Waseda University Doctoral Dissertation

Research on Robust Local Feature Extraction Method for Human Detection

TANG, Shaopeng

Graduate School of Information, Production and Systems

Waseda University

Feb. 2011

Abstract

Human detection is an essential branch of the computer vision, concerned with the analysis of images or videos to extract useful information. It can be used in many applications ranging from intelligent vehicle to surveillance system. It also plays an important role in robotics and user interfaces. The target of this dissertation is to develop robust feature extraction algorithms that encode image regions as high dimensional feature vectors that support high accuracy human/non-human decisions. In this dissertation, several human detectors are proposed to improve the detection rate; some acceleration methods are also discussed to reduce the detection time for practical applications.

In chapter 1, the background knowledge of object detection and human detection is introduced. An overview of this dissertation is also presented in this chapter.

In chapter 2, histogram of template (HOT) feature is proposed. The predefined templates and formulas are used to encode the image regions, to provide the cue for classification. The templates and formulas are designed to extract the characteristic information of the human body. Not only the gradient information is used, like the traditional human detector, but also the texture information is considered as well, which shows high discriminative abilities in human detection. Besides, because the basic unit for feature extraction is template, which is a three pixels' combination, the HOT feature can reflect the relationship between three pixels, instead of focusing on only one. Experiments are performed on INRIA dataset, which shows the proposed

HOT feature is more discriminative than histogram of orientated gradient (HOG) feature, under the same training method. The detection error tradeoff (DET) curve is used to evaluate the performance of features. At 10^{-4} false positive per window (FPPW), HOT feature reduces the miss rate by nearly 3%, compared to HOG feature, when kernel support vector machine (SVM) is used. Compared to covariance matrix (COV) feature, it improves the performance by 1%. Take into account that COV feature uses variable sub window, which can improve the performance efficiently, and the computation complexity, we could say that HOT feature outperforms HOG and COV feature. We also evaluate the three features by using a random ensemble strategy, and experiment shows that HOT feature also has advantage in discriminative ability. By experiment analysis, we can see that by using more templates or formulas, we could further improve the detection rate. HOT feature also has property of illumination-invariant. It means that the normalization strategy is not necessary for this feature and only integer calculation can support the feature extraction, which may be useful for hardware acceleration.

In chapter 3, we extend the original HOT feature in two directions: an appearance based feature and a motion based feature are proposed respectively. A multi-scale block histogram of template (MB-HOT) feature is used to detect human by appearance. HOT feature can get higher detection rate partially because that the feature is extracted from the middle level (template level), not from the pixel level directly. The MB-HOT feature further improves the performance by extending the middle level. For the definition of template, the three pixels' combination is replaced by three blocks' combination. So the feature can be extracted from more macrostructures levels. We compare this feature with original HOT feature, HOG,

multi-scale HOG, and COV feature. Experiments on INRIA dataset show that this feature outperforms HOT, HOG, and multi-scale HOG. At 10^{-4} FPPW, the result is improved by 5%, compared to HOT feature when linear SVM is used. The performance is nearly the same but a little better than COV. Considering the COV feature uses the different training method and the variable sub window strategy, and it has high computation complexity, we can say that our feature is better than COV feature. A motion based feature is also proposed to capture the relative motion of human body. This feature is calculated in optical flow domain. The experiment is done on CAS dataset. The result of our feature is better than optical flow feature, Karhunen-Loeve transform (KLT) feature and intra motion histogram central difference feature. We also do the experiment on video sequences to show that the combination of the detection responds obtained by MB-HOT and motion based feature can reduce the false detections for human detection from videos.

In chapter 4, we give the graphics process unit (GPU) based implementation for MB-HOT feature, to accelerate the feature extraction. In the feature extraction, there exist lots of parallel calculations. We optimize the workflow of the calculation, to make it suitable for GPU based implementation. Our experiment is done on GTX 285 and CUDA framework. For a natural image (640×480), it takes about 48.924 milliseconds for feature extraction and classification, which means that this method can meet the real time requirement.

In chapter 5, we propose a pose invariant human detector. Before the detection stage, we first estimate the poses of humans, which are represented by a set of two dimensional points, and the features are extracted from the bounding boxes of the different parts. In this way, the characteristic feature for each part can be obtained and the redundant information of the background can be removed. The pose estimation is defined as a multi-output regression problem. A new definition of loss function is proposed to find the mapping function, by which the images can be mapped into the pose space. We compare our method with other pose-invariant feature, and it can get higher detection rate at 10^{-4} FPPW.

In chapter 6, we conclude the contributions of this dissertation.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Image Understanding and Object Detection

Image understanding is the branch of the computer science, the purpose of which is to analysis of images to extract useful information from the world. It covers wide range of image processing fields and plays an important role in computer vision.

Human learn information from the natural. Because of the limitation of the human ability, lots of research focuses on the machine learning to help human to get more information. Human get the information from many fields, but the most are from the human vision system. So people want to develop a machine vision system to replace the human vision system.

The purpose of image understanding is to give the explanations of provided images and extract the valuable information. Images are provided by different observation system, and reflect the objective reality of this world， from which we could get the valuable information. So the image understanding is a very important work. But because of the lag of image understanding, there is not efficient solution for such kind of the work. In this application at first, after we obtain the images and process them, the information will be extracted by human. With development of technology, we could get huge data of images. So we want to use computer to complete this job automatically. Many researchers have been working long time in order to design the computer vision system that can not only perceive the world by vision information, but also emulated the capability of human in visual perception.

Compared with image process such as image enhancement, image de-noising and so on, image understanding is a high level vision task since it involves the knowledge on the semantics of images. Low-level image processing deals with the original image data like intensity or color values, while high-level computer vision performs on the abstract data such as object location, object size, object shape and mutual relations among objects. Based on the information contained from the images, high level computer vision system can decide to take certain action.

Object detection is an essential part of image understanding. There have been wide applications of object extraction in real world, such as face detection, human detection, motion estimation, robot vision system and so on.

Generally speaking, research for object detection is focusing on representation, learning and recognition. Representation means how to represent an object. Learning method is used to learn common property of class of objects. Recognition identifies the object in an image using models using models learned from learning stage. So an object detector can be divided into two parts. First is an image feature set and another is a detection algorithm. Features include sparse or dense representation of image regions as feature vector; and the detection method specifies how to use these features to get the classification decisions. Feature extraction captures the texture information, gradient information or contour information of the object, and extracts the discriminative features to represent this kind of object. There are two views on how to computer feature vector. One approach is based on sparse features extracted from a set of salient image regions. This method is based on that not all images regions contain useful information. Some regions may be more important than some cluttered regions. The alternative approach is to calculate the feature densely. The point is that at the early stages of visual scene analysis, all images regions may be of equal importance, and it is best that we don't lost any information. In the detection stage, there are also two groups of methods: generative approach and discriminative approach. Typically, Bayesian graphical models are used in generative method to characterize these parts and to model their co-occurrences. Discriminative uses machine learning method to classify each feature vector, to judge that it belongs to the object or not.

1.2 Human Detection

Human detection is becoming a popular research topic recently, because it can be used in many practical applications.

Consider the image content analysis system, for example, statistics shows that the digital camera owner can take as many as about ten thousands photos in just two or three years, even he doesn't use it frequently. It is very tedious to search and manage these pictures manually. It is very useful that if we could develop some image manage software to automatically add tags to these images. This could help us to search what we want quickly. Most images contain human, so human detection will become an important for such kind of tools. For films and videos, the human detection will form an integral part of applications for video on demand and automatic content management. In conjunction with face and activity recognition, this may facilitate the search some designated contents or search for relevant sub sequences. From Figure 1-1, we can see some photos selected from the album. They are all personal digital images. In this thesis, we will mainly discuss the human detection problem from the images like this. We extract the features or descriptors for either the whole body or various sub-parts, and construct the detector based on this.

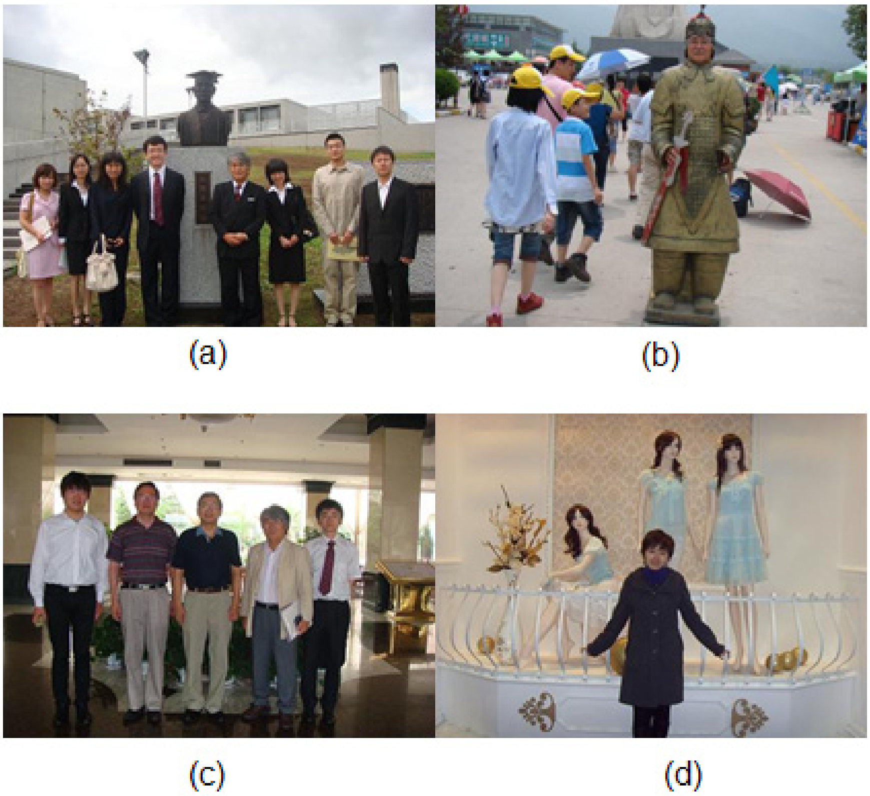


Figure 1-1 Some photos selected from the album

Beside, the human detection can be used in the intelligent vehicle system. Pedestrian accident is one of the largest sources of traffic-related injuries. Robust human detector can reduce such kind of accident efficient. Majority of the accidents occur either at the pedestrian crossings or while reversing the car. Because of the speed, drivers has no time to take some actions to deal with the human suddenly appear in front of the cars; or the rear mirrors don’t provide a full view of the scene behind the car and this situation becomes worse for kids and babies, owing to their smaller size. In either case, it's the inability of the driver to locate the pedestrians that cause the trouble. Real time pedestrian detection system can deal with this situation and avert a possible collision efficiently. From Figure 1-2 (a), we can see that the rear mirror doesn't provide the full view and the child is in danger in this case. Figure 1-2 (b) is a case for automatic driver system. It can detect the human in front of the car, and take some actions to prevent the accident. Human detection is key part for such kind of applications.

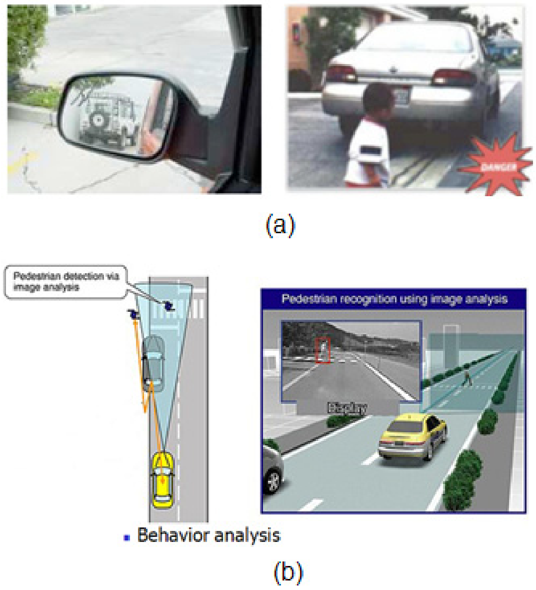

DANGER
(a)
Pedestrian detection via image analysis
Pedestrian recognition using image analysis
Display
▪ Behavior analysis
(b)

Figure 1-2 (a) possible accident assumption; (b) automatic driver system

Another application of human detection is that it can be used in the surveillance system. Surveillance system is the monitoring of the behavior, activities, or other information, usually of the people. It can be used in supermarket, security system and traffic management system, to give a warning or record the inappropriate behavior. It can also provide some information for analysis, such as the customers' number counting in supermarket. Just like the multimedia analysis, there is huge data we should deal with in the surveillance system. It will take a long time to search the content that we want. In some cases, the surveillance system needs the help of human. The human will watch the videos captured by the system and find the inappropriate behaviors. Because of the limitation of human understanding, it is easy for human to make some mistakes, such as lose some key information. Automatic surveillance system should be developed for such kind of applications. It is generally believed that a successful surveillance system will contain a figure-ground segmentation to detect humans in images, tracking to maintain temporal coherence, and finally a recognition part to recognize identity, actions and so on. A robust human detection method is the foundation for this kind of system. In Figure 1-3, we can see some surveillance system. There are many views, and human understanding is easy to miss some important information.

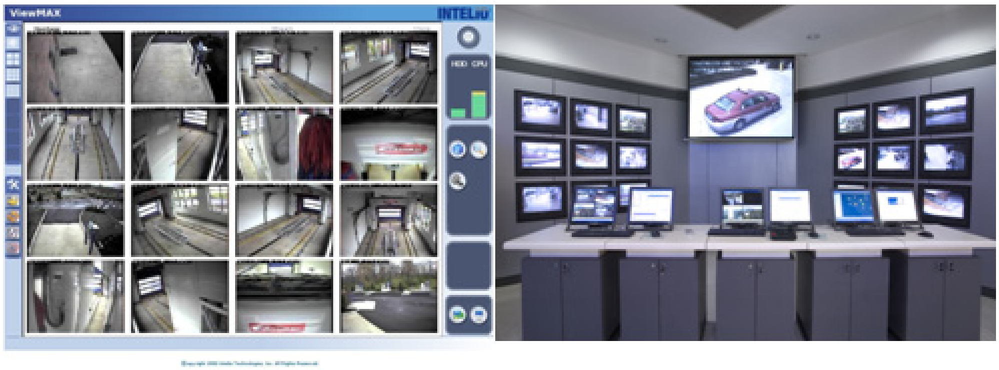


Figure 1-3 Typical Surveillance System

Human detection has many difficulties. Recent years, although many research focus on this field, the detection rate is still far away from the practical application. So unlike the face detection which has already been used in commercial product, the human detection still needs improvement. The first difficulty is that the appearance of human is different. People wear different clothes. So some powerful features used in the face detection, such as the skin color, can't not be used the human detection. People have different poses; the shape of human body is not fixed. The sizes of the human body in the images are different, because of the distance to camera is different. So we should consider the scale problem. These appearance differences can be seen in Figure 1-4.

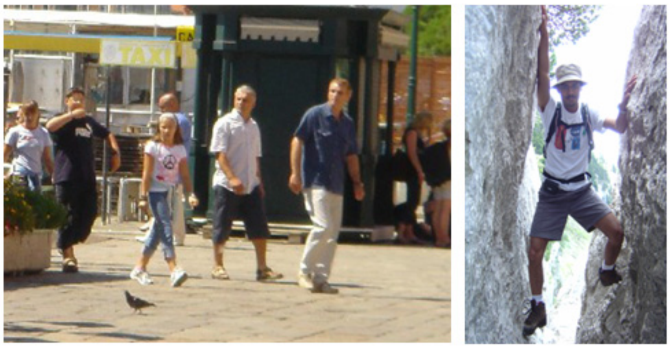


Figure 1-4 Appearance difference

Secondly, the backgrounds are clutter and vary from image to image. For example, the images that are taken from the outdoor scene and the images that are taken from the indoor scene are different. The detector must be capable of distinguishing human from the complex background region. In the detection stage, there exists some false detection in the background. If we change the configuration of the detector, we could reduce the miss detection; but at the same time, the false detection increases.

Thirdly, the illumination condition is different. Direct sunlight or dim lighting at night has influence on result of human detection. Although models of color and illumination invariance have made significant advances, they still are far from being effective solutions when compared to human and mammalian visual system which are extremely well adapted to such changes.

1.3 Summaries of Contributions

The dissertation concentrates on how to detect the human from the images and videos. Robust features for human detection are discussed to improve the detection rate.

The histogram of template feature is proposed. We define some templates and formulas to extract the contour information and the texture information of human body. They are designed to obtain the shape of local part of the human body, and they can reflect the properties of local edge of the human body shape. We could calculate the histogram of pixels that satisfy templates for each formula, and the characteristic of the shape of the local part of human body could be represented well.

This feature is extracted from the template level, not the pixel level like the HOG feature. It means that we calculate the values for each template, and generate the histograms of templates. It can reflect the relationship of three pixels. So this feature is more macrostructures than features which are extracted from the pixel, and has more discriminative ability. Both the gradient information and texture information are used in this feature. By using the definition of the formulas, we make two types of information homologous. In this way, we could get higher detection rate than only one type information used feature. We calculate the value by comparing the value of pixel and the value of neighboring pixel. We focus on the relationship of the pixel but not value of each pixel. So this feature has some properties of local binary pattern, such as the illumination-invariant. The illumination condition has no big influence on the detection result. HOG feature uses normalization operation to achieve the effect of the illumination. Floating calculation is necessary for the normalization. The normalization stage is not necessary for our feature. It means that only integral calculation can support our feature. This is very convenient for hardware based acceleration.

Based on the histogram of template feature, we propose a multi-scale block histogram of template to further improve the detection. One advantage of the HOT feature is that it is extracted from the template level, and it is more macrostructure than previous features. The MB-HOT feature is developed from the HOT feature and extends the template level. For HOT feature, the template is three pixels' combination. In MB-HOT feature, it is extended to three blocks' combination. So it can reflect the relationship of three blocks. In this way, the feature could be extracted from more macrostructure level. Theoretically, it can improve the result, and experiments on INRIA dataset show that it actually improves the performance a lot. Even linear kernel is used for training stage and classification, the result is nearly the same with the HOT feature when non-linear kernel is used.

A motion based feature is proposed to detect the human from the videos by using the optical flow information. This feature is also developed from the HOT feature, but gives some modifications to make it suitable to extract the motion information of the human body, especially the relative motion of the limb. First modification is about the dimension problem. Because the value in optical flow domain is two dimensional, different from the gray domain in which the value is one dimensional. So the definitions of formulas are changed. The second modification is about how to capture the relative motion. Because the HOT feature or the MB-HOT feature has already captured enough motion boundary information, the combination of appearance based feature and motion based feature can't improve the detection rate well. The block is divided into outer cell and inter cell, and the feature is extracted from the outer cell cells. The value of the outer cells is subtracted by the value of the inter cell. The motivation is that it can capture relative displacements of limbs to the background.

For the combination feature, we propose a histogram of modified local binary pattern. It is extracted from the images convolved with Gabor filter, as a feature vector to represent texture, and it can be considered as the supplement of gradient information. Because they use different types of information, they have low error dependency and can get higher detection rate.

In order to solve the pose-variant problem, we propose a pose-invariant feature. Before the feature extraction stage, we first estimate the poses of human, which are represented by a set of two dimensional points. The feature is extracted from the bounding boxes of different parts. The pose estimation problem is defined as a multi-output regression problem. A new definition of loss function is given to find the mapping function, by which the images can be mapped into the pose space. Orientated gradient is mainly used in pose estimation, which reduces the number of sub windows which are used in boosting based regression method, compared to

Haar-like feature. Experiments on INRIA dataset shows that pose estimation can increase the detection rate efficiently by correctly detection human shows pose are different from ordinary.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows: the histogram of template feature is firstly discussed in Chapter 2. Then the extensions of HOT feature: MB-HOT feature and a motion based feature are proposed in Chapter3. In Chapter 4, some acceleration methods are discussed. A combination feature and a pose-invariant feature are proposed in Chapter 5. Finally, Chapter 6 concludes the whole dissertation.

2 Histogram of Template Feature

2.1 Background and Related Work

Human detection technique is widely used in many applications ranging from image analysis, smart cars, and visual surveillance to behavioral analysis. In recent years, lots of research work has been focused on this field. But human detection is still a challenging task because of many difficulties. Most natural humans have large variations, such as the appearance, the pose and so on. Difference in clothes brings further challenge because some features such as skin color in the face detection can’t be used in this application. Besides, complex backgrounds, illumination, occlusions and different scales must be considered in the detection. A robust detector must be independent for all these variations.

The gradient information is efficient for the object detection. A lot of human descriptors contain the gradient information more or less [1-6]. Histogram of orientated gradient (HOG) [1] and covariance matrix (COV) [3] are excellent descriptors using the gradient information. HOG is a gray-level image feature formed by a set of normalized gradient histogram. In [1-2], HOG feature is compared with many other features, such as Haar-like feature, Wavelet feature and so on, and it gets the best performance. Covariance matrix integrates coordinates and intensity derivatives into a matrix. They represent the gradient information well, and can get a good result on some human datasets. But only using the gradient information may be not enough to detect humans from complex backgrounds or images in low resolution.

The texture information also has some discriminative abilities in the human detection. Some research work has been done on the feature of local binary pattern (LBP) [7] and Gabor filter [8]. Gabor filter and LBP are widely used in texture classification and face recognition. They represent the intensity information well. But only using these features is not enough to get the good result. The original definition of LBP is not suitable for the human detection. It must be combined with other features such as Laplacian EigenMap (LEM) in [8]. In [7], two variants of LBP: Semantic-LBP and Fourier-LBP are proposed. The modified definition of LBP makes it suitable for the human detection.

Besides the feature extraction, the training method is also very important for the human detection. They are two key components for the pattern classification problem. The features extracted from a large number of training samples are used to train a classifier. Support vector machine (SVM) [9] and various boosting methods [10] are efficient to train a classifier in practical applications. The SVM has some advantages. It is easy to train and the global optimum is guaranteed. The variance caused by the suboptimal training is avoided for the fair comparison. The boosting method combined with the cascade strategy is widely using in real-time applications.

The boosting method aims at producing an accurate combined classifier from a sequence of weak classifiers, which are fitted to iteratively reweighted versions of the data. The cascade strategy saves detection time and makes it possible to detect object real time.

So there are two research directions for the human detection: finding more discriminative local features [1, 3], and developing more efficient training methods [11].

