

Pedestrian Lane Detection for Assistive Navigation

A Project Report

submitted by

B Shravan (13EC209)

PrithviRaj Badri (13EC208)

P Hari Charan (13EC138)

G Mohan krishna (13EC119)

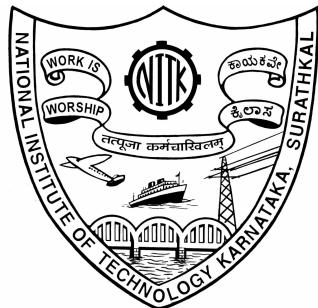
under the guidance of

Dr.A V Narasimhadhan

in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY



DEPARTMENT OF ELECTRONICS AND COMMUNICATION

ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE - 575025

April 16, 2017

ABSTRACT

Ever increasing technological advancements have substantially contributed to autonomous technology. Automatically finding paths is a crucial and challenging task in autonomous navigation systems. Vision based lane detection is essentially important in many fields however the lane detection in unstructured lanes is still very challenging at current stage, and the existing methods often suffer from high computational complexity. Lane Detection is a problem that involves Computer vision and Image understanding techniques which include methods for acquiring, processing, analyzing and understanding digital images.

This project models vision based Pedestrian lane detection for Assistive navigation in unstructured scenes. Lane detection aims to locate the lane region in a scene. It is a key component in assistive navigation for blind people, autonomous vehicles and mobile robots. The lanes may vary significantly in color, texture, and shape and are not indicated by any painted markings. An efficient lane detection system helps travelers enter the lane in the right direction and avoid the dangers of straying outside the lane region.

This project consists method from which a lane appearance model is constructed adaptively from a sample image region which is identified automatically from the image vanishing point based on color tensor and dominant local orientations of color edge pixels and also introduces two important features namely Area feature and Texture feature that makes the sample lane selection more robust to textured, indoor and outdoor lanes. Pedestrian lane detection algorithm can also be extended for occlusion handling.

TABLE OF CONTENTS

ABSTRACT	i
1 Introduction	1
1.1 Problem definition	1
1.2 Previous work	1
1.3 Motivation	2
1.4 Overview	3
2 Description	4
2.1 Vanishing Point Estimation	4
2.2 Lane Border Detection and Sample Lane Selection	4
2.3 Lane Segmentation	5
3 Implementation and Results	6
3.1 Vanishing point Estimation	6
3.2 Image Segmentation	9
3.2.1 K-means clustering	10
3.2.2 Graph based Segmentation	11
3.3 VP estimation using other methods	13
3.3.1 Hough based method	13
3.3.2 Gabor Based method	14
3.4 Sample Lane Selection	15

3.5	Lane Segmentation	19
3.5.1	Shape context	21
3.6	Results	25
3.7	Conclusion	26

LIST OF FIGURES

1.1	Block Diagram of Pedestrian Lane Detection System	3
3.1	Vanishing Point Estimation on Image1	8
3.2	Vanishing point	9
3.3	K-means image segmentation	11
3.4	N-cut graph segmentation	12
3.5	Efficient graph based algorithm	12
3.6	Hough Transform	13
3.7	Detected Line Segments	13
3.8	Gabor orientations	14
3.9	Imaginary Rays	15
3.10	Sample lane selection on Image1	19
3.11	Significance of texture feature	19
3.12	Sample lane selection on Image2	20
3.13	Shape Templates	22
3.14	Detected lane	23
3.15	Detected lane	23

LIST OF TABLES

3.1	Brick Lane Image	18
3.2	Smooth Lane	18
3.3	Indoor lane	18

CHAPTER 1

Introduction

Computer vision deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do [1]. Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and in general, deals with the extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. Pedestrian Lane Detection is one such problem that involves Computer vision and Image understanding techniques.

1.1 Problem definition

Lane detection aims to locate the lane region in a scene. It is a key component in assistive navigation for blind people, autonomous vehicles and mobile robots. An efficient lane detection system helps travellers enter the lane in the right direction and avoid the dangers of straying outside the lane region. Algorithms for vehicle detection [4,7,8,10,14,16,18], zebra-crossing lines based detection [19]. Existing methods involves road detection algorithms using various color models [5,6,12] texture based algorithms [9] using Gabor filters and vanishing point estimation. Also B-Snake [13] based lane detection and tracking algorithm without any cam- era parameters was proposed which works on marked lanes. Although there are many algorithms that are developed for road detection, only a few methods proposed for pedestrian lane detection, which are mostly concerned with pedestrian lanes having white markers [11,16,18,19,20]. So, detection of pedestrian lanes in such unstructured scenes plays an important role and efficient detection algorithms to be found under all conditions.

1.2 Previous work

Current vision-based approaches for detecting pedestrian lanes in unstructured scenes can be divided into two categories:

(i) lane segmentation, and (ii) lane-border detection.

In the lane segmentation approach, algorithms rely on color and texture of lane surfaces to differentiate the lane pixels from the background [5,6,9,12]. Different color spaces and classifiers have been used. Previous methods used are Gaussian models in the RGB color space to represent the on-road and off-road classes [5,6,12]. Also using the RGB space, it captures the variability of the road surface with multiple color histograms, and the background with a single color histogram, and in the hue- saturation-intensity (HSI) color space achromatic pixels are classified using intensity only, whereas other pixels are classified by thresholding their chromatic distance to the training colors. Because the color models are trained off-line, these methods do not cope well when the data varies from that of training data.

To address this problem, several methods exists which can directly obtain lane pixels from the input image and these methods determine the sample lane regions in different ways. For example, considering small random areas at the bottom and in the middle of the input image but it tends to include non-lane regions. The performance of these methods depends on the quality of the sample regions, which in turn relies on prior knowledge about the walking lane.

In the lane-border detection approach, the lane boundaries are determined using the vanishing point [3,9]. In other methods, the lane borders are detected among the edges pointing to the vanishing point. These methods are effective only when the lane region is homogeneous and differs significantly from non-lane regions in terms of color and texture. Because this approach relies only on edges for lane-border detection, it is sensitive to background edges. Most of the work is done for straight lanes detection and a little work has been made for the curved lanes and here we consider detecting of such lanes that occur in unstructured scenes under all possible conditions.

1.3 Motivation

Algorithms for finding the pedestrian lane can be extended to locate open roads for self-driven cars or robots. It enables a vision-impaired person to find the walkable path and maintain his or her balance while walking a challenging task that at present is performed mostly using a white cane. It also allows a smart wheelchair to traverse a pedestrian lane with little guidance from the disabled user. Pedestrian lane detection is also useful for

autonomous vehicles in sensing off-limit regions or pedestrians in a scene.

1.4 Overview

Detecting unstructured pedestrian lanes can be divided into three main stages:

- (i) Vanishing Point Estimation
- (ii) Lane Border Detection and Sample Region Selection
- (iii) Lane Segmentation.

The Fig.?? is the block diagram that represents input-output relation using the above mentioned stages.

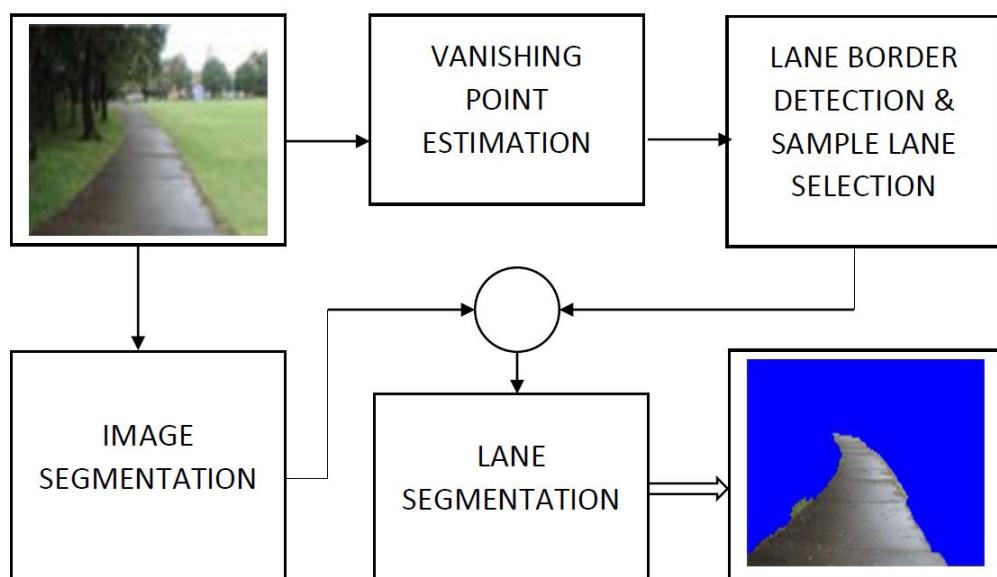


Figure 1.1: Block Diagram of Pedestrian Lane Detection System

CHAPTER 2

Description

By using vanishing point and constructing the lane model directly from the image makes more adaptive to different illumination conditions and lane surfaces. Detection of pedestrian lanes in unstructured scenes , which is adaptive to different illumination conditions, can be divided into three main stages as mentioned earlier, their brief description is discussed in this chapter.

