STEVENS INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

VANISHING POINT DETECTION IN HOUGH TRANSFORM SPACE

Nikhil Mahajan
10374674

Prof. Dr. Hong Man

CPE-645 : Image Processing and Computer Vision SPRING 2013

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Vanishing Points has remained an active field of research for various image processing objectives, viz. robot localization, typography etc. Hough Transform space has been proposed for detecting vanishing points after every pixel of image is scanned & edge detected. Peaks of Hough Image (which corresponds to line segments) are then extracted, using L_p-norms in linear space and further, K-Means algorithm is implemented to form clusters.

I. INTRODUCTION

In three-dimensional (3D) space, depending on the relative pose of the lines to the image plane, parallel lines may or may not be parallel equivalent to those in 2D image space. A vanishing point is the apparent convergence of parallel lines in a real space, located at infinity or otherwise has a finite location on the image plane. Vanishing points which lie on the same plane in the image defines a line, the so-called vanishing line.

Figure 1. shows three vanishing points and vanishing lines of a cube. Finite vanishing point is defined by a point on the image plane and a vanishing point at infinity is defined by a direction on the image plane. If the camera geometry is known, each vanishing point corresponds to an orientation in the scene and vice versa

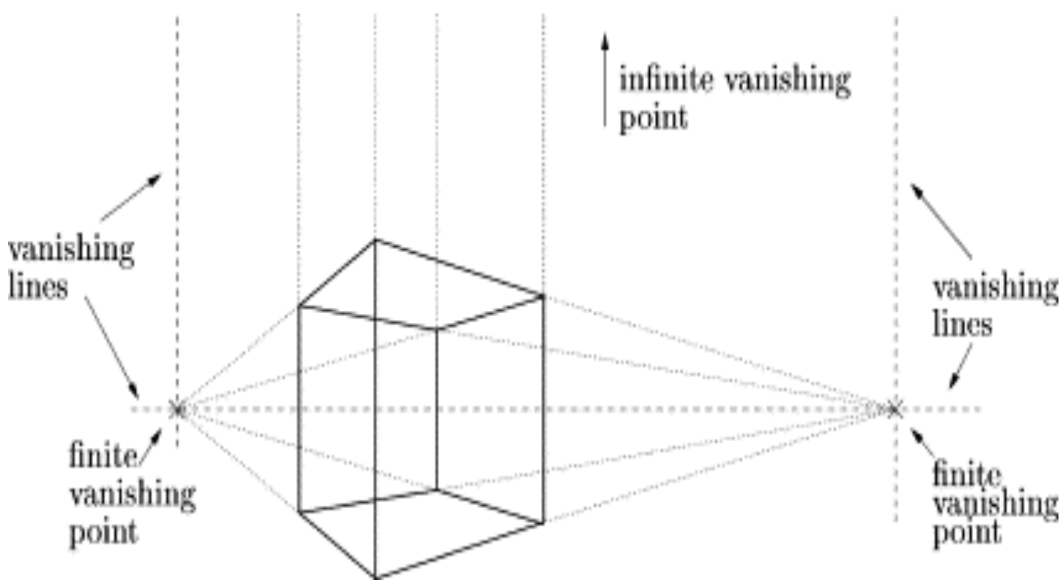


Fig. 1. The three vanishing points and vanishing lines of a cube.

Analysis of vanishing points provides strong cues for inferring information about the three-dimensional (3D) geometry of the scene depicted such as depth and object dimension. Analyzed cues have been successfully exploited for computing internal parameters of a camera[1] [2], characterize the direction of a road and automatically drive a vehicle[3], estimate distances to the retrieve object surfaces and their orientation in respect to the viewer [4].

Most of the existing methods to detect automatically these vanishing points stand on the use of Hough Transform, explicitly or not[5]. In this work, an analytical method to determine the vanishing point in the Hough Parameters space is presented. Hough-space method detects rectangles in an image by considering relations among the Hough peaks imposed by the underlying geometry of a rectangle.

---

[1] C. Rother, "A New Approach for Vanishing Points Detection in Architectural Environments," Image and Visual Computing ,vol. 20, pp. 647-656,September 2002.

[2] B. Caprile and V. Torre, "Using Vanishing Points for Camera Calibration," International Journal of Computer Vision vol. 4, pp. 127-140,February 1990

[3] Shih-Ping Lioua and Ramesh C. Jaina, "Road Following Using Vanishing Points," Computer Vision, Graphics and Image Processing, vol. 39, pp.116-130, January 1987.

