An Asymmetrical Diffusion Framework for Junction Analysis

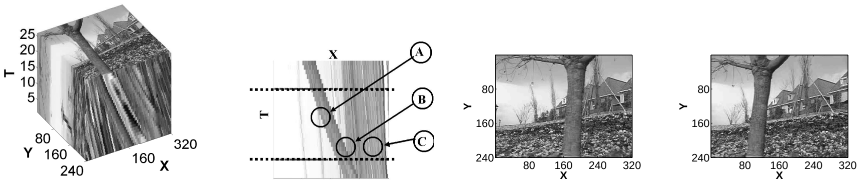
Shawn Arseneau and Jeremy R. Cooperstock
Centre for Intelligent Machines
McGill University
Montreal, Canada
arseneau,jer@cim.mcgill.ca

Abstract

The diffusion framework exhibits many promising properties for the purposes of junction analysis. In its most common form, images are diffused either isotropically or with respect to gradient information in an anisotropical fashion. This information is then collected into an orientational distribution function (ODF) and the resulting features are modeled as *'X'*, *'Y'* or *'T'*-shaped junctions. Specific to the spatio-temporal domain, and in particular, a 2D spatio-temporal slice, points of kinetic-based occlusion are identified by *T*-junctions while points of kinetic-transparency form *X*-junctions. The challenge is that most forms of diffusion are symmetric in their representation and are unable to properly distinguish between these two junction types. This work proposes to diffuse information asymmetrically and investigates the differences between weighting the iterative diffusion isotropically versus as an ODF-shaped region of influence function.

1 Introduction

Occlusion plays a pivotal role in many motion-based, machine vision tasks. From target tracking to scene segmentation, occlusion aids in the identification of motion boundaries and hence serves as a viable precursor for such applications. The challenge; however, is that occlusion is often downplayed or ignored altogether in many motion estimation / segmentation routines to facilitate model estimation: ignoring a valuable source of information [1, 9]. Occlusion detection is well defined within the spatio-temporal domain and is facilitated through the field of junction analysis, both of which will be detailed in the following section. Two competing approaches: convolution-type filtering and diffusion frameworks will be reviewed in section 2, while a novel approach designed specifically to distinguish the asymmetrical nature of occlusion patterns is proposed in section 3. This algorithm is then applied to both simulated and real-world data in section 4 and directions of future work will be reviewed in section 5.



(a) (b) (c) (d)

Figure 1: Spatio-temporal volume created by the *flower garden* sequence. The 3D volume created by stacking the images over time (a), where a slice taken from y=80 (b), indicates using the bottom, dashed line the corresponding contribution of image (c) while the top, dashed line corresponds to that of (d)

2 Background

The following section introduces the spatio-temporal domain as a reasonable representation within which to identify points of occlusion. Such points are discerned using methods from junction analysis and previous approaches are investigated that are capable of creating asymmetrical ODFs that highlight occlusion-like features.

2.1 Spatio-Temporal Domain

The spatio-temporal domain is well-suited to the detection of occlusionary events as they are depicted as the junction of contours [3, 16]. This effect is illustrated in figure 1. High gradient regions in the spatial domain give rise to contours in the spatio-temporal domain as can be seen by the slice extracted in figure 1b. These high gradient regions can be formed from either an object's motion boundary or from a textural pattern on the object itself. This is an important distinction as not all contours in the spatio-temporal domain necessarily correspond to object boundaries. The event of kinetic-based occlusion, when one object moves in front of another with respect to the camera's viewing angle, is depicted as a merging of spatio-temporal contours while disocclusion is detailed as a bifurcation or splitting of a contour into two such contours [16]. This pattern, akin to a T-junction shape, is suitably described using junction analysis. Transparency, which also involves an object moving in front of another, results in an X-shaped junction as both features are visible, albeit with a much less consistent contour, over a finite span of time. It is occlusion that will be the focus of this work as it is more consistent in its depiction with respect to the gradient magnitude in the spatio-temporal domain, and forms the more prevalent phenomenon in real-world, target tracking applications.

