

Spring 2016

Pedestrian Detection Using Basic Polyline: A Geometric Framework for Pedestrian Detection

Liang Gongbo

Western Kentucky University, tliang130@gmail.com

Follow this and additional works at: <http://digitalcommons.wku.edu/theses>



Part of the [Artificial Intelligence and Robotics Commons](#), and the [Computer Engineering Commons](#)

Recommended Citation

Gongbo, Liang, "Pedestrian Detection Using Basic Polyline: A Geometric Framework for Pedestrian Detection" (2016). *Masters Theses & Specialist Projects*. Paper 1582.
<http://digitalcommons.wku.edu/theses/1582>

This Thesis is brought to you for free and open access by TopSCHOLAR®. It has been accepted for inclusion in Masters Theses & Specialist Projects by an authorized administrator of TopSCHOLAR®. For more information, please contact todd.seguin@wku.edu.

PEDESTRIAN DETECTION USING BASIC POLYLINE:
A GEOMETRIC FRAMEWORK FOR PEDESTRIAN DETECTION

A Thesis
Presented to
The Faculty of the Department of Computer Science
Western Kentucky University
Bowling Green, Kentucky

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
Gongbo Liang

May 2016

PEDESTRIAN DETECTION USING BASIC POLYLINE:
A GEOMETRIC FRAMEWORK FOR PEDESTRIAN DETECTION

Date Recommended 04/07/2016




Dr. Qi Li, Director of Thesis



Dr. James Gary



Dr. Zhonghang Xia

 4/18/16

Dean, Graduate Studies and Research Date

DEDICATION

I dedicate this thesis to my parents, Zhongyi Liang and Zhuoru Wang, who are a great inspiration to me.

Thank you for all the love I have been given.

I can never repay you.

ACKNOWLEDGMENTS

This thesis would not have been completed without the guidance of my professors, support of my friends and the love of my family.

I would like to express my deepest gratitude to my advisor, Dr. Qi Li, for his support, guidance, and patience during this research. His intelligence and encourage have made this research enjoyable and fruitful. I am very fortunate knowing and work with him.

I would like to sincere thank Dr. James Gary and Dr. Zhonghang Xia for serving on my committee, provided valuable advice, reviewing my work, and all other effort I have been given.

I would also like to thank the faculty and staff of the Department of Computer Science. With out all your help, my Master's student could not been completed.

Finally, I would like to thank my wife, Yang Liu. Thank you for always being at my side to support and encourage me.

CONTENTS

Chapter 1. Introduction.....	1
Chapter 2. Related Work.....	4
2.1 Haar wavelet.....	5
2.2 Histogram of Oriented Gradients	6
2.3 Global Chamfer Matching	7
2.4 Shapelets.....	7
2.5 Shape Context	8
Chapter 3. Basic Framework	10
Chapter 4. Extraction of 1-Piece Polylines	12
4.1 Dominant Point Detection	12
4.2 Piecewise Linearity Verification.....	13
4.3 Partitioning.....	13
Chapter 5. Pedestrian Detection	15
5.1 Hypothesis Box Generation.....	15
5.2 Amplify, Shrink, Shift Hypothesis Box.....	18
5.3 Constraints of Hypothesis Boxes	20
5.3.1 Boundary Constraint.....	21
5.3.2 Noisy Polyline Constraint	21
5.3.3 Leg Constraint.....	22
5.3.4 Arm Constraint.....	24
5.4.5 Head Constraint	26
5.4 Select of Output Hypothesis Boxes	28

Chapter 6. Experimental Design	29
6.1 Polyline extraction.....	29
6.2 Hypothesis Box Generation.....	30
6.2.1 Modify Hypothesis Box Size.....	30
6.2.2 Modify the Hypothesis Box Location.....	31
6.3 Output result.....	32
Chapter 7. Discussion	36
7.1 Test Dataset and Evaluation Measure.....	36
7.1.1 Test Dataset.....	36
7.1.2 Measurement.....	36
7.2 Detection Result.....	37
7.2.1 Detection Result.....	37
7.2.2 Comparing Result.....	38
7.3 Failure Analysis.....	39
7.3.1 Failure Cause by Asymmetric.....	39
7.3.2 Failure Cause by Output Selection Algorithm.....	41
7.3.3 Failure Cause by Occlusion and/or Crowd.....	42
7.3.4 Low Contrast.....	43
7.4 Future Study.....	43
7.4.1 More Detection Models.....	43
7.4.2 A Voting System/Scoring Hypothesis Box.....	43
7.4.3 Machine Learning and Classifier.....	44
Chapter 8. Conclusion	45
APPENDIX A: MORE DETECTION RESULTS WITH EXTRACTED LINES	46

APPENDIX B: COMPARISON RESULT WITH DALAL AND TRIGGS' ALGORITHM.....	50
Literature Cited	54

LIST OF FIGURES

Figure 1. Two handwritten digits.	9
Figure 2 The proposed framework.	11
Figure 3 Dominant points detection and connected component partitioning.....	14
Figure 4. Generate Hypothesis boxes.....	17
Figure 5 Hypothesis boxes for people in different position.	18
Figure 6. Amplifying and shifting a hypothesis box.	20
Figure 7 Parallel and Symmetric Lines for Pedestrians.	23
Figure 8 Three Scenarios of Arm Constraint.	25
Figure 9 Head Search Area.....	26
Figure 10 Three Scenarios of Head Constraint.	27
Figure 11 All hypothesis boxes after applied constraints.....	33
Figure 12 Detection results for three return methods.....	34
Figure 13 Some Detection Results.	37
Figure 14 Comparison Result.....	38
Figure 15 Failure caused by violation of symmetry.....	39
Figure 16 Failure caused by violation of symmetric.	40
Figure 17 Failure caused by output selection algorithm.	41
Figure 18 Failure caused by occlusion or crowd.....	42

LIST OF TABLES

Table 1. Comparison Result with Dalal and Triggs' Algorithm	39
---	----

PEDESTRIAN DETECTION USING BASIC POLYLINE:
A GEOMETRIC FRAMEWORK FOR PEDESTRIAN DETECTION

Gongbo Liang

May 2016

57 Pages

Directed by: Dr. Qi Li, Dr. James Gary, and Dr. Zhonghang Xia

Department of Computer Science

Western Kentucky University

Pedestrian detection has been an active research area for computer vision in recently years. It has many applications that could improve our lives, such as video surveillance security, auto-driving assistance systems, etc. The approaches of pedestrian detection could be roughly categorized into two categories, shape-based approaches and appearance-based approaches. In the literature, most of approaches are appearance-based. Shape-based approaches are usually integrated with an appearance-based approach to speed up a detection process.

In this thesis, I propose a shape-based pedestrian detection framework using the geometric features of human to detect pedestrians. This framework includes three main steps. Give a static image, i) generating the edge image of the given image, ii) according to the edge image, extracting the basic polylines, and iii) using the geometric relationships among the polylines to detect pedestrians.

The detection result obtained by the proposed framework is promising. There was a comparison made of this proposed framework with the algorithm which introduced by Dalal and Triggs [7]. This proposed algorithm increased the true-positive detection result by 47.67%, and reduced the false-positive detection number by 41.42%.

Chapter 1. Introduction

Nowadays, machines are becoming more and more important to people's daily life. How to empower a machine with the ability to interact with people is one of the most interesting and practical challenges for scientists and engineers. In order to interact with people in motion, machines are expected to have computer vision. Object detection and recognition has been an active research area in computer vision, and pedestrian detection or people detection is an important topic in the area of object detection. Automatic pedestrian detection has many applications including video surveillance security, auto-driving assistance systems, etc. For example, studies showed that 4,735 of 32,719 annual traffic crash fatalities involve pedestrians in the United States in 2013 [50]. The large number of traffic crash fatalities involving pedestrians an important motivation to build an auto-driving assistance system with the ability of detecting pedestrians [17].

In the literature, there are two basic types of schemes for automatic pedestrian detection. One scheme is based on exhaustive scanning of an input image under different scales with a sliding window; the other scheme is based on hypotheses by evidence. The following is a list of well-known pedestrian detection methods: shape context detection [15, 29, 30, 40], searching of particular patterns or body parts [5, 27, 28, 12], stereo vision and neural networks detection [43, 3, 10, 16, 20], global Chamfer matching detection [14, 21, 25, 36], detection using Haar wavelet and Haar-like wavelet [32, 17, 43,45], detection using the Histogram of Oriented Gradients feature [7, 8, 39, 44, 47],

detection using more than one well-known methods [42, 34, 38, 17], using general methodology of body plans [12], etc.

In this thesis, I propose a pedestrian detection framework that can detect pedestrians in static scenes, i.e., without motion input. It does not involve a training step either. The proposed pedestrian detection framework is based on geometric features of pedestrians that are designed upon 1-piece polylines, i.e., line segments. Pedestrians may have various appearances, under different lighting conditions, clothing styles, and poses. Some basic geometric features, such as the head-height ratio of a person, the distance from hip joint to shoulder, however, remain the same under different appearances. Existing pedestrian detection methods usually apply a learning process based on a large set of training images that cover various appearances of pedestrians and non-pedestrian objects/scenes to address this issue. This proposed framework is thus complementary to learning-based detection methods, and it may be integrated with the latter ones to reduce the search space and thus the computational cost [24].

In the proposed framework, we first generate an edge image of the input image, then extract the 1-piece polylines from the edge image using the extraction method introduced in [24]. Next, for each vertical polyline, we detect whether there is a parallel polyline existing within in certain distance. If a parallel polyline exists, we assume the vertical polyline is a potential leg. We generate several hypothesis boxes for each potential leg according to our pre-built models. At last, we apply different constraints to filter out the objects other than pedestrian.

The detection result obtained by the proposed framework is promising. I made a comparison of this proposed framework with Dalal and Triggs' algorithm. According to

the comparison, the true-positive detection rate of Dalal and Triggs' algorithm is 29.97%, and for mine is 44.25%, which increase the rate of Dala and Triggs' algorithm by 47.67%. The number of false-positives is reduces by 41.42%. See section 7.1 and 7.2 for comparison details.

Chapter 2. Related Work

Various approaches of pedestrian detection have been proposed in the literature. The approaches could be roughly categorized as shape-based approaches and appearance-based approaches. Most of approaches in the literature of pedestrian detection are appearance-based. Shape-based approaches are usually integrated with an appearance-based approach to speed up a detection process.

A shape-based pedestrian detection approach relies on the search for the specific characteristics of pedestrians, which might include vertical symmetry and strong presence of edges. In a shape-based pedestrian detector, pedestrians are more than likely to be discovered through human beings' strong vertical symmetry shape in the detected area, even including the people who have different poses, wear different clothing, and people on the move. Gavrilu proposed a distance transform based shape matching approach to reduce the brute-force search space of candidate regions of pedestrians [14].

An appearance-based pedestrian detection approach generally applies a classifier, such as a linear/kernelSVM, AdaBoost, Radial Basis Functions (RBFs), etc., to appearance features. Gavrilu applied a RBF classifier to a rich set of intensity features [14]; Papageorgiou et al. applied linear SVM to Haar wavelet coefficients [32, 31]; Dalal and Triggs applied linear SVM to the Histogram of Oriented Gradients (HOG) features [7]; Sabzmeydani and Mori proposed the so-called shapelet features that are a set of mid-level features and are created as a combination of oriented gradient response by AdaBoost [35]; Mohan et al. proposed a two-stage hierarchical classification scheme called Adaptive Combination of Classifiers (ACC) [28]. In Stage 1, four distinct

classifiers are constructed with respect to four body components (the head, legs, left arm, and right arm) and a predefined pedestrian shape model is applied to ensure the four detected body components are in the proper geometric configuration; In Stage 2, a classifier is constructed with respect to the combination (i.e., concatenation) of four body components [28]. Mikolajczyk et al. proposed a three-stage scheme for pedestrian detection [27]. In Stage 1, orientation features in multi scales are detected and grouped; In Stage 2, seven distinct classifiers are constructed with respect to seven body part; In Stage 3, a joint Probabilistic model is applied to the Assembly of detected body parts [27].

In the following section, several frequently used features in pedestrian detection are explained in detail.

2.1 Haar wavelet

Wavelet is a natural mathematical structure that provides description of patterns [25]. Haar wavelet is a popular method that is used for pedestrian detection such as [31, 32]. According to Papageorgiou and Poggio [32], it was the first people detection system described in the literature that does not rely on motion, tracking, background subtraction, or any other assumptions on the scene structure. Haar wavelets are able to identify locally oriented intensity difference features at different scales and are also efficiently computable [32].

Papageorgiou and Poggio proposed the Haar wavelet feature detection in [32]. The study introduced a dense overcomplete representation using wavelets, which were at the scale of 16 and 32 pixel and were 75 % overlapped. Low frequency changes in contrast, including vertical, horizontal, and diagonal, can be encoded by using three

different types. Therefore, the feature vector for a 64 : 128 pixel detection window was 1326 dimension long. Only the maximum response was used to cope with lighting differences for each color channel. Based on the window's mean response for each direction, normalization of each direction was performed. Additionally, because of clothing differences, according to the authors, there was no information in the class of people the wavelet coefficient's sign. Thus, the authors only retained the absolute values for each coefficient.

2.2 Histogram of Oriented Gradients

Since the local object appearance and shape can be characterized by the distribution of local intensity gradients, Dalal and Triggs introduced a pedestrian detection algorithm according to the Histogram of Oriented Gradients [7]. The fundamental idea was that by dividing image into small cells, accumulating a local 1-D histogram of gradient directions over the pixels of each cell, and then accumulating a measure of local histogram over some larger spatial blocks, the result could be used to normalize all the cells in the particular blocks, which were Histogram of Oriented Gradient (HOG) descriptors [7].

In [7], Dalal and Triggs introduced the HOG feature detection in details in their study. Centered differences in x- and y- direction were used to compute image derivatives. The 8 by 8 pixel cell histograms were formed and inserted by the gradient magnitude, which interpolate in x, y and orientation. Each block contained 2 : 2 cells, and each cell was overlapped by one cell in each direction. An additional hysteresis step was used to normalize the L_2 long blocks to avoid the domination of the feature vector of

one gradient entry. The resulting final vector contains all normalized block histograms, with a total dimension of 3780 for a 64 : 128 detection window.

2.3 Global Chamfer Matching

The global chamfer matching detection method matches object shape silhouettes with image structure. A silhouette will be shifted over into an image. A distance between a silhouette and the edge image at each image location will be calculated, based on a distance transform that was computed for each image pixel the distance to the nearest feature pixel. The lower the distances were, the better the matching of the image and the silhouette was.

According to Gavrilu, the foundation of the above system largely relied on shape matching using distance transform (DT). Various pedestrian appearances were modeled with silhouette templates. To Match a template T and an image I , steps of computing the feature image of I and applying a DT to obtain a DT-image were involved. A variety of DT algorithms were existing. The Chamfer transform is one of them, which computes an approximation of the Euclidean distance using integer arithmetic. After computing the DT, the relevant template T is transferred over the DT-image of I . The pixel values of the DT-image which lie under the pixels of the transformed template would determine the matching measure $D(T, I)$. The lower distance means the better matching between the image and template at this location [14].

