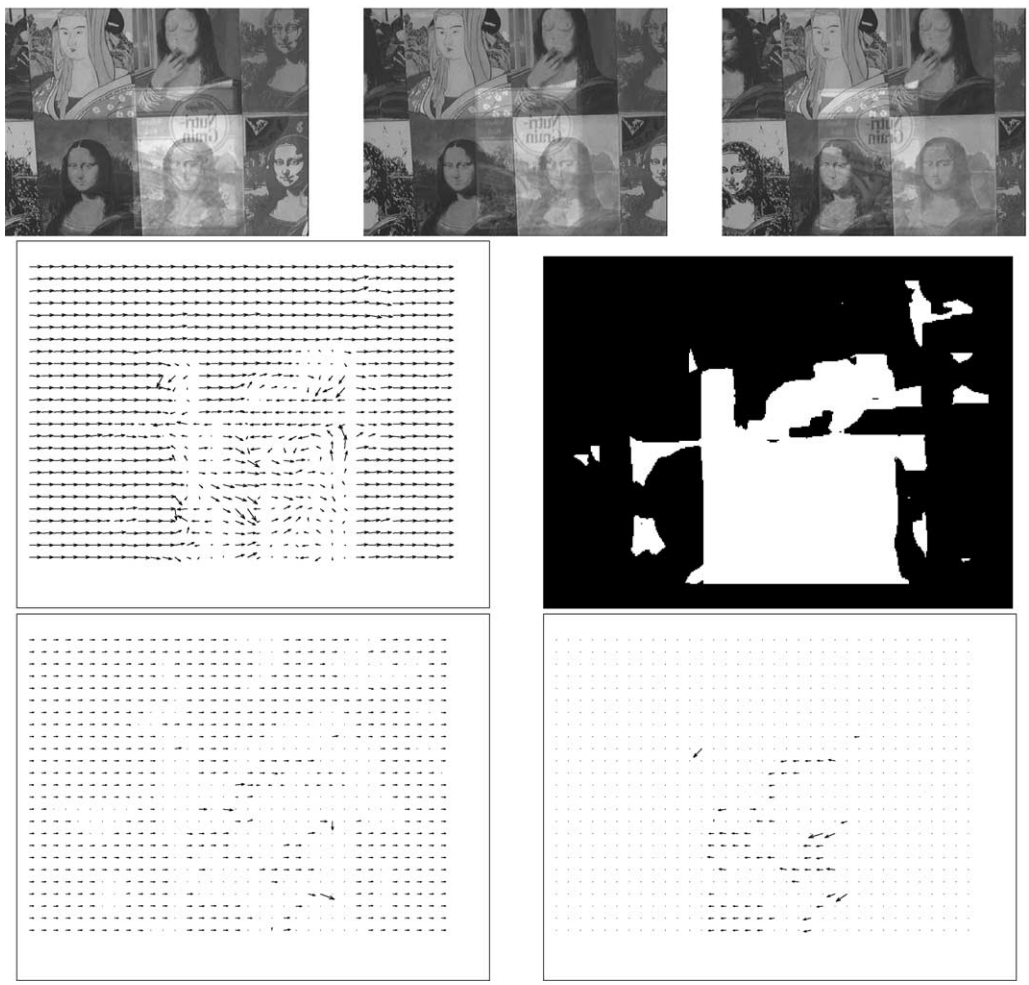

Fig. 11. *Row 1*: The first, 16-th and 32-th frame of the image sequence. Each frame has 288 × 384 pixels. *Row 2 Left*: Estimation results using the single motion model in the 16-th frame. *Row 2 Right*: Marked two motion candidate regions according to the eigenvalue analysis. *Row 3*: Optical flow of two signals using the EM algorithm on our spectral orientation signatures.

the extreme point analysis introduced in Section 2.4. The estimation results without distribution compensation are better than the results with compensation. Also, proper initialization reduces the number of iterations greatly in the EM algorithm. In addition, 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. In the spectral orientation signature, however, the curves 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 desired global minimum. The spectral estimation results are not comparable to the spatial estimation results because the quality of two orientation signatures are not at the same level, as already shown in Fig. 8.

