Discrete Driver Assistance

Reinhard Klette[1], Ruyi Jiang[2], Sandino Morales[1], and Tobi Vaudrey[1]

[1] The University of Auckland, Auckland, New Zealand
[2] Shanghai Jiao Tong University, Shanghai, China

Abstract. Applying computer technology, such as computer vision in driver assistance, implies that processes and data are modeled as being discretized rather than being continuous. The area of stereo vision provides various examples how concepts known in discrete mathematics (e.g., pixel adjacency graphs, belief propagation, dynamic programming, max-flow/min-cut, or digital straight lines) are applied when aiming for efficient and accurate pixel correspondence solutions. The paper reviews such developments for a reader in discrete mathematics who is interested in applied research (in particular, in vision-based driver assistance). As a second subject, the paper also discusses lane detection and tracking, which is a particular task in driver assistance; recently the Euclidean distance transform proved to be a very appropriate tool for obtaining a fairly robust solution.

Keywords. Discrete mathematics, driver assistance, stereo analysis, lane detection, distance transform

1 Vision-Based Driver Assistance

Driver assistance systems (DAS) are developed to (i) *predict* traffic situations, (ii) *adapt* driving and car to current traffic situations, and (iii) *optimize* for safety. Vision-based DAS applies one or multiple cameras for understanding the environment, to help achieve goals (i-iii).

After specifying a processing model, possibly in continuous space, any specification for its algorithmic use will depend on discrete mathematical models, such as numerical algorithms [13], or concepts in discrete mathematics such as adjacency sets $A(p)$ of pixels p, digital straight lines, or distance transforms, which are examples from digital geometry [14]. Typically, continuous models are used in motion analysis up to the moment when mapping those concepts into algorithms, but matching techniques for multi-ocular vision typically already start with a discrete model.

This paper is organized as follows: Section 2 describes techniques applied in binocular correspondence analysis, followed by Section 3 with (further) illustrations of matching results in vision-based DAS. Lane detection via distance transform is the subject of Section 4. A few conclusions are given in Section 5.

2 Stereo Algorithms

Stereo algorithms are designed for calculating pairs of corresponding pixels in concurrently recorded images. After calibration and rectification [6], images L (left) and R (right) are in *standard stereo geometry* (i.e., parallel optical axes, coplanar image planes, aligned image rows), defined on pixels of an $M \times N$ grid Ω. See Figure 1; the third view has been used in [17] for prediction error analysis.

Thus, stereo pixel correspondence is basically a 1D search problem, compared to motion pixel correspondence (e.g., [10]) which is a continuous 2D search problem. Two corresponding pixels $p_L = (x, y)$ and $p_R(x - \Delta(x, y), y)$ identify a *disparity* $\Delta(x, y)$ which defines the *depth* $bf/\Delta(x, y)$, where b is the *base distance* between both focal points, and f is the uniform focal length of both cameras (after rectification). However, the search should also account for disparity consistency between adjacent scan lines (e.g., between rows y, $y - 1$, and $y + 1$).

2.1 Data and Continuity Terms

This *stereo matching problem* is an instance of a general *pixel labeling problem*: given is a finite set $\mathcal{L}$ of labels $l, h, \ldots$; define a labeling Δ which assigns to each pixel $p \in \Omega$ (in the *base image*; we assume L to be the base image) a label $\Delta_p \in \mathcal{L}$. Consider a *data term* of penalties $D_p(\Delta_p)$ for assigning label Δ_p to pixel p. The simplest data term is given by $D_x(l) = |L(x, y) - R(x - l, y)|^b$, for a fixed row y, $1 \leq x \leq M$, and b either 1 or 2, assuming that image pairs are *photo-consistent* (i.e., corresponding pixels have about the same value).

[7] compares various data terms within a *the-winner-takes-all strategy*: for each pixel $p = (x, y)$ in the left image, a selected data term is applied for all potential matches $q = (x - l, y)$ in the right image, for $l \geq 0$; that l is taken as disparity which defines a unique (within the whole row) minimum for this cost function; if there is no such unique global minimum then the disparity at p remains undefined. – For example, results in [7] indicate that the census cost function seems to be very robust (w.r.t. image data variations) in general.

