Modeling trajectories of free moving objects with smooth flow fields

Mykhaylo Nykolaychuk, Christian Rössl, Holger Theisel, Klaus Richter

Fraunhofer IFF
Magdeburg Germany
{mykhaylo.nykolaychuk,klaus.richter}@iff.fraunhofer.de

Otto-von-Guericke-Universität
Magdeburg Germany
{roessl,theisel}@isg.cs.uni-magdeburg.de

ABSTRACT

We present a new method for smooth global analysis and representation of image-based motion data with topological methods for flow fields, useful for further activity analysis. A video sensor captures motion of subjects and yields discrete samples of velocity vectors in the image domain. We show how to construct a smooth, continuous, globally defined bidirectional flow field which approximates the directional part of given samples. We then apply a topological analysis of this continuous flow field which yields a segmentation into regions of similar flow behavior, i.e., regions of similar motion. This segmentation (topology) can be seen as global model of typical motions and used for further analysis like detecting atypical motions. We tested our method empirically and provide results for two different scenarios with human motion and traffic motion, respectively.

Keywords: **motion analysis, flow field topology**

1 INTRODUCTION

Motion analysis is an important research topic in computer vision with a great variety of applications. In the present scenario, our goal is to "learn" typical motion within a spatial domain that is captured by digital video. Such analysis provides the fundamentals to classify motion and to decide algorithmically whether subjects are moving in an admissible way.

There are many applications for such a tool. Examples are automatic monitoring of traffic, e.g., in a factory environment with self-directed vehicles and human workers interacting. The aim is to detect potentially dangerous situations such that action can be taken to avoid any damage or injuries.

Another example is automatic control of security areas like airports, where atypical behavior of subjects within the moving crowd is detected. It is quite easy to imagine that vector fields may provide a good tool to model such crowd movement.

In this paper, we present a method for motion analysis based on *vector field topology* which was introduced in [4]. The input is a video sequence from which trajectories of moving subjects are extracted. Figure 1 shows example scenes, arrows indicate possible motion directions. The set of all trajectories are converted into a flow field, more specifically into a tensor field as vector orientation, i.e., their sign, should not have any influence. We then regard the topology of the tensor field to segment it into regions of similar flow behavior. The resulting segmentation may serve as basis, e.g., for methods that detect atypical flow. A typical result of our method is shown in Figure 10.

The remainder of the paper is organized as follows. In Section 2 we start with previous work on the field of image-based motion detection and trajectories classification. In Section 3 we describe the data acquisition process. Section 4 presents our approach to generate a bidirectional vector field from scattered vector samples. Topology extraction is discussed in Section 5, and we show experimental results in Section 6. Section 7 concludes the paper with a summary and discussion on future work.

2 RELATED WORK

Video-based analysis systems generally have a very complex structure. They span different levels of abstraction: from the low-level detection and tracking of moving objects in video streams (trajectories) to the high-level behavior analysis (scene understanding) [16]. A common aim is to describe the observed data and to detect atypical or threatening events in real-time, i.e., to find a high-level interpretation. This becomes complicated when complex situations (i.e. scenes containing many objects and interactions) in a completely unsupervised ways are observed and evaluated.

Our work can be seen as bridge between these two levels. The input to our method are extracted trajectories, and the output is a global representation as the so-called topological skeleton of these motion data, which provides a segmentation.

Analysis of human motion and analysis of traffic are two big areas of research. We refer to surveys in [7] and [11] for detailed overview of human and traffic motion analysis respectively.

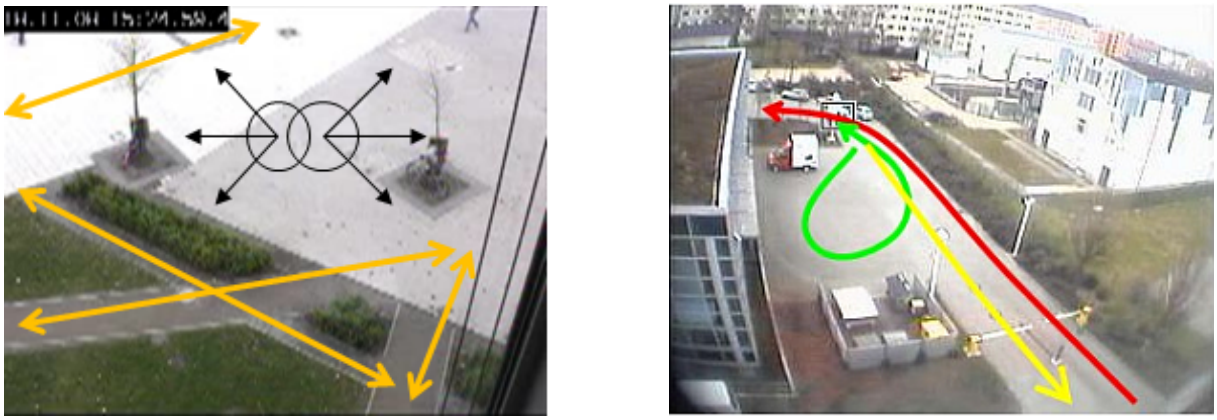

Figure 1: Motion scenes of persons (left) and vehicles (right)

Most related to our work is the approach of Hu et al. [6] which considers environments where tracking of individual objects is hard or even impossible (i.e. crowds): first, flow vectors are computed for each video frame, then after some filtering a global motion flow field is generated, and representative modes (sinks) for each motion pattern are extracted and tracked. A collective global representation of the discovered motion patterns (super tracks) is used for event-based video matching. This approach is suitable when no individual analysis of moving objects is required. In contrast our method provides potential for both, global and individual analysis.