2.4 Shapelets

The term of shapelet was coined by Refreiger in [33], which was used to name a series of localized basis functions for different shapes. Later, in [21] Kovesi applied this

term on referring a basis of finite support for describing shape. Sabzmeydani and Mori's introduced a learning algorithm using shapelet features to discriminate between pedestrians and non-pedestrians in [35]. They believed that the most important cue for pedestrians detection in static image would be the shape, especially the pieces of shape like the stereotypical omega pattern formed by head and shoulder of a pedestrian [35]. Their algorithm contains three steps:

1. Low-level Features: In each image, they extracted the gradient responses of different directions and then computed the local average of those responses around pixels. These low-level gradients were used to build shapelets.
2. Shapelet Features: They used Adaboost to select a subset of the low-level features to generate the shapelet feature in each of sub-windows in the detection window. The shapelet features consisted of the combination of gradients with different orientations and strengths at different locations inside sub-windows.
3. Final Classifier: The shapelet features could only describe local neighborhoods of an image; therefore the classification power was not strong enough. So, they applied AdaBoost again to train the final classifier, which used shapelet features as the input.

2.5 Shape Context

Shape Context was originally proposed in [1]. It has shown admirable potential on pedestrian detection using ISM framework [23, 37]. Belongie and Malik used the following words to introduce the idea of shape matching and object recognition using

shape context: “Regarded as vectors of pixel brightness values and compared using L_2 norms, they are [Figure 1] very different. However, regarded as shapes they appear rather similar to a human observer. Our objective in this paper is to operationalize this notion of shape similarity, with the ultimate goal of using it as a basic for category-level recognition [1].” The recognition goal that was mentioned in the statement was able to be achieved by using shape context according to Belongie and Malik.

In [1], Belongie and Malik defined that shape context is a shape descriptor that is used to describe the distribution of a point to rest points of the shape with respect to the given point. The problem of recognizing the similarity of two shapes could be transferred to the recognition of the correspondences between points on two shapes. A descriptor, shape context, is given to each point of the two similar shapes, of which the corresponding points should have the similar shape contexts.

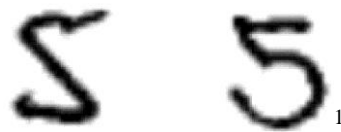


Figure 1. Two handwritten digits.

¹ The examples were borrowed from the work of Belongie et al. [1].

Chapter 3. Basic Framework

The pedestrian detection framework that is proposed in this thesis uses the geometric features of humans to detect pedestrians using basic polylines. It contains three main steps: i) edge detection, ii) 1-piece polylines extraction, and iii) generation of hypothesis boxes.

Edge detection is the first step of this framework. This step generates the edge image of an input image under multiple scales. There are many existing edge detectors that can be used for this purpose. We choose the Canny edge detector because it is one of the most commonly used edge detection algorithms in a wide range of applications.

The second step is the extraction of 1-piece polylines (in this thesis also called basic polylines) based on the edge image. The basic polylines extraction process could be done by three steps: i) dominant points detection, ii) piecewise linearity verification, and iii) partitioning. (Details are presented in Chapter 4.)

The last step of the proposed framework applies geometric relationships of the 1-piece polylines to detect pedestrian candidate regions. Specifically, we first generate hypothesis boxes for each vertical 1-piece polyline if there is a parallel vertical line existing within certain distance of the vertical basic polyline. Some adjustments might be needed after the hypothesis boxes generation, such as amplifying or shrinking the boxes, finding the best location of the box, etc. Finally we apply different constraints to those hypothesis boxes to eliminate the hypothesis boxes that do not contain pedestrians according to the geometry features of humans. Figure 2 illustrates the proposed framework.

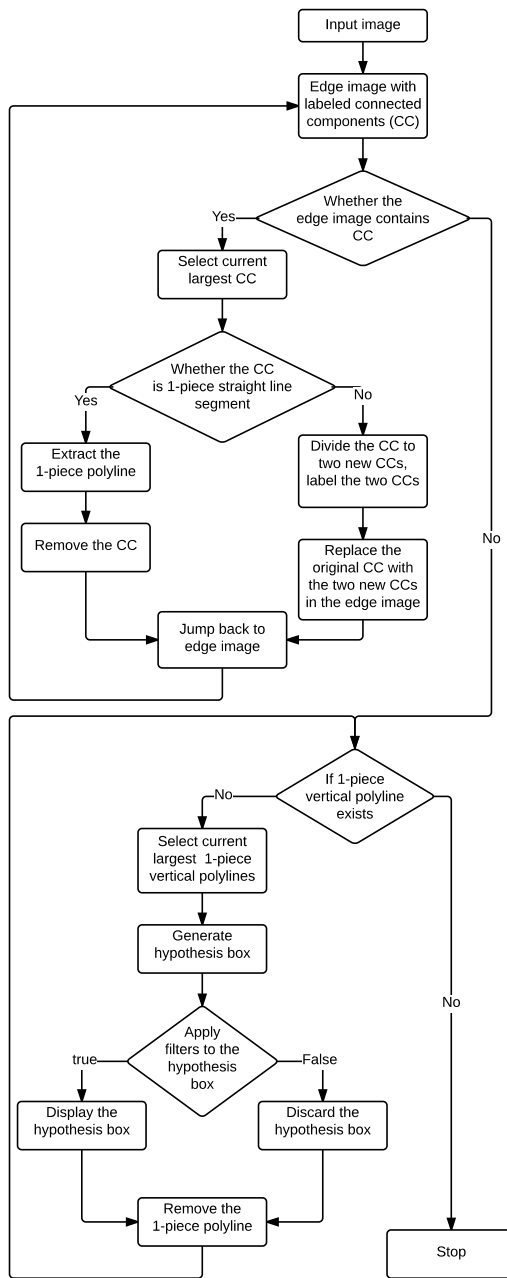


Figure 2 The proposed framework.

Chapter 4. Extraction of 1-Piece Polylines

In [24], we proposed the polylines extraction scheme for 1-/2-piece polylines. For this proposed pedestrian detection framework, we only used 1-piece polylines technique. 2-piece polylines might be added in the future. The explanation of the three consecutive steps of 1-piece polylines extraction follows.

4.1 Dominant Point Detection

Dominant points are the special points in a connected component. A non-linear connected component should have three dominant points. A linear connected component, however, will only have two dominate points.

Given a connected component P , denote v_1 and v_2 a pair of points in P such that their Euclidean distance is the maximum among all possible pairs of points, e.g.,

$$(v_1, v_2) = \operatorname{argmax}_{p_i \neq p_j} \|p_i - p_j\|,$$

where $\|\bullet\|$ denotes the 2-norm of a vector. v_1 and v_2 are then considered as the end points of the dominant axis of P , and thus two dominant points.

Then, we compute the third dominate point v_3 of P by maximizing the sum of the distance between p and v_1 and the distance between p and v_2 , i.e.,

$$v_3 = \operatorname{argmax}_{p \in P - \{v_1, v_2\}} (\|p - v_1\| + \|p - v_2\|).$$

If P is a straight-line segment, every $p \in P$ has the property that $\|p - v_1\| + \|p - v_2\|$ is a constant, i.e., $\|v_1 - v_2\|$.

For instance, in Figure 3, p_1 , p_2 , and p_3 are three arbitrary points in the connected component. Since the distance between p_1 and p_2 is the largest among the distances of

any other two points of the connected component, we say p_1 and p_2 are the first and second dominate points for the connected component. The point p_3 is the third dominate point because the sum of the p_1p_3 and p_2p_3 is the larger than the sum of distances from any other points to p_1 and p_2 .

4.2 Piecewise Linearity Verification

Given a connected component P , after three dominant points are found, we will then verify the piecewise linearity of the connected component. Specifically, given a point p and two vertices v_1 and v_2 , we compute the distance from p to the line segment v_1v_2 by

$$dist(p, v_1, v_2) = \left| (p - v_1) \cdot \frac{v_2 - v_1}{\|v_2 - v_1\|}^\perp \right|,$$

where \cdot is the dot produce, and $(a, b)^\perp$ is defined as $(-b, a)$. Given three vertices v_1 , v_2 , and v_3 , we call p on the boundary of a triangle or an *inlier* if

$$\min\{dist(p, v_1v_2), dist(p, v_1v_3), dist(v_2v_3)\} \leq \epsilon,$$

where ϵ is the displacement threshold (in pixel unit). In this framework, we use 2 pixels as the displacement threshold to verify linearity.

4.3 Partitioning

After linearity verification, if a connected component fails the verification, meaning that the connected component is not a straight-line segment, we need to apply partitioning on the connected component. In our proposed pedestrian detection framework, we simply have three potential partitioning lines, v_1v_2 , v_1v_3 , and v_2v_3 . Then, we use balancing, b , as the criteria to select the official partitioning line.

First, we apply three partitions with respect to the three partitioning lines. For each partition, we save the pixels on one side of the line in C_1 and the other the pixels on the other side of the line in C_2 . Next, for each partitioning line, we compute the balancing by

$$b = \frac{\min(C_1, C_2)}{\max(C_1, C_2)}.$$

Finally, we select the partitioning line with maxima value of balancing b as our partitioning line to partition the connected component.

For instance, according to the dominant points, which are detected in Figure 3, we can have three partitioning lines, p_1p_2 , p_1p_3 , and p_2p_3 . Since the balancing value of p_1p_2 is much closer to 1 than other two, which means that it has the maxima value of balancing among the three partitioning lines, line p_1p_2 will be chosen as the partitioning line for this connected component. (See Figure 3)

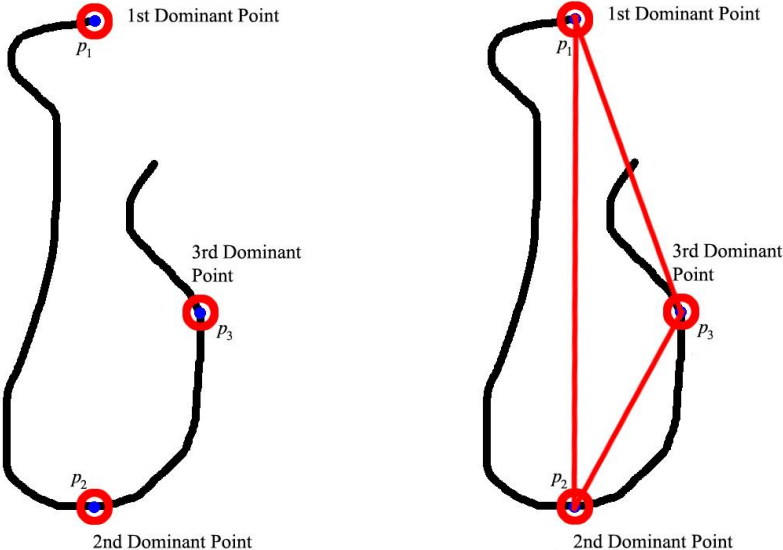


Figure 3 Dominant points detection and connected component partitioning

Chapter 5. Pedestrian Detection

Pedestrians usually have some distinguishing geometric features, which makes pedestrians distinguishable from other objects that we can see on streets, such as high symmetry in front view and profile view, similar body proportions among different people, and parallel or symmetric edges for legs and arms. In this proposed geometric pedestrian detection framework, we will take advantages of these kinds of features.

The basic pedestrian detection includes the following tasks: i) hypothesis box generation, ii) hypothesis box modification and localization, iii) hypothesis box filtration according to predefined constraints, and iv) hypothesis boxes output selection.

5.1 Hypothesis Box Generation

Hypothesis boxes are the core objects for this proposed framework. A hypothesis box is similar to a bounding box, which is widely used to bind an object, with only one difference between them. For every bounding box, it must bind at least one object inside the box. Hypothesis box, on the other hand, may or may not contain any objects inside it. In this framework, hypothesis boxes are generated for every vertical 1-piece polylines if the polyline has a parallel basic polyline existing within certain distance. If a hypothesis box satisfies all the constraints, it might be outputted as the bounding box that is used to bind the detected pedestrian.

For each vertical 1-piece polyline, l , which are extracted in the previously step, two kinds of hypothesis boxes will be generated. One type of hypothesis boxes, hb_l , suggests that the polyline l is one edge of a leg or a portion of a leg in the front view of a

pedestrian. The other type of hypothesis box, hb_2 , assumes that polyline l is one edge of a leg or a portion of a leg in a profile view of a pedestrian.

The size of hb_1 is initial as $2.2l : 0.55l$. The size of hb_2 is initial as $2l : 1.5l$.

Usually the leg-to-height ratio of an adult is 1:2 [4]. However, for different people, the leg-to-height ratio might be various, but the differences should not be large. We initially make an assumption that the polyline l is an edge of a leg. If the assumption is true, the height of this pedestrian should be $2l$. Since in the real word the ratio maybe various, we lengthen the height by 10% to $2.2l$. The leg-to-width ratio usually is around 1:0.5, according to [4]. For the same reason with the leg-to-height ratio, we widen the width of leg by 10% to make the leg-to-width ratio to 1:0.55. Then, we got the initial hypothesis box size of a pedestrian's front view, which is $2.2l : 0.55l$.

The hypothesis boxes of profile view are generated based on the same theory. The maximum width of any person in profile view and front view is the person with open arms. According to Leonardo da Vinci's Vitruvian Man [6], the background of Figure 4, the width of a person with open arms equals to the height of the person. Let us assume the length of leg is l , and the height of this person equals to $2l$; therefore, the maximum width of this person in any situation should be no wider than $2l$. As we all know, when a pedestrian is under a normal walking position in profile view, the arm movement should less than 180 degree, which means usually the angle of two arms should not equal or over 180 degree, in the other words, the wide of a pedestrian will less than $2l$, if the height of the pedestrian is $2l$. Therefore, we chose $1.5l$ as the wide of the hypothesis box of profile view. When any pedestrian is walking, the overall height from ground to the top of the head of the pedestrian will change all the time. However, it

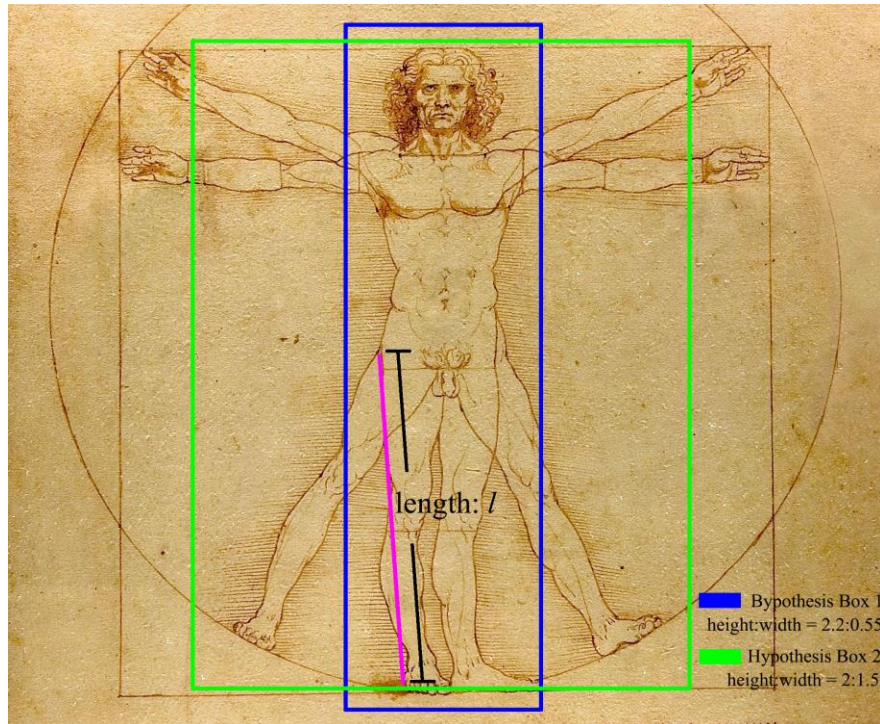


Figure 4. Generate Hypothesis boxes²

should be smaller than the height of the pedestrian in a straight up standing position. Therefore, we chose the height of pedestrians in a straight up standing position as the height of the hypothesis box of profile view.