The main contribution of this chapter is along the first direction. It focuses on building a more powerful local feature for the human detection. A new feature, histogram of template, is proposed. It extracts the texture information as well as the gradient information, and makes the two different types of information homologous. Compared with features using the gradient information, such as the HOG feature, the proposed feature shows more discriminative ability. Besides, this feature can encode the relationship of three pixels in one template. Compared with features that deal with each pixel independently, HOT feature can get higher detection rate. Last, the HOT feature has some properties of local binary pattern, such as illumination-invariance. So the normalization is not so important for the detection result. This property can be used to reduce computation complexity in some circumstance.

Human detection algorithms now can be separated into three groups.

The first group of methods is based on local features [1-4, 6-7, 11-13]. They extract some features from sub regions of images in the training dataset, to train a classifier by support vector machine (SVM) or boosting methods, such as Adaboost or Logitboost. For a new image, they extract the same features and send them to the classifier which will give a classification result. In [14], a local receptive fields (LRF) feature is extracted using multilayer perceptrons by means of their hidden layer. In [13], Haar wavelet is used as human descriptor. SVM in [9] is used to train the classifier. [1] uses the HOG feature as descriptor for the human detection, and [2] is developed from this one. It integrates the cascade-of-rejecter approach, and uses the Adaboost method in [11] to choose best sub window in each stage. In [11] Haar-like feature is used to detect humans. It uses the integral image to speed up the detection process. Cascade rejection method is proposed to make real-time human detection possible. In [3] the covariance matrix feature is used as human descriptor, to represent the coordinates, and the gradient information of humans. Covariance matrix can be formulated as connect Riemannian manifold. Each matrix can be treated as a point in Riemannian manifold, and can be mapped into a vector space. An edge let descriptor is used in [12] for the human detection. Different from just combining the orientations in horizontal and vertical direction in [4], it combines the orientations in edge let defined direction, which makes it more efficient for the human detection. This group of methods has a good performance, and if enlarge the training dataset, the detection rate can be improved. The workflow of the learning based method can be seen in Figure 2-1.

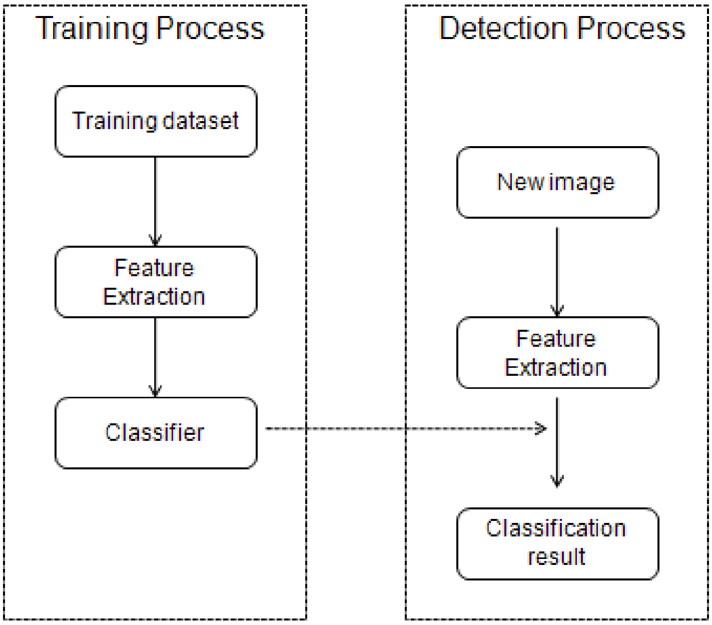


Figure 2-1 Workflow of learning based method

The second group of methods is based on local appearance, and [15-18] are based on this. They detect the interesting points in the training images and use the patches around the interest points to construct a codebook. When given a new image, they first find the similar patches in codebook and all patches vote for the positions of humans.

The third group is based on chamfer matching. They use human templates to find the most marching regions in the edge map of an input image. [19-20] are based on this method. In [19] a direct template matching approach for the global shape-based human detection is proposed, and [20] is developed from this but uses some hierarchical templates to reduce the detection time and solve the occlusion problem to some extent. These methods may not give a good result when there are too many edge clusters in the edge map.

Our method belongs to the first group. It uses HOT feature to extract the texture information and the gradient information for the human detection. Two types of information are made homologous to increase the discriminative abilities of the proposed feature. The covariance matrix feature in [3] gets higher detection rate than HOG feature in [1]. But the training method is different. [3] uses the logitboost method and the size of sub windows is variable, but the SVM training method and the fixed sub window strategy are used in [1]. So it is hard to say that whether the covariance matrix feature is more discriminative or the training method is better. The HOT feature is compared with the HOG feature using the same training method, for the fair comparison.

2.2 Definition of Histogram of Template Feature

First, I will give two most popular features for human detection: HOG feature and COV feature.

HOG is developed from the SIFT algorithm [5]. For calculating the HOG feature, the image is divided into blocks. The blocks overlap with each other. Each block contains four cells. Cell is the basic unit for the feature calculation. For each pixel $I(x, y)$, the orientation $\theta(x, y)$ and the magnitude $m(x, y)$ of the gradient are calculated by

$$dx = \mathrm{I}(x+1, y) - \mathrm{I}(x-1, y) \qquad \text{Equation 2.1}$$

$$dy = \mathrm{I}(x, y+1) - \mathrm{I}(x, y-1) \qquad \text{Equation 2.2}$$

$$m(x, y) = \sqrt{dx^2 + dy^2} \qquad \text{Equation 2.3}$$

$$\theta(x, y) = \tan^{-1}(dy / dx) \qquad \text{Equation 2.4}$$

A histogram is calculated for each cell, and the length of each bin is the sum of magnitude of the pixels whose orientations are in the corresponding interval. In [4], each block contains 2×2 cells, so a block can be represented by a 36-dimensional vector.

COV calculates a vector for each pixel in a sub window:

$$[x, y, |I_x|, |I_y|, \sqrt{I_x^2 + I_y^2}, |I_{xx}|, |I_{yy}|, \arctan\frac{|I_x|}{|I_y|}]^T \qquad \text{Equation 2.5}$$

Where x, y are pixel locations, and I_x, I_{xx}, I_y, I_{yy} are intensity derivatives. The last term is the edge orientation. So for each sub region, we calculate a set of 8-dimensional vectors, and a covariance matrix can be obtained from these vectors:

$$C_R = \frac{1}{S-1}\sum_{i=1}^{S}(z_i - \mu)(z_i - \mu)^T \qquad \text{Equation 2.6}$$

Where μ is the mean, S is the number of these vectors. Due to the symmetry of covariance matrix, only the upper triangular part is stored as the feature for the detection. A descriptor of a sub region is a 36-dimensional vector.

The HOG and the COV feature are mainly depended on the gradient information. There are some disadvantages of gradient-based features.

Sometimes, the gradient information is ambiguous. The same gradient may correspond to the different curves. See Figure 2-2 for example. Point P is the intersection of curve A and curve B. Only using the gradient information of P is not enough to discriminate A and B. But if the template feature is used, because the smooth degrees are different, P on A is more likely to meet the second template and the P on B is more likely to meet the first template.

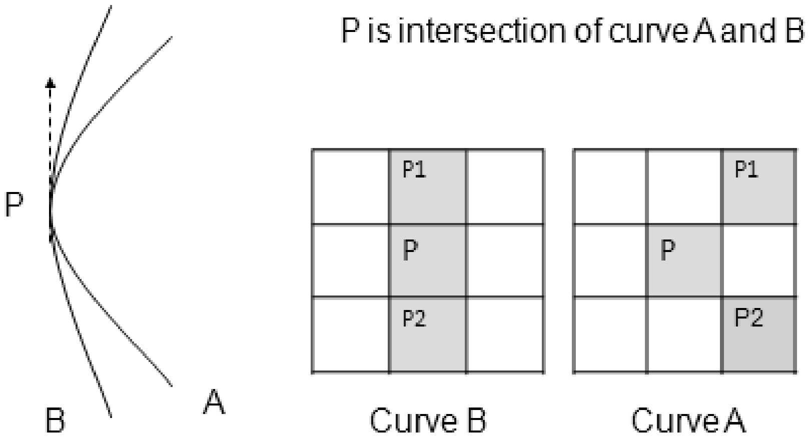


Figure 2-2 Disadvantage of Gradient based feature. It may be ambiguous in some circumstance, if only gradient information is used.

Gradient based features almost only use the gradient information for the detection, and drop the texture information in the original image, although three channels of color image are used in gradient calculation. The texture information also shows discriminative abilities in LBP based features [7-8] and local appearance based features [15-18]. So if texture information can be used with the gradient information, more accurate detection result can be obtained.

Histogram of Template feature is proposed here. Some templates are given to define the spacial relationship of three pixels. See Figure 2-4 for example.

In Figure 2-4, 12 connected templates are given. In our experiment, the templates (1) to (8) are used for the feature calculating. 12 templates can be used for more accurate result.

These templates are used in some formulas. The texture information and the gradient information are also used in these formulas, to give a concrete definition of this feature. The formulas are designed to capture the shape of the human body, and have reasonable computation complexity.

For texture information, two formulas are given as following. First is:

$$F(T) = \begin{cases} 1 \; if \; I(P) > I(P1) \; and \; I(P) > I(P2) \\ 0 \; other \; wise \end{cases} \qquad \text{Equation 2.7}$$

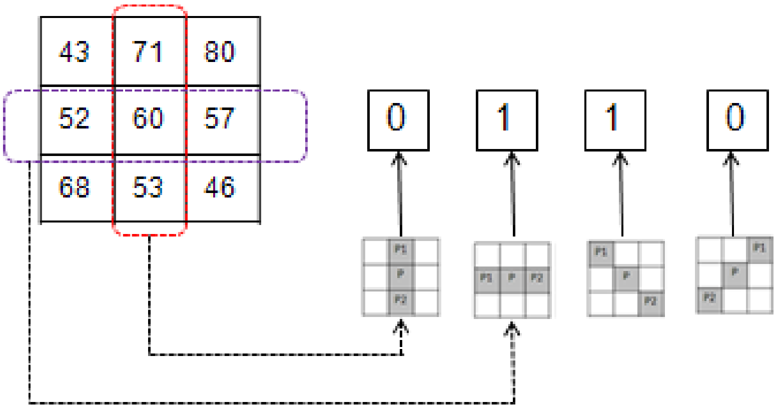


Figure 2-3 Calculation for the Equation 2.7

For each template, if the intensity value of P is greater than the other two, it is regarded that the pixel P meets this template. This is can be seen in Figure 2-3. It can capture the pixels that have the greatest value in one template, and the histogram of pixels that satisfy each template in a sub window can reflect the properties of local part of human body well.

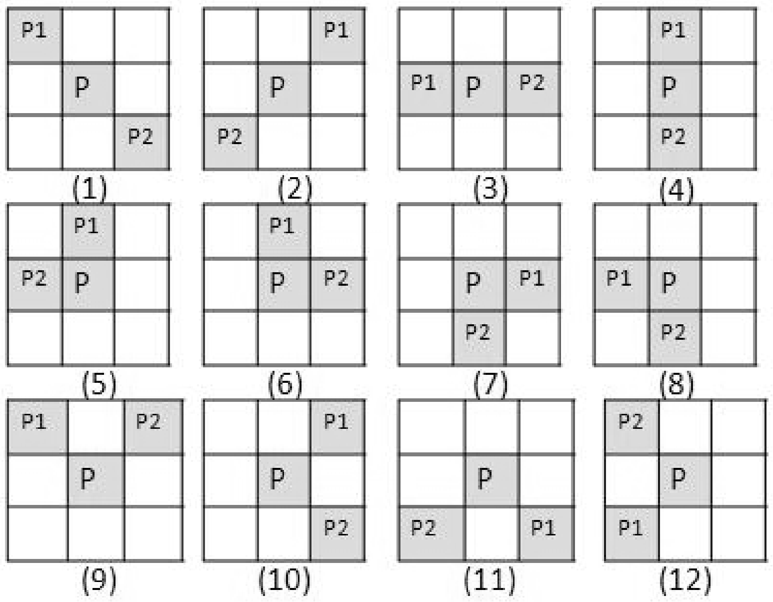


Figure 2-4 There are 12 templates here. They are three pixels' combination.

For each sub window, the number of pixels meeting each template is calculated to get a histogram. See Figure 2-5. For example, eight templates are used to extract the feature.

The histogram has eight bins and each bin corresponds to one template. The value of each bin is the amount of pixels which meet the requirement of this template in this sub region.

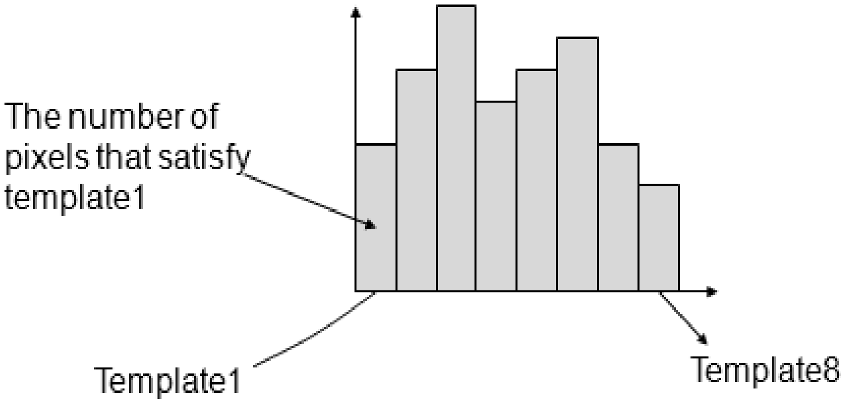
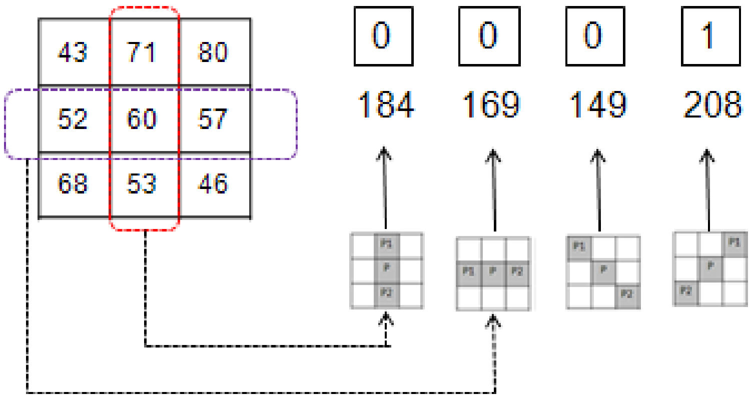


Figure 2-5 Example of histogram of template for one formula; 8 templates are used, and they correspond to 8 bins. The value of each bin is the number of pixels that meeting corresponding template.

The second formula is:

$$F(T_i)=\begin{cases}1 \ \mathit{if}\ \ i=\underset{i}{\operatorname{argmax}}\{I(P_i)+I(P1_i)+I(P2_i)\}\\ 0 \ \mathit{other\ wise}\end{cases} \qquad \text{Equation 2.8}$$



43	71	80
52	60	57
68	53	46

Figure 2-6 Calculation for the second Equation 2.8

The sum of intensity values of three pixels in template k is greater than the values of other templates; it is can be regarded that P meets template k. A histogram can be calculated by using the second formula. By using this formula, we could find the template that has the greatest sum. They can be regarded as the basic unit of human body shape and the shape of human body can be represented well. See Figure 2-6.

For the gradient magnitude information, there exist similar formulas:

$$F(T) = \begin{cases} 1 \ \textit{if} \ \ M(P) > M(P1) \ \textit{and} \ \mathrm{M}(P) > M(P2) \\ 0 \ \textit{other wise} \end{cases} \qquad \text{Equation 2.9}$$

$$F(T_i) = \begin{cases} 1 \ \textit{if} \ \ i = \arg\max\limits_{i} \{M(P_i) + M(P1_i) + M(P2_i)\} \\ 0 \ \textit{other wise} \end{cases} \qquad \text{Equation 2.10}$$

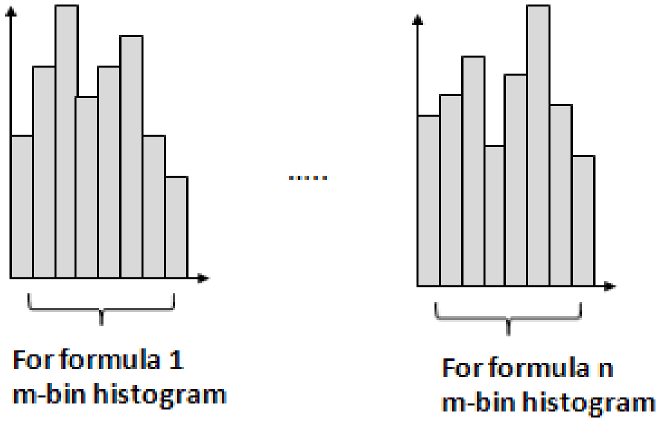


Figure 2-7 Final HOT feature for a sub window. It is a $m \times n$ dimensional vector. In our experiment, m=8 and n=4.

For the gradient orientation information, $[0, 2\pi)$ is first divided into nine scopes. It calculates which scope the orientation of the pixel belongs to, and uses the serial number of scope to replace the orientation value. Then, the following formula can be used to extract the orientation information.

$$F(T) = \begin{cases} 1 \ \mathit{if} \ \mathrm{Ori}(P) = Ori(P1) \ \mathit{and} \ Ori(P) = Ori(P2) \\ 0 \ \mathit{other\ wise} \end{cases} \qquad \text{Equation 2.11}$$

Eight templates are usually used to extract the feature, so for each formula, an eight-dimensional vector can be obtained. These vectors are combined together as the final feature. See Figure 2-7.

Then, we want to discuss why we use these formulas. There are two reasons.

First, these formulas can capture the shape of human body well. These formulas are designed to obtain the shape of local part of the human body. Eq2.7, Eq2.8 use gray value, Eq2.9, Eq2.10 use gradient magnitude value and Eq2.11 uses gradient orientation value. Formula Eq2.8 and Eq2.10 could capture the template that has the greatest sum of values, and formula Eq2.7 and Eq2.9 capture the pixels that are greater than the other two pixels in one template. Formula Eq2.11 captures pixels that have the same gradient orientation with the other two pixels in one template. They can reflect the properties of local edge of the human body shape. See Fig.1 for example. Gradient value is used in Eq2.9, Eq2.10. When Eq2.9 is used, the pixels in red line are easy to satisfy template (2) and when Eq2.10 is used, they are easy to satisfy template (1). Similar, the pixels in the green line are easy to satisfy template (3) when formula Eq2.9 is used, and they are easy to satisfy template (4) when formula Eq2.10 is used. By using such kind of formulas, we could extract the shape information efficiently. We could calculate the histogram of pixels that satisfy templates for each formula, and the characteristic of the shape of the local part of human body could be represented well. We give the definition of formula Eq2.7 and Eq2.8 in the same way, but the gray value is used. For formula Eq2.11, it uses the gradient orientation value. We want to find the pixels that have the same orientation values with their neighboring pixels in the templates. The pixels in the red line and the black line in Figure 2-8 may have the same orientation respectively. It can reflect the change of the orientation of the shape of human body. We give the definition of formula by the above assumption, and experimental result confirms this.

Second, the computation complexity is small when these formulas are used. There are only addition and comparison operation. It is easy for hardware based acceleration.

I also want to explain why these templates are used. This is because that these selected templates can represent the shape of human body well. The three pixels in one template should contain some shape information of human body, so it can have discriminative abilities to detect human from images.

There are many templates for three-pixel combination, but maybe many of them don’t contain useful shape information for detection. Increasing the number of templates can improve the detection rate, because it brings more information, some of which may be useful and some of which may be not. But if we increase the number of templates, the length of feature will also increase. It will bring more computation. And the result improved by increasing templates is limited. There exists limitation for detection rate.

So maybe it is not necessary to use all combination of three pixels. The result of 8 templates is better than HOG and COV. If more accurate result is requested, 12 templates can be used. But it bring more computation time. The time consumption of 8 templates based feature is nearly the same with HOG, but that of 12 templates are twice, according to our experiment, without software optimization or hardware acceleration.

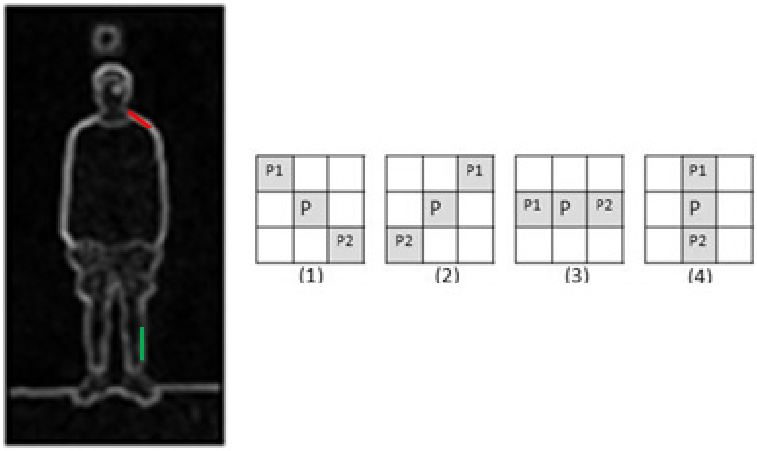
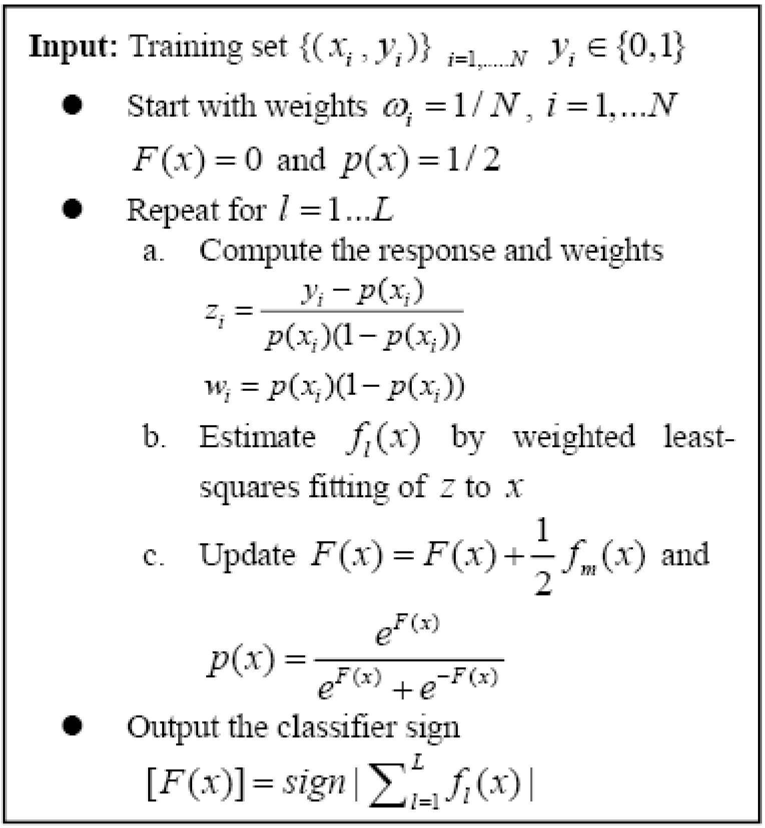


Figure 2-8 By using these formulas, the local shape of human body can be represented efficiently

The integral image can be used for feature extraction. For example, if 4 formulas and 8 templates are used, the histogram has 32 bins. 32 additional images are used. One image corresponds to one bin. If the pixel in the original image satisfies one template for one formula, the value of the pixel in the corresponding additional image is 1; otherwise it is 0. Then, by constructing the integral images of the additional image, we could get the 32-bin histogram for each sub window quickly.