2.1 Vanishing Point Estimation

The vanishing point in an image is often located using either line segments or local orientations. For unstructured scenes with non-straight edges, using local orientations is more suitable than using line segments for vanishing point estimation. However, most existing methods based on local orientations have high computational complexity and are sensitive to background edges. Furthermore, they rely on the intensity channel only, even though color channels provide photometric information that can lead to more robust detection of edges and local orientations. Here to improve the accuracy and efficiency of vanishing point detection, it is good to employ color tensor [21] to capture image structure and focusing on edge pixels only. Thus using the local orientations and edge strength for each pixel, obtained from the color tensors, vanishing point is observed which is useful for the subsequent stages that follows.

2.2 Lane Border Detection and Sample Lane Selection

Because the appearance i.e, color, edge, shape, texture of pedestrian lanes in unstructured scenes varies significantly and is strongly affected by illumination conditions, it is difficult to obtain a robust appearance model with off-line training. Hence, it is more reasonable to construct an appearance model adaptively and directly from the input image. Existing methods usually select the sample region as a small region at the bottom or center of the input image. However, the sample region selected in such a manner tends to include

non-lane regions. Hence, lane border is detected and the sample region is automatically defined using the vanishing point found in previous stage.

2.3 Lane Segmentation

In this, the input image is segmented initially into color homogeneous sub-regions using the graph-based segmentation algorithm [23,24]. This algorithm initializes sub-regions as single pixels. Adjacent sub-regions are then merged iteratively, according to the color difference between the sub-regions. Each sub-region is represented by a color feature and a shape feature [25], and the sample lane region obtained from the previous stage can be used to detect the pedestrian lane. Here we consider two color spaces, the RGB and the illumination invariant space (IIS) [26]. Compared to the RGB, the IIS is less sensitive to illumination conditions and shading.

CHAPTER 3

Implementation and Results

In this chapter, description of each and every step is elaborately given and their implementations are also discussed. The results section shows the obtained outputs in each step on various lanes and significance of area and texture feature during sample lane selection step is well explained.

3.1 Vanishing point Estimation

As discussed in chapter 2 about vanishing point estimation in brief, its implementation and results obtained are discussed in this chapter.

Simply summing differential structure of various color channels may result in cancellation even when evident structure exists in the image. Rather than adding direction information in $(0, 2\pi)$ it is appropriate to add orientation information in $(0, \pi)$. Tensors describe the local orientation rather than direction. Tensors are well suited to combine first order derivatives for color images. To obtain accurate vanishing point in an efficient way we use color tensor to capture image structure and also we focus only on edge pixels.

Color tensor is a tool [21] for analyzing the local differential structure of a color image. Here the color tensor is applied on multiple channel instead of only intensity channel. Also color channels provide photometric information that lead to more robust detection of edges and local orientations. Color tensor or Structure tensor adequately handles the vector nature of the image. Using both the features from color tensor and photometric invariant derivatives we obtain the photometric invariant features such as edges, corners, optical flow and curvature. Consider an image with three color channels: $F = F_k; k = 1, 2, 3$. Let D_{kx} and D_{ky} denote the derivatives of F_k along the horizontal and vertical direction, respectively.

$$G = \begin{bmatrix} G_{xx} & G_{xy} \\ G_{xy} & G_{yy} \end{bmatrix}$$

where

$$G_{xx} = \mathbf{w} * \left[\sum_{n=1}^3 D_{k,x} \cdot * D_{k,x} \right] \quad (3.1)$$

$$G_{yy} = \mathbf{w} * \left[\sum_{n=1}^3 D_{k,y} \cdot * D_{k,y} \right] \quad (3.2)$$

$$G_{xy} = \mathbf{w} * \left[\sum_{n=1}^3 D_{k,x} \cdot * D_{k,y} \right] \quad (3.3)$$

where \mathbf{w} is smoothing filter and $\cdot *$ is elementwise product.

Based on Eigen value analysis of the color tensor, we estimate dominant local orientation θ and the edge strength λ for all image pixels as

$$\theta = \frac{1}{2} \arctan\left(\frac{2G_{xy}}{G_{xx} - G_{yy}}\right) + \frac{\pi}{2} \quad (3.4)$$

$$\lambda = \frac{1}{2}(G_{xx} + G_{yy} + \sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}) \quad (3.5)$$

The dominant orientations and edge strengths of all image pixels are computed in **MATLAB**. The result obtained is shown in Fig:3.1b, The dominant orientation is computed w.r.t horizontal direction.

In estimating the vanishing point, instead of focussing on every pixel we consider only edge pixels. So to compute edge pixels, canny edge detection algorithm [1,2] is used. The algorithm is briefly explained as follows.

Canny Edge Detection algorithm

- Smooth the input image with a Gaussian filter.
- Compute the gradient magnitude and angle images.
- Apply non-maxima suppression to the gradient magnitude image.
- Use double thresholding and connectivity analysis to detect and link edges.

The reasons for choosing canny edge detector are

- Low error rate.
- Edge points are localized.
- Single edge point response.

The edge detected pixels after applying canny edge-detection algorithm is shown in Fig:3.1a.

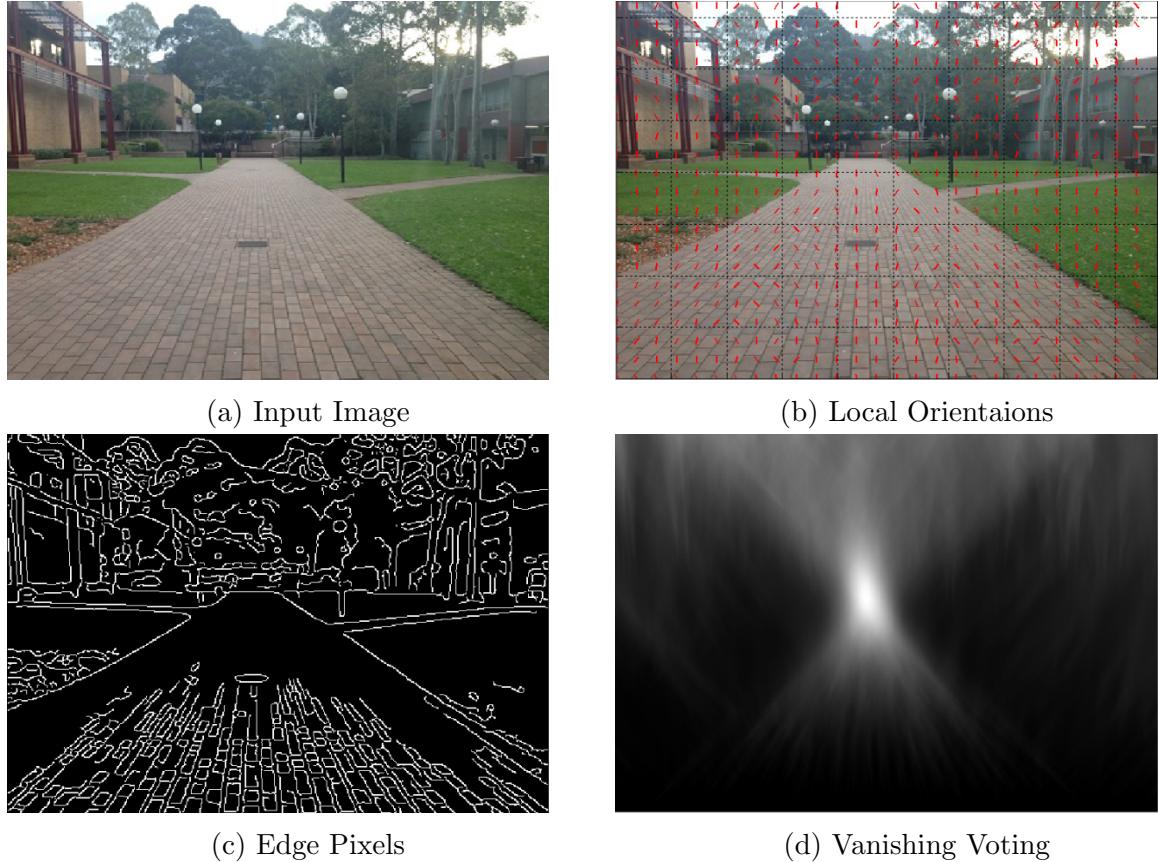


Figure 3.1: Vanishing Point Estimation on Image1

Vanishing point voting

To compute vanishing point [3,22], all the edge pixels $p = (x_p, y_p)$ and candidate pixels $v = (x_v, y_v)$ are considered such that $y_v > y_p$. (considering the edge pixels below the candidate pixel). A pixel contributes significantly to vanishing point when its dominant orientation θ_p is more towards the probable vanishing point candidate. So to give more weight to such pixels, their contribution is considered only when Δ_{vp} , the difference between dominant local orientations at pixel v and the angle of the vector connecting

v to edge pixel p is less than a positive threshold, i.e. Voting is considered only when $\Delta_{vp} < |\theta_p - \angle l_{vp}|$ threshold. A threshold of 5° was used .