In order to test the performance of the EM algorithm on determining the number of motions, we propose an example of single motion with a velocity $(1, -1)$. Both spatial and spectral EM algorithms should converge to one speed, 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, while the spectral EM algorithm converges to $(1.057, -1.045)$ and $(0.951, -1.011)$ after 2 iterations. This result is not surprising since the curves in the spectral orientation signature are blurred. To confirm the spectral EM algorithm can converge 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 determine the number of motions exactly and properly initial values play a critical role for data with 'bad' quality.

In Fig. 9 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 resolution of different filters. Inside the white window in the 16-th frame, we apply G_1, B_3, and our conic kernel 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°, 45°) and (180°, 45°) and are ideally consistent with the motions.

Fig. 10 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 kernel 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 [18,19] we detect two motion candidate regions and the regions with the aperture problem. Only in the multiple motion candidate regions apply we Eq. (8)-based EM algorithm to estimate motions in the spatial domain (row 4).

In Fig. 11, we demonstrate a real example of transparency sequence. It contains a right moving portrait and a mirrored left moving muesli package. We detect multiple motion candidates using the eigenvalue analysis [19]. Then, we 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. This 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 [18,19].

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. In addition, it facilitates the convergence of the EM algorithm when its results are used as the starting values of the EM estimation.

Acknowledgements

We thank G. Birkelbach and H. Farid for their helpful suggestions and discussions. The financial support provided to W. Yu by German Academic Exchange Service (DAAD) and the US National Institutes of Health grant R01-HL-44803-06 (principal investigator: J.S. Duncan) is gratefully acknowledged. G. Sommer is supported by Deutsche Forschungsgemeinschaft (DFG) grant 320/1-3. K. Daniilidis is grateful for the following support: NSF-IIS-0083209, NSF-EIA-0120565, NSF-IIS-0121293, NSF-EIA-9703220, a DARPA/ITO/NGI subcontract to UNC, and a Penn Research Foundation grant.

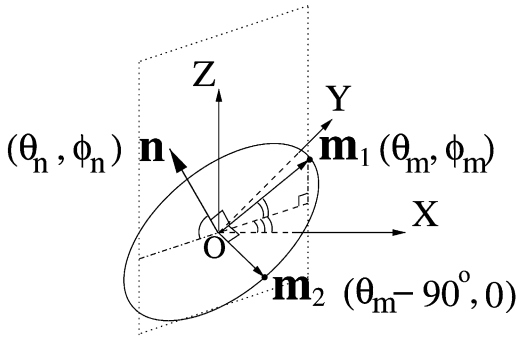

Fig. 12. The relation between (θ_n, ϕ_n) and (θ_m, ϕ_m). The circle contains all possible unit vectors on a 3D plane. The dotted plane containing the normal vectors $\mathbf{n}$ and $\mathbf{m}_1$ is a vertical plane perpendicular to the XY plane.

Appendix A. The relation between (θ_n, ϕ_n) and (θ_m, ϕ_m)

In Fig. 12, 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 point (θ_m, ϕ_m)) and $\mathbf{m}_2$ (pointing to the point $(\theta_m - 90°, 0)$). As $\mathbf{m}_2$ is also perpendicular to $\mathbf{m}_1$, $\mathbf{m}_2$ is then the normal vector of the dotted plane containing $\mathbf{n}$ and $\mathbf{m}_1$. Since $\mathbf{m}_2$ lies in the horizontal XY plane, the dotted plane is then perpendicular to the XY plane. In this vertical plane we have

$$\phi_n + 90° + \phi_m = 180°. \tag{A1}$$

In addition, the vertical plane always divides the circle equally since it passes through the origin. As the angles in the θ direction are periodic, we have

$$|\theta_n - \theta_m| = 180°.$$

Without affecting the estimation of the velocity, we simply take

$$\theta_n - \theta_m = 180°. \tag{A2}$$

Then we obtain Eq. (7).

References

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

[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, H.S. Sawhney, Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding, Proceedings of the International Conference on Computer Vision, Boston, MA, June 20–23, 1995, pp. 777–784.