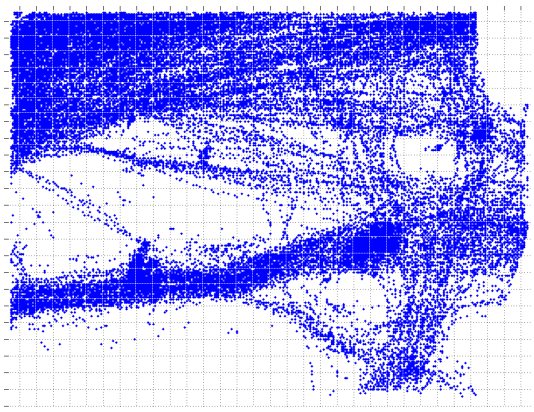

Figure 2: Raw motion data

Motion data analysis using trajectory data gained increasing interest recently [8, 10]. A lot of trajectory classification methods have been developed in the fields of pattern recognition [1] and video surveillance [14]. One important characteristic of those methods are that they use the shapes of whole trajectories to do classification. For example such as the hidden Markov model (HMM) [1] by modeling a whole trajectory with a single mathematical function.

Many approaches employ neural networks such as self-organizing maps (SOM) and use whole trajectories for classification. For instance in [14], each trajectory is encoded to a feature vector using its summary information (e.g., the maximum speed). But some valuable information (e.g., time, motion direction, etc.) could be lost due to this encoding.

Moving-object anomaly detection is a hot topic of research and very closely related to trajectory classification. Li et al. [10] propose an anomaly detection method based on motifs (i.e., trajectory features). This method is limited to extracting motifs without taking other attributes into account.

Morris and Trivedi [12] learn topology scene descriptors (POI) and modeled the activities among POIs with HMMs. The approach is suitable to detect abnormal activities and shows good performance when used in structured scenes.

Pusiol et al. [17] propose a method to learn scene logical regions (scene topology) in an unsupervised way. Topology of the scene is learned using the regions where the person usually stands and stops, so-called slow regions. Transitions among these slow regions are learned as primitive events from which histograms are generated to recognize activities of other subject.

Generation of semantic-based trajectories can be used to simulate real-world behavior. Pfoser and Theodoridis [15] present several examples of how real-world movement characteristics can be described with appropriate semantics. While we do not aim at a semantic analysis in this work, we are confident that our resulting global model of motion may serve as basis for such analysis.

Another approach to classify trajectories is clustering (see, e.g., [8]). However, significant features are likely to appear only locally at parts of trajectories, they do not characterize trajectories globally as a whole. Also discriminative features appear not only as common movement patterns but also as regions.

In contrast to this, we propose a different approach based on a *global* model: significant features are extracted as *topology* of motion data which are represented by a smooth flow field.

3 DATA ACQUISITION

We use a video camera in combination with optical flow to extract motion data from a video sequence. We chose an off-the-shelf video sensor Vitracom SiteView EP for this purpose.

In our setup the sensor is fixed and captures a certain region (see Figure 1). For any detected motion of any subject we are provided with a sequence of a subject's id, its positions in image-space and a time-stamp. These data can be used to construct trajectories of individual subjects. All our computations are performed in image-space which is sufficient for our purpose as in this work we do not combine data from multiple sensors.

Due to the complexity of the scene the acquired data suffers from disturbing effects like distortion, overlapping, occlusion, or fusion of moving objects. In order to extract reliable raw data, we apply a smoothing step on the individual trajectories. The rationale for this preprocess consists firstly in the fact that filtering in this way respects the temporal and spatial coherence as motions are processed individually. Secondly, we can rely on a simple and efficient finite differences scheme for curve smoothing rather than smoothing scattered vector data. In order to achieve this, we minimize a certain bending energy for trajectory smoothing.

Finally, we overlay velocity samples from all trajectories within the image domain. The result is a set of vector samples scattered in the domain. Figure 2 shows such raw data.

4 VECTOR FIELD GENERATION

While the scattered vector samples describe our vector field, we cannot apply them directly for topology extraction and segmentation. We need to generate a suitable continuous parametric model from the given discrete set of data points. A standard approach is least-squares approximation of a bivariate function, e.g., tensor-product B-spline.

Unfortunately, this approach does not work in our setting: our goal is fitting a *bidirectional* vector field (or a tensor field), i.e., we want to fit accurately the direction of velocity vectors but not their orientation. This *orientation-invariance* cannot be modeled by a linear least-squares fit.

In the following, we provide a different formulation of the problem which leads to solving an eigenvalue problem.