For instance, in Figure 4, the pink line segment is one edge of a leg, which the length is l . The blue box is the hypothesis box for front view of this person, of which height-width-ratio equals to 2.2:0.55. The green box is the hypothesis box of profile view, of which the height-width-ratio of profile view hypothesis box is 2:1.5.

Figure 5 illustrates the hypothesis box for pedestrians in different positions in different views.

² The background image of Figure 4, it was cropped of *Vitruvian Man*, a drawing by Leonardo da Vinci around 1490. The drawing is used to show the proportions of man.

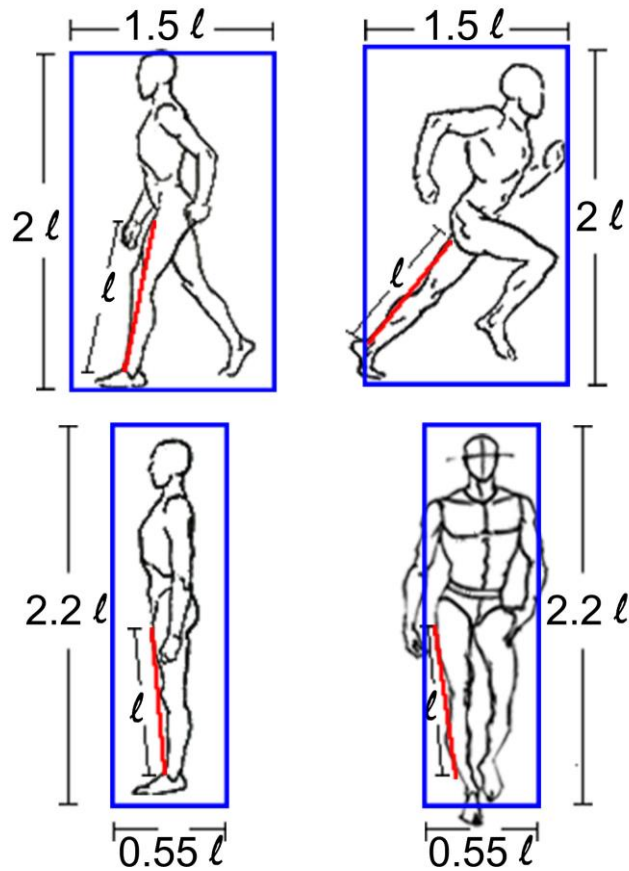


Figure 5 Hypothesis boxes for people in different position.

5.2 Amplify, Shrink, Shift Hypothesis Box

Initially, we make the assumption that a vertical basic polyline l is one edge of a leg. In the real world, however, things are usually not perfect. In most cases, it will be lucky if we can extract a polyline that perfectly fits a leg. If the length of l is not equal to the leg, and if the hypothesis boxes are generated according to l , meaning that the size of the hypothesis boxes are inaccurate. They are either larger or smaller than the ground truth hypothesis boxes. To address this potential issue, it is necessary to amplify or shrink a hypothesis box based on real situations.

Similarly, in the initial step, we only assume that a polyline l is one edge of a leg. However, a pedestrian usually has two legs, resulting four edges. We have not specified which leg l belongs to or which edge of the leg it is. To address this potential issue, the hypothesis box needs to be shifted to the left and to the right in order to find the best location.

Humans are generally symmetry. Pedestrians are either walking or standing will inherit the symmetry feature of humans. Thus, we take advantage of this feature when we adjust a hypothesis box. When we amplify, shrink, or shift a hypothesis box, we want to select its size and location, which will make the hypothesis box most symmetrical regarding to polylines inside the hypothesis box. (Figure 6 shows of the operations of amplifying and shifting a hypothesis box.)

In Figure 6, the pink line segment is the extracted basic line segment for one edge of one leg. Since it is less than the length of the whole leg, the generated hypothesis box according to that line segment is smaller than the ground truth hypothesis box, which therefore needs amplification. However, it is still not clear whether it belongs to the left leg of the pedestrian or the right leg, as well as whether it is an inside edge or outside edge. Thus, the hypothesis box needs to be shifted. From the second figure to the fourth figure in the first row, it shows the amplifying process of the original blue hypothesis box. The four figures in the second row show the steps of shifting the hypothesis box. The red hypothesis box in the last figure in Figure 6 is the returned hypothesis box with correct size and correct location after the amplification and shifting.

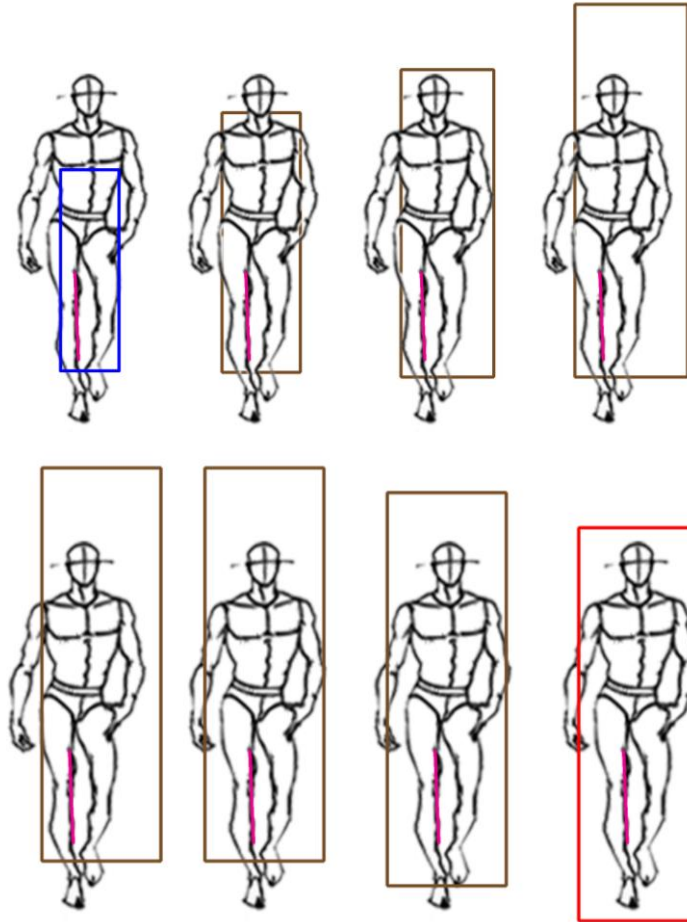


Figure 6. Amplifying and shifting a hypothesis box.

5.3 Constraints of Hypothesis Boxes

One of the criteria that the proposed geometric framework relies on is symmetric figure of human. Since a human is generally symmetric, this feature is a very good criterion to distinguish humans from other objects on the street. However, many other manmade or nature objects may be symmetric, e.g., poles, chairs, cars, tables, and the front view of many objects, bicycles, dogs, cats, and birds. In order to eliminate other symmetric objects when detecting humans, we need to apply various constraints to the hypothesis box, which allows us to reduce the overall time complexity as well.

In this section, I will introduce five constraints in the order of the application in the framework. They are: i) boundary constraint, ii) noisy polyline constraint, iii) leg constraint, iv) arm constraint, and v) head constraint.

5.3.1 Boundary Constraint

The boundary constraint is used to check the boundary of hypothesis box. If a hypothesis box is out of the boundary of the image, then discard the hypothesis box.

5.3.2 Noisy Polyline Constraint

This constraint is used to eliminate two types of noisy polylines. One is the polylines that are shorter than certain length, and the other one is polylines which are longer than specific percentage of the height of the hypothesis box.

For this proposed pedestrian detection framework, longer polylines such as polylines for legs and arms, will have more contribution to the human detection than shorter ones, such as a polyline of finger, but we still do not want to have those polylines which are longer than certain percentage of the height of the hypothesis box because these long line segments might not belong to pedestrians in the hypothesis box.

If polyline l is an edge of a leg of a specific pedestrian, and leg-to-height ratio is around 1:2. Thus [4], the height of this person is $2l$. The longest line segment we can extract from this pedestrian should not longer than 85% of $2l$, which equals the distance from foot to shoulder.

Assume the head height is h and the length of leg is l for a particular person. According to [4] the head-to-height ratio for human is usually between 1:5 and 1:7. If the head-to-height ratio is 1:7 and leg-to-height ratio is 1:2, then leg-to-head ratio with respect to a person is $2l:7h$ =height-of-a-person. When this person is standing or walking

without carrying any long object, the longest straight-line segment we may be able to extract from a person is the line from foot to shoulder. If we denote the longest straight-line as l_2 , then $l_2 = 7h - 1h = 6h$, since $h = 2l \div 7$, then $6h = \frac{6}{7} \times 2l \approx 1.7l = 0.85 \times 2l$. Thus, if any single polyline longer than $1.7l$ or 85% of $2l$ in a hypothesis box, it means this polyline is not from this person. Thus, we discard this polyline.

Short noisy polylines might cause extra work. Thus, short noisy polylines need to be eliminated as well. Any line segments that less than $1/16$ of l (l is an edge of a leg of a specific person) are considered as noisy lines, and all the noisy lines will be discarded. $1/16$ of l is similar with the length of one eye of this person. In any $640 : 480$ pixels image, if a person is located in the center of the image, and the height of the person is about $1/2$ of the height this image, the length of his or her leg will be around 120 pixels, and $1/16$ of the leg length is 8 pixels. It is a reasonable size for noisy lines.

5.3.3 Leg Constraint

In a normal case, a pedestrian usually has two legs; for each leg we should be able to extract at least two edges, one inside edge and one outside edge. If the knee is bent, four edges from one leg may be found, including the inner and outside edges for both thigh and shin. Thus, at least four line segments are supposed to be extracted for each pedestrian. In addition, the two edges from one pair of inner and outside edges, such as the inner and outside edges of left shin, should be general parallel to each other, and two edges from different legs could be general either parallel or symmetry to each other when people are in natural walking or standing positions.

In real-world situation, it is difficult to extract exactly four edges for any pedestrian due to some unexpected situations. For example, one leg of a pedestrian is

blocked by some other objects or other person. It is also challenging to extract a line segment that covers exactly the whole leg including thigh and shin since most time pedestrian need to bend their legs while walking. However, the parallel or symmetry features still exist among the lines of legs. (See Figure 7.)

We have three criteria to determine whether a leg is existing in a hypothesis box . First, if there are two pairs of parallel or symmetric polylines existing in the bottom half of a hypothesis box, we assume the original polyline l which is used to generate this hypothesis box is an edge of a leg; secondly, if there are one pair of parallel polylines and one pair of symmetry polylines existing in the bottom half of the hypothesis box, we assume the original polyline l which is used to generate this hypothesis box is an edge of a leg as well; thirdly, if there are one pair of parallel polylines or one pair of symmetry polylines existing in the bottom half of the hypothesis box, we assume the original polyline l which is used to generate this hypothesis box may be an edge of a leg. We

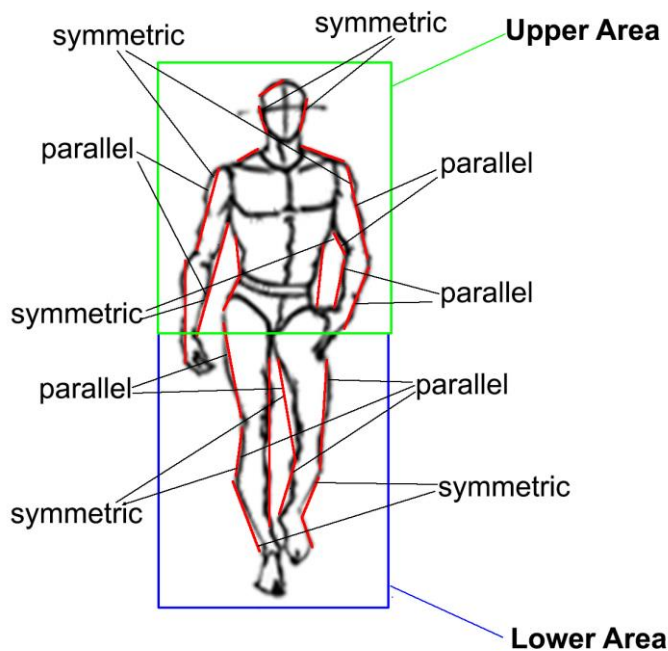


Figure 7 Parallel and Symmetric Lines for Pedestrians.

keep the hypothesis box but give lower confidence score to it. (The confidence score will be discussed in the Discussion section 7.4.2.) If none of the three patterns exists in the bottom half of a hypothesis box, we assume that the polyline l is not from a leg, and the hypothesis box will be discarded.

5.3.4 Arm Constraint

Similar with the constraint of leg, this constraint is used to filter out hypothesis boxes which do not contain any possible arms. This constraint keeps any hypothesis box that possibly has arms, but a lower confidence score is given to the hypothesis box. If a hypothesis box does not satisfy this constraint, it will be discarded.

For the front view of a pedestrian, unlike legs, the inside edges of arms might not be easy to detect by edge detectors. In many cases, the inside edge might be blocked or partially blocked by the body of the person or undistinguishable because the cloth color of arms and body are close to each other. However, the outside edge is more than likely to be able to detect. For a normal walking or standing person, without carrying or holding any objects, the two outside edges of two arms of a specific person generally has two different relationships: i) parallel to each other, and ii) symmetric to each other.