[4] S.S. Beauchemin, J.L. Barron, The computation of optical flow, ACM Computing Surveys 27 (1995) 433–467.
[5] S.S. Beauchemin, J.L. Barron, The frequency structure of 1d occluding image signals, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 200–206.
[6] J. Bigün, G.H. Granlund, J. Wiklund, Multidimensional orientation estimation with application to texture analysis and optical flow, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (8) (1991) 775–790.
[7] D.J. Fleet, K. Langley, Computational analysis of non-Fourier motion, Vision Research 34 (1994) 3057–3079.
[8] T.C. Folsom, R.B. Pinter, Primitive features by steering, quadrature, and scale, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (11) (1998) 1161–1173.
[9] W.T. Freeman, E.H. Adelson, The design and use of steerable filters, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1991) 891–906.
[10] N.M. Grzywacz, A.L. Yuille, A model for the estimate of local image velocity by cells in the visual cortex, Proceedings of the Royal Society of London, B 239 (1990) 129–161.
[11] H. Gu, Y. Shirai, M. Asada, MDL-based segmentation and motion modeling in a long image sequence of scene with multiple independently moving objects, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1) (1996) 58–64.
[12] H. Haußecker, H. Spies, Motion, in: B. Jähne, H. Haußecker, P. Geißer (Eds.), Handbook of Computer Vision and Applications, vol. 2, Academic Press, New York, 1999, pp. 309–396, Chapter 13.
[13] D.J. Heeger, Optical flow using spatiotemporal filters, International Journal of Computer Vision 1 (4) (1988) 279–302.
[14] F. Heitger, L. Rosenthaler, R. Von der Heydt, E. Peterhans, O. Kuebler, Simulation of neural contour mechanisms: from simple to end-stopped cells, Vision Research 32 (5) (1992) 963–981.
[15] B.K.P. Horn, Robot Vision, MIT Press, Cambridge, MA, 1986.
[16] P.V.C. Hough, A method and means for recognising complex patterns, US Patent 3,069,654, 1962.
[17] C.L. Huang, Y.T. Chen, Motion estimation method using a 3d steerable filter, Image and Vision Computing 13 (1995) 21–32.
[18] B. Jähne, Spatio-Temporal Image Processing, Springer, Berlin, 1993.
[19] B. Jähne, H. Haußecker, H. Scharr, H. Spies, D. Schmundt, U. Schurr, Study of dynamical processes with tensor-based spatiotemporal image processing techniques, in: H. Burkhardt, B. Neumann (Eds.), Proceedings of the Fifth European Conference on Computer Vision, Freiburg, Germany, June 2–6, Springer LNCS 1407, vol. II, 1998, pp. 322–335.
[20] M. Michaelis, G. Sommer, Junction classification by multiple orientation detection, in: J.O. Eklundh (Ed.), Proceedings of the Third European Conference on Computer Vision, Stockholm, Sweden, May 2–6, Springer LNCS 800, vol. I, 1994, pp. 101–108.
[21] P. Perona, Deformable kernels for early vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (5) (1995) 488–499.
[22] T. Poggio, F. Girosi, Networks for approximation and learning, Proceedings of the IEEE 78 (9) (1990) 1481–1497.
[23] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, Cambridge University Press, Cambridge, 1992.
[24] M. Shizawa, T. Iso, Direct Representation and Detection of Multi-Scale, Multi-Orientation Fields Using Local Differentiation Filters, IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, June 15–17, 1993, pp. 508–514.
[25] M. Shizawa, K. Mase, A Unified Computational Theory for Motion Transparency and Motion Boundaries Based on Eigenenergy Analysis, IEEE Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, June 3–6, 1991, pp. 289–295.
[26] E.P. Simoncelli, H. Farid, Steerable wedge filters for local orientation analysis, IEEE Transactions on Image Processing 5 (9) (1996) 1377–1382.
[27] E.P. Simoncelli, D.J. Heeger, A model of neuronal responses in visual area MT, Vision Research 38 (5) (1998) 743–761.
[28] Y. Weiss, Smoothness in Layers: Motion Segmentation Using Nonparametric Mixture Estimation, IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, June 17–19, 1997, pp. 520–526.
[29] Y. Xiong, S.A. Shafer, Moment, and hypergeometric filters for high precision computation of focus, stereo and optical flow, International Journal of Computer Vision 24 (1) (1997) 25–59.
[30] L. Xu, E. Oja, P. Kultanen, A new curve detection method: randomized Hough transform (RHT), Pattern Recognition Letters 11 (5) (1990) 331–338.
[31] W. Yu, K. Daniilidis, G. Sommer, Approximate orientation steerability based on angular Gaussians, IEEE Transactions on Image Processing 10 (2) (2001) 193–205.