Real Projective Plane Mapping for Detection of Orthogonal Vanishing Points

Markéta Dubská
http://www.fit.vutbr.cz/~idubska
Adam Herout
http://www.fit.vutbr.cz/~herout

Graph@FIT
Brno University of Technology
Czech Republic

Abstract

This paper deals with the detection of orthogonal vanishing points. The aim is to efficiently cope with the clutter edges in real-life images and to determine the camera orientation in the Manhattan world reliably. We are using a modified scheme of the Cascaded Hough Transform where only one Hough space is accumulated – the space of the vanishing points. The parameterization of the vanishing points – the "diamond space" – is based on the PClines line parameterization and it is defined as a mapping of the whole real projective plane to a finite space.

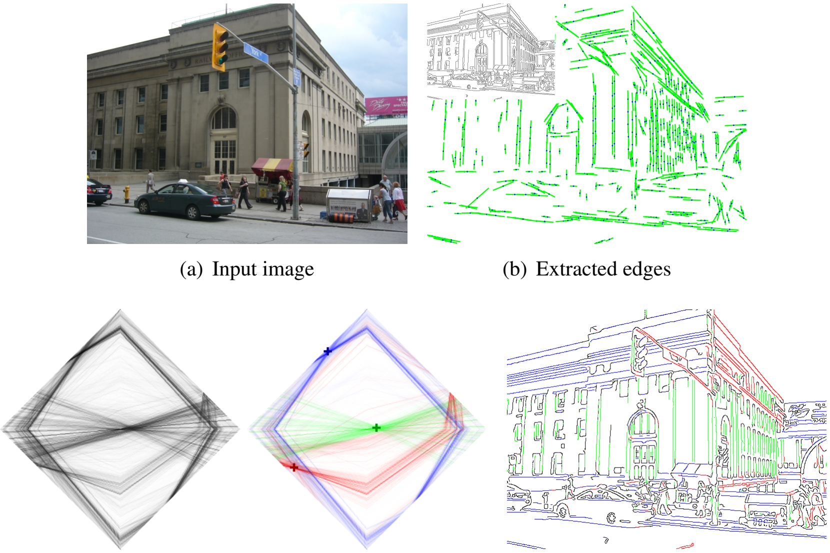
Our algorithm for detection of vanishing points operates directly on edgelets detected by an edge detector, skipping the common step of grouping edges into straight lines or line segments. This decreases the number of configuration parameters and reduces the complexity of the algorithm. Evaluated on the York Urban DB, our algorithm yields 98.04 % success rate at 10° angular error tolerance, which outperforms comparable existing solutions.

Our parameterization of vanishing points is in all aspects linear; it involves no goniometric or other non-linear operations and thus it is suitable for implementation in embedded chips and circuitry. The iterative search scheme allows for finding orthogonal triplets of vanishing points with high accuracy and low computational costs. At the same time, our approach can be used without the orthogonality constraint.

1 Introduction

Vanishing points are an important feature of most real-life images. In Manhattan worlds, they can be used for camera orientation [1, 7], camera calibration [12], scene reconstruction [6], clustering of scene edges for scene understanding [7] and other tasks [13]. At the same time, the vanishing points tend to be very stable and supported by various parts of the scene and their detection is thus robust against various distortions. Having the vanishing points reliably and efficiently detected facilitates many other computer vision tasks.

Detection of vanishing points has a long history behind. Hough transform has been a popular tool [19]; originally it used the Gauss sphere [2] and it was later bent in different ways. A generalized linear Hough transform approach was proposed by Tuytelaars *et al.* [20] as the Cascaded Hough Transform (CHT). Li *et al.* [14] used 1D Hough transform cascaded in a different manner for separate detection of in-frame and distant vanishing points.



(a) Input image

(b) Extracted edges

(c) Diamond space (right: color coded contributions)

(d) Color-coded edges

Figure 1: Detection of orthogonal vanishing points. Bottom row depicts the affiliation of edges/contributions to different orthogonal VPs by color.