For any hypothesis boxes of front view for pedestrians which contains three polylines a_1 and a_2 in the top half of the hypothesis box, i) if a_1 is parallel or symmetric to a_2 , and if the Euclidean distance of a_1 and a_2 are similar to $\frac{1}{2}l$ (l is the leg length of this person), then these two polylines are two arms; ii) if a_1 is not parallel nor symmetric to a_2 , but the Euclidean distance of a_1 and a_2 are similar to $\frac{1}{2}l$, these two polylines may be two arms, and we keep the hypothesis box, but we give it a lower confidence score to this hypothesis box; iii) if there are some line segments, l' , exist in

the top half of the hypothesis box, and if the vertical Euclidean distance from one end of l' to the top vertex of l or the top vertex of the mirror image of l (l is the polyline of the leg of this person) within certain threshold, l' is possible an arm, but we give the hypothesis box a even lower confidence score than number ii. For number iii, the distance from hip joint to shoulder is used as the threshold. Even for different people, the ratio of distance from hip joint to shoulder and the distance from foot to hip joint are similar. Thus, if the distance from one vertex of a line (v_1) in the top half of the hypothesis box to the hip joint (the top vertex of l), equals to the threshold, it is possible that the v_1 is shoulder.

If a hypothesis box does not satisfy any of the three conditions, then we say there are no arms in the hypothesis box, indicating that the hypothesis box does not include any person. The hypothesis box will be discarded under such circumstance. (See Figure 8.)

For profile view of a person, the similar conditions are used as well. The only difference is that the Euclidean distance of a_1 and a_2 is much smaller than front view of this person.

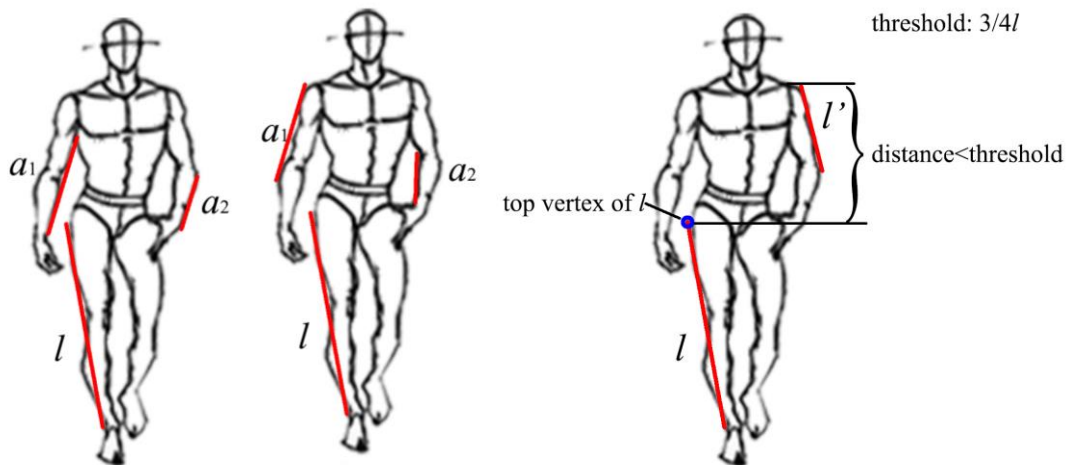


Figure 8 Three Scenarios of Arm Constraint.

5.4.5 Head Constraint

Head is another characteristic that is used to identify a pedestrian. Head feature plays important role in several existing pedestrian detection algorithms, such as [27, 28]. From observing edge images and polylines extracted by the edge images, some patterns of heads are identified. The following features will be used for this constraint in order to determine a head.

First of all, a search area for head needs to be defined. In many cases, collars might appear distinguishable symmetric pattern. Thus, we want to include neck and upper chest in the head search area as well. Due to this reason, the height of the head search area is set 50% longer than the head. Pedestrians may wear hats. We want to consider this situation, too. Thus, the width of the head search area is set also 50% wider than the head. For instance, if the head height equals about $\frac{1}{4}$ of l , l is the length of leg,

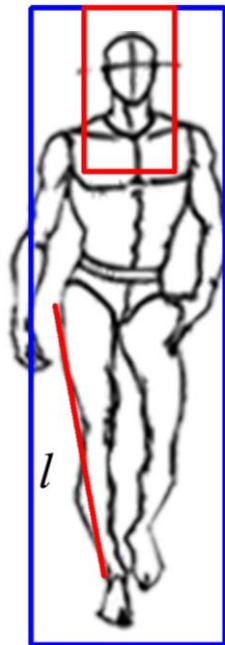


Figure 9 Head Search Area.

and the width of the head equals $\frac{1}{6}$ of l . Thus, the dimension for the head search area for this person is $\frac{1}{4}l$ for width and $\frac{3}{8}l$ for height. Since the height of a hypothesis box is generated as 10% taller and wider than the person, additional search area need to be added. Therefore, the final head search area dimension is $0.3l$ wide and $0.575l$ high. The head search area starts from the top of the hypothesis box and is placed in the horizontal center of the hypothesis box. In Figure 9, polyline l is the leg; blue box is the hypothesis box generated according to l ; the red box is the search area of head.

Generally speaking, we are searching three types of patterns in the searching area:

i) pattern of heads and collars, ii) pattern of heads or collars, and iii) any line segments in the area. For pattern of head, at least one pair of parallel polylines or at least a pair symmetric polylines should be observed. For pattern of collar, we look are interested in a pair of symmetric polylines. (See Figure 10 for more illustration.) If pattern i exists in

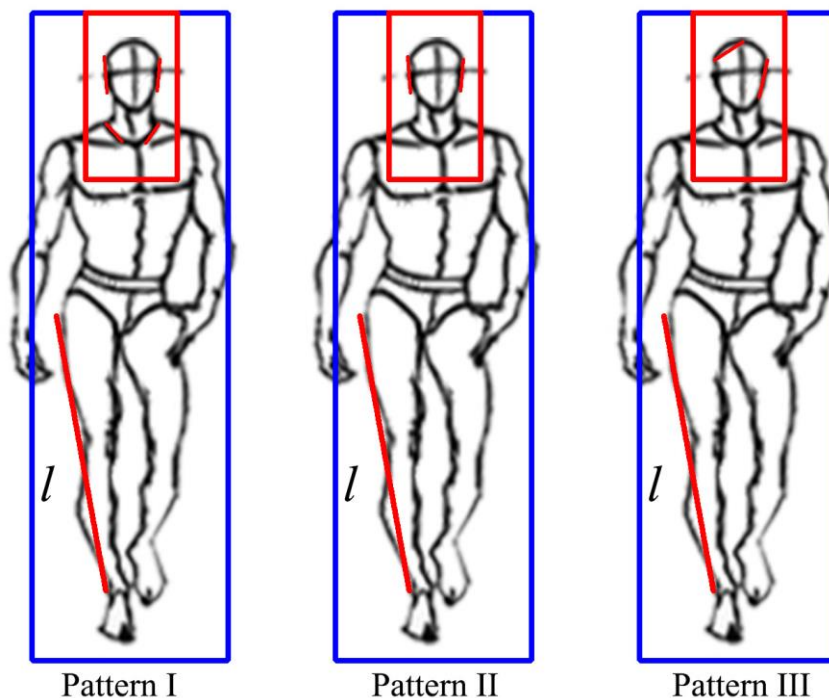


Figure 10 Three Scenarios of Head Constraint.

the search area, there is a head in the hypothesis box. Thus, we will say there is a person in the hypothesis box. Similarly, if pattern ii exists, there is a head in the hypothesis box. Thus, we believe there is a person in the hypothesis box, but a lower confidence score to the hypothesis box will be given to it. If only pattern iii exists, there is possibly a head in the hypothesis box, which means there is probably a person in the hypothesis box. We will keep the box, but an even lower confidence score will be given to this hypothesis box. However, if none of the patterns exist in the search area, we assume there is no head in the hypothesis box. Thus, the hypothesis box does not include any pedestrians. The hypothesis box will be discarded.

5.4 Select of Output Hypothesis Boxes

The last step of the pedestrian detection step is how to output the hypothesis boxes. There are many hypothesis boxes was generated during the detection process. Even after all the constraints are applied, it is possible that there are more than one hypothesis boxes left for one pedestrian. Thus, the most representative hypothesis box for the pedestrian should be selected as the output result.

We first cluster all the hypothesis boxes for an image. For each cluster, we select the most symmetric hypothesis box as the output result. Then, we output all the selected hypothesis boxes from different clusters as the final result of the pedestrian detection process.

Chapter 6. Experimental Design

6.1 Polyline extraction

Our original goal is to extract both 1-piece and 2-piece polylines, and use both 1- /2-piece polylines for detection. We were going to follow approach of [24]. However, not long after the research is started, we started to think that maybe we do not have to apply 2-piece polylines to this framework.

For each connected component in an edge image, we first detect the two dominant points. After the first two dominate points are detected, piecewise linearity verification will be preformed by first generating a line segment, l , which uses the two dominate points as vertices. Then, for each point, p , of the connected component, we compute the Euclidean distance, d , from p to l . If d is smaller or equal to the predefine threshold, t , the connected component is linear. It is considered as a 1-piece polyline. The linearity threshold for this framework is 2 pixels.

If d is larger than t , the connected component is not linear. We need to partition the connected component. For each non- linear connected component, we have three partitioning lines, which connected to dp_1 , dp_2 , and dp_3^3 , denoting as $\overrightarrow{dp_1dp_2}$, $\overrightarrow{dp_1dp_3}$, and $\overrightarrow{dp_2dp_3}$. Then, we select the partitioning line with the most balanced partitioning result. The most balanced means the number of pixels on one side of the partitioning line (pixNum_1) is most similar to the number of the pixels on the other side of the

³ dp_1 , dp_2 , and dp_3 represent the first dominate point, the second dominate point, and the third dominate point, respectfully. Non-linearity connected component has three dominate points. (See section 4 for detail)

partitioning line (pixNum_2). For each of the three partitioning line, we compute the balance, b . If pixNum_1 is not equal to pixNum_2 . Then,

$$b = \frac{\min(\text{pixNum}_1, \text{pixNum}_2)}{\max(\text{pixNum}_1, \text{pixNum}_2)}$$

The partitioning line with maxima value of b is the partitioning line for this connected component.

6.2 Hypothesis Box Generation

Hard coded width-height ratio is a big disadvantage of the current hypothesis box generating system, which is heavily relying on edge detection and line segment extraction. This relying-on relationship raises two issues: i) the generated hypothesis box might be too large or too small; ii) the generated hypothesis box may not align properly with the pedestrian.

Section 6.2.1 will discuss the strategy of amplifying or shrinking hypothesis boxes to address Issue i; Section 6.2.2 will elucidate solving Issue ii by shifting hypothesis boxes around.

6.2.1 Modify Hypothesis Box Size

To address the Issue i, we amplify or shrink is applied to each hypothesis box to find the most suitable size to cover the pedestrian. In order to better achieve the goal, we adopt the supersampling strategy.

For each extracted vertical line segment, l , we would create several samples with a variety of lengths. Typically, the samples range from 50% of l to 200% of l . Then, hypothesis boxes of each sample are generated. To choose which hypothesis box is the most suitable one for this pedestrian among all the hypothesis boxes that are generated

by the samples, we need to use the parameters of top-bottom weight balance and left-right balance (symmetry), which will be introduced in Section 6.2.2. Basically, the most balanced hypothesis box is the chosen one.

6.2.2 Modify the Hypothesis Box Location

To solve Issue ii, we need to shift the hypothesis box around.

Top-down Weight Balance

To identify the best vertical location of hypothesis boxes, a top-down weight balance is used as criteria. The implementation of this top-down weight balance is fairly simple. We count the number of pixels in the top half of the hypothesis box and the bottom half of the hypothesis box. If both of the numbers are not equal to zero, compute the quotient. The hypothesis boxes are ranked from one to zero according to the quotients. Then, certain number of hypothesis boxes is returned as candidate hypothesis boxes to enter the step of symmetric verification.

Symmetry

For finding the best horizontal location of hypothesis boxes, we tried two different methods to find out the most symmetric location of each hypothesis box. The first method is left-right weight balance. Similar with the top-down weight balance explained before, for this method we count the number of pixels in left part and right part of the hypothesis box, and then compute the quotient. This method is easy to implement; however, it is still relying on the edge detection and 1-piece polyline extraction algorithms.

The second method is an based on the histogram of gray values symmetry detection method introduced in [2]. For each hypothesis box, firstly, extract the image

covered by the hypothesis box and create a new image using the extracted portion; secondly, change the new image to gray scale image; next, divide the gray scale image to two parts, left part and right part; then, compute the histogram of gray values of each of the two parts; finally, apply dot product to the two values to verify the similarity of the two parts. The result for dot product should be between 0 and 1. 0 means completely different, and 1 means identical to each other.

6.3 Output result

Depending on the size of an image, there may be over one thousand vertical lines extracted, which means there might be over one thousand potential hypothesis boxes generated. Even after applying various constraints, there may still be a large number of hypothesis boxes left. Output of the right hypothesis boxes becomes the last task for this detection framework. (See Figure 11)

The idea is: first, cluster all the hypothesis boxes into different clusters; then, return the most representative one from each cluster.

A few attempts were made for clustering, but several attempts were made to explore the most accurate way for the selection of output hypothesis boxes. I am going to explain them in the following part.

Clustering

Firstly, we rank all the hypothesis boxes from largest to smallest and create a list, which is denoted as *hb_list*, to list all the hypothesis boxes. Then, select the current largest hypothesis box on *hb_list*, denoting it as *hb_l*. Next, create a clustering of *hb_l* and add itself into the clustering. Fourthly, compare all other hypothesis boxes, *hb*, with

hb_l . For each hb , if hb_l covers more than 50% of hb , add the hb into the clustering of hb_l and remove hb from hb_list . After comparing with all the hypothesis boxes in hb_list , remove hb_l from the hb_list as well. Lastly, check the hb_list , if there are any other hypothesis boxes left on the list, repeat the procedure again; otherwise, the clustering is completed.

Best Hypothesis Box Selection

Several attempts are made to select the output hypothesis box, which represents the cluster best. The first attempt uses the median. Since we know in any of clusterings or groups, the member on either end will not present the group best. Therefore, we use the median member as the output result for each clustering. It is an easy implementation, and quick attempt. Right after the implementation, we find out that this method does not work. The reason is that in many clusterings, the output hypothesis box selected by this method does not represent the clusterings well.