Given are samples of velocity vectors $\mathbf{v}_i = (u_i, v_i)^\top \in \mathbb{R}^2, i = 1, \ldots, m$ at points $(x_i, y_i)^\top$ in the image plane. We want to find a parametric vector field

$$\mathbf{w}(x,y) = \sum_{j=1}^{n} b_j(x,y)\,(U_j, V_j)^\top \;, \qquad (1)$$

such that $\mathbf{w}(x_i, y_i)$ is as parallel as possible to $\mathbf{v}_i$. Here, $b_j(\cdot)$ represent any suitable bivariate basis functions, e.g., tensor-product B-splines. And we have to determine the unknown coefficients $U_j, V_j,\ j = 1 \ldots, n$.

Note that we are only interested in matching the direction of vectors and disregard both, orientation and magnitude. The first property is essential to our approach, while the latter one can easily be restored in a second step by a linear least-squares fit of a scalar magnitude field.

4.1 Eigenvalue problem

In order to model the problem as a minimization, we reformulate it: instead of $\mathbf{w}$ and $\mathbf{v}$ being parallel, we require that $\mathbf{w}(x_i, y_i)$ are as *orthogonal* to $\mathbf{v}_i^\perp =: (-v_i, u_i)^\top$ as possible, where $(\cdot)^\perp$ denotes a rotation by $\frac{\pi}{2}$. For the best fitting bidirectional vector field we have

$$E = \sum_{i=1}^{m} \left(\mathbf{w}(x_i, y_i)^\top \mathbf{v}_i^\perp \right)^2 \to \min \;, \qquad (2)$$

i.e., we penalize non-orthogonality. We rewrite (2) in matrix notation such that

$$E = \mathbf{r}^\top \mathbf{r} \;.$$

Then we have each element of $\mathbf{r} \in \mathbb{R}^m$ as

$$\begin{aligned} \mathbf{r}_i &= \mathbf{w}(x_i, y_i)^\top \mathbf{v}_i^\perp \\ &= -v_i\, \mathbf{b}(x_i, y_i)^\top \mathbf{U} + u_i\, \mathbf{b}(x_i, y_j)^\top \mathbf{V} \\ &= \left(u_i \mathbf{b}(x_i, y_i)^\top \right) \mathbf{V} - \left(v_i \mathbf{b}(x_i, y_i)^\top \right) \mathbf{U} \;. \end{aligned}$$

Here, we write the two components of $\mathbf{w}(x,y)$ in (1) as scalar products $\mathbf{b}(x_i, y_i)^\top \mathbf{U}$ and $\mathbf{b}(x_i, y_j)^\top \mathbf{V}$, respectively, with a vector $\mathbf{b} = (b_1(x,y), \ldots, b_n(x,y))^\top$ of basis functions and coefficient vectors $\mathbf{U} = (U_1, \ldots, U_n)^\top$ and $\mathbf{V} = (V_1, \ldots, V_n)^\top$.

Then $\mathbf{r}$ is expressed as

$$\mathbf{r} = \mathbf{C}\mathbf{V} + \mathbf{D}\mathbf{U} = (\mathbf{C}, \mathbf{D})\,(\mathbf{U}, \mathbf{V})^\top \;,$$

and the definition of the matrices $\mathbf{C}, \mathbf{D} \in \mathbb{R}^{m \times n}$ is given by the element-wise notation above.

For the quadratic energy term $E = \mathbf{r}^\top \mathbf{r}$ we obtain

$$\begin{aligned} E &= (\mathbf{V}^\top, \mathbf{U}^\top) \begin{pmatrix} \mathbf{C}^\top \mathbf{C} & \mathbf{C}^\top \mathbf{D} \\ \mathbf{D}^\top \mathbf{C} & \mathbf{D}^\top \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \mathbf{U} \end{pmatrix} \\ &=: \mathbf{x}^\top \mathbf{A}\, \mathbf{x}^\top \;. \end{aligned}$$

It is obvious that the global minimum $E = 0$ is attained at $\mathbf{x}^\top = (\mathbf{V}^\top, \mathbf{U}^\top) = \mathbf{0}$. This trivial solution, however, is not the one we are looking for. By adding the additional constraint $||\mathbf{x}||^2 = 1$ we avoid the trivial solution.

(Note that we are interested only in directions of vectors and not in their magnitude, so there is an infinite number of feasible solutions which differ only in scale. Hence, we may restrict the above norm of $\mathbf{x}$ to any positive number.)

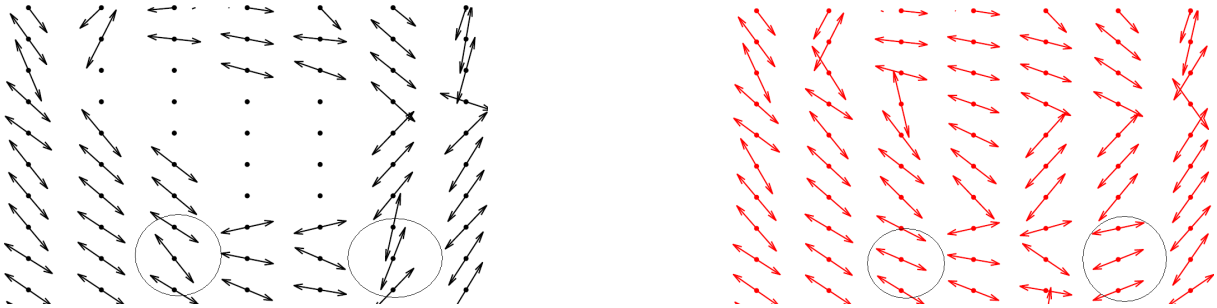

Figure 3: Vector field without (left) and with (right) regularization.

