Data-Driven Extraction of Curved Intersection Lanemarks from Road Traffic Image Sequences

K. Mück[1], H.-H. Nagel[1,2], and M. Middendorf[1]

[1] Institut für Algorithmen und Kognitive Systeme,
Universität Karlsruhe (TH), Postfach 6980, 76128 Karlsruhe, Germany,
{klaus|nagel|markusm}@ira.uka.de
Phone: +49 721 608–4044; Fax: +49 721 608–6116
[2] Fraunhofer-Institut für Informations- und Datenverarbeitung (IITB),
Fraunhoferstr. 1, 76131 Karlsruhe, Germany

Abstract. Segmentation of optical flow fields, estimated by spatio-temporally adaptive methods, is – under favourable conditions – reliable enough to track moving vehicles at intersections *without using vehicle or road models*. Already a single image plane trajectory per lane obtained in this manner offers valuable information about where *lane markers* should be searched for. Fitting a hyperbola to an image plane trajectory of a vehicle which crosses an intersection thus provides concise geometric hints. These allow to separate images of direction indicators and of stop marks painted onto the road surface from side marks delimiting a lane. Such a 'lane spine hyperbola', moreover, facilitates to link side marks even across significant gaps in cluttered areas of a complex intersection. Data-driven extraction of trajectory information thus facilitates to link *local* spatial descriptions practically across the entire field of view in order to create *global* spatial descriptions. These results are important since they allow to extract required information from image sequences of traffic scenes without the necessity to obtain a map of the road structure and to make this information (interactively) available to a machine-vision-based traffic surveillance system.
The approach is illustrated for different lanes with markings which are only a few pixels wide and thus difficult to detect reliably without the search area restriction provided by a lane spine hyperbola. So far, the authors did not find comparable results in the literature.

1 Introduction

Geometric results derived from (model-based) tracking of road vehicles in traffic image sequences can already be transformed into *conceptual* descriptions of road traffic. The generation of such descriptions from video recordings of road traffic at inner-city intersections – see, e. g., [2,3,4,6,11] – presupposes, however, the availability of knowledge about the spatial lane structure of the intersection. In addition to knowledge about the geometric arrangement of lanes, knowledge about lane attributes is required such as, e. g., which lane might be reserved for left or right turning traffic – see Figure 1 for illustration.

D. Vernon (Ed.): ECCV 2000, LNCS 1843, pp. 411–427, 2000.
© Springer-Verlag Berlin Heidelberg 2000

So far, this kind of knowledge had to be provided a-priori by the designer(s) of a system, either by a qualitative interactive extraction from image sequence data or by digitizing a map of the intersection. Even if a city administration provides a map of an intersection comprising all significant lane markings, however, experience has shown that such a map may not be up-to-date.

Fig. 1. The left panel shows a representative frame from a traffic intersection video sequence recording traffic from the incoming arm of road A (upper left quadrant) through the intersection to the outgoing arm of road B at the bottom. The right panel shows another frame from this same sequence, recorded while pedestrians where allowed to walk and no vehicles happen to be in the field of view of the recording video camera. Lane markings have been extracted from this frame.

This state of affairs naturally suggests an attempt to *automatically* extract the lane structure of an intersection from the video sequence recording the road traffic to be analyzed. The next section outlines our approach, followed by a more detailed description in Sections 3 and 4, illustrated by experimental results obtained by an implementation of this approach. Additional experimental results are presented in Section 5. A concise overview of relevant publications is followed by a comparison with our approach and by conclusions in Section 6.

2 Basic Assumptions and Outline of the Approach

Experience with *data-driven* image segmentation approaches – regardless whether edge- or region-oriented – has shown that success at an affordable computational expense depends critically on the exploitation of appropriate *implicit* knowledge about the depicted scene, including its illumination, and about the imaging conditions. We venture that progress results if such implicit knowledge can be explicated and thus made amenable to scrutiny, a precondition for further improvement.

This contribution assumes that vehicles which approach, cross, and leave an intersection stay predominantly within their lane. Image plane trajectories of vehicles thus provide crucial information about the number and geometry of lanes as well as about admitted driving directions. The image region corresponding to a lane can thus be conceived to be a *ribbon* of approximately constant width. The vehicle trajectory constitutes the 'spine' of such a 'lane ribbon'.

Lanes around an innercity intersection are composed in general of three segments: a straight 'approaching' arm, a straight-line or circular-arc segment across the intersection proper, and a straight 'leaving' arm. As will be shown, these characteristics can be captured astonishingly well and concisely by one arm of a hyperbola. The assumed 'hyperbolic spine' of a lane thus constitutes a powerful cue for lane interpolation across the intersection area.

Direction markers painted onto the road surface are expected near the spine of a lane ribbon and parallel to it whereas stopping lines cut nearly perpendicular across the lane ribbon. Sidemarks delimit the lane along the ribbon borders in the approaching and leaving arm segments, but are often omitted within the intersection area itself. All lane markers are assumed to be elongated bright blobs surrounded by dark road surface.

These quite general – but nevertheless essentially appropriate – assumptions can be exploited to the extent that we succeed to extract vehicle trajectories in the image plane *without introducing knowledge about position, orientation, shape, and motion* of vehicles in the scene. Instead, we assume that vehicles can be represented as blobs with sufficient *spatiotemporal* contrast – i. e. not necessarily purely spatial contrast – to segment them from the remainder of the imaged scene. We assume, moreover, that such 'object image candidates (OICs)' *move smoothly* in the image plane.