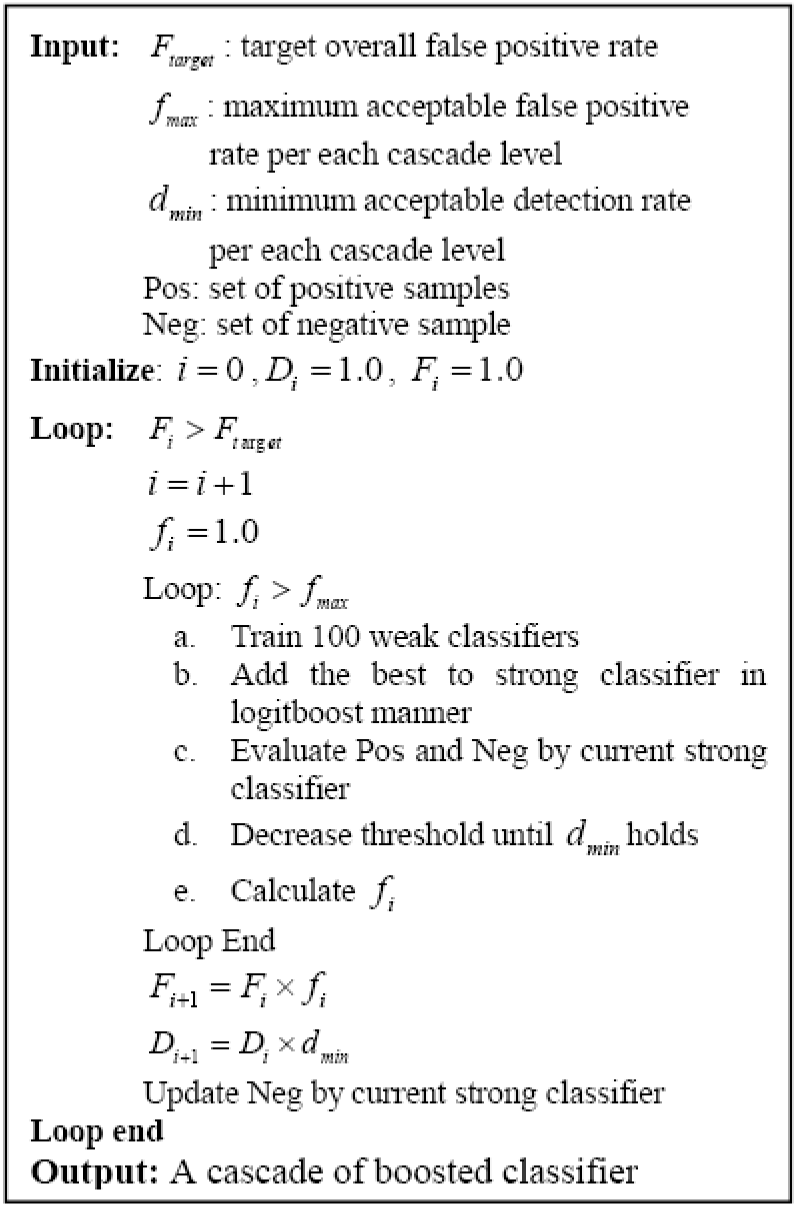
Compared with HOG feature, HOT feature has three advantages. First is that it not only uses gradient information, but also uses texture information. Although HOG feature also uses three channels of color image for gradient calculation, the texture information is ignored and it is not treated as a cue for detection. The second is that HOT feature is more macrostructures. HOG is actually an orientation voting, and after the gradient is computed, the feature is calculated from pixel level. The HOT feature is specific pattern voting and the feature is extracted from middle level, which contains several three pixels' combination. So HOT feature is more discriminative, and experiment confirms this point. The third is that HOT feature is illumination-invariant, so the normalization is not necessary as HOG feature.



```
Input: Training set $\{(x_i, y_i)\}_{i=1,....N}$ $y_i \in \{0,1\}$
●   Start with weights $\omega_i = 1/N$, $i = 1,...N$
    $F(x) = 0$ and $p(x) = 1/2$
●   Repeat for $l = 1...L$
    a.  Compute the response and weights
        $z_i = \frac{y_i - p(x_i)}{p(x_i)(1 - p(x_i))}$
        $w_i = p(x_i)(1 - p(x_i))$
    b.  Estimate $f_l(x)$ by weighted least-squares fitting of $z$ to $x$
    c.  Update $F(x) = F(x) + \frac{1}{2} f_m(x)$ and
        $p(x) = \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}}$
●   Output the classifier sign
    $[F(x)] = sign \mid \sum_{l=1}^{L} f_l(x) \mid$
```

Figure 2-9 Logitboost training method

The training method is also very important for the detection result. A reasonable training method improves the result efficiently. So for the fair comparison of different features, the effect of training method should be considered. Support vector machine and many boosting methods, such as Adaboost, Logitboost and Gentleboost, are widely used in many tasks. In our experiment, SVM is used for comparison.



```
Input:  $F_{target}$ : target overall false positive rate
        $f_{max}$ : maximum acceptable false positive
            rate per each cascade level
        $d_{min}$ : minimum acceptable detection rate
            per each cascade level
        Pos: set of positive samples
        Neg: set of negative sample
Initialize: $i = 0$, $D_i = 1.0$, $F_i = 1.0$
Loop:   $F_i > F_{t\arg et}$
        $i = i + 1$
        $f_i = 1.0$
        Loop: $f_i > f_{max}$
          a.  Train 100 weak classifiers
          b.  Add the best to strong classifier in
              logitboost manner
          c.  Evaluate Pos and Neg by current strong
              classifier
          d.  Decrease threshold until $d_{min}$ holds
          e.  Calculate $f_i$
        Loop End
        $F_{i+1} = F_i \times f_i$
        $D_{i+1} = D_i \times d_{min}$
        Update Neg by current strong classifier
Loop end
Output: A cascade of boosted classifier
```

Figure 2-10   The training algorithm for cascade rejection strategy.

First, the boosting based training method. The Logitboost training method can be used in this application. The workflow of Logitboost method can be seen in Figure 2-9. In the loop, the best weak classifier is added. Because the final classifier contains many weak classifiers, the cascade rejection method can be used to accelerate the classification. If an image is considered as the human, it has to pass all the cascades. If it can’t pass one cascade, it doesn't have to pass the following cascades and is regarded as non human directly. The workflow of training for the cascade rejection method can be seen in Figure 2-10.

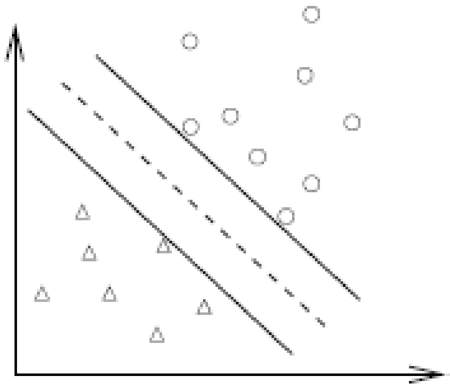


Figure 2-11 Decision plane of SVM

SVM in [9] is effective for learning with small sampling in high-dimensional spaces. The objective of SVM is to find a decision plane that maximizes the inter-class margin. See Figure 2-11. The feature vectors are projected into a higher dimensional space by kernel function. The kernel function makes it possible to solve the linear non-separable problems and the mapping function is not necessarily known explicitly. So the decision rule is give by the following formula.

$$f(x) = \sum_{i=1}^{N_s} \beta_i K(x_i, x) + b \qquad \text{Equation 2.12}$$

Where x_i are support vectors, N_s is the number of support vectors. $K(x, y)$ is the kernel function. So the training process of SVM is to find the proper parameters of Eq2.12.

Compared with boosting methods, SVM needs more computational resources and it is difficult for real-time application. The size of sub windows should be fixed. It is hard to take the variable sub-window size strategy due to the computation problem, although the variable window strategy can improve the performance efficiently. But for the comparison purpose, SVM is suitable. The training time is less and the optimization is guaranteed. The difference of the performance caused by the optimization can be ignored. The parameters of SVM are controllable. The suitable parameters can be selected avoiding the difference caused by the parameter difference. In our experiment, LibSVM [21] is used. RBF and linear kernel functions are used in our experiment respectively.

2.3 Experimental Result

2.3.1 Data Set

The experiment is performed on INRIA dataset [22]. It is widely used for the human detection in still images. The database contains 1774 human annotations and 1671 person free image. This dataset is made up of a training dataset and a testing dataset. 1208 human annotations and 1218 non-human images are used for the training stage, and the left images for testing. For positive samples, left-right reflections are also used. So, 2416 positive samples are used for training. More detail can be seen in [22]. There are varieties of variations in human pose, clothing, lighting, clutters and occlusions, so it is difficult for the human detection and it is suitable as a benchmark for comparison between different methods. Some examples can be seen in Figure 2-12.

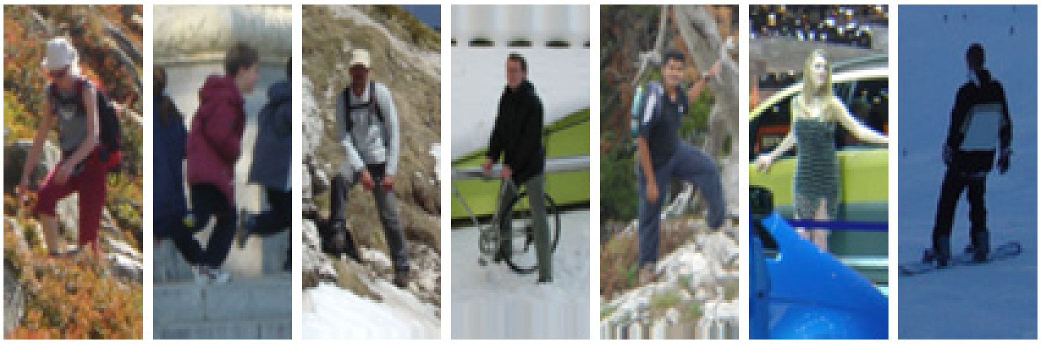


Figure 2-12 Selected positive samples in INRIA dataset

2.3.2 Evaluation Method

For the purpose of comparison, we need to quantify the different detector's performance. There are two methods to evaluate the detection method. The first is the window level detectors using detection error tradeoff (DET), or receiver operating characteristics (ROC) curves to evaluate the binary classifier performance. The second is recall-precision (RP) curves to measure the accuracy. In both case the detectors will output confidence scores for detections. The larger this value is, it has higher possibility that it belongs to the human. For all detectors, both evaluation methods start from the lowest possible score, evaluate the respective parameters such as number of false positives, recall rate or precision rate, and then progressive increase the threshold until they reach the highest possible score. Each tested threshold provides a point on the curve.

We don't use the RP curves for comparison of the features, because the definition of precision in RP curves uses the number of false positives. It is significantly high for a window level classifier as it often tests million of negative examples compared against few thousand positive examples. The other disadvantage is that localization results can be defined in multiple ways. For example, how much overlapping predictions are made should all predications be considered correct or only one prediction as true positive and rest as false positives. These issues bring additional parameters in the evaluation.

In our experiments, we use the DET curves. It forms natural criterion for our binary classification tasks as the measure the proportion of true detections against the proportion of false positives. They plot miss rate versus false positives on a log-log scale. The definition of the miss rate is as follow:

$$MissRate = 1 - \mathrm{Re}call = \frac{FalseNegatives}{TruePositives + FalseNegatives} \qquad \text{Equation 2.13}$$

We plot false positives per window (FPPW) along x-axis and miss rate along y-axis. Lower values denote the better classifier performance. DET plots are used extensively in many evaluations. The present the same information as ROC curves but allow small probabilities to be distinguished more easily.

We often use a false positive rate of 10^{-4} FPPW as reference point for results. This may seen arbitrary but it is no more so than. Small FPPW are necessary for detector to be useful in practice owing to be the number of window tested, typically of the order of from thousand to ten thousands windows. In a multi-scale detector, 10^{-4} corresponds to a raw error rate of about 0.8 false positives per image tested. At these FPPW, the DET curves are usually very shallow so even very small improvements in miss rate are equivalent to large gains in FPPW at constant miss rate.

2.3.3 Experimental Analysis

In order to show the advantage of the proposed feature, we design the following three experiments. In the first experiment, we compare our feature with HOG feature and COV feature. We use the same strategy with [1]. The re-sample strategy and normalization strategy are also used in our experiment. The work flow of the re-sample strategy can be seen in Figure 2-13. The size of sub window and the stride between sub windows are provided by [1].

In the second experiment, a random ensemble strategy is used. So we don’t have to consider the size of sub window and the stride. The comparison is fairer.

In the third experiment, we evaluate the length of the proposed feature. Only 8 templates are used in the first experiment, and we show that if more templates are used, the performance would be further improved.

In the forth experiment, we show the results with respect of the change of parameters and training strategies. We want to show that the result of our feature in different configurations.

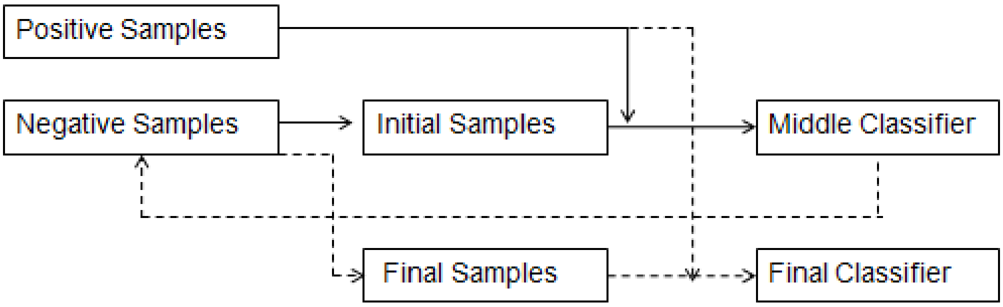


Figure 2-13 The workflow of the Re-sample strategy

I want to explain the parameters used in our method first. For feature extraction part, there are two parameters, the size of sub window and the stride of sub windows; for training part, if Ker-SVM is used, the parameters include C value and g value. Cross-validation can be used for choosing C and g. it is a common method and is widely used in such kind of work. LibSVM also provides tool for cross-validation. Only training images are used in cross-validation. So the size of sub window and the stride of sub windows are key parameters in our experiment.

(1) In the first experiment, we compare the HOT feature with the HOG and COV feature. According to [1], using RGB information for gradient calculation will improve the performance by 1.5% at 10-4 false positives per window (FPPW). Re-sample strategy will improve 5% at 10-4 FPPW. Re-sample strategy means that positive samples and some negative samples random selected from natural negative samples in training dataset are used for training first. The middle classifier is obtained. Then, this classifier is applied to nature negative images and selects the hard negative samples. The initial samples and the hard negative samples are used for training final classifier. See Figure 2-13.

In our experiment, only gray information is used in implementation, but re-sample strategy is used in experiment. It can improve performance efficiently. Better result is expected if RGB information is used for gradient calculation for HOT feature. We use nearly the same strategy with the HOG feature. The Re-sample and normalization are used in our experiment, just like the HOG in [1]. The size of sub window and the stride are also provided by [1].

In the re-sample stage, 2416 positive samples and 12800 negative samples random selected are used as the initial training dataset. And there are 39000 hard negative samples in our experiment. The initial training dataset and the hard negative samples are used to get the final classifier. The Linear kernel and the RBF kernel are used for training respectively. C value and g value of RBF kernel are selected by cross-validation method, which is a common method in SVM. LibSVM provides this tool. C and g are obtained by only using the training dataset. The C value is 128 and g value is 0.00048828125. We also use the default C and g to evaluate the performance of our feature. We can see the result returned by the cross-validation tool from Figure 2-14

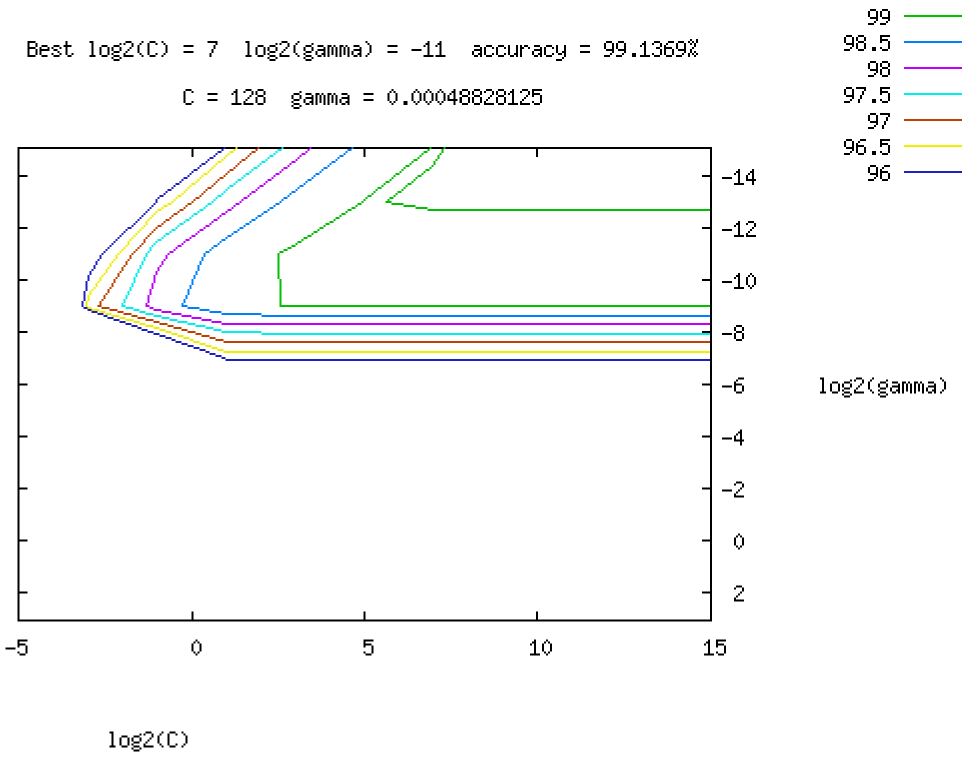


Figure 2-14 Result of the cross-validation

From Figure 2-15, it can be seen that the result is nearly the same. The size of block is set as 16×16 and the stride between two blocks is 8. They are the same with the HOG feature. The feature length of a block is 32, since the first eight templates in Figure 2-4. So the length of the feature for a 64×128 image is 3360. It is shorter than HOG and COV, which means that the computation complexity and the memory consumption is less. The comparison result can be seen in Figure 2-15. The data of HOG and COV are copied from [3] for comparison. The configuration for each feature can be seen in Table 2-1.

From Figure 2-15, it can be seen that the HOT feature gets higher detection rate than HOG and COV feature at 10^{-4} FPPW. Usually, we compare the miss rate at 10^{-4} FPPW [1]. Take into account that COV feature uses variable sub window. It can improve the detection rate a lot compared with using fixed sub window [2-3, 7]. We could say that our feature outperforms HOG and COV.

Table 2-1 The experiment configurations of three features

	HOG	COV	HOT
Feature Dimension	36	36	32
Re-Sample	Y	Y	Y
Sub window size	16×16	Variable	16×16
Stride	8	N	8
Training method	SVM	Logitboost	SVM
Normalization	Y	Y	Y
Unbalance data	N	N	N

(2) In the second experiment, we try to reduce the influence of block size and stride. A random ensemble strategy in [7] is used.

From an image (64×128) in our experiment, lots of sub windows in different sizes and on different positions can be extracted. The minimum sub window size is set as $K \times K$. This size is incremented in a step of K horizontally and vertically, or both. Finally we can get all possible sub windows: $W_{subwin} = \{r_i\}$. In our experiment, K=8. So the cardinality of W_{subwin} is 4896. Smaller K will give more sub windows.

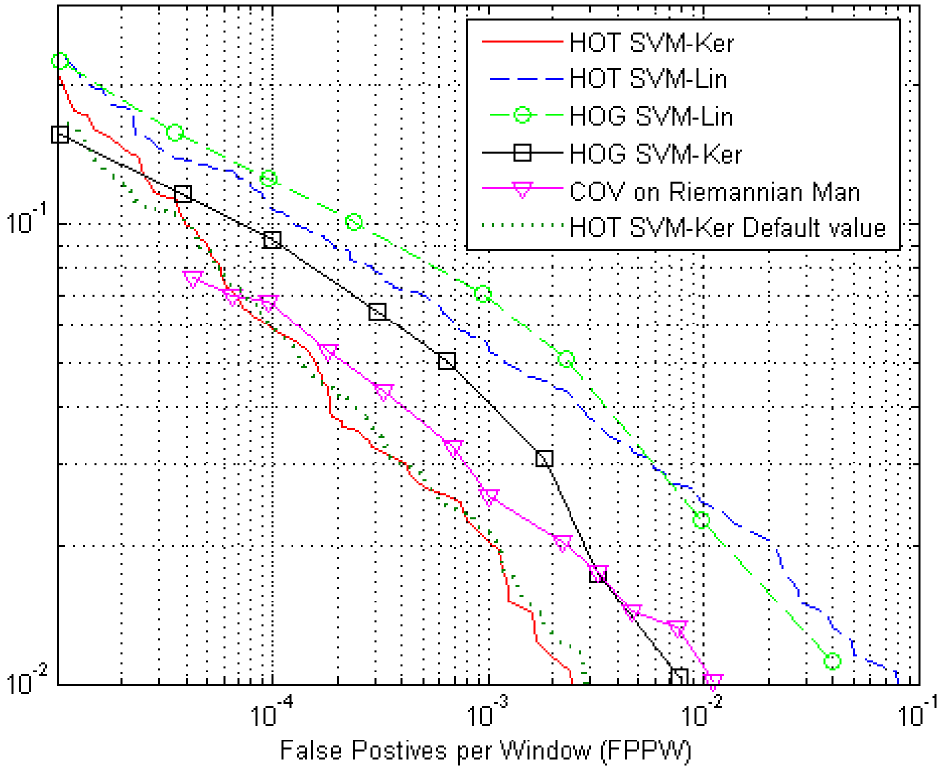


Figure 2-15 Comparison with methods of HOG [1] and COV [3]. The curves of HOG and COV are copied from [3].