Introducing the Lagrange multiplier λ the minimization problem now reads as

$$E' = \mathbf{x}^\top \mathbf{A}\,\mathbf{x}^\top - \lambda(\mathbf{x}^\top \mathbf{x} - 1) \to \min$$

The matrix $\mathbf{A}$ is symmetric and positive definite, and we find the minimum by setting the gradient $\nabla E' = 0$, which yields

$$\mathbf{A}\mathbf{x} - \lambda \mathbf{x} = 0\,,$$

and which in turn means that the minimum is attained for an *eigenvector* of $\mathbf{A}$. In fact, we are looking for the eigenvector corresponding to the smallest eigenvalue.

There are efficient numerical methods for solving the eigenvalue problem, i.e., for computing the smallest eigenvalue and hence a single eigenvector.

While the dimensions of the matrix $\mathbf{A} \in \mathbb{R}^{2m\times 2m}$ can be rather large — on a $N \times N$ grid we have $m = N^2$ — $\mathbf{A}$ is sparse if the basis functions $b_j(x,y)$ have local support which is the case for B-splines. State-of-the art algorithms exploit the sparsity pattern of the system matrix [9].

In summary, we compute a parametric vector field $\mathbf{w}(x,y)$ from discrete samples $\mathbf{v}_i$ by solving an eigenvalue problem. The solution satisfies the condition that $\mathbf{w}$ is as parallel as possible to vectors $\mathbf{v}_i$ in least-squares sense.

4.2 Regularization

There may be relatively large parts of the domain, which do not contain any vector samples. We are aware of the fact that the approximation method described above may not be able to "interpolate" meaningful vectors across such regions. (Instead, the solution to the eigenvalue problem yields arbitrary vectors of very small magnitude, see Figure 3 (left).)

This is a well-known effect that depends on the particular choice of basis functions: roughly speaking, one cannot expect a contribution from basis functions without having enough samples within their support. For B-splines, this means the Schoenberg-Whitney conditions should be satisfied, which is generally not the case in our setup (see, e.g., [5]) The standard approach to solve this problem is adding a regularization term, i.e., taking into account additional constraints usually on smoothness. We penalize the norm of first order partials. The eigenvalue problem then reads as

$$(\mathbf{A} + \alpha \mathbf{R}^\top \mathbf{R})\mathbf{x} - \lambda \mathbf{x} = 0\,,$$

where $\mathbf{R}^\top \mathbf{R}$ captures the regularization term, and $\alpha > 0$ is a small weight. Figure 3 shows the effect of regularization.

4.3 Velocity bias

The formulation as is effectively yields a weighted least-squares approximation: (2) does not strictly penalize angles but includes the magnitude of the samples. For our purpose, such weighting is neither meaningful nor does it provide better results. There is no reason, why faster movement should influence the fitting of the directional component. For this reason, we normalize all samples such that $||\mathbf{v}_i|| = 1, i = 1,\ldots,m$. Note that this does not interfere with recovering speed as a scalar value: this would be second step resulting in least-squares fitting a scalar field. As we are interested in vector field topology, speed is currently not of interest to us.

4.4 Discussion of our setting

The formulation of the problem allows for an arbitrary choice of basis functions b_j. A natural and well-established choice are tensor product B-splines. Higher polynomial degree q yields higher order smoothness, i.e., C^{q-1} continuity, of the bidirectional flow field $\mathbf{w}$ as well as higher order approximation. In our experiments with $q = 1,2,3$ we found that in fact the simplest choice $q = 1$, i.e., a bi-linear function $\mathbf{w}$ shows sufficiently good results. At the same time, this is the most efficient model to evaluate. In some situations we even observed that higher order smoothness does not capture "turbulent" regions of the sample set as well, because due to larger support, more samples with potentially diverging directions are taken into account.

Regions without any data, i.e., regions with no observed motion, should be masked out for our application, they provide trivial segments. Regularization generally yields plausible data for such regions and improves vector field near these segment boundaries.

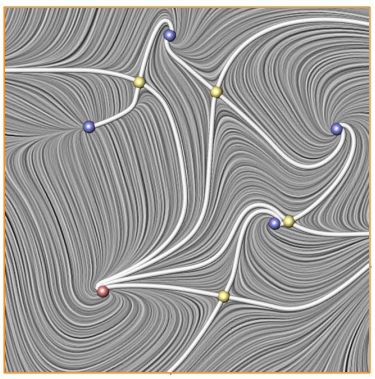

Figure 4: Example of a topological skeleton of a vector field

However, we found that for our purpose the difference in using or not using a regularization term is not significant.

