21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Fast and efficient vanishing point detection in indoor images

Demetrios Gerogiannis, Christophoros Nikou*, Aristidis Likas
Department of Computer Science, University of Ioannina, Greece
Email: {dgerogia,cnikou,arly}@cs.uoi.gr

Abstract

A method for detecting a vanishing point in structured images is presented. The method relies on the detection of line segments from an edge map by representing clusters of edge points by the long axes of highly eccentric ellipses. The extracted lines provide a set of candidate vanishing points computed by their intersections, which are assigned weights proportional to the lengths of the line segments they belong to. Then, a voting scheme is applied through an accumulator array generated by griding the image frame. The votes of each grid cell are weighted by the Π*-sigmoid kernel allowing cells to contribute to their neighbors.*

1 Introduction

Human-made scenes, such as roads, buildings and their facades or indoor corridor boundaries have a large number of parallel lines in the 3D space. In the framework of a pinhole camera model, two parallel lines are projected onto a pair of converging lines in the 2D image space provided that their 3D plane is not fronto-parallel to the image plane. The common point of intersection of all 3D parallel lines (generally belonging to different planes) in the 2D image is called the vanishing point. The detection of a vanishing point in an image is a crucial step in many computer vision applications, like robot navigation, camera calibration, single view 3D scene reconstruction and pose estimation.

In the related literature, there are two main categories of methods for vanishing point detection. There are techniques requiring knowledge of the intrinsic parameters of the camera, which exploit the notion of 3D parallelism and prominent structures of the scene orthogonal to each other, also called *Manhattan directions* [4, 5]. There are also techniques assuming no knowledge of the internal camera parameters, such as the method in [2] using the Helmholtz principle for image partitioning, the Expectation-Maximization (EM) framework adopted in [7] or the non-iterative algorithm based on consensus sets [11].

In contrast to the above methods, which may base their estimation in the existence of three orthogonal vanishing points, images acquired in structured environments such as roads or corridors are a specific category where the detection of a single vanishing point may be sufficient for the underlined application (e.g. vision-based robot motion along a corridor). The general strategy consists in partitioning the image into accumulator cells collecting votes from the line segments having their intersection in the specific cell. The detection of peaks in the accumulator space provides the vanishing points [8, 10].

In this paper, a method for detecting a single vanishing point in structured environments is presented. The method consists two main steps. At first, line segments are detected using the direct split-and-merge (DSaM) algorithm [6] which allows the robust detection of lines even in cases of points deviating from the line under Gaussian noise assumptions. This modeling allows to capture points which would be rejected as noise in a standard framework like the Hough transform. Secondly, a voting step is applied through a kernel, where candidate vanishing points are assigned weights proportional to the lengths of the line segments they belong to. Therefore, longer line segments which are more probable in indoor environments (e.g. the intersection of wall and ground) are more probable to contribute to the determination of the vanishing point.

2 Vanishing Point Detection

Given an indoor scene (e.g. a corridor), the first step of the method consists in detecting the edges of the image. Therefore, the probabilistic boundaries are first

*This work is co-financed by the European Union (European Regional Development Fund- ERDF) and Greek national funds through the Operational Program THESSALY- MAINLAND GREECE AND EPIRUS-2007-2013 of the National Strategic Reference Framework (NSRF 2007-2013)

computed [9] though in simpler, non textured environments, the output of the standard but established Canny edge detector detector [3] is generally acceptable.

The next step consists in fitting line segments to the extracted edges. To this end, our recently proposed direct split-and-merge (DSaM) algorithm [6] was applied. The rational of that algorithm is based on the assumption that 2D points may be considered collinear if the minimum eigenvalue of their covariance matrix is smaller than a predefined threshold. By these means, line segments are modeled by the major axes of ellipses with very high eccentricity capturing points which may lie in the neighborhood of the line segment due to noise or illumination changes. Through an iterative scheme, by initializing the algorithm with one cluster containing all the 2D edge points, every cluster whose ellipse does not satisfy the aforementioned threshold is split into two clusters. The procedure is repeated until there is no ellipse to split. Then, a merge process follows in order to combine neighboring clusters with collinear major axes to reduce the complexity of the model. The line segment of the corresponding cluster is modeled by the major axis, i.e. the eigenvector associated with the maximum eigenvalue. Figure 1 summarizes the DSaM algorithm. The points in red correspond to the probabilistic boundary pixels [9] of the image in figure 1(a).