2.2 Junction Analysis

Within the field of junction analysis, there are two primary approaches. The first and more standard form can be categorized as convolution-based filters where a quadrature pair of kernels are convolved at either a single or multiple levels of scale. A typical example would be the commonly-used Gabor kernels [4]. The second, is to use the diffusion framework to propagate information locally to amalgamate structural information [8, 11].

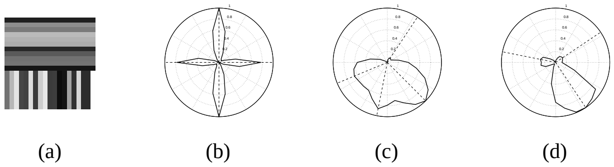


Figure 2: (a) Sample T-junction type pattern with the corresponding ODF when applying (b) Gabor with $\sigma_y = (\frac{1}{2})\sigma_x$, (c) one-sided filters and (d) the wedge filters (dashed lines denote local maxima)

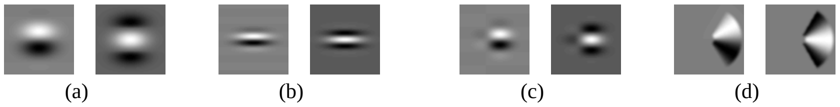
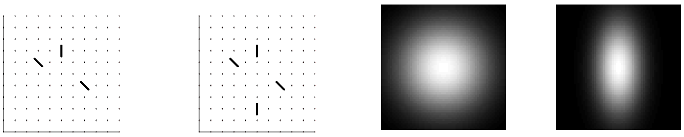


Figure 3: Quadrature pair of Gabor filters at frequency of 0.75 with (a) σ_x=σ_y and (b) $\sigma_y = (\frac{1}{2})\sigma_x$, (c) OneSided filters [12], and (d) Wedge filters [15]

2.2.1 Convolution-Based Approaches

To discern between the various types of junctions, local orientation must first be inferred. The standard approach is to populate an orientational distribution function (ODF), akin to an orientational histogram, where the saliency of the gradient measure is calculated at each of the discretized angles [15]. For example, applying a quadrature-pair of Gabor filters, results in an ODF as seen in figure 2b where the maximum response corresponds to the underlying, dominant orientation patterns from the original image. Conceptually, the original filter pair, as seen in figure 3b, are rotated with respect to the x-axis and convolved with the image data. The response at that angle is represented in polar form in figure 2b where the radius component denotes the saliency of an orientation at the given polar angle. Gabor filters are ill-suited to discriminate between T and X junctions as the filters themselves are symmetric and produce the same ODF response. To overcome this limitation, further work has been done to create asymmetrical-type, quadrature pairs of filters to address this shortcoming. Early work by Perona involved an end-point filter that was specifically targeted to identify end-points of lines/edges, while later work by Simoncelli and Farid improved on the accuracy by designing a set of polar-based Gabor kernels, known as wedge-filters [12, 15]. Yu further improved the results by building a rotated averaging wedge approach that was more robust to the effects of noise while being more discriminating at differing scale values [19]. All of the aforementioned approaches suffer from the same problem as the ODF signature that they produce does not necessarily indicate the underlying pattern sought and that they tend to describe with much information, what *diffusion* approaches have the potential of describing in a more succinct fashion [5, 7]. Also, for the Gabor, end-point and the wedge filter approaches, there are several possible parameter variations, namely the different frequency parameter choices, which must be accounted for, requiring the use of a large bank of such filters to properly account for the various possible patterns [4, 12, 15].