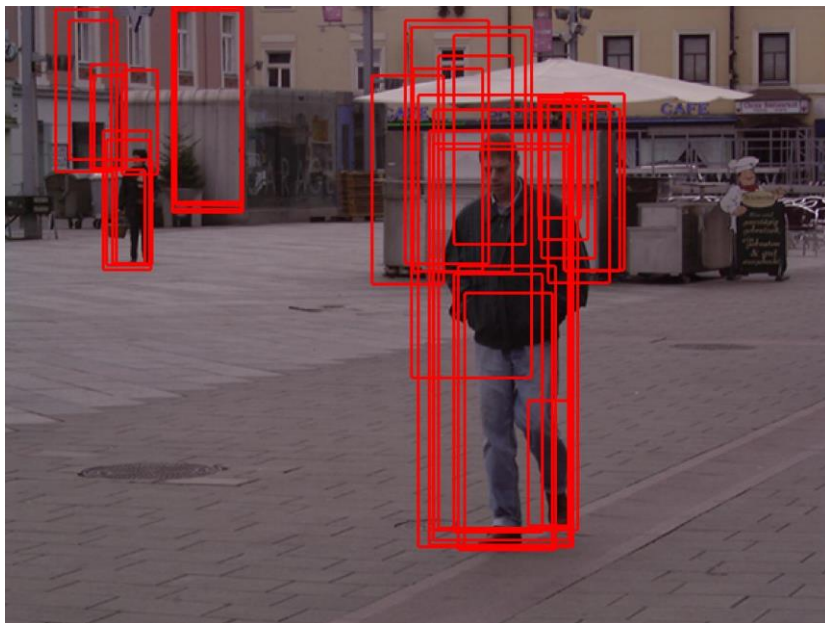


Figure 11 All hypothesis boxes after applied constraints

Next method is to use head as the criteria to select the returning hypothesis box. We check the head area for each hypothesis boxes in the clustering, and select the hypothesis box of which the head area is the most symmetric. This method is much better than the one we tried before. However, another strategy occurred to us while we are implementing the second method.

The last method is an extension of the second attempt. We apply symmetric checking to the whole hypothesis box. Since humans are general symmetric, if the hypothesis box covers a person, the head area should be symmetric, too. However, it is possible that a hypothesis box is not covering any person, but accidentally the head area of

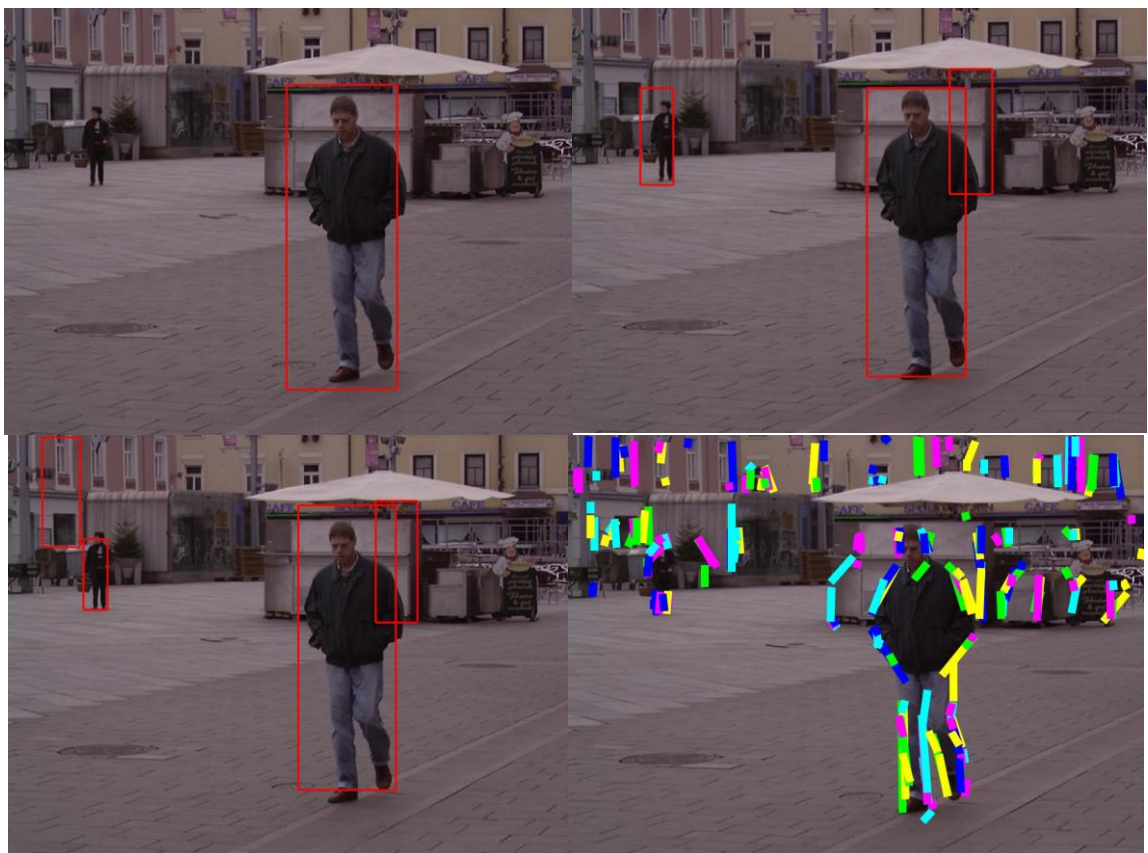


Figure 12 Detection results for three return methods.

The first image was returned according to the median member method (1st method), the second image was returned by head symmetry (2nd method), the third image was returned by the symmetry for entire hypothesis box, the fourth image shows the extracted line.

this hypothesis box is symmetric. Therefore, when applying the symmetric checking to the whole hypothesis box, not only the head area is applied but other relevant area is applied too. This will prevent this scenario from happening. (See Figure **12** for the result of the three return methods.)

Chapter 7. Discussion

7.1 Test Dataset and Evaluation Measure

7.1.1 Test Dataset

For this thesis the testing data set used is a subset of INRIA Person Dataset. INRIA Person Dataset is one of the most prevalent datasets used to test pedestrian detection algorithms for still images. Some other datasets might be used in the future, including Caltech, Caltech-Japan, ETH, TUD-Brussels, and Daimler. According to the website of INRIA Personal Dataset, the images of this data set were taken from personal digital image collections over a long time period. The majority of the images have high resolution. Some of the images were cropped. Few of the images were taken from the Internet using Google images. [49] For the subset of INRIA Person Dataset, which we used to test the algorithm, we manually selected 155 positive images, and randomly selected 187 negative images.⁴ For the positive images, we are focusing to the images with pedestrians wearing normal dress, regular cloth and pants, the front view of pedestrians, and the image was taken at the eye level. There are total 287 pedestrians among the test images.

7.1.2 Measurement

For this paper, the single frame evaluation measurement introduced in the PASCAL object detection challenges [10] is used as the measurement to determine

⁴ Positive images are the images contain at least one pedestrian inside it. Negative images are the images do not contain any pedestrian.

whether the detection is successful or not. The basic rationale of this measurement is to compare the output hypothesis box, BB_{hb} , of each person in an image with the ground-truth bounding box, BB_{gt} , of the person. If the intersect of BB_{hb} and BB_{gt} , divided by the union of BB_{hb} and BB_{gt} larger than 50%, the detection is successful. The computation can follow the formula below:

$$ratio = \frac{BB_{hb} \cap BB_{gt}}{BB_{hb} \cup BB_{gt}}$$

7.2 Detection Result

7.2.1 Detection Result

For the detection result, see Figure 13 and APPENDIX A.

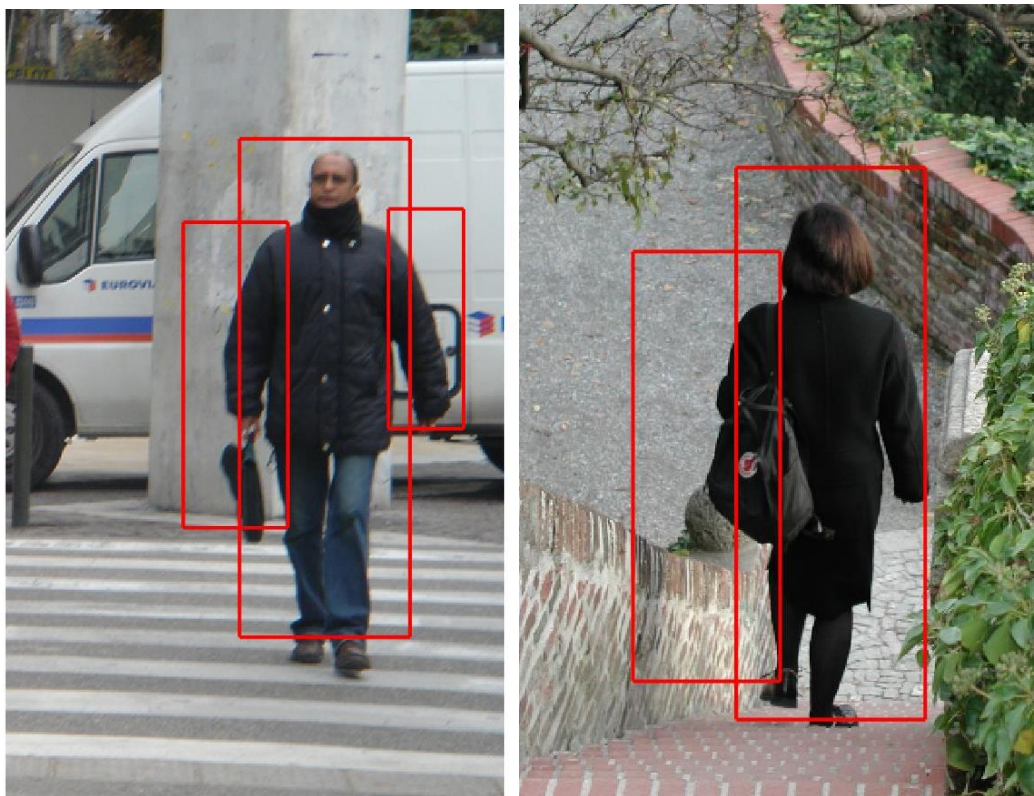


Figure 13 Some Detection Results.

7.2.2 Comparing Result

Since pedestrian detection is an active research area, there are many other detection algorithms existing. Even Mathworks' Matlab, one of the most popular and common tools that has been using for computer vision, added a people detection function into its newest version, Matlab 2015b, as part of the *vision* package, which is called *PeopleDetector*. The *vision.PeopleDetector* function was implemented using Histogram of Oriented Gradient features and a trained Support Vector Machine classifier [48]. The method was originally introduced in Dalal and Triggs' paper, [7]. For convenience, the method is named Dalal and Triggs' method in this thesis.

We applied the algorithm in this study and the Dalal and Triggs' algorithm to 342 testing images. In total, 287 pedestrians were involved in the testing image. Our algorithm successfully detected 127 pedestrians, and Dalal and Triggs' only detected 86 pedestrians. Since these two detection algorithms used different detection methods, the detection results do not have much overlapping with each other. There are only 41 pedestrians detected by both of the two detectors. (See Figure 14 and APPENDIX B for comparison results.)

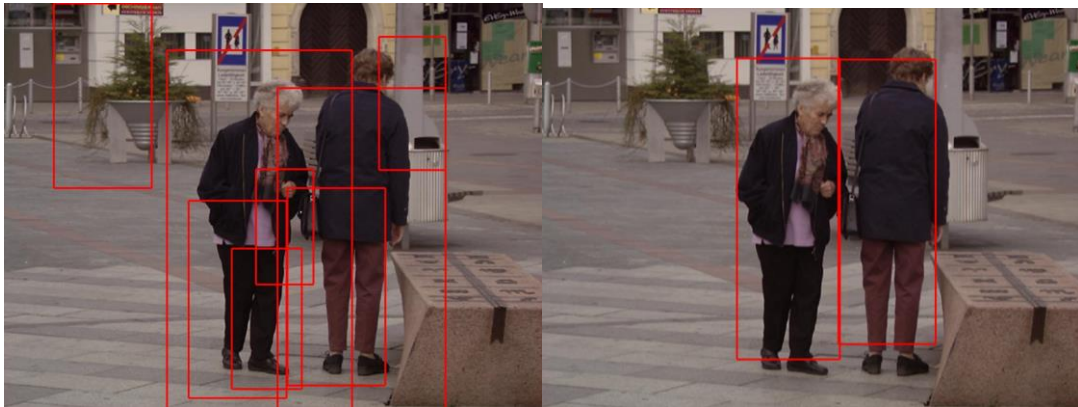


Figure 14 Comparison Result.

The left image was detected by Dalal and Triggs' algorithm, and the right one was detected by our algorithm.

Another thing we noticed is that the Dalal and Triggs' algorithm would generate much more false-positive detections. Dalal and Triggs' totally made 367 false-positive detections, and our algorithm only made 215 false-positive detections. (Also see Table 1.)

Table 1. Comparison Result with Dalal and Triggs' Algorithm .

	Our Algorithm	Dalal and Triggs'
Total Pedestrians ⁵	287	287
Successfully Detection	127	86
False-positive Detection	217	367

7.3 Failure Analysis

After analyzing the failure cases, we discovered that failures were usually caused by one or more of four factors: i) pedestrians are not always symmetric; ii) output selection algorithm does not select the best hypothesis box; iii) occlusion and/or crowd appears in the image; and iv) image with low contrast.

7.3.1 Failure Cause by Asymmetric

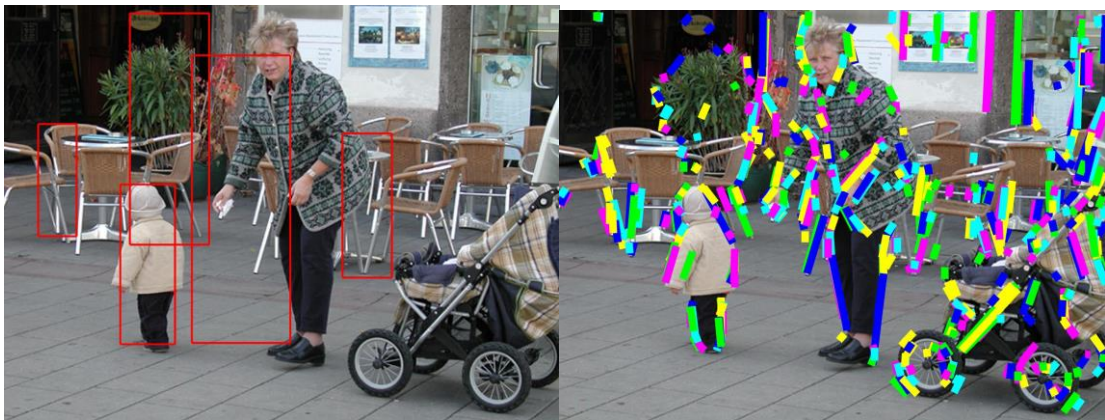


Figure 15 Failure caused by violation of symmetry.

⁵ The number of total pedestrians in testing image.

Current detection models are taking advantage of symmetry feature of pedestrian. Generally, a pedestrian in an erect posture is symmetric. However, not all the people will walk with an erect posture, for instance, the lady in Figure 15.

The lady stoops to the kid in front of her. In the front or back view, even a pedestrian bending down his or her body, the pedestrian is still generally symmetric. Nevertheless, it may not be true in other angles of view. From the 1-piece line image (the second picture in Figure 15), the lady is in the 45-degree angle of camera, if without her right arm, the symmetry feature still exists, meaning that our current detection models should still cover this scenario. However, in this specific image, her right arm violates the symmetric feature of pedestrian, which caused the detection failure.