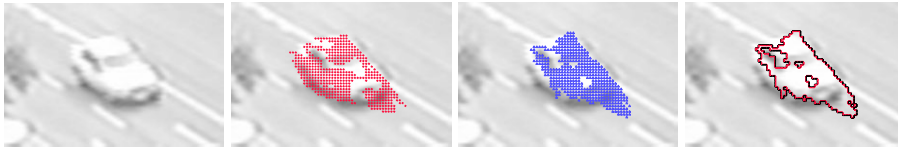
The transformation of these assumptions into an algorithm for the extraction of lane descriptions from intersection traffic image sequences will be treated in more detail in the next section.

3 Extraction of Lane Spines from Image Sequences

Automatic machine-vision-based traffic surveillance at an intersection avoids several complications if a *single* camera records the *relevant intersection area* (no tracking beyond the field of view of a camera into that of another, more simple camera calibration, no inter-camera correspondence problems, etc.). The price to be paid for this simplification consists in rather small image structures which have to be detected, tracked, and classified, since a large field of view at a given resolution results in small object images. The following discussion treats only the evaluation of *monocular* image sequences.

3.1 Detecting and Tracking the Image of a Moving Road Vehicle

If a vehicle trajectory is expected to provide the information about where in the image one should search for lane markings, then this trajectory must be extracted with a minimum of a-priori knowledge. Detection and tracking of moving



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

Fig. 2. The leftmost panel (a) shows an enlarged section around a vehicle to be tracked, cropped from the 375th frame of the sequence illustrated in Figure 1. Panel (b) shows the accepted OF-vectors overlayed to this enlarged section whereas the OIC-mask generated for the first subsequence ending with this frame is shown as an overlay in panel (c). Panel (d) shows OIC-mask contours obtained for this vehicle, overlayed to the 375th frame of this sequence. The OIC-mask position has been advanced from frame m to $m+1$ by the average optical flow vector $\overline{\mathbf{u}_m}$ obtained from the OF-blob determined for this vehicle in frame m. The centers of these OIC-masks form a trajectory obtained by a purely data-driven approach.

vehicles will be investigated by estimation and segmentation of densely populated Optical Flow (OF) fields, despite the considerable computational expenses involved: compared with, e. g., change detection approaches, OF-field segments provide immediately usable information about magnitude and direction of the shift velocity of greyvalue structures. In many cases, such information allows to exclude alternative interpretations of results. If executed properly, moreover, the estimation of OF-fields provides additional information about the reliability of an estimation and, thereby, further supports the algorithmic analysis of any result in doubt.

In view of the rather small image structures to be tracked, we adopted – albeit in a modified manner – a recently published approach towards OF-estimation [13]: optical flow is computed by determination of the eigenvector corresponding to the smallest eigenvalue of the so-called 'Greyvalue Local Structure Tensor' $\overline{\nabla g (\nabla g)^T}$, a weighted average (over a local environment of the current pixel) of the outer product of $\nabla g = (\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}, \frac{\partial g}{\partial t})^T$, where ∇g denotes the gradient of the greyvalue function $g(x, y, t)$ with respect to image plane coordinates (x, y) and time t. Spatio-temporal adaptation of the filter masks for gradient estimation improves the trade-off between noise reduction and separation of different greyvalue structures.

A 4-connected region of OF-estimates is then selected as an 'OF-blob', provided at each pixel position within such a region

1. the OF-magnitude exceeds a minimum threshold (separation between stationary background and moving vehicles),
2. the smallest eigenvalue of $\overline{\nabla g (\nabla g)^T}$ is smaller than a threshold (i. e. the greyvalue structure remains essentially constant in the OF-direction), and
3. the two larger eigenvalues of $\overline{\nabla g (\nabla g)^T}$ both exceed a minimum threshold (i. e. the greyvalue variation in both spatial directions is sufficient to reliably estimate an OF-vector).

OF-blobs which can be persistently detected at mutually (according to the OF-estimate) compatible image plane locations are aggregated into an 'Object Image Candidate (OIC)' as follows. We assume that the *shape* of a vehicle's image does not change significantly during a subsequence of n ($\simeq 12$) frames. Let $\overline{\mathbf{u}_k}$ denote the average OF-vector of the k-th OF-blob within a subsequence ($k = 1, 2, \ldots, n-1$). The k-th OF-blob is shifted forward by $\sum_{j=k}^{n-1} \overline{\mathbf{u}_j}$ (with components of the resulting sum vector rounded to the next integer value) and 'stacked' on top of the OF-blob extracted from the last frame within this subsequence. Among the pixel locations supporting this stack, we retain as an *initial* OIC-mask only those which are covered by at least p % (with, e. g., $p = 45$) of the n possible entries from the stacked OF-blobs. The OIC-contour is then taken as a rough estimate for the image shape of a moving vehicle.

For the first subsequence, $\mathrm{OIC}_{1_{\mathrm{initial}}}$ is accepted in this form. In the case of later subsequences, however, consecutive OIC-masks are merged in order to adapt an OIC-mask to possible changes in appearance of an object image. An OIC-mask OIC_{i-1} obtained from the $(i-1)$-th subsequence is shifted by $\overline{\mathbf{u}_{(i-1)_n}} + \sum_{j=1}^{n-1} \overline{\mathbf{u}_{i_j}}$ to the location of $\mathrm{OIC}_{i_{\mathrm{initial}}}$ obtained from the next (i-th) subsequence. The two OIC-masks are stacked, each with the weight (i. e. hit count) determined at its generation. Analogously to the generation of an initial OIC-mask, all pixel locations are retained which received a weight of at least p % of the $2n$ possible entries from the two stacked OIC-masks.

Figure 2 illustrates the generation of an OIC-mask.

3.2 Extraction of the 'Lane Spine'

Fig. 3. Left panel: The OIC-mask contours, shown every 20 half-frames, with small dark 'x' denoting the center of an OIC-Mask. Right panel: The 'lane spine', a hyperbola fitted according to equ. 1 – using every half-frame – to the OIC-mask trajectory shown in the left panel.

