INSTITUT FÜR INFORMATIK

UND PRAKTISCHE MATHEMATIK

Multiple Motion Analysis Using 3D Orientation Steerable Filters

Weichuan Yu, Gerald Sommer, Kostas Daniilidis

Bericht Nr. 2008

5. December 2000

CHRISTIAN-ALBRECHTS-UNIVERSITÄT

KIEL

Institut für Informatik und Praktische Mathematik der
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40
D – 24098 Kiel

Multiple Motion Analysis Using 3D Orientation Steerable Filters

Weichuan Yu, Gerald Sommer, Kostas Daniilidis

Bericht Nr. 2008
5. December 2000

e-mail: {wy,gs}@ks. informatik.uni-kiel.de

The address of K. Daniilidis is: GRASP Lab, University of Pennsylvania, Philadelphia, PA 19104-6228, USA.
e-mail: kostas@grip.cis.upenn.edu.

Multiple Motion Analysis Using 3D Orientation Steerable Filters

Weichuan Yu, Gerald Sommer, Kostas Daniilidis

5. December 2000

Abstract

In this paper, we study the characterization of multiple motions from the standpoint of orientation in spatiotemporal volume. Using the fact that multiple motions are equivalent to multiple planes in the derivative space or in the spectral domain, we apply a new 3D steerable filter for motion estimation. This new method is based on the decomposition of the sphere with a set of overlapping basis filters in the feature space. It is superior to principal axis analysis based approaches and current 3D steerability approaches in achieving higher orientation resolution. Our approach is more efficient and robust than a similar spatiotemporal Hough transform and outperforms existing EM algorithms applied in the derivative space.

In occlusion estimation, we use an eigenvalue analysis based multi-window strategy to detect and to eliminate outliers in the derivative space. This technique purifies input data and improves therefore the precision of the estimation results. Furthermore, based on the spatial coherence in image sequences we use the "shift-and-subtract" technique to localize occlusion boundaries and to track their movement in occlusion sequences. Our technique can be also used to distinguish occlusion from transparency and to decompose transparency scenes into multi-layers.

1 Introduction

The study of optical flow has a long history. With respect to different criteria the related flow estimation algorithms may be divided into global or local techniques, one frame or multiple frame methods, spatial or spectral domain based approaches, probabilistic or non-probabilistic models, and so on. The reader is referred to [4, 21] for a general survey of flow estimation methods. While the research of single motion estimation is becoming mature, the estimation and analysis of multiple motions are still challenging problems. Generally, we use the behavior in the neighborhood of a considered point due to the aperture problem. Such assumptions are either explicit in area-based techniques or implicit in filter-based schemes where the addressed neighborhood is the filter support. Conventional flow estimation methods are based on the single motion assumption (there is only one single motion inside the neighborhood) and the smoothness assumption (the motion is piecewise-smooth). For example, the well known brightness change constraint equation (BCCE) [24] is based on these assumptions

$$I_x u + I_y v + I_t = 0, \tag{1}$$

where I_x, I_y, and I_t denote the spatio-temporal partial derivatives of the image intensity and (u, v) is the optical flow vector. These assumptions are violated if the neighborhood contains motion boundaries or multiple transparent motions.

In this paper, we would like to address the problem of multiple motion analysis from the point of view of orientation analysis using a new steerable filter. In the following, we would like to review some related works at first.

1.1 Spatial Approaches

Nagel and Enkelmann [33] first addressed the violation of single motion model at motion boundaries. They used a regularization term to penalize motion discontinuities and thus rejected motion boundaries in the computation of optical flow. Weickert and Schnörr further extended this regularization term into spatio-temporal space [44]. Similarly, Black and Anandan [10] treated occlusion regions as *outliers* of the motion constraint and set lower weights to these regions in the estimation. The concept of *outlier* represents exactly the relationship between the pixels near occlusion boundaries and the pixels with a single motion: The spatio-temporal partial derivatives of the pixels with a single motion form a plane in the derivative space and the derivatives of the pixels near occlusion boundaries deviate from this plane due to motion discontinuities.

Based on this concept, probabilistic methods were proposed to model occlusion boundaries [28] and to estimate motions near occlusion boundaries [13, 7]. The outlier was also considered as noise in statistic methods (e.g. [38]) as well as in the Hough transform based approaches (e.g. [12]).

A second group of approaches was based on segmenting regions where parametric models of flow can be fitted [47]. But the segmentation is in turn dependent on the optical flow. This forms a "chicken-and-egg" dilemma: We need appropriate segmentation to estimate optical flow accurately; while we need accurate optical flow to segment images properly. In order to bypass this dilemma, iterative methods were used to refine the region segmentation and motion estimation gradually [24]. Bergen *et al.* [7] proposed an iterative method based on the "shift-and-subtract" strategy to estimate two motions. They subtracted pixels connected to one motion during the refinement of the parameters of the other motion and *vice versa*. Irani *et al.* [26] applied this iterative method in the temporal integration to blur out uninterested regions and track objects even with non-consistent speeds. The iteration principle was used in the statistic approaches as well. Researchers have recently elaborated algorithms based on the expectation-maximization (EM) principle [14] for motion estimation (e.g. [46]).

In presenting explicit multiple motion models many researchers have made contributions. Wang and Adelson represented multiple motions with a multi-layer model [43]. This model is consistent with the daily knowledge naturally. But the local motion estimation technique they used for estimation is still based on the BCCE. Fleet *et al.* [15] explicitly modeled the occlusion boundary in the spatial domain as a step edge in both components of the optical flow field and used the steerability theory to detect the boundary. Black and Fleet further proposed to use the Bayesian framework to determine which pixels belong to the motion boundary regions [11].

1.2 Spectral Approaches

Motion estimation was also addressed from the point of view of orientation analysis. Adelson and Bergen [1] pointed out that motion is equivalent to spatio-temporal orientation and introduced a spatio-temporal energy model for single motion representation. This was the first optical flow algorithm based on the spectral analysis.

Bigün *et al.* connected the orientation analysis with symmetry detection [8, 9]. They pointed out that a single motion can be described as a linear symmetric image, whose spectrum is a line passing through the origin in the frequency domain. They fixed the orientation of the spectral line by minimizing a moment measure in the frequency do-

main. Jähne also used a 3D structure tensor [29] to detect symmetry and to estimate motion [27]. He further introduced an eigenvalue based coherence measure to distinguish different kinds of motions such as single constant motion and motion discontinuities.

Both Bigün *et al.* [8, 9] and Jähne [27] used the principal axis analysis method to estimate motion. The principal axis analysis is also called principal component analysis (PCA), Karhunen-Loève transform (KLT), and Hotelling transform in different literature. It decomposes a signal into an orthogonal basis using eigenvector analysis or singular value decomposition (SVD). The problem of principal axis analysis is that it is only suitable to detect one dominant orientation because the rest eigenvectors are orthogonal to the first eigenvector, no matter what kind of structure the signal has. In other words, the orientation resolution of the principal axis analysis is not sufficient to solve a non-orthogonal multiple orientation problem.

1.3 Gabor Based Approaches

In order to determine the orientation of the motion plane in the frequency domain, Heeger sampled the spectrum of the image sequence with twelve Gabor filters [22]. Heitger *et al.* proposed a variant of 2D Gabor filter whose odd and even parts have zero mean values [23]. They called this variation "stretched Gabor" filter. Similarly, Xiong and Shafer used hypergeometric filters to sample the spectrum for motion estimation [49]. The hypergeometric filter is based on one kind of special function called *confluent hypergeometric function* in the mathematical physics. It is very similar to a Gabor filter in shape, but its DC component is zero. For details about the *confluent hypergeometric function* the reader is referred to [34].

Grzywacz and Yuille [19] further pointed out that the angular support of a filter in the spectral domain is a measure of angular uncertainty. This uncertainty can be expressed as the angle between two tangential lines of the support, which pass through the spectral origin. In orientation analysis, the filters at different frequencies are desired to have the same angular uncertainty. This is exactly the property of Gabor wavelets [30, 48].

One main concern of Gabor/hypergeometric filter based approaches is the enormous complexity of computation in sampling the spectral domain with fine resolution. Another concern is the positive skewness in the filter responses of the Gabor wavelets [19]. If we have only one single motion, the maximum is still well localized in spite of the skewness. But if we have multiple motions instead, the overlapping of different filter responses, specially the overlapping of the skewness will disturb the locations of maximal values [51]. Thus, the estimation results are no more convincible.

1.4 Our Contribution

In this paper, we address the problem of multiple motion analysis from the point of view of orientation analysis by pointing out that multiple motions are equivalent to multiple planes in the derivative space or in the frequency domain. Correspondingly, the motion parameters are determined by the normal vectors of these planes. This claim is a generalization of the spatio-temporal energy model of single motion [1].

Since the angle between two motion planes can be arbitrary, we need a filter with fine orientation resolution to estimate motion parameters exactly. Though the enormous complexity and the skewness obstruct the application of Gabor based approaches in multiple motion analysis, the idea of sampling the orientation space locally is still attractive since it can achieve fine orientation resolution. The remaining problem is how to reduce the computation complexity. To solve this problem, we propose a new 3D orientation steerable filter.

This paper is constructed as follows: The following section studies occlusion and transparency in detail for a better understanding of multiple motions. Then we introduce the new 3D steerable filter and compare it with current 3D steerable filters as well as 3D orientation histogram in section 3. In the same section we also display the filter responses of 3D planes and confirm the comparisons between the spatial- and spectral-motion models. After that we use our filter for multiple motion estimation in section 4 and compare it with the 3D Hough transform and the expectation-maximization (EM) algorithm. Section 5 further discusses the multi-window strategy in occlusion estimation and the precision improvement of estimation results after eliminating outliers. Later on we use the "shift-and-subtract" technique to localize and to track motion boundaries in occlusion sequences in section 6. In section 7 we show experiment examples. Finally, this paper is concluded in section 8.

2 Understanding Multiple Motions

In this section, we explain that occlusion is equivalent to multiple planes both in the (I_x, I_y, I_t) space and in the frequency space and transparency is equivalent to multiple planes *only* in the frequency space. Correspondingly, multiple motion analysis is equivalent to orientation analysis of planes.

2.1 Spatial Observation of Multiple Motions

Although occlusion and transparency can be decomposed into multiple layers, they are based on different decomposition principles. We illustrate this difference in figure 1, where we observe that occlusion is more local than transparency in the spatial domain: while occlusion involves a step-function at the occlusion boundary, transparency results from the overlapping of two motions in the entire window. Consequently, it is difficult to describe both kinds of multiple motions using a unified model in the spatial domain. Thus, we turn to the frequency domain to look for the solution.