Another example is Figure 16. As we can see, the gentleman in Figure 16 is violating the symmetric feature as well. The first picture in Figure 16 was the original

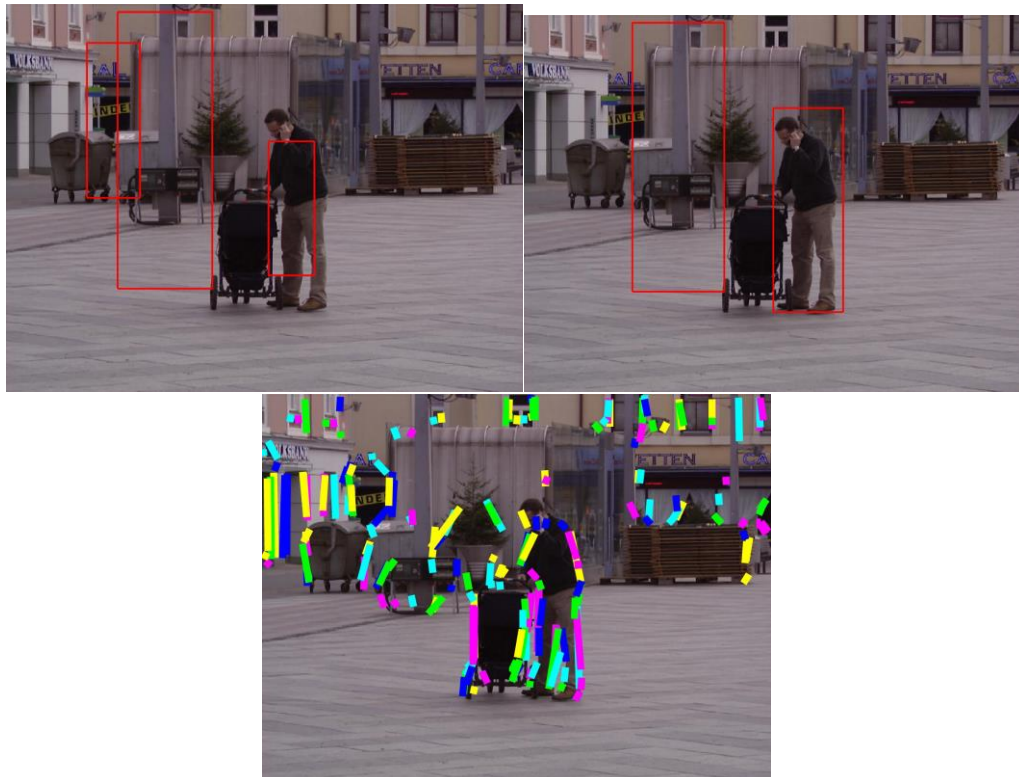


Figure 16 Failure caused by violation of symmetric.

detection result. Even though the detection result is not categorized as failure according to our evaluation measurement, we still can improve the detection result by not using the symmetric criteria in the step of outputting hypothesis boxes. The detection result can be observed as the second picture in Figure 16.

By introducing more detection models, we might be able to solve this problem. Even though more detection models might lead us into overfitting problem, it is still worth to try in the future. If we can overcome this failure, it would dramatically increase the accuracy.

7.3.2 Failure Cause by Output Selection Algorithm

In many testing cases, our detection algorithm is able to generate the hypothesis box that covers the pedestrian pretty well, however, the output algorithm might choose

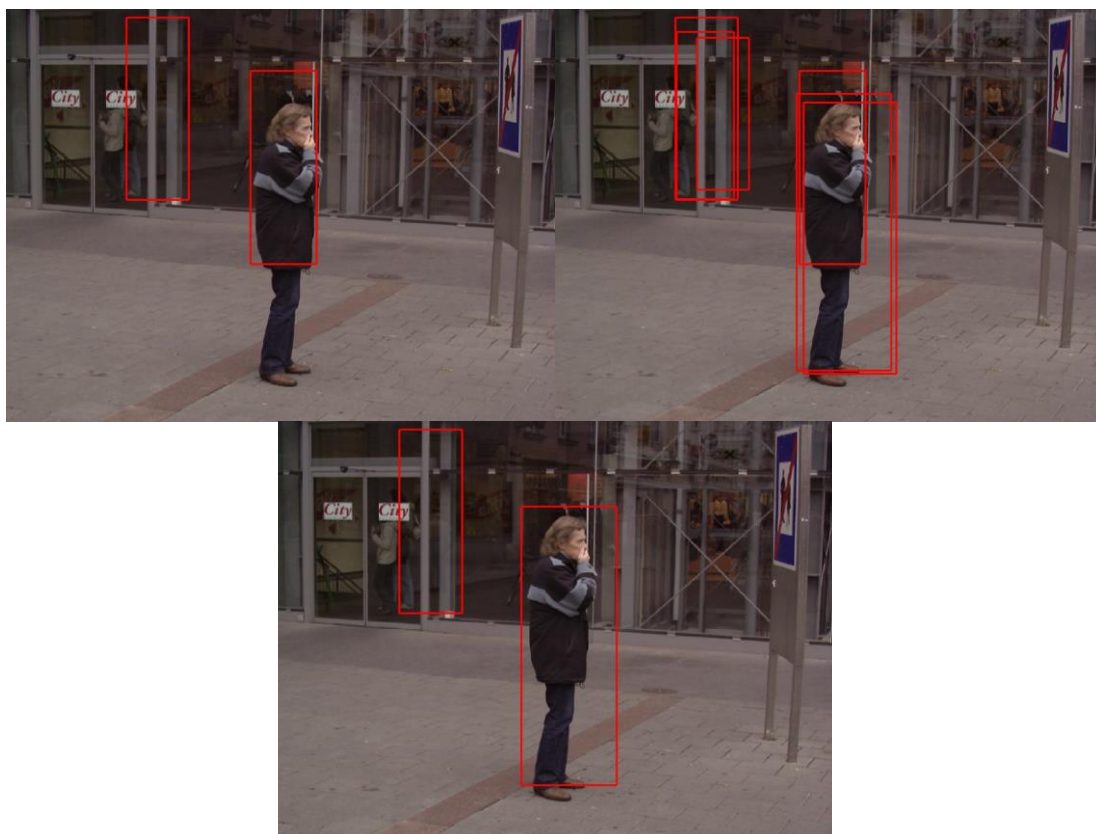


Figure 17 Failure caused by output selection algorithm.

some one else other than the best one, for instance Figure 17.

The first image of Figure 17 is the outputted detection result, and the third image shows all the hypothesis boxes generated during the detection period, after applying all the constraints. We can easily find out there are some hypothesis boxes covering the pedestrian better than the one returned. The second image used median of clustering of hypothesis boxes as the return box for that clustering. We can see the detection is improved significantly. Therefore, in the future, if we can develop a better output selection algorithm, the detection result should be improved.

7.3.3 Failure Cause by Occlusion and/or Crowd

In our detection result, failure often appears among crowd or occlusion, such as people in the background of Figure 18. Both of crowd and occlusion will affect the accuracy of 1-piece polyline extraction, which is the feature needed for detection. It is possible to solve the problem, if we add some more detection models. For instance, we can add a model that the hypothesis box has fixed width and various heights. The width

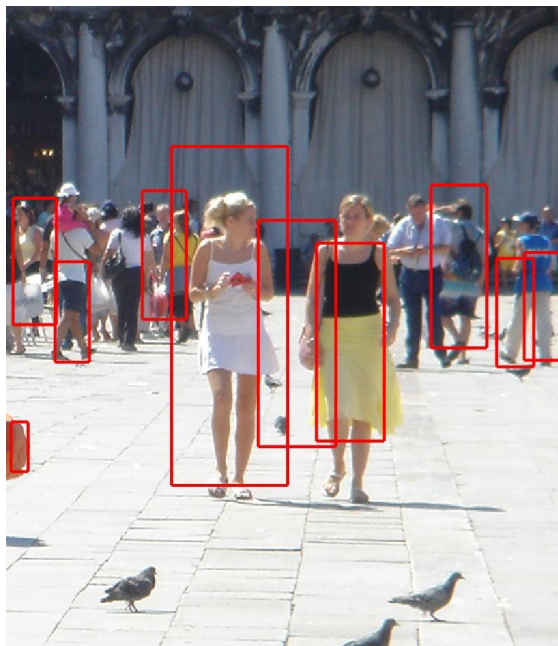


Figure 18 Failure caused by occlusion or crowd.

could equal to the wide of image or certain percentage of the image. Then we might be able to use the density of vertical lines as the criteria to make the detection decision.

7.3.4 Low Contrast

Low contrast of images is another reason for failure detection. In low contrast image, it is very difficult to extract 1-piece polylines accurately since current edge detectors generally are not working well with low contrast image.

7.4 Future Study

Even though there is much work to do before this simple line based pedestrian detection framework could be implemented in the industry, the result is still very promising. There are many possible ways we can improve the algorithm. Followings are some suggestions for future study.

7.4.1 More Detection Models

Currently, we only have two detection models, including a model for front view of a pedestrian and a model for profile view of a pedestrian. In the future, more detection models should be added to the system, such as models for two or three people walking together, models for pedestrian with stroller, models for adult walking with kids, so on and so forth.

7.4.2 A Voting System/Scoring Hypothesis Box

In the future, a voting system to this framework to help on selection of hypothesis boxes should also be added. During the detection step, several constraints would give a low confidence score to some hypothesis boxes that are not very confident in detection. Currently, we have not used the scoring system. We hope in the future

work, we can introduce a voting system as an addition to current framework. The voting system will be helpful to eliminate some low scoring hypothesis boxes, which will not only help with increasing the detection accurate, but also will improve the time efficiency of the algorithm.

7.4.3 Machine Learning and Classifier

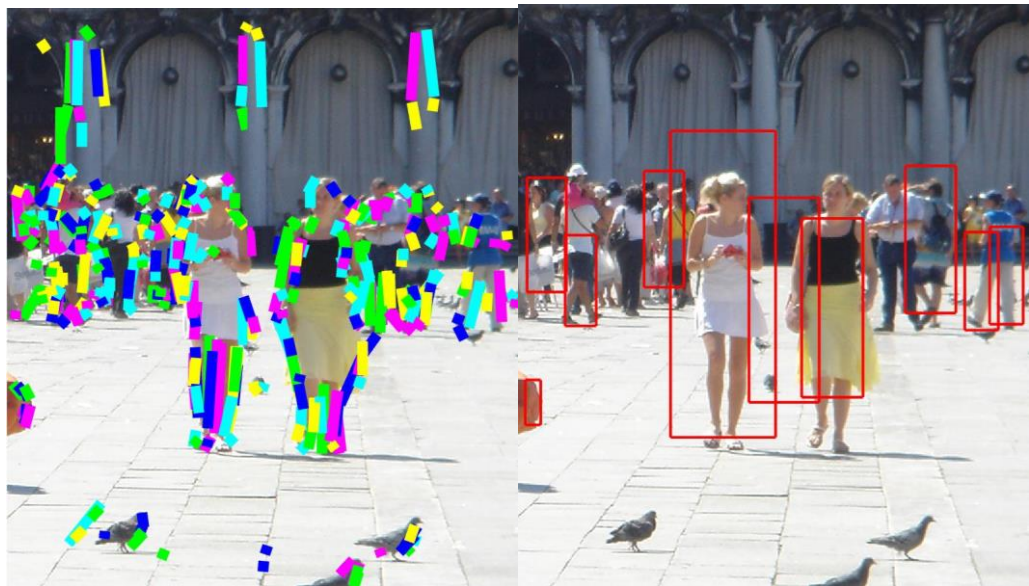
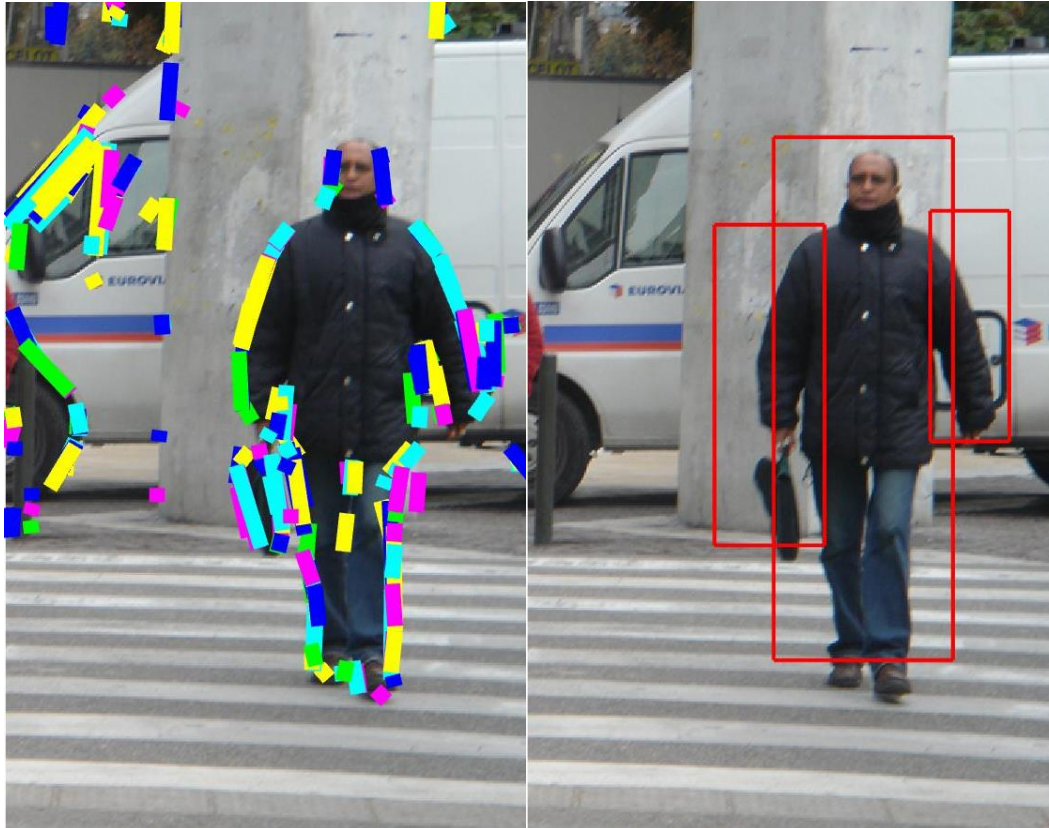
For this basic line segment based pedestrian framework, the geometric feature of pedestrian is solely applied. There is no training and learning step involved. We have not used neither classifiers nor machine learning strategy. In the future, we are looking forward to including some machine learning strategy into this proposed geometric framework, possibly neural network. We are confident if we do so, it will dramatically increase the detection accuracy.