5 TOPOLOGY EXTRACTION

In this section we show how to extract the topological skeleton of the smooth bidirectional flow field $\mathbf{w}$. This skeleton provides the segmentation of the image domain in regions of similar motion.

5.1 Topological skeleton

One of the most important features of a vector field is its *topological skeleton* which has been introduced as a visualization tool in [4]. The topological skeleton of a 2d vector field $\mathbf{v}(x,y)$ essentially consists of a collection of *critical points*, i.e., $\mathbf{v}(x,y) = \mathbf{0}$ and special stream lines called *separatrices* which separate the flow into areas of different flow behavior.

The attractiveness of the topological skeleton as a visualization tool lies in the fact that even a complex flow behavior can be expressed (and visualized) by using only a limited number of graphical primitives. At the same time, the topological skeleton yields a segmentation of the domain.

Helman and Hesselink [4], consider first order critical points, i.e. critical points with a non-vanishing Jacobian. Based on an eigenvector analysis of the Jacobian matrix, these critical points are classified into *sources*, *sinks*, *centers* and *saddles*. For the description of the topology, saddle points are of particular interest. In addition so called *boundary switch points* [2] may separate regions of different inflow/outflow behavior across the boundary of the flow.

Figure 4 shows an example of a vector field together with the topological skeleton. For an excellent introduction to vector field topology and related computational methods we refer to [19].

5.2 Extraction of the topological skeleton

For extracting the topological skeleton, we proceed in two steps. First, we find critical points. Then we integrate streamlines staring from critical points to construct separatrices and hence segment boundaries.

Finding critical points.

We consider a bidirectional flow which is essentially a 2nd order symmetric *tensor field* $\mathbf{T}(x,y)$ where

$$\mathbf{T} = \begin{pmatrix} w_1(x,y)^2 & w_1(x,y)\,w_2(x,y) \\ w_1(x,y)\,w_2(x,y) & w_2(x,y,)^2 \end{pmatrix},$$

with $\mathbf{w}(x,y) = (w_1(x,y), w_2(x,y))^\top$.

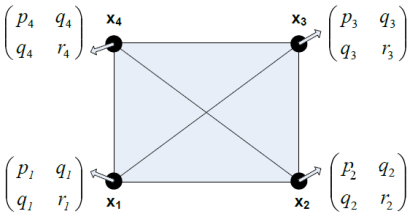

Figure 5: Cell subdivision into triangles

Then critical points (or *degenerate points*) are points (x_c, y_c) in the domain where the eigenvalues λ_1 and λ_2 of $\mathbf{T}(x_c, y_c)$ are equal and hence we have

$$\mathbf{T}(x_c,y_c) = \mathbf{Q}\begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\mathbf{Q}^\top = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix},$$

where $\lambda = \lambda_1 = \lambda_2$ [3, 18]. This means, in degenerate points *any* vector is an eigenvector to $\mathbf{T}$. This is valid in any coordinate system $\mathbf{Q}^\top$.

Such points are characterized by the two conditions

$$\begin{cases} \mathbf{T}_{11}(x_c,y_c) - \mathbf{T}_{22}(x_c,y_c) = 0 \\ \mathbf{T}_{12}(x_c,y_c) = 0 \end{cases}$$

Finding critical points corresponds to finding roots of the two components of the above equations.

Finally, critical points are classified in *wedges* or *trisectors* based on certain partials:

$$a = \frac{\partial}{\partial x}(\mathbf{T}_{11} - \mathbf{T}_{22}),\quad b = \frac{\partial}{\partial y}(\mathbf{T}_{11} - \mathbf{T}_{22}),$$
$$c = \frac{\partial}{\partial x}\mathbf{T}_{12},\quad d = \frac{\partial}{\partial y}\mathbf{T}_{12},$$

(see [3]), with $\delta = ab - cd$, for a wedge $\delta > 0$ and for a trisector: $\delta < 0$.

The detection of critical points consists essentially of a numerical root finding algorithm. As we are not interested in finding higher order critical points, we simplify the setting such that elements of the tensor field $\mathbf{T}$ are interpolated independently.

Even more, we prefer linear interpolation to bi-linear interpolation within grid cells (or generally higher order interpolation). For linear interpolation we subdivide each quad cell uniformly into either two (inserting

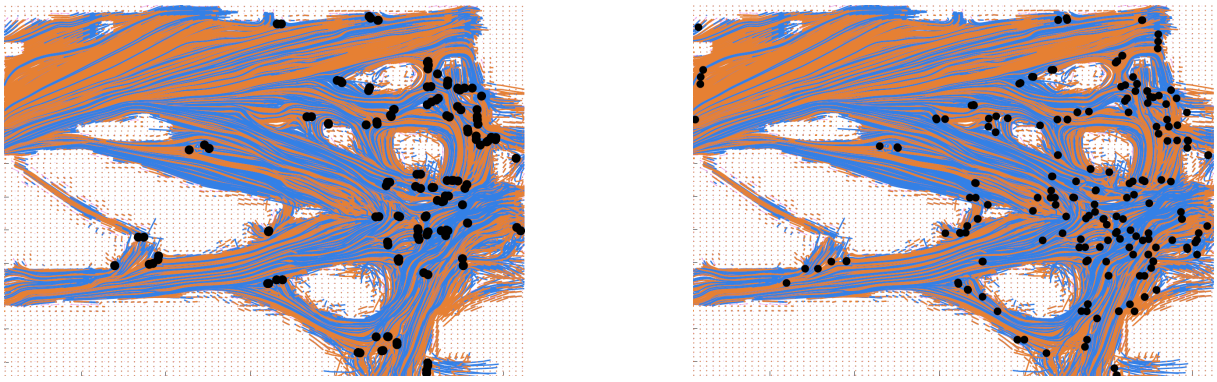

Figure 6: Detected degenerate points for linear (left) and bilinear (right) interpolation.