(a) (b) (c) (d)

Figure 4: (a) Slanted asymmetrical input (T-junction), (b) symmetrical input (X-junction), (c) isotropic and (d) anisotropic weighting function (rotated to align with vertical axis)

2.2.2 Diffusion-Based Approaches

An alternative to the local convolution-based approaches is to propagate local structural information from pixel to pixel, accumulating a local description of the surrounding pattern. Diffusion, also referred to as regularization of data, is well-suited to this task. It seeks to strike a balance between maintaining the original information, through the data consistency term, while biasing the local model using the diffusion or regularization term, consecutively shown as the two terms in Equation 1:

$$E(I) = \int_{\Omega} \left[\frac{\alpha}{2} (I - I_o)^2 + \phi(\tau) \right] d\Omega \tag{1}$$

where the ϕ-function represents the regularization term with respect to τ that typically takes some form of gradient information, α a smoothing term to balance the data consistency versus the amount of regularization desired in the output image I where I_o is the original data. Ω is the domain over which the data is integrated, such as the spatial domain when applied to images [18]. The general application domain of diffusion ranges from medical imaging to image enhancement [2,14]. There are several choices that the gradient form may take such as structure tensors or directional derivatives. To detail previous approaches, we will examine the benefits of symmetric structure tensors. To highlight the issue faced in the distinction between T and X junctions as it relates to the spatio-temporal domain, the two test cases illustrated in figures 4(a,b) are used.

The most common type of diffusion is isotropic where information is passed from the center node to the neighboring nodes as a function of relative proximity without regard to the relative direction. Often referred to as blurring in the image processing domain, this approach is adept in removing noise from an image at the expense of losing high gradient edges. For example, the test cases from figure 4 tend to slowly blur as seen from the results in figures 5. The ellipses reflect the certainty measure along the major and minor elliptical axis directions, where a stick-thin ellipse corresponds to a measure of high certainty and a circle represents high uncertainty as to the underlying gradient direction. The diffusion results, after each iteration, are represented by a single structure tensor. As expected, the results approach a case where all of the nodes are represented by the same direction and certainty. Another approach is to bias the diffusion based on the underlying gradient direction, otherwise known as anisotropic diffusion [13,18]. This approach aids in only smoothing data *along* edges rather than *across* them, hence maintaining contour information while reduce noise components in images. An example of the anistropic kernel aligned with a vertical orientation is shown in figure 4d. Applying this approach to the test cases using the single result representation, yields similar results to the isotropic case.

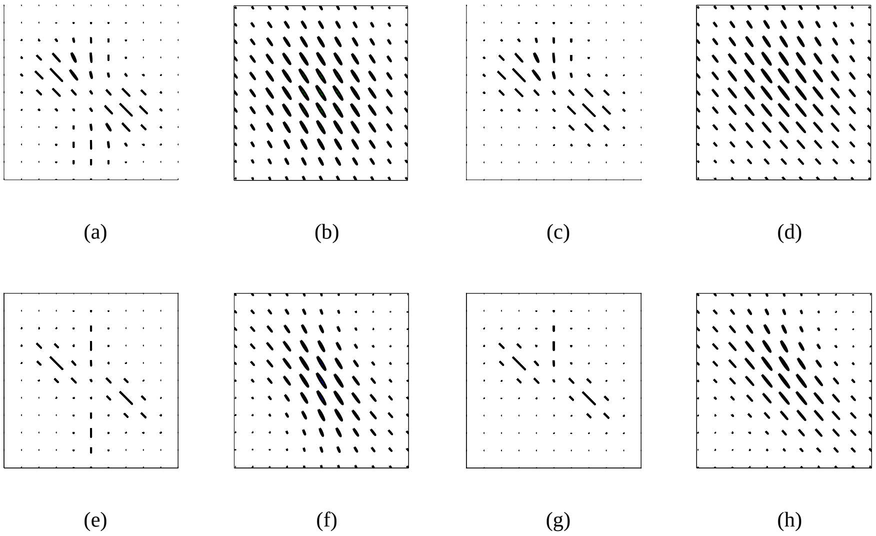
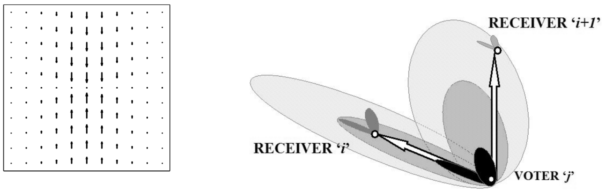