But here the voting is biased towards higher pixels in the image, to eliminate this bias, distance measure is also included. Here the pixels which are close are given more weight than the pixels far away and this is done by considering μ_p which is ratio of length of vector l_{vp} and the diagonal length of the image L (i.e.) $\mu_p = l_{vp}/L$. So the vanishing point score is calculated using the equation

$$s(v, p) = \exp(-\Delta_{vp}\mu_p), \quad if \quad \Delta_{vp} < 5^\circ \\ = 0, \quad otherwise$$

So $s(v, p)$ is high when pixel p has similar orientation to l_{vp} and is close to pixel v. The vanishing point score of each pixel v is sum of $s(v, p)$ due to all edge pixels P.

$$S(v) = \sum_{p \in P} s(v, p)$$

The pixel with maximum Score $S(v)$ is selected as vanishing point. Voting scheme can be verified in Fig:3.1d. The vanishing point on two images are shown in red dot in Fig:3.2.



Figure 3.2: Vanishing point

3.2 Image Segmentation

Lane is initially segmented into color homogeneous sub-regions using image segmentation algorithm. K-means and Graph based segmentation algorithms are used to segment the image. Comparing both algorithms, is observed that graph-based segmentation is best suited in terms of time complexity and accuracy.

3.2.1 K-means clustering

Transform the image from RGB space to color-space “Lab”

The “Lab” space [1,2] consists of a luminosity layer ‘L’, chromaticity-layer ‘a’ indicating where color falls along the red-green axis, and chromaticity-layer ‘b’ indicating where the color falls along the blue-yellow axis. All of the color information is in the ‘a’ and ‘b’ layers. You can measure the difference between two colors using the Euclidean distance metric.

Applying the k-means clustering

The whole information about the image is defined in terms of the colors on ‘a and ‘b axis. Using Euclidean distance metric for k-means clustering algorithm, we can find clusters which in-turn are segments for the image. The k-means is applied on a lane image and it is shown in Fig:3.3

Disadvantages:

1. Choosing number of clusters.
2. Here, only the color information is considered for segmenting the image. The spatial information is ignored.

Requirements:

1. Considering both the color information as well as spatial relationships.
2. Adaptive number of segments.
3. Aims to extract the global impression of an image.

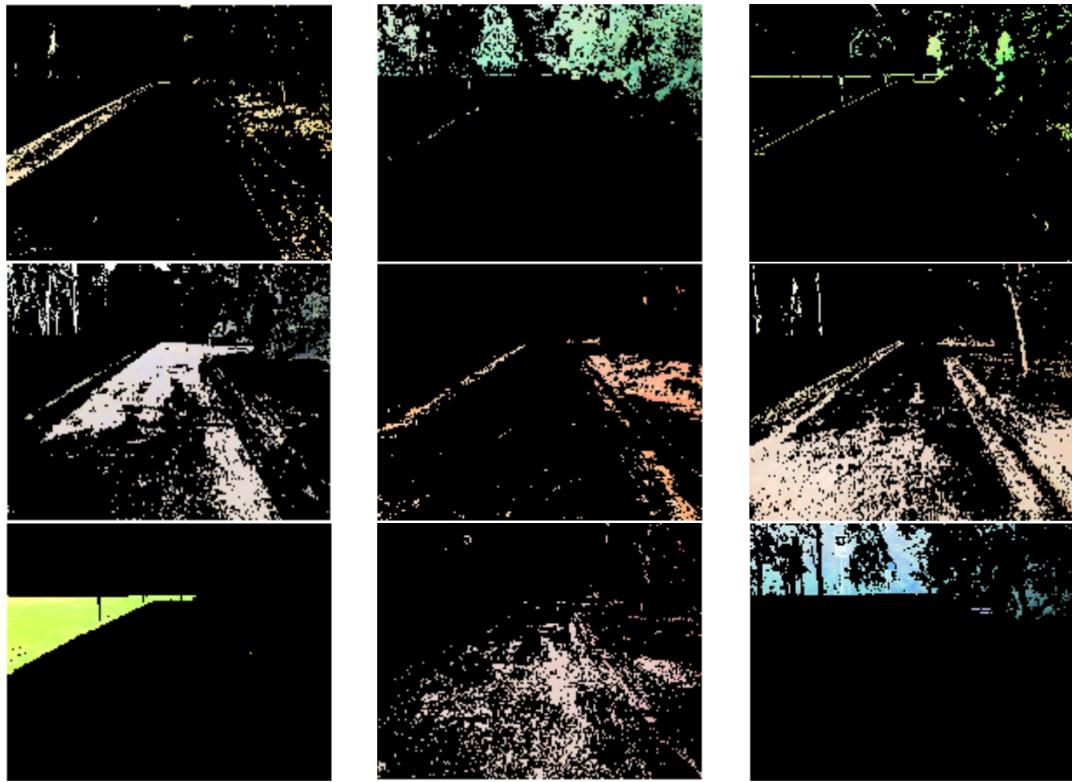


Figure 3.3: K-means image segmentation

3.2.2 Graph based Segmentation

N-cut

It aims at extracting the global impression of the image. The approach is most related to the graph theory. The set of points in the arbitrary feature space are represented as a weighted undirected graph $G = (V, E)$, where the nodes of the graph are the points in the feature space, and an edge is formed between every pair of nodes. The weight on each edge is a degree of similarity between nodes i and j .

Next in grouping of vertices, partitioning the vertices into disjoint sets V_1, V_2, \dots, V_n , where by some metric the similarity among the vertices in a set V_i is high and across different sets V_i, V_j is low.

Things needed:

1. Need a criterion for good partition.
2. Need an algorithm to compute it efficiently. (like gradient descent algorithm to compute global minimum of a cost function efficiently).

The algorithm [23] treats an image pixel as a node of the graph, and considers image segmentation as a graph partitioning problem.

The graph based segmentation using N-cut is applied on an image and the segmented output is shown in Fig:3.4.

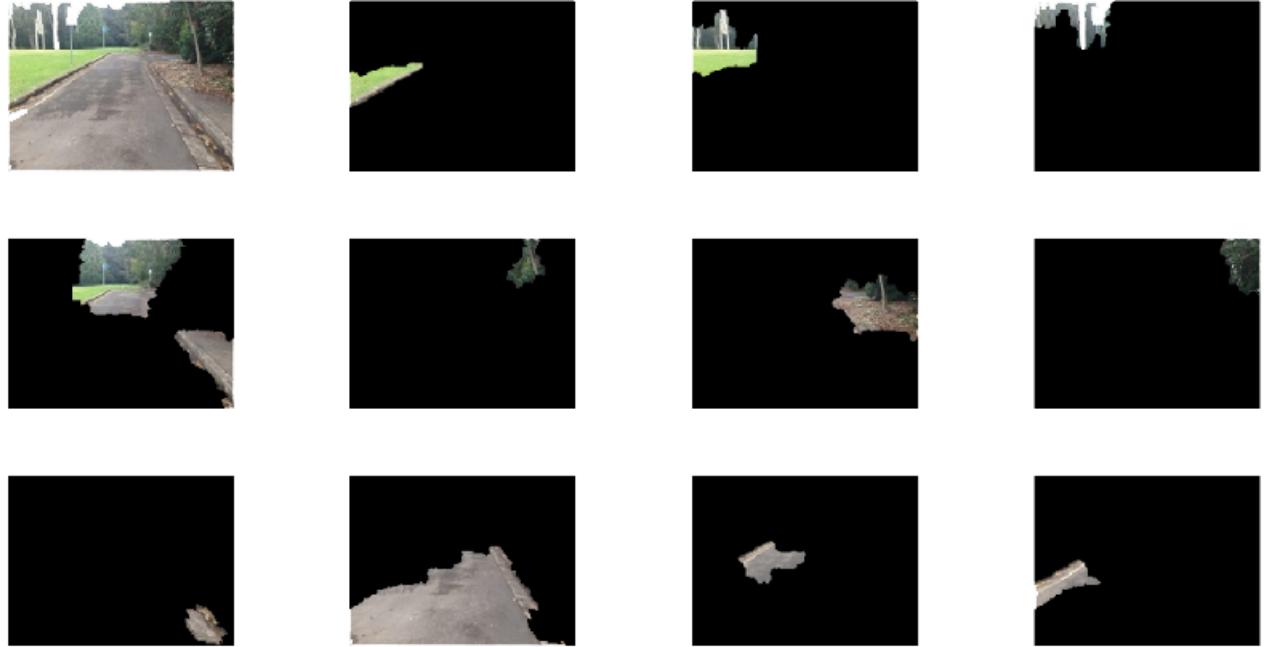


Figure 3.4: N-cut graph segmentation

Efficient Algorithm

The problem with N-cut algorithm is that it works better on smaller resolution images. The efficient graph-based segmentation in [24] can be used even to higher resolution images where it captures perceptually important components. The segmentation output is shown in Fig.3.5

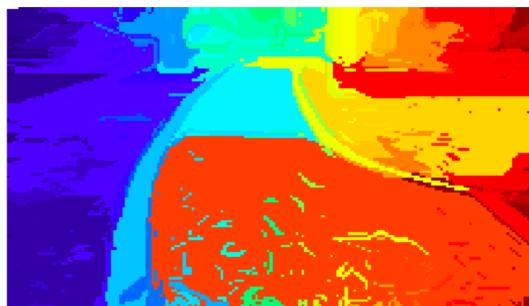


Figure 3.5: Efficient graph based algorithm

3.3 VP estimation using other methods

3.3.1 Hough based method

In this method, we first apply the Hough transform [1,2] on the edge map to find line segments. Next, we compute the vanishing point by voting the intersections of line pairs in another Hough transform. The Hough transform is applied and shown in Fig: 3.6 on input lane and obtained output regarding hough space and detected line segments is shown in Fig:3.7. Finally to obtain vanishing point from the detected line segments, apply another hough transform to get the point with more number of line intersections.