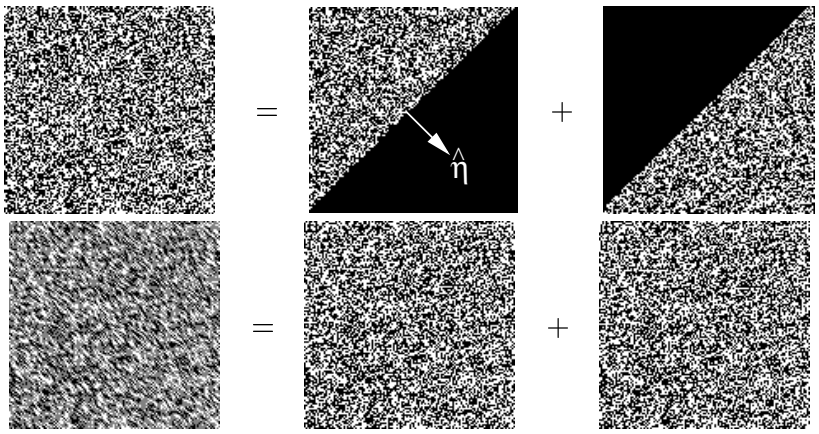

Figure 1: Difference between occlusion and transparency. Here random dot regions represent motions and dark regions denote static status. The occluding signal is moving with the speed $(1, 1)$ [pixel/frame] and the occluded signal with $(1, -1)$ [pixel/frame] and the speeds of transparent motions are $(1, 1)$ [pixel/frame] and $(1, -1)$ [pixel/frame] as well. **Top:** One frame of the occlusion sequence is decomposed into two layers by a Heaviside unit step function (equations (2) and (3)). There is motion discontinuity only at the boundary. The term $\hat{\eta}$ denotes the unit vector normal to the occluding boundary. **Bottom:** One frame of the transparency sequence is a simple superposition of two layers (equation (9)). Multiple motions exist in the entire window.

2.2 Spectral Analysis of Occlusion

The spectrum of multiple motions was first analyzed by Fleet and Langley [16]. Assuming that an occlusion boundary is a characteristic function $\chi(\mathbf{x})$, they modeled the

occlusion in the spatial domain as follows:

$$I(\mathbf{x},t) = \chi(\mathbf{x}-\mathbf{v}_1 t)I_1(\mathbf{x}-\mathbf{v}_1 t) + [1-\chi(\mathbf{x}-\mathbf{v}_1 t)]I_2(\mathbf{x}-\mathbf{v}_2 t), \tag{2}$$

where $I_1(\mathbf{x})$ is a 2D **occluding** signal moving with velocity $\mathbf{v}_1 = (v_{1x}, v_{1y})^T$ and $I_2(\mathbf{x})$ is a 2D **occluded** signal moving with velocity $\mathbf{v}_2 = (v_{2x}, v_{2y})^T$.

Beauchemin and Barron [5, 6] were the first who formulated an exact model in the frequency domain. They modeled the occlusion in the spatial domain with a Heaviside unit step function $U(\mathbf{x})$ for $\chi(\mathbf{x})$:

$$U(\mathbf{x}) = \begin{cases} 1 & \mathbf{x}^T\hat{\eta} \geq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where $\mathbf{x}$ denotes 2D spatial Cartesian coordinates and $\hat{\eta}$ is the unit vector mentioned above.

We denote the spatial frequency vector as $\kappa = (\omega_x, \omega_y)^T$ and the temporal frequency as ω_t. Then, the Fourier transform of the image sequence reads

$$\begin{aligned} \tilde{I}(\kappa,\omega_t) \quad &= \tilde{U}(\kappa)\delta(\kappa^T\mathbf{v}_1+\omega_t) * \tilde{I}_1(\kappa)\delta(\kappa^T\mathbf{v}_1+\omega_t) + \tilde{I}_2(\kappa)\delta(\kappa^T\mathbf{v}_2+\omega_t) \\ &\quad -\tilde{U}(\kappa)\delta(\kappa^T\mathbf{v}_1+\omega_t) * \tilde{I}_2(\kappa)\delta(\kappa^T\mathbf{v}_2+\omega_t), \end{aligned} \tag{4}$$

where $*$ means convolution and $\tilde{}$ denotes the Fourier transform of the corresponding signal. The spectrum of the Heaviside unit step function is given by

$$\tilde{U}(\kappa) = 2\pi[\pi\delta(|\kappa|) + \frac{\delta(\kappa^T\hat{\eta}_\perp)}{i\kappa^T\hat{\eta}}], \tag{5}$$

where $\hat{\eta}_\perp$ denotes a unit vector perpendicular to $\hat{\eta}$. Taking the properties of the impulse function into account, we obtain (see [52] for detail)

$$\tilde{I}(\kappa,\omega_t) = [2\pi^2\tilde{I}_1(\kappa) + A(\kappa)]\delta(\kappa^T\mathbf{v}_1+\omega_t) + (1-2\pi^2)\tilde{I}_2(\kappa)\delta(\kappa^T\mathbf{v}_2+\omega_t) + B(\kappa,\omega_t), \tag{6}$$

with

$$A(\kappa) = \frac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp) * \tilde{I}_1(\kappa), \tag{7}$$

$$B(\kappa,\omega_t) = \frac{2\pi}{i\kappa^T\hat{\eta}}\delta(\kappa^T\hat{\eta}_\perp)\delta(\kappa^T\mathbf{v}_1+\omega_t) * \tilde{I}_2(\kappa)\delta(\kappa^T\mathbf{v}_2+\omega_t). \tag{8}$$

The first two terms of expression (6) are two oriented planes passing through the origin of the frequency space. Their normal vectors, namely $(u_1, v_1, 1)$ and $(u_2, v_2, 1)$ contain the velocities. The second term is the exact spectrum of the occluded signal. The first term contains an additional distortion term $A(\kappa)$ on the plane of the occluding spectrum. However, here we are only interested in the orientation of the plane and the term $A(\kappa)$ does not disturb the orientation. Actually, $A(\kappa)$ strengthens this spectral plane. Therefore, we do not consider it as distortion of orientation analysis. The main discriminating term is the third one, $B(\kappa, \omega_t)$, which is a convolution between a line and a plane in 3D frequency space. Only when $\tilde{I}_2(\kappa)$ is discrete we may have some shifted *lines* formed by the spectral plane $\kappa^T \hat{\eta}_\perp = 0$ and the spectral plane of the occluding signal $\kappa^T \mathbf{v}_1 + \omega_t = 0$ [6]. Otherwise, $B(\kappa, \omega_t)$ is a 3D volume filling the entire frequency space. This volume depends on *both* speeds of the occluding and the occluded signal as well as the normal of the occluding boundary.

If the energy of the distortion term $B(\kappa, \omega_t)$ is very high, we will not be able to recognize these two planes. Fortunately, the critical factor in the amplitude of $B(\kappa, \omega_t)$ is the hyperbolic term $\frac{2\pi}{i\kappa^T \hat{\eta}}$ which reduces very quickly with the increase of $|\kappa|$. In most regions of the spectral domain, say for $|\kappa| \geq 1$, the amplitude of the distortion is much less than that of signals. Therefore, we may consider the spectrum only above a lower bound of the frequency and identify the two dominant planes.

2.3 Spectral Analysis of Transparency

Transparency may be viewed as a special case of occlusion by simply substituting $\chi(\mathbf{x} - \mathbf{v}_1 t)$ with a real constant a $(a \in (0,\ 1))$ [5]

$$I(\mathbf{x}, t) = aI_1(\mathbf{x} - \mathbf{v}_1 t) + (1 - a)I_2(\mathbf{x} - \mathbf{v}_2 t). \tag{9}$$

The corresponding spectrum is then characterized by two oriented planes without distortion

$$\tilde{I}(\kappa, \omega_t) = a\tilde{I}_1(\kappa)\delta(\kappa^T \mathbf{v}_1 + \omega_t) + (1 - a)\tilde{I}_2(\kappa)\delta(\kappa^T \mathbf{v}_2 + \omega_t). \tag{10}$$

2.4 Spectral Model of Multiple Motions

Though in the case of occlusion there exists a distortion term, the main energy proportion is on the two spectral planes due to the hyperbolic nature of the distortion term. In addition, we may low-stop the energy spectrum to abandon the strong disturbance at

low frequencies. Thus, both occlusion and transparency are characterized as multiple spectral planes passing through the origin, and the corresponding motion speeds are described by the normal vectors of these planes.

This model can be viewed as a generalization of the spatio-temporal energy model of single motion [1, 22]. At first sight it is very similar to the work of Shizawa and Mase [41], which assumed blindly that multiple motions are additive superposition of two single motions in the frequency domain or in the derivative space [41]. But there are two different points in our work:

- Shizawa and Mase proposed that multiple motions are characterized as multiple planes both in the (I_x, I_y, I_t)-space and in the frequency domain. We argue that even the model is correct in the (I_x, I_y, I_t)-space, it is not feasible since we can hardly assume that the intensity profile is differentiable in case of transparency.
- At low frequencies multiple planes are disturbed by the distortion term of the occlusion according to our analysis. We have to truncate low frequency components in order to fit multiple planes robustly.

2.5 Comparison between Spatial and Spectral Model

According to the analysis in the above subsection, the spectral model describes both occlusion and transparency, while the spatial model can describe only occlusion. If we do not have *a priori* knowledge about motions, we should use the spectral model.

It should be noticed that though we have a thorough analysis of the spectral motion model, there exists a severe problem in obtaining the energy spectrum of the image sequence due to the block effect of the discrete Fourier transform (DFT). This is actually one main problem in the frequency-based techniques. In order to avoid the block effect of DFT, we take a local Fourier transform (LFT), i.e. Gaussian windowed DFT. According to the convolution theorem in the Fourier analysis, Gaussian windowed DFT of the image sequence is equivalent to the convolution between the spectrum of the image sequence and the Gaussian function. Consequently, the spectrum is blurred after the LFT. Thus, the spectral model has worse resolution than the spatial model. For compensation we have to enlarge the window to improve the frequency resolution with finer interval. But this compensation has also limitations. First, the constant motion assumption in this enlarged neighborhood is more fragile. Second, using larger window means including more frames in the estimation, but we can hardly assume that the motion is constant over very large time interval. Thus, if we have only occlusion, we

prefer to apply the spatial model and use the concept of *outliers* to treat the derivatives near occlusion boundaries.

3 3D Gaussian Steerable Filter

In the previous section, we mentioned that the parameters of multiple motions are described by the normal vectors of motion planes. In order to extract the parameters of these planes exactly, we need a filter with fine orientation resolution. But we have to face an enormous complexity of computation while constructing or rotating such filters.

In order to attenuate this conflict, the concept of steerability was introduced [18] and many 2D steerable filters have been applied in image processing and low level computer vision ([32, 35, 31, 42, 17]). However, till now there were only a few approaches dealing with 3D steerability [18, 2] which mainly use the global decomposition method.

Freeman and Adelson [18], being the first who introduced the concept of steerability into 3D filtering, interpolated derivatives of 3D Gaussian functions with a set of basis filters which are rotated copies of the original filter. The corresponding interpolation functions are trigonometric functions of the orientation parameters. For example, they need only three basis filters to synthesize the first derivative of 3D Gaussian filter $G_1(x, y, z)$ in an arbitrary direction. The enormous complexity in rotating this filter to different directions is therefore strongly attenuated.

Andersson designed another 3D steerable filter in the frequency domain [2], whose basis filters read

$$B_{li}(\overline{u}) = G(\rho)(\hat{n}_{li} \cdot \hat{u})^l, \tag{11}$$

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. In his approach, the basis filters with the same order have the same shape. Furthermore, the orientation of basis filters is arranged in such a way that the basis filters with the same order also span evenly on the sphere surface. Correspondingly, the synthesized filters of order l are rotated copies of one basis filter with the same order. 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 $\frac{(l+1)(l+2)}{2}$.