Figure 5: Isotropic results against the X-junction data after (a) one and (b) ten iterations as well as on the T-junction data after (c) one and (d) ten iterations. Results of anistropic diffusion against X-junction data after (e) one and (f) ten iterations as well as on the T-junction data after (g) one and (h) ten iterations.

It is the representation of the diffusion process as a single result after each iteration that prevents the identification of junctions. Also with these approaches, it is only possible to represent linear segments at a given node as opposed to considering that the local structure may be a curve as well. To address the latter concern, tensor voting sought to propagate data based on curvature in addition to proximity within the diffusion framework [8, 17]. The kernel, or *voting field* as it is referred to in their work, was constructed as a function of proximity between nodes, as well as biasing towards lines rather than curves through their decay function. The tensor voting technique was able to populate sparse and noisy data to extract the underlying structure with remarkable results [10, 17]. The junction detection approach was taken to be all those locations with locally uncertain (ball-shaped) gradient tensors surrounded by certain (stick-shaped) tensors [10]. Although this may be appropriate for sparse data, this approach is ill-suited for the dense structural information portrayed within the spatio-temporal domain. Also, on a more fundamental level, not all nodes with tensors of high uncertainty, surrounded by tensors with higher certainty necessarily denote a junction. To identify a junction at the node level, the representation cannot take the form of a single tensor, as a junction inherently implies *multiple* underlying orientations. Relaxation labeling addresses this concern by allowing multiple representations at a given node [6, 11]. This iterative process adds support to those nodes that tend to agree with the general model of its neighboring node. On a per-node basis, the degree of similarity between two nodes is derived through a compatibility function. Several implementations have been designed such that the compatibility function is based on co-circularity, co-heliocity as well as the normal and tangential curvature components to the vector/tensor fields [2, 11, 14]. Although impressive results are achieved in identi-



(a) (b)

Figure 6: (a) sample asymmetrical voting field aligned with vertical axis (arrowheads point towards the x-axis) and (b) example of ODF-shaped ROI

fying the perceived boundary between texture flow fields, the estimates at the node level are still symmetric in their representation thus preventing a direct distinction between the depictions of *T* and *X*-junctions.

3 Asymmetrical Voting

Combining the concepts of an asymmetrical, orientation distribution function and the diffusion framework, is most easily described in the context of a political election as does the approach of tensor voting in their description of voting fields [10]. Diffusion, at its core, seeks to spread information from a center node, which will be referred to as the voter, to the neighboring nodes, or receivers, as shown in figure 6b. The ballot, the information itself that is propagated, is comprised of two components. The first is the ballot *form*, which in the case of junction analysis, takes the form of an angular orientation. The second part is the ballot's *strength*, which is weighted by the proximity of the voter to the receiver and the relative direction between them, as in anisotropic diffusion. The final step is to propagate asymmetrical ballots from the initial, symmetric gradient information. Note that in this work, gradients are aligned tangent rather than normal to the gradient edge to facilitate clearer depictions. There are two stages to the proposed approach: the first is to transform the symmetrical information into a collection of asymmetrical (unidirectional) ballots. The next stage is the iterative process that provides the diffusion of information.

In the first stage, the initial, symmetrical ballot is transformed into an asymmetrical/unidirectional voting field. For example, for a vertically-directed gradient, the voting field would be aligned as per figure 6a. Conceptually, all of the ballots within the voting field point towards the axis perpendicular to the original gradient vector direction.