the diagonal) or four (inserting the quad's barycenter, Figure 5) triangles and formulate the above condition in terms of barycentric coordinates. This is due to two reasons: firstly, we have to consider a lower polynomial degree for root finding, which makes this task more efficient. And secondly, the number of detected critical points decreases, and we find exactly one or no (first order) critical point within each linear piece.

Figure 6 compares amount and distribution of critical points for linear and bi-linear interpolation. In this example, we detect 62 and 145 degenerate points for linear interpolation splitting each cell into two or four (as shown in the Figure 5) triangles, respectively. Bilinear interpolation yields 228 degenerate points. For our results we use linear interpolation and $1:4$ subdivision of cells as this constitutes a good compromise between efficiency and finding all relevant points.

Finding separatrices.

Separatrices are streamlines which emanate from degenerate points in two (wedge) or three (trisector) directions. We use a 4th order Runge-Kutta method for numerical integration on $\mathbf{w}$. The standard algorithm for vector field integration is modified such that the orientation is chosen at the starting point and kept consistent during integration.

6 RESULTS

We tested our approach with two real-world data sets where motion of vehicles and motion of human subjects were captured, respectively (see also [13]). The video sensor is fixed for both setups and provides a perspective view of the scenes (Figure 8 and 10, right).

The vehicle data was acquired over a 24 hour period. Vehicle motion is more homogeneous in this case. Humans were captured in the area where many people walk in different directions over a 30 minutes time interval.

The motion data is rather complex: there are several obstacles, and persons can move freely around them, in general. However, we expect most motion along several paths though the region. Computations were performed on 56×56 (vehicles) and 53×53 (human subjects) grids, respectively. Our current, non-optimized implementation requires approximately 5 seconds for computation of each, the flow field and the critical points. Integration of separatrices takes about 20 seconds.

Figure 7 (left) shows the raw data of vehicle movements as acquired from the sensor. The least-squares approximated bidirectional vector field is shown in Figure 7 (right). There are homogeneous regions in the scattered raw data, such as the region near the gateway, which results in homogeneous vector field regions. Regions of less homogeneous motion, e.g, where particular trajectories intersected, are less homogeneous in the bidirectional vector field. Regions without motion data yield zero vectors in the vector field. This shows effectiveness of the bidirectional vector field model and its generation.

Figure 8 visualizes the extracted tensor field topology for the moving vehicles on the grid (left) and as an overlay on the perspective image (right). Regions of similar motion are well-separated, and this example shows nicely that the topological skeleton (black) conveys the properties of the vector field very well with only a limited number of primitives. There are rather smooth regions where vehicles enter the scene at the gateway. Fundamental motions such as vehicles turning are present. There are also several regions with high amount of degenerate points: in fact, these regions represent critical regions within the domain. The white circle marks an area with a barrier that lets the cars in and out. The region near the trash cans marked by the black circle is highly frequented by both, vehicles and human subjects.

The motion of human subjects shown in Figure 9 is more complex, hence the resulting vector field is less homogeneous. Figure 10 visualizes topology of the motion and makes it straightforward to detect regions with no motion, homogeneous and less homogeneous motion. The white circle marks a region where different directions of motion intersect, and hence this region shows a rather complex behavior. Note that such arbi-

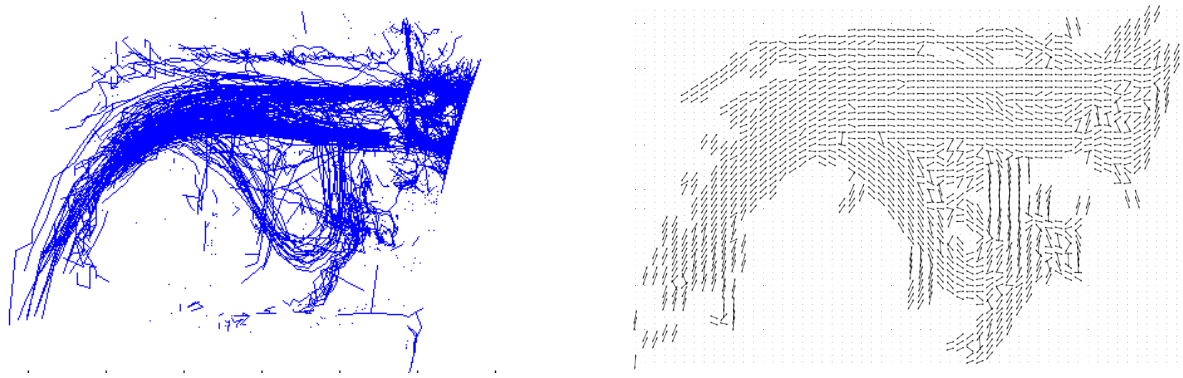

Figure 7: Raw data (left) and generated vector field (right)

