MARRYING LEVEL-LINE JUNCTIONS FOR OBSTACLE DETECTION

Nikom SUVONVORN

Department of Computer Engineering
Prince of Songkhla University
Hat Yai, Songkhla, 90112, Thailand

François LE COAT, and Bertrand ZAVIDOVIQUE

Institut d'Electronique Fondamentale
Bât 220, Universite Paris-Sud
Centre d'Orsay, F91405 Orsay cedex, France

ABSTRACT

In this paper, we present an application of stable marriages algorithms, applied to level-line junctions, for obstacle detection. The method is implemented on our PiCar stereovision based system, an intelligent vehicle. The performance of the method is tested for precision and safety in real situations and compared with two other methods based on dynamic programming and correlation.

Index Terms— Obstacle detection, stereovision, stable marriage, level-line junctions,

1. INTRODUCTION

Obstacle detection is a key function to be implemented on an intelligent vehicle. Many methods with different approaches have been proposed. The technique implemented on our PiCar system [1] is based on stereo analysis. Obstacles are identified from the disparity field. In that case, the matching step plays the important role for precision and detectability. The two matching methods – Le Coat's [2] and Birchfield's [3] – already implemented on PiCar rely respectively on dynamic programming and correlation. In this paper, we present an alternative combinatorial optimization from a paradigm called "stable marriages". A sparse disparity field is obtained by applying the suitable algorithm to level-line junctions.

The paper is organized as follows. First, we introduce the PiCar vision system. Then, our method to marrying level-line junctions, is presented. Finally, performances are compared between methods by experimenting in real situations.

2. PICAR VISION SYSTEM

The PiCar vision system builds on several components implemented within the real time, stamped, and data flow RTMaps environment. The end image processing result is figured as a Bird View with superimposed obstacle detection and road way tracking. The view is an inverse road plane perspective reconstructed through bilinear image interpolation. That enables to represent the vehicle and its exo-system in a non projective rectified world. All computations are performed over proprioceptive (gyros, odometers ...) and exteroceptive (cameras, GPS ...) sensors.

Most of the image processing layer is using the 1.0 OpenCV library. The stereoscopic disparity map addressed in this paper assumes epipolar rectified images [4]. It is restricted to a Region Of Interest (ROI) bounded up and down by the horizon and a 3 meters blind zone and covering the relevant road way. Disparities lay thus inside a trapezoidal region delimiting the forward vehicle's trajectory.

The driving direction is determined with no need of any line mark, thanks to IPM [5] (Inverse Perspective Mapping) and a Hough transform that makes vote for the road-border transition. Obstacles are detected both longitudinally and laterally in the U and V-disparity dual projection spaces [6], given the slope of cameras wrt. the ground plane and the image height of the horizon. A calibration measure is required to associate the height in the images, related by an hyperbolic relation to longitudinal distances in the real world.

3. MARRYING LEVEL-LINE JONCTIONS

This section presents firstly the EFLAM model for level-line junction extraction. Then, "stable marriages" algorithms for junction matching are described.

3.1. The EFLAM model

Let $I(\mathrm{p})$ be the image intensity at pixel p. Image I can be decomposed into *level sets*. Boundaries of level sets are called *level lines*. Level lines obtained from the level sets can overlay but never cross. *Level-line flow* is locally defined as the overlaying level lines for λ from u to v : $F_{u,v} = \{L_\lambda / \lambda \in]u, v]\}$, with flow extension: $\mathcal{E} = v - u$. *Level-line junction* is then a point between any four pixels in the image where at least two level line flows merge or split. To extract robust level line junctions, the EFLAM model is applied.

The EFLAM model [7] stems from the assumption that level-line junctions are points of interest. Let $\tilde{\mathrm{p}}$ be a junction point located between four neighbor pixels $\tilde{\mathrm{p}}_i$. Then, given a radius ρ, the FLAM's area $\mathcal{R}_i$ according to pixel $\tilde{\mathrm{p}}_i$ is defined as :

$$\mathcal{R}_i = [\ \mathrm{p}_j \in N(\tilde{\mathrm{p}}_i) / \mathcal{C}_{\mathrm{p}_j} = 1\ ] \qquad (1)$$

Where $\mathcal{C}_{\mathrm{p}_j} = \begin{cases} 1 & \text{if} \quad |I(\tilde{\mathrm{p}}_i) - I(\mathrm{p}_j)| \leq \frac{\mathcal{E}}{2} \\ 0 & \text{else} \end{cases}$.