[4] R. Lodola, A. Mecocci and U. Salvatore, "Outdoor Scenes Interpretation Suitable for Blind People navigation," Proc. of Image Proc. and Its Application, Edimburgh, 4-6 July 1995, pp. 256-260

[5] Eyelyne Lutton, Henri Maitr and Jaime Lopez-Karhe, "Contribution to the Determination of Vanishing Points using Hough Transform," IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 16, No. 4, pp. 430-442, April 1994.

II. HOUGH TRANSFORM

The Hough Transform[6] is a powerful method for detecting linear structures in images. For a binary image, associated with each pixel (x, y) is a continuous set of all possible lines of infinite extent, which could pass through that point. Such lines are parameterized by equation;

$\rho = x \cos(\theta) + y \sin(\theta)$; as illustrated.

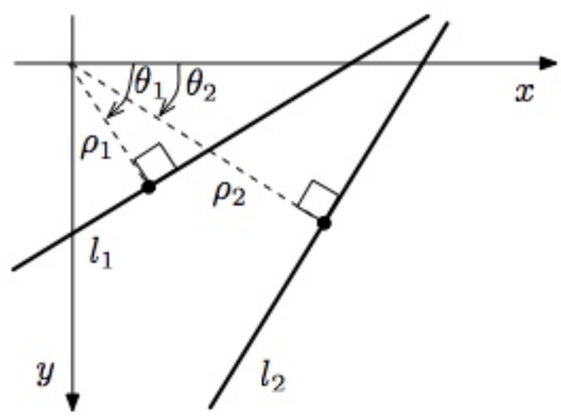


Fig 2. Illustration of Hough parameters as they relate to lines in image space. Here line l1 is uniquely identified by its perpendicular to the origin, which has length ρ_1, and angle from x-axis given by θ_1. Similarly for l2, length ρ_2 and angle from x-axis given by θ_2.

Without any loss of information, θ can be evaluated over a range [-π/2, π/2]. For the original image having Width W & Height H, ρ can range over $[-\rho_{max}, \rho_{max}]$, where $\rho_{max} = \sqrt{W2+H2}$. The Hough space is defined by a finite rectangle $[-\rho_{max}, \rho_{max}]$ x [-π/2, π/2],

[6] Rafael C. Gongalez, Richard E. Woods, "Digital Image Processing", Third Edition, Pearson, ISBN 978-81-317-1934-3

split into rectangular bins. For each non-zero point (x, y) in the original binary image, the Hough transform draws a “sinusoid” in Hough space by incrementing the bin counts along the discretized curve $\rho = x \cos(\theta) + y \sin(\theta)$ by one.

These bins are labeled by a set of parameters, correspond to possible lines in image space. Conversely, if several points in image space lie along a line the bin counts for (ρ, θ) correspond to the line through these points will be high. Peaks in the Hough domain correspond to a high likelihood of a line in the image domain.

The Hough procedure is readily extended to gray-level images, where the bins are incremented in proportion to the gray-level value of the pixel in original image space. In order to detect strong lines using the Hough transform, it is profitable to pre-process the image, detecting edges using an edge detection technique viz. Sobel, Prewitt etc.

III. EDGE DETECTION

Edge detection is the approach used most frequently for segmenting images based on abrupt (local) changes In intensity i.e. significantly reduce the amount of data in an image, while preserving the structural properties to be used for further processing.

I. Canny Edge Detector

Introduced by John F Canny [JFC][7] in 1986, the canny edge detection algorithm consists of following basic steps:

i. Smoothing: Blurring of the image to remove noise.
ii. Finding gradients: The edges should be marked where the gradients of the image has large magnitudes.
iii. Non-maximum suppression: Only local maxima should be marked as edges.
iv. Double thresholding: Potential edges are determined by thresholding.
v. Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

This method has an edge over basic edge detection methods by taking into account factors such as image noise and nature of edges themselves.

[7] John Canny. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6):679–698, Nov. 1986.

IV. VANISHING POINT DETECTION

A. One Vanishing point

The Hough transform procedure generates a set of n Hough peaks (ρ_1,θ_1), (ρ_2,θ_2),…. (ρ_n, θ_n) and their bin counts interpreted as weights $w_1, w_2, \ldots w_n$. A large weight w_k, for instance, corresponds to a large Hough bin count at point (ρ_k, θ_k) giving credence to the line parameterized by; $\rho_k = x \cos(\ _k) + y \sin(\ _k)$ in proportion to its length in the original image. Suppose n lines are to intersect at the same point, or nearly the same point (x_0, y_0), we can extract that intersection point by solving the approximation problem:

$$\begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \\ \vdots & \vdots \\ \cos(\theta_n) & \sin(\theta_n) \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \approx \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \end{bmatrix}$$