As mentioned in Section 2, image plane trajectories of vehicles crossing the intersection illustrated in Figure 1 can in good approximation be described by a hyperbola. In analogy with established practice for ellipses [16], we fit the following polynomial

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{1}$$

to a sequence of OIC-mask centers such as shown in Figure 3. The free parameter in this homogeneous equation is fixed by the requirement $A - C = 1$, in analogy to the well-known trace requirement $A + C = 1$ for ellipse fitting (see, e. g., [16]). The degenerate case of a straight-line trajectory is automatically detected and treated separately. The axes of the right-handed eigenvector coordinate system of the resulting conic are fixed by requiring that the apex of the hyperbola intersects the first axis and the estimated vehicle trajectory opens into the positive direction of this axis.

4 Extraction of Lane Structures from Image Sequences

In innercity intersections such as those shown in Figure 1, lanes are in general delimited at the sides by bright lines, by sidewalks with a visible curb, or by a surface with color or texture visibly different from the lane surface. We generally expect, therefore, to find edge elements marking the side of a lane image. The problem consists in the challenge to reliably detect a sufficiently large fraction of these edge elements in order to facilitate their concatenation into a coherent lane delimiter. At this point, we introduce a-priori knowledge about (continuous or interrupted) lane side boundary markings in the form of a 'lane model'. In principle, one could define such a model in the (2-D) image plane or in the (3-D) scene. We decided to define the model in the scene, because the dependencies between the 'lane spine' and the hyperbolas representing the lateral lane boundaries are more simply described in the scene. If we know the projective transformation by the camera system, a lane representation in the scene domain can be transformed without any further heuristics into the image plane. Our lane model will be denoted as a 'hyperbolic ribbon'. This hyperbolic ribbon will be fitted to the edge elements tentatively selected as marking the side boundary of a lane image.

4.1 Hyperbolic Ribbon as Lane Model

As mentioned in section 2, the spine of curved intersection lanes can in good approximation be described by a hyperbola. Based on such a 'hyperbolic spine', we are able to construct a ribbon of hyperbolas to describe the lateral delimiters of a curved lane. A lane has one delimiter at the right and one at the left side. So it seems to be enough to construct a ribbon of three hyperbolas. One hyperbola can be defined by five parameters $[m_x, m_y, \theta, a, b]$. The parameters m_x and m_y represent the center point, θ describes the orientation and a, b specify the form

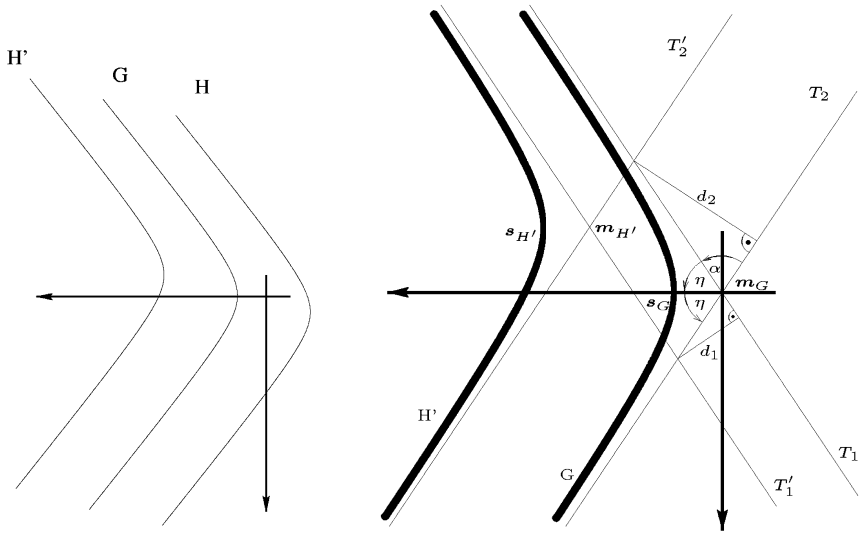


Fig. 4. The left panel depicts the 'lane spine' G as shown in the right panel of Figure 3, together with the two 'lane boundary' hyperbolas H, H'. These three hyperbolas represent a lane as a 'hyperbolic ribbon'. The 'leaving' arm is wider than the 'approaching' arm of the lane, but the asymptotes of all hyperbolas are parallel. The coordinate system shown refers to the lane spine G, i. e. $m_{Gx} = m_{Gy} = 0$ and $\theta_G = 0$. The right panel illustrates how to construct one side of a hyperbolic ribbon with different widths. The center point $\boldsymbol{m}_{H'}$ of the hyperbola H' can be constructed with the normalized directions of the asymptotes T_1 and T_2, the angle $\alpha = \pi - 2\eta$, and a scale factor based on the widths d_1, d_2. The direction and orientation of the hyperbola H' are the same, respectively, as those of the lane spine G, i. e. the normalized directions of the asymptotes T_1', T_2' are the same as the normalized directions of T_1, T_2. The position of the apex $\boldsymbol{s}_{H'}$ is defined by the distance between the apices $\boldsymbol{s}_G$ and $\boldsymbol{s}_{H'}$.

of a hyperbola. A ribbon of three – nominally unrelated – hyperbolas therefore requires the specification of 15 parameters.

The outer hyperbolas can be derived from the hyperbolic spine inside the ribbon with a few basic and useful assumptions about the symmetry of a lane. These assumptions, based on official guidelines for constructing innercity intersections, lead to a large reduction of the number of parameters while defining a hyperbolic ribbon (see Figure 4). All of the following definitions are related to the eigensystem of the lane spine G which is shifted by $(m_x, m_y)^T$ and rotated by θ relative to the coordinate system of the scene, according to the parameters $[m_x, m_y, \theta, a, b]$ of lane spine G:

- *Orientation*: All hyperbolas have the same orientation θ. This reduces the number of parameters from 15 by two to 13.
- *Shape*: Each hyperbola has a pair of asymptotes. These asymptotes represent the straight parts of a lane. We assume, therefore, that all aperture

angles have the same value η with $\tan(\eta) = b/a$. This reduces the number of parameters by two more to 11.