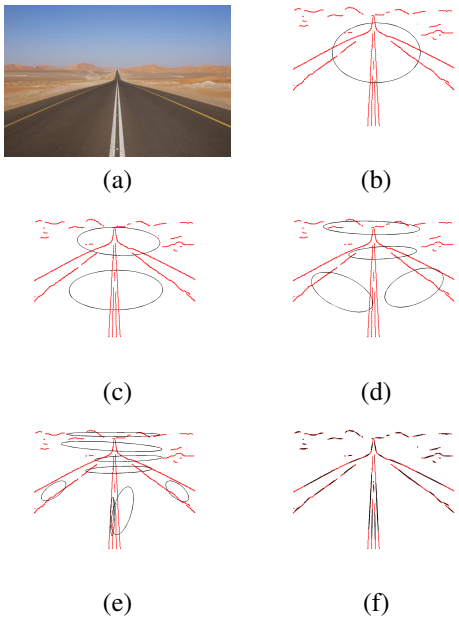

Figure 1. (a) Original color image. (b)-(e) The first four iterations of the split process and (f) the final line segments.

After the determination of the line segments in terms of the long axes of highly eccentric ellipses, the set $\mathcal{C}_{vp}$ of candidate vanishing points is constructed by computing all the pairwise intersection points between all the lines. To further improve the efficiency of the method, intersection points that lie outside the image plane could be ignored but this issue is optional and depends on the specific application. For example, in a corridor, the vanishing point lies within the image plain and intuitively the vanishing line usually appears somewhere in the viewer's horizon. On the other hand, if the algorithm is to be used by a robot navigation system, the detection of the vanishing point outside the image plane may indicate an abrupt turn.

Thence, a weight $w(\mathbf{p}) = |l_1^{\mathbf{P}}||l_2^{\mathbf{P}}|$ is assigned to each point $\mathbf{p} \in \mathcal{C}_{vp}$ which is equal to the product of the lengths $|l_1^{\mathbf{P}}|$ and $|l_2^{\mathbf{P}}|$ of the two line segments whose intersection is the candidate vanishing point $\mathbf{p}$. Thus, a candidate vanishing point produced by short segments, or one long and one short segment, is attributed with a small weight.

In the final step of our workflow, the vanishing point is computed by selecting one of the candidate points, which is achieved through a voting scheme. This step is similar in spirit to the approach presented in [10]. However, the main difference with respect to that method is that, in our algorithm, each line segment votes only for the intersection points belonging to it while in [10] a line segment votes for every candidate vanishing point (even if it does not belong to the segment).

In a voting scheme, an important factor is the size of the accumulator array bins, which in our case is a grid covering the image support $G \equiv [0,\, G_w] \times [0,\, G_h] \subset \mathbb{R}^2$. The grid is uniformly divided into equally sized cells B_i, $i = 1, ..., N$, using a scaling factor $\sigma \in (0,\, 1]$ imposing a bin size of $[\sigma G_w, \sigma G_h]$.

In order to assign weights to each candidate vanishing point present in a given bin, a kernel function centered at each array bin is employed. To this end, a 2D Π-Sigmoid kernel is applied to each bin [1], imposing thus a fuzzy concept to the borders of the cell. The support of a 1D Π-Sigmoid kernel:

$$k(x) = \frac{1}{b-a}\left[\frac{1}{1+e^{-\lambda(x-a)}} - \frac{1}{1+e^{-\lambda(x-b)}}\right], \quad (1)$$

with $\lambda > 0$, which is depicted in figure 2, approximates a uniform kernel whose borders are fuzzified in order to avoid abrupt changes. Thus, points under the plateau contribute equally with their votes while the contribution of points lying at the extremities falls off quickly but it does not become zero, depending on the value of the parameter λ. The larger the value of λ the less fuzzy the kernel borders become and consequently the edges of the kernel are very steep (fig. 2). By these means,

the capture region of the kernel allows points from the neighboring bin to contribute with a relatively low non-zero weight. A 2D Π-Sigmoid kernel $\Pi_s(\mathbf{x}; \mathbf{a}, \mathbf{b}, \lambda)$