In other words, we find the best-fit sinusoid parameterized by (x_0, y_0) directly in the Hough domain by forming a matrix $A \in R^{nx2}$ and vector $\rho \in R^n$, and seek to make the difference $\| Ax - \rho \|_2$, $x \in R^2$ as small as possible. Formally, x is a solution to the optimization problem which is convex for any L_p-norm ($p \geq 1$) and is readily solved for several norms and choices of p viz. L_1, L_2 etc.

a. *L_1 norm:* In this case, no closed-form solution for x can be given, however problem is readily reformulated as an inequality constrained linear program, for which fast interior point iterative methods (e.g., log-barrier, primal-dual) exist.

b. L_2 *norm:*

When A is full rank, the optimal x satisfies the normal equations, $A^T x = A^T \rho$, i.e. $x = (A^TA)^{-1} \rho$. If the two columns of A are given by $a_1, a_2 \in R^n$, then A^TA is 2×2 symmetric with elements:

$$A^T A = \begin{bmatrix} a_1^T a_1 & a_1^T a_2 \\ a_2^T a_1 & a_2^T a_2 \end{bmatrix}$$

for which there is an explicit formula for inverse, thus the optimal x is

$$\begin{aligned} x &= (A^T A)^{-1} A^T \rho \\ &= \frac{1}{\det(A^T A)} \begin{bmatrix} a_2^T a_2 & -a_1^T a_2 \\ -a_2^T a_1 & a_1^T a_1 \end{bmatrix} \begin{bmatrix} a_1^T \rho \\ a_2^T \rho \end{bmatrix} \end{aligned}$$

c. *Quadratic* W *-norm:*

The weighted least-squares setting, where quadratic norm $\| \cdot \|_W$ induced by the n × n matrix $W = \mathrm{diag}(w_1, w_2, ..., w_n)$ defined as,

$$\|x\|_W = (x^T W x)^{1/2} = \|W^{1/2} x\|_2$$

is equivalent to least-squares L_2-norm problem. Here, one interprets the weights w_j as the degree to which one should incorporate the line parameterized by (ρ_j, θ_j) contributing to the intersection point. The Hough transform bin counts serve as the weights, with high bin counts corresponding to large weights.

B. Multiple Vanishing Point

If an image consists of multiple vanishing points, K-mean [8] algorithm can be implemented to cluster the line intersections. K means cluster the intersections with a good direction-to vanishing point but even if vanishing point is visible in the original image, cluster centroids are adversely “pulled” away from the purported vanishing point by stray lines intersecting with existing lines, and thus by themselves provide poor estimates of the vanishing point location. The stray lines create extra pairwise intersections, changing the cluster centroids’ locations.

As a result, we classify n lines corresponding to peaks in hough transformed image rather than n(n-1)/2 points, as belonging to one of K vanishing points. The Algorithm 1, ascertains set of lines in the Hough domain all intersect at the same point in the image domain as shown in figure 3.

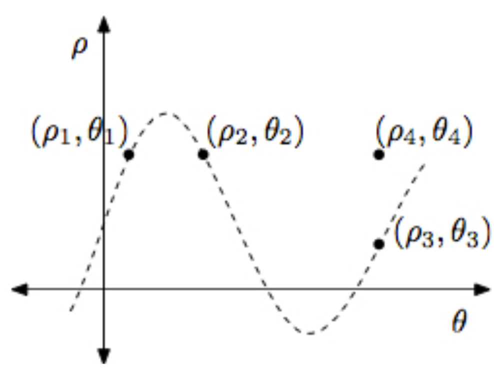


Fig.3 Visualization in the Hough domain.

---

[8] Richard O. Duda, Peter E. Hart, David G. Stork, “Pattern Classification, 2nd Edition”, Wiley, ISBN: 978-0-471-05669-0

In figure 3, Points $(\rho_1, \theta_1),\ldots,(\rho_4, \theta_4)$ in Hough space corresponds to lines in image space. Here, $(\rho_1,\theta_1),\ldots,(\rho_3,\theta_3)$ lie on the same sinusoid, whose best-fit parameters (x, y) is in accordance with Equation $\rho = x\cos(\theta) + y\sin(\theta)$. (ρ_4,θ_4) is not on the same sinusoid, thus does not give credence to a mutual intersection.