Random ensemble means that n_w sub windows are random selected from W_{subwin}. In our experiment, n_w=150. A set of sub windows can be obtained: $\{r_j, j=1,2,...n_w\}$. For each sub window, a feature is calculated. The features for all random selected sub windows can be obtained as: $\{f_j, j=1,2,...n_w\}$. So the final feature for this detection window can be represented as $F=\{f_1, f_2 ... f_{n_w}\}$. If the lengthen for each sub window is d, the dimension of the final feature is $d \times n_w$.

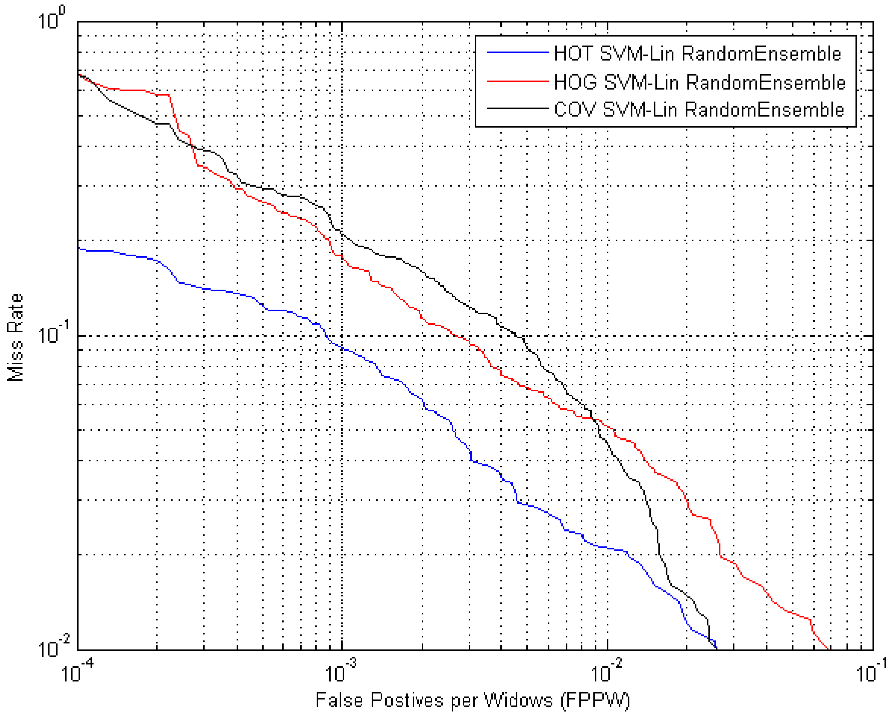


Figure 2-16 Comparison of three features Random ensemble strategy is used here. The influence of parameters can be ignored

By using the random ensemble strategy, the influence of block size and stride can be ignored, because all sub windows are random selected from w_{subwin} . HOT, HOG and COV are evaluated by using this strategy. The initial training dataset in the first experiment is used for training. Lin-Ker SVM is used, so there is no C value and g value. In this evaluation strategy, there is no any parameter for the feature extraction. The comparison of three features can be seen in Figure 2-16. Our feature outperforms HOG and COV in this experiment, which shows the discriminative ability of our feature.

(3) In the third experiment, lengthen of HOT feature is evaluated. The HOT feature is computed by using formulas and templates. The first eight templates in

Figure 2-4 are used for the 32-dimensional feature for a sub window in the above experiments. If more templates are used, the detection result will be improved. The performance is evaluated in this experiment.

The template is actually the three pixels' combination. There are 9×8×7 possible templates in all in a 3×3 region. We only consider the connected one. Some connected templates containing central pixel can be seen in Figure 2-4 and Figure 2-17. Other connected templates can be obtained by shifting these 20 templates.

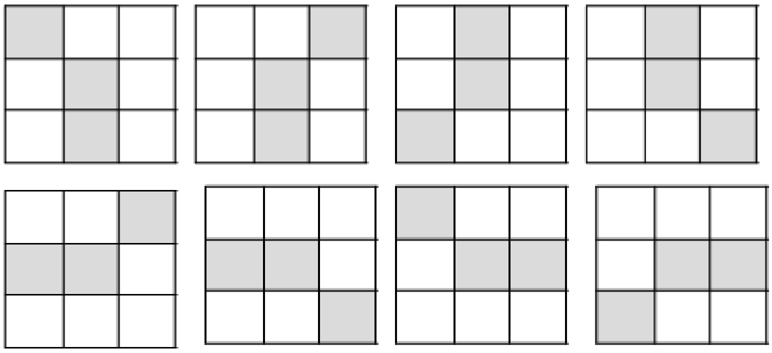


Figure 2-17 Another 8 connected templates containing central pixel

The detection result of 12 templates and 20 templates can be seen in Figure 2-18.

In Figure 2-18, the initial training dataset and SVM of RBF kernel are used for training. For 8 templates case, the first 8 templates in Figure 2-4 are used. For 12 templates case, all templates in Figure 2-4 are used. For 20 templates case, all corrected templates containing central pixel in Figure 2-4 and Figure 2-17 are used. From Figure 2-18, it can be seen that when increase the number of templates, the detection result is improved. But the improvement is limited when the number of template is increased from 12 to 20. When use more templates, the length of the feature will increase. It means that more time is needed for the classification. So it may be not necessary to use all the three-pixel combination. 8 templates or 12 templates are suitable for the human detection. It is a tradeoff between the detection rate and the computation complexity. Mention that in the first and second experiment, only the first 8 templates are used.

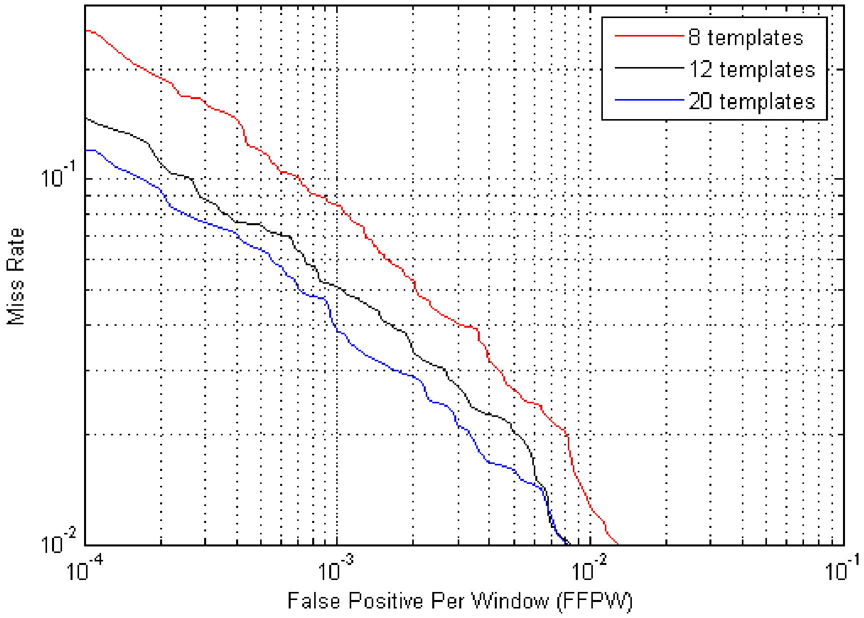


Figure 2-18 Detection result when more templates are used

(4) In the fourth experiment, the performances of HOT feature in different parameter configurations and training strategies are evaluated. We want to show the performance of proposed feature in the different configuration. For the training strategy, we consider the normalization strategy, unbalance data strategy; for the parameter, we evaluate the size of sub window and the stride between two neighboring sub windows. The initial training dataset in the first experiment is used for training.

Normalization schemes: In experiment, Non-norm, L1-norm and L2-norm strategy are used for comparison. Let v be un-normalized feature vector. The schemes are:

(a) Non-norm $v -> v$; Equation 2.14

(b) L1-norm $v -> v/(\|v\|_1 + \xi)$; Equation 2.15

(c) L2-norm $v -> v/\sqrt{\|v\|_2^2 + \xi^2}$. Equation 2.16

See Figure 2-19 for performance comparison. L2-norm outperforms Non-norm and L1-norm schemes, but the difference is not too much. So this step is not necessary for HOT feature. HOT feature has illumination-invariant property itself. This operation can't be ignored for HOG feature because of illumination. It means that we could accelerate the feature extraction by abandoning the normalization. So only the integer calculation is needed in the feature extraction. It is easy for hardware based acceleration. In the first experiment, L2 is used for fair comparison because HOG uses L2.

Unbalance data: Since the number of negative samples is larger than that of positive samples, more negative images are used for training than positive images. It is reasonable that different penalty value for different classes may increase the detection rate. C of negative samples: C of positive samples is set as 3:1, 2:1, 1:1, 1:2 and 1:3 for comparison. See Figure 2-20. It can be seen that if the penalty value of positive samples is greater than that of negative samples, the performance can be improved. But the difference is not too much. In the first experiment, we don’t consider it, and the same penalty is used for positive samples and negative samples for fair comparison.

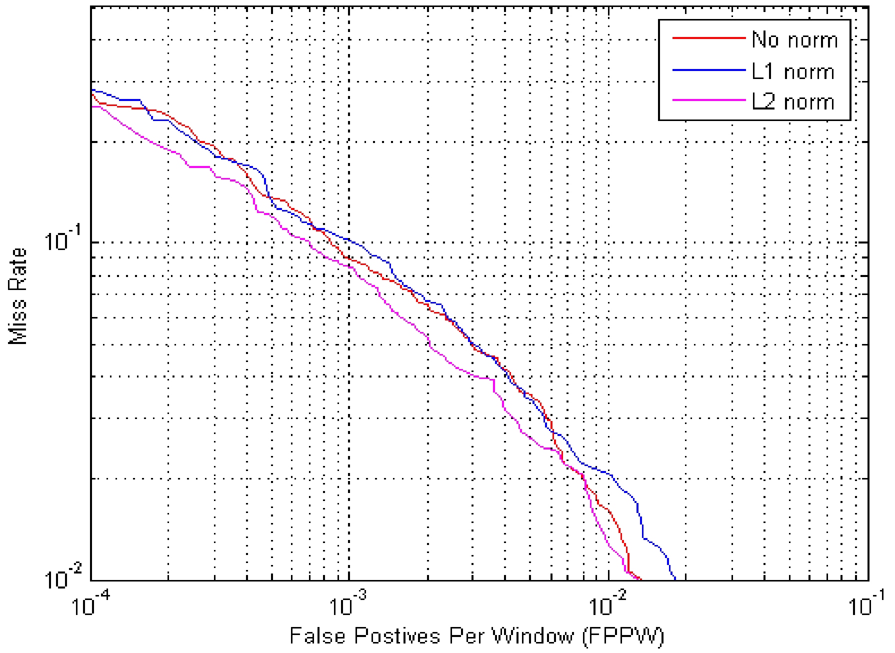


Figure 2-19 Comparison of different normalization schemes

Sub window size: For an inputting detection image (64×128), it is first divided into many sub windows (block). Sub windows can overlap with each others. Since the size of sub window is fixed, suitable size should be decided for training and detection. In experiment, 12×12, 16×16 and 20×20 are used for comparison. See Figure 2-21. 20×20 has the best performance. The result of 16×16 and 12×12 are nearly the same as 20×20. For fair comparison, 16×16 is used in first experiment.

Stride between sub windows: the distance between two neighboring blocks. The area of overlap region of two sub windows is decided by this value. The less this value is, the longer the final feature is. In experiment, 4, 8 and 16 are evaluated. See Figure 2-22. 8 is used in the first experiment, just like the HOG feature.

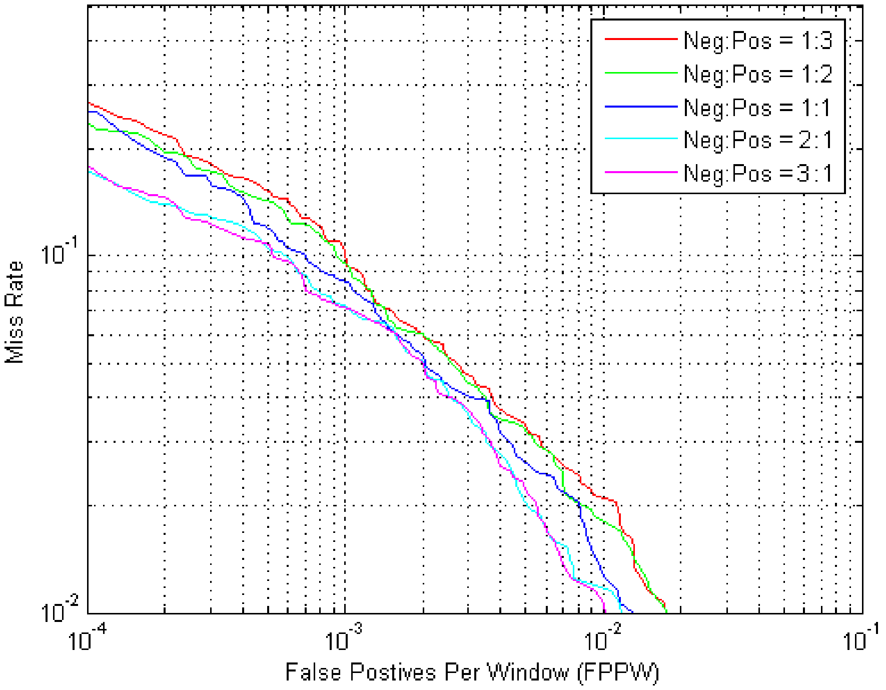


Figure 2-20 Different penalty values for the different classes

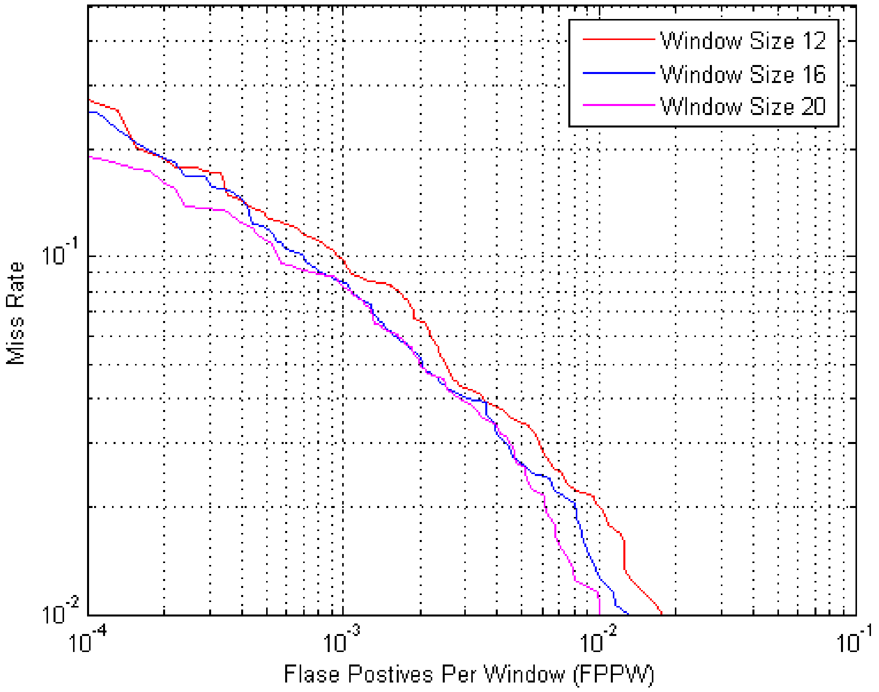


Figure 2-21 The size of the sub window

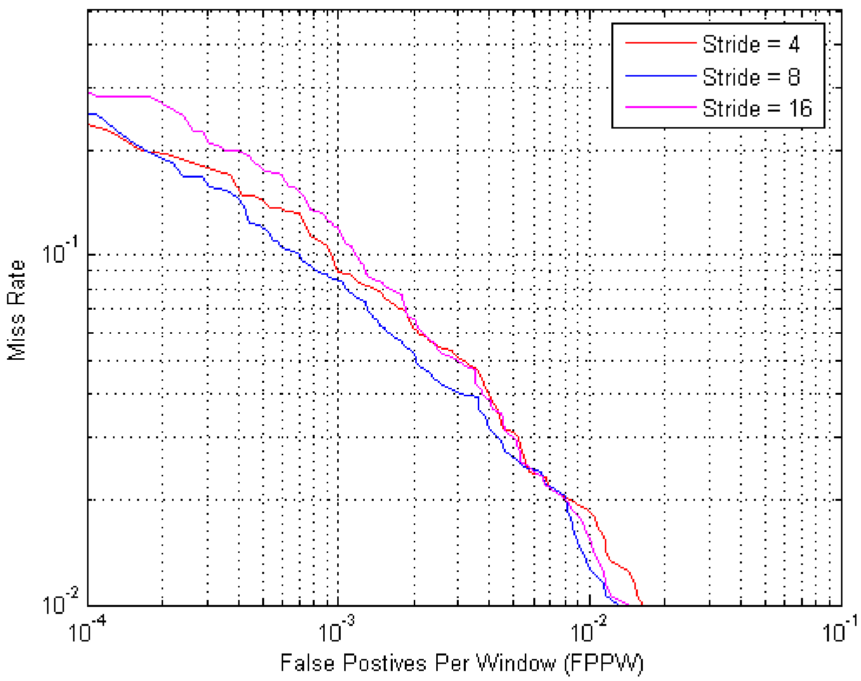


Figure 2-22 Stride between two sub windows

Finally, some detection results from natural images by using the classifier obtained in the first experiment can be seen in Figure 2-23.

2.4 Conclusions

The main contribution of this paper is to propose a more discriminative feature. We propose a new feature that has higher detection rate than HOG and COV in the same dataset, and we design four experiments to show the discriminative ability, and to prove that this is not achieved by parameter tuning.

In our first experiment, in order to compare our feature with HOG feature, we use the same training method and the same parameters with the HOG feature. The training strategies (Re-sample and normalization) and the parameters (size of sub window and stride of sub window) are provided by the original paper of HOG. When Ker-SVN is used, the C and g value are select by cross-validation tool provided by LibSVM, which is a common method in SVM based training method. Only the training images are used in cross-validation. We also give the result when the default C and g are used. In this case, our feature could get the higher detection rate on the same dataset, so we could say that our feature outperforms HOG feature.

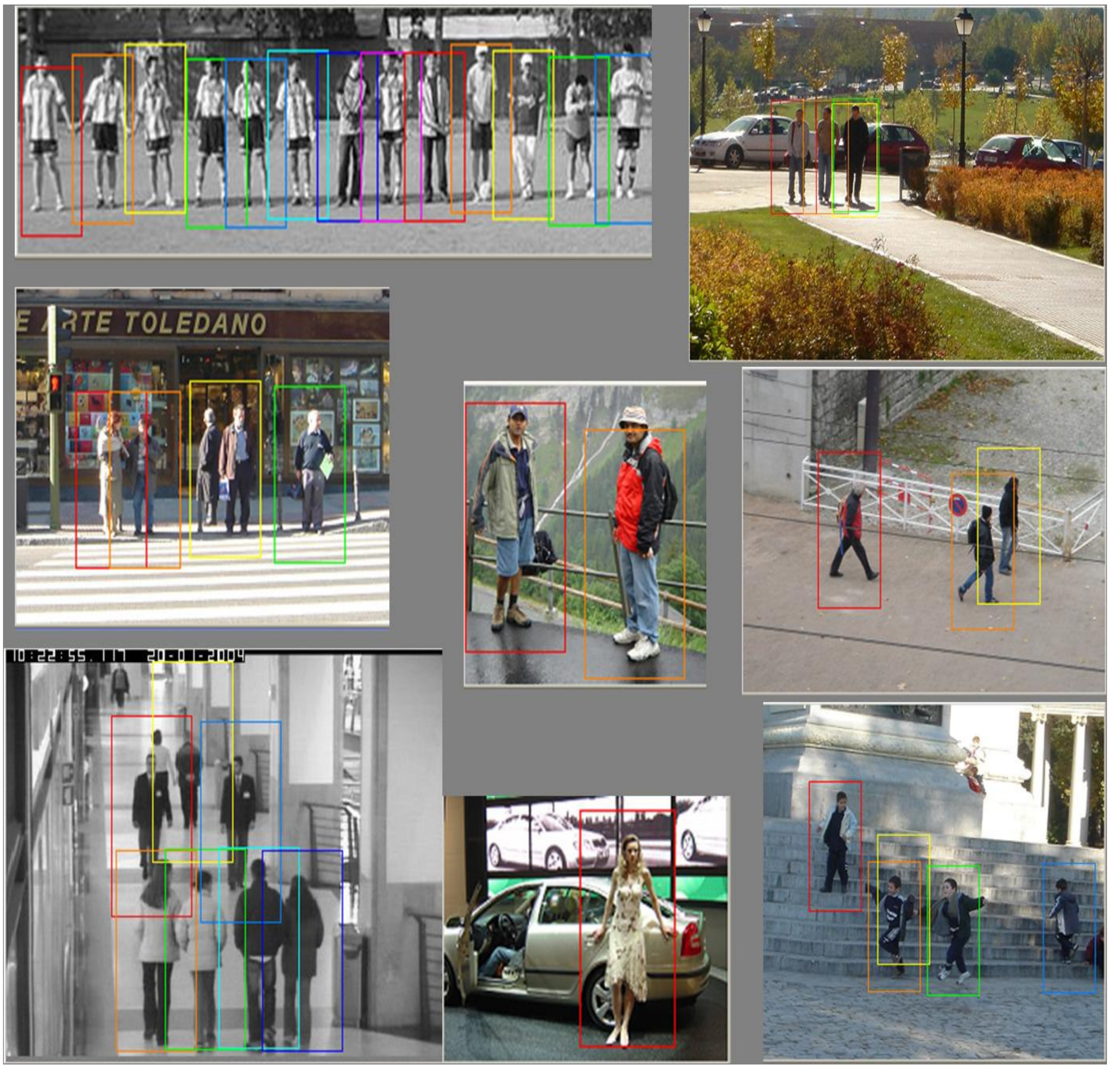


Figure 2-23 Detection result of natural images

We also compare our feature with COV feature in this experiment. In the original paper of COV, the logitboost training method is used. It means that the size of sub window is variable. It can improve the detection rate a lot. Although fixed sub window size is used in our method, HOT feature gets high detection rate than COV. And considering the computation complexity, the COV feature has to calculate the covariance matrix for each sub window, but our feature only contains addition and comparison operation. The computation complexity is less and it is easy for hardware based acceleration. So we could say our feature outperforms COV.