– *Position*: The 'approaching' arm and the 'leaving' arm of a lane at an innercity intersection can differ in their width. The parameters d_1 and d_2 take this observation into account. Figure 4 shows how the center point of the hyperbola H' is moved. It can be computed from the center point of G as follows:

$$\boldsymbol{m}_{H'} = \begin{pmatrix} (d_1 + d_2)/(2\sin(\eta)) \\ (d_1 - d_2)/(2\cos(\eta)) \end{pmatrix}. \tag{2}$$

The position of the center point of the hyperbola H can be computed analogously:

$$\boldsymbol{m}_{H} = -\begin{pmatrix} (d_1 + d_2)/(2\sin(\eta)) \\ (d_1 - d_2)/(2\cos(\eta)) \end{pmatrix}, \tag{3}$$

Equations 3 and 2 can be derived based on the geometric context shown in the right panel of Figure 4. This reduces the number of parameters further by two parameters to 9.

– *Apex*: The last free parameter defines the position of the apex. This position specifies, too, the acuity of the hyperbola at this point, because the center point, orientation and aperture angle are fixed. A hyperbola G in the normal form

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \tag{4}$$

has its apex at $(a, 0)^T$. Multiplying this vector by a factor p moves the apex along the first axis: $(p{\cdot}a, 0)^T$. At the same time, the hyperbolas have to retain their shape. This can be achieved by multiplying the shape parameter b with the same factor p. The aperture angle η — where $tan(\eta) = (p{\cdot}b)/(p{\cdot}a) = b/a$ — will thus remain unchanged. The distance between the apices $\boldsymbol{s}_{H'}$ and $\boldsymbol{s}_G$ as well as between $\boldsymbol{s}_H$ and $\boldsymbol{s}_G$ is defined by

$$\|\boldsymbol{s}_{\boldsymbol{H'}}\boldsymbol{s}_{\boldsymbol{G}}\| = \|\boldsymbol{s}_{\boldsymbol{H}}\boldsymbol{s}_{\boldsymbol{G}}\| = \tau \cdot \frac{d_1 + d_2}{2}. \tag{5}$$

Usually the width of the curved section of a lane is greater than in the straight lane sections. This can be taken into account by setting τ to values greater than 1. The factors p_H and $p_{H'}$, referring to the lane boundary hyperbolas H and H', respectively, can be computed by solving:

$$p_H \cdot a = a + \frac{d_1 + d_2}{2\sin(\eta)} - \sqrt{\left(\tau\frac{d_1 + d_2}{2}\right)^2 - \left(\frac{d_2 - d_1}{2\cos(\eta)}\right)^2}, \tag{6}$$

$$p_{H'} \cdot a = a - \frac{d_1 + d_2}{2\sin(\eta)} + \sqrt{\left(\tau\frac{d_1 + d_2}{2}\right)^2 - \left(\frac{d_2 - d_1}{2\cos(\eta)}\right)^2}. \tag{7}$$

Using these assumptions, the initial number of 15 parameters can be reduced to only 8 parameters for describing a complete hyperbolic ribbon:

$$< (m_x, m_y)^T_G, \theta_G, a_G, b_G, d_1, d_2, \tau >$$

where the index G refers to the lane spine G.

This model approximates a *lateral lane delimiter* by an infinitely thin line. In order to take the finite width of a lane delimiter into account, we extend the model: we use a hyperbolic ribbon for each lane delimiter, this time parameterized for the width dw of the lane delimiter. The complete lane model can thus be described by the tuple $< (m_x, m_y)^T_G, \theta_G, a_G, b_G, d_1, d_2, \tau, dw_1, dw_2 >$ of parameters. The parameters dw_1, dw_2 represent the width of the left and right lane boundary delimiter, respectively. Two separate parameters are necessary in general since lane delimiters between adjacent lanes may differ in width from those delimiting the road sides.

4.2 Fitting Hyperbolic Ribbons to Edge Elements

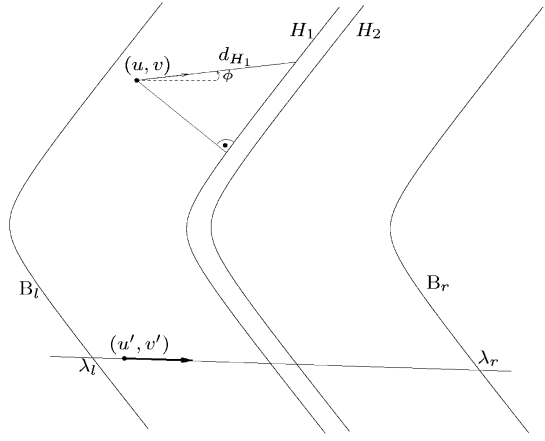


Fig. 5. The hyperbolas H_1 and H_2 model one lateral boundary of a lane. All edge elements inside the hyperbolic tolerance ribbon limited by the hyperbolas B_l and B_r are candidates for the fitting step. An edge element has a position and a direction, so we can interpret an edge element as a line. As shown in the lower part of this Figure, the intersection of this line with the hyperbolas B_l and B_r results in the parameters λ_l and λ_r. If $\lambda_l < 0 < \lambda_r$, the edge element inside the hyperbolic ribbon limited by B_l and B_r will be fitted to the hyperbola H_1, otherwise to H_2. The edge element shown at position $(u, v)^T$ in the upper part of this Figure has the distance d_{H_1} from H_1. This distance measure takes into account the gradient orientation at the point $(u, v)^T$. If we consider an edge element as a line, this distance can be derived by intersecting this line with the hyperbola H_1.