Algorithm 1: K-Means ~ Best L_2

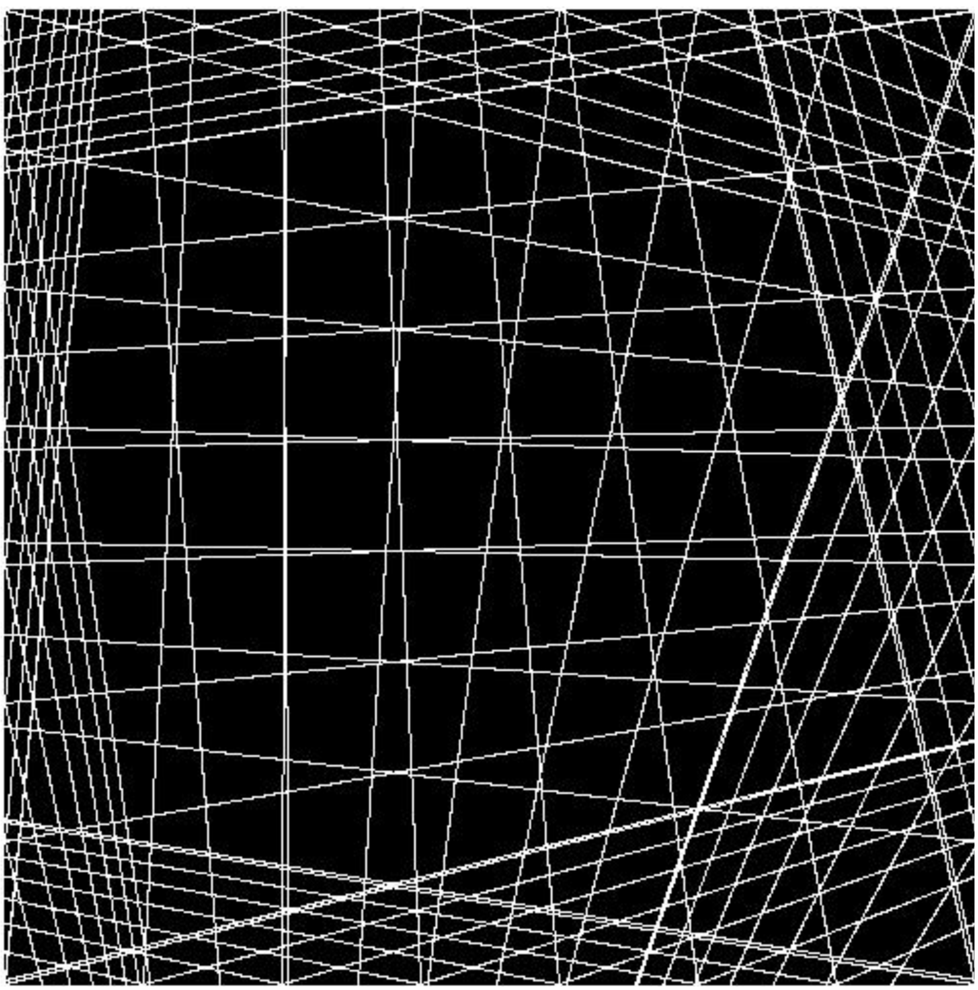
Input: number of vanishing points: K,
input lines: $(\rho_1, \theta_1), \ldots, (\rho_n, \theta_n)$.

$\mathcal{L} \leftarrow \{1, 2, \ldots, n\}$
$best\_err \leftarrow \infty$
$\mathcal{V} \leftarrow \emptyset$
for all pairwise disjoint partitions $S_1, S_2, \ldots, S_K \subseteq \mathcal{L}$, with $S_i \cap S_j = \emptyset, i \neq j$, and $\bigcup_{j=1}^{K} S_j = \mathcal{L}$ **do**
  for $j = 1$ to K **do**
    $A_j \leftarrow$ rows S_j of A in equation (1).
    $\rho_j \leftarrow$ rows S_j of ρ in equation (1).
    $x_j^* \leftarrow \arg\min_{x \in \mathbb{R}^2} \|A_j x - \rho_j\|_2$
  end for
  $cum\_err \leftarrow \sum_{j=1}^{n} \|A_j x_j^* - \rho_j\|_2$
  if $cum\_err < best\_err$ **then**
    $best\_err \leftarrow cum\_err$
    $\mathcal{V} \leftarrow \{x_1^*, x_2^*, \ldots, x_K^*\}$
  end if
end for
return $\mathcal{V}$ as the K vanishing point estimates.