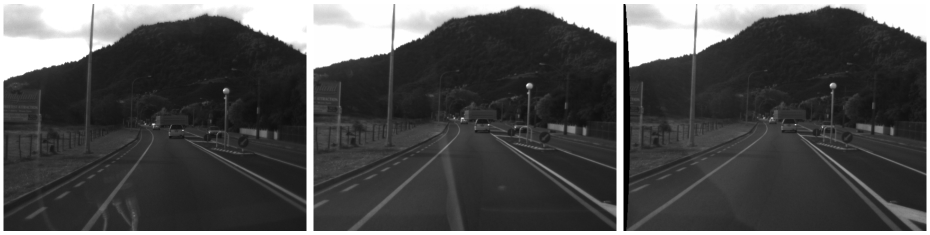

Fig. 1. One time frame of image sequences taken with three cameras (called: *third*, *left*, and *right camera* - from left to right) installed in HAKA1, test vehicle of the *.enpeda..* project. Note the reflections on the windscreen, and differences in lightness (e.g., image of right camera is brighter than the other two). Left and right views are rectified.

The minimization of the following *energy* (or: *cost*) *functional* E defines a basic approach for solving the stereo matching problem:

$$E(\varDelta) = \sum_{p \in \varOmega} \left(D_p(\varDelta_p) + \sum_{q \in A(p)} C(\varDelta_p, \varDelta_q) \right) \tag{1}$$

This functional combines a data term with a *continuity term* $C(\varDelta_p, \varDelta_q)$, which is often simplified to a unary symmetric function $C(|\varDelta_p - \varDelta_q|)$, for assigning labels $\varDelta_q$ to adjacent pixels $q \in A(p)$. Further terms may be added (e.g., for occlusion, or ordering constraint). The continuity term assumes that projected surfaces are *piecewise smooth* (i.e., neighboring pixels represent surface points which are at about the same distance to the cameras). A convex function C supports efficient global optimization, but leads to oversmoothed results [11].

Common choices for a unary continuity function are either a simple step function (Pott's model), a linear function, or a quadratic function, where the latter two need to be truncated for avoiding oversmoothing. A simple choice is also a two-step function, which penalizes small disparity changes at adjacent pixels with a rather low weight (to allow for slanted surfaces), but penalizes larger disparity changes with a higher weight.

Finding a global minimum $\varDelta$, which minimizes the energy in Equation (1), assuming a continuity function which is not enforcing some kind of over-smoothing, is an NP-hard problem; see [15]. Purely local matching strategies (e.g., hierarchical correlation based methods) failed to provide reasonable approximate solutions. Strategies favored recently follow some semi-global optimization scheme.

2.2 Semi-Global Paradigms for Sub-Optimal Solutions

Basically, current stereo algorithms follow one of the following three paradigms: scanline optimization often implemented by (DP) *dynamic programming* [18], (BP) *belief propagation* [3], or (GC) *graph-cut* [15]. These paradigms aim at finding a sub-optimal solution to the stereo matching problem.

SGM using Scanline Optimization. *Semi-global matching* (SGM) is commonly identified with applying scanline optimization along several digital rays, all incident with the start pixel p in the base image [9]. Original dynamic programming stereo [18] was defined for energy minimization along a single scan line. Assume that row y remains constant; matching aims at minimizing

$$E_m(\varDelta) = \sum_{x=1}^{m} \left(D_x(\varDelta_x) + \sum_{\hat{x} \in A(x)} C(\varDelta_x, \varDelta_{\hat{x}}) \right) \quad \text{with} \quad E(\varDelta) = E_M(\varDelta) \tag{2}$$

Value m defines the *stage* of the dynamic optimization process; when arriving at stage m we have assignments of labels $\varDelta_x$, for all x with $1 \leq x < m$ (possibly

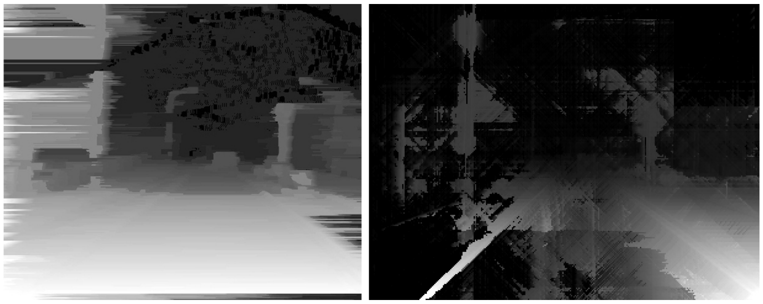

Fig. 2. Left: streaks in a single-line DP result (for left and right image as shown in Figure 1). Right: visible search line patterns in the calculated depth map for 8-ray DP using mutual information as cost function (also known as SGM MI).