Bazin *et al.* [3] argue that approaches based on the Hough transform detect vanishing points separately, failing to detect (optimal) triplets of orthogonal vanishing points which are useful for explaining images of Manhattan worlds. Bazin *et al.* suggest to look in the vicinity of an initial hypothesis described by three angles by exhaustive search or by RANSAC [4]. RANSAC and its modifications are popular for detection of vanishing points [4, 10, 15, 17]. Some works deal with infinite (i.e. ideal) vanishing points by separately treating finite and infinite ones [9]. Another popular class of algorithms is based on EM algorithms [7, 18].

Is the Hough transform obsolete for finding orthogonal vanishing points? Does it have to be inefficient and prone to uneven discretization errors? In this paper, we revisited the vanishing point detection and we propose an algorithm for detection of orthogonal triplets of vanishing points (Fig. 1). It is inspired by the CHT [20], modified so that only one limited accumulation space is used throughout the whole process. Moreover, the parameterization of lines (and consequent vanishing points) is linear: goniometric or other non-linear functions are avoided by the algorithm. This parameterization one-to-one maps the real projective plane serving as the plane of projection of the pinhole camera to a limited diamond-shaped space with an acceptable mapping contraction.

Contrary to the majority of approaches, we accumulate image edgelets (edges with gradients) directly, deliberately skipping the step of grouping them into line segments, which is costly and sensitive to several configuration parameters. The *diamond accumulator space* is used also in the second phase of the algorithm – searching for *orthogonal triplets* of vanishing points. The search procedure we propose is converging very quickly (in the experiments, a single step of the optimization procedure is used). Our method is evaluated on the York Urban DB [7] and it outperforms the existing approaches in the detection rate. At the same time, by avoiding the grouping into line segments and by fast convergence of the orthogonalization algorithm, our approach calls for an efficient and real-time implementation.

2 Proposed Real Projective Plane Mapping

Our algorithm builds upon a recent convenient parameterization of lines for the Hough transform presented by Dubská *et al.* [8] and later proven to be usable for a modified CHT scheme by Havel *et al.* [11]. This parameterization is a point-to-line-mapping (PTLM): each point is mapped to a line in the dual space and vice versa. Even for a limited original space, the dual space (Hough space) covers the whole real projective plane. Dubská *et al.* solved this problem by attaching finite subspaces of two dual spaces one to another: the second space has one of the parallel axes negated [8].

The main idea of the Cascaded Hough Transform (CHT) [20] is to first accumulate the edge points in the dual space, then detect maxima (corresponding to lines) and accumulate these maxima again into the next Hough space. Since the set of maxima of concurrent lines lie on one line, in the second Hough space, their representations intersect in one point – the vanishing point.

The PClines parameterization [8] uses a mapping between the real projective plane containing the original image and the space of parallel coordinates for the detection of lines. In the following text, subscript p is used for coordinate axes in the space of parallel coordinates and subscript c for Cartesian coordinate axes. To distinguish between points and lines, the representation $[x,y,w]$ is used for a 2D point in homogeneous coordinates (in order to deal with ideal points in infinity) and (a,b,c) denotes a line defined by equation $ax+by+c=0$.

In the parallel coordinates, point $[x,y,1]$ is represented as a line intersecting the parallel axes $\mathbf{x}_p$, $\mathbf{y}_p$ at value x or y, respectively. An ideal point $[x,y,0]$ is represented as a line parallel to both the coordinate axes. The dual statement also holds for lines: an arbitrary line is represented by exactly one point in the space of parallel coordinates. For easier referencing to objects in the parallel coordinates, we define a second Cartesian coordinate system $\mathbf{u}_c, \mathbf{v}_c$ (see Fig. 2 middle, green axes). Let d be the distance between $\mathbf{x}_p$ and $\mathbf{y}_p$. The mappings for points and lines are as follows:

$$\begin{aligned} \mathcal{S}_d^p([x,y,w]) &= (-x+y,-dw,dx), \\ \mathcal{S}_d^l((a,b,c)) &= [db,-c,a+b]. \end{aligned} \tag{1}$$

In the spirit of the CHT, the second Hough transform can be applied, i.e. points are mapped again to lines and lines to points. Let D be the distance between the parallel axes in the second parallel space. The composition of mappings (1) is then:

$$\mathcal{SS}_{dD}^p([x,y,w]) = (\mathcal{S}_D^l \circ \mathcal{S}_d^p)([x,y,w]) = [-dDw,-dx,-x+y-dw], \tag{2}$$

$$\mathcal{SS}_{dD}^l((a,b,c)) = (\mathcal{S}_D^p \circ \mathcal{S}_d^l)((a,b,c)) = (db+c,Da+Db,-dDb). \tag{3}$$