Chapter 8. Conclusion

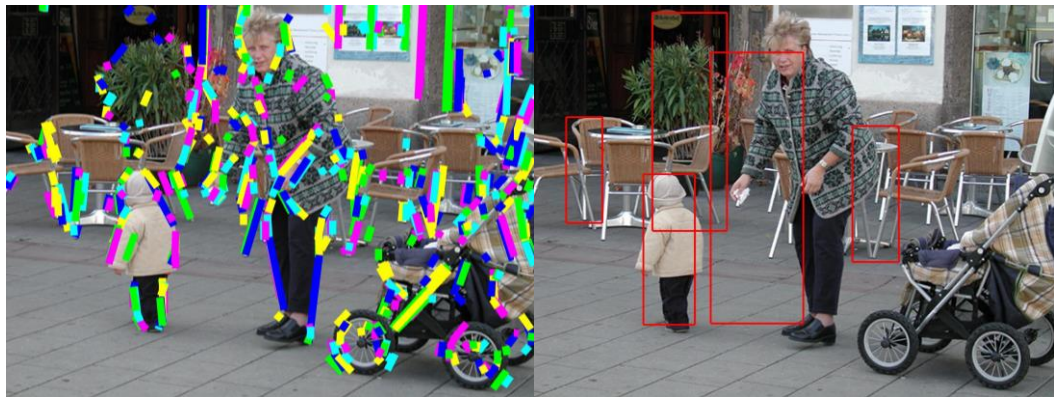
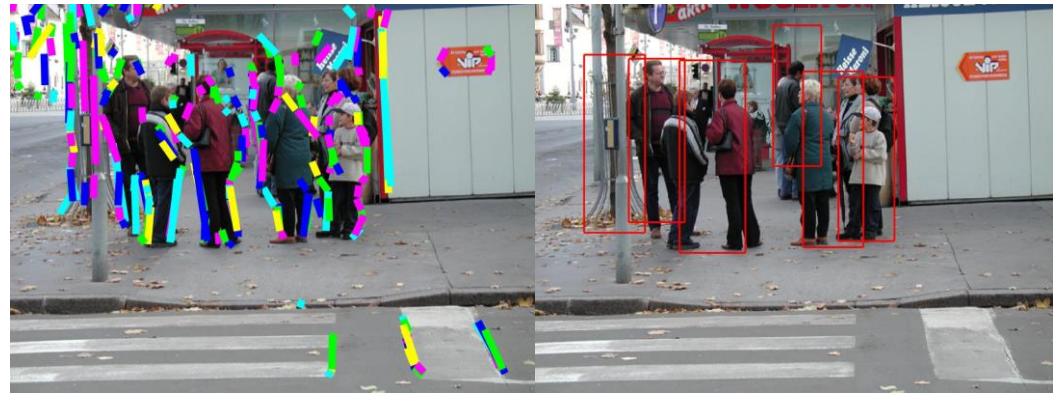
The pedestrian detection framework proposed in this thesis is using basic line segments and the geometric relationships between line segments as the cue to detect pedestrians. It is a fairly new attempt on the topic of pedestrian detection. There is still a lot of room for improvement.

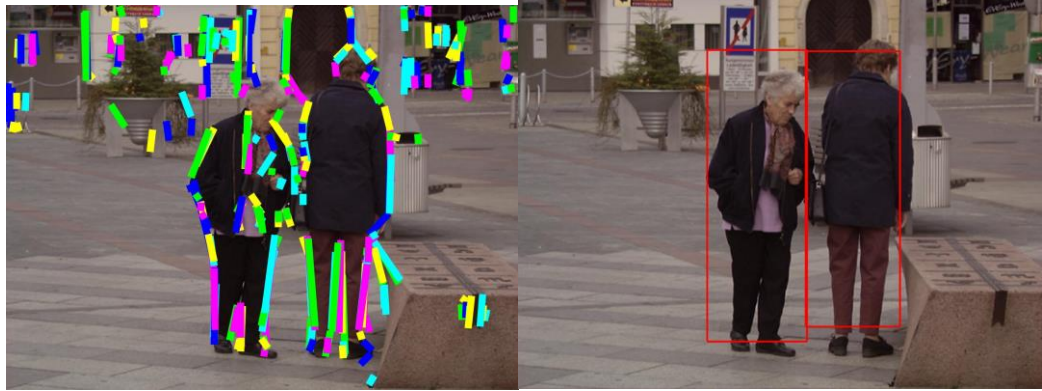
The result of current work is promising. Compared with Dalal and Triggs' pedestrian detection algorithm, the detection result improved by 48%, and the false-positive detection reduced by 41%. In the future work, we hope to include more detection models introducing a hypothesis boxes voting system, adding machine-learning feature. We believe in such way, the detection result will become even better.

APPENDIX A: MORE DETECTION RESULTS WITH EXTRACTED
LINES



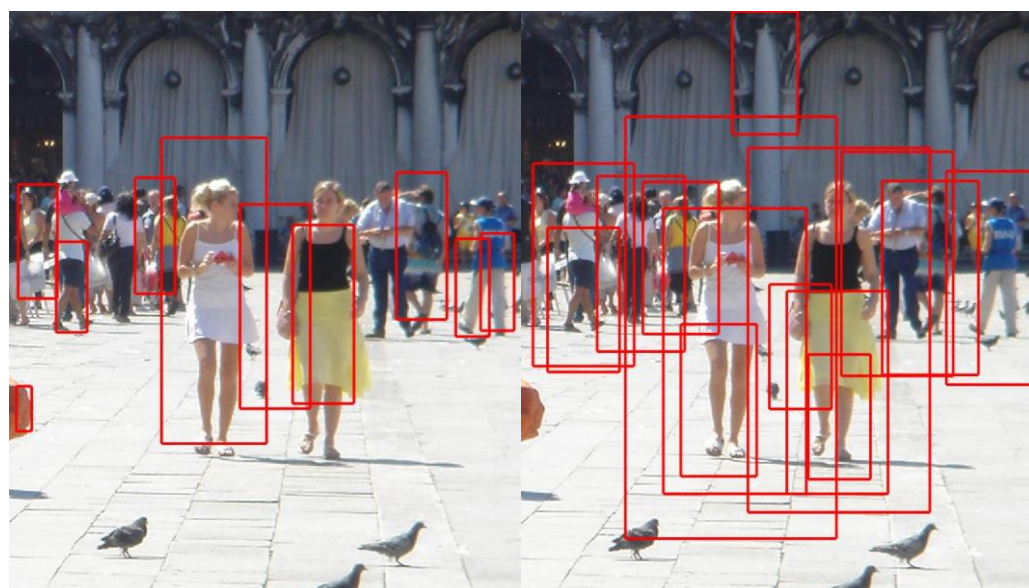
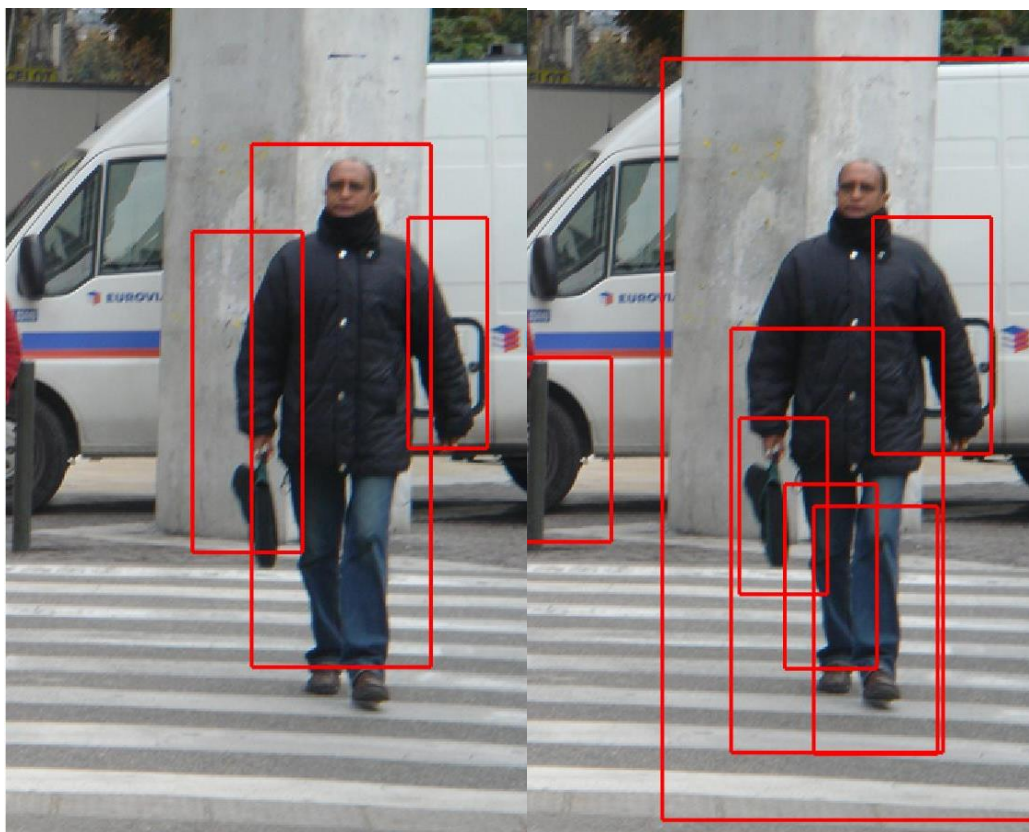


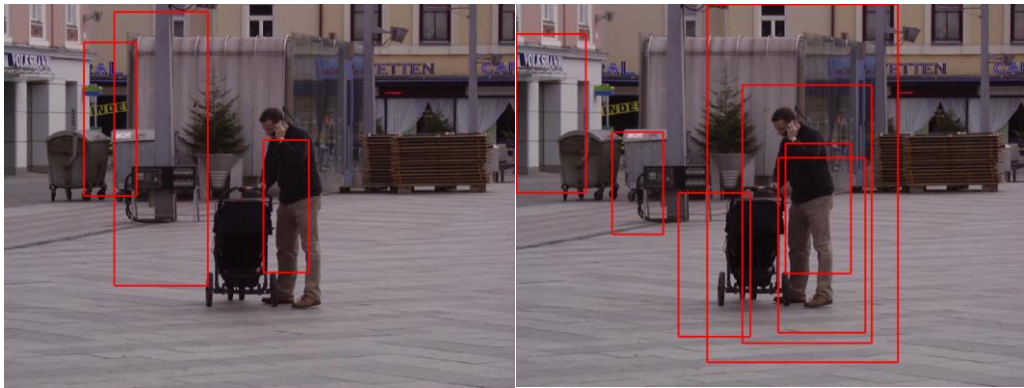
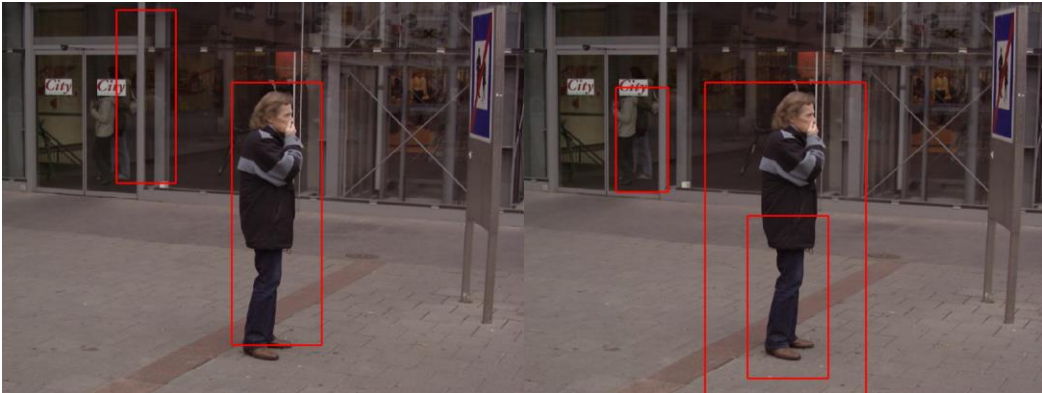
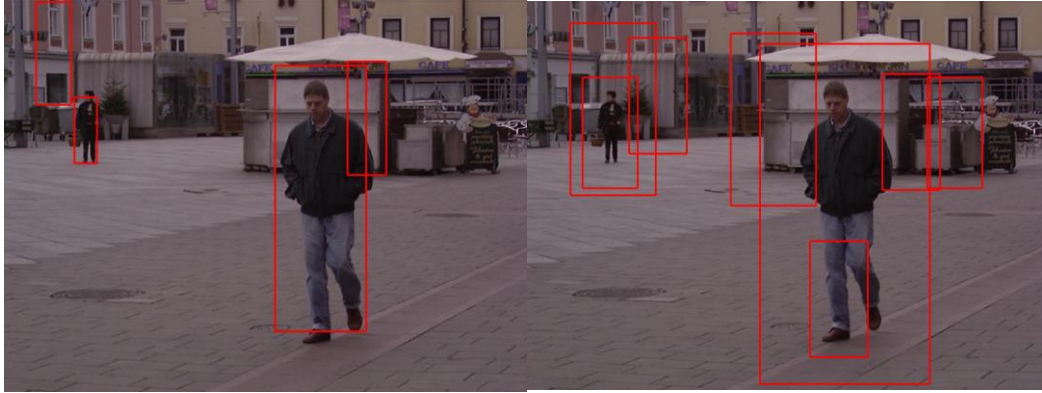