Here we would like to address the problem of orientation resolution in both approaches. As a matter of fact, we only need to analyze the performance of one basis filter since the synthesized filters are rotated copies of one basis filter. In the work of Freeman and Adelson, the derivatives of Gaussian functions have coarse orientation resolution due to their large angular supports in the orientation space, as shown in figure 4. In Andersson's work, the basis filters are centered at the vertices of the corresponding regular polyhedron. In order to span the whole sphere surface with a set of basis filters, the angular support of each basis filter should not be smaller than a facet of the corresponding regular polyhedron. It is confirmed that even for an icosahedron, which is a matching regular polyhedron with the smallest facet, the corresponding support of one facet is not yet small enough [24]. Thus, the steerable filter proposed by Andersson does not provide sufficiently fine resolution, either. In order to improve the resolution, we propose a new kind of 3D steerable filter in the following subsection.

3.1 Definition of Our 3D Steerable Filter

To analyze local 3D orientation naturally, we first 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 2). Here, the goal is to build an orientation signature $S(\theta, \phi)$ from $I(r, \theta, \phi)$. In order to have fine orientation resolution, we use *conic kernels* with small angular supports as basis filters to sample the orientation space locally. A *conic kernel* centered at (θ_i, ϕ_j) reads

$$B_{(\theta_i,\phi_j)}(r, \theta, \phi) := \frac{G_0^{(\theta_i,\phi_j)}(\theta, \phi)}{\mathcal{N}_{R_{min},R_{max}}^{(\theta_i,\phi_j)}(r)}, \tag{12}$$

where $\mathcal{N}_{R_{min},R_{max}}^{(\theta_i,\phi_j)}(r)$ is a weighting function along the radial direction and it is independent of the angular part of the filter. We will come back to the design of $\mathcal{N}(r)$ later. The angular part of the kernel is a 2D Gaussian function in the orientation space coordinated with (θ, ϕ)

$$G_0^{(\theta_i,\phi_j)}(\theta, \phi) := \frac{1}{2\pi\sigma^2} e^{-\frac{(\mathcal{D}(\theta,\theta_i))^2+(\phi-\phi_j)^2}{2\sigma^2}}, \tag{13}$$

where σ denotes the scale of the 2D Gaussian function. Since the angles along the θ direction are periodic, we define $\mathcal{D}(\cdot)$ to represent the minimal circular difference between θ and θ_i ($\theta, \theta_i \in [0, 2\pi]$)

$$\mathcal{D}(\theta, \theta_i) := \min(|\theta - \theta_i|, |\theta - \theta_i - 2\pi|, |\theta - \theta_i + 2\pi|). \tag{14}$$

Theoretically, a Gaussian function is not compactly supported. Thus, in implementation we only consider the part of $G_0^{(\theta_i,\phi_j)}(\theta,\phi)$ inside the circular mask with a diameter D, as shown in figure 2.

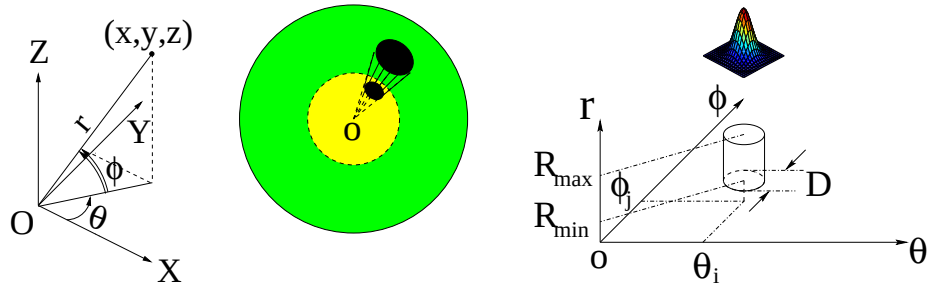

Figure 2: A *conic kernel* centered at (θ_i, ϕ_j) with radial boundaries R_{min} and R_{max}. **Left:** The definition of the spherical coordinate system. **Middle:** The filter kernel in the 3D Cartesian coordinate system. The keypoint is at the center of the sphere. **Right:** The filter kernel with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

After applying such a *conic kernel* on $I(r,\theta,\phi)$, we get a basis filter response as a local sample located at (θ_i, ϕ_j)

$$A_{(\theta_i,\phi_j)} := \sum\sum_{\{(\theta,\phi)|\sqrt{(\theta-\theta_i)^2+(\phi-\phi_j)^2}\leq\frac{D}{2}\}} G_0^{(\theta_i,\phi_j)}(\theta,\phi) \sum_{r=R_{min}}^{R_{max}} \frac{I(r,\theta,\phi)}{\mathcal{N}_{R_{min},R_{max}}^{(\theta_i,\phi_j)}(r)}. \tag{15}$$

Now let us consider the sampling of (θ, ϕ) plane using a set of basis filters. It is known that a sphere surface forms a rectangular region in the (θ, ϕ) plane. For this rectangular region it is impossible to have a tessellation with circular cells. Instead, we may overlap neighboring basis kernels to cover the whole rectangular region, as shown in figure 3. In this arrangement we observe that this rectangular region is periodic along the θ direction and is mirror-symmetric about the boundary along the ϕ direction. These periodic and mirror-symmetric properties help to solve the boundary problem.

In order to construct the orientation signature $S(\theta, \phi)$ from a set of samples $A_{(\theta_i,\phi_j)}$, we use 2D Gaussian functions with local support $G_0^{(\theta_i,\phi_j)}(\theta,\phi)$ again as interpolation functions yielding

$$S(\theta,\phi) := \sum_{\theta_i}\sum_{\phi_j} A_{(\theta_i,\phi_j)} G_0^{(\theta_i,\phi_j)}(\theta,\phi). \tag{16}$$

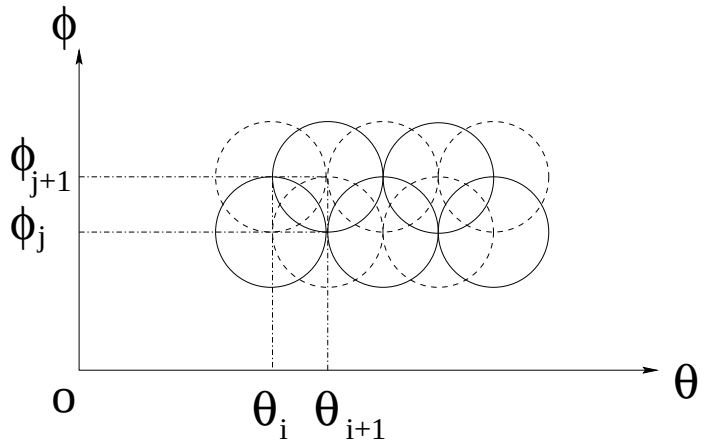

Figure 3: The sampling of (θ, ϕ) plane using a set of *conic kernels*. The horizontal or vertical distance between two neighboring masks is equal to the radius of one mask.

The legality of using 1D Gaussian functions as interpolation functions was already proved in [36]. Our approach can be viewed as an extension into 2D feature space. So far, we have defined an analytic model of 3D orientation analysis based on angular Gaussian functions.

3.2 Comparisons with Current 3D Steerable Filters

Current 3D steerability approaches are based on the global decomposition principle. In contrast, our 3D filter is based on the local decomposition principle. This difference leads our approach to have higher orientation resolution. In figure 4 we show the filter G_1 in the work of Freeman and Adelson [18], Andersson's third order filter [2] (whose ten basis filters span evenly on the sphere surface with the finest angular support), and our filter, respectively. We also display the angular supports of these filters in the (θ, ϕ) space since the orientation resolution of a filter can be measured by the angular support of this filter. This can be done by integrating the filter kernels over the radial variable. Note that the angular support of the filter in the spatial domain is the same as that in the frequency domain since the Fourier transform is an isometric mapping. The irregularity in the (θ, ϕ) space with $|\phi| > 40^\circ$ is caused by the discrete representation of filter kernels. We notice that 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 fix the orientation of *one* plane in the single motion estimation [25]. Obviously, G_1 cannot detect multiple planes simultaneously. The orientation resolution of Andersson's filter

is only a litte bit better. Compared with these two steerable filters, our filter has a much higher orientation resolution.

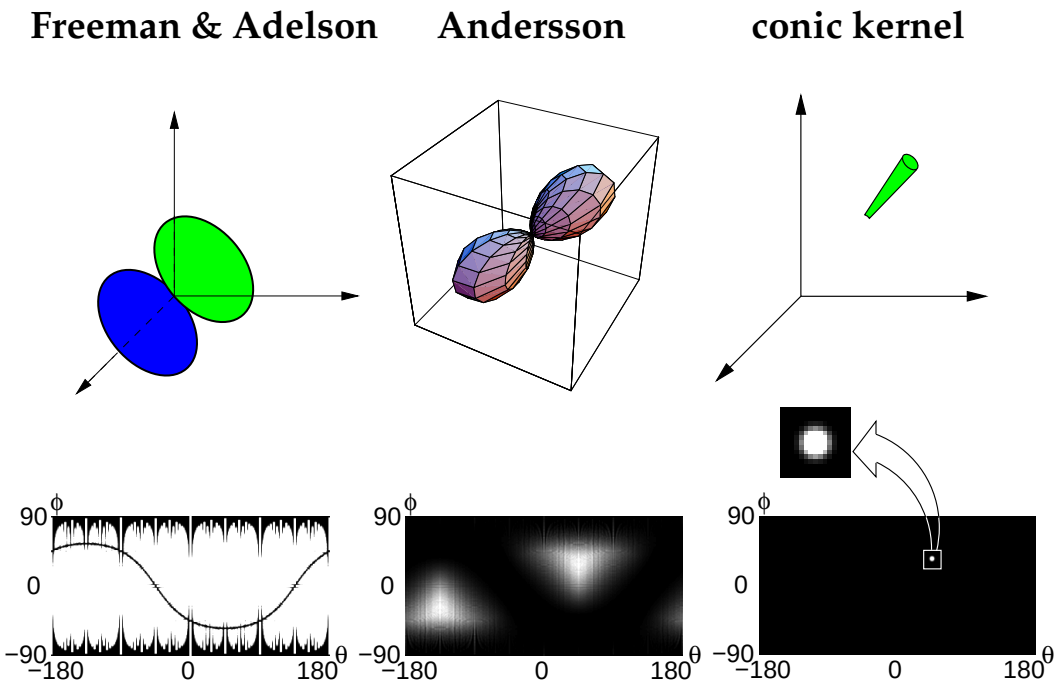

Figure 4: **Row 1:** The rendering image of filter kernels. **Left:** The filter G_1 in the work of Freeman and Adelson (redrawn from [25]). **Middle:** The third order filter in Andersson's approach (redrawn from [2]). **Right:** Our filter. **Row 2:** The corresponding angular supports for above filters centered at $\theta = 45.00°, \phi = 35.26°$ are shown with white regions in the (θ, ϕ) space. These supports are actually measurements of the orientation resolution of the filters. For clarity we enlarge the angular support of our filter in an extra image.

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 and our filter are based on different decomposition principles, we can compare their complexity only by considering the computational burden per pixel in the input data. Concretely,

- The filter G_1 is composed of three basis filters with the *global* support, i.e. each basis filter covers the input data completely. Thus, each pixel in the input data is involved in the scalar product as well as in the interpolation procedure three times.

- The third order filter in Andersson's approach has ten basis filters. Thus, each pixel in the input data is involved in the scalar product and the interpolation procedure ten times.
- Our filter is based on the local decomposition principle. In figure 3 we observe that the quadratic area bounded by four lines $\theta = \theta_i$, $\theta = \theta_{i+1}$, $\phi = \phi_j$, and $\phi = \phi_{j+1}$ is covered by four quarter circular masks. Without studying the overlapping exactly, we may roughly say that a pixel in this quadratic area is involved in the scalar product four times. As the interpolation function has the same support as the basis filter we know that a pixel in this quadratic area is involved in the interpolation four times as well.