$N(\tilde{\mathrm{p}}_i)$ denotes the set of pixels within the circular mask (with radius ρ) around the pixel $\tilde{\mathrm{p}}_i$ and $\mathcal{E}$ is a (preset) threshold. The figure 1(a) sketches the EFLAM model with FLAM's areas around a Y junction.

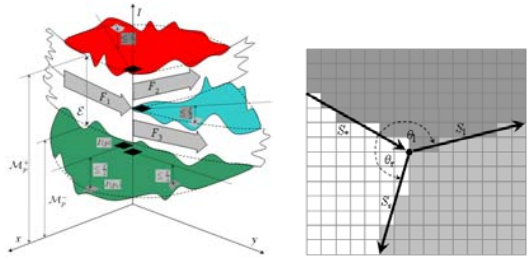


Fig. 1. (a) EFLAM model, (b) Primitive

For each area $\mathcal{R}_i$, the average image intensity $\mathcal{M}_{\tilde{\mathrm{p}}_i}$ is determined as :

$$\mathcal{M}_{\tilde{\mathrm{p}}_i} = \frac{\sum_{[\, p_j \in N(\tilde{\mathrm{p}}_i) / \;\; \mathcal{C}_{\mathrm{p}_j}=1\,]} I(\mathrm{p}_j)}{\mathcal{N}_m} \tag{2}$$

$\mathcal{N}_m$ is the number of pixels in the circular mask. Then, the maximum level line flow $F^*_{\tilde{\mathrm{p}}}$ (and its extension flow $\mathcal{E}$) passing through the junction point $\tilde{\mathrm{p}}$ can be defined according to its neighbors as:

$F^*_{\tilde{\mathrm{p}}} = F^L_{\tilde{\mathrm{p}},\mathcal{M}^{+-}_{\tilde{\mathrm{p}}},\mathcal{M}^{++}_{\tilde{\mathrm{p}}}} \cup F^M_{\tilde{\mathrm{p}},\mathcal{M}^{-+}_{\tilde{\mathrm{p}}},\mathcal{M}^{+-}_{\tilde{\mathrm{p}}}} \cup F^R_{\tilde{\mathrm{p}},\mathcal{M}^{--}_{\tilde{\mathrm{p}}},\mathcal{M}^{-+}_{\tilde{\mathrm{p}}}}$, with transparent notations and $\mathcal{M}^{++}_{\tilde{\mathrm{p}}} \geq \mathcal{M}^{+-}_{\tilde{\mathrm{p}}} \geq \mathcal{M}^{-+}_{\tilde{\mathrm{p}}} \geq \mathcal{M}^{--}_{\tilde{\mathrm{p}}}$. The flow $F^*_{\tilde{\mathrm{p}}}$ is defined as the entering flow towards the junction point while the others are flows out. Note that since the order of intensities around a junction point is fixed (increasing or decreasing order), by refering to $F^*_{\tilde{\mathrm{p}}}$ all flows can be determined automatically.

Thus, the junction variation $\mathcal{V}_{\tilde{\mathrm{p}}}$ is defined as the sum of squared variations of regions $-$ if the difference between the maximum and minimum intensities is equal or higher than $\mathcal{E}$ $-$, if not the variation is null :

$$\mathcal{V}_{\tilde{\mathrm{p}}} = \begin{cases} \sum_{i\in[1,4]} \mathcal{V}^2_{\mathcal{R}i} & \text{if} \quad \mathcal{M}^{++}_{\tilde{\mathrm{p}}} - \mathcal{M}^{--}_{\tilde{\mathrm{p}}} \geq \mathcal{E} \\ 0 & \text{else} \end{cases} \tag{3}$$

Where the variation of each area $\mathcal{V}_{\mathcal{R}_i}$ is defined by:

$$\mathcal{V}_{\mathcal{R}_i} = \begin{cases} \frac{\mathcal{N}_m}{2} - \mathcal{N}_{\tilde{\mathrm{p}}_i} & \text{if} \quad \frac{\mathcal{N}_m}{16} \leq \mathcal{N}_{\tilde{\mathrm{p}}_i} < \frac{\mathcal{N}_m}{2} \\ 0 & \text{else} \end{cases} \tag{4}$$