excluding pixels close to the left border of the left image), and we have not yet assignments for $x \geq m$; we select $\varDelta_m$ by taking that $l \in \mathcal{L}$ which minimizes

$$D_m(l) + C(l, \varDelta_{m-1}) + E_{m-1}(\varDelta) \tag{3}$$

(Obviously, the term $E_{m-1}(\varDelta)$ can be deleted for the minimization task.) At $m = 1$ we only have $\varDelta_1 = 0$, for $m = 2$ we may decide between $l = 0$ or $l = 1$, and so forth. When arriving at stage $x = M$, we have an optimized value $E(\varDelta)$ (modulo the applied DP strategy); we identify the used labels for arriving at this value by backtracking, from $x = M$ to $x = M - 1$, and so forth.

Dynamic programming propagates errors along the used digital line; here, this occurs along image rows, from left to right, resulting in horizontal streaks in the calculated depth map. Disparities in adjacent pixels of the same line, or in adjacent rows may be used to define a continuity term in the used energy function for reducing this streak effect. A further option is to combine forward DP also with the following backward DP strategy:

$$E^m(\varDelta) = \sum_{x=m}^{M} \left(D_x(\varDelta_x) + \sum_{\hat{x} \in A(x)} C(\varDelta_x, \varDelta_{\hat{x}}) \right) \quad \text{with} \quad E(\varDelta) = E^1(\varDelta) \tag{4}$$

where $\varDelta_m$ is selected as that $l \in \mathcal{L}$ which minimizes

$$D_m(l) + C(l, \varDelta_{m-1}) + E_{m-1}(\varDelta) + C(l, \varDelta_{m+1}) + E^{m+1}(\varDelta) \tag{5}$$

($E_{m-1}(\varDelta)$ and $E^{m+1}(\varDelta)$ can be ignored again.) Obviously, this requires to proceed up to $x = m$ with 'normal' DP both from left and from right, then combining values along both digital rays into one optimized value $\varDelta_m$ at $x = m$ following Equation (5). This increases the time complexity, compared to the simple approach in Equation (3), and time-optimization is an interesting subject.

This double-ray approach was generalized to optimization along multiple digital rays [9], thus approximating the global (NP-complete; see above) solution.

For a digital ray in direction $\mathbf{a}$, processed between image border and pixel p, consider the segment $p_0 p_1 \dots p_{n_\mathbf{a}}$ of that digital ray, with p_0 on the image border, and $p_{n_\mathbf{a}} = p$; the energy contribution along that ray at pixel p is defined via scanline optimization as in Equation (3). All used digital rays $\mathbf{a}$ (ending at p) are assumed to have identical impact; the label at pixel p is obtained by generalizing Equation (5); we assign that disparity l which minimizes

$$D_p(l) + \sum_{\mathbf{a}} \left[C(l, \mathit{\Delta}_{n_\mathbf{a}-1}) + E_{n_\mathbf{a}-1}(\mathit{\Delta}) \right] \tag{6}$$

Labeling $\mathit{\Delta}$ in Equation (6) obtains thus a further value $\mathit{\Delta}_p$ at pixel p. Again, all those $E_{n_\mathbf{a}-1}(\mathit{\Delta})$ can be deleted for minimization, and algorithmic time optimization (say, scanline optimization along all lines at first, and then combining results) leads to a feasible solution. [8] also includes a second-order prior into the used energy function.

Belief Propagation. Belief propagation is a very general way to perform probabilistic inference; the BP stereo matching algorithm in [3] passes messages (the "belief" which is a weight vector for all labels) around in a 4-adjacency image grid. Message updates are in iterations; messages are passed on in parallel, from a pixel to all of its 4-adjacent pixels. At one iteration step, each pixel of the adjacency graph computes its message based on the information it had at the end of the previous iteration step, and sends its (new) message to all the adjacent pixels in parallel.