From above analysis, our filter needs a little bit more computation than the filter G_1 but much less computation than Andersson's filter.

It should be noticed that a complexity comparison is only fair, when the corresponding filters provide (about) the same orientation resolution. This is not the case for the three 3D steerable filters mentioned above. Actually, neither the filter G_1 nor Andersson's filter can achieve the same fine orientation resolution that our filter provides. One possibility to achieve such a fine orientation resolution using global decomposition method is to generalize the steerable wedge filter [42] from 2D space to 3D space with considerably higher effort [53], which is not yet implemented according to current literature.

3.3 Comparisons with 3D Orientation Histogram

Our filter is related to the extended Gaussian image (EGI). An EGI maps the surface of a 3D object onto the unit sphere surface, in which a small facet of the object is transformed into a point whose orientation is the same as that of the small facet and whose weight is the area of the small facet. For convex objects it is proven that their corresponding EGIs are unique.

In practice, we usually use the discrete approximation of EGI, which is referred to as 3D orientation histogram. In order to construct the 3D orientation histogram, we must at first tessellate the unit sphere surface. According to the ideal tessellation criteria, the sphere surface should be divided into cells with the same area and the same rounded shape. In addition, these cells should be located as a regular pattern and should provide fine angular resolution [24]. Unfortunately, these criteria cannot be fulfilled at the same time.

With respect to the orientation resolution and decomposition principle our approach is

very similar to the 3D orientation histogram. Both techniques achieve high orientation resolution and both methods decompose the sphere locally. However, there are still differences between them.

- The 3D orientation 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 treats both convex and concave objects. But we must fix the keypoint and the radial boundaries at first.
- The 3D orientation histogram works on a unit sphere surface, while our approach projects the sphere onto the (θ, ϕ) space and works on this orientation space. Though after this non-isometric mapping we lose the rotational symmetry, we gain both easier structure display and post-processing as compensation. For example, on the surface of this paper sheet we cannot display all parts of a great circle of a sphere using the 3D orientation histogram. We have to imagine in our mind that there are parts hiding behind the paper sheet. In contrast, we can display the great circle completely on the (θ, ϕ) space, though with some deformation.
- The basis cells in the 3D orientation histogram cannot fulfill all ideal tessellation criteria simultaneously. Furthermore, the basis cells are not *isotropic*, i.e. they are not rotation invariant for *every* direction. In contrast, our approach provides *isotropic* cells in the (θ, ϕ) space (not on the sphere!) satisfying these criteria.
- The 3D orientation histogram is based on the tessellation of the unit sphere. Each pixel on the sphere is involved *once and only once*. Our approach samples the (θ, ϕ) plane with Gaussian kernels. Since these basis kernels overlap, each pixel on the sphere is therefore involved *several times*. From the point of view of computation complexity, our method needs more computation than the 3D orientation histogram. Of course we can divide the (θ, ϕ) plane with a set of small rectangular cells which do not overlap. But these cells are then non-isotropic. Thus, overlapping is actually the price of having isotropic cells.

3.4 Compensation Issue

Now we come to the design of the weighting function $\mathcal{N}(r)$ (see equation (12)). It is known that the horizontal angle θ and the vertical angle ϕ are defined differently in the spherical coordinates. 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 lie on a small circle. If we divide the whole (θ, ϕ) space with a homogeneous grid, it turns out that the higher the latitude

value is, the denser the grid points are on the sphere surface. This kind of non-uniform distribution was addressed in [24] in detail.

We may average this non-uniform distribution in 3D space by designing the weighting function $\mathcal{N}(r)$ as the sum of discrete weights in the basis kernels so that the filter response is relatively insensitive to the non-uniform distribution. This compensation "strengthens" the outputs of filter kernels with a few points and "suppresses" those outputs of filter kernels with many points. As a result, we are no more able to know the real distribution density of points in the (θ, ϕ) space. However, the density information is desirable in many motion estimation approaches. 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 [37]. The distribution density actually works as a weighting factor in the parameter regression procedure. If we lose the distribution density information, the estimation result will be much worse. For this reason, we would like to preserve the distribution density information by setting $\mathcal{N}(r)$ as a real positive constant.

3.5 Our Filter Responses of 3D Planes

For the sake of motion estimation, we are interested in our filter responses of 3D planes. In the 3D Cartesian coordinate system, a plane passing through the origin $(0, 0, 0)$ with a unit normal vector $\mathbf{n} = (n_1, n_2, n_3)^T$ reads

$$xn_1 + yn_2 + zn_3 = 0. \tag{17}$$

In order to represent this plane with parameters θ and ϕ, we convert the Cartesian coordinates into spherical coordinates $(x, y, z) \to (r, \theta, \phi)$ and $(n_1, n_2, n_3) \to (1, \theta_n, \phi_n)$. After wiping out the radial variable r we acquire an equation of the 3D plane with variables θ and ϕ

$$\cos(\phi)\cos(\phi_n)\cos(\theta - \theta_n) + \sin(\phi)\sin(\phi_n) = 0. \tag{18}$$

Different planes have different representations in the (θ, ϕ) space. For horizontal and vertical planes whose normal vectors are parallel to the coordinate axes, their corresponding representations in the (θ, ϕ) space are straight lines, as shown in figure 5. In contrast, tilted planes turn into periodic curves in the (θ, ϕ) space, as shown in figure 6. These curves look like trigonometric functions with different amplitudes and phases. For each curve, if we know the extreme point with the maximal ϕ coordinate, ϕ_m, and

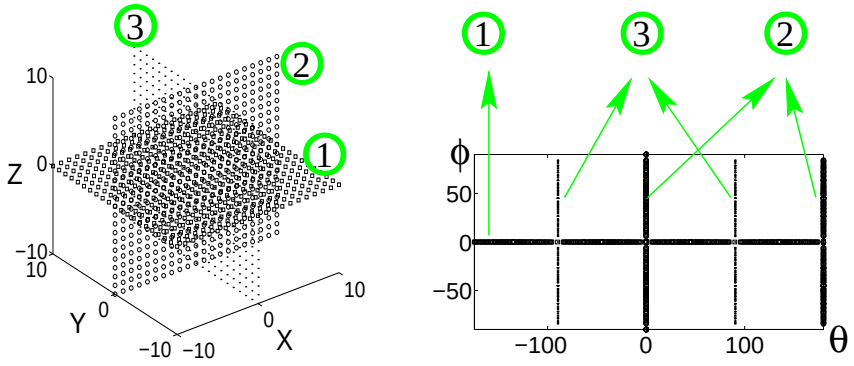

Figure 5: **Left:** Three special planes in the Cartesian coordinates with normal vectors $(0, 0, 1)$, $(1, 0, 0)$, and $(0, 1, 0)$, respectively. **Right:** Special planes in the (θ, ϕ) space.

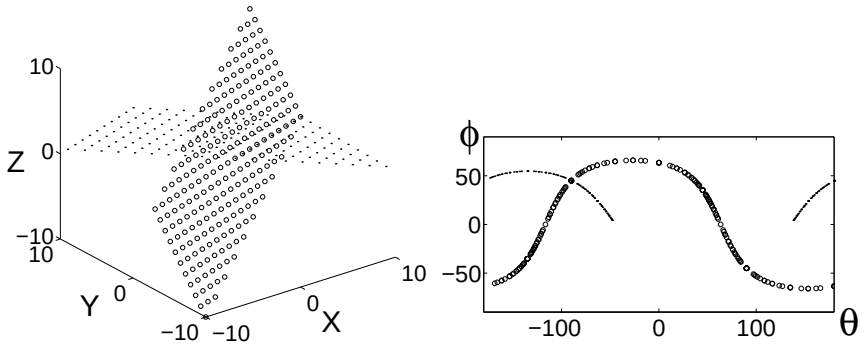

Figure 6: **Left:** A plane with normal vector $(-2, 1, 1)$ and a plane with normal vector $(1, 1, 1)$ in the Cartesian coordinates. **Right:** The corresponding curves in the (θ, ϕ) space. Since the components on the plane with normal vector $(1, 1, 1)$ have only positive z coordinates, we observe only positive ϕ coordinates on the corresponding curve.

the corresponding θ coordinate, θ_m, then we can find out the normal vector of the corresponding plane (see [51] for derivation)

$$\begin{cases} \theta_n &= \theta_m \pm \pi \\ \phi_n &= \frac{\pi}{2} - \phi_m \end{cases} . \tag{19}$$

Here we use $+$ or $-$ sign to determine θ_n in equation (19) in such a way that the third component n_3 of the normal vector is positive. Then we can use θ_n and ϕ_n for motion estimation. Besides, the θ_m locates in the middle of two zero-crossing points of the θ axis, while these two zero-crossing points have a distance of π along the θ axis. This nice property is very useful in determining the number of motions automatically as well

as in obtaining properly initial values of motion parameters, as we will show in section 4. 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 the least square estimation (LSE) algorithm is applicable; for multiple curves we may apply the EM algorithm. We will give particulars of this point in section 4.

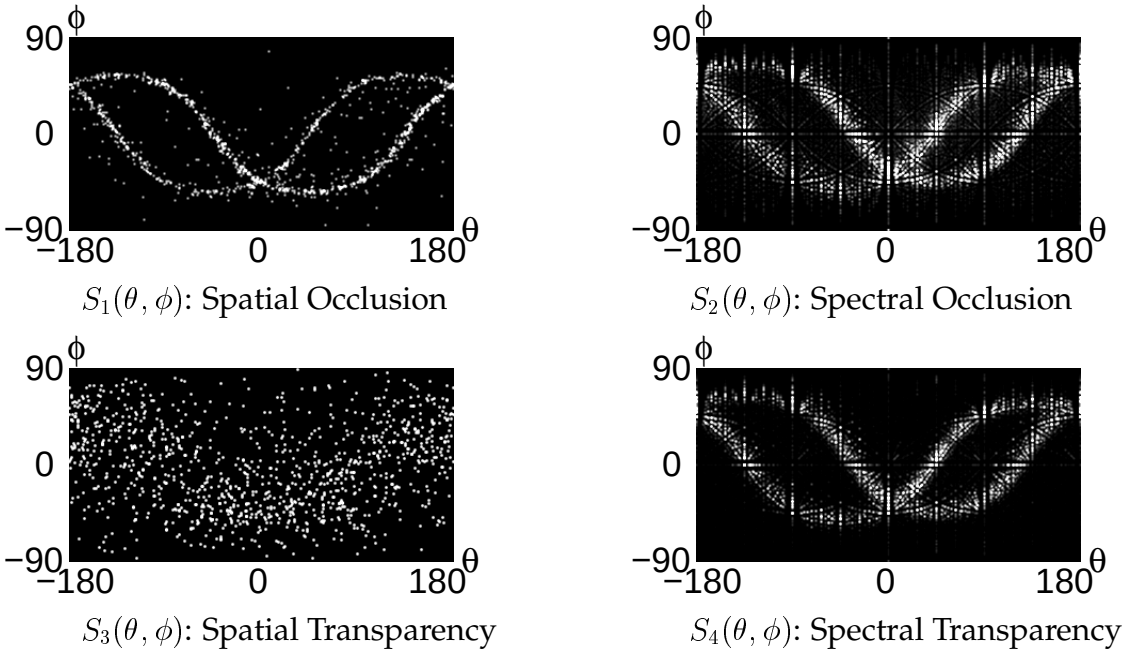

Figure 7: Orientation signatures of occlusion and transparency sequences in figure 1. **Top Left:** The orientation signature of the occlusion sequence based on the spatial motion model. We use a $33 \times 33 \times 1$ window in the derivative space to obtain this signature. **Top Right:** The signature of the occlusion sequence based on the spectral motion model. We use a $32 \times 32 \times 32$ window to obtain this signature. **Bottom Left:** The signature of the transparency sequence based on the spatial model. The distribution of points is nearly random. **Bottom Right:** The signature of the transparency sequence based on the spectral model.