Where $\mathcal{N}_{\tilde{\mathrm{p}}_i} = \sum_{[\mathrm{p}_j \in N(\tilde{\mathrm{p}}_i)]} \mathcal{C}_{\mathrm{p}_j}$. The junction extraction process is designed as a recursive automaton. When it stops, extracted junctions are then coded into primitives.

$$\overrightarrow{P_Y} = \left[\begin{array}{cccccc} \mathrm{p} & \overrightarrow{S}^* & \overrightarrow{S}^L & \overrightarrow{S}^R & \theta^L & \theta^R \end{array} \right] \tag{5}$$

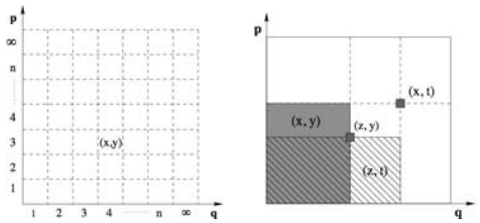
Where $\overrightarrow{S} = \left[\begin{array}{cccc} \mathrm{p} & \theta & L & C(n_l, n_r) \end{array} \right]$. Then, the candidates have to be prepared for matching. Each primitive will create a preference list containing its potential mates (primitives of the other image to be possibly paired). The preferences are set based on the junction similarity.

3.2. Stable marriages algorithms

After preselecting matching candidates by creating the preference list for each primitive, an algorithm of stables marriages suitably designed for such problems will be used for pairing.

The stable marriage problem was first studied by Gale and Shapley [8]. In this problem, two finite sub-sets M and W of two respective populations, say men and women, have to match. Assume n is the number of elements, $M = \{m_i\}_1^n$ and $W = \{w_j\}_1^n$. Each element x creates its preference list $l(x)$ i.e. it sorts all members of the opposite sex from most to less preferred. A matching $\mathcal{M}$ is a one to one correspondence between men and women. If (m, w) is a matched pair in $\mathcal{M}$, we note $\mathcal{M}(m) = w$ and $\mathcal{M}(w) = m$ and ρ_m is the rank of m in the list of w (resp. ρ_w the rank of w in the list of m). Man m and woman w form a *blocking pair* if (m, w) is not in $\mathcal{M}$ but m prefers w to $\mathcal{M}(m)$ and w prefers m to $\mathcal{M}(w)$.

Note that $(m, \mathcal{M}(m))$ and $(\mathcal{M}(w), w)$ are *blocked pairs*. The situation where (m, w) is blocking $(m, \mathcal{M}(m))$ and $(\mathcal{M}(w), w)$ is called *blocking situation* (see figure.2(b)). If there is no blocking pair, then the marriage $\mathcal{M}$ is stable. Gale and Shapley [8]designed an algorithm with complexity $O(n^2)$ that guarantees stability. But it is unfair to one half of the population and it may make every couple unhappy in that men get their best possible choice and women their worst, or conversely.



(a) (b)

Fig. 2. (a) Marriage table : the pair (x,y), y is the 3^{rd} choice of x and x is the 4^{th} choice of y, (b) Blocking situation in a marriage table.

The *marriage table* [9] (see figure.2(a)) is a representation of the stable marriages problem designed to meet the three criteria of stability, sex equality and global satisfaction. It is a table with $(n + 1)$ lines and $(n + 1)$ columns. Lines (resp. columns) frame the preference orders of men, $\{1 \cdots p \cdots N \;\infty\}$ (resp. women, $\{1 \cdots q \cdots N \;\infty\}$). The cell (p, q) contains pairs (m, w) such that w is the p^{th} choice of m, and m is the q^{th} choice of w. Cells can thus contain more than one pair or none. The cell (p, ∞) (resp. (∞, q)) contains the pairs

(m, w) where w is the p^{th} choice of m (resp m the q^{th} choice of w) but m does not exist in her preference list (resp. w is not in his preference list).

So, the global satisfaction of matching can be defined by $\overline{S} = \sum_{(m,w)\in M}(\rho_m + \rho_w)$. Note that a solution with maximum global satisfaction would get matched pairs around the origin of the table (bottom-left). Conversely, sex equality tends to fit the diagonal of the marriage table. It is defined as $\overline{E} = \sum_{(m,w)\in M} |\rho_m - \rho_w|$.