Fitting hyperbolic ribbons to edge elements is based on the approach of [10]. Although the lane model is defined in the scene, the projected image of this model

is fitted to the edge elements in the image: the sum of Mahalanobis distances between edge elements and the projected model – parameterized according to the actual parameter estimates (which constitute the components of the filter state vector) – is iteratively optimized. The finally accepted state vector represents the matched lane in the scene.

The estimation process is initialized by the lane spine mentioned above. The widths of the arms of a lane are initially set to the *same* standard value. The typical lane width, the typical width at the curved section of a lane, and the typical width of the lateral lane boundary delimiters can be found in guidelines for constructing innercity intersections. All parameters can thus be initialized by reasonable values.

The distance function takes the gradient orientation of an edge element into account (see Figure 5). Let an edge element $\boldsymbol{e} = (u, v, \phi)^T$ and the hyperbola H be given by

$$\boldsymbol{e}: \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \end{pmatrix} + \begin{pmatrix} u \\ v \end{pmatrix}, \tag{8}$$

$$H: \; Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{9}$$

The signed distance function $d_H(\boldsymbol{e}, \boldsymbol{x})$ which quantifies the distance between an edge element $\boldsymbol{e}$ and the hyperbola H under the actual state $\boldsymbol{x}$ (with $\boldsymbol{x} = (m_{xG}, m_{yG}, \theta_G, a_G, b_G, d_1, d_2, \tau, dw_1, dw_2)^T$) is defined by:

$$d_{\mathrm{H}}(\boldsymbol{e}, \boldsymbol{x}) = \begin{cases} \lambda_1, & |\lambda_1| < |\lambda_2| \\ \lambda_2, & \text{otherwise} \end{cases} \tag{10}$$

with

$$\lambda_1 = \frac{-p/2 + \sqrt{p^2/4 - qr}}{r}, \qquad \lambda_2 = \frac{-p/2 - \sqrt{p^2/4 - qr}}{r},$$

$$r = A\cos^2(\phi) + B\cos(\phi)\sin(\phi) + C\sin^2(\phi),$$

$$p = 2A\cos(\phi)u + B(\cos(\phi)v + \sin(\phi)u) + 2C\sin(\phi)v + D\cos(\phi) + E\sin(\phi),$$

$$q = Au^2 + Buv + Cv^2 + Du + Ev + F \quad .$$

Note that a negative distance is possible.

The lane is modeled with two lane delimiters. Each of them consists of two outer hyperbolas H_1, H_2 (the hyperbolic spine is only used for the construction of the hyperbolic ribbon, but not for the computation of the distance between edge elements and the projection of the lane model). Fitting all edge elements of the entire image to a lane model will consume too much time. It is necessary, therefore, to decide which edge element should be associated with which hyperbola. The first constraint is satisfied by building a hyperbolic tolerance ribbon around each of the delimiters (see Figure 5). All edge elements inside such a hyperbolic tolerance ribbon between the hyperbolas B_l and B_r are considered candidates for the fitting process. The association of an edge element with the correct hyperbola (H_1 or H_2) is based on the typical greyvalue distribution near a lane

delimiter since delimiters are in general brighter than the rest of the lane. An edge element represents this property by the gradient direction at this position: this direction has to be approximately perpendicular to the assigned hyperbola. It is easy to decide whether an edge element has to be associated with the left or the right hyperbola of the delimiter: An edge element considered as a directed line can be intersected with the boundaries of the hyperbolic tolerance ribbon. Let $\lambda_l = d_{B_l}(\boldsymbol{e}, \boldsymbol{x})$ and $\lambda_r = d_{B_r}(\boldsymbol{e}, \boldsymbol{x})$ denote the distances of the edge element to the intersections of this line with the hyperbolas B_l and B_r. If $\lambda_l < 0 < \lambda_r$, the edge element will be associated with hyperbola H_1, otherwise with hyperbola H_2 (see Figure 5).

5 Experimental Results

Figure 6 shows two successfully detected lanes with fitted models overlayed. The fitting process describes the lane delimiters essentially correct. A small discrepancy in the curved section of the lane shown in the left panel corresponds to the variation of the lane width as described above.

Both lanes differ in width between their approaching (2.85 m) and their leaving (3.50 m) arms. In addition, the widths at the stop line is a little bit smaller (2.75 m) than in the incoming straight sections of both arms. The lane spine derived for the right lane was a bit off the lane center since the driver of the respective car did not drive strictly along the lane center. The starting state for the right boundary delimiter thus was initialized too far to the right. The fitting process could not fully correct this state initialization. A possible solution for such a problem could consist in exploiting a longer observation of the scene: rather than relying on the first trajectory in order to derive a lane spine, one could select a more appropriate one, based on a statistical analysis of several trajectories.

The left panel of Figure 7 shows *all* hyperbolas used in fitting a lane. The enlarged sections allow to verify the quality of fit in more detail.

The approach has been extended in order to treat *neighboring* lanes with essentially the same parameter set as illustrated by the right panel of Figure 7: assuming equal widths and orientation for both lanes and symmetrical apex displacements for boundary spines, we only need one additional parameter, namely for the width of the common boundary delimiter separating the two neighboring lanes. In this manner, we are able to fit six hyperbolas – which can differ substantially with respect to their position and orientation *in the image plane* – to about 4900 edge elements, using only 11 parameters. A backprojection of lane delimiters extracted in this manner into the road plane and a comparison with official map information about lane markings at the depicted intersection yields good agreement (see Figure 8).