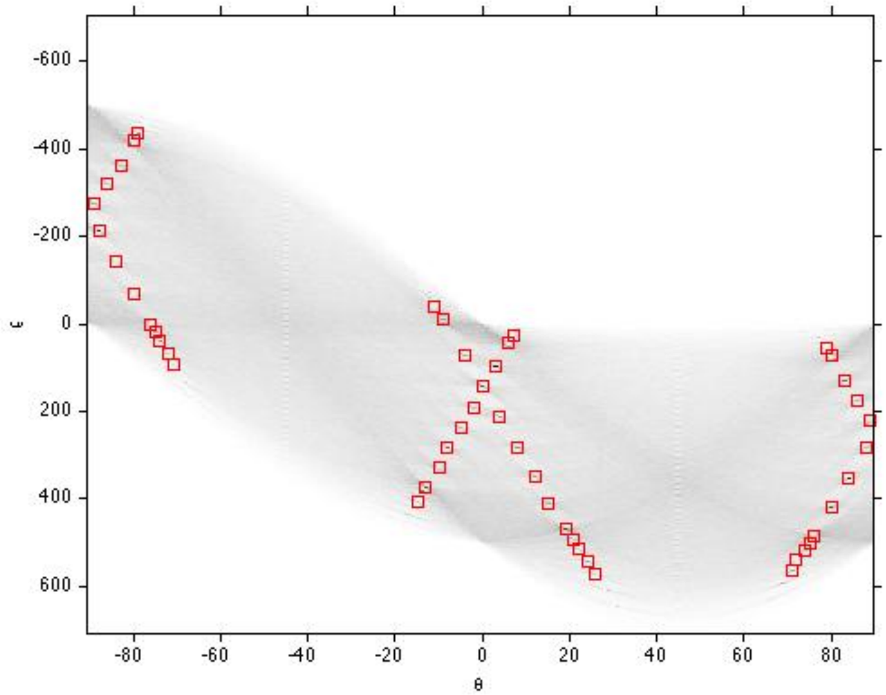
V. RESULTS

Despite the fact, Algorithm 1 is exhaustive; its execution time is small in comparison to Hough transform line extraction. Also the Hough line fidelity, which is heavily dependent on threshold parameters in the edge detection and Hough peak finding steps, has a big impact on the extending vanishing point.

a. 500x300 image generated with Vanishing Points



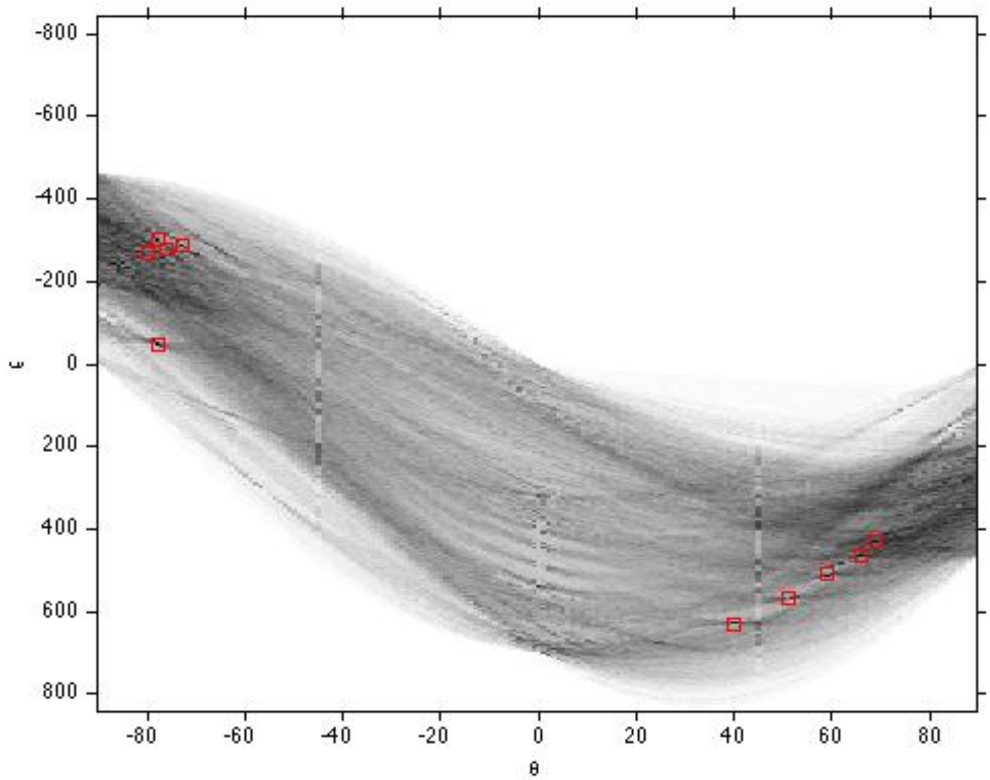
b. Hough Transform with largest peaks denoted by red color.



c. Image – Gaussian Smoothed (Gray-scale - Original) & Canny Edge Detection.