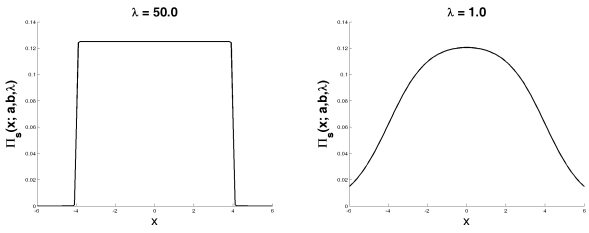

Figure 2. $\Pi_s(\mathbf{x}; \mathbf{a}, \mathbf{b}, \lambda)$ **for** $\lambda = 50$ **and** $\lambda = 1$**.**

with parameters $\mathbf{a} = (\alpha_1, \alpha_2)$, $\mathbf{b} = (b_1, b_2)$, with $\alpha_i \leq b_i$, and λ is a separable function that may be generated from the product of two $1D$ kernels:

$$\Pi_s(\mathbf{x}; \mathbf{a}, \mathbf{b}, \lambda) = \prod_{d=1}^{2} \frac{\frac{1}{1+e^{-\lambda(x_d - a_d)}} - \frac{1}{1+e^{-\lambda(x_d - b_d)}}}{b_d - a_d}.$$

Parameters $\mathbf{a}$ and $\mathbf{b}$ control the width of the kernel, while the slope λ controls the fuzziness of the kernel.

Thus, the total votes casted to cell B_i are computed by:

$$V(B_i) = \sum_{\mathbf{p} \in \mathcal{C}_{vp}} w(\mathbf{p}) \Pi_s(\mathbf{p}; \mathbf{a}_i, \mathbf{b}_i, \lambda). \quad (2)$$

The voting process is concluded by detecting the dominant cell B^* according to:

$$B^* = \arg\max_{B_i} \{V(B_i)\}. \quad (3)$$

Finally, the coordinates of the vanishing point are computed as the weighted average of all the candidate vanishing points with respect to the Π-Sigmoid kernel centered at the cell B^*:

$$\mathbf{p}^* = \frac{\sum_{\mathbf{p} \in \mathcal{C}_{vp}} w(\mathbf{p}) \mathbf{p} \Pi_s(\mathbf{p}; \mathbf{a}^*, \mathbf{b}^*, \lambda^*)}{\sum_{\mathbf{p} \in \mathcal{C}_{vp}} w(\mathbf{p}) \Pi_s(\mathbf{p}; \mathbf{a}^*, \mathbf{b}^*, \lambda^*)}, \quad (4)$$

where $\mathbf{a}^*, \mathbf{b}^*, \lambda^*$ are the parameters of the kernel corresponding to B^* in (3). Note that all the candidate points contribute to the solution. However, the importance of the points in the dominant cell is overwhelming. The steps of the method are summarized in Algorithm 1. In order to increase the robustness of the algorithm and to speed it up, line segments that are shorter than a threshold T and their orientation is close to horizontal or vertical within θ degrees are pruned. Although this rule could be omitted, it was deduced that setting the value of parameter T to 5% of the size of the diagonal of the image and $\theta = \pm 15^o$ improves significantly the performance of the method.

Algorithm 1 Vanishing point detection algorithm

- **input:** A color image.
- **output:** The coordinates of the vanishing point.
  - Detect the edges of the image.
  - Apply the DSaM algorithm [6] to detect line segments.
  - Prune segments whose length is below T and their orientation is vertical or horizontal within $\pm\theta^o$.
  - Compute the coordinates of pairwise intersection points between all segments.
  - Voting
    - Calculate the votes for each cell B_j, $j = 1, \ldots, N$ using (2).
    - Find the dominant cell using (3).
    - Compute the vanishing point using (4).

3 Experimental Results

To evaluate our method, we created two sequences, with 35 and 18 frames respectively, with a frame size of 320×240 pixels each. The images in each sequence were captured periodically by a robot moving on a specific course. The sequences represent an indoor corridor under various illumination conditions. To make the task more challenging, in the second sequence, a person walking towards the robot appears in all of the frames. Then, 5 individuals were asked to detect manually the vanishing point in each image. The ground truth vanishing point for each image was considered to be the mean point indicated by the volunteers. The standard deviation of the various vanishing points provided by the humans is 11 pixels which is approximately 3.5% of the shorter image dimension.