Let $m^i_{q \to p}$ denote the message send from pixel q to adjacent pixel p at iteration i, defined for all $l \in \mathcal{L}$ as follows:

$$m^i_{q \to p}(l) = \min_{h \in \mathcal{L}} \left(C(h, l) + D_q(h) + \sum_{r \in A(q) \setminus p} m^{i-1}_{r \to q}(h) \right) \tag{7}$$

l is just one of the $|\mathcal{L}|$ possible labels at p, and h runs through $\mathcal{L}$ and is again just a possible label at q. We accumulate at p a vector of length $|\mathcal{L}|$ of all messages received from all $q \in A(p)$, and this contains at its position $l \in \mathcal{L}$ the following:

$$D_p(l) + \sum_{q \in A(p)} m^i_{q \to p}(l) \tag{8}$$

Besides $D_p(l)$, we also have the sum of all the received message values for $l \in \mathcal{L}$. Instead of passing on vectors of length $|\mathcal{L}|$, a belief propagation algorithm typically uses $|\mathcal{L}|$ *message boards* of the size of the images, one board for each label l. At the end of an iteration t, that disparity with minimum cost is selected as being the result for pixel $p \in \mathit{\Omega}$.

BP fails in cases of photometric inconsistencies between left and right image; [5] showed that some edge preprocessing of both images is of benefit, and [19] performed a systematic study which shows the residual images (rather than original input images) carry the important information for correspondence analysis.

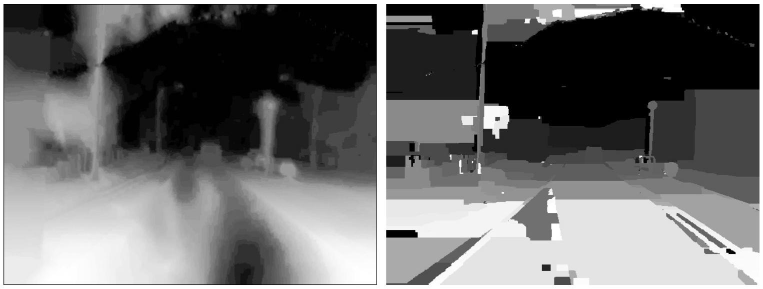

Fig. 3. Left: BP result (for left and right image as shown in Figure 1). Right: GC result.

Graph-Cut. Consider 4-connected pixels of the base image; this defines an undirected graph (Ω, A) with nodes Ω and edges A. Assume two additional nodes s and t, called *source* and *sink*, respectively, with directed edges from s to all the nodes in Ω, and from those nodes to t. This defines altogether an undirected graph $G = (\Omega \cup \{s,t\}, A \cup \{s\} \times \Omega \cup \Omega \times \{t\})$. Edges in this graph are weighted by $w(p,q)$ (continuity values to undirected edges and data values to undirected edges), also called *capacities*.

An (s,t)*-cut* of G is a partition of $\Omega \cup \{s,t\}$ into subsets S and $\overline{S}$, with $s \in S$ and $t \in \overline{S}$. The energy $E(S)$ of such an (s,t)-cut is the sum of all weights of edges connecting S with $\overline{S}$:

$$E(S) = \sum_{p \in S, q \in \overline{S}, \{p,q\} \in A} w(p,q) \tag{9}$$

A *minimum* (s,t)*-cut* is an (s,t)-cut with minimum energy. Ford and Fulkerson (see [4]) proved that the calculation of a minimum cut is equivalent to the calculation of a maximum flow; the calculation of a min-cut is commonly implemented via calculating a max-flow. Used algorithms have about $\mathcal{O}(n^4)$ worst case run time, but run in practice in about $\mathcal{O}(n^3)$ expected time or better.