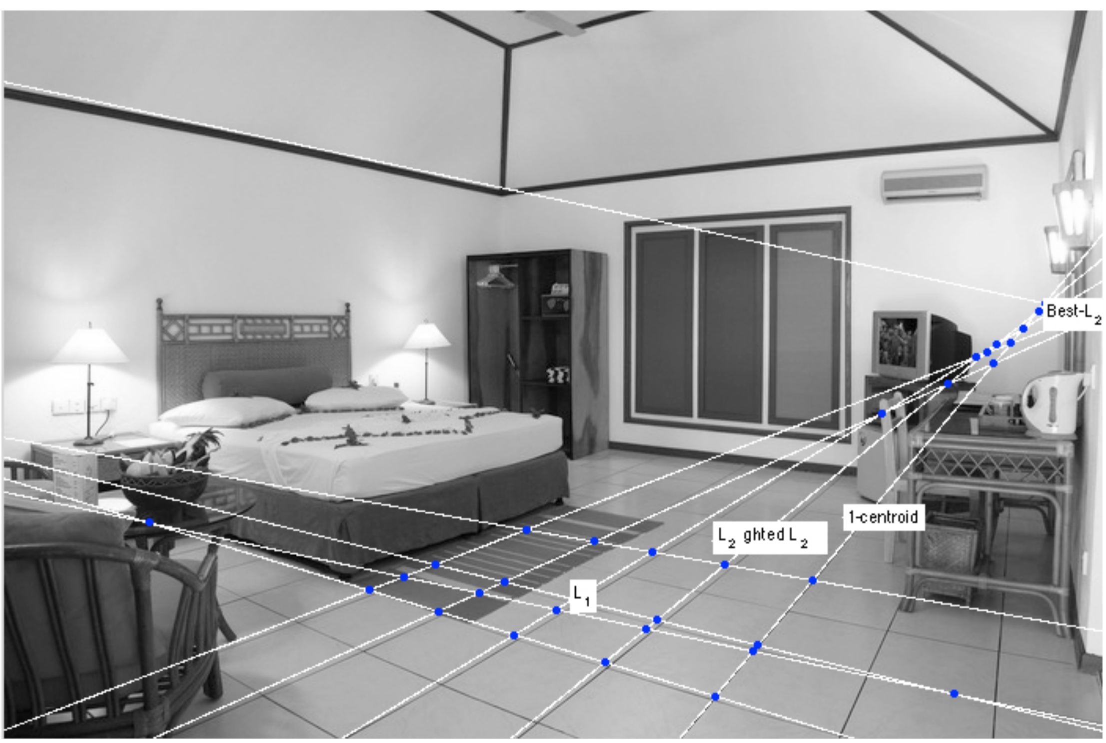
d. Canny Edge Detection.

c. Hough Transform with largest peaks denoted by red squares.



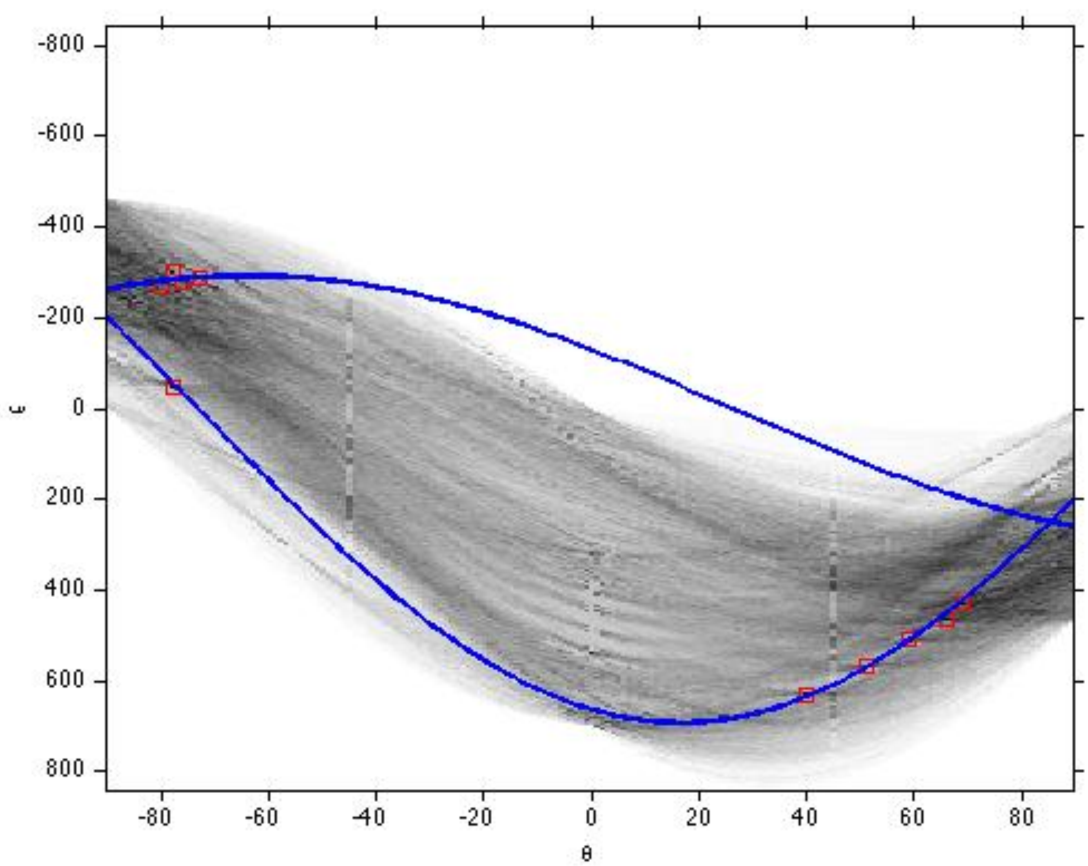
d. Lines corresponding to peaks (Hough Transform)