After explaining the filter responses of 3D planes, we can confirm the comparisons between the spatial and the spectral model now. In figure 7 we display the orientation signatures of both occlusion and transparency. These orientation signatures are obtained by applying our 3D steerable filter on the derivative space or on the energy spectrum of the image sequences shown in figure 1. Note that multiple planes in both derivative space and spectral domain pass through the origin. Therefore, it is very easy to set the origin as the keypoint for the application of our 3D steerable filter. This is a great

advantage compared with 2D junction characterization, where we have to locate the keypoints at first [53]. Since both occlusion and transparency sequences have the same motion parameters, we expect that their orientation signatures have the same curves. A comparison between two rows shows that the spectral model can treat both occlusion and transparency, while the spatial model can treat only occlusion. In the spectral signatures, we observe distortions outside two main curves in $S_2(\theta, \phi)$, while in $S_4(\theta, \phi)$ these distortions disappear. Besides, a comparison between $S_2(\theta, \phi)$ and $S_1(\theta, \phi)$ confirms that the spectral model has coarser resolution than the spatial model since the spectrum is blurred by LFT.

Taking into account that we have to use a large window (here the window size is $32 \times 32 \times 32$) to obtain the orientation signature in the spectral domain and that the constant motion assumption is easily violated in such a large window, we prefer to use the spatial model for occlusion analysis.

4 Multiple Motion Estimation Using 3D Steerable Filter

After applying our 3D filter on the input sequence, we obtain an orientation signature. For parameter extraction we still need further processing like the EM algorithm. Since the equation (17) based 3D Hough transform as well as the planar EM algorithm can extract the orientation parameters of planes **directly**, we face the following question: Why do we project the 3D data onto the 2D feature space before extracting parameters? In order to answer this question, we must analyze the 3D Hough transform and the EM algorithm in more detail.

The Hough transform is a sampling and searching method for parameter extraction. Concretely, in equation (17) we would like to extract the normal vector (n_1, n_2, n_3) from a set of points coordinated with $(I_{ix}, I_{iy}, I_{it})(i = 1, \cdots, N)$. For each point (I_{ix}, I_{iy}, I_{it}) we draw the corresponding vectors in the (n_1, n_2, n_3) space satisfying the equation

$$I_{ix}n_{1j} + I_{iy}n_{2j} + I_{it}n_{3j} = 0, \qquad (j = 1, \cdots) \tag{20}$$

where (n_{1j}, n_{2j}, n_{3j}) denotes the j-th vector normal to (I_{ix}, I_{iy}, I_{it}). After going through all points, we search in the (n_1, n_2, n_3) space the position with the maximal number of vector intersections. From the corresponding coordinates (n_{1m}, n_{2m}, n_{3m}) we obtain the desired motion parameters

$$\begin{cases} u_m &= \frac{n_{1m}}{n_{3m}} \\ v_m &= \frac{n_{2m}}{n_{3m}} \end{cases}. \tag{21}$$

In practice, we sample the parameter space with a finite interval and relax the orthogonal criterion with a positive threshold ε yielding replace the above equation (20) with

$$\mid I_{ix}n_{1j} + I_{iy}n_{2j} + I_{it}n_{3j} \mid \leq \varepsilon. \tag{22}$$

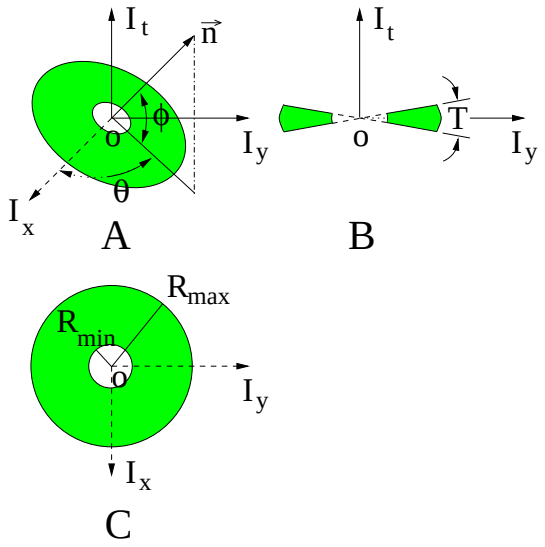

Figure 8: The 3D Hough transform is equivalent to a filter with a concave disk shape. **A:** General projection drawing of the filter mask. The vector $\mathbf{n}$ is normal to the filter mask. **B:** Side view of the filter mask. The angular thickness T of the disk is determined by the clustering threshold ε in equation (22). **C:** Vertical view of the filter mask.

The equation (22) based 3D Hough transform is equivalent to a 3D filter with a concave disk shape centered at the origin of the 3D space (figure 8). This disk is actually a collection of *relaxed* normal vectors of all possible planes containing (I_{ix}, I_{iy}, I_{it}). Comparing our filter shape (figure 2) with the shape of this disk filter we conclude that our filter samples the orientation space more efficiently than the 3D Hough transform. This conclusion is also confirmed by the comparison between our filter response of a plane (figure 6) and the Hough image of a point, which is actually the impulse response of the concave disk filter in 3D space (figure 9). It is interesting that the Hough image of a 3D point is very similar to our steerable 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 our filter response of a 3D plane consists of filter responses of a lot of points we may confirm the above conclusion easily. This efficiency enables our filter to reduce the enormous memory requirement in Hough based approaches [50], especially the gigantic overlapping of the Hough curves (figure 9). As a result, we can extract the parameters of motion planes with much less complexity.

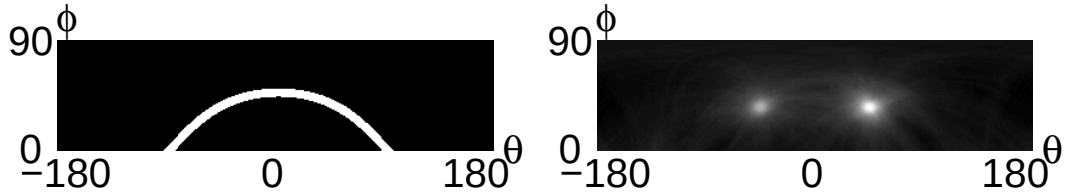

Figure 9: **Left:** All unit vectors satisfying equation (22) form a curve similar to our filter response of a 3D plane in the (θ, ϕ) space. The width of the curve is determined by the clustering threshold ε in equation (22). **Right:** The Hough image of the random dot occlusion sequence in figure 1.

Another point is that the intersections of different curves in the Hough image are blurred due to the introduction of ε, as shown in figure 9. Consequently, the global maximal position is no more a peak, but a smooth *mono-modal* distribution. Though 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 *mono-modal* distribution are unknown and we do not know how to get rid of the neighbors of the global maximum automatically in searching the second maximum. We have such an example in figure 9, where we display the Hough image of the random dot occlusion sequence shown in figure 1. After finding out the global maximal position $(\theta_{n1}, \phi_{n1}) = (46°, 36°)$ as the first normal vector and extracting the corresponding speed $(u_1, v_1) = (0.9561, 0.9901)$, we do not know how to get rid of the neighbors of this global maximal position *automatically*. In order to test if the second maximal position is correct, we cut out the neighbors of the global maximum by setting the region with $\theta > 0$ in the Hough image to zero and further search the maximal position. This time we obtain two points with $(\theta_{n2}, \phi_{n2}) = (-46°, 36°)$ and $(\theta_{n3}, \phi_{n3}) = (-45°, 36°)$. Correspondingly, the second motion has two possible parameters, $(u_2, v_2) = (0.9561, -0.9901)$ or $(u_2, v_2) = (0.9732, -0.9732)$, and we do not know which one is the desired motion parameter. This problem is even worse, when these two maxima locate near each other.

This problem is easier to solve in our filter responses. According to the description in section 3.5, we may determine the number of motions by analyzing zero-crossing points of the θ axis and obtain properly initial values by searching along ϕ direction starting from θ_m, which is the middle point of two zero-crossing points lying on the θ axis (these two zero-crossing points should have a distance of π).

The EM algorithm consists of subsequent iterations of the expectation and maximization step until there is no significant difference in the parameter estimates. Whereas the maximization step is the usual maximum-likelihood parameter estimation given the as-

signment of points to groups, the expectation step is regrouping the points by updating membership weights [28, 46]. Since the EM algorithm is an iterative method, it has no closed-form solution. Generally, we do not know the number of motions exactly. In order to fix the number of motions, Weiss introduced the smoothness constraint into the motion model [45] to avoid the "overfitting" problem. Ayer and Sawhney [3] and Gu *et al.* [20] applied the minimum description length (MDL) principle of the information theory to obtain the minimal number of motions. These approaches fix the number of motions implicitly. Besides, the convergency and the robustness of the EM algorithm are very much dependent on the initial values. According to the analysis in the last paragraph, using our filter in the EM algorithm helps to solve these two problems.

In short, though our filter does not extract motion parameters directly, it reduces the dimension of data and makes the access of parameters easier. Both properties help to improve the performance of current estimation algorithms.

5 Outlier Issue in Occlusion Estimation

In this section and in the next section, we further address two other points in occlusion analysis. Currently, the EM algorithm includes the outliers in the estimation. This makes the estimation fragile, especially if the number of outliers is comparable to the number of *normal* pixels since the EM algorithm is purely based on statistics. Our motivation is to improve the quality of input data before extracting motion parameters. According to our observation this is possible by combining current techniques.

5.1 Detection of Outliers

We assume that the motions in image sequences are piecewise-smooth with possible occlusion. In the spatio-temporal derivative space, we observe the following relations according to [27]:

- For a single constant translational motion we have a plane with a normal vector parallel to $(u, v, 1)$, where (u, v) denotes the optical flow vector. The eigenvalues of this plane satisfy

$$\sigma_1 \geq \sigma_2 > \sigma_3 = 0. \tag{23}$$

- For a single constant motion having the aperture problem, the plane above degenerates into a line whose corresponding eigenvalues satisfy

$$\sigma_1 > \sigma_2 = \sigma_3 = 0. \tag{24}$$

- For occlusion we observe multiple planes plus distortions [52] with three positive eigenvalues

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 > 0. \tag{25}$$

Thus, we can judge if there are multiple motions from different combinations of eigenvalues even *without* knowing motion parameters. In practice, the eigenvalues may deviate from their standard values due to noise or derivative approximation error. Thus, instead of checking if $\sigma_3 = 0$, we set a threshold λ_{31} for multiple motion detection. If $\sigma_3 > \lambda_{31}\sigma_1$, we conclude that there are multiple motions. We may also check the aperture problem by defining another threshold λ_{21} between σ_2 and σ_1. Here we set $\lambda_{31} = \lambda_{21} = 0.2$.