Let Δ be a labeling of Ω. Any α*-expansion* of Δ into Δ' satisfies that

$$\Delta'_p \neq \Delta_p \Longrightarrow \Delta'_p = \alpha$$

for every pixel $p \in \Omega$. The following *expansion-move algorithm* is a greedy algorithm which runs in practice in near-linear time:

```
start with an arbitrary labelling $\Delta$ on $\Omega$;
do { success := false;
    for each label $\alpha \in \mathcal{L}$ {
        calculate the minimum-energy $\alpha$-expansion $\Delta'$ of $\Delta$;
        if $E(\Delta') < E(\Delta)$ then { $\Delta := \Delta'$;   success := true}   }   }
until success = false;
return $\Delta$
```

The calculation of the minimum-energy α-expansion is performed by applying a min-cut (meaning, via max-flow) algorithm. An α-extension either keeps an old label $\varDelta_p$ or assigns the new label α; this defines a partition of the graph into set S (old label) and $\overline{S}$ (new label). Equation (9) defines the energy for this partitioning (labeling). See, for example, [1] for more details.

3 Improving Stereo Results by Preprocessing

Obviously, the illustrated resulting depth maps are not satisfactory for DAS. Errors are often due to varying illumination conditions or other real world imaging effects, and this is different to studies using only ideal images taken indoors

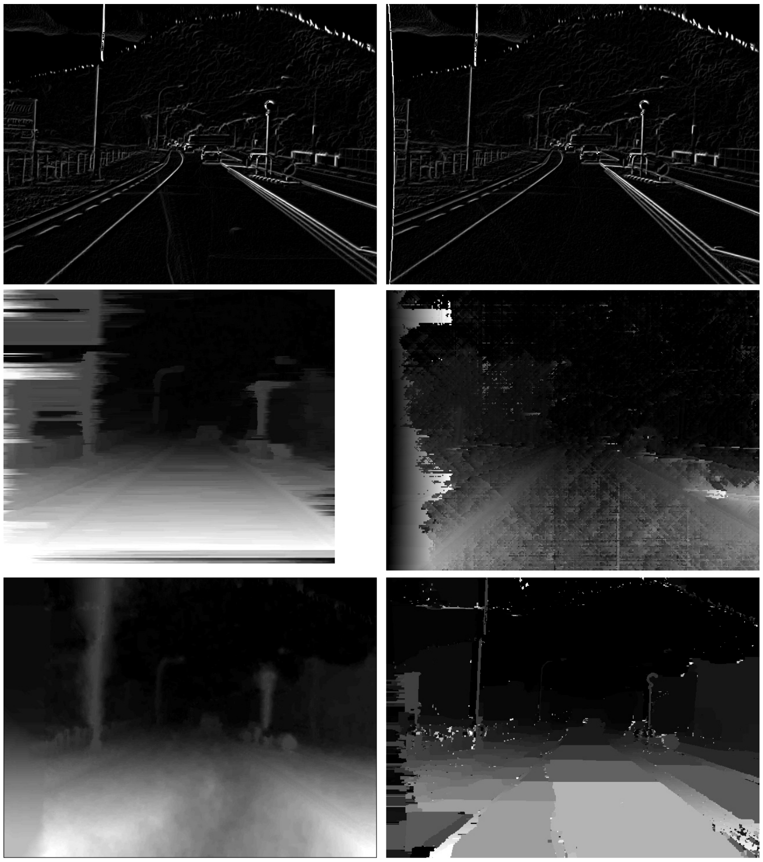

Fig. 4. Top: Sobel edge maps of the left-right stereo pair in Figure 1, used as stereo input pair. Middle: depth maps of single line DP (left) and Birchfield-Tomasi cost function in 8-ray DP (also known as SGM BT). Bottom: BP (left) and GC results.

or under controlled conditions. The paper [19] identified residual images as a promising type of input data for stereo or motion correspondence algorithms.

3.1 Edge Operators

Earlier than [19], the paper [5] studied the effect of edge-preprocessing on stereo matching, showing that edge-preprocessed input data improve resulting depth maps in general, especially when applying BP.