Mappings $\mathcal{SS}_{dD}$ are transformations of one infinite space to another infinite space [5]. In the case of line detection, the infinite space can be replaced by two finite dual spaces [8, 20]. Dubská *et al.* [8] flip the $\mathbf{y}_p$ axis, put it in $-d$ distance and form a twisted $\mathcal{T}$ space along the straight $\mathcal{S}$ space (1). Mapping to the $\mathcal{T}$ space is:

$$\begin{aligned} \mathcal{T}_d^p([x,y,w]) &= (x+y,-dw,dx), \\ \mathcal{T}_d^l((a,b,c)) &= [db,-c,a-b]. \end{aligned} \tag{4}$$

The CHT based on the PClines parameterization can be done by using all four combinations of the mappings. All four possible cascaded transformations for a single point using $\mathcal{S}$ and

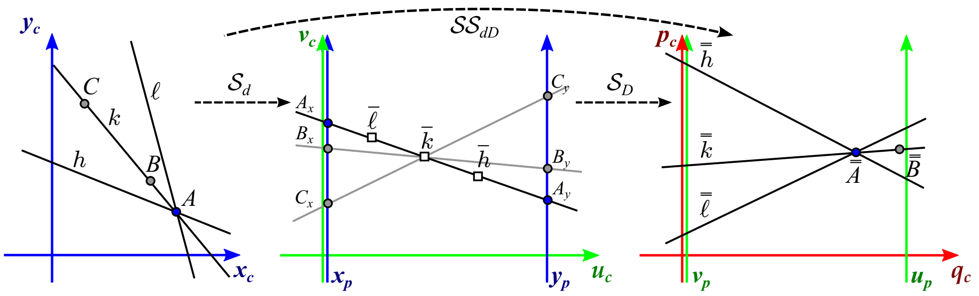


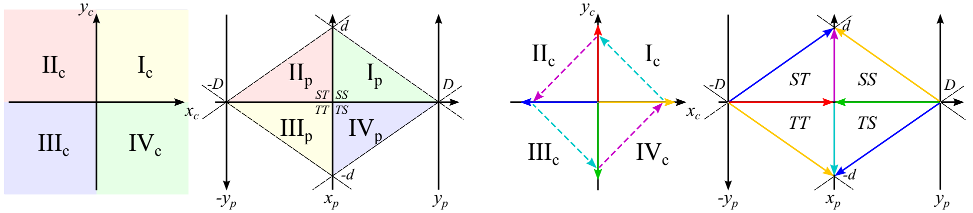
Figure 2: Two cascaded PClines transformations via the straight $\mathcal{S}$ space. **left:** Original image space with points and lines. **middle:** The same objects in parallel coordinates. A new Cartesian coordinate system is defined (green, u_c, v_c). **right:** Second transformation to parallel coordinates u_p, v_p.

$\mathcal{T}$ spaces are the following:

$$\begin{aligned}
\mathcal{SS}^p_{dD}([x,y,w]) &= (\mathcal{S}^l_D \circ \mathcal{S}^p_d)([x,y,w]) = [-dDw, -dx, -x+y-dw],\\
\mathcal{ST}^p_{dD}([x,y,w]) &= (\mathcal{S}^l_D \circ \mathcal{T}^p_d)([x,y,w]) = [-dDw, -dx, -x+y+dw],\\
\mathcal{TS}^p_{dD}([x,y,w]) &= (\mathcal{T}^l_D \circ \mathcal{S}^p_d)([x,y,w]) = [-dDw, -dx, x+y-dw],\\
\mathcal{TT}^p_{dD}([x,y,w]) &= (\mathcal{T}^l_D \circ \mathcal{T}^p_d)([x,y,w]) = [-dDw, -dx, x+y+dw].
\end{aligned} \tag{5}$$

It should be noted that these four mappings transform the whole projective plane into a finite space (composed of four finite parts, each for one transformation) – see Fig. 3(a).