In case of occlusion, if we can purify multiple planes from outliers (i.e. distortions), we may improve the precision of estimation results. The remaining question is how to detect these outliers. We observe that if we have occlusion in a window, the occlusion boundaries should locate somewhere *inside* this window, though we do not know their exact positions. Based on this observation, we use a multi-window strategy to eliminate outliers before estimation. We detect outlier regions using small windows and mark these regions as outliers. In a large window containing these small windows, the pixels outside outlier regions are then guaranteed to be *normal* pixels. Using only these *normal* pixels for estimation, we avoid the disturbance of outliers and improve therefore the precision of estimation results in the large window.

It should be noticed that we also abandon some *normal* pixels by marking outliers with small windows. Therefore, we prefer to reduce the size of the small window so that this loss is as small as possible. On the other side, in order to provide robust eigenvalue analysis we must have an adequate number of pixels in the small window. Taking into account that the occlusion boundaries are local in every image frame and that the motions are assumed to be piecewise-smooth, we solve this conflict by limiting the spatial size of the small window, but extending its temporal size to include pixels from other frames as well (e.g. from frames (t_0-1) and (t_0+1), where t_0 denotes the current frame).

In order to verify that the *normal* pixels remaining are still adequate for estimation, we define a reliability measure which is a ratio between the number of *normal* pixels

remaining and the total number of pixels in the large window

$$r_m := \frac{\mathcal{N}_i}{\mathcal{N}_{all}}, \quad (i = 1, 2) \tag{26}$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ denote the number of remaining pixels of the occluding and occluded signal. If either of these two ratios is below a threshold, we have to enlarge the window to include more pixels for estimation.

5.2 Precision Improvement after Eliminating Outliers

In figure 10 we show the result of outlier detection in the random dot occlusion sequence (figure 1). We also show the orientation signatures before and after eliminating outliers. After eliminating outliers the curves in the (θ, ϕ) space are more clear. Consequently, we obtain better estimation results (table 1). In order to analyze the effect of window size in the estimation, we reduce the estimation window from 33×33 to 17×17. In the 17×17 window, the number of outliers is easier to be comparable to the number of *normal* pixels. As a result, the disturbance of outliers increases strongly. In contrast, if we eliminate outliers before estimation, we still can obtain reasonable results. This example displays vividly that the EM algorithm is purely based on statistics.

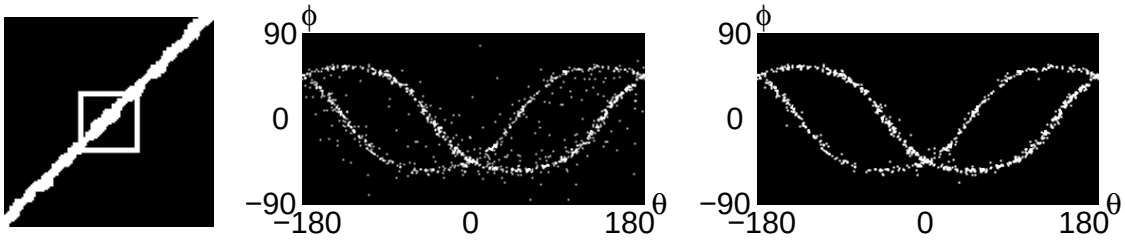

Figure 10: **Left:** Marked outliers in the random dot occlusion sequence in figure 1 after eigenvalue analysis using a $5 \times 5 \times 3$ window. The white box here shows the estimation window across the occlusion boundary. **Middle:** Orientation signature of 3D data in the (I_x, I_y, I_t) space before eliminating outliers. **Right:** Orientation signature after eliminating outliers. Two curves are more clearly to see. See tables 1, 2 and 3 for estimation results.

window size	eliminating outliers	occluding speed	occluded speed
33×33	before	$(0.986, 0.999)$	$(0.986, -0.988)$
	after	$(0.998, 0.999)$	$(0.990, -0.995)$
17×17	before	$(0.880, 0.971)$	$(0.859, -0.869)$
	after	$(0.988, 1.013)$	$(0.993, -0.998)$

Table 1: Estimation results before and after eliminating outliers with different window sizes. For comparison we apply the EM algorithm with same parameters and initial values before and after eliminating outliers: $\sigma_r = 0, 1$, $(u_{10}, v_{10}) = (0.8, 0.3)$, and $(u_{20}, v_{20}) = (1.2, -0.1)$.

6 Localization and Tracking of Occlusion Boundaries

After obtaining multiple motion parameters in the boundary regions, we would like to localize occlusion boundaries in one frame and further track their movement. Fleet *et al.* [15] modeled an occlusion boundary explicitly as an edge in a local circular mask with six parameters, i.e. four motion parameters of both occluding and occluded signals, the orientation of this boundary, and the distance between the boundary and the center of the mask. This model is only suitable for a straight-line boundary.

The spectral model of the occlusion boundary [52, 6] also assumes implicitly that the boundary is an edge (equation (3)). If the boundary has other contours, the term $U(\mathbf{x})$ in equation (3) has to be changed. Consequently, the spectrum of $U(\mathbf{x})$ changes in equation (5) as well. Since the distribution of distortions is not yet studied under this circumstance, we cannot propose an explicit model in the spectral domain to describe all possible boundaries.

Instead of using an explicit boundary model, we apply the "shift-and-subtract" technique to localize motion boundaries. The "shift-and-subtract" technique is based on the spatial coherence of the image sequence [7, 46]. Assume we have three successive frames I_{t-1}, I_t, and I_{t+1}. We first shift the frame I_{t-1} with two estimated speeds $\mathbf{v}_1$ and $\mathbf{v}_2$ to form the shifted frames $I_{t-1}(\mathbf{x} + \mathbf{v}_1)$ and $I_{t-1}(\mathbf{x} + \mathbf{v}_2)$. Then we calculate two difference images $\Delta I_{t,1}$ and $\Delta I_{t,2}$

$$\begin{cases} \Delta I_{t,1}(\mathbf{x}) &= I_t(\mathbf{x}) - I_{t-1}(\mathbf{x} + \mathbf{v}_1) \\ \Delta I_{t,2}(\mathbf{x}) &= I_t(\mathbf{x}) - I_{t-1}(\mathbf{x} + \mathbf{v}_2) \end{cases}. \tag{27}$$

If the multiple motions are occlusion, we will observe one region with zero intensity

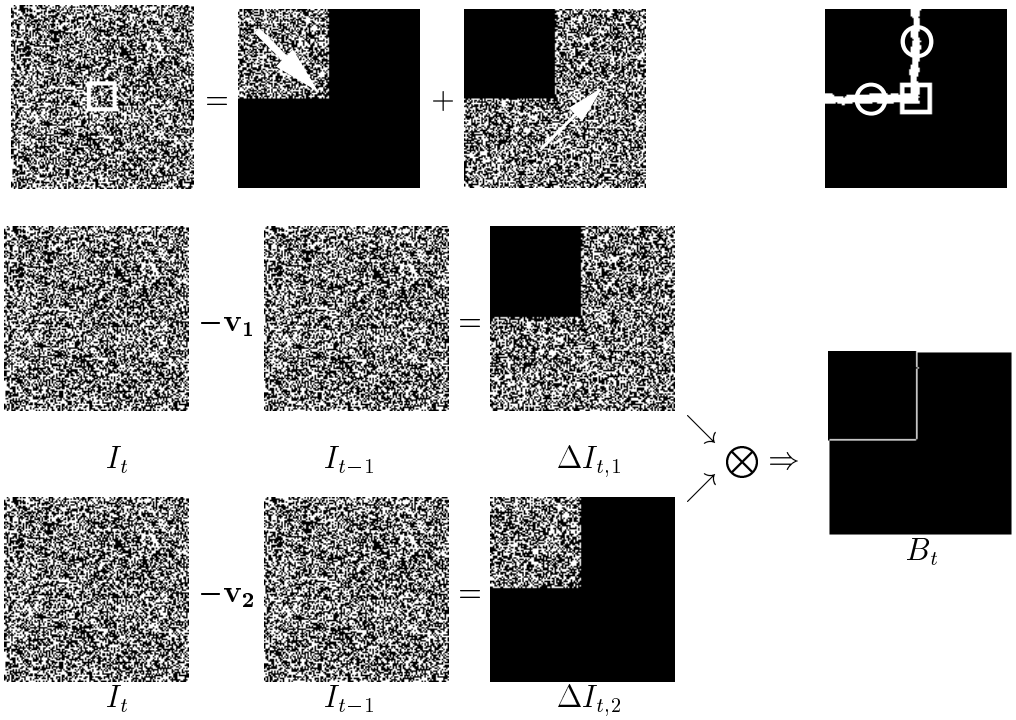

Figure 11: **Row 1 Left:** One frame from a occlusion sequence. It is composed of one occluding signal moving right-down and one occluded signal moving right-up. **Row 1 Right:** Marked occlusion regions after eigenvalue analysis. While the explicit model [15] can describe the straight-line boundary parts marked with circles, it cannot describe the boundary corner marked with the square window. **Row 2:** "Shift-and-Subtract" with the occluding speed. **Row 3:** "Shift-and-Subtract" with the occluded speed. **Between Row 2 and 3 Right:** The localized occlusion boundaries. Here we do not consider the border problem.

in each one of $\Delta I_{t,1}$ and $\Delta I_{t,2}$. These two regions are complementary in coordinates (figure. 11). Their intersection indicates the location of boundaries B_t. Thus, we extract the boundary information in a simple way without using an explicit model.

By repeating the same process on frames I_t and I_{t+1}, we obtain the shifted boundaries B_{t+1} and track therefore the movement of occlusion boundaries. Since the occlusion boundaries move consistently with the occluding signal, we solve the foreground/background ambiguity [11] as well.

The "shift-and-subtract" technique also distinguishes occlusion from transparency, as there is no zero region in either $\Delta I_{t,1}$ or $\Delta I_{t,2}$ in case of transparency. Furthermore, we

can use this technique to decompose transparency scenes into their multi-layer representations [43] (figure 14).

7 Experiments

7.1 Synthetic Occlusion Analysis

We begin with the occlusion sequence in figure 1, whose orientation signature is shown in figures 7 and 10. We apply the EM algorithm based on the orientation signature for estimation. We use the orientation signatures both with and without averaging compensation to confirm the analysis in section 3.5. In the spatial model, we further compare the estimation results before and after eliminating outliers.

In order to compare the spatial and spectral motion model, we set the same tolerance parameter $\sigma_r = 0.1$ and the same initial values in both spatial and spectral EM algorithm. In the first estimation test, we set the initial values of motion parameters arbitrarily as $(u_{10}, v_{10}) = (0.8, 0.3)$ and $(u_{20}, v_{20}) = (1.2, -0.1)$. In the second estimation test, we set initial values as $(u_{10}, v_{10}) = (0.9, 1.1)$ and $(u_{20}, v_{20}) = (0.9, -1.1)$ according to the extreme point analysis introduced in section 3.5. The results in tables 2 and 3 show that if we have properly initial values, the iteration number of the EM algorithm reduces.

In tables 2 and 3, the spatial EM algorithm provides more accurate results than the spectral EM algorithm and needs less iterations. We also confirm that the estimation results without averaging compensation are better than the results with compensation. Moreover, if we can detect and eliminate outliers before estimation, we can improve the estimation results.