The initial gradient information at voter j is represented in polar form as (ρ_j, θ_j). The form and strength of the ballot sent from voter j to receiver i are represented as (B_{ij}^F, B_{ij}^S) respectively, and are calculated as per equations (2-4):

$$B_{ij}^F = \begin{cases} \theta_j & \left|\theta_j - \phi_{ij}\right| > \frac{\pi}{2} \\ \theta_j + \pi & otherwise \end{cases} \tag{2}$$

$$B_{ij}^S = \rho_j \cdot G_{ij}(\sigma_x) \cdot \Psi_{ij}(\tau) \cdot R(\theta_j) \tag{3}$$

$$\Psi_{ij}(\tau) = \begin{cases} 0 & -\frac{1}{2}\tau \leq x_{ij} \leq \frac{1}{2}\tau \\ 1 & otherwise \end{cases} \tag{4}$$

where τ is the minimum distance between nodes, x_{ij} the distance between nodes (i, j) along the x-axis, ϕ_{ij} is the polar angle formed from voter 'j' to receiver 'i' with respect to the x-axis, and G_{ij} is a 2D Gaussian with $\sigma_y = (\frac{1}{2})\sigma_x$ rotated with respect to the gradient direction using $R(\theta_j)$. The votes are collected at the receiver from N of its neighboring voters, and combined using a summation of 1D, periodic Gaussians shifted with respect to the ballot form, to create the ODF as follows:

$$ODF_i(\theta) = \alpha\left[\xi_i(\theta) \cdot \rho_i\right] + (1-\alpha)\left[\sum_{k=1}^{N} B_{ik}^{S} \cdot G_{2\pi}\left(B_{ik}^{F}, \sigma\right)\right] \tag{5}$$

$$\xi_i(\theta) = \begin{cases} 1 & \theta_i = \theta \\ 0 & otherwise \end{cases} \tag{6}$$

where $k \neq i$, α is the diffusion coefficient and the $G_{2\pi}$ is a periodic Gaussian, which is a 1D Gaussian sampled between [-3,3] to correspond with angular values between [0,2π)), as the ODF data, in polar-form, is also periodic. Once every node has tallied their respective collection of ballots into an ODF, the iterative process continues to diffuse the information as per the *iterative, asymmetrical diffusion* equation 7:

$$ODF_i^t(\theta) = \sum_{j=1}^{N}\left[\beta_{ij}\left(\alpha \cdot ODF_i^{t-1}(\theta)\right) + (1-\beta_{ij})\left((1-\alpha) \cdot ODF_j^{t-1}(\theta) \cdot ODF_j^{t-1}(\phi_{ij})\right)\right] \tag{7}$$

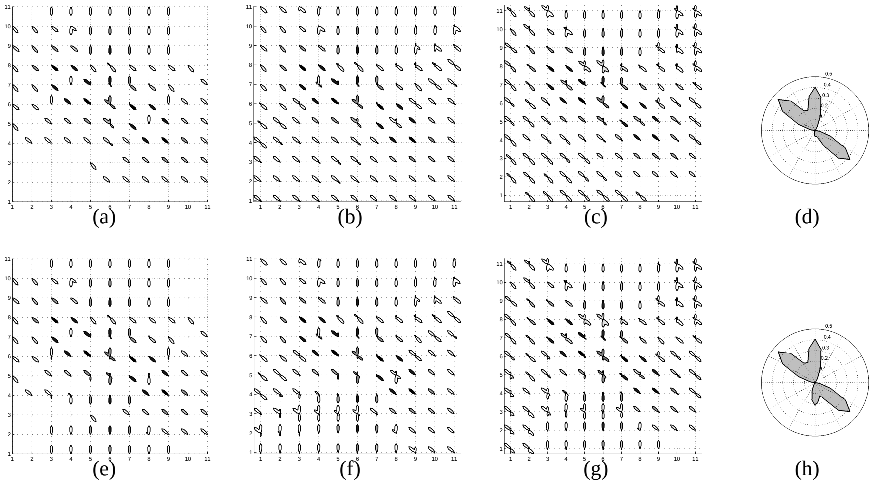
$$\beta_{ij} = \begin{cases} 1 & i = j \\ 0 & otherwise \end{cases} \tag{8}$$