Consider, for example, transformation (2) and points from quadrant IV$_c$ (Fig. 3(a)). A regular point, i.e. $P = [x, y, 1]$, $x \geq 0$, $y \leq 0$, is transformed to $\mathcal{SS}^p_{dD}(P) = [-dD, -dx, -x + y - d]$. Because $d, D > 0$ (distances between the parallel axes), the position of the point's



(a) Mapping of quadrants.

(b) Arrangement of triangular subspaces.

Figure 3: Subspaces of the mapping. (a) *left:* Quadrants of the original infinite Cartesian space. *right:* Quadrants of the PClines space (two attached spaces of parallel coordinates). (b) *left:* After cascading two mappings, the target quadrants are bounded, triangular. *right:* These triangular areas can be rearranged and attached one to another at edges which represent identical points in the original space. The dashed lines stand for the ideal line of the complete real projective plane (x_c, y_c space is infinite).

image can be derived:

$$\begin{aligned} d,D > 0,\ x \geq 0 &\Rightarrow -dD < 0,\ -dx \leq 0 \\ x \geq 0,\ y \leq 0,\ d > 0 &\Rightarrow -x+y-d < 0 \\ &\Rightarrow \frac{-dD}{-x+y-d} > 0 \\ &\Rightarrow \frac{-dx}{-x+y-d} \geq 0. \end{aligned} \tag{6}$$

This proves that all points from $\mathrm{IV_c}$ are transformed into $\mathrm{I_p}$ quadrant by the $\mathcal{SS}$ transformation. And what is more, all these points lie in a single half-plane defined by line $\ell_b = (d,D,-dD)$ (the line passing through points $[D,0]$ and $[0,d]$):

$$(d,D,-dD)\cdot\left[\frac{-dD}{-x+y-d},\frac{-dx}{-x+y-d},1\right] = \frac{-dDy}{-x+y-d} \leq 0. \tag{7}$$

Ideal points $P = [x,y,0],\ x \geq 0,\ y \leq 0$ with images $\mathcal{SS}^{p}_{dD}(P) = [0,-dx,-x+y]$ lie on the line segment between points $[0,0]$ and $[0,d]$.

To conclude, **all** points – both regular and ideal – from $\mathrm{IV_c}$ quadrant are transformed by $\mathcal{SS}$ to the intersection of $\mathrm{I_p}$ quadrant and the closed lower half-plane defined by line $dx+Dy-dD=0$. Proofs for mappings (5) are analogous. It should be only pointed out that $\mathrm{I_c}$ quadrant is mapped to a part of $\mathrm{III_p}$ quadrant, $\mathrm{II_c}$ to $\mathrm{II_p}$ and $\mathrm{III_c}$ to $\mathrm{IV_p}$ – see Fig. 3(a).

These four parts, each mapped by a different mapping ($\mathcal{SS},\mathcal{ST},\mathcal{TS}$ or $\mathcal{TT}$) can be attached because images of the axes $\mathbf{x}_c,\mathbf{y}_c$ and the ideal line always lie on the borders of two segments (Figure 3(b)). Therefore, point mappings between the original plane and the joined diamond space are:

$$[x,y,w]_o \rightarrow [-dDw,-dx,\operatorname{sgn}(xy)x+y+\operatorname{sgn}(y)dw]_d \tag{8}$$

$$[x,y,w]_d \rightarrow [Dy,\operatorname{sgn}(x)dx+\operatorname{sgn}(y)Dy-dDw,x]_o, \tag{9}$$

where sgn stands for a non-zero signum function. However, in the joined space, the image of a straight line is not a line anymore. The result of the mapping is a polyline whose number of segments depends on the number of quadrants the line passes. The sequence of endpoints defining the polyline is in Eq. (10). When a line passes through just two quadrants (vertical lines, horizontal lines, lines through the origin), one segment always degenerates to a point (i.e. two adjacent points from (10) are identical). Singular cases, lines $x=0$, $y=0$ and the ideal line, are handled according to Fig. 3(b).

$$\begin{aligned} &\alpha = \operatorname{sgn}(ab), \quad \beta = \operatorname{sgn}(bc), \quad \gamma = \operatorname{sgn}(ac) \\ &(a,b,c)_o \rightarrow \left[\frac{\alpha a}{c+\gamma a},\frac{-\alpha c}{c+\gamma a}\right],\left[\frac{b}{c+\beta b},0\right],\left[0,\frac{b}{a+\alpha b}\right],\left[\frac{-\alpha a}{c+\gamma a},\frac{\alpha c}{c+\gamma a}\right] \end{aligned} \tag{10}$$

The target space (Fig. 3) has a diamond shape. In the further text, we will refer to the particular accumulation space as the **diamond space**. It should be noted that in practical implementations, the triangular quadrants can be rearranged in the computer memory so that the space is a horizontal rectangle for better memory utilization.