In order to test the performance of the EM algorithm on determining the number of models automatically, we propose an example of one moving signal with the 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, this result is not surprising. This fact indicates that in the EM algorithm we do not know the number of motions exactly. In order 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

Occlusion Estimation with Arbitrarily Initial Values

model	outlier	averaging	iteration	occluding	occluded
spatial model	before elimination	yes	3	$(0.927, 0.998)$	$(0.949, -0.971)$
		no	3	$(0.986, 0.999)$	$(0.986, -0.988)$
	after elimination	yes	3	$(0.985, 1.002)$	$(0.977, -0.991)$
		no	3	$(0.998, 0.999)$	$(0.990, -0.995)$
spectral model	not available	yes	7	$(1.187, 1.194)$	$(1.112, -1.147)$
		no	4	$(0.898, 0.948)$	$(1.106, -1.099)$

Table 2: Estimation results of the occlusion sequence shown in figure 1. In both spatial and spectral EM algorithms we use the same parameters: $\sigma_r = 0, 1$, $(u_{10}, v_{10}) = (0.8, 0.3)$, and $(u_{20}, v_{20}) = (1.2, -0.1)$.

Occlusion Estimation with Properly Initial Values

model	outlier	averaging	iteration	occluding	occluded
spatial model	before elimination	yes	1	$(0.938, 1.005)$	$(0.923, -0.960)$
		no	1	$(0.980, 0.997)$	$(0.963, -0.974)$
	after elimination	yes	1	$(0.987, 1.002)$	$(0.967, -0.986)$
		no	1	$(0.994, 0.997)$	$(0.978, -0.988)$
spectral model	not available	yes	2	$(1.182, 1.191)$	$(1.110, -1.145)$
		no	2	$(0.966, 1.002)$	$(1.007, -1.026)$

Table 3: Occlusion estimation with properly initial values $(u_{10}, v_{10}) = (0.9, 1.1)$ and $(u_{20}, v_{20}) = (0.9, -1.1)$. The other conditions are the same as those in table 2. The estimation results in both tables are in the same precisions level since the input data does not change.

EM algorithm converges to $(1.004, -1.029)$ after 2 iterations. From this result we also confirm that the spatial model provides finer resolution than the spectral model.

7.2 Real Occlusion Analysis

In this subsection we analyze real occlusion sequences. In figure 12 we show the well known "flower garden" occlusion sequence, in which a left moving trunk covers the left moving flower bed and houses. We first estimate motions using the single motion model. At the occlusion boundaries the results are not correct, as shown in row 2 of figure 12. After the eigenvalue analysis we detect two motion candidate regions and the

regions with the aperture problem, which are shown in row 2 as well. We observe that the regions with the aperture problem are very large in the sky and along the trunk. In the two motion candidate regions we apply the spatial EM algorithm with the properly initial values to estimate multiple motions and display the results in row 3. In row 4 we apply the "shift-and-subtract" technique. Before and after the shifting, there is no difference inside the regions with the aperture problem. As a result, we observe only the boundaries of the trunk in the difference image $\Delta I_{t,1}$. In fact, the difference images $\Delta I_{t,1}$ and $\Delta I_{t,2}$ can be viewed as the result of occlusion segmentation. We further localize occlusion boundaries from $\Delta I_{t,1}$ and $\Delta I_{t,2}$ (row 4). The boundaries are badly connected since the nonzero regions in $\Delta I_{t,1}$ are discrete due to the aperture problem.

In figures 13 we have an occlusion example in which a right moving box covers a left moving picture. The image is rich in texture so that we need not to face the aperture problem. We first estimate motions using the single motion model. The incorrect results at the occlusion boundaries are clear to see. We confirm this observation using the eigenvalue analysis and mark all possible multiple motion regions in row 2. In this sequence we would like to test the effect of eliminating outliers. For performance comparison we apply the EM algorithm vertically along the occlusion boundary before and after eliminating outliers. Since we do not know the ground truth exactly, it is a little bit difficult to compare the precision of estimation results. But we observe that on each side of the boundary there is almost no speed difference among pixels. Therefore, we may use the estimation results with a large window as ground truth since there are much more *normal* pixels than *outliers* in such a large window. In the results with a small window we can observe the improvement after eliminating outliers clearly. In the window centered at $(160, 137)$ the results are not reasonable because there are only *four* pixels of the occluded signal remaining after the outliers are eliminated. This example demonstrates vividly the necessity of introducing reliability measure (equation (26)). By using the "shift-and-subtract" technique we further localize the occluding boundary which is displayed as intersection of zero regions in ΔI_1 and ΔI_2. This "shift-and-subtract" technique works also for boundaries with complex contour like the corner of the box.

7.3 Real Transparency Analysis

In figures 14 we show a real transparency sequence to compare the spatial and spectral multiple motion models. It contains a right moving portrait and a mirrored left moving muesli package. For this sequence we apply the BCCE based algorithm to estimate single motion and use the eigenvalue analysis to determine the multiple motion candidates at first. Only in these candidate regions, where the single motion model fails, we apply

the spatial and the spectral EM algorithm for motion estimation. We see that the spatial EM algorithm is not able to estimate transparent motions correctly, while the spectral EM approach works well. 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. Similarly, the multiple motion candidate regions are not in line with the package shape since some regions of the package have the aperture problem.

After obtaining the motion parameters, we further decompose the transparency scene into multi-layers with the "shift-and-subtract" technique. The results are shown in difference images $\Delta I_{t,1}$ and $\Delta I_{t,2}$.

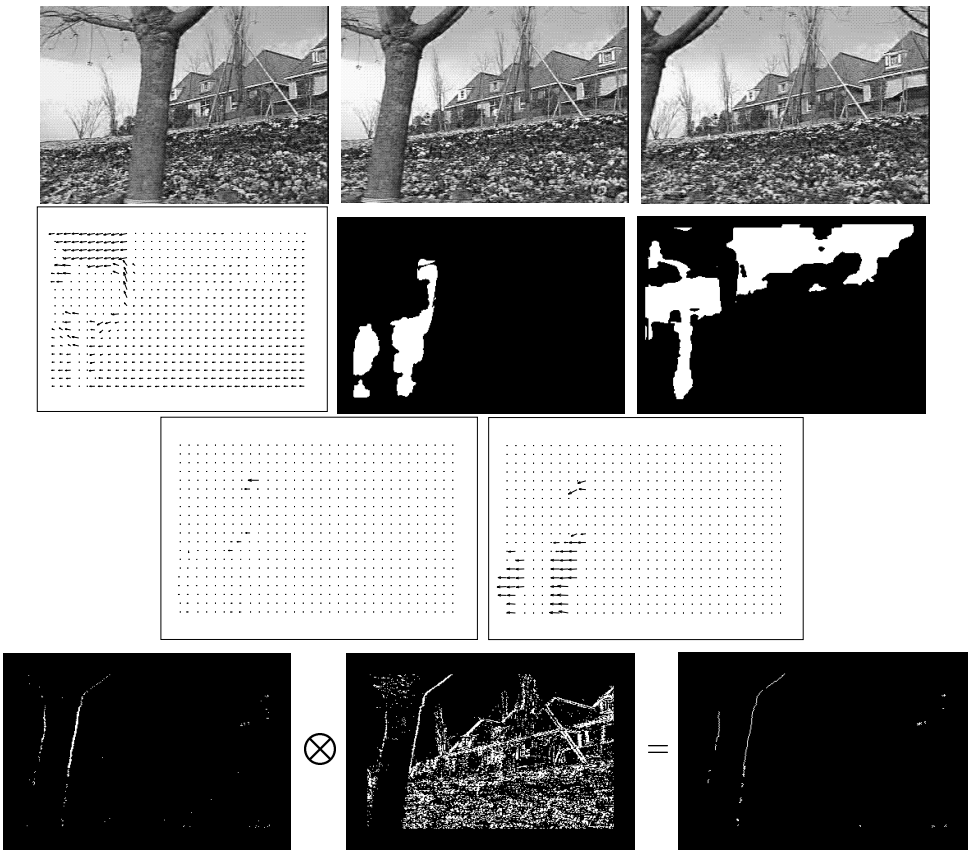

Figure 12: **Row 1:** The 17-th, 32-th and 48-th frames of the "flower garden" sequence. Each frame has 240×352 pixels. Here we consider the 32-th frame as the central frame. **Row 2 Left:** Estimation results using the single motion model. At motion boundaries the results are not correct. **Row 2 Middle:** Two motion candidate regions according to the eigenvalue analysis. **Row 2 Right:** Regions with the aperture problem. **Row 3:** Optical flow applying the spatial EM algorithm. **Row 4:** Detection and localization of motion boundaries. **Row 4 Left:** Difference image $\Delta I_{t,1}$. **Row 4 Middle:** Difference image $\Delta I_{t,2}$. **Row 4 Right:** Detected motion boundaries.

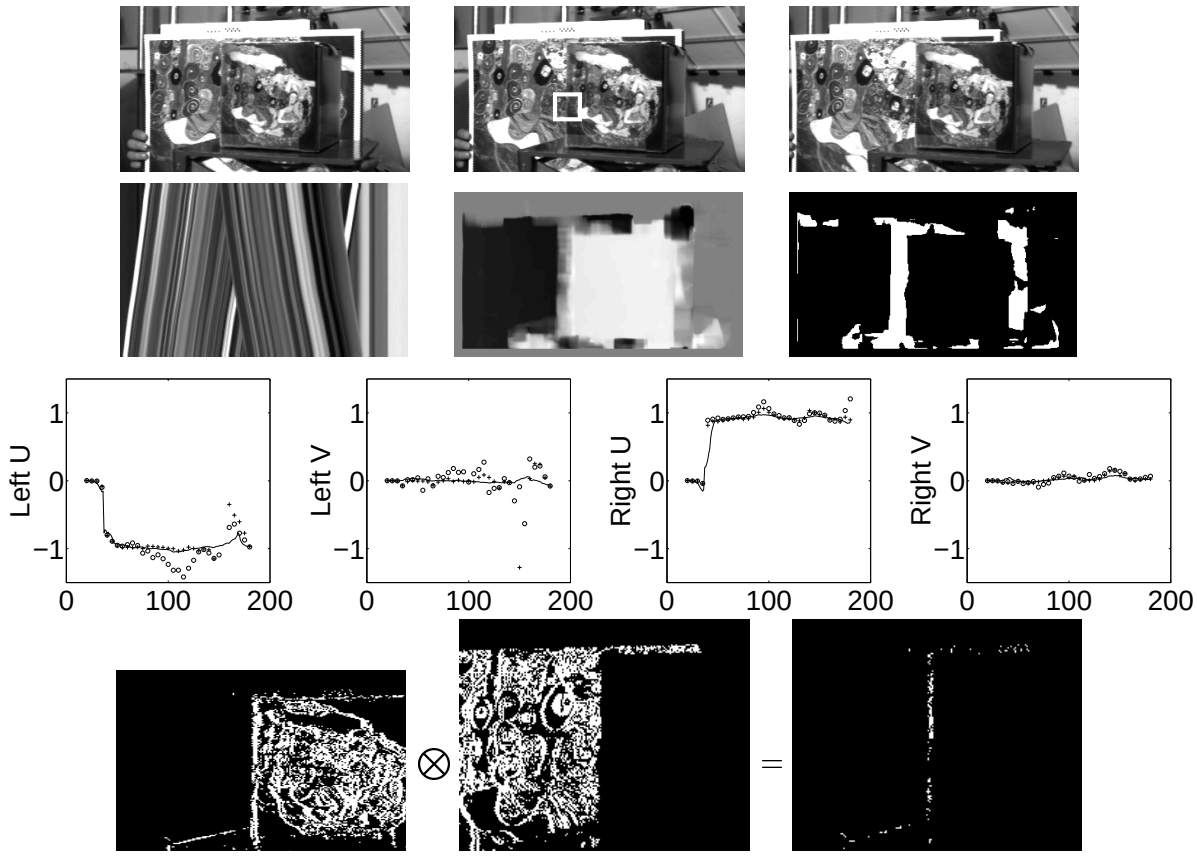

Figure 13: **Top:** The first, 16-th, and 32-th frame of an occlusion sequence. Each frame has 200×350 pixels. The white box in the 16-th frame is centered at $(122, 137)$. **Row 2 Left:** The epipolar slice of the sequence along row 122 with the first frame at the top of the slice. **Row 2 Middle:** The result of the single motion estimation algorithm. Since the vertical speed components are almost zero in this sequence, we show only horizontal speed components and use black color for negative speed (moving to the left) and white color for positive speed (moving to the right). **Row 2 Right:** Two motion candidate regions after the eigenvalue analysis. **Row 3:** Estimation results along column 137 using a 15×15 window. We use the results with a 31×31 window as the ground truth and draw them with solid lines. We draw the results before eliminating outliers with circles and the results after eliminating outliers with crosses. For comparison we draw different speed components separately. For clarity of drawing we sample the results with an interval of 5 pixels along column 137. **Row 4:** The segmentation result after "shift-and-subtract". For clarity we enlarge the occlusion boundary region.