The approach reported up to here has been extended even further in order to cope with situations such as those illustrated by the left upper panel in Figure 9: since this sequence comprises only fifty frames, no vehicle trajectory covers a lane across the entire intersection. We thus fit multiple trajectory segments (see

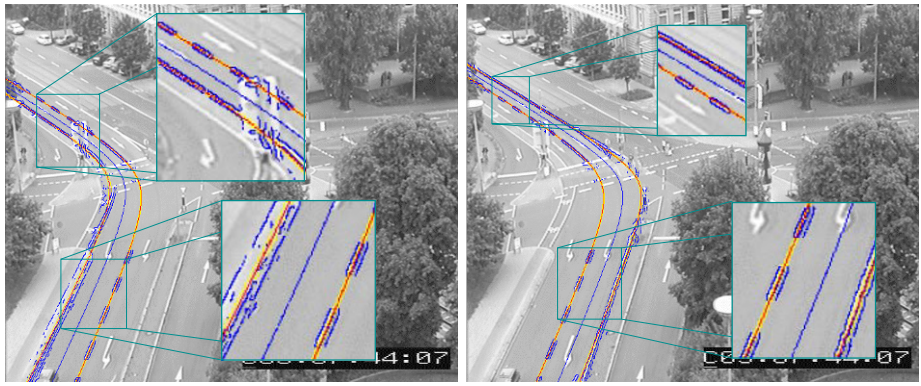


Fig. 6. The left panel shows a fit of a hyperbolic ribbon to the *left* lane, computed according to the approach described in Section 4. The hyperbolic spine is drawn in blue, the hyperbolas in the *centre* of the lane delimiters are painted red, while the hyperbolas which model the *boundaries* of the lane delimiters are painted yellow. All edge elements which are taken into account during the fitting process are painted in blue, too. Within the upper enlarged subwindow, it can be clearly seen that edge elements due to the large traffic sign which occludes part of this lane do not endanger an appropriate fit. The right panel shows a fit of a hyperbolic ribbon to the *right* lane, in analogy to the left panel.

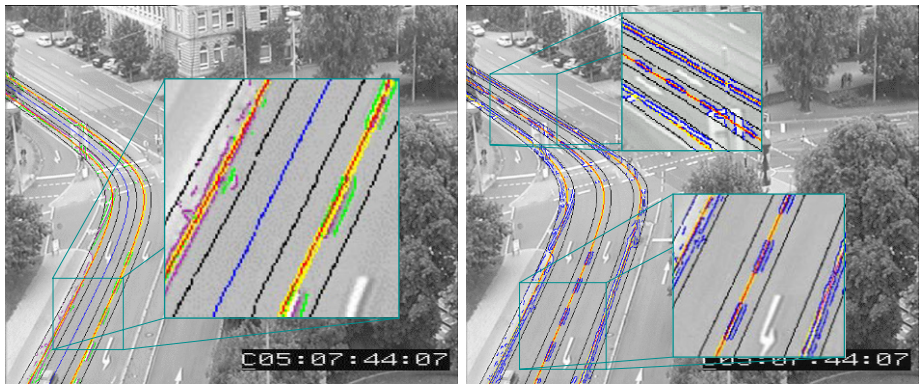


Fig. 7. The left panel shows *all* hyperbolas exploited for the fit to the left lane, together with an enlarged section. The black hyperbolas bound the area in which edge elements are supposed to belong to the enclosed lane delimiter. One can clearly detect the differently (colored in either pink or green) oriented edge elements on the sides of bright blobs corresponding to short lane markings. (Notice the effect of greyvalue overshoot in the scanline direction for a transition between the bright lane markings and the dark background which results in unexpected additional edge elements colored in green, clearly visible in the insets!) The right panel shows a *simultaneous* fit of two hyperbolic lane models side-by-side (using a **joint** parameter set, thus significantly reducing the number of parameters required to describe **both** lanes) to two neighboring lanes.