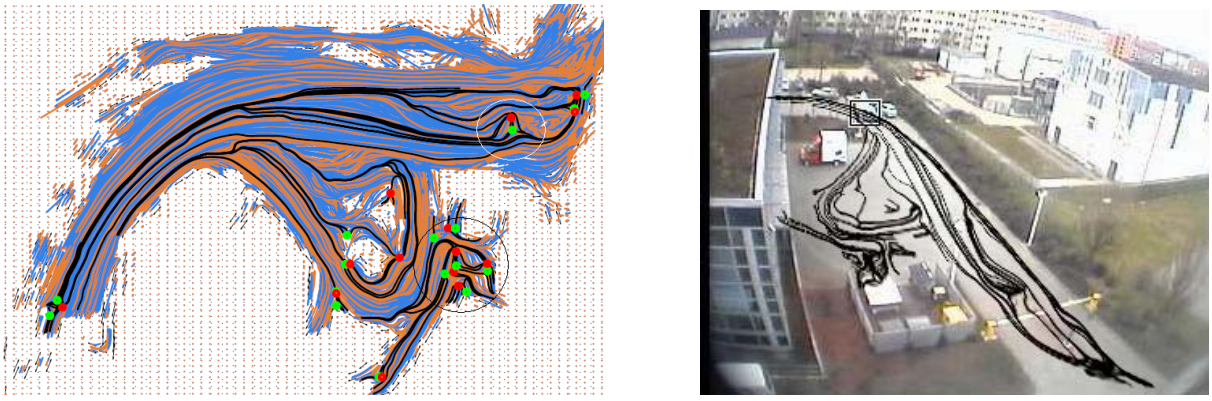

Figure 8: Topology of vehicles motion trajectories

trary intersections cannot be represented accurately by the underlying tensor field. In fact the whole region should be represented by a single (higher order) critical point. Such interpretation of the results and classification is left for future work.

In summary, we observe that the topological skeleton provides a meaningful segmentation of motion within the scene and give a global representation of the monitored area.

7 CONCLUSIONS

We demonstrated how methods for analyzing flow fields can be applied to analysis of motion data. In order to achieve this, we first approximate scattered velocity data with a smooth bidirectional vector field, i.e., essentially with a tensor field. Then we extract its topological structure for segmentation of the domain into regions of similar motion. Topological skeleton is an appropriate global representation of typical motions in a monitored area. Our results prove the efficiency of this approach.

The output of our method may serve as a basis for further analysis. Assume that the extracted topological skeleton reflects a typical or normal condition for the observed region. The change of the skeleton, i.e. the deviation from normal movements can then be quickly identified. Another application of our method is classification of individual movements or extraction of basic movements. This requires additional object tracking.

For future work we plan to extract fundamental motions of our data which enables on-the-fly classification of trajectories in categories of similar motion. At the same time newly acquired motion should be integrated in the current segmentation, i.e. real-time update of topological skeleton. Conversely, such classification could be used to fit flow fields for every motion category which in turn can be used to improve the segmentation. It might be useful to simplify and/or smooth the topological skeletons, especially when data from multiple sensors or data from a moving sensor are combined.

An ultimate goal is to infer specific actions from motion data in order to identify non-admissible actions in any kind of security area.

ACKNOWLEDGMENTS

The authors are partially funded by the German Federal Ministry of Eductaion and Research (BMBF) within the ViERforES project (no. 01IM08003C).

REFERENCES

[1] F.I. Bashir, A.A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *IP*, 16(7):1912–1919, July 2007.