3 Detection of Orthogonal Vanishing Points

Let us recall that the previous section introduced a mapping which transforms an infinite plane (including ideal points and the ideal line) to a bounded diamond space. This parameterization is per partes linear (a line or line segment is transformed to a polyline of three or two segments) and it is defined by simple linear operations (10). This mapping was constructed by direct application of cascading (as in the CHT) on the PClines parameterization [8]. Note that this mapping maps points of the original plane to *points* in the dual space.

We propose to directly apply this parameterization on short line segments – *edgelets* – detected by a suitable detector in the input image (ellipse fitting to clusters of pixels is used in the experiments). The input image is placed into the center of the projective plane and its size is normalized so that it fits into a fixed interval around the origin $(-\mu, \mu)$ (see Fig. 5 for an example). We deliberately skip the common step of grouping these edgelets into more refined lines and accumulate evidence of edgelets directly into the diamond space. The highest peak in the diamond space is the most voted vanishing point; edgelets contributing to it are removed from the accumulation space and the second highest response is sought for and the procedure is repeated in order to find more vanishing points (Fig. 1). By using Eq. (9), each detected diamond space maximum is projected to the image plane; then, by using the camera internal parameters, it is reprojected to the world coordinates $\bar{\mathbf{v}}$.

In a Manhattan world scenario, three projections of *orthogonal* vanishing points are of interest. Three non-orthogonal candidates $(\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \bar{\mathbf{v}}_3)$ are found by using the above described procedure (Fig. 4). An orthogonalized triplet $(\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3)$ is computed as:

$$\begin{aligned}
\hat{\mathbf{v}}_1 &= \bar{\mathbf{v}}_1 \\
\hat{\mathbf{v}}_2 &= [\bar{\mathbf{v}}_i]_{\hat{\mathbf{v}}_1}, \quad i \in \{2,3\} \text{ s.t. } \bar{\mathbf{v}}_i \cdot [\bar{\mathbf{v}}_i]_{\hat{\mathbf{v}}_1} \text{is maximal; } [\mathbf{v}]_{\mathbf{n}} = \mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n} \\
\hat{\mathbf{v}}_3 &= \hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2
\end{aligned} \tag{11}$$

The Manhattan vanishing points (orthogonal triplet) are then found by optimization:

$$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) = \underset{\dot{\mathbf{v}}_1 \perp \dot{\mathbf{v}}_2 \perp \dot{\mathbf{v}}_3}{\arg\min} \left\{ \|(\dot{\mathbf{v}}_1, \dot{\mathbf{v}}_2, \dot{\mathbf{v}}_3), (\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \bar{\mathbf{v}}_3)\| - \alpha \left(\frac{h(\dot{\mathbf{v}}_1)}{h(\bar{\mathbf{v}}_1)} + \frac{h(\dot{\mathbf{v}}_2)}{h(\bar{\mathbf{v}}_2)} + \frac{h(\dot{\mathbf{v}}_3)}{h(\bar{\mathbf{v}}_3)} \right) \right\}, \tag{12}$$

where $\|(\dot{\mathbf{v}}_1, \dot{\mathbf{v}}_2, \dot{\mathbf{v}}_3), (\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \bar{\mathbf{v}}_3)\|$ stands for the sum of angular differences between the vanishing points and the initial detections, $h(\mathbf{v}_i)$ is the response from the diamond Hough space and α is a balancing parameter; in our experiments $\alpha = 1$. Note that all vanishing point coordinates must be normalized.