where ODF_i^t is the orientational distribution function at node 'i' at iteration time = 't' and $ODF_j^{t-1}(\phi_{ij})$ is the ODF-shaped ROI term. This form of diffusion uses the ODF at a node **both** as the ballot that is being propagated to its neighbors and as the *region of influence* (ROI) function to determine the ballots' strength. An illustration of this process is shown in figure 6b where the V shaped ODF at voter 'j' is propagated to receivers $(i, i+1)$; however, with less strength to '$i+1$' as it is less aligned with the underlying ODF-shaped ROI.

4 Results and Discussion

To investigate the importance of the ODF-shaped ROI term in the asymmetrical diffusion of equation 7, two different versions of the iterative process were implemented. The first replaces this term with an isotropic, 2D Gaussian function while the second implements the ROI as per equation 7. Both the isotropic and the ODF-shaped ROI functions were applied to the test cases shown in figure 4(a,b) and are depicted in figure 7.

The first column depicts the results of the initial voting stage as per equations (2-6). The second column denotes the application of the isotropic ROI function. With this approach, the results are propagated from the voter nodes to the receiving nodes without a bias towards the underlying estimate at the voting node. Although the results are satisfactory, figure (7b,f), the ODF information becomes blurred as is expected with an isotropic



(a) (b) (c) (d)

(e) (f) (g) (h)

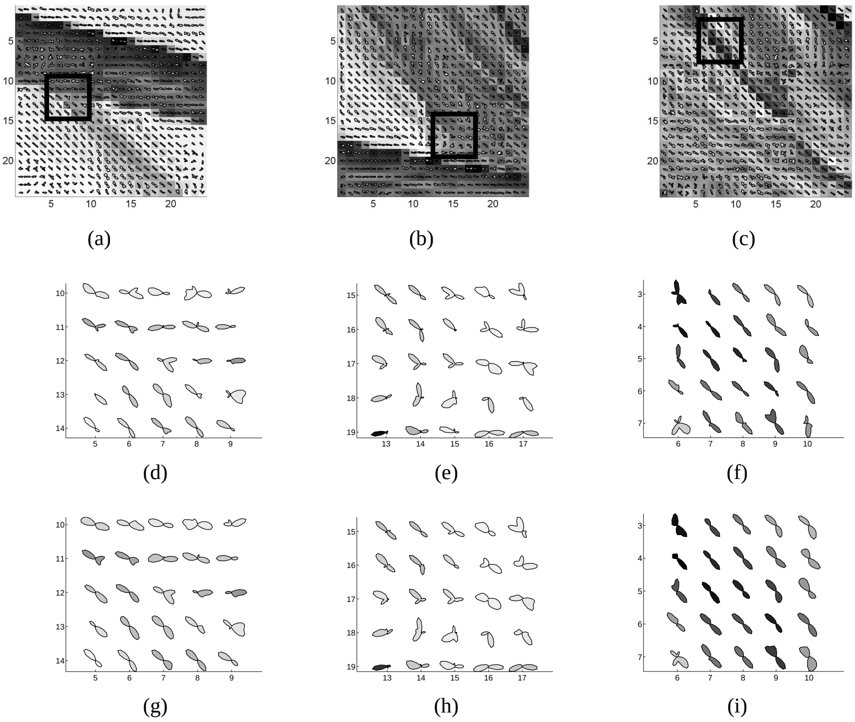
Figure 7: Top row: results against slanted T input, bottom row: against slanted X input of figure (4a-b) respectively. First column: results after initial voting stage, second column: results after ten iterations with isotropic ROI function, third column: results after ten iterations with ODF-shaped ROI function and fourth column: close-up ODFs from central position (6,6) from the third column. All node-wise ODFs have been normalized such that black-white ODFs correspond to high-low certainty values respectively.

approach. With the ODF-shaped ROI, information is diffused in a logical fashion by selectively updating the receiving nodes most likely to share common properties and are illustrated in the third column.