In order to investigate the dependence of the final result on the values of the parameters λ of the Π-Sigmoid kernel and the grid resolution tuned by σ, experiments were performed, examining the mean detection error and the execution (in MATLAB) time with respect to those parameters. Note that as the grid resolution decreases the algorithm demands more execution time because it integrates a larger number of kernels. The results are summarized in Table 1, where we have tested the behavior of the algorithm for two configurations for the parameter λ, namely $\lambda = 10$ and $\lambda = 50$. As it may be observed, the proposed method exhibits a consistent behavior since the variation of the detection error is rather small concerning the different configurations

of the parameters. The pair of parameters $\lambda = 50$ and $\sigma = 0.05$ is a good compromise between detection error and execution time. The method provides, in general, accurate results considering that its detection error is in average 2% of the image diagonal. Moreover, as the algorithm was developed in MATLAB it may be further accelerated and easily integrated in embedded systems.

Table 1. Algorithm Performance

$\lambda = 10$				
σ	0.05	0.08	0.10	0.20
Time per image (sec)	0.2	0.1	0.1	0.1
Error (pixels)	5.4	6.7	7.6	15.4
$\lambda = 50$				
σ	0.05	0.08	0.10	0.20
Time per image (sec)	0.2	0.1	0.1	0.1
Error (pixels)	5.4	6.8	7.5	15.4

We also compared our direct split-and-merge framework (DSaM) to the Hough Transform (HT), which is widely used for line detection. We kept the proposed voting scheme in both algorithms. At first, the HT needs is relatively difficult to be tuned due to the tedious task of determining the bin sizes. Moreover, the HT provided large errors (of the order of 15 pixels) and thus failed to correctly detect the vanishing point in indoor images because it was affected by spurious points.

Representative results of our method are given in figure 3. The images depict frames of an indoor corridor, with and without obstacles. The corresponding error between real and computed vanishing point is 1.54 pixels. The green lines correspond to the line segments computed by the DSaM algorithm and represent the image edges. The blue lines represent the edges contributing to the detection of the vanishing point. The red star sign depicts the vanishing point as it was computed by our method, while the yellow circle is the ground truth.

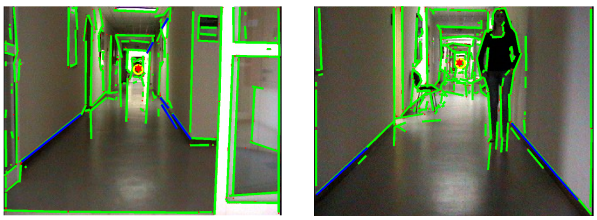

Figure 3. Representative results (the figure is better viewed in color).

4 Conclusion

In this work, a vanishing point detection algorithm based on an iterative split-and-merge scheme for line segment detection and a voting scheme was presented. Our method performs significantly better compared to a strategy adopting the widely used HT for vanishing point localization. A perspective of this work is to extend the algorithm to outdoor environments where unstructured image features and clutter make the problem more challenging.

References

[1] A. Alivanoglou and A. Likas. Probabilistic models based on the pi-sigmoid distribution. In *3rd IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 36–43, 2008. 2

[2] A. Almansa, A. Desolneux, and S. Vamech. Vanishing point detection without any a priori information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):502 – 507, 2003. 1

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986. 2

[4] J. M. Coughlin and A. Yuille. Manhattan world: compass direction from a single image by Bayesian inference. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 941–947, 1999. 1

[5] P. Denis, . H. Elder, and F. J. Estrada. Efficient edge-based methods for estimating Manhattan frames in urban imagery. In *European Conference on Computer Vision (ECCV)*, volume 2, pages 107–210, 2008. 1

[6] D. Gerogiannis, C. Nikou, and A. Likas. A split-and-merge framework for 2D shape summarization. In *7th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Dubrovnik, Croatia, 4-6 September 2011. 1, 2, 3

[7] J. Kosecka and W. Zhang. Video compass. In *European Conference on Computer Vision (ECCV)*, pages 476–491, 2002. 1

[8] B. Li, K. Peng, X. Ying, and H. Zha. Vanishing point detection using cascaded 1D Hough Transform from single images. *Patter Recognition Letters*, 33:1–8, 2012. 1

[9] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *8th International Conference on Computer Vision (ICCV)*, volume 2, pages 416–423, July 2001. 2

[10] C. Rother. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20:647–655, 2002. 1, 2

[11] J. P. Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *12th IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, September 27 - October 4, 2009. 1