Figure 4 shows four resulting depth maps for the left-right stereo pair in Figure 1, but after applying the (3 × 3) Sobel edge operator. In case of 8-path DP, the use of the Birchfield-Tomasi cost function (SGM BT) shows better

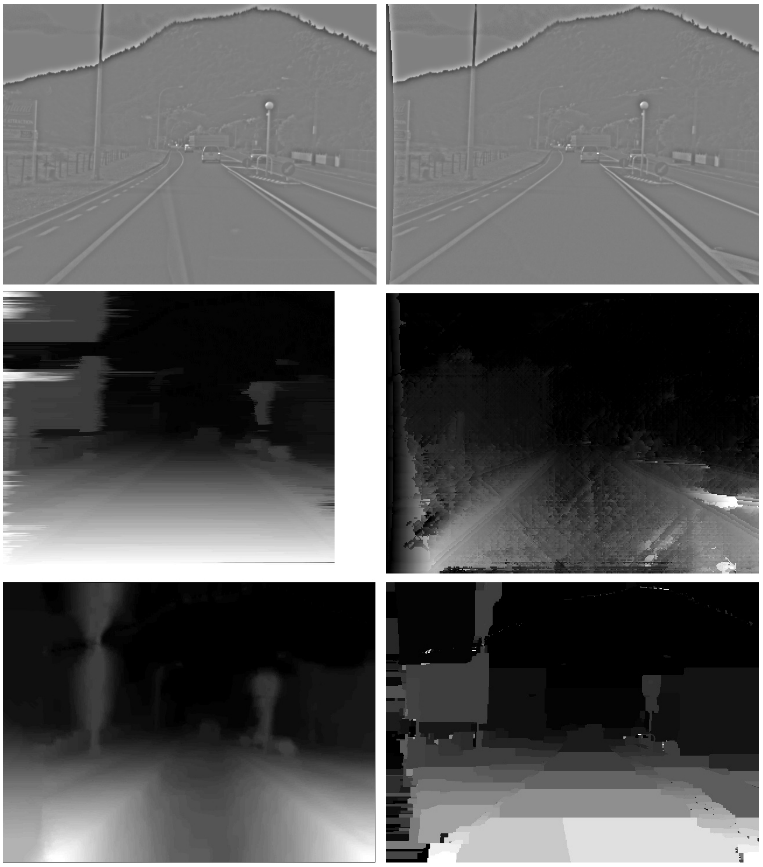

Fig. 5. Top: residual images using 40 iteration of 3×3 mean, used as stereo input pair. Middle: depth maps of single line DP (left) and Birchfield-Tomasi cost function in 8-ray DP (also known as SGM BT). Bottom: BP (left) and GC results.

results compared to the use of mutual information (SGM MI). – Edge images are high-frequency components of images, and the same is true for residual images.

3.2 Residual Images

We consider an image I as being a composition $I(p) = \mathbf{s}(p) + \mathbf{r}(p)$, for $p \in \Omega$, where $\mathbf{s} = S(I)$ denotes the *smooth component* and $\mathbf{r} = I - \mathbf{s}$ the *residual*. We use the straightforward iteration scheme:

$$\mathbf{s}^{(0)} = I, \quad \mathbf{s}^{(n+1)} = S(\mathbf{s}^{(n)}), \quad \mathbf{r}^{(n+1)} = I - \mathbf{s}^{(n+1)}, \quad \text{for } n \geq 0.$$

Figure 4 shows four resulting depth maps for the left-right stereo pair in Figure 1, but on the residual images defined by a 3×3 mean operator and $n = 40$ iterations.

As a general conclusion, single-line DP is quite robust and provides a fast and approximate depth map ("a good draft", and some kind of temporal or spatial propagation of results might be useful [16]), SGM-BT fails absolutely on original data but seems to perform better than SGM-MI on preprocessed images, BP is highly sensitive to illumination changes, and improves very nicely when using optimized parameters on preprocessed input data, and GC also improves on preprocessed input data.

4 Lane Detection and Tracking

Lane detection and tracking has been a successful research subject in DAS. [12] reviews briefly related work in vision-based DAS and discusses a new lane model, also providing two algorithms for either time-efficient or robust lane tracking (to be chosen in dependency of current road situation).

4.1 Bird's-Eye View and Edge Detection

The proposed lane detection and tracking algorithms work on a single image sequence. However, the figures show results for both stereo sequences in parallel, illustrating this way the robustness of the method with respect to different camera positions.