In order to further prove the effectiveness of proposed feature, we provide the second experiment. Random ensemble strategy is used. It means that the sub windows are random selected. So we don’t have to care about the size of sub window and the stride. Lin-SVM is used for evaluation. There is no parameter in this experiment. Our feature outperforms HOG and COV in this experiment. This can further prove the effectiveness of our feature.

In the third experiment, we show the results if more templates are used. In the first and second experiment, only 8 templates are used. If more templates are used, the performance can be further improved. It is a tradeoff between the detection rate and the computation complexity.

In fourth experiment, our feature with respect of change of parameters and training strategies is evaluated. We want to show that our feature is not affected by parameters too much.

From Figure 2-19, Figure 2-20, Figure 2-21 and Figure 2-22 in revised paper, you can see that the difference of performance is not too much. This experiment is designed to show the result in different parameter configurations. The parameter configuration used in the first experiment is provided by HOG. Some parameters used here even could get better result than the parameter configurations used in the first experiment. From these experiments, we find some other advantages of the propose feature.

From Figure 2-19, the difference of result of different normalization strategy is not too much. When HOG feature is used, the different between the feature with normalization and the feature without normalization is huge. So the normalization is necessary for HOG, but it is not necessary for our feature. It means that our feature doesn’t have to use normalization, so it can be done only by integer calculation, which is easy for hardware acceleration. It is another advantage of proposed feature.

From Figure 2-20, you can see that if different C values are used for positive samples and negative samples, the result can be improved. This is because in the training stage, the numbers of positive samples and negative samples are different. We have more negative samples than positive sample. So it is reasonable that we use different C value. But in our first experiment, we don’t consider this, and the same C value is used for fair comparison.

In conclusion, a new feature for human detection is proposed in this paper. A histogram of pixels meeting different templates is used as a feature for the human detection. It integrates the texture information and the gradient information together, and shows more discriminative ability than only gradient based feature, even if the length of feature is shorter. Other advantage is that HOT feature is illumination-invariant, so the normalization is not the necessary step for detection, which is very useful for some hardware accelerators that only support integer calculation. In our experiment, the size of the sub window is fixed. It is expected that the variable window size and the boosting method will further improve the performance of the HOT feature. The computation of the HOT feature is parallel, so it is easy for hardware acceleration. Besides, integral image can also be used for computation. These factors make it possible for real-time application.

3 Extension of HOT Feature

3.1 Multi-scale Block HOT Feature

3.1.1 Definition

The main advantage of HOT feature is that this feature is extracted from the middle level, which contains all possible 3×3 pixel regions. See Figure 3-1. For HOG feature, after the orientation and magnitude of gradient are calculated for each pixel, it calculates the histogram by orientation voting. The basic unit for voting is pixel. So it is can be regarded that HOG feature is extracted from pixel level in the feature extraction. HOT feature computer values for the middle level. The basic unit of this level is 3×3 region. So, HOT feature is more macrostructures than the feature extracted from pixel level.

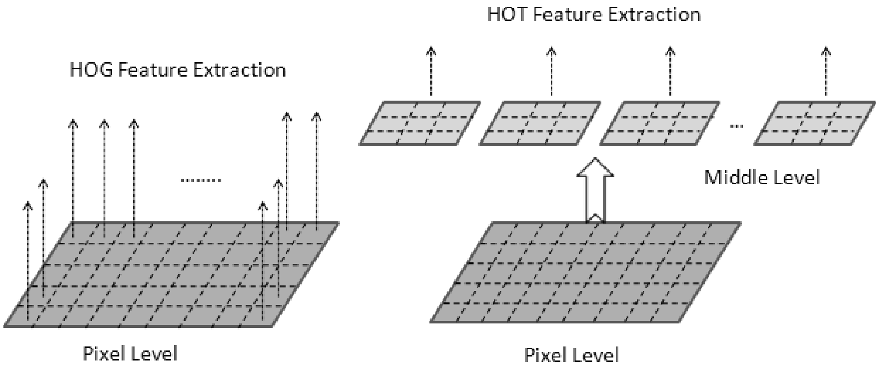


Figure 3-1 Feature extraction from different levels

Multi-scale block histogram of template (MB-HOT) feature is developed from HOT feature by extending the middle level. For HOT feature, the middle level only contains 3×3 pixel region. It is extended to 3×3 block region for MB-HOT feature. A block contains $s \times s$ pixels. $s = \{1,2,3...\}$. See Figure 3-2. The template is defined in a 3×3 block region. The value of each block is the average value of all pixels in this block. So in the middle level, it not only contains 3×3 pixels regions, but also 6×6, 9×9, 12×12… pixels regions. It is more macrostructures than the original HOT feature. In the feature extraction, we calculate the feature for different s respectively. When $s = i$, the feature can be represented as f_i. The final feature is $F = \{f_1, f_2, ...f_i...\}$.

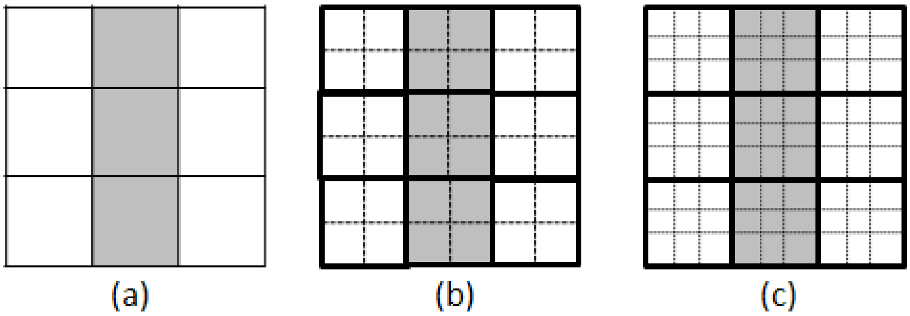


Figure 3-2 One template used in MB-HOT feature. s =1 for (a), s =2 for (b) and s =3 for (c). The value of each block the average value of all pixels in this block

Four formulas and 8 templates are used here for feature extraction.

Figure 3-3 shows that all pixels meeting the formula Eq2.9 when the upper left templates in Figure 2-4 are used.

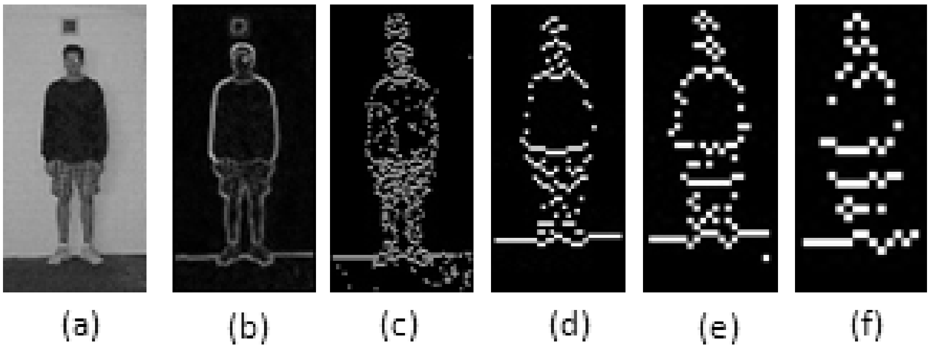


Figure 3-3 (a) original image; (b) gradient image; (c) for formula (3), all pixels meeting the upper left template in Figure 2-4, when s=1; (d) when s=2; (e) when s=3; (f) when s=4

The feature extracted from the different scale templates contains more shape information of human body than that only one scale is used. In Figure 3-4, we give the histogram of 3×3 regions meeting 8 templates for 4 formulas. It contains 32 bins. Figure 3-4 (a), (b), (c), (d) show the histograms by using the different scale templates respectively.

3.1.2 Experiment Result

In our experiment, we compare our MB-HOT feature with HOG feature and HOT feature by using linear SVM.

The comparison is done on INRIA dataset [22]. It is widely used for human detection in still image. The database contains 1774 human annotations and 1671 person free image. This dataset is made up of the training dataset and the testing dataset. 1208 human annotations and 1218 non-human images are used in the training stage, and the left images for testing. For positive samples, the left-right reflections are also used. So, 2416 positive samples are used for the training. More detail can be seen in [22].

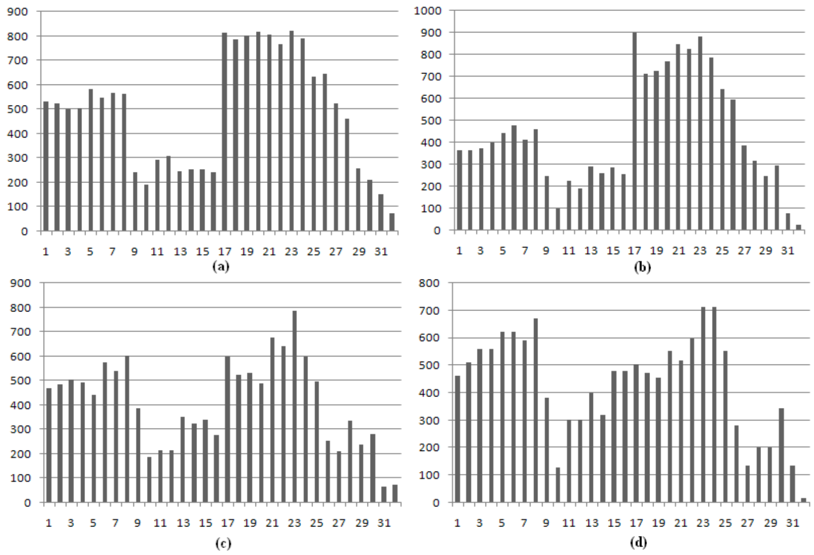


Figure 3-4 Histograms generated by using different scale templates. X axis has 32 bins and each bin corresponds to one template for each formula. Because 8 templates and 4 formulas are used, we have 32 bins. Y axis is the number of 3×3 regions that satisfy each template for each formula. They are extracted from Fig.4 (a). s =1 for (a), s =2 for (b), s =3 for (c), s =4 for (d). They show different statistically property

Re-sample strategy is used in our experiment. Re-sample strategy means that the positive samples and some negative samples random selected from natural negative samples in the training dataset are used for training first. The middle classifier is obtained. Then, this classifier is applied to the nature negative images and selects the hard negative samples. The initial samples and the hard negative samples are used for training the final classifier.

Different from the strategy in [1], color normalization is not used in our feature. Using RGB is report to improve performance [1]. But we show that our MB-HOT outperform competitors without the color information, with which further improvement is expected.

3 (X-) scales are used in the experiment. Linear kernel SVM is used for training. The comparison result can be seen in Figure 3-5. The data of other features are copied from respective papers [1, 3, 23-24].

From Figure 3-5, it can be seen that the Multi-scale HOT outperforms HOT[24], HOG[1] and Multi-resolution HOG[23]. The performance is nearly the same but a little better than COV[3]. Considering that the COV feature uses the different training method and the variable sub window strategy which can improve the result efficiently, and it has the high computation complexity, we can say that our feature is better than COV feature.

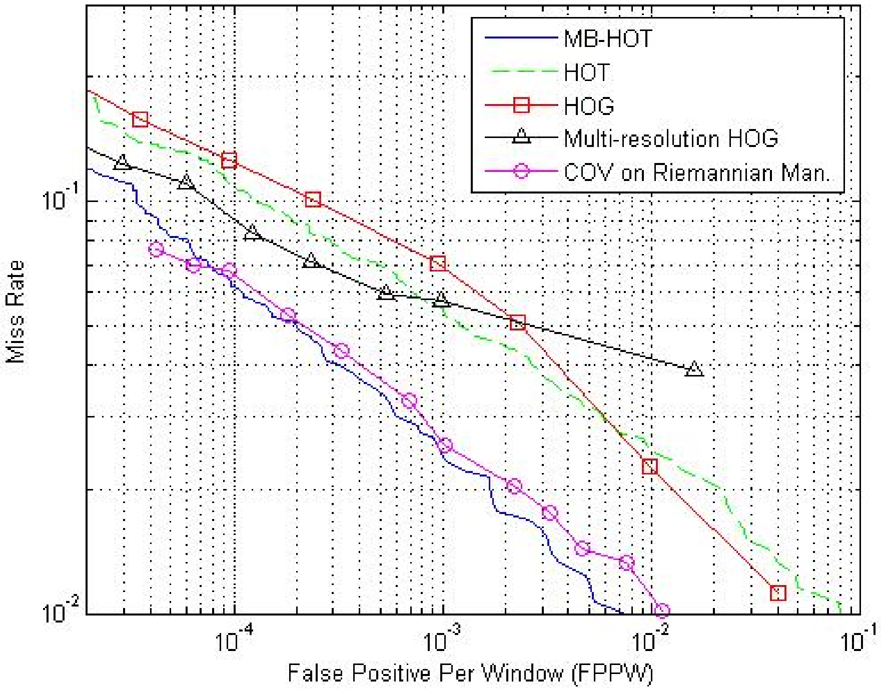


Figure 3-5 Comparison with other features. The curves are copied from respective papers

3.2 Motion Based Detector

3.2.1 Definition

The motion of human is different from the motion caused by other objects. Some detectors [25-27] are constructed by using the motion information. The combination of appearance based detector and motion based detector can detect the standing people and moving people efficiently from videos [27].

In order to extract the motion information, the optical flow is first calculated by current frame and last frame. Then the feature is extracted from the optical flow field. The optical flow value is two-dimensional. So the formulas used here is different from the histogram of template feature. Besides, the motion based feature is focus on capturing the relative moment of human body, because the global motion boundary information can be extracted by the appearance based feature. Another reason is that capturing the relative motion can reduce the optical flow caused by the motion of camera.

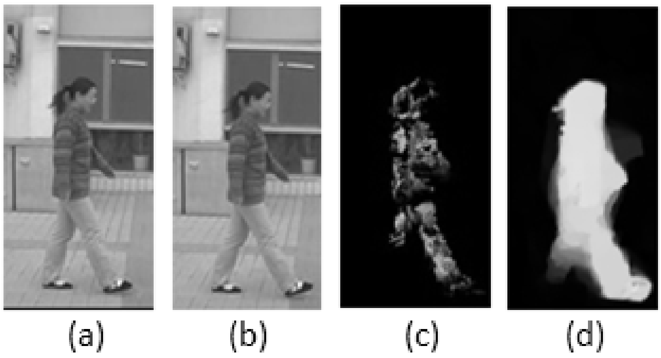


Figure 3-6 (a) Last frame; (b) Current frame; (c) Optical flow by [28]; (d) Optical flow by [29]

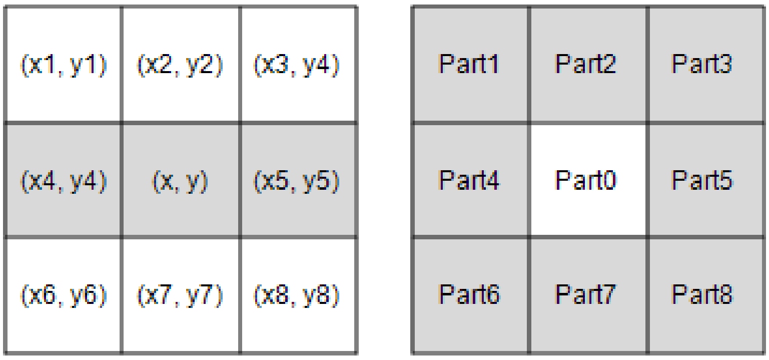
First we focus on how to calculate the dense optical flow, which is a popular method to estimate the motion. Optical flow calculation is a pixel correspondence problem. Given a pixel in the first image, we want to look for a nearby pixel in the second frame. It is supposed that the brightness is constant and the motion is small. So the displacement vector can be obtained. But if there is a great displacement and the changing illumination, the result is not good. The method in [28, 30] is used in our approach. The result can be seen in Figure 3-6(c).

In the feature extraction, our feature is based on the histogram of templates feature, but gives some modifications. First is about the dimension problem. Optical flow value is two-dimensional, different from gray image. So the definition of formulas is changed. Take formula Eq2.7 for example. The formula Eq2.7 is replaced by formula Eq3.1. The same improvement is applied to formulas Eq2.8, Eq2.9 and Eq2.11, which make our feature suitable to deal with 2 dimensional values.

$$F(T)=\begin{cases}1 \; if \; I_x(P)>I_x(P1) \; and \; I_x(P)>I_x(P2) \\ \quad and \; I_y(P)>I_y(P1) \; and \; I_y(P)>I_y(P2) \\ 0 \; other \; wise\end{cases} \qquad \text{Equation 3.1}$$

Where $I_x(P)$ denotes the x (horizontal) component of optical flow of pixel P, and $I_y(P)$ is the y (vertical) component of optical flow.

The second improvement is about the relative motion of human body. Because the appearance based feature has already captured enough motion boundary information, the combination of appearance based feature and motion boundary based feature can’t improve the detection result efficiently. So our motion based feature focus on local motion of human, such as the motion of legs and arms. In order to get relative motion, the cell is divided into 9 parts. See Figure 3-7. The optical flow value of pixel in 8 outer parts is subtracted by value of corresponding pixel in central part. And the histogram of template feature is extracted from 8 outer parts. The motivation is that if the person's limb width is approximately the same size as the part size, it can capture relative displacements of limbs to the background and nearby limbs [27]. Another reason is that if the camera is moved smoothly, the subtraction operation can reduce the effect caused by the moving camera.



(x1, y1)	(x2, y2)	(x3, y4)
(x4, y4)	(x, y)	(x5, y5)
(x6, y6)	(x7, y7)	(x8, y8)

Part1	Part2	Part3
Part4	Part0	Part5
Part6	Part7	Part8

Figure 3-7 The pixel value in optical flow field and the structure of cell

After we get the motion based detector, we could combine the MB-HOT feature and motion based feature together to get more accurate detection result in video. The workflow can be seen in Figure 3-8.

3.2.2 Experiment Result

In our second experiment, we compare our motion based feature with other features in optical flow field.

The Chinese academy of sciences (CAS) dataset [31] is selected for computing the positive samples. This dataset is also used in [26] to get the positive samples. It contains six different subsets of 12 people walking in six different directions. See Figure 3-9

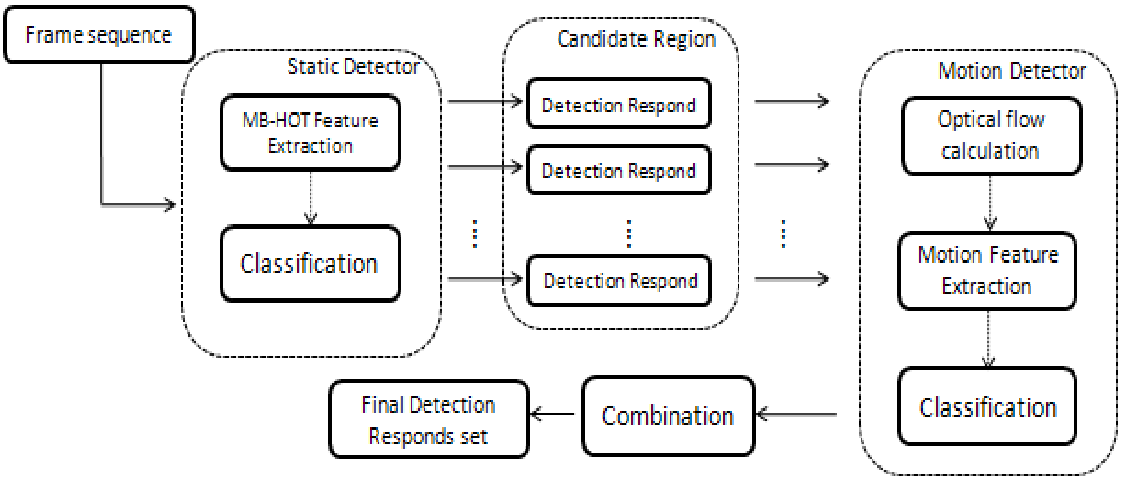


Figure 3-8 The workflow of our method

The regions containing human is manually marked and extracted. The corresponding optical flow is calculated by the current frame and the last frames. The optical flow method in [28, 30] is used. The positive samples are divided into the training dataset and the testing dataset. The samples in testing dataset are not included in training dataset.

Some videos without human are also selected as negative samples. We manually mark the region containing other objects but not human. The optical flow is extracted. Some examples can be seen in Figure 3-10.

At last, 3000 positive samples and 3000 negative samples are selected for training, which are scaled into 64×128. 1000 positive samples and 1000 negative samples are selected for testing, which are selected from different video sequences. Some other nature videos are also prepared for testing. See Figure 3-11

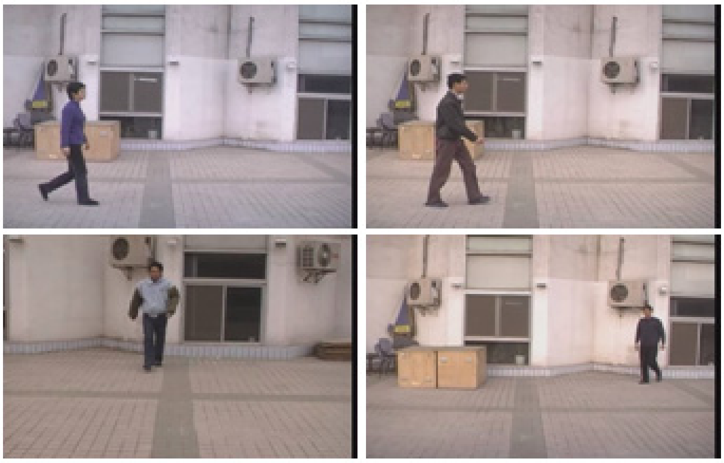


Figure 3-9 Some samples in CAS dataset

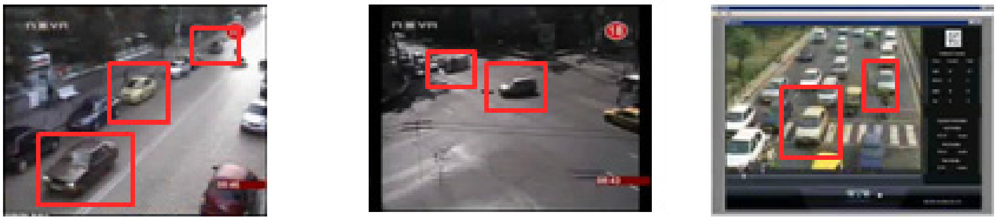


Figure 3-10 Some negative samples

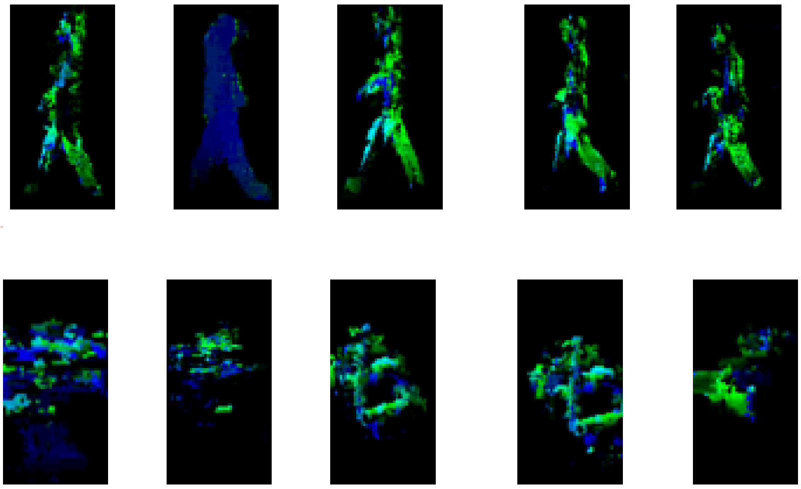


Figure 3-11 Some positive samples and negative samples in optical flow field

We compare our feature with [25-27]. In [25] optical flow (OF) is used as feature directly. In [26] Karhunen-Loeve transform (KLT) coefficient is taken as feature. In [27], intra motion histogram central difference (IMHCD) feature which is developed from HOG feature is used to code motion information. We implement these three methods and test them in our dataset. The comparison can be seen in Figure 3-12. It can be seen that our feature outperform the other motion based features.