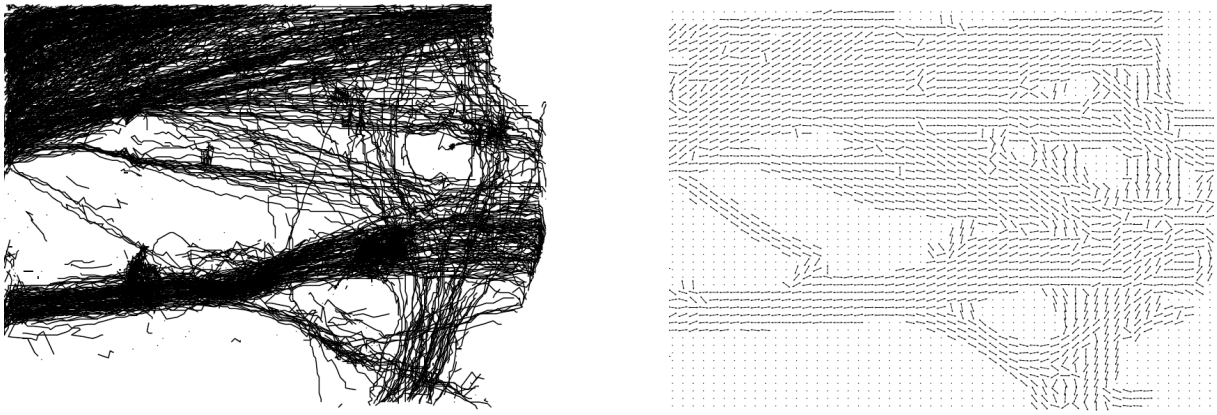

Figure 9: Raw motion trajectories and estimated vector field

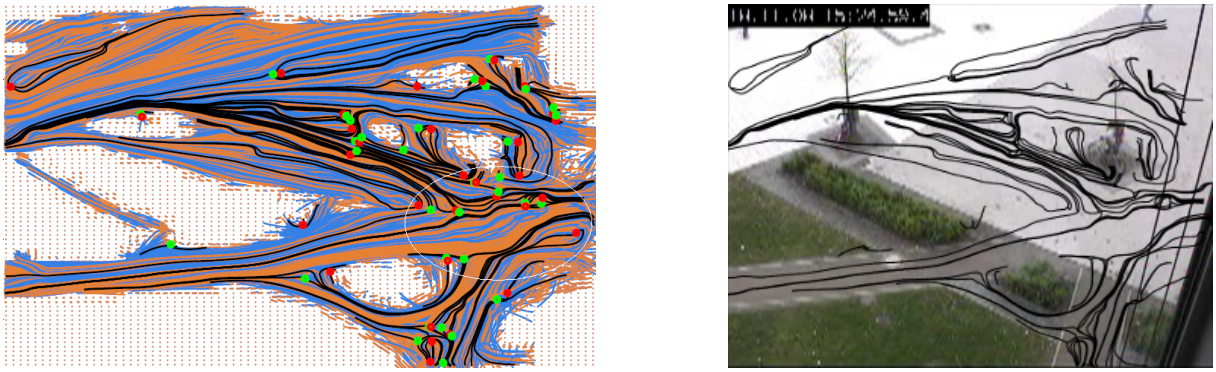

Figure 10: Topology of persons motion trajectories

[2] W. de Leeuw and R. van Liere. Collapsing flow topology using area metrics. In D. Ebert, M. Gross, and B. Hamann, editors, *Proc. IEEE Visualization '99*, pages 149–354, Los Alamitos, 1999.

[3] T. Delmarcelle and L. Hesselink. The topology of symmetric, second-order tensor fields. In *IEEE Visualization*, pages 140–147, 1994.

[4] J. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *Computer*, 22(8):27–36, 1989.

[5] Josef Hoschek and Dieter Lasser. *Grundlagen der geometrischen Datenverarbeitung*. B.G.Teubner, Stuttgart, 1992.

[6] Min Hu, Saad Ali, and Mubarak Shah. Detecting global motion patterns in complex videos. In *ICPR*, pages 1–5, 2008.

[7] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 21:359–381, 2003.

[8] Jae-Gil Lee, Jiawei Han, Xiaolei Li, and Hector Gonzalez. Traclass: trajectory classification using hierarchical region-based and trajectory-based clustering. *PVLDB*, pages 1081–1094, 2008.

[9] R. B. Lehoucq, D. C. Sorensen, and C. Yang. Arpack users' guide: Solution of large scale eigenvalue problems with implicitly restarted arnoldi methods., 1997.

[10] Xiaolei Li, Jiawei Han, Sangkyum Kim, and Hector Gonzalez. Roam: Rule- and motif-based anomaly detection in massive moving object data sets. In *Proceedings of 7th SIAM International Conference on Data Mining*, 2007.

[11] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2):90–126, 2006.

[12] Brendan T. Morris and Mohan M. Trivedi. Learning and classification of trajectories in dynamic scenes: A general framework for live video analysis. In *AVSS '08: Proceedings of the 2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, pages 154–161, 2008.

[13] Mykhaylo Nykolaychuk. Analyse von Bewegungstrajektorien mit Techniken der Strömungsvisualisierung. Master's thesis, Otto-von-Guericke Universität Magdeburg, Fakultät für Informatik, June 2009.

[14] Jonathan Owens and Andrew Hunter. Application of the self-organizing map to trajectory classification. In *VS '00: Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, page 77, 2000.

[15] Dieter Pfoser and Yannis Theodoridis. Generating semantics-based trajectories of moving objects. In *International Workshop on Emerging Technologies for Geo-Based Applications*, pages 59–76, 2000.

[16] C. Piciarelli and G. L. Foresti. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, 27(15):1835–1842, 2006.

[17] Guido Pusiol, Francois Bremond, and Monique Thonnat. Trajectory based primitive events for learning and recognizing activity. In *Second IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS2009)*, 2009.

[18] X. Tricoche, G. Scheuermann, and H. Hagen. Scaling the topology of symmetric, second-order planar tensor fields. In G. Farin, B. Hamann, and H. Hagen, editors, *Hierarchical and Geometrical Methods in Scientific Visualization*, pages 171–184. Springer, Berlin, 2002.

[19] Tino Weinkauf. *Extraction of Topological Structures in 2D and 3D Vector Fields*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik (H. Theisel), March 2008.