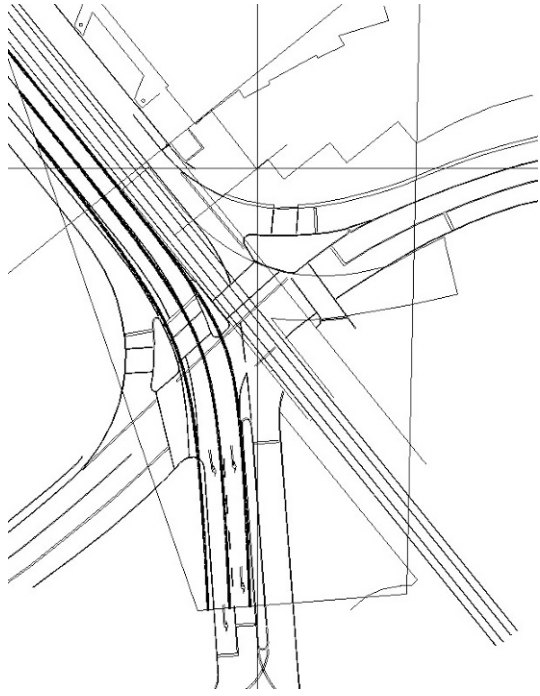
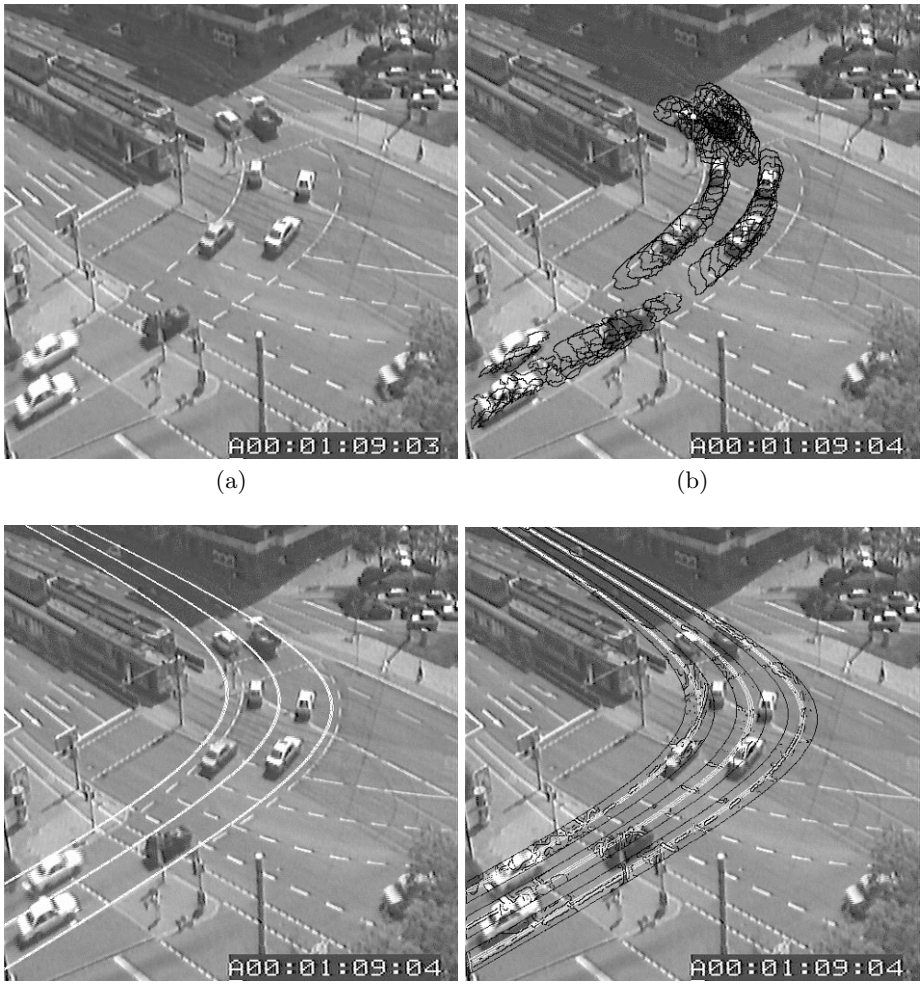


Fig. 8. This image shows the backprojections of two simultaneously fitted lanes (see Figure 7) into a map of the intersection provided by the official land registry. The comparatively small deviations in the lower part of the image are attributed to small systematic errors of the camera calibration. As Figure 7 shows, the lanes were fitted well in the image plane both for the incoming and the leaving arm. It can be seen that the incoming lanes become smaller close to the stopping line in order to allow for an additional 'bicycle lane' for those bicyclists who want to continue straight ahead.

the top right panel in Figure 9) from *different* vehicles in neighboring lanes to a *jointly* parameterized *pair* of two neighboring 'lane spine hyperbolas'. The initial estimates for these lane spine hyperbolas are shown in the lower left panel and the final result for the boundary limits in the lower right panel of Figure 9. The remaining deficiency regarding the 'leaving' arm is due to the lack of an appropriate initialization and of sufficient contrast to facilitate a correction in the estimation of lane boundary delimiters. This Figure illustrates both the power and current limits of our approach.

6 Comparison and Conclusions

The idea to exploit data-driven tracking of moving objects in video sequences in order to derive descriptions of developments in the scene has found increasing interest recently, due to methodological improvements in the detection and tracking of moving objects. The continuous decrease of the price/performance



(a) (b)

(c) (d)

Fig. 9. The upper left panel (a) shows an overview of another traffic intersection. In the upper right panel (b), object contours of *all* vehicles were overlayed. This variation of the basic method was used due to the shortness – 100 halfframe – of the video sequence. The starting state of the fitting process based on the object contour centroids is shown in panel (c). Despite of the short sequence, the model of two lanes could be fit successfully to the visible lane delimiters (see panel (d)). The quite bad result in the upper left quadrant of the lower right panel (d) is due to missing tracking data – and thus missing object contour centroids – in this area.

ratio for computers has significantly accelerated research in this area. Recent investigations have concentrated on tracking moving bodies in order to extract significant *temporal events* (see, e. g., [14], [5]) or to build coarse scene models in addition, but based on multiple calibrated cameras [7]. Others use data-driven tracking in order to recover 3D trajectories under the 'ground plane constraint' and exploit the knowledge acquired thereby to control an active camera platform in order to keep a moving body within the field of view of the camera set-up [1].

Lane finding is an important subtask for machine-vision-based control of road vehicles. In such a context, execution time is at a premium, with the effect that preference is given to computationally simple algorithms. As long as a vehicle guided by machine-vision just has to follow an (at most slowly curving) road or to continue straight ahead across an intersection, lane boundaries can be approximated sufficiently well by low-order polynomials of the horizontal image coordinate as studied, for example, by [8]. In [12], similar simple road configurations are investigated for lane keeping purposes by machine-vision. Although [9] addresses this same problem of lane keeping, its author analyzed the image of a slowly curving lane recorded by a video-camera mounted behind the windshield of a driving car and concluded that it can be well approximated *in the image plane* as a hyperbola – essentially due to the effect of perspective projection under the conditions mentioned. In our case, the lane spine is modeled by a hyperbola in the *scene*: the straight line sections enclosing the curved section are due to the lane structure across an intersection in the scene – as opposed to be due to a perspective effect in the image plane associated with a constant curvature lane in the scene. Since we exploit the hyperbolic lane structure in the scene in order to reduce the number of parameters to be estimated for lane boundary delimiters and neighboring lanes, our approach turns out to be able to cope successfully with a considerable number of potentially distractive edge elements.