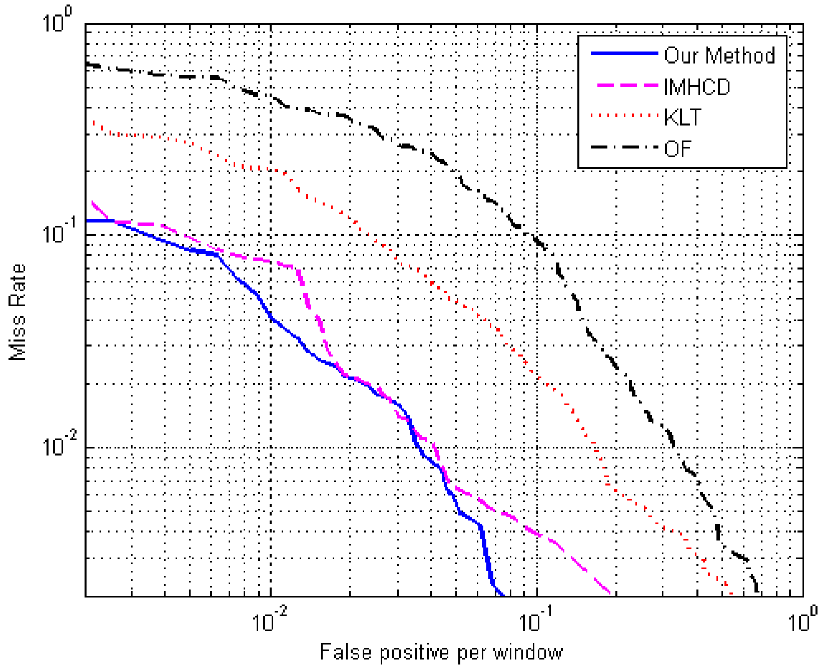


Figure 3-12 Comparison with IMHCD, OF and KLT feature

In second experiment, we test our method on 6 video sequences. 3 sequences are selected from CAS dataset, and we record other three sequences. They are all not used in the training stage. The cameras are fixed or move slightly. The detection result can be seen in Figure 3-13. We change the threshold of detector to get the different points in ROC curves. Some false responds can be removed by motion based detector from the responds obtained by appearance detector. In our implementation, we only use the very simple model to combine the result of the static and motion detector. If the probability obtained by static detector is greater than a pre-defined threshold or the sum of probabilities obtained by static detector and motion detector is greater than another pre-defined threshold, they all would be considered as the positive samples. In this case, some standing people with low probability values returned by static detector would be missed. The more complicated probability model will be developed to improve the performance. The frame coherence will be considered.

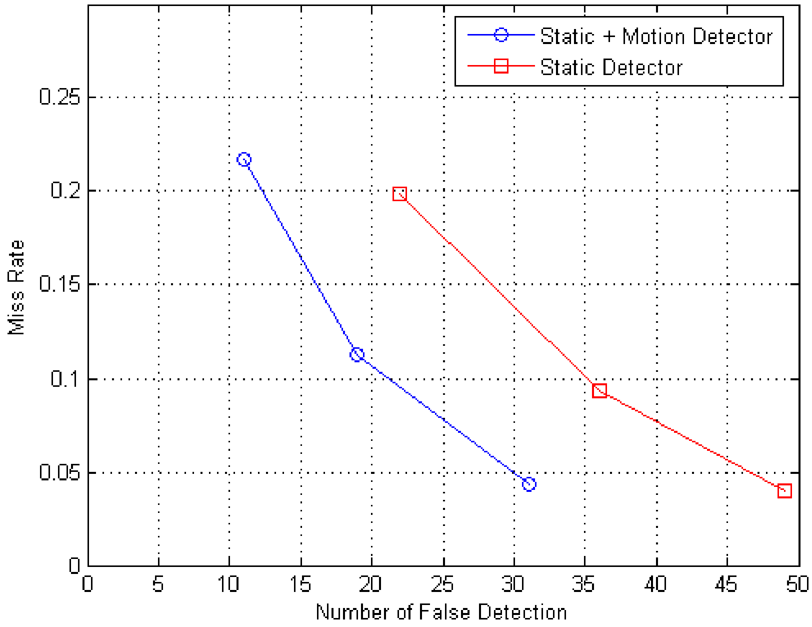


Figure 3-13 Comparison of static detector and static & motion detector

Some detection results can be seen in Figure 3-14.

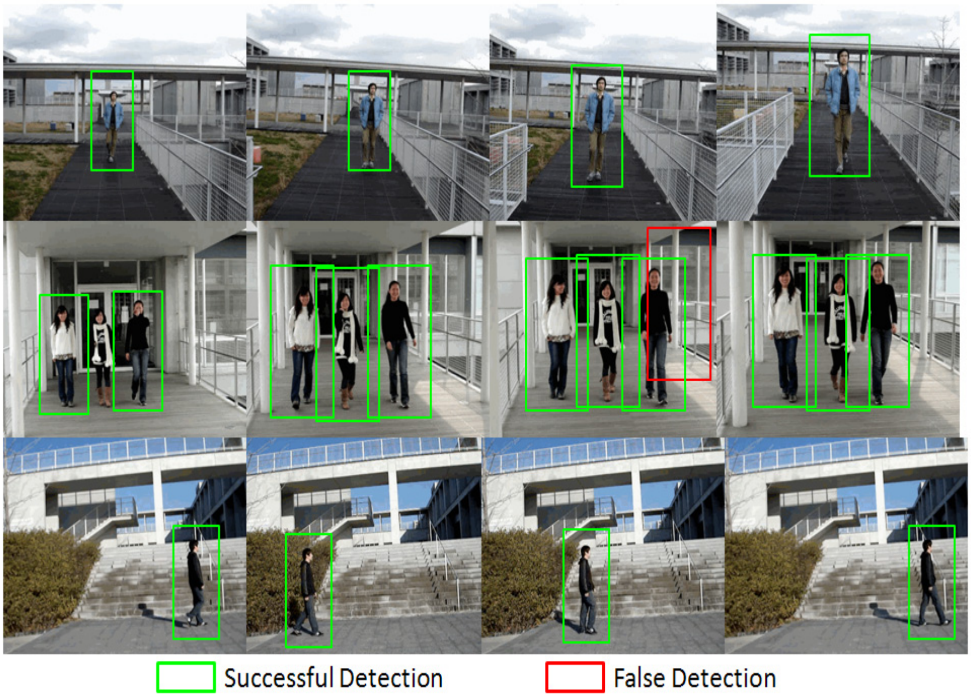


Figure 3-14 Some detection results of video sequences

Mention that in the appearance based detectors, because the sliding window strategy is used, for one human, there may be more than one responds. This can be seen in Figure 3-15.

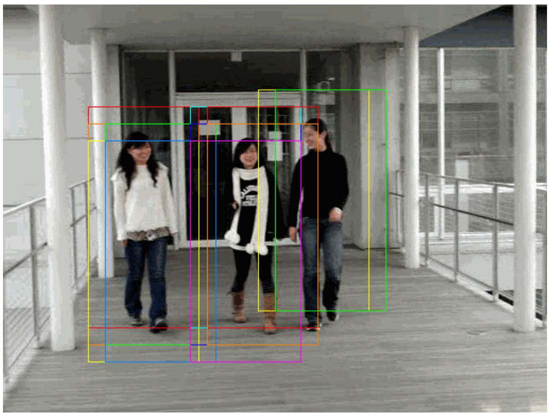


Figure 3-15 More than one responds for one human

We have to find the responds associated with the same person. Mean shift clustering method is used in our experiment.

Mean shift considers feature space as an empirical probability density function. If the input is a set of points then mean shift considers them as sampled from the underlying probability density function. If the dense regions are present in the feature space, then they correspond to the model of the probability density function.

For each data point, mean shift associates it with the nearby peak of the dataset's probability. Eq.3.2 is the definition of the mean shift vector.

$$M_h\left(x\right) \equiv \frac{\sum_{i=1}^{n} K(\frac{x_i - x}{h}) w(x_i)(x_i - x)}{\sum_{i=1}^{n} K(\frac{x_i - x}{h}) w(x_i)} \qquad \text{Equation 3.2}$$

The mean shift algorithm can be specified as follow: first, we fix a window around each data point; second, compute the mean of data within the window; thirdly, shift the window to the mean and repeat till convergence. Mean shift treats the points the feature space as a probability density function. Dense regions in feature space correspond to local maxima or modes. So for each data point, we perform gradient ascent on the local estimated density until convergence. The stationary points obtained via gradient ascent represent the modes of the density function. All points associated with the same stationary point belong to the same cluster.

For the application of human detection, the input is (x, y, s). (x, y) is the center of the detection respond, and s is the scale. For each respond, we will calculate the cluster center. Finally, the output is (x, y, s, n). n is the number of the responds associated with this cluster center.

Some experiment result about this merge strategy can be seen in Figure 3-16.

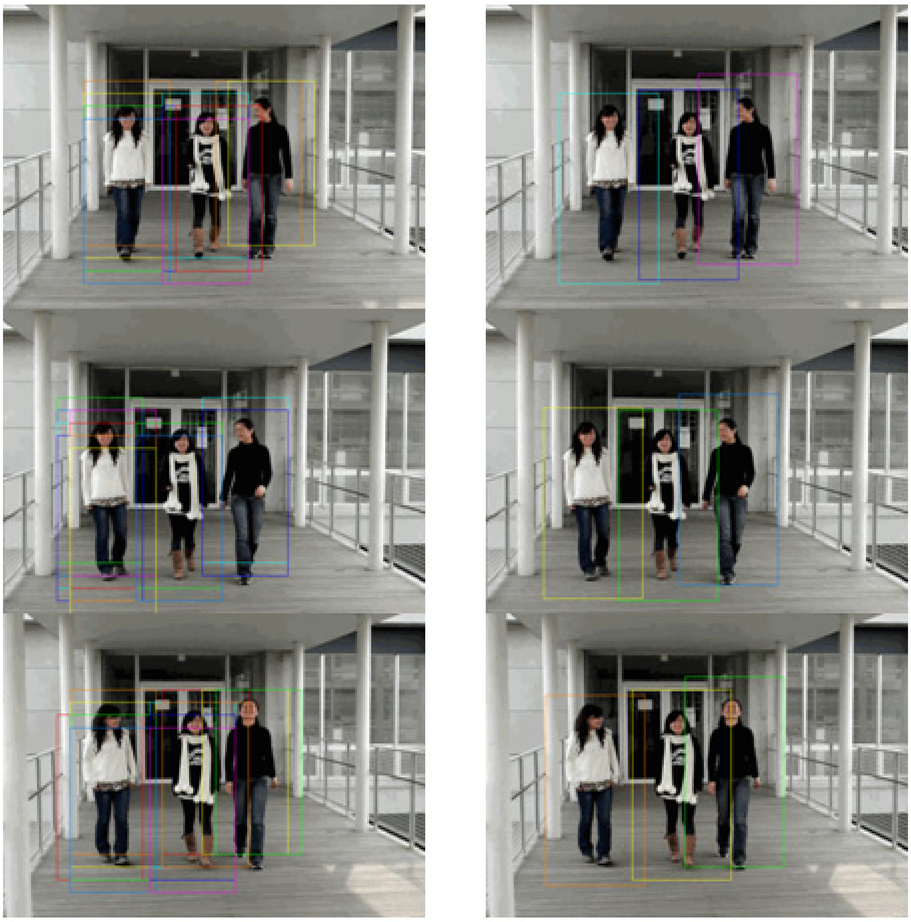


Figure 3-16 Experiment results of the merge strategy

4 Acceleration Methods

4.1 Background and Related Work

Up to now, the computation complexity is still the bottleneck for many applications. Because in most case, sliding window is used as searching strategy, thousand of sub windows with different scales should be detected for one frame. Although for some applications when camera is fixed, background subtraction method can be used to reduce the detection area, when camera is moving, we have to detect all possible positions and scales. The huge computation makes it difficult for real time requirement. Some tracking systems [32-33] are developed on assumption that all human have already been detected, and the detection process is done offline. So real time human detector is necessary for practical applications. There are some software based acceleration methods. For some features which are developed from the histogram, the integral image can be used to accelerate the histogram calculation. Cascade-Rejection structure is used in [2-3, 11]. Only positive samples have to pass all the cascades. For intelligent vehicles, some road subtraction and clustering methods can be used to estimate the position and scale of pedestrian [34]. This reduces detection windows more or less, but not sufficient for some real-time applications.

Recently, Graphics process unit (GPU) shows high computation ability, especially for parallel calculation. The concept of General Purpose GPU is proposed, focusing on solve the parallel computation problem by using graphics chip. GPU is not dedicated to computer graphics applications, but also widely used in image processing and computer vision. Some GPU based implementations of HOG feature [35-36] are proposed to solve the computation problem. The GPU based implementation for feature extraction and classification can obtain high speed for detection. The programming framework of GPU can be seen in Figure 4-1. The CPU could save the image in the system memory. In the detection stage, this image is first loaded into the video memory. The pixel processors can access video memory and perform some operations. The pixel processor is programmable, and we could change the value of pixel, change output position and access other pixels.

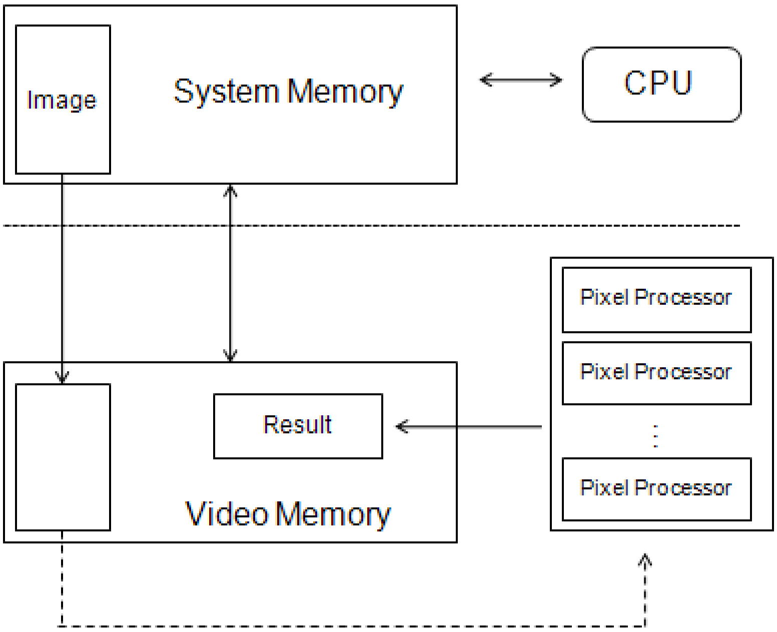


Figure 4-1 Programming framework of GPU

New computation architecture of GPU can conceal the detail of rendering pipeline and user can focus on the parallel algorithm design and efficient memory access.

4.2 Integral Image

In [11], the concept of the integral image is proposed to accelerate the feature extraction for Haar-like feature. This image can make the computation of the rectangular features be done in a constant time, therefore enable the system to operate fast.

First, I give the definition of the integral image. In the integral image, the value at location (x, y) is the sum of the pixel value about and to the left of (x, y) inclusive. See Figure 4-2

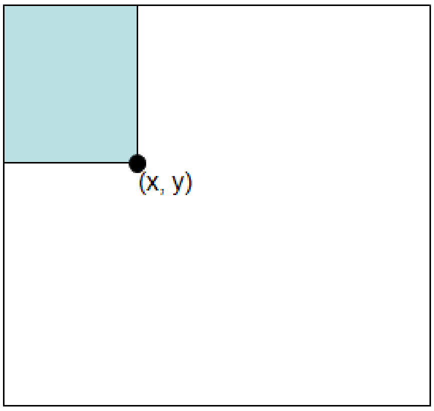


Figure 4-2 The definition of integral image

By using the following two recurrences, where $i(x,y)$ is the pixel value of original images at the given location and $v(x,y)$ is the cumulative column sum. We can calculate the integral image representation of the image in a single.

$$\begin{aligned} v(x,y) &= v(x,y-1) + i(x,y) \\ ii(x,y) &= ii(x-1,y) + v(x,y) \end{aligned} \qquad \text{Equation 4.1}$$

Given the integral image, the sum of pixel values within a rectangular region of the image aligned with the coordinate axes can be computed with four array references in constant time. For example, see Figure 4-3. We want to calculate the sum value of pixels in region S. This value can be obtained by using the following formula:

$$S = ii(L4) + ii(L1) - ii(L2) - ii(L3) \qquad \text{Equation 4.2}$$

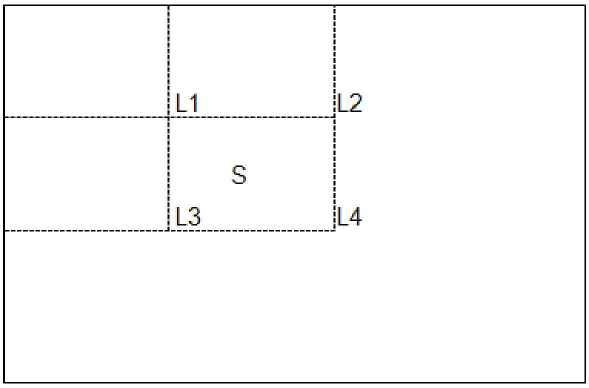


Figure 4-3 Rapid calculation of rectangular feature using integral image

By introducing the concept of integral image, we can obtain various size of orientation of rectangular feature in an image. Now, we focus on how to use integral image to accelerate the feature extraction for HOT feature.

Because we use four formulas and eight templates for feature calculation, it can be considered as that we calculate 32 values for each pixel in a cell, and we want to get the histogram for each type of value. In this way, we could use 32 additional images for acceleration. When we calculate a value for one pixel, we store the value in corresponding additional image. At last, we calculate the integral images for all the additional images. And if we want to get the 32 bin histogram for a cell, it can obtained by the 32 additional integral images. See Figure 4-4.

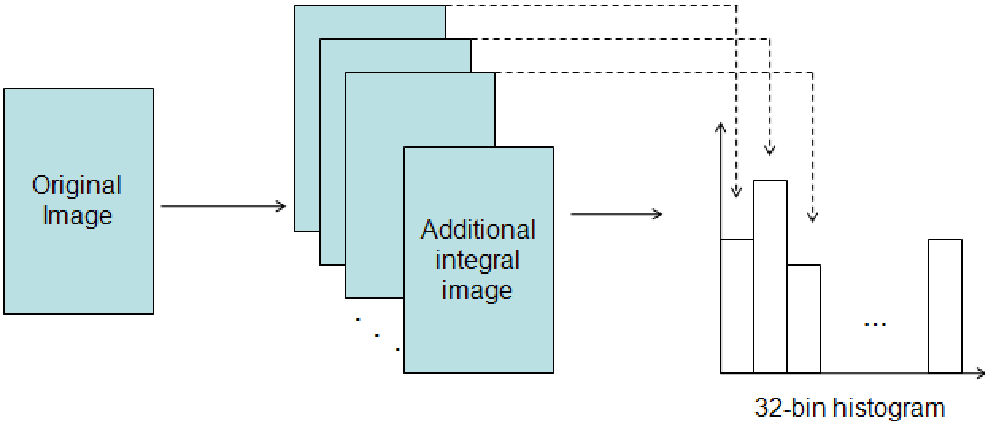


Figure 4-4 Integral images for HOT calculation

From the Table 4-1, we can see the calculation time for integral image based acceleration method.

Table 4-1 Calculation Time

Image Size	320×240	640 ×480
CPU: Normal	156 s	622 s
CPU: Integral Image	9.062 s	63.875 s

4.3 GPU Based Acceleration

GPU shows the high parallel computation ability. Especially after the CUDA computation framework is proposed by NVIDIA. GTX285 and CUDA are used in our method to get a real time detector.

In CUDA, the parallel calculation can be achieved by using the thread block. The structure of the thread block can be seen in Figure 4-5. CUDA threads may access data from multiple memory spaces during their execution. Each thread has private local memory. Each thread block has a shared memory visible to all threads of the block and with the same lifetime as the block. Finally, all threads have access to the same global memory. There are also two additional read-only memory spaces accessible by all threads: the constant and texture memory spaces. The global, constant and texture memory space are optimized for different memory usages. Texture memory also offers different addressing models, as well as data filtering, for some specific data formats. The memory hierarchy can be seen in Figure 4-6.