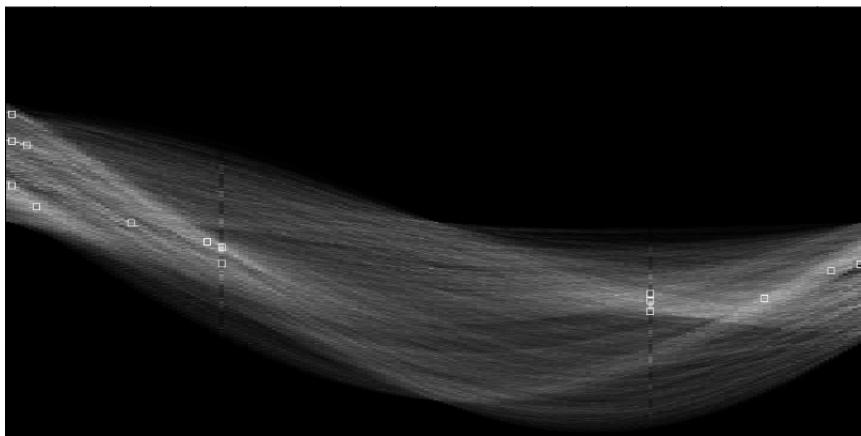


Figure 3.6: Hough Transform



Figure 3.7: Detected Line Segments

3.3.2 Gabor Based method

In this method we use four Gabor filters [22] to estimate the local dominant orientation at each pixel p in the intensity image. Each pixel along ray r will accumulate a voting score according to its distance to pixel p . Finally, the image pixel with the highest voting score is considered as the image vanishing point. The gabor dominant orientations are computed and shown in Fig:3.8.

A 2-D Gabor kernel g for a preferred orientation ϕ_n is given by

$$g_{\omega_0, \phi_n} = \frac{\omega_0}{\sqrt{2\pi k}} e^{\frac{-\omega_0^2(4a^2+b^2)}{8k^2}} \cdot [e^{i\omega_0 a} - e^{\frac{-k^2}{2}}] \quad (3.6)$$

where $a = x \cos \phi_n + y \sin \phi_n$, $b = -x \sin \phi_n + y \cos \phi_n$, $k = \frac{\pi}{2}$, $\lambda = 4\sqrt{2}$, $\omega_0 = \frac{2\pi}{\lambda}$

The image is convolved with kernel defined above as shown in equation 3.12. And the energy referred as gabor energy at each pixel is computed using equation 3.13.

$$\hat{I}_{\psi_n}(p) = I(p) \otimes g_{\psi_n}(p) \quad (3.7)$$

$$E_{\psi_n}(p) = \sqrt{Re(I_{\psi_n}(p))^2 + Im(\hat{I}_{\psi_n}(p))^2} \quad (3.8)$$

More specifically, to estimate the dominant orientation ϕ_n at each pixel location $p(x;y)$ in the image, an input grayscale image is convolved with four oriented Gabor filters at $0^\circ, 45^\circ, 90^\circ$ and 135° , and Gabor energy responses are computed for each pixel location.

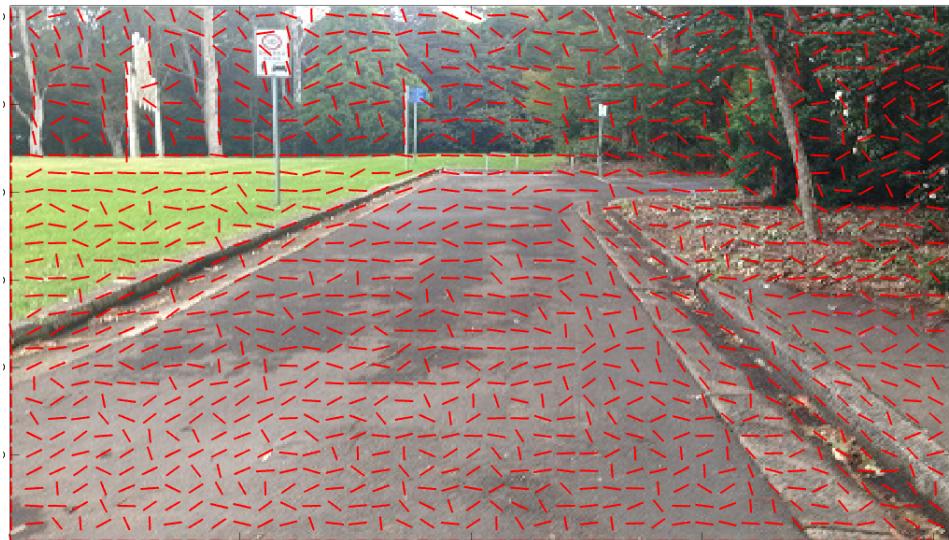


Figure 3.8: Gabor orientations

3.4 Sample Lane Selection

During the lane segmentation step, we require color feature and shape feature of the connected sub-regions that are formed from image segmentation process [3]. To acquire the color feature we need sample lane region so that color feature can be calculated for all the set of connected regions, to detect the lane region. The appearance (includes color, shape, texture) of the lanes in unstructured scenes vary significantly and difficult to obtain robust model with offline training (such as with color model to separate lane region from non-lane regions, predefined shape models, etc.).

It is better to obtain the sample lane directly from the image itself. Using vanishing point we can obtain the appearance model and verify using color and orientation features of the lane borders and lane regions. Since most part of the lane can be approximated with straight lines, we use set of rays to denote the lane borders. The rays are taken from ϕ_{min} to ϕ_{max} in Fig:3.9.

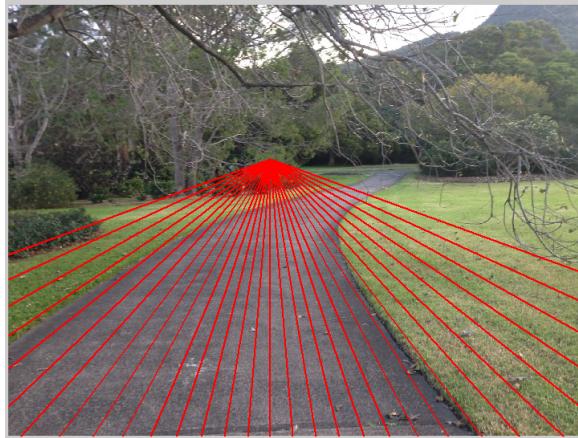


Figure 3.9: Imaginary Rays

Algorithm:

For each ray, two features are calculated Orientation difference feature, Color difference feature. For a ray pair, three features are defined Color uniformity of the region formed by the ray pair, Area of the region (given by number of pixels) and the texture of the region.

Therefore totally for a ray pair(ray_i, ray_j),seven features are computed.

- Orientation difference for ray_i and neighbouring pixels
- Color difference for region on both sides of ray_i

- Orientation difference for ray_j and neighbouring pixels
- Color difference for region on both sides of ray_j
- Color uniformity of the region
- Area feature with gaussian distribution characterized from histogram of the area of regions.
- Texture of the region

Orientation difference feature:

Consider the neighbour pixels on either side of the ray within range (threshold is set).
Find the orientation difference between the ray and the neighbouring pixels.

$$d_o = \frac{1}{Nr} * \sum_{p \in Nr} (\theta(r) - \theta(p)) \quad (3.9)$$

Where Nr = no of neighbouring pixels

$\theta(r)$ = orientation of the ray

$\theta(p)$ = orientation of the neighbouring pixel

Color difference:

Two regions are formed on either side named left and right region. For both the regions mean color is computed and difference between the mean colors is calculated with normalization factor included.

Color uniformity:

To find the color uniformity, the histogram for all the three color channels of a region formed by the ray pair is computed.

Area feature:

This is the number of pixels comprising the region formed by the ray pair. First the histogram of the areas formed by the ray pair is computed. The mean area of the region is computed and with some predefined variance a gaussian function is built.

Texture feature:

An important approach to region description is to quantify its texture content. Although no formal definition of texture exists, intuitively this descriptor provides measures of properties such as smoothness, coarseness, and regularity [1]. Statistical approaches yield characterizations of textures as smooth, coarse, grainy, and so on. An approach used frequently for texture analysis is based on statistical properties of the intensity histogram. One class of such measures is based on statistical moments of intensity values. The expression for the nth moment about the mean is given by

$$\mu = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \quad (3.10)$$

where z is a random variable indicating intensity, $p(z)$ is the histogram of the intensity levels in a region, L is the number of possible intensity levels. The second moment (the variance) is of particular importance in texture description. It is a measure of intensity contrast that can be used to establish descriptors of relative smoothness. Smoothness measures the relative smoothness of the intensity in a region.

$$Smoothness : R = 1 - \frac{1}{1 + variance} \quad (3.11)$$

It is 0 for a region of constant intensity and approaches 1 for region with large excursions in the values of its intensity levels. In practice, the variance, used in this measure is normalized to the range $[0, 1]$ by dividing it by $(L - 1)^2$.