e. Least Square Interactions



Original grayscale image with vanishing points and 10 strongest lines (corresponding to floor tiles) superimposed. Blue solid circles, marks the line, and estimates of vanishing point locations using the methods described. In order of worst to best: K = 2 centroid, L_2, Weighted L_2, L_1, and Best-L_2 (Improved K-Mean : Algorithm 1).

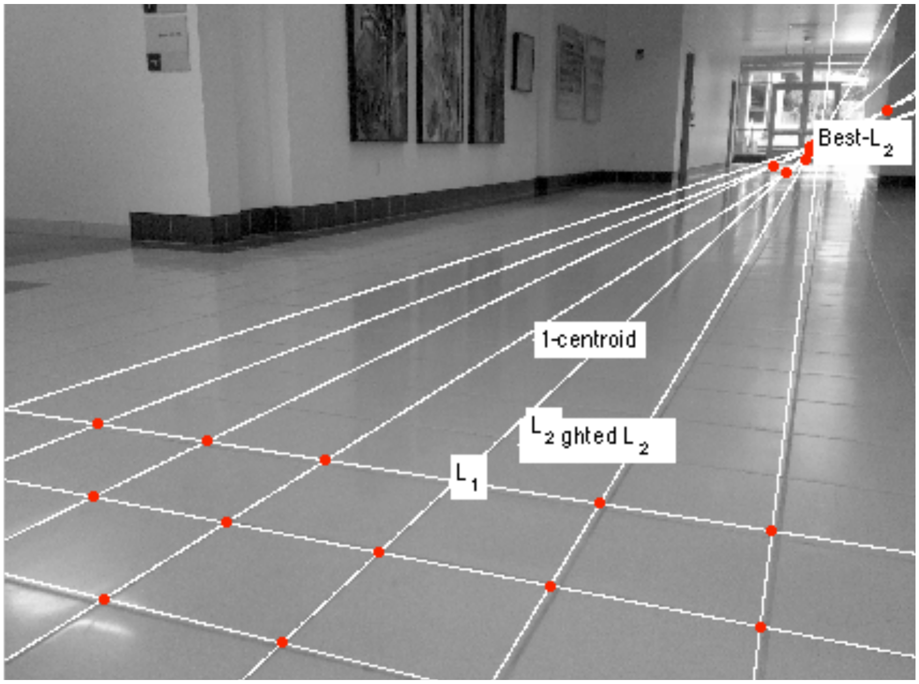
f. Hough-domain view of the edge map, with 10 peaks denoted by red squares.