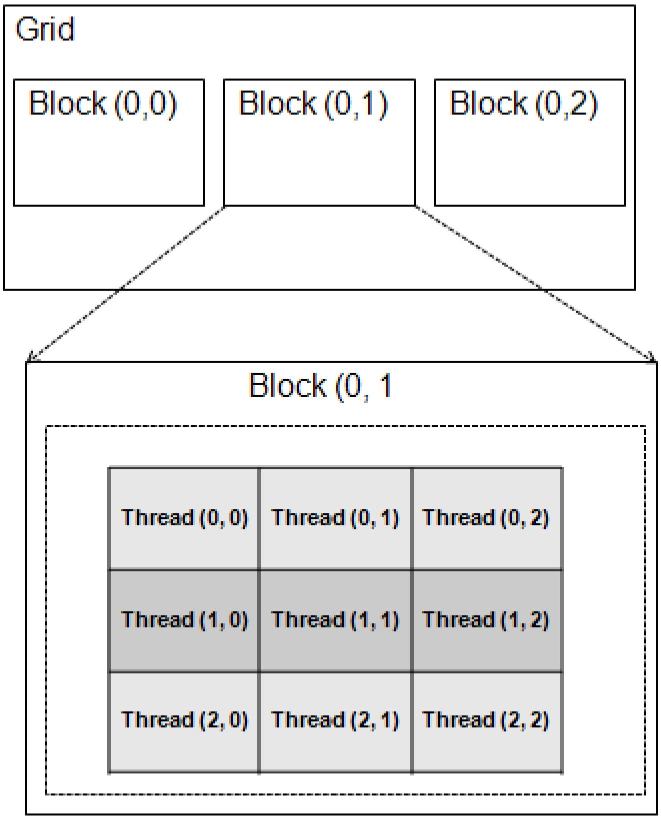


Figure 4-5 Thread block

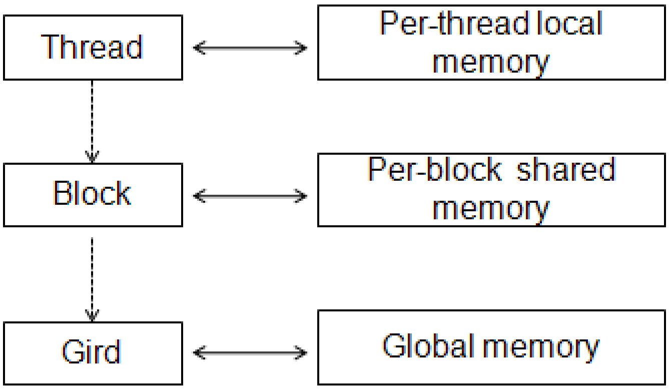


Figure 4-6 Memory Hierarchy

The flow of GPU based implementation can be seen in Figure 4-7. In the offline stage, we train the classifier, which contains a set of support vectors and weigh values. Before the detection stage, they are loaded into the video memory. In the online detection stage, the image is first loaded into the video memory, and we set up the scale pyramid for this image. Then we calculate the magnitude values for each pixel in the pyramid. Then we calculate the status table. We compute 32 values for each pixel, and store them in the status table. After that we could compute the histogram table, and the features for all detection windows can be extracted. The classification result can be obtained by using support vectors and weight values.

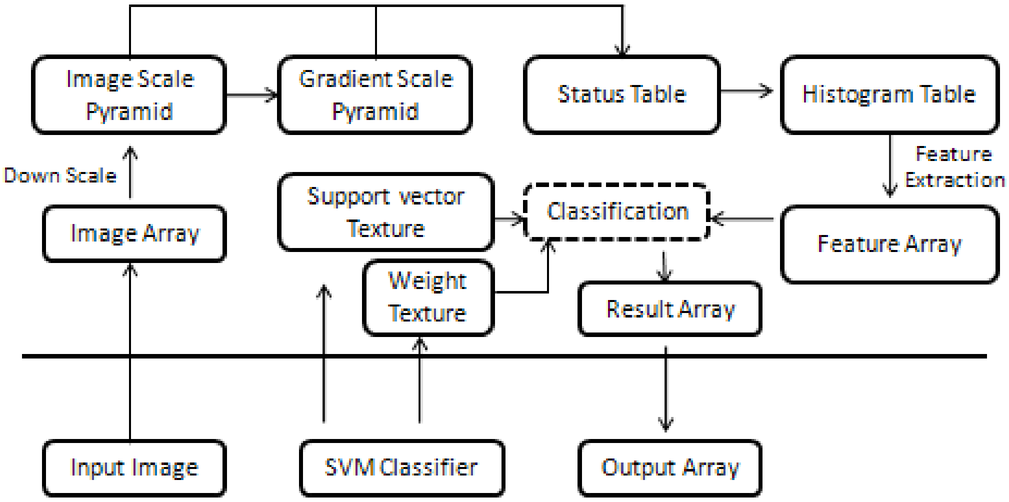


Figure 4-7 Workflow of GPU based MB-HOT detector

Down Scale: in order to detect the human varying in size, a scale pyramid is built. We first scale the image, because the detection window in our implementation is fixed. We change the size of the inputting image in order to detect the human varying in size. This is the Y scale space. For each image in the Y scale pyramid, we are gonging to scale it for feature extraction. This is the X scale space. See Figure 4-8. Linear interpolation is used. A thread block contains 16×16 threads and one thread corresponds to one pixel in the scaled image.

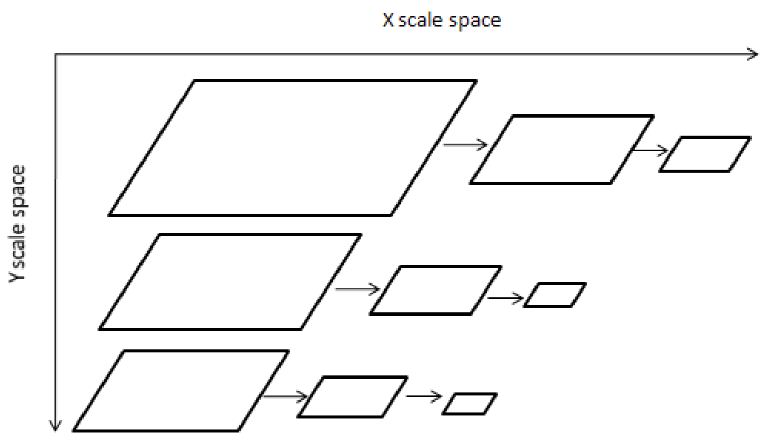


Figure 4-8 Scale pyramid. Y scale space is for detecting human varying in size. X scale space is for feature calculation

Gradient Calculation: for each pixel in scale pyramid, a gradient magnitude is calculated. A thread block contains 16×16 threads, to deal with a 16×16 pixel region. One thread is dedicated to get the magnitude value of one pixel. 18×18 pixels are copied to the shared memory for each thread block, to get the coalesced access.

Status Table Calculation: for each pixel in scale pyramid, a status table is calculated. Status table is a 32 bit value. See Figure 4-9. Each bit corresponds to one template for different formulas. If this pixel meets one template, the value of corresponding bit is 1; otherwise 0. The design of this part is similar to gradient calculation. For a thread block, 18×18 gray values and 18×18 magnitude values are copied to shared memory.

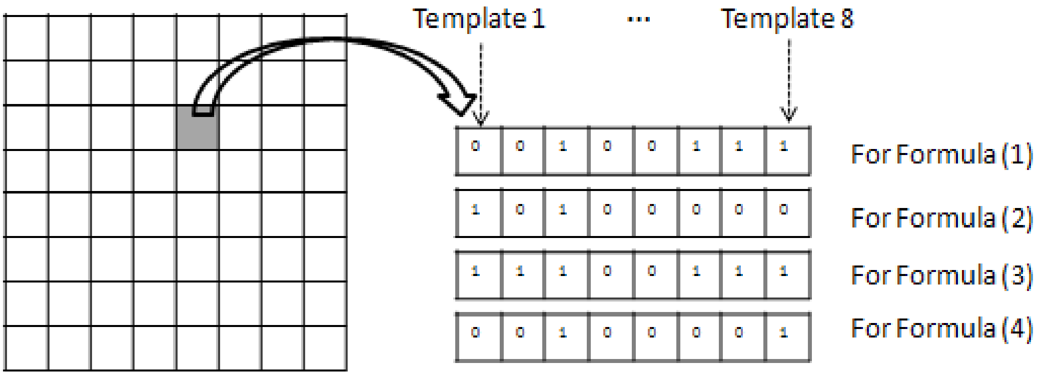


Figure 4-9 32 bit status table

Feature Calculation: calculate a feature for a 64×128 detection window. It contains 7×15 cells in the first level of X scale space, 3×7 cells in the second level and so on. A cell contains 16×16 pixels. The stride between 2 cells is 8 pixels. A thread block contains 32 threads. It calculates a 32 dimensional vector for a 16×16 pixels region (a cell). A thread is used to compute one value of this 32-d vector. Four histograms are copied from the histogram table. Because the histogram in the last step is calculated from an 8×8 region, 4 histograms are added together to get a histogram of 16×16 region. The features of all cells in different scale levels are combined together after normalization, as the feature of this detection window.

Linear SVM: A classifier is trained offline by using LIBSVM [21]. Classifier is a set of support vectors. These support vectors are stored in texture memory before detection process. The method in [35] is used. Each thread block is responsible for one detection window and each thread computes weighted sums corresponding to each column of the window.

For motion based detector, the GPU based implementation is similar to MB-HOT method. The difference is that we only detect the responds returned by

MB-HOT feature. The corresponding regions in the current frame and last frame are used to compute the optical flow. In our experiment, the optical flow of each frame is pre-calculated by using the method in [28, 30]. In [29], a GPU based optical flow calculation method is proposed. The result can be seen in Figure 4-10. We will integrate it in our detection framework for future work. The division of labor for GPU based acceleration can be seen in Figure 4-11.

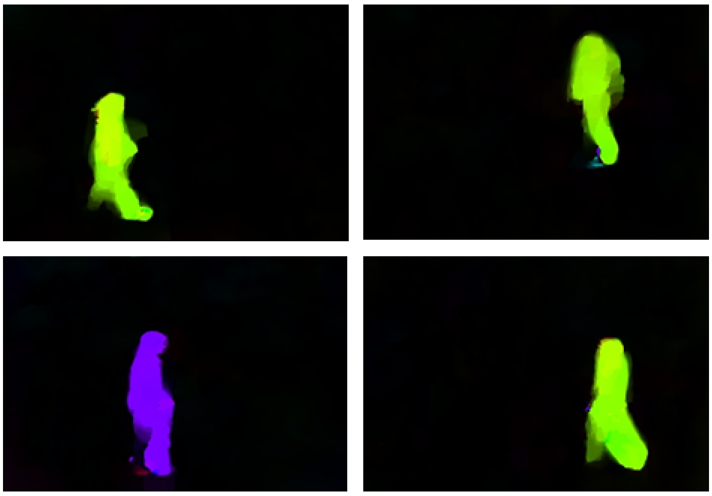


Figure 4-10 GPU based optical flow calculation

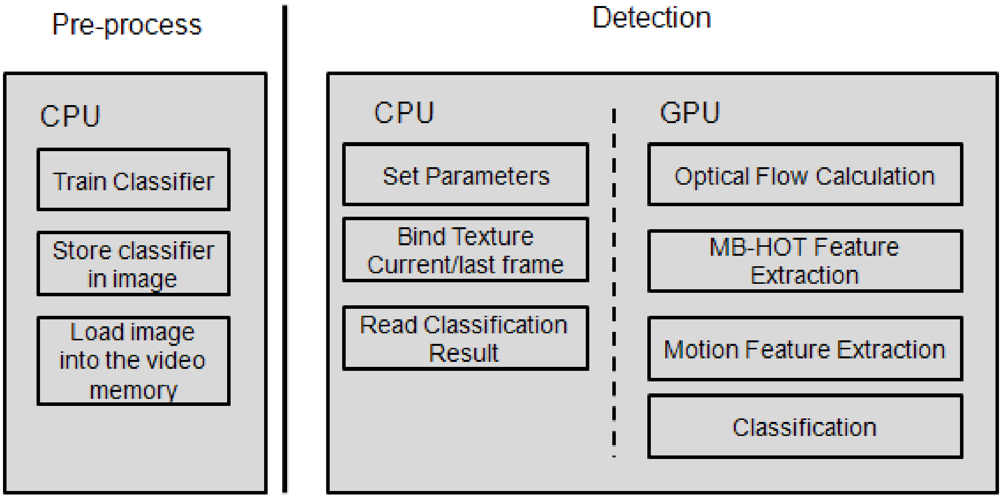

Pre-process
Detection
CPU
Train Classifier
Store classifier in image
Load image into the video memory
CPU
Set Parameters
Bind Texture Current/last frame
Read Classification Result
GPU
Optical Flow Calculation
MB-HOT Feature Extraction
Motion Feature Extraction
Classification

Figure 4-11 Division of labor for GPU based acceleration

In the experiment, the calculation time is evaluated. The experiment environment is as following: Intel Quad CPU is used. It has 4 cores, and the frequency for each core is 2.83GHz. System memory is 8 GB. GTX 285 is used for GPU calculation. Visual studio C++ 2005 and CUDA programming framework are used as developed tools.

For MB-HOT feature, 2 scales are used in X scale space for feature extraction and the factor is 0.5. 2 scales are used in Y scale space to detect human varying in size; the factor is 0.8. The size of detection window is 64×128. The stride between two detection windows is 8 pixels. LibSVM [21] is used to get the classifier. These support vectors are obtained offline and are loaded into the texture memory before the detection stage. The calculation time and memory consumption for each stage can be seen in Table 4-2 and Table 4-3. It can be seen that although the sliding window strategy is used, MB-HOT feature can meet the real time requirement.

Table 4-2 Calculation time consumption for each step (Millisecond)

Image Size	320×240	640×480
Down Scale	0.391	0.984
Gradient	0.06	0.16
Status Table	0.26	0.32
Histogram	0.225	0.46
Feature Calculation	6.2	31
Classification	2.1	16
Overall	9.236	48.924

Table 4-3 Calculation memory consumption for each step (Mega Byte)

Image Size	320×240	640×480
Down Scale	0.16	0.63
Gradient	0.16	0.63
Status Table	0.63	2.52
Histogram	0.31	1.26
Feature Calculation	11.61	83.32
Classification	27.06	27.06
Overall	39.93	147.1

5 Pose-invariant Human Feature

Pedestrian detectors make most errors on pedestrians in configurations that are uncommon in the training dataset. See Figure 5-1. An articulation-insensitive feature should be developed to cope with the large variability of human poses, for the robust human detection.

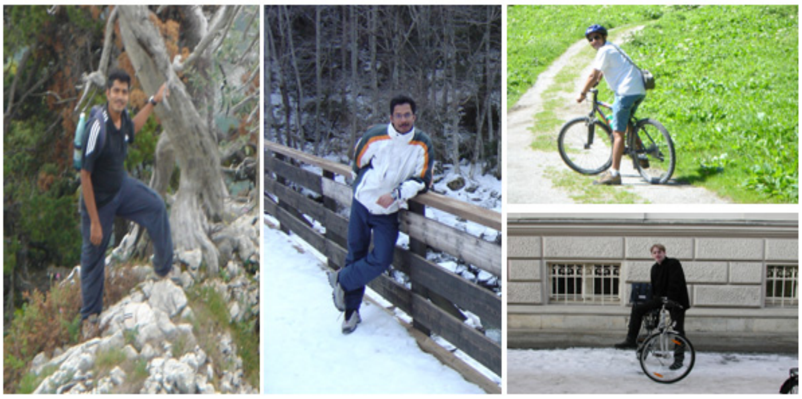


Figure 5-1 Variability of human poses in images. Shape information is used as the main discriminative cue in most pedestrian detector. The pose variety makes the shapes of human various, and worsens the difficulties for the detection

Pose estimation is an efficient solution for pose variant problem [37-39]. Before the detection stage, we could estimate the human poses by template matching, structure learning and so on. After the human body configurations are estimated, we can get the bounding boxes of body parts. We resize the bounding boxes and align the orientations of them to a canonical orientation. The features are extracted from these bounding boxes respectively and are combined together as the final feature, to achieve the pose-invariant effect for pedestrian detection. Due to the high performance of HOG feature, it is used in our detector as a low level feature. The key problem is that a fast and accurate pose estimation method should be developed.

For the regression problem, numerous of algorithms were proposed in the machine learning literature. The nonparametric kernel regression (NPR), the kernel ridge regression (KRP) and the support vector regression (SVR) are most popular date-driven regression method. The detail can be seen in [40].

In this chapter, a pose-invariant human detection method is proposed. Before detection stage, we first estimate the poses of humans, which are represented by a set of two dimensional points, and the features are extracted from the bounding boxes of different parts. In this way, the characteristic feature for each part can be obtained and the redundant information of background can be removed, which can improve the detection accuracy efficiently. The workflow of this feature can be seen in Figure 5-2. The pose estimation problem is defined as a multi-output regression problem. A new definition of loss function is given to find the mapping function, by which the images can be mapped into the pose space. Orientated gradient is mainly used in pose estimation, which reduces the number of sub windows which are used in boosting based regression method, compared to Haar-like feature. The HOG features are extracted from all parts of human configuration and are combined together as final feature. Experiment on INRIA dataset shows that pose estimation can increase the detection rate efficiently by correctly detecting the humans whose poses are different from ordinary.

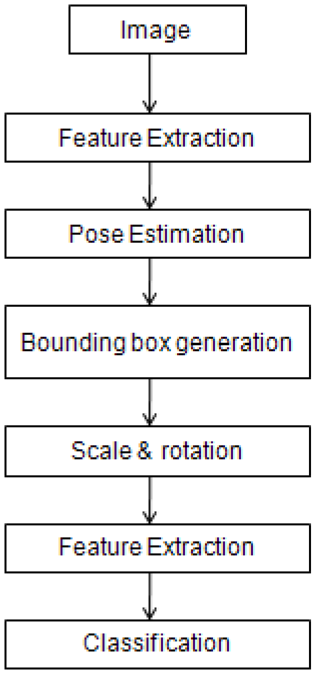


Figure 5-2 The workflow for pose-invariant feature

In this section the pose estimation problem is treated as a multi-output regression problem. Given a training set $\{\mathrm{y}_i, \mathrm{x}_i\}_1^N$ with inputs $\mathrm{x}_i \in \mathbb{R}^d$ and outputs $y_i \in \mathbb{R}^{2q}$, the goal is to find a function that maps x to y, such that the expected value of loss function is minimized.

The main contribution of this section is that the image based multi-output regression method is used to estimate the human pose for detection. We propose a new loss function which is more dedicated to pose estimation with purpose for detection, and prove that the new loss function is also suitable for boosting regression method. The HOG is used for multi-output regression, which reduce the number of sub windows we should consider in the training stage, compare to Haar-like feature.

The pose estimation can be modeled as a multi-output regression problem. Given a training set $\{\mathrm{y}_i, \mathrm{x}_i\}_1^N$ with inputs $\mathrm{x}_i \in \mathbb{R}^d$ and outputs $y_i \in \mathbb{R}^{2q}$, the goal is to find a function that maps x to y, such that the expected value of function $E_{x,y}[\psi(y, F(x))]$ is minimized. The L^2 loss function is usually used [41-42], which is defined as follow:

$$\psi(y, F(x)) = |y - F(x)|^T\,|y - F(x)| = \|y - F(x)\|^2 \qquad \text{Equation 5.1}$$

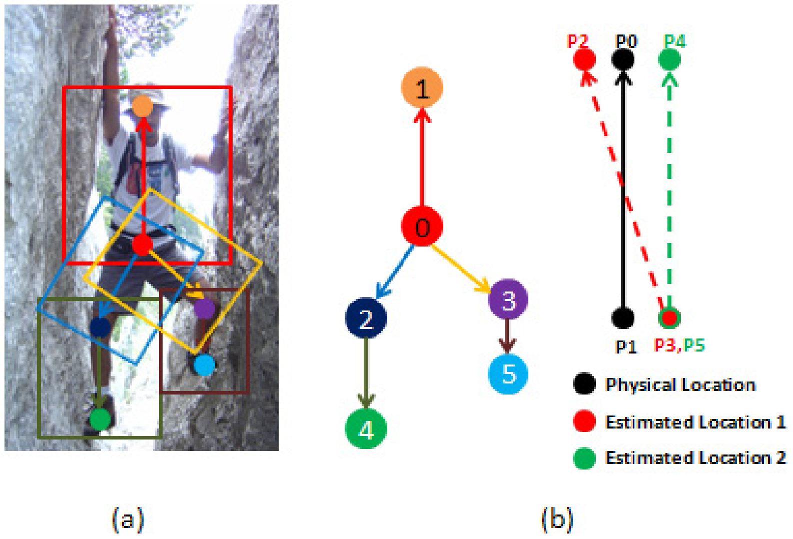


Figure 5-3 (a) Bounding boxes for parts. They are resized and rotated before feature extraction; (b) axle lines of bounding boxes are of importance for feature extraction. Loss function should represent the difference of axle lines. If the loss function in previous works is used, the estimated location 1 has the same value of location 2, but the location 2 is better because it has the same orientation with the physical location

The human poses actually consist of several points in two-dimensional space: $\{P_j\}_{j=1}^q$ where $P_j \in \mathbb{R}^2$. The above loss function considers the estimated pose as a $2q$-dimensional vector, and takes it as a whole. We can't get a nicer map function by using the above loss function in some cases, since sometimes the lines between two points are of importance for feature extraction. See Figure 5-3. The loss function should represent the difference between the physical axle lines and estimated axel lines. We define the new loss function as follow:

$$\psi = \sum_{j=1}^{j=q} \{ \| P_{root,j} - P'_{root,j} \|^2 + \alpha \| (P_j - P_{root,j}) - (P'_j - P'_{root,j}) \|^2 \} \qquad \text{Equation 5.2}$$

P_j is the j th point in physical human pose and $P_{root,j}$ is the root of this point. $(P_{root,j}, P_j)$ is the axel line. P'_j is the estimation of P_j. We use ψ to evaluate the difference between the physical and estimated axle lines.

The mapping function $F(x)$ can be found by minimizing $J(F(x))$.

$$J(F(x)) = \sum_{i=1}^{N} \sum_{j=1}^{q} \{ \| y_{i,j}^{root} - F_j^{root}(x_i) \|^2 + \alpha \| (y_{i,j} - y_{i,j}^{root}) - (F_j(x_i) - F_j^{root}(x_i)) \|^2 \}$$

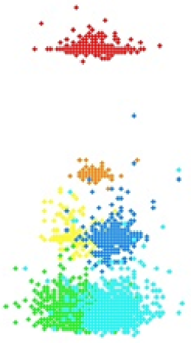
$y_{i,j}$ is the j th point of the i th training image, and $y_{i,j}^{root}$ is the root point of $y_{i,j}$. $F_j(x_i)$ is the estimated point of $y_{i,j}$ and $F_j^{root}(x_i)$ is the corresponding estimated root point.