From the Tables: 3.1, 3.2, 3.3 it is observed that standard deviation and measure of smoothness values are well differentiated for the lane region when compared to non-lane region and other regions. Hence standard deviation and measure of smoothness can be used as features and the lane score for these features is given by

$$Texture\ feature = e^{-0.005 * Standard\ deviation - Smoothness} \quad (3.12)$$

Thus using all these features the region with maximum feature product is selected as sample lane.

	Lane and non-lane regions	Lane region	Non-lane region
Average Gray Level	161.77	171.07	139.53
Standard Deviation	18.27	10.16	36.05
Measure of Smoothness	0.0051	0.0016	0.0196
Third Moment	-0.0614	0.0011	-0.5148
Measure of uniformity	0.0220	0.0306	0.0095
Entropy	6.0084	5.3242	7.0601

Table 3.1: Brick Lane Image

	Lane and non-lane regions	Lane region	Non-lane region
Average Gray Level	113.75	118.11	124.33
Standard Deviation	14.35	8.0328	11.9
Measure of Smoothness	0.0032	0.001	0.0022
Third Moment	0.024	0.0001	0.0133
Measure of uniformity	0.0222	0.0364	0.0262
Entropy	5.8291	5.0364	5.5722

Table 3.2: Smooth Lane

Area feature and Texture feature are dependent and are important in getting correct sample lane. Due to introduction of these features and their significance discussed, the sample lane selection becomes more robust to textured, indoor and outdoor lanes. The results can be verified as shown in Fig:3.10. The Fig:3.10b is region with maximum uniformity feature and Fig:3.10d is region with maximum feature product with the inclusion of area and texture feature. The significance of texture feature is better shown in Fig:3.11. The Fig:3.12 shows the sample lane selection on other lane. The significance of both area feature and texture feature are well explained in Results section.

	Lane and non-lane regions	Lane region	Non-lane region
Average Gray Level	160.09	194.20	154.93
Standard Deviation	43.5526	12.2682	31.2509
Measure of Smoothness	0.0283	0.0023	0.0148
Third Moment	-1.0321	-0.0025	-0.1988
Measure of uniformity	0.0080	0.0296	0.0110
Entropy	7.2852	5.5048	6.8655

Table 3.3: Indoor lane

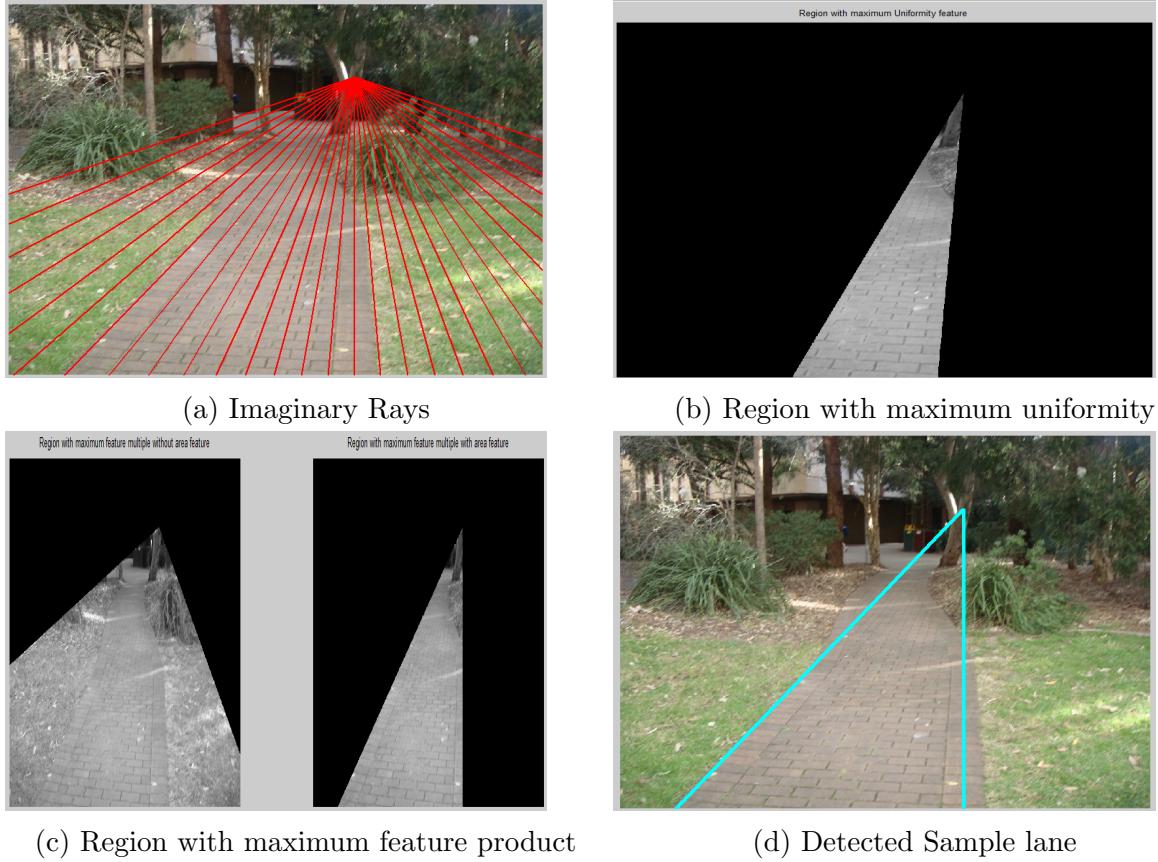


Figure 3.10: Sample lane selection on Image1

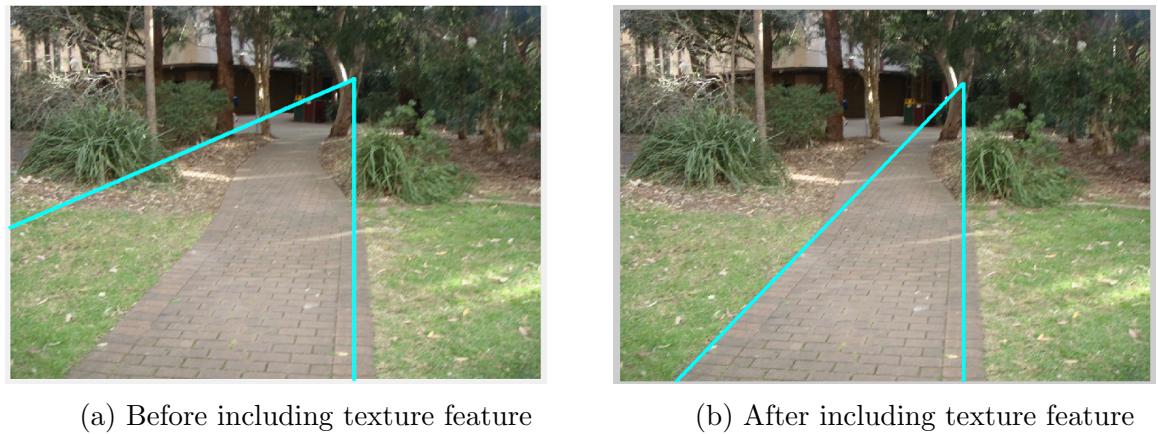


Figure 3.11: Significance of texture feature

3.5 Lane Segmentation

In this stage, input image is segmented initially into color homogenous sub-regions. As discussed in the previous section of image segmentation using K-means clustering and Graph segmentation using N-cuts, the segmented regions are obtained. But due to their own disadvantages , as discusssed previously, there is need of using efficient segmentation algorithm which captures perceptually important components and is given in [24]. The