The minimization (12) is done by iterative search, starting with three different initial orthogonalized triplets of vanishing points $(\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3)$, produced by using different members

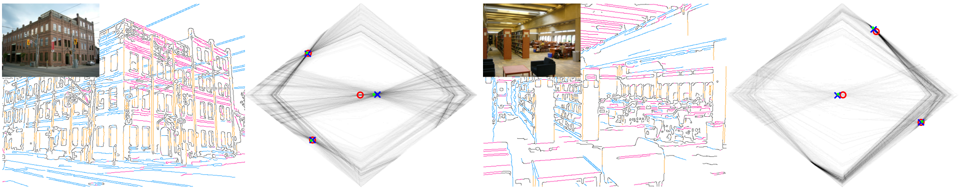


Figure 4: Two examples of the diamond accumulation space (best viewed on screen). **red circles:** Non-orthogonal vanishing points $(\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \bar{\mathbf{v}}_3)$. **blue crosses:** Orthogonal triplet of vanishing points. **green dots:** Ground truth vanishing points.

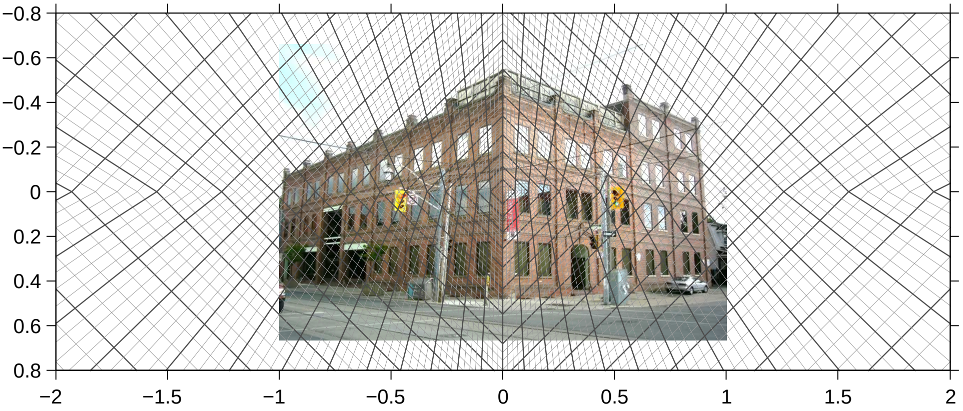


Figure 5: Visualization of the accumulators ("pixels") from the diamond space projected to the image plane. Image normalization $\mu = 1$.

of the original, non-orthogonal detection selected as the first vanishing point $\bar{\mathbf{v}}_1$ in Eq. (11). In the experiments below, only a *single* search iteration is performed; it turned out that one iteration is enough for finding a sufficiently accurate solution.

4 Experimental Results

This section shows the results of measurements of the discretization accuracy of the accumulation space (Sec. 4.1) and the performance of the detection of orthogonal triplets of vanishing points on the York Urban Dataset [7] (Sec. 4.2). The MATLAB source codes and evaluated data on the YUD can be downloaded from our website[1].

4.1 Accumulator Space Sampling Error

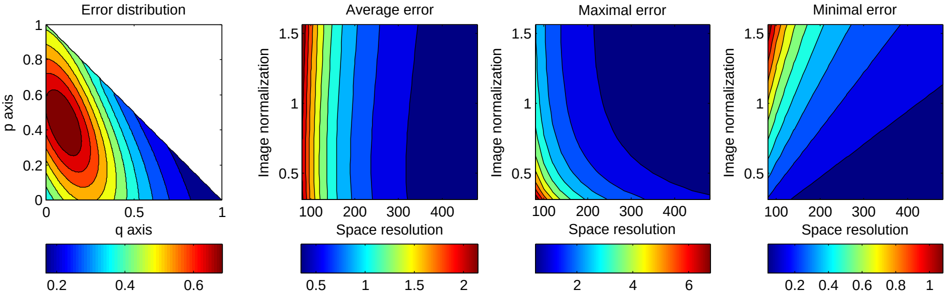
Figure 5 shows the discretization of the accumulator space in the original image. The shape of the accumulators or "pixels" is different in the **x** and **y** axis because the diamond space was made with the **y** axis inverted – this is an arbitrary selection.