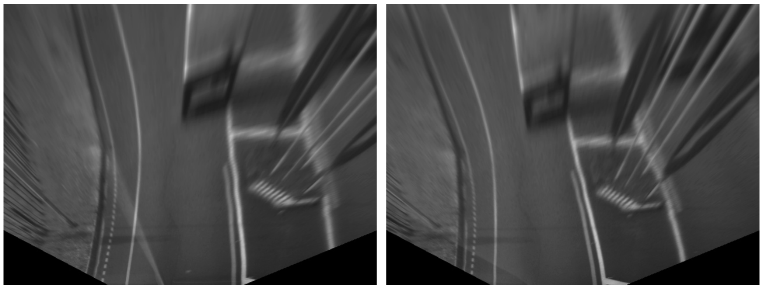

Fig. 6. Bird's-eye views (for left and right image as shown in Figure 1).

The process starts with mapping a given image into a bird's-eye view (i.e., a homography mapping a perspective image into an orthographic top-down view), based on calibrated projections of corners of a rectangle in front of the car; see Figure 6 for two resulting bird's-eye views. This is followed by an edge detection method which aims at detecting vertical step edges (such as lane marks) rather than horizontal edges; small artifacts are eliminated from these binarized edge images which would otherwise disturb the subsequent distance transform.

4.2 Distance Transform and Lane Tracking

Consider a distance transform, applied to a binary edge map, which labels every pixel $p \in \Omega$ by its shortest distance to any edge pixel; see, for example, [14], for distance transforms in general. Experiments have been performed with various kinds of distance transforms, and preference was given to the Euclidean distance transform (EDT). For example, [2] proved that a 2D EDT can efficiently calculated by two subsequent 1D EDT. (The developed procedure for calculating lower envelops is also applicable for calculating lower envelops in a BP algorithm while using a truncated quadratic continuity function.)

[20] suggested the *orientation distance transform* (ODT) which separates EDT values into a row and a column component, represents as complex numbers. We use the real (i.e., row) part of the ODT. See Figure 7 for binary input edge maps and resulting RODT (i.e., real ODT) maps, where gray values increase with measured distance.

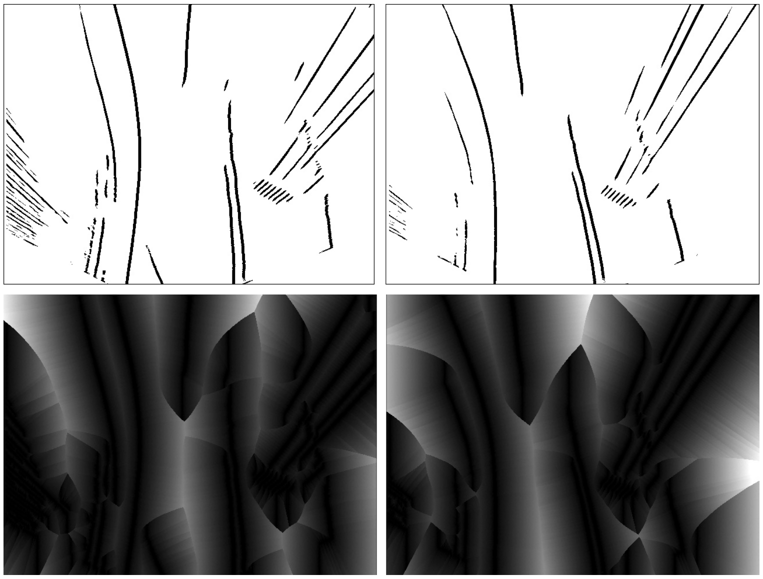

Fig. 7. Detected edges (top) and RODT results (bottom). Both for bird's-eye views as shown in Figure 6. In RODT, dark = low distance, white = high distance.

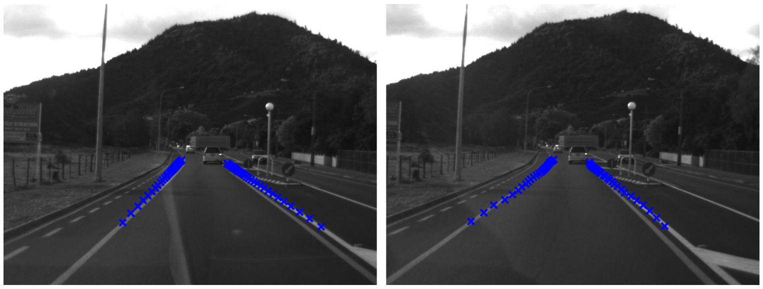

Fig. 8. Detected lanes (for left and right image as shown in Figure 1), illustrating robustness of the technique (i.e., independence of camera position).