The second test was to apply the iterative, asymmetrical diffusion process to the spatio-temporal slice of the garden sequence in figure 1b. It should be noted that this real-world image sequence is not restricted to horizontal motion whereby junctions are solely a result of occlusion. Other phenomenon, such as vertical motion, may give rise to junctions in the slice image; however, it remains a standard benchmark in motion analysis and occlusion detection, via junction analysis, is a first stage within that process. In figure 8, results are shown for the identified locations from figure 1b. The top row of figure 8 indicates the ODFs calculated per node overlaid on the original image data, the middle row are results after one iteration, and the bottom row show results after ten iterations and illustrate how locations of junctions remain stable across iterations through its three-lobed appearance. At location ’C’ (figure 8, third column) the ODFs tend to smooth out towards standard symmetrical responses as expected. These results were obtained with the ODF-shaped ROI function, as per equation 7.

5 Conclusion

An iterative, asymmetrical diffusion process is proposed to create local, orientation distribution functions. These ODFs can be used to identify T-junctions, through their three-lobed appearance, to identify occlusion within a spatio-temporal slice or provide features



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 8: Top row: close-up of regions A,B,C from figure 1b with overlaid results for ten iterations ($t = 10$), and respective close-ups from top row (denoted by black rectangles) for $t = 1$ (middle row) and $t = 10$ (bottom row)

for detection/segmentation applications in general. Future areas of work will explore using the coherence value of the structure tensor for the variance in equation 5 to add further robustness to the process and investigate the framework within the 3D domain.

References

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

[2] O. Ben-Shahar and S. Zucker. Hue fields and color curvatures: A perceptual organization approach to color image denoising. In *IEEE Computer Vision and Pattern Recognition*, volume 2, pages 713–720, 2003.

[3] R. Bolles, H. Baker, and D. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1:7–55, 1987.

[4] D. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1(4):279–302, 1988.

[5] A. Hoogs, R. Collins, R. Kaucic, and J. Mundy. A common set of perceptual observables for grouping, figure-ground discrimination, and texture classification. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 25(4):458–474, 2003.

[6] R. Hummel and S. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 5(3):267–287, May 1983.

[7] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: Cue integration in image segmentation. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 918–925, 1999.

[8] G. Medioni, M. Lee, and C. Tang. *A Computational Framework for Feature Extraction and Segmentation*. Elsevier Science, Mar. 2000.

[9] C. Ngo, T. Pong, H. Zhang, and R. Chin. Motion characterization by temporal slices analysis. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 768–773, 2000.

[10] M. Nicolescu and G. Medioni. Motion segmentation with accurate boundaries - a tensor voting approach. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 382–389, 2003.

[11] P. Parent and S. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 11:823–839, 1989.

[12] P. Perona. Steerable-scalable kernels for edge detection and junction analysis. In *Proc. European Conference on Computer Vision*, pages 3–18, 1992.

[13] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Machine Intell.*, 33(7):629–639, 1990.

[14] P. Savadjiev, J. Campbell, G. Pike, and K. Siddiqi. 3d curve inference for diffusion mri regularization. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 123–130, 2005.

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

[16] W. Thompson. Structure-from-motion by tracking occlusion boundaries. In *Workshop on Visual Motion*, pages 201–203, 1989.

[17] W. Tong, C. Tang, P. Mordohai, and G. Medioni. First order augmentation to tensor voting for boundary inference and multiscale analysis in 3d. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 26(5):594–611, May 2004.

[18] D. Tschumperlé and R. Deriche. Diffusion pde's on vector-valued images. *IEEE Signal Processing Magazine*, pages 16–25, September 2002.

[19] W. Yu, K. Daniilidis, and G. Sommer. Rotated wedge averaging method for junction characterization. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 390–395, 1998.