By choosing a different normalization of the input image and/or the resolution of the accumulation space, it is possible to find suitable parameters for bounding the final discretization error. The discretization is defined by the space resolution, image normalization and by the camera parameters. A meaningful error metric for an accumulator cell is the max deviation between all vectors projected to the cell and the representative vector computed for the given cell. This error is plotted in Fig. 6.

4.2 Vanishing Point Detection Accuracy

The detection accuracy is evaluated on the York Urban Database [7], consisting of 102 images, each with three orthogonal ground truth vanishing points. The parameters (space resolution, image normalization) were obtained from the training set (half of the DB) and the

[1] http://medusa.fit.vutbr.cz/pclines/



(a) Error Distribution (b) Error Dependence

Figure 6: Error of the accumulator space: $d = D = 1$, camera and image parameters from the York Urban DB [7]. (a) Distribution of the angular error for a quarter of the diamond space with resolution 320 and normalization $\mu = 1$. (b) Dependence of average/maximal/minimal error in the diamond space on the space resolution and image normalization. The space resolution is critical for the average error and by using a different normalization it is possible to bound the lower and upper error.

method	[18]	[9]	[16]	[15]	[17]	[10]	**ours**
correct VP [%]	94.35	100*	80.7/84.6/79.5/69.3	99.3**	90.03	93	88.04/**98.04**
avg. err [°]	3.5	1.63	-	-	<3	-	1.87/**1.41**

Table 1: Detection rate at $< 10°$ angular error tolerance in comparison with previous works. For our method, the value for directly detected VPs $(\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \bar{\mathbf{v}}_3)$ are reported, followed by the orthogonalized VPs according to Sec. 3 in bold. * Column [9] should be omitted from fair comparison because the authors seem to apply the tolerance on individual angles (pitch, yaw, roll), not on the overall angular difference, which is the case of other works (and our evaluation). ** Column [15] should also be treated lightly because the authors feed the detector only with edges user-annotated as belonging to one of the vanishing points (no outliers), while other works are working with all the edges in the image, including clutter.

evaluation was done on the complete set in order to be comparable with previous works. We used two means of evaluation popular in the literature: detection rate with 10° angular error tolerance [9, 10, 15, 16, 17, 18], Table 1, and cumulative histogram of the count of correctly recognized vanishing points based on the angular error tolerance [2, 13, 14], Figure 7.

5 Conclusions

We presented a new method for detecting orthogonal or non-orthogonal vanishing points in real-world images. The method is based on the Cascaded Hough Transform, but the procedure is modified so that there is only one accumulation space – the diamond space. This simplifies the detection process and lowers the number of various thresholds and configuration parameters; also, this parameterization is linear. Then, we proposed a quickly converging iterative algorithm for detection of orthogonal triplets of vanishing points. This refinement procedure works directly on the accumulated diamond space by simple sampling and it is

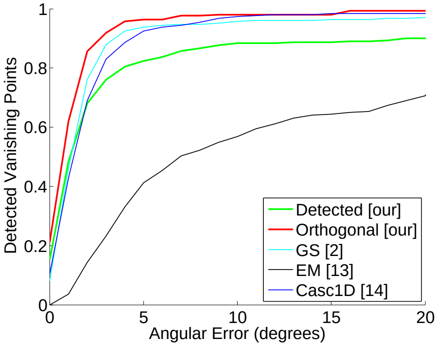


Figure 7: Cumulative histogram of the number of correctly detected vanishing points. *Horizontal axis:* average angular error of the detected vanishing points from the ground truth. *Vertical axis:* fraction of vanishing points detected with the given error tolerance. **green:** Our detection algorithm without the orthogonalization, i.e. $(\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \bar{\mathbf{v}}_3)$ in Sec. 3. **red:** Our detection after search for orthogonal triplet of vanishing points, i.e. result of Eq. (12). **GS**, **EM** and **Casc1D** are algorithms used in [2, 13, 14].

thus very efficient. The MATLAB implementation of our parameterization is publicly available together with our results.

We evaluated the proposed methods on the York Urban Dataset. The results show that our algorithm outperforms existing solutions that are using the same evaluation methodology. At the same time, it is computationally efficient – the algorithmic steps are simple, the accumulation works directly with the edgelets (no clustering to line segments is used) and the iterative search converges quickly (a single iteration on three alternative initial hypotheses was used in the tests).