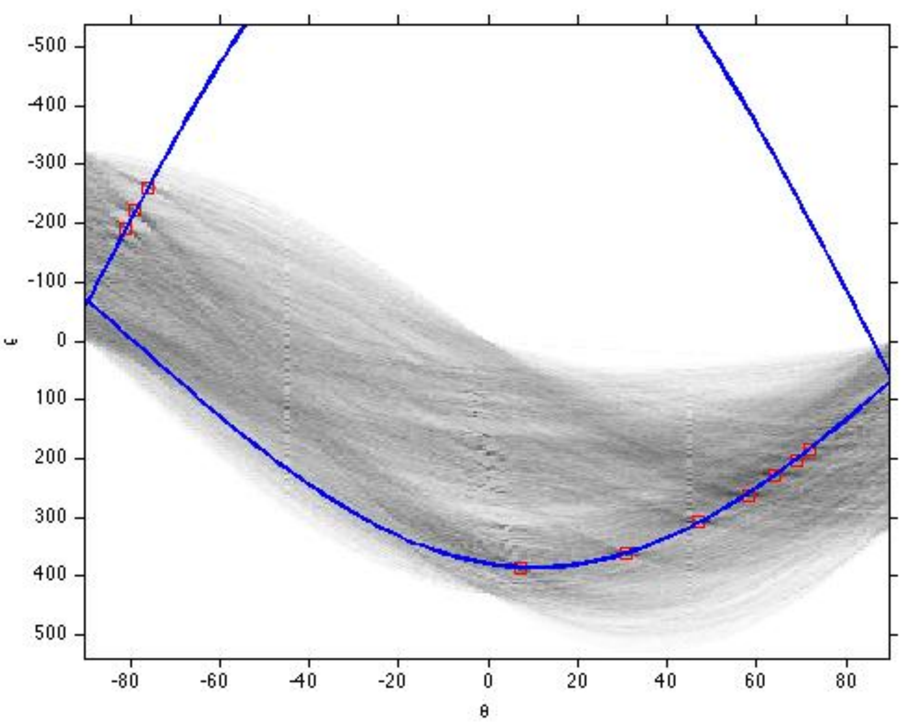


Here the assumption of K = 2 vanishing points is a poor one. As K-Means algorithm suffers from “overfitting”.

Elapsed CPU Time : 2.0700 seconds

g. Hough Transform for a different room.





Here K = 2, best fits the hough peaks with sinusoids explaining the same.

Elapsed CPU Time : 2.3800 second

VI. CONCLUSION

The problem of extracting vanishing points has remained an active field of research since the 1970's, owing primarily to problems of accuracy and robustness. The methods presented are able to extract the vanishing points in indoor environments, which can be used to reconstruct a partial 3-dimensional indoor map.

After tuning thresholding parameters, and relevant Hough transform resolutions, the vanishing points could be found in few seconds. Though the K-Means algorithm is not an optimal one, further work can be done to optimize the vanishing point procedure for robustness, speed and positional errors decrease.

Higher order Hough Transform can also be implemented for spherical, polar projects & RANSAC[9] algorithms. Today, mobile devices has reached a high level of real time computational capabilities, possibility of porting the application to mobile phone for 3-Dimensional vector maps generation would be an interesting interest area.

[9] Fayez Tarsha-Kurdi, Tania Landes, Pierre Grussenymer, "Hough-Transform and Extended RANSAC Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data", ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, Espoo : Finland (2007)

BIBLIOGRAPHY

1. C. Rother, “A New Approach for Vanishing Points Detection in Architectural Environments,” Image and Visual Computing ,vol. 20, pp. 647-656,September 2002.

2. B. Caprile and V. Torre, “Using Vanishing Points for Camera Calibration,” International Journal of Computer Vision vol. 4, pp. 127-140,February 1990.

3. Shih-Ping Lioua and Ramesh C. Jaina, “Road Following Using Vanishing Points,” Computer Vision, Graphics and Image Processing, vol. 39, pp.116-130, January 1987.

4. R. Lodola, A. Mecocci and U. Salvatore, “Outdoor Scenes Interpretation Suitable for Blind People navigation,” Proc. of Image Proc. and Its Application, Edimburgh, 4-6 July 1995, pp. 256-260.

5. Eyelyne Lutton, Henri Maitr and Jaime Lopez-Karhe, “Contribution to the Determination of Vanishing Points using Hough Transform,” IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 16, No. 4, pp. 430-442, April 1994.

6. Rafael C. Gongalez, Richard E. Woods, “Digital Image Processing”, Third Edition, Pearson, ISBN 978-81-317-1934-3.

7. John Canny. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6):679–698, Nov. 1986.

8. Richard O. Duda, Peter E. Hart, David G. Stork, “Pattern Classification, 2nd Edition”, Wiley, ISBN: 978-0-471-05669-0.

9. Fayez Tarsha-Kurdi, Tania Landes, Pierre Grussenymer, “Hough-Transform and Extended RANSAC Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data”, ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007, Espoo : Finland (2007).

10. C. Jung and R. Schramm, “Rectangle detection based on a windowed hough transform,” in *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on*, 17-20 2004, pp. 113–120.

11. CVX : Matlab Software for Disciplined Convex Programming. http://cvxr.com/cvx/