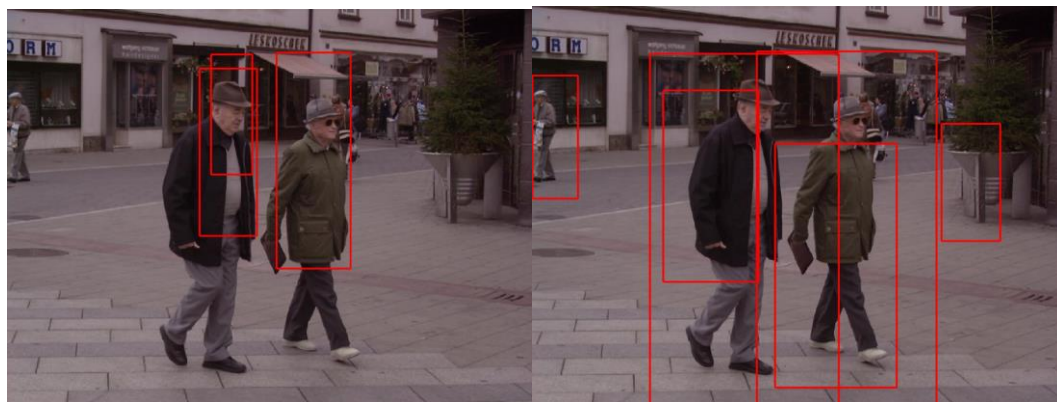
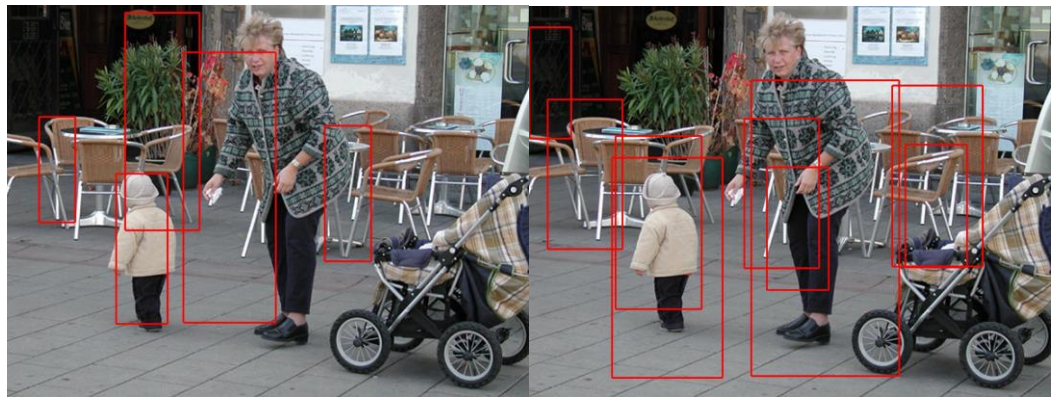
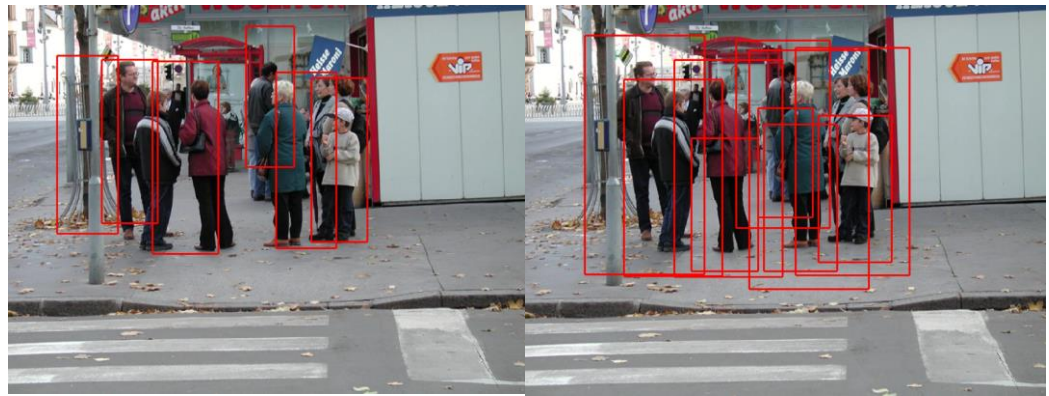
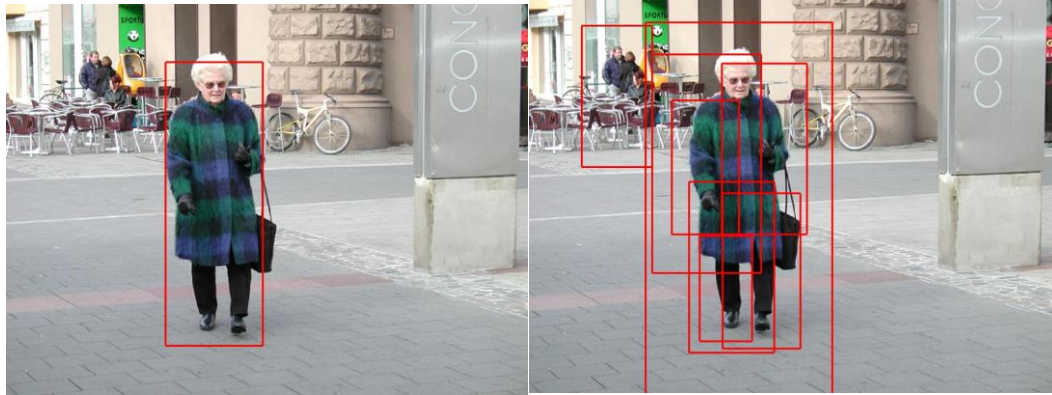


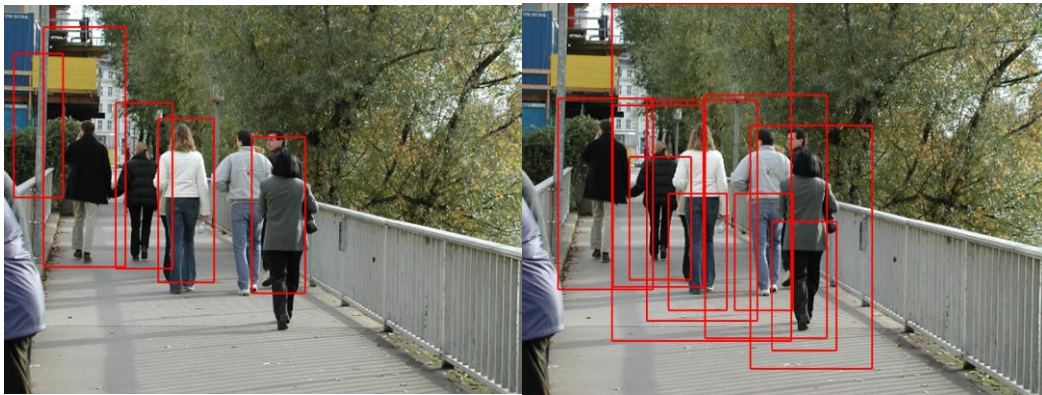
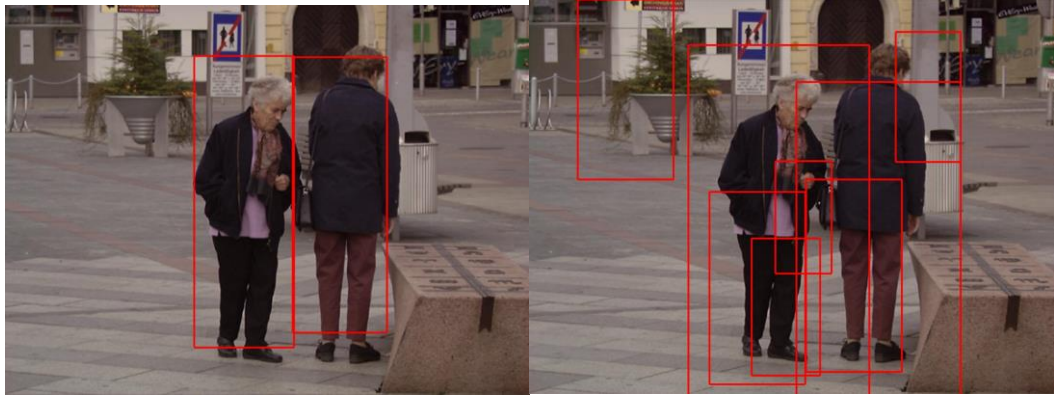
APPENDIX B: COMPARISON RESULT WITH DALAL AND TRIGGS' ALGORITHM

(Right column is the detection result of Dalal and Triggs' algorithm)









Literature Cited

1. Belongie, S., Malik, J., Puzicha, J. Shape matching and object recognition using shape contexts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24,4 (2002), 878-885.
2. Bertozzi, M., Broggi, A., Chapuis, R., Chausse, F., and Fascioli, A. Shape-based pedestrian detection and localization. In *Intelligent Transportation System Proceeding* 1 (2003), 328-333.
3. Bertozzi, M., Binelli, E., Broggi, A., and Rose M.D. Stereo vision-based approaches for pedestrian detection. In *Computer Vision and Pattern Recognition Workshop in Computer Vision and Pattern Recognition* (2005), 16.
4. Bogin, B., and Varela-Silva, M. Leg length, body proportion, and health: a review with a note on beauty. In *International Journal of Environmental Research and Public Health* 7, 3 (2010), 1047-1075.
5. Curio, C., Edelbrunner, J., Kalinke, T., Tzomakas, C., and Seelen, W. Walking pedestrian recognition. In *Processing of IEEE Internal. Conference on Intelligent Transoprtion System'99*. (1999), 292-297.
6. Da Vinci, Leonardo. *Vitruvian Man*. (1490).
7. Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1* (2005), 886-893.
8. Dalal, N., Triggs, B., and Schmid, C. Human detection using oriented histograms of flow and appearance. In *9th European Conference on Computer Vision Proceedings 2* (2006), 428-441.
9. Dollar, P., Wojek, C., Schiele, B., and Perona, P. Pedestrian detection: an evaluation of state of the art. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 4 (2012), 743-761.
10. Enzweiler, M., and Gavrilu, D.M., Monocular pedestrian detection: survey and experiments. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31, 12 (2009), 2179-2195.
11. Everingham, M., Van Gool, L., Williams, C.K.I., Winn J., and Zisserman. The pascal visual object classes (voc) challenge. In *International Journal of Computer Vision* 88, 2 (2010), 303-338.

12. Felzenszwalb, P., and Huttenlocher, D. Efficient matching of pictorial structures. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2 (2000), 66-72.
13. Forsyth, D., and Fleck, M., Body plans. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (1997), 17-19.
14. Gavrila, D. Pedestrian detection from a moving vehicle. In Proceedings of 6th European Conference on Computer Vision 2 (2000), 37-49.
15. Gavrila, D. M., and Geibel, J. Shape-based pedestrian detection and tracking. In Proceedings of. IEEE Intelligent Vehicles Symposium (2002), 8-14.
16. Gavrila, D.M., and Munder, S. Multi-cue pedestrian detection and tracking from a moving vehicle. In International Journal of Computer Vision, 73, 1 (2007), 41–59.
17. Geismann, P., and Schneider, G. A two-staged approach to vision-based pedestrian recognition using Haar and HOG features. In IEEE Intelligent Vehicles Symposium (2008), 554-559.
18. Geronimo, D., Lopez, A., Ponsa, D., and Sappa, A. Haar wavelets and edge orientation histograms for on-board pedestrian detection. In Proceedings of the 3rd Iberian conference on Pattern Recognition and Image Analysis 1 (2007), 418-428
19. Geronimo, D., Lopez, A.M, Sappa, A.D., and Graf, T. Survey on pedestrian detection for advanced driver assistance system. In IEEE Transactions. Pattern Analysis and Machine Intelligence, 32, 2 (2010), 1239-1258.
20. Hirschmüller, H. Stereo processing by semi-global matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30, 2 (2008), 328–341.
21. Katz, I., and Aghajan, H. Multiple camera-based chamfer matching for pedestrian detection. In International Conference on Distributed Smart Cameras (2008), 1-5.
22. Kovesi, P. Shapelets correlated with surface normal produce surfaces. In. 10th International Conference on. Computer Vision 2 (2005), 994-1001.
23. Leibe, B., Seemann, E., and Schiele, B. Pedestrian detection in crowded scenes. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2005), 878-885.
24. Li, Q., Liang, G., Gong, and Y. A geometric framework for stop sign detection. In Signal and Information Processing (ChinaSIP), IEEE China Summit and International Conference on Signal and Information Processing (2015), 258-262.

25. Ma, T., Yang, X., and Latecki, L. J. Boosting chamfer matching by learning chamfer distance normalization. In Proceedings of 13th European Conference on Computer Vision (2010), 450-463.
26. Mallat, S. A theory for multiresolution signal decomposition: the wavelet representation. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 11, 7 (1989), 674-693.
27. Mikolajczyk, K., Schmid, C., and Zisserman, A. Human detection based on a probabilistic assembly of robust part detectors. In Proceedings of 8th European Conference on Computer Vision (2004), 69-81.
28. Mohan, A., Papageorgiou, C., and Poggio, T. Example-based object detection in images by components. In IEEE Transactions on Pattern Analysis and Machine Intelligence (2001), 349-361.
29. Mori, G., and Malik, J. Estimating human body configurations using shape context matching. In Proceedings of 7th European Conference on Computer Vision 3 (2002), 666-680.
30. Mori, G., Ren, X., Efros, A. and Malik, J. Recovering human body configurations: combining segmentation and recognition. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2004), 326-333.
31. Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., and Poggio, T. Pedestrian detection using wavelet templates. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1997), 193-199.
32. Papageorgiou, C., and Poggio, T. A trainable system for object detection. In International Journal of Computer Vision 38, 1 (2000), 15-33.
33. Refegier, A. Shapelets: i. a method for image analysis. In Monthly Notices of the Royal Astronomical Society, 338, 1 (2003), 35-47.
34. Ronfard, R., Schmid, C., and Triggs, B., Learning to parse pictures of people. In 7th European Conference on Computer Vision Proceedings 4 (2002), 700-714.
35. Sabzmeydani, P., Mori, G. Detecting pedestrians by learning shapelet features. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007), 1-8.
36. Schreiber, D., Belezni, C., and Rauter, M, GPU-accelerated human detection using fast directional chamfer matching. In 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (2013), 614-621.
37. Seemann, E., Leibe, B., and Schiele, B. Multi-aspect detection of articulated objects. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2006), 1582-1588.

38. Shahua, A., Gdalyahu, Y., and Hayon, G., Pedestrian detection for driving assistance systems: single-frame classification and system level performance. In IEEE Intelligent Vehicles Symposium, Parma, Italy (2004), 1-6.
39. Suard, F., Rakotomamonjy, A., Bensrhair, A., and Broggi, A. Pedestrian detection using infrared images and histograms of oriented gradients. In IEEE Intelligent Vehicles Symposium (2006), 206-212.
40. Thayananthan, A., Stenger, B., Torr, P.H.S., and Cipolla, R. Shape context and chamfer matching in cluttered scenes. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2003), 734-741.
41. Tuzel, O., Porikli, F., and Meer, P. Human detection via classification on Riemannian manifolds. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2007), 1-8.
42. Viola, P., Jones, M.J., and D. Snow. Detecting pedestrians using patterns of motion and appearance. In Proceedings of 9th IEEE International Conference on Computer Vision (2003), 734-741.
43. Xing, W., Zhao, Y., Cheng, R., Xu, J., Lv, S., and Wang, X. Fast pedestrian detection based on haar pre-detection. In International Journal of Computer and Communication Engineering 1, 3 (2012), 207-209.
44. Walk, S., Majer, N., Schindler, K., and Schiele, B. New features and insights for pedestrian detection. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010), 1030-1037.
45. Zhang, S., Bauckhage, C., and Cremers, A. B. Informed Haar-like features improve pedestrian detection. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2014), 947-954.
46. Zhao, L., and Thorpe, C. Stereo- and neural network-based pedestrian detection. In IEEE Transaction on Intelligent Transportation System 1, 3 (2000), 148-154.
47. Zhu, Q., Yeh, M., Cheng, K., and Avidan, S. Fast human detection using a cascade of histograms of oriented gradients. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2006), 1491-1498.
48. MathWork, vision.PeopleDetector Sysytem object.
<http://www.mathworks.com/help/vision/ref/vision.peopledetector-class.html>.
49. INRIA person dataset. <http://pascal.inrialpes.fr/data/human/>.
50. Traffic safety fact, 2013 data. Department of Transportation (2015).