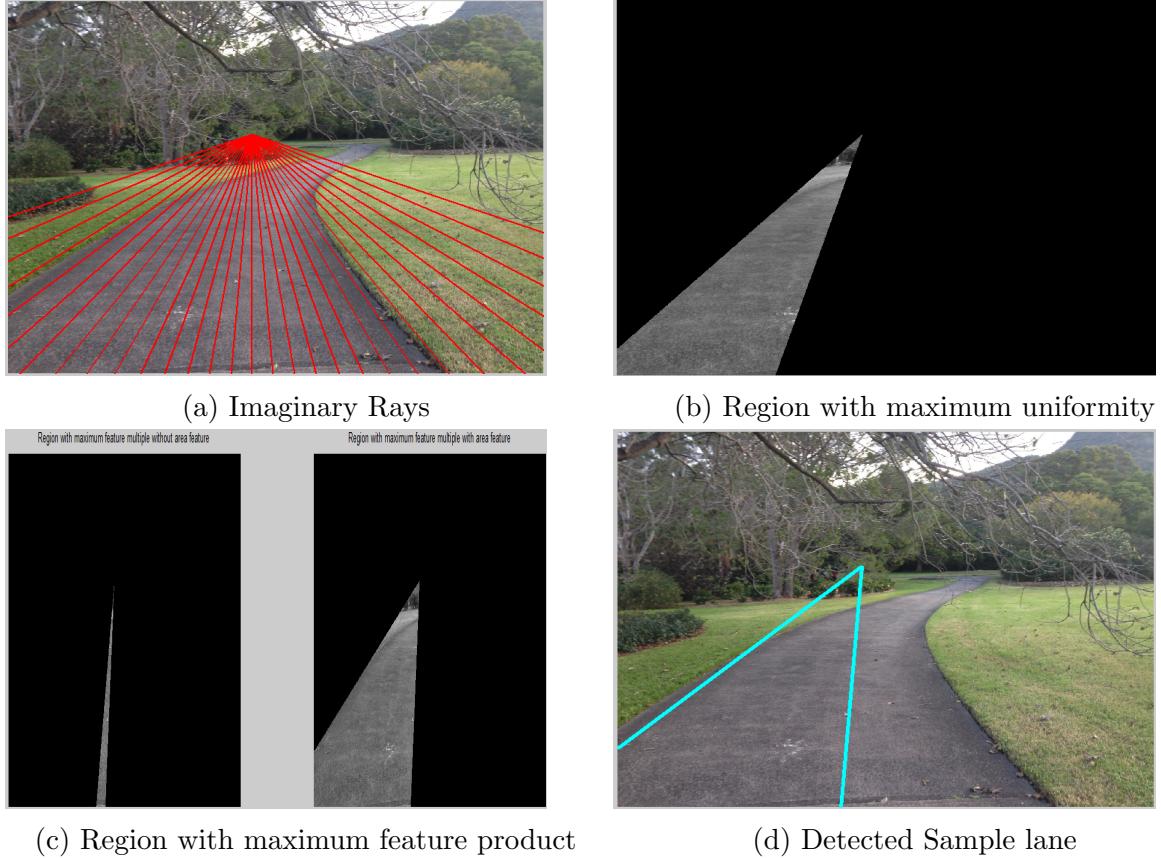


Figure 3.12: Sample lane selection on Image2

image segmentation result is shown in figure 3.16. After segmentation, pedestrian lane is detected. Let $R = \{R_1, R_2, \dots\}$ be set of color homogeneous sub-regions. Pedestrian lane is treated as set Z of connected subregions. Two sub-regions R_i and R_j are said to be connected if there exist two pixels $p_i \in R_i$ and $p_j \in R_j$ that are connected (e.g. 4-connected pixels). A connected region is represented by a color feature and shape feature.

Color feature:

The color feature is the mean of all color pixels in Z. The lane score for a given color feature \mathbf{c} is $p(c/L)$ where L is sample lane detected where $p(c/L)$ is class-conditional probability density function for the lane class. From the sample lane, we obtain the color histogram of the pixels and using this the lane score for color feature is estimated. We use two color spaces: Red-Green-Blue(RGB) [1] and Illumination Invariant Space(IIS) [26]. Compared to the RGB, the IIS is less sensitive to illumination conditions and shading. RGB to the IIS conversion is as follows:

$$C_1 = \arctan\{R/\max(G, B)\} \quad (3.13)$$

$$C_2 = \arctan\{G/\max(R, B)\} \quad (3.14)$$

$$C_1 = \arctan\{B/\max(R, G)\} \quad (3.15)$$

Shape feature:

The shape feature is extracted using the shape context [25]. The shape context are well known for their robustness and their invariance to scale and rotation. Matching of shape using shape context has some cost involved. A score corresponding to cost is calculated using shape templates in Fig:3.13.

3.5.1 Shape context

For each point p_i on the first shape, we want to find the best matching point q_j on the second shape. This is a correspondence problem which can be cleared by using shape context. Shape context is a novel descriptor which plays a vital role in shape matching. Consider the set of vectors originating from a point to all the other sample points on a shape. These vectors express the configuration of the entire shape relative to the reference point. Consider a shape with sample points on its contour.

The shape context of the sampling point p is the histogram h_p of the angles and distances from the remaining sampling points to p. Consider a point p_i on the first shape and a point q_j on the second shape. Let $C_{ij} = C(p_i, q_j)$ denote the cost of matching these 2 points.

Therefore the dissimilarity between the shape contexts of the two points p and q is represented as

$$C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \quad (3.16)$$

Where $h_i(k)$ and $h_j(k)$ denote the $K - bin$ normalised histogram at p_i and q_j respectively. The cost C_{ij} may include an additional term based on local appearance similarity at points p_i and q_j . This is particularly useful when we are comparing shapes deriving from gray-level images instead of line drawing.

On a single shape, the shape contexts of the points p and q are different, i.e. $C(p, q)$ is high. And on two similar shapes, the shape contexts of two corresponding points p and q are similar, so $C(p, q)$ is low.

Bipartite graph matching

Given the set of costs C_{ij} between all the pairs of points p_i on first shape and q_j on the second shape. The matching should be one-one. Therefore we make a permutations of the matching points the find the permutation which has total cost minimum.

$$H(p_i) = \min \sum (C(p_i, q_j)) \quad (3.17)$$

There exists many efficient algorithms like hungarian method which is done in $O(N^3)$. Let $T = \{T_1, T_2, \dots\}$ be a set of shape templates for pedestrian lanes. Examples of the shape templates obtained are shown in Fig:3.13.

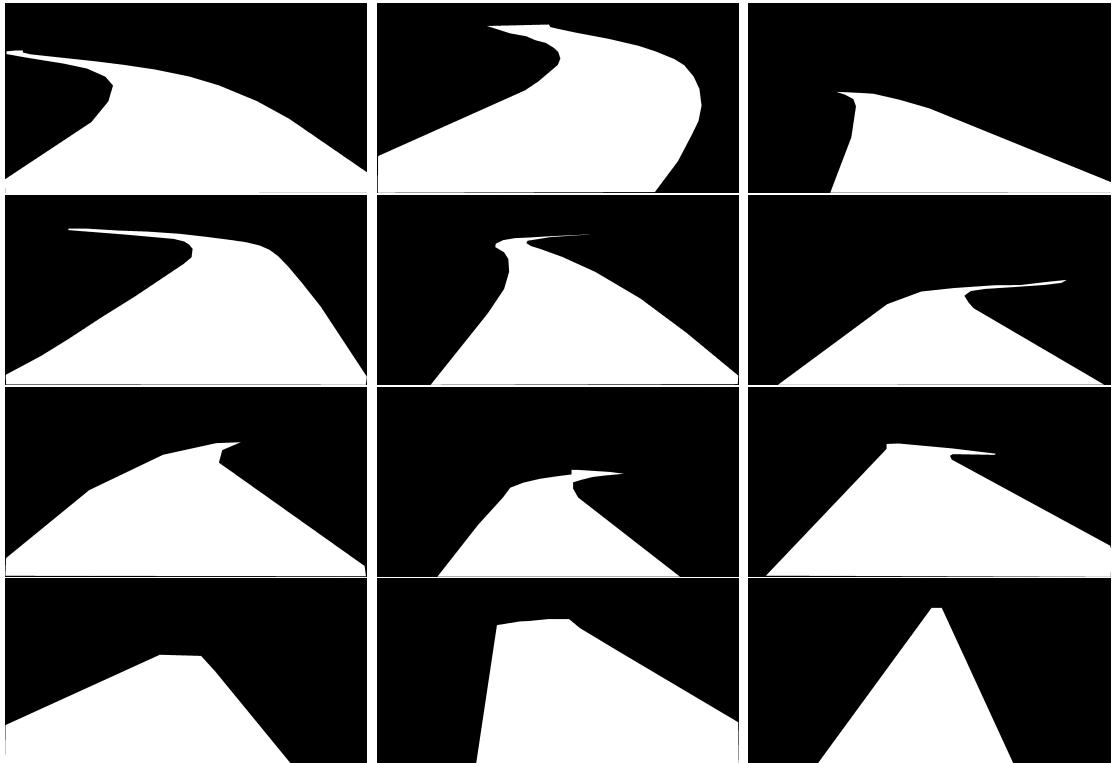


Figure 3.13: Shape Templates

To obtain shape feature \mathbf{s} , the outer contour of region Z is sampled in a similar way as the templates. The matching cost $D(\mathbf{s}, T)$ between \mathbf{s} and a template T is modeled as follows:

$$D(\mathbf{s}, T) = \frac{1}{|\mathbf{s}|} \sum_{p \in \mathbf{s}} \min_{q \in T} C(p, q) \quad (3.18)$$

Where $|\mathbf{s}|$ denotes the number of sampling points on \mathbf{s} . The smaller is the matching cost $D(\mathbf{s}, T)$, the higher is the similarity between \mathbf{s} and T . Now we have set of shape templates, need to find that shape template which has minimum cost and assign the cost

value as shape feature to the connected region. The lane score for the shape feature of connected region is given by

$$g_2(s) = \exp(-\lambda * \min(D(\mathbf{s}, T))) \quad (3.19)$$

The lane score for the connected region is given by

$$g(Z) = g_1(c) * g_2(s) \quad (3.20)$$

The optimal region Z^* of R is found by maximizing the lane score. That is finding the subset of R that has maximum $g(Z)$. Conventionally one can find all the subsets of R and find that Z , connected region which has maximum lane score. But that will be computationally expensive of order $O(2^N)$. Hence there is need of effective algorithm to find the connected region Z . For this, use of greedy-search algorithm in [3] can be of great use. Below one such algorithm goes by iteratively adding or removing subregions under some constraints is discussed.