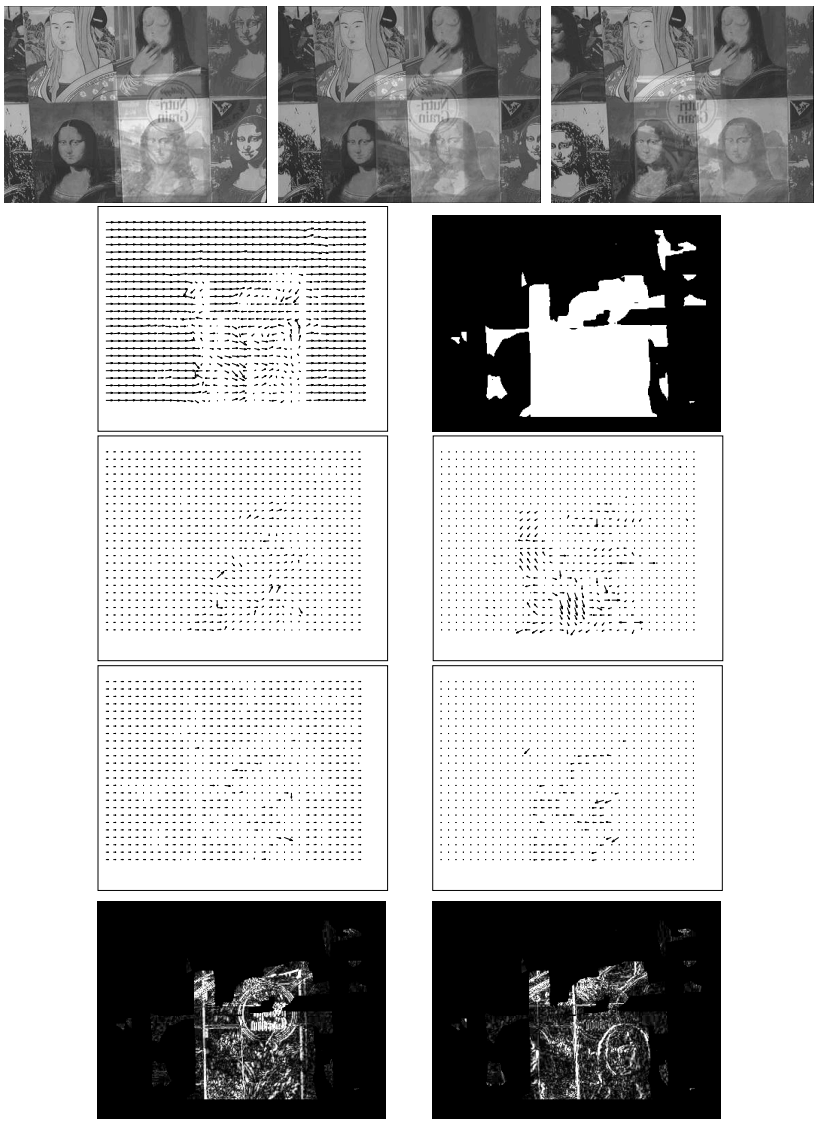

Figure 14: Comparison of spatial- and spectral-EM algorithms on real transparency sequence. **Row 1:** The first, 16-th and 32-th frames of the image sequence. Each frame has 288×384 pixels. **Row 2 Left:** Estimation results using the single motion model in the 16-th frame. **Row 2 Right:** Marked two motion candidate regions according to the eigenvalue analysis. **Row 3:** Optical flow of the spatial EM approach. The estimation results are not correct in the transparent region. **Row 4:** Optical flow of the spectral EM approach. **Bottom:** Decomposition of the transparency scene into two layers using the spectral EM results and the "shift-and-subtract" technique. **Bottom Left:** Difference image $\Delta I_{t,1}$. **Bottom Right:** Difference image $\Delta I_{t,2}$.

8 Conclusion and Discussions

In this paper, we studied multiple motion analysis from the standpoint of orientation analysis. After pointing out that multiple motions are equivalent to multiple planes in the derivative space or in the frequency domain, we proposed a new kind of 3D orientation steerable filter in motion estimation. This method is superior to principle axis analysis based approaches and current 3D steerability approaches in achieving high orientation resolution. Comparisons showed that this new method is similar to the 3D Hough transform, but more efficient and robust. Besides, it also improves the performance of the EM algorithm.

We implemented our method in the feature space directly. Though projecting the sphere surface onto 2D feature space is not an isometric mapping and the rotation symmetry is lost after projection, this transform benefits structure display and post-processing.

In occlusion estimation we further proposed to eliminate outliers in the derivative domain. Compared with current probabilistic approaches, which include the outliers in the estimation, our method improves the *quality* of input data and therefore provides more exact results.

In order to localize occlusion boundaries and to track their movement, we utilized the spatial coherence inside the frame and applied the "shift-and-subtract" technique. We did not use an explicit local model of the boundary region. But we still obtained the desired information about the occlusion boundaries. Furthermore, multiple motions can be segmented very efficiently by combining estimation techniques and spatial coherence [46]: The region with the same motion parameters can be figured out by calculating the difference between two frames with estimated speeds.

The spatial coherence information is also a key cue to distinguish occlusion and transparency in the spatial domain. Actually, it is not difficult to distinguish occlusion from transparency in the frequency domain. For example, we can look at a set of estimation results by shifting the observing window and observe their variation. Since occlusion is more local than transparency, the number of motions changes from two to one after the observing window has crossed occlusion boundaries, while in case of transparency the number of motions remains the same. We may also observe the relative ratio between data points outside the motion planes and those on the motion planes [52]. This ratio is much larger in case of occlusion than in case of transparency since all energy of the transparency lies on the dominant planes. However, in the frequency domain we cannot localize motion boundaries due to the well known uncertainty principle: The spectrum of the observing window provides us no localization information inside the

window. Therefore, we must go back to the spatial domain to detect and to localize motion boundaries, where the coherence information plays a very important role in image segmentation and scene analysis. The "shift-and-subtract" technique and the recently introduced *normalized cut* approach [39, 40] remind this point vividly.

Acknowledgment

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

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

[5] S. S. Beauchemin and J. L. Barron. A Theory of Occlusion in the Context of Optical Flow. In F. Solina, W. Kropatsch, R. Klette, and R. Bajcsy, editors, *Advances in Computer Vision*, pages 191–200. Springer Wien New-York, November, 1997.

[6] 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.

[7] J. R. Bergen, P. J. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(9):886–895, 1992.

[8] J. Bigün and G. H. Granlund. Optimal orientation detection of linear symmetry. In *Proc. Int. Conf. on Computer Vision*, pages 433–438, London, UK, June 8-11, 1987.

[9] 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.

[10] M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[11] M.J. Black and D.J. Fleet. Probabilistic detection and tracking of motion discontinuities. In *Proc. Int. Conf. on Computer Vision*, volume I, pages 551–558, Kerkyra, Greece, Sep. 20-27, 1999.

[12] M. Bober and J. Kittler. Estimation of complex multimodal motion: an approach based on robust statistics and hough transform. *Image and Vision Computing*, 12(10):661–668, 1994.

[13] P. Bouthemy. A maximum likelihood framework for determining moving edges. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11:499–511, 1989.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.

[15] D.J. Fleet, M.J. Black, and A.D. Jepson. Motion feature detection using steerable flow fields. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 274–281, Santa Barbara, CA, June 23-25, 1998.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

[23] 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.

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

[25] Chung-Lin Huang and Yng-Tsang Chen. Motion estimation method using a 3d steerable filter. *Image and Vision Computing*, 13:21–32, 1995.

[26] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12:5–16, 1994.

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

[28] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 760–761, New York, NY, June 15-17, 1993.

[29] Knutsson. *Filtering and Reconstruction in Image Processing*. PhD thesis, Department of Electrical Engineering, Linkoeping University, Linkoeping, Sweden, 1982. Dissertation No. 88.

[30] T. S. Lee. Image Representation Using 2D Gabor Wavelets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.

[31] M. Michaelis. *Low Level Image Processing Using Steerable Filters*. PhD thesis, Institute of Computer Science, University Kiel, Germany, 1995. also available as Tech. Report No. 9716.

[32] 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.

[33] H.H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:565–593, 1986.

[34] Arnold F. Nikiforov and Vasilii B. Uvarov. *Special Functions of Mathematical Physics*. Birkhaeuser Verlag, 1988.

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

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

[37] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.

[38] B. G. Schunck. Image flow segmentation and estimation by constraint line clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(10):1010–1027, 1989.

[39] J. Shi and J. Malik. Normalized cuts and image segmentation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 731–737, Puerto Rico, June 17-19, 1997.

[40] J. Shi and J. Malik. Motion segmentation using normalized cuts. In *Proc. Int. Conf. on Computer Vision*, pages 1154–1160, Bombay, India, Jan. 4-7, 1998.

[41] 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.

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

[43] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 361–366, New York, NY, June 15-17, 1993.

[44] J. Weickert and C. Schnörr. Räumlich-zeitliche Berechnung des optischen Flusses mit nichtlinearen flußabhängigen Glattheitstermen. In *DAGM Symposium Mustererkennung*, pages 317–324, Bonn, Germany, Sep. 15 - 17, 1999.

[45] 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.

[46] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 321–326, San Fransisco, CA, June 18-20, 1996.

[47] S.F. Wu and J. Kittler. A gradient-based method for general motion estimation and segmentation. *Journal of Visual Communication and Image Representation*, 4:25–38, 1993.

[48] R. P. Würtz. Object recognition robust under translations, deformations, and changes in background. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):769–774, 1997.

[49] 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.

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

[51] W. Yu. *Local Orientation Analysis in Images and Image Sequences Using Steerable Filters*. PhD thesis, Institute of Computer Science, University Kiel, Germany, 2000.

[52] W. Yu, K. Daniilidis, S. Beauchemin, and G. Sommer. Detection and characterization of multiple motion points. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume I, pages 171–177, Fort Collins, CO, June 23-25, 1999.

[53] W. Yu, K. Daniilidis, and G. Sommer. Approximate orientation steerability based on angular Gaussians. *IEEE Trans. Image Processing*, 2001. to appear.