The BZ [10] algorithm consists of scanning the marriage table cells in order to first maximize both criteria concurrently. It scans anti-diagonals forward from maximum to minimum global satisfaction while each one is read in swinging from center to sides meaning maximum to minimum sex equality. In each cell, pairs are married if both partners are free. After all cells have been visited, the table is then revisited again to remove the blocking situations: a blocking pair gets married and corresponding blocked pairs are released. The process repeats until there is no more blocking situation (case "stability") or the iteration number is greater than the population size ("instability").

To recover stability fully, a new direct blocking-pair-removal procedure was designed and tested. The three criteria are conjointly satisfied by BZ in 95% cases, requiring a raise in complexity from $O(n^2)$ to $O(n^3)$. For the 5% remaining instances, a careful case analysis is completed leading to four types of instability, and to the SBZ algorithm [10]. It starts from the best previous one in terms of stability, BZ, and removes selected blocking pairs depending on their type. While stability of the matching is then fully supported – the important progress from the BZ algorithm – both satisfaction and equity are increased in 60% to 70% of the processed cases. However, when satisfaction and equity drop it is in a proportion of about half the BZ score. For stabilizing BZ results with this procedure, the complexity remains $O(n^3)$ in practice. The SBZ is then used for our junctions matching.

4. EXPERIMENTATION AND COMPARISON

The experiment was carried out in real time on the PiCar vehicle. The test is run in two sequences : other vehicle, and then cyclist/pedestrian for obstacle. One thus builds varied situations summarized in the following :

- obstacle is close (resp. obstacle is far): the effectiveness of the method is then measured by the maximal/minimal distances from PiCar to the obstacle.

- obstacle leaves the field of view (resp. obstacle enters the field): the effectiveness is measured by the smallest size of a detected obstacle in the image.

Morever, one could vary the noise conditions present on and around the road (by modifying the environment) and test the robustness with respect to these disturbances.

- the rate of discontinuity in the obstacle detection, whether caused by the system or the environment, is a funding comparative data too.

- the speed of the method must be considered because it can influence all other factors.

Vehicle detection, over 12 minutes and 29 seconds

	Our method	Le Coat	Birchfield and al.
(a)	49m70	25m00	15m20
(b)	0,81% 286 pixels	2,29% 806 pixels	3,88% 1364 pixels
(c)	82,77%	66,35%	55,27%
(d)	3-4	4-5	4-5

Cyclist detection, over 5 minutes and 24 seconds

	Our method	Le Coat	Birchfield and al.
(a)	36m00	23m30	20m70
(b)	0,69% 243 pixels	1,33% 464 pixels	1,36% 480 pixels
(c)	91,66%	55,20%	38,75%
(d)	3-4	4-5	4-5

Table 1. Comparasion between methods (a) The maximum distance from detection (m.) (b) Minimal size of detectable object (%) on the window of 35133 pixels (c) The rate of detectability (%) (d) The speed (images per second)

Methods' performances are compared through these references. The figures 3 and 4 show "left" images of the vehicle/cyclist sequences. In each image, the yellow lines defines a zone in which the obstacle must be detected. This region of interest is specific to the modality "obstacle detection on the road", in order to reduce the computing time. The blue line underlines the detected obstacle. The real distance (in meters) between PiCar and the obstacle, displayed on the top-left image, is computed after identical configuration and calibration of the camera.

Table 1 outlines comparative numbers between methods. It is clear by comparison that our method gives encouraging results – in terms of precision and detection rate, thanks to the marriage of selected junctions. However, the method is slower than the others, mainly due to the combinatorics necessary to meet all three conditions of global satisfaction, fairness and stability is not compensated by the reducing junctions extraction process. Note that the latter can be accelerated independently – e.g. run in parallel by zones or pipelined over more images.

5. CONCLUSION

We presented in this paper a matching method for obstacle detection. The method applies a "stables marriage" algorithm to match level-line junctions, extracted according to the EFLAM model. It is implemented and tested on the PiCar test-bed to analyze image sequences in real-time. The comparison was completed with two methods (Le Coat's and Birchfield's)