So far, we did not encounter an example where a vehicle trajectory has been used to extract a global *quantitative* description of multiple lane borders in an image sequence of a nontrivial traffic scene.

We exploit image-plane trajectories of vehicles in order to collect evidence in the image plane about the exact position of side marks in the form of edge segments in very *restricted* image regions, thereby significantly reducing the danger of picking up unwanted edge elements or edge segments. Fitting an hyperbola to vehicle trajectories enables us to interpolate the (frequently 'invisible') part of the lane within an intersection area. *Local* spatial descriptors can thereby be linked along the vehicle trajectory into the remainder of the field of view, thus establishing *global* spatial descriptors. An added attraction of our approach is seen in the fact that the transformation of hyperbolic curves under perspective projection can be studied in closed form. This should facilitate investigations regarding a quantitative transfer of spatial descriptions within an image into spatial *scene* descriptions.

Future research will not only be devoted to increase the robustness of the approach reported here, but also to develop estimation procedures for *initializa-*

tion parameters which have so far been set interactively, such as the initial width for a lane. The accumulation of information about lanes from *multiple* vehicle trajectories certainly belongs into this category, too, picking up ideas reported by, e. g., [15].

Robust automatic extraction of lane boundaries should facilitate the detection and classification of additional road markers painted onto a lane. Such a capability allows to determine the 'semantics' of such a lane, for example that it constitutes a lane reserved for left turning traffic – for example, see Figure 7. This information will significantly simplify to characterize traffic behavior at the *conceptual* or even *natural language* level of description.

Acknowledgments

We thank M. Haag for discussions and help during our investigations and for valuable comments on a draft version of this contribution.

Partial support of this research by the Deutsche Forschungsgemeinschaft (DFG) is gratefully acknowledged.

References

1. K.J. Bradshaw, I.D. Reid, and D.W. Murray: "The Active Recovery of 3D Motion Trajectories and Their Use in Prediction", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-19, pp. 219–233, 1997.
2. H. Buxton and S. Gong: "Visual Surveillance in a Dynamic and Uncertain World", *Artificial Intelligence*, Vol. 78, pp. 431–459, 1995.
3. S. Dance, T. Caelli, and Z.-Q. Liu: "A Conceptual, Hierarchical Approach to Symbolic Dynamic Scene Interpretation", *Pattern Recognition*, Vol. 29(11), pp. 1891–1903, 1996
4. J.H. Fernyhough, A.G. Cohn, and D.C. Hogg: "Generation of Semantic Regions from Image Sequences", in Proc. *Fourth European Conference on Computer Vision (ECCV'96)*, 15-18 April 1996, Cambridge/UK; B. Buxton and R. Cipolla (Eds.), Lecture Notes in Computer Science LNCS 1065 (Vol. II), pp. 475–484, Springer-Verlag Berlin, Heidelberg, New York 1996.
5. J.H. Fernyhough, A.G. Cohn, and D.C. Hogg: "Building Qualitative Event Models Automatically from Visual Input", Proc. ICCV'98, Bombay/India, pp. 350–355, January 1998.
6. R. Gerber and H.-H. Nagel: "(Mis-?)Using DRT for Generation of Natural Language Text from Image Sequences", in Proc. *Fifth European Conference on Computer Vision*, 2-6 June 1998, Freiburg/Germany; H. Burkhardt and B. Neumann (Eds.), Lecture Notes in Computer Science LNCS 1407 (Vol. II), pp. 255–270, Springer-Verlag Berlin, Heidelberg, New York 1998.
7. W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee: "Using Adaptive Tracking to Classify and Monitor Activities in a Site", Proc. CVPR'98, pp. 22–29, June 1998.
8. F. Guichard and J. Ph. Tarel: "Curve Finder Combining Perceptual Grouping and a Kalman Like Fitting", Proc. International Conference on Computer Vision ICCV'99, 20–27 September 1999, Kerkyra (Corfu), Greece, pp. 1003–1008.

9. A. Guiducci: "Parametric Model of the Perspective Projection of a Road with Applications to Lane Keeping and 3D Road Reconstruction", Computer Vision and Image Understanding **73**:3 (1999) 414–427.
10. F. Heimes, H.-H. Nagel, and Th. Frank: "Model-Based Tracking of Complex Innercity Road Intersections", *Mathematical and Computer Modelling*, Vol. 22(9-11), pp. 189–203, 1998.
11. R.J. Howarth: "Interpreting a Dynamic and Uncertain World: Task-Based Control", *Artificial Intelligence*, Vol. 100, pp. 5–85, 1998.
12. Ch. Kreucher and S. Lakshmanan: "LANA: A Lane Extraction Algorithm that Uses Frequency Domain Features", IEEE Trans. on Robotics and Automation **15**:2 (1999) 343–350.
13. H.-H. Nagel and A. Gehrke: "Spatiotemporally Adaptive Estimation and Segmentation of OF-Fields", in Proc. *Fifth European Conference on Computer Vision (ECCV'98)*, 2-6 June 1998, Freiburg/Germany; H. Burkhardt and B. Neumann (Eds.), Lecture Notes in Computer Science LNCS 1407 (Vol. II), pp. 86–102, Springer-Verlag Berlin, Heidelberg, New York 1998.
14. N. Johnson and D. Hogg: "Learning the Distribution of Object Trajectories for Event Recognition", Proc. BMVC'95, pp. 583–592, 1995.
15. M. Mohnhaupt and B. Neumann: "On the Use of Motion Concepts for Top-Down Control in Traffic Scenes", Proc. ECCV'90, Antibes/France, O. Faugeras (Ed.), LNCS 427, pp. 598-600, April 1990.
16. Z. Zhang: "Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting", *Image and Vision Computing*, Vol. 15, pp. 59–76, 1997.