In a predefined start row (near to the image's bottom) we identify a left and right boundary point for the current lane based on the calculated RODT values; these two boundary points initialize a particle filter for lane detection. The subsequent lane tracking module applies either an efficient (but less robust; designed for good road conditions), or a robust (but less efficient) algorithm.

Figure 8 shows final results of lane detection (using robust method), for left and right sequence. Some kind of unification might be considered; however, our experience shows that the method performs very robust on a single image sequence. Both tracking algorithms are operating in the bird's-eye views, and both are using results of the RODT for evaluating possibilities of finding lane boundaries.

The RODT not only provides information about the expected centerline of a lane but also about lane boundaries. However, it takes slightly more computation time than the total for generating the bird's-eye view, edge detection, and removal of artifacts.

5 Conclusions

This paper informs the reader about a few subjects where discrete mathematics have met program development in recent vision-based DAS, and proved to be very useful for defining fairly efficient, accurate or robust techniques. The discussed stereo techniques have been proposed elsewhere, and we provided a brief and uniform presentation, together with experimental illustration. The lane detection and recognition solution was reported in [12]. Vision-based DAS is expected to move further ahead, from low-level stereo and motion analysis into advanced subjects for understanding complex traffic scenes, and further interactions with discrete mathematics are certainly coming this way.

References

1. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis Machine Intelligence*, **23**:1222–1239, 2001.
2. P. F. Felzenszwalb and D. P. Huttenlocher. Distance transform of sampled functions. Cornell Computing and Information Science, TR 2004-1963, September 2004.
3. P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Int. J. Computer Vision*, **70**:41–54, 2006.
4. L. R. Ford and D. R. Fulkerson. Flows in networks. Technical report R-375-PR, US Air Force Project RAND, 1962.
5. S. Guan, R. Klette, and Y. W. Woo. Belief propagation for stereo analysis of night-vision sequences. In Proc. *PSIVT* (T. Wada et al., eds.), LNCS 5414, pages 932–943, 2009.
6. R. I. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
7. S. Herman and R. Klette. The naked truth about cost functions for stereo matching. MI-tech TR 33, University of Auckland, 2009.
8. S. Hermann, R. Klette, and E. Destefanis. Inclusion of a second-order prior into semi-global matching. In Proc. *PSIVT* (T. Wada et al., eds.), LNCS 5414, pages 633–644, 2009.
9. H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In Proc. *CVPR*, volume 2, pages 807–814, 2005.
10. B. K. P. Horn, and B. G. Schunck. Determining optical flow, in *Artificial Intelligence*, **17**:185–203, 1981.
11. H. Ishikawa. Exact optimization for Markov random fields with convex priors. *IEEE Trans. Pattern Analysis Machine Intelligence*, **25**:1333–1336, 2003.
12. R. Jiang, R. Klette, S. Wang, and T. Vaudrey. Lane detection and tracking using a new lane model and distance transform. MI-tech TR 39, University of Auckland, 2009.
13. R. Kimmel. *Numerical Geometry of Images.* Springer, New York, 2004.
14. R. Klette, and A. Rosenfeld. *Digital Geometry.* Morgan Kaufmann, San Francisco, 2004.
15. V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In Proc. *ECCV*, pages 82–96, 2002.
16. Z. Liu and R. Klette. Dynamic programming stereo on real-world sequences. In Proc. *ICONIP* (M. Köppen et al., eds.), Part I, LNCS 5506, pages 527–534, 2009.
17. S. Morales and R. Klette. Prediction error evaluation of various stereo matching algorithms on long stereo sequences. MI-tech TR 38, University of Auckland, 2009.
18. Y. Ohta and T. Kanade. Stereo by two-level dynamic programming. In Proc. *IJCAI*, pages 1120–1126, 1985.
19. T. Vaudrey and R. Klette. Residual images remove illumination artifacts for correspondence algorithms! In Proc. *DAGM*, 2009 (to appear).
20. T. Wu, X. Q. Ding, S. J. Wang, and K. Q. Wang. Video object tracking using improved chamfer matching and condensation particle filter. In Proc. *SPIE-IS & T Electronic Imaging*, volume 6813, pages 04.1–04.10, 2008.