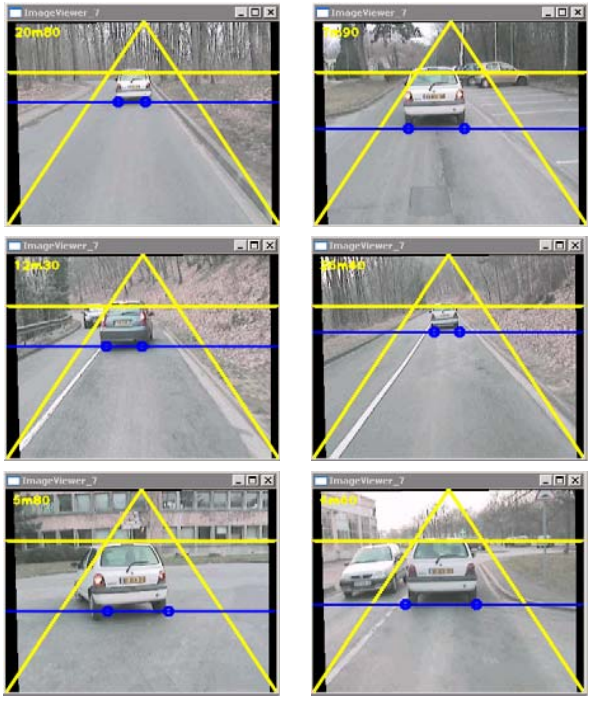


Fig. 3. Result of vehicle detection by our method

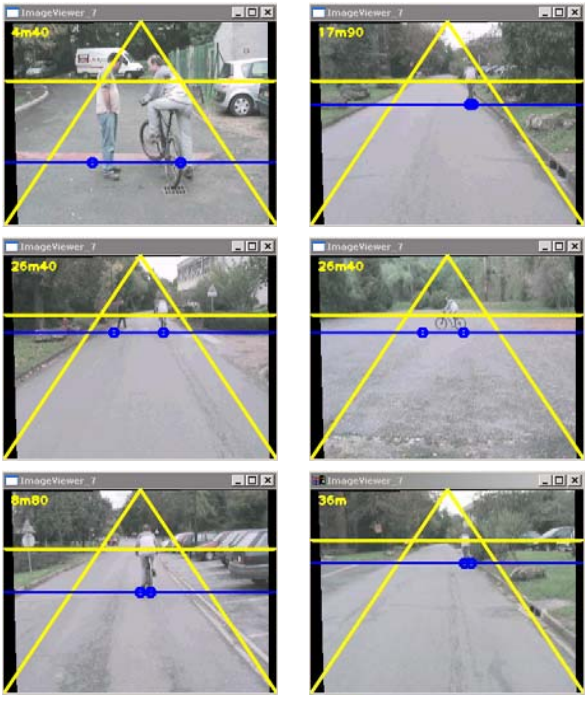


Fig. 4. Result of cyclist detection by our method

based on dynamic programming and correlation. They were already implemented on the vehicle and considered satisfactory for low speed range. The results show that our method brings a significant improvement in terms of precision and detection rate, but increases the computing time in its current implementation.

6. REFERENCES

[1] S. Bouaziz, M. Fan, A. Lambert, T. Maurin, and R. Reynaud, "Picar: experimental platform for road tracking applications," *Intelligent Vehicles Symposium*, pp. 495–499, June 2003.

[2] François LE COAT, *Vers un appariement dynamique d'images : conception, implantation et évaluation d'un algorithme de mise en correspondance d'images.*, Ph.D. thesis, Université de Paris-Sud, janvier 2000, PhD dissertation (in French).

[3] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 269–293, December 1990.

[4] Zhengyou ZHANG, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[5] Massimo BERTOZZI, Alberto BROGGI, and Alessandra FASCIOLI, "Stereo inverse perspective mapping: Theory and applications," *Image and Vision Computing Journal*, vol. 16, no. 8, pp. 585–590, 1998.

[6] Raphael Labayrade and Didier Aubert, "Robust and fast stereovision based obstacles detection for driving safety assistance," *Machine Vision Applications, journal of IEICE*, January 2004.

[7] N. Suvonvorn and B. Zavidovique, "Eflam: A model to level-line junction extraction," in *International Conference on Computer Vision Theory and Applications*, Setbal, Portugal, February 2006.

[8] D. Gale and L.S. Shapley, "College admissions and the stability of marriage," *American Mathematical Monthly*, vol. 69, pp. 9–15, 1962.

[9] B. Zavidovique, N. Suvonvorn, and Guna S. Seetharaman, "A novel representation and algorithms for (quasi) stable marriages," in *International Conference on Informatics in Control, Automation and Robotics*, Barcelona, Spain, September 2005.

[10] N. Suvonvorn and B. Zavidovique, "A stable marriages algorithm to optimize satisfaction and equity," in *International Conference on Image Analysis and Recognition*, Povoa de Varzim, Portugal, September 2006.