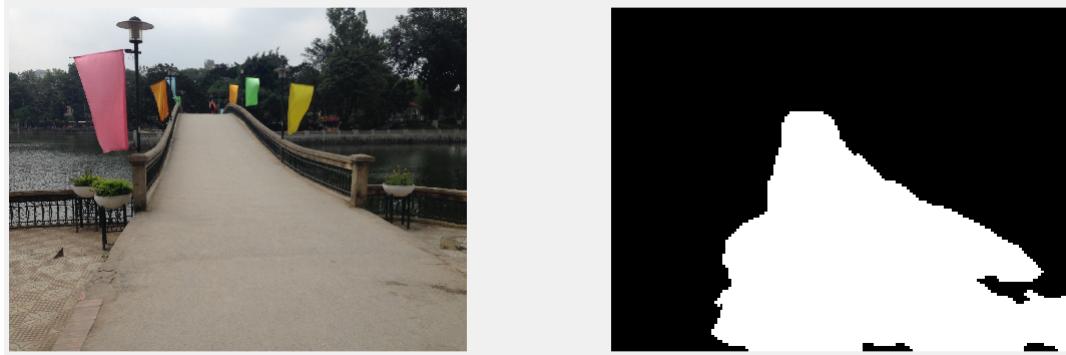


Figure 3.14: Detected lane

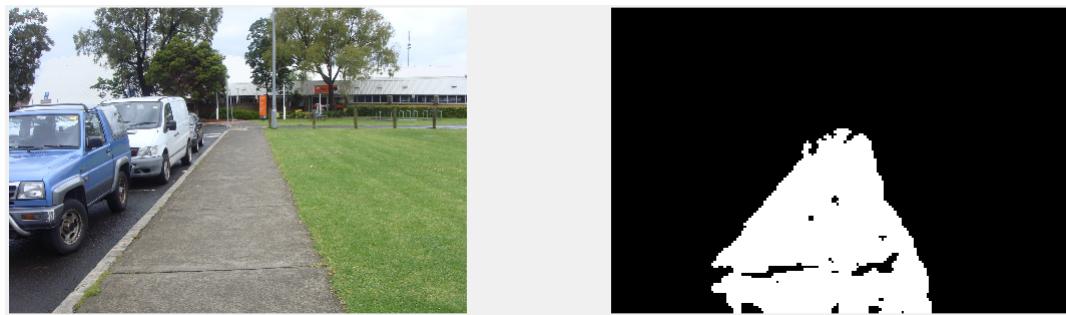


Figure 3.15: Detected lane

Greedy-search Algorithm

$$R' \leftarrow \{R_i \in R \mid p(\mathbf{c}_i \mid L) \geq \tau_c\}$$

$$Z^* \leftarrow \operatorname{argmax}_{R_i \in R'} p(\mathbf{c}_i \mid L)$$

continue \leftarrow TRUE

while *continue* **do**

$$R_{add} \leftarrow \{R_i \in \{R' - Z^*\} \text{ so that } Z^* \cup R_i \text{ is a connected set}\}$$

$$R^+ \leftarrow \operatorname{argmax}_{R_i \in R_{add}} g(Z^* \cup R_i)$$

$$R_{rmv} \leftarrow \{R_i \in Z^* \text{ so that } \{Z^* - R_i\} \text{ is a connected set}\}$$

$$R^- \leftarrow \operatorname{argmax}_{R_i \in R_{rmv}} g(Z^* - R_i)$$

if $g(Z^* \cup R^+) > g(Z^*)$ **and** $g(Z^* \cup R^+) \geq g(Z^* - R^-)$ **then**

$$Z^* \leftarrow Z^* \cup R^+$$

else if $g(Z^* - R^-) > g(Z^*)$ **then**

$$Z^* \leftarrow Z^* - R^-$$

else

$$\text{i}ncrease \leftarrow FALSE$$

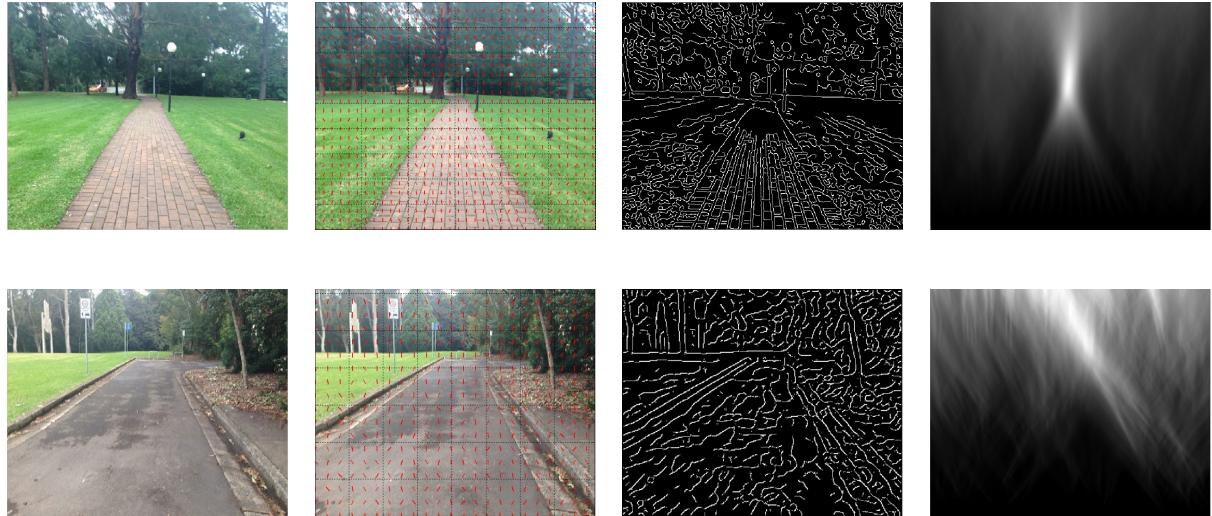
end if

end while

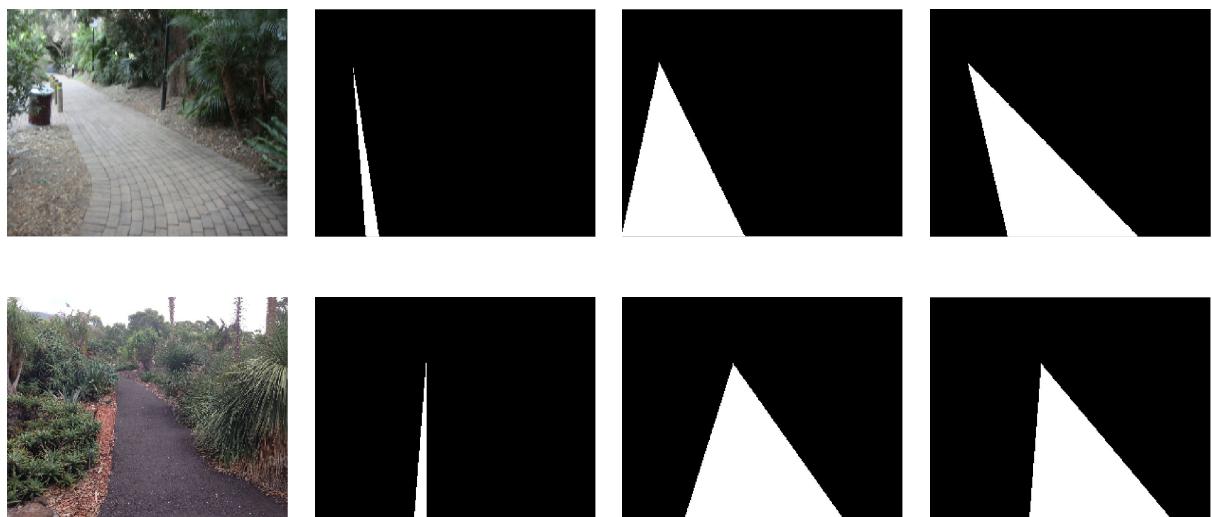
The greedy-search algorithm discussed above is applied on two lanes and the obtained pedestrian lanes are shown in the Fig:3.14, 3.15.