Since the algorithm is using a linear parameterization of the Hough space, it requires no goniometric or other non-linear operations and it can be formulated in a fixed-point arithmetic. It is thus predestined to be cheaply computed at interactive frame rates and to prevail on embedded processors and programmable circuitry. We are working on its experimental implementation for this class of devices. Another direction of undergoing exploration is including depth data – since the simple and cluttered edgelets are able to accumulate to correct vanishing points, normals of depth patches could be used in a similar manner.

Acknowledgements

This work was supported by the TACR grant TA01010931, by the CEZMSMT project IT4I - CZ 1.05/1.1.00/02.0070, and by project V3C, TE01020415. We would like to thank our colleagues: to Jan Navratil for suggesting the name “diamond space” and Roman Juranek for help with MATLAB.

References

[1] Matthew E. Antone and Seth Teller. Automatic recovery of relative camera rotations for urban scenes. In *Proc. CVPR*, 2000.

[2] Stephen T. Barnard. Interpreting perspective images. *Artif. Intell.*, 21(4):435–462, November 1983. ISSN 0004-3702. doi: 10.1016/S0004-3702(83)80021-6.

[3] J.-C. Bazin, C. Demonceaux, P. Vasseur, and I. Kweon. Rotation estimation and vanishing point extraction by omnidirectional vision in urban environment. *IJRR*, 31, 2012.

[4] Jean-Charles Bazin and Marc Pollefeys. 3-line RANSAC for orthogonal vanishing point detection. In *Proc. IROS*, 2012.

[5] Prabir Bhattacharya, Azriel Rosenfeld, and Isaac Weiss. Point-to-line mappings as Hough transforms. *Pattern Recognition Letters*, 23(14):1705–1710, 2002.

[6] R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes. In *Proc. BMVC*, 1999.

[7] Patrick Denis, James H. Elder, and Francisco J. Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *Proc. ECCV*, 2008.

[8] Markéta Dubská, Adam Herout, and Jiří Havel. PClines – line detection using parallel coordinates. In *Proc. CVPR*, 2011.

[9] W. Elloumi, S. Treuillet, and R. Leconge. Tracking orthogonal vanishing points in video sequences for a reliable camera orientation in manhattan world. In *CISP*, 2012.

[10] Wolfgang Förstner. Optimal vanishing point detection and rotation estimation of single images from a legoland scene. In *ISPRS*, 2010.

[11] Jiří Havel, Adam Herout, and Markéta Dubská. Vanishing points in point-to-line mappings and other line parameterizations. *Pattern Recognition Letters*, 34:703–708, 2013.

[12] Ciarán Hughes, Patrick Denny, Martin E. Glavin, and Edward Jones. Equidistant fish-eye calibration and rectification by vanishing point extraction. *PAMI*, 2010.

[13] Jana Košecká and Wei Zhang. Video compass. In *ECCV*. 2002.

[14] Bo Li, Kun Peng, Xianghua Ying, and Hongbin Zha. Vanishing point detection using cascaded 1D hough transform from single images. *Pattern Recognition Letters*, 2012.

[15] F.M. Mirzaei and S.I. Roumeliotis. Optimal estimation of vanishing points in a manhattan world. In *ICCV*, 2011. doi: 10.1109/ICCV.2011.6126530.

[16] M. Nieto and L. Salgado. Non-linear optimization for robust estimation of vanishing points. In *ICIP*, 2010. doi: 10.1109/ICIP.2010.5652381.

[17] Marcos Nieto and Luis Salgado. Real-time robust estimation of vanishing points through nonlinear optimization. In *RTIVP*, 2010.

[18] Marcos Nieto and Luis Salgado. Simultaneous estimation of vanishing points and their converging lines using the em algorithm. *Pattern Recognition Letters*, 2011.

[19] Jefferey A. Shufelt. Performance evaluation and analysis of vanishing point detection techniques. *IEEE Tran. Pattern Analysis and Machine Intelligence*, 21:282–288, 1999.

[20] Tinne Tuytelaars, Marc Proesmans, Luc Van Gool, and Esat Mi. The cascaded Hough transform. In *Proceedings of ICIP 1998*, pages 736–739, 1998.