3.6 Results

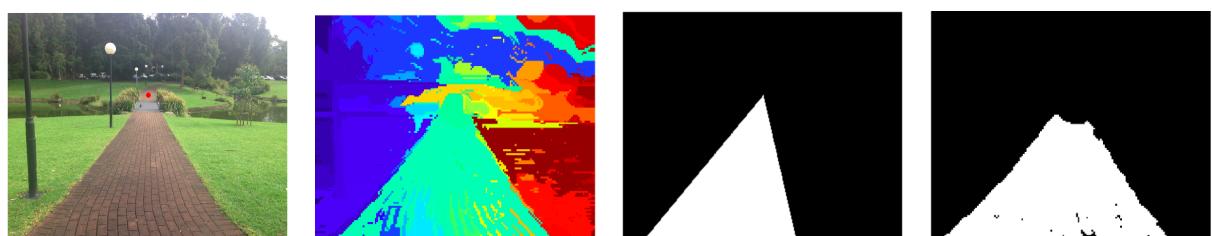
Vanishing point estimation

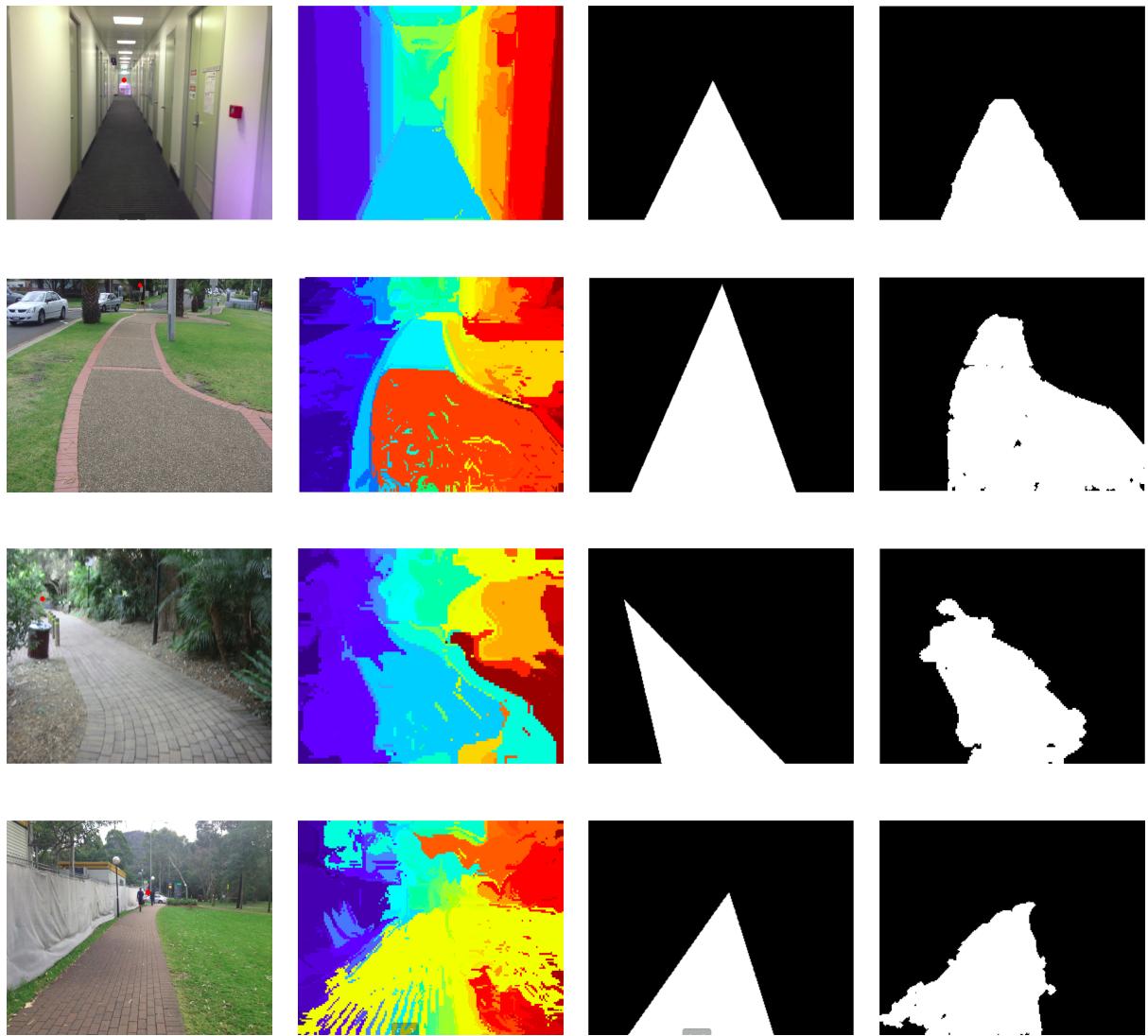


Significance of Area & Texture feature



Lane detection





3.7 Conclusion

A method for pedestrian lane detection in unstructured environments is implemented by combining color, edge and shape features and using the vanishing point to automatically determine a sample lane region. Robustness of sample region is enhanced by introducing Area and Texture feature that can be applied to textured, indoor and outdoor lanes. Results shows that sample lanes selected are good with inclusion of area and texture feature and the lane segmented outputs are observed. Graph-based image segmentation is used and is best suited for this project because it captures perceptually important components and it is efficient in terms of time complexity. Greedy-based algorithm is implemented to detect the lane effectively than that of conventional method.

REFERENCES

- [1]. R. E. W. Rafael C. Gonzalez, Digital Image Processing-Prentice Hall, Third edition. Pearson- Prentice Hall, 2008.
- [2]. S. L. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing using MATLAB - Second edition. Gatesmark Publising, 2009.
- [3]. M. C. Le, S. L. Phung, and A. Bouzerdoum, Pedestrian lane detection in unstructured environments for assistive navigation, in 2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8, Nov. 2014.
- [4]. S. Se and M. Brady, Road feature detection and estimation, in M. Machine Vision and Applications, vol. 14, pp. 157–165, 2003.
- [5]. J. Crisman and C. Thorpe, A color vision system that tracks roads and intersections, in IEEE Transactions on Robotics and Automation, vol. 9, pp. 49–58, IEEE, Feb. 1993.
- [6]. T. C. Ceryen Tan, Tsai Hong and M. Shneier, Color model-based real-time learning for road following, in IEEE Intelligent Transportation Systems Conference, pp. 939–944, IEEE, Sept. 2006.
- [7]. Z. G.-y. S. Yun and Y. Yong, A road detection algorithm by boosting using feature combination, in IEEE Intelligent Vehicles Symposium, pp. 364–368, IEEE, 2007.
- [8]. T. G. J. M. lvarez and A. M. Lpez, Vision-based road detection using road models, in 16th IEEE International Conference on Image Processing (ICIP), pp. 2073–2076, IEEE, Nov. 2009.
- [9]. C. Rasmussen, Texture-based vanishing point voting for road shape estimation, in Proceedings of British Machine Vision Conference, pp. 470–477, Sept. 2004.
- [10]. J. Y. A. H. Kong and J. Ponce, General road detection from a single image, in IEEE

Transactions on Image Processing, vol. 19, pp. 2211– 2220, IEEE, Aug. 2010.

- [11]. M. A. S. J. R. M. M. B. Boquete, Autonomous robots, vol. 16, pp. 95– 116, Jan. 2004.
- [12]. H. W. Yinghua He and B. Zhang, Color based road detection in urban traffic scenes, in Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems, vol. 1, pp. 730–735, IEEE, Oct. 2003.
- [13]. D. S. Yue Wanga, Eam Khwang Teoha, Lane detection and tracking using b-snake, in Image and Vision Computing, vol. 22, pp. 269–280, April 2004.
- [14]. J. M. lvarez and A. M.Lopez, Road detection based on illuminant in- variance, in IEEE Transactions On Intelligent Transportation Systems, vol. 12, IEEE, March 2011.
- [15]. Z. G.-y. S. Yun and Y. Yong, A road detection algorithm by boosting using feature combination, in IEEE Intelligent Vehicles Symposium, Istanbul, pp. 364–368, IEEE, 2007.
- [16]. J. S. C. Oh and K. Sohn, Illumination robust road detection using geometric information, in 15th International IEEE Conference on Intelligent Transportation Systems, pp. 15661571, IEEE, 2012.
- [17]. B. G. HumanWare, Url: store.humanware.com/hus/braillenote-gps-software-and-receiver-package.html., 2015.
- [18]. M. B. S. Se, Road feature detection and estimation, in Mach. Vis. App, vol. 14, pp. 157–165, 2003.
- [19]. T. S. M.S. Uddin, Bipolarity and projective invariant-based zebra crossing detection for the visually impaired, in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recog- nition Workshops, pp. 22–30, IEEE, 2005.
- [20]. S. H. V. Ivanchenko, J. Coughlan, Detecting and locating crosswalks using a camera phone, in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–8, IEEE, 2008.
- [21]. T. G. J. van de Weijer and A. W. M. Smeulders, Robust photometric invariant features from the color tensor, in IEEE Transactions on Image Processing, vol. 15, pp.

118–127, IEEE, Jan. 2006.

- [22]. J. A. S. P. Moghadam and W. S. Wijesoma, Fast vanishing-point detection in unstructured environments, in *IEEE Transactions on Image Processing*, vol. 21, pp. 425–430, IEEE, Jan. 2012.
- [23]. J. Shi and J. Malik, Normalized cuts and image segmentation, in *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 22, IEEE, August 2008.
- [24]. D. P. H. Pedro F. Felzenszwalb, Efficient graph-based image segmentation, in *International Journal of Computer Vision*, vol. 59, pp. 167–181, 2004.
- [25]. S. Belongie, J. Malik, and J. Puzicha, Shape matching and object recognition using shape contexts, vol. 24, pp. 509–522, Apr 2002.
- [26]. H. F. Ng and Y. W. Chu, Illumination invariant color model for image matching and object recognition, in *2008 Eighth International Conference on Intelligent Systems Design and Applications*, vol. 1, pp. 95–99, Nov 2008.