We suppose that the point on the ass is the root point for all the other points and $y_{i,1}$ is the root point of y_i, so the $J(F(x))$ can be rewritten as:

$$\begin{aligned}
J(F(x)) &= \sum_{i=1}^{N} \sum_{j=2}^{q} \{ \| y_{i,1} - g(x_i) \|^2 + \alpha \| (y_{i,j} - y_{i,1}) - (F_j(x_i) - g(x_i)) \|^2 \} \\
&= \sum_{i=1}^{N} \sum_{j=2}^{q} \| y_{i,1} - g(x_i) \|^2 + \sum_{i=1}^{N} \sum_{j=2}^{q} \alpha \| (y_{i,j} - y_{i,1}) - (F_j(x_i) - g(x_i)) \|^2 \\
&= q \sum_{i=1}^{N} \| y_{i,1} - g(x_i) \|^2 + \alpha \sum_{i=1}^{N} \sum_{j=2}^{q} \| (y_{i,j} - y_{i,1}) - (F_j(x_i) - g(x_i)) \|^2 \\
&= q \sum_{i=1}^{N} \| y_{i,1} - g(x_i) \|^2 + \alpha \sum_{i=1}^{N} \sum_{j=2}^{q} \| u_{i,j} - k_j(x_i) \|^2 \\
&= q \sum_{i=1}^{N} \| y_{i,1} - g(x_i) \|^2 + \alpha \sum_{i=1}^{N} \| u_i - k\ (x_i) \|^2 \\
&= M(k(x))
\end{aligned}$$

Where $k_j(x_i)=F_j(x_i)-g(x_i)$, $u_{i,j}=y_{i,j}-y_{i,1}$ and $g(x_i)$ is the estimated root point of x_i. $F(x)$ can be obtained by using boosting method to calculate $k\ (x)$ and $g\ (x)$.

For human pose estimation, we suppose that the point on the ass is the root point for all the other points. We select the point on ass as the root node as the following reasons: firstly the ass node is in the central of human body, so it is easily to generate bounding box for each part; secondly the error of estimated location of root node should be minimized for more accurate human detection, and the ass point is more easily located than the other points. See Figure 5-4.



	E	S^2
● head	(32.46, 23.84)	18.19
● ass	(32.16, 63.62)	2.87
● left knee	(28.00, 83.49)	15.09
● left foot	(27.70, 106.00)	43.28
● right knee	(37.79, 63.51)	21.00
● right foot	(38.24, 105.86)	54.62

Figure 5-4 Space distribution of human parts; we mark more than 800 images in the INRIA training dataset

In order to compute $g\ (x)$, the SVR and PCA method are used to locate the root point. The algorithm flow can be seen in Figure 5-5.

Input: $\{\mathrm{y}_{\mathrm{i}}, \mathrm{x}_{\mathrm{i}}\}_1^{\mathrm{N}}, y_i \in \mathbb{R}^{2q}, x_i \in \mathbb{R}^d$

- $r_i = p(y_{i,1}) : \mathbb{R}^2 \to \mathbb{R}^1$, by using PCA method;
- Find suitable w by minimizing:

$$\frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N}\left|r_i - g^{'}(x_i)\right|_{\xi}$$

Where $g^{'}(x) = \sum_{n=1}^{N} w_n k(x;x_n)$

And $k(x;x_n)$ is reproducing kernel function.

Output: $g(x) = p^{-1}(g^{'}(x)) : \mathbb{R}^d \to \mathbb{R}^2$

Figure 5-5 Algorithm for computing $g(x)$

After obtaining the $g\ (x)$, we focus on compute $k\ (x)$. The boosting method can be used to calculate it.

If boosting method is used, this function can be assumed to take a linear form:

$$k\ (x) : \mathbb{R}^d \to \mathbb{R}^{2q-2} = \sum_{t=1}^{T} \lambda_t h_t(x) \qquad \text{Equation 5.3}$$

In which $h(x)$ is a weak function.

The final function is approximated by adding a new weak function iteratively.

$$k^{'}(x) = k(x) + \lambda h(x) \qquad \text{Equation 5.4}$$

At each round, we select suitable λ and h by following formula:

$$(\lambda^{'}, h^{'}) = \arg \min_{\lambda, h} M(k{+}\lambda h) \qquad \text{Equation 5.5}$$

$M(k^{'})$ can be computed as:

$$\begin{aligned} M(k^{'}) &= q\sum_{i=1}^{N}\left\|y_{i,1}-g(x_i)\right\|^2 + \alpha\sum_{i=1}^{N}\left\|u_i - k^{'}(x_i)\right\|^2 \\ &= q\sum_{i=1}^{N}\left\|y_{i,1}-g(x_i)\right\|^2 + \alpha\sum_{i=1}^{N}\left\|u_i - k(x_i) - \lambda h(x_i)\right\|^2 \\ &= M(k) + \alpha\lambda^2\sum_{i=1}^{N}\left\|h(x_i)\right\|^2 - 2\alpha\lambda\sum_{i=1}^{N}(u_i - k(x_i))(h(x_i))^{\mathrm{T}} \end{aligned}$$

In order to get minimum value of $M(k^{'})$, the λ can be calculated by $\partial(M(k^{'}))/\partial(\lambda) = 0$.

$$\lambda^{'} = \frac{\sum_{i=1}^{N}(u_i - k(x_i))(h(x_i))^{\mathrm{T}}}{\sum_{i=1}^{N}\left\|h(x_i)\right\|^2} \qquad \text{Equation 5.6}$$

So the minimum value of $M(k^{'})$ is as following:

$$M(k^{'}) = M(k)(1 - \alpha\frac{(\sum_{i=1}^{N}(u_i - k(x_i))(h(x_i))^{\mathrm{T}})^2}{(\sum_{i=1}^{N}\left\|h(x_i)\right\|^2)}\frac{1}{q\sum_{i=1}^{N}\left\|y_{i,1}-g(x_i)\right\|^2 + \alpha\sum_{i=1}^{N}\left\|u_i - k\ (x_i)\right\|^2})$$

It is obvious that the value is reduced and the h can be computed by:

$$\begin{aligned} h^{'} &= \arg\max_{h}\{\alpha\frac{(\sum_{i=1}^{N}(u_i - k(x_i))(h(x_i))^{\mathrm{T}})^2}{(\sum_{i=1}^{N}\left\|h(x_i)\right\|^2)} \\ &\quad \frac{1}{q\sum_{i=1}^{N}\left\|y_{i,1}-g(x_i)\right\|^2 + \alpha\sum_{i=1}^{N}\left\|u_i - k\ (x_i)\right\|^2}\} \qquad \text{Equation 5.7} \\ &= \arg\max_{h}\frac{\left|\sum_{i=1}^{N}(u_i - k(x_i))(h(x_i))^{\mathrm{T}}\right|}{\sqrt{(\sum_{i=1}^{N}\left\|h(x_i)\right\|^2)}} \\ &= \arg\max_{h}\varepsilon(h) \end{aligned}$$

We can get $M(k)$ by adding $(\lambda^{'}, h^{'})$ iteratively at each round.

The work flow for computing the mapping function can be seen in Figure 5-6.

```
Input: ${y_i, x_i}_1^N, y_i \in \mathbb{R}^{2q}, x_i \in \mathbb{R}^d$
  ● Compute $g(x)$;
  ● Compute $u_i = \{(y_{i,j} - y_{i,1})\}_{j=2}^{q} \in \mathbb{R}^{2q-2}$
  ● Set $k(x) = 0$
  ● Loop: $t: 1 \rightarrow T$
        (a) Compute $k_t(x) = \lambda_t h_t(x)$
              $\lambda_t$ is determined by Eq5.6
              $h_t(x)$ is determined by Eq5.7
        (b) $k(x) = k(x) + k_t(x)$
        (c) Check convergence
  ● End Loop
Output: $F(x) = J(g(x), k(x)): \mathbb{R}^d \rightarrow \mathbb{R}^{2q}$
```

Figure 5-6 Algorithm for computing mapping function

Now, we focus on the weak function pool. How to setup this pool will be discussed. The size of images for training and testing is usually 64×128. In our experiment, we use the variable size strategy. The minimum sub windows are sampled with 1/K of the width and height of its parent detection window, and the size is incremented in a step of 1/K either horizontally or vertically, or both. Finally, we get the set of all valid sub windows. In our experiment, there are 3025 valid sub windows. For each sub window, a weak function is trained. The definition of weak function can be seen in Eq.5.12.

$$h(x):\mathbb{R}^d \to \mathbb{R}^r \qquad \text{Equation 5.12}$$

HOG feature is used here for training the regression functions. Each weak function consists of r single output regression functions. For one sub window, the training is done for r times, and gets r regression method. This can be seen in Figure 5-7.

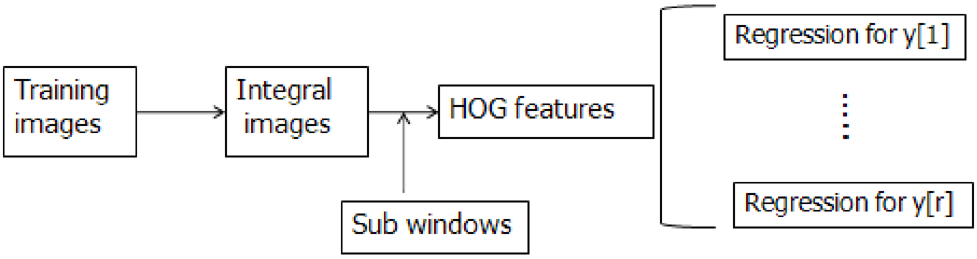


Figure 5-7 HOG feature is used for calculating regression method

In this way, we could get the pool of the weak function pool. By using the method in Figure 5-6, we could get the last mapping function. By this function, we could get the pose estimation and get the bounding boxes for each part of the human body, which can be used in the feature extraction.

Another advantage of this method is that it can be easily improved to solve the occlusion problem in some cases. For example, the surveillance system assumes that human walk on a ground plane and the image is captured by a camera looking down to the ground. This assumption can be seen in Figure 5-8.

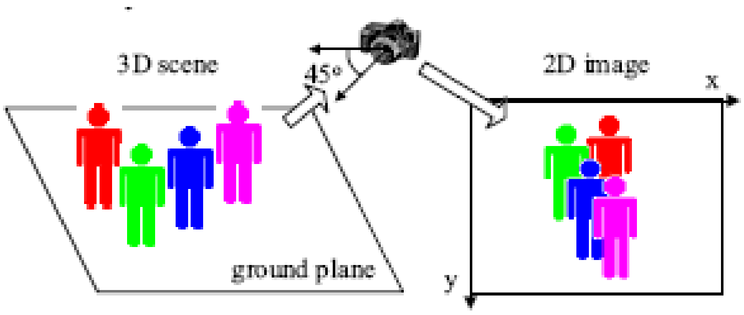


Figure 5-8 3D assumption for surveillance system

It can be deduced that he further the human from the camera, the smaller y-coordinates. The relative depth of humans could be obtained by comparing their y-coordinates. The shape of human is first detected as a box, and then modeled as an ellipse.

Detection is started in the area with higher y-coordinate. This could promise that first detected human without occlusion. Then, the region information of detected human is kept. When other area is detected, the information of occluded region could be obtained. The classifier of which corresponding region is occluded is abandoned.

The cascade rejection structure is used here. The full body is divided into head and should, torso, leg, left of body, right of body. The classifiers are trained respectively using concerning information. Each classifier corresponds to one cascade of final classifier. If it is considered as human, it must pass all the cascades. Using the area information provided by already detected human, the cascades that are in occluded region are omitted. This is can be seen in Figure 5-9.

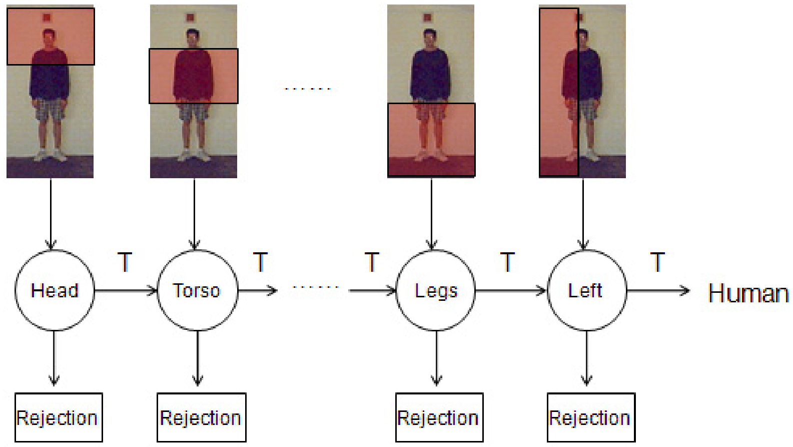


Figure 5-9 Cascade rejection structure for part based detection

In the experiment, we prove that our method is better than other pose-invariant features such as structure learning based method in [37]. We mark the pose of human manfully for 800 positive images in the training dataset in INRIA dataset. The pose information is used for calculating the regression function. In the feature extraction part, the HOG feature is used, just like the structure learning method. The linear SVM is used. In this experiment, we use three points to represent the pose of the human body: the head, the ass and the feet. Two bounding boxes are used for the feature extraction. We compare the result with [37] in which ten rounds are used. This is can be seen in Figure 5-10.

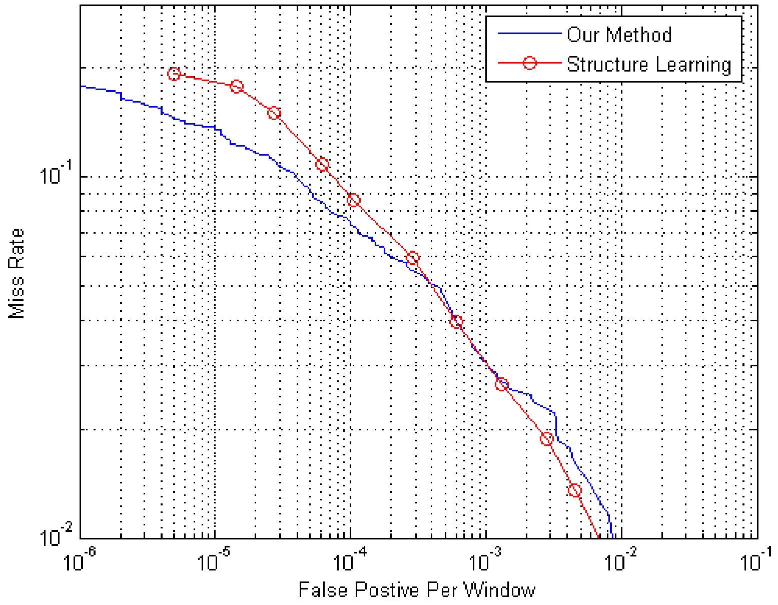


Figure 5-10 Comparison with structure learning method when ten rounds is used

6 Conclusion

In this dissertation, the problem of human detection is mainly discussed. In order to improve the detection rate, we propose several detectors. These detectors can extract the characteristic feature of human body, and improve the performance efficiently, without bring more calculations.

Firstly, the histogram of template feature is proposed. Some formulas and templates are designed to extract the feature from gradient information and texture information. Compared with HOG feature, it has three advantages. First is that this feature is extracted from the template level. We calculate the value for each template, and construct the histogram for template. HOG feature calculate values (orientation value and magnitude value) for each pixel, and actually, it is an orientation voting. The basic unit for voting is pixel. HOT feature is actually a template voting. The basic unit for voting is template. It is more macrostructures, and can extract the characteristic of human. Second is that it integrates the texture information and the gradient information together. So it can show more discriminative abilities than only gradient information based feature, such as HOG feature, even the lengthen of the feature is shorter. Third is that HOT feature is illumination-invariant, so the normalization is not the necessary step for detection, which is very useful for some hardware accelerations that only support integer calculation.

Secondly, some extensions of HOT feature are proposed to further improve the detection rate and efficiency. Multi-scale block HOT feature and a motion based feature are proposed respectively. MB-HOT feature extends the template level by replacing the three pixels' combination by three blocks' combination. In this way, the feature is extracted from more macrostructures levels. We do experiments to show that the features extracted from the different scale templates contain more shape information of human body than that only one scale is used, and it is more discriminative than original HOT feature. The target of motion based feature is to detect the human by using motion information, especially the relative moment of human body. This feature is extracted from the optical flow domain. The definition of formulas and the region for extracting the feature are changed, to make it suitable for extracting the relative motion by using optical flow domain.

Thirdly, we give acceleration method for proposed features. First, we use integral images to save the values of templates. In this way, we could get the histograms for each cell quickly, while some repeating calculations can be avoided. Second, we give the GPU based implementation. The feature calculation part contains lots of parallel calculation, so we re-organize the workflow of feature extraction to make it suitable for GPU based implementation. Experiment shows that the proposed method can reduce the calculation time efficiently.

Finally, a pose invariant feature is proposed, to detect human in different poses. The feature is extracted from the estimated parts of human body. In this way, the redundant information of background can be removed efficiently. The pose estimation problem is defined as a multi-output regression problem. A new definition of loss function is given to find the mapping function, by which the images can be mapped into the pose space. Orientated gradient is mainly used in pose estimation, which reduces the number of sub windows which are used in boosting based regression method, compared to Haar-like feature. Experiment shows that pose estimation can increase the detection rate efficiently by correctly detecting the humans whose poses are different from ordinary.

In conclusion, in order to solve the human detection problem and make it possible for practical applications, several features are proposed to detect human from images or videos. The experiments and analysis can show that these proposed features have advantages respectively and can detect human robustly.

Acknowledgement

First and foremost, I would like to appreciate my supervisor Professor Satoshi Goto for his constant encouragement, guidance, and support during my research. Without his guidance, help and suggestions, I can’t finish my studies smoothly. I also express my appreciation to my second supervisor Professor Yasuo Matsuyama, Professor Takeshi Yoshimura, and Professor Masahiro Okuda for the guidance throughout my work. Thank them for their concise and accurate advice to me.

I also thank Ms. Maruyama Hiromi, Mr. Ning Jiang, Mr. Yijun Lu, Mr. Axel Beaugendre and Mr. Jiu Xu for working together with me on the topic of object detection and tracking method. I also express my sincere thanks to Dr. Xin Jin and Dr. Dajiang Zhou for giving me so much valuable support. I also thank Mr. Zhenxing Chen and Mr. Han Jiang for their help. I also thank all the members of Professor Goto’s laboratory for their cooperation.

I also express my sincere gratitude to Waseda University Ambient SoC Global COE Program of MEXT, for the support to my research. I also thank the support from the CREST project of Japan Science and Technology Agency.

Finally, I thank my families and friends for their kind support over the years. Thanks to my parents and my wife, for your love, support and understanding.

Publications

Journal Papers

[1] Shaopeng.Tang and Satoshi.Goto, "Histogram of template for pedestrian detection," IEICE Trans. INF. & SYST., VOL.E93-D, NO.7, pp. 1737-1744, July 2010

[2] Shaopeng.Tang and Satoshi.Goto, "Accurate human detection by appearance and motion," IEICE Trans. INF. & SYST., VOL.E93-D, NO.10, pp 2728-2736, Oct. 2010

International Conference Papers

[1] Shaopeng.Tang, Lili Wang, Aimin Hao, "A Method for Terrain Rendering without Seams Based on Image," in 10th International Conference on Computer-Aided Design and Computer Graphics (CAD/GRAPHICS), Beijing, 2007, pp. 481-485.

[2] Shaopeng.Tang and Satoshi.Goto, "Partially Occluded Human Detection By Boosting SVM," in 5th International Colloquium on Signal Processing & Its Applications, Malaysia, 2009, pp. 224-227.

[3] Shaopeng.Tang and Satoshi.Goto, "Pedestrian Detection with an Ensemble of Localized Features," in IEEE International Symposium on Circuits and Systems (ISCAS), Taipei, 2009, pp. 2838-2841.

[4] Shaopeng.Tang and Satoshi.Goto, "Human detection using motion and appearance based feature," in ICICS, Macao, 2009, pp. 1-4.

[5] Shaopeng.Tang and Satoshi.Goto, "Histogram of template for human detection," in ICASSP, Dallas, 2010.3.

[6] Shaopeng.Tang and Satoshi.Goto, "Multi scale block histogram of template feature for pedestrian detection," in ICIP, HongKong, 2010.9.

Reference

[1] N.Dalal and B.Triggs, "Histograms of oriented gradients for human detection," in *Conference on Computer Vision and Pattern Recognition*, 2005.

[2] Q.Zhu*, et al.*, "Fast human detection using a cascade of histograms of oriented gradients," in *IEEE Conf. on Computer Vision and Pattern Recognition*, New York, 2006, pp. 1491-1498.

[3] T.Oncel and P.Fatih, "Pedestrian detection via classification on riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, Oct. 2008.

[4] K.Mikolajczyk*, et al.*, "Human detection based on a probabilistic assembly of robust part detector," in *ECCV*, 2004, pp. 69-82.

[5] D.Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV,* vol. 60, pp. 91-110, 2004.

[6] Z.Lin and S.Larry, "A Pose-Invariant Descriptor for Human Detection and Segmentation," in *ECCV*, 2008.

[7] M.Yadong and Y.Shuicheng, "Discriminative local binary patterns for human detection in personal album," in *CVPR*, 2008.

[8] L.Nanni and A.Lumini, "Ensemble of multiple pedestrain reprssentations," *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, June, 2008 2008.

[9] B. Scholkopf and A. Smola. (2002) Learning with kernels support vector machines, regularization, optimization and beyond.

[10] J.Friedman*, et al.*, "Additive logistic regression: a statistical view of boosting," *Ann. Stat.,* vol. 28, pp. 337-407, 2000.

[11] P.Viola and M.Jones, "Rapid object detection using a boosted cascade of simple features," in *Coference on Computer Vision and Pattern Recognition*, 2001.

[12] B.Wu and R.Nevatia, "Detection and tracking of Multiple, partially occluded humans by bayesian combination of edgelet based part detectors," *IJCV,* 2007.

[13] P. Papageorious and T. Poggio, "A trainable system for object detection," *IJCV,* vol. 38, pp. 15-33, 2000.

[14] S.Munder and D.M.Gavrila, "An experimental study on pedestrian classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, November 2006.

[15] B.Leibe*, et al.*, "Pedestrian detection in crowded scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, 2005, pp. 878-885.

[16] B.Leibe*, et al.*, "Combined object categorization and segmentation with an implicit shape model," in *ECCV*, 2004, pp. 17-32.

[17] S.Agarwal and D.Roth, "Learning a sparse representation for object detection," in *ECCV*, 2002.

[18] E.Seemann*, et al.*, "Towards robust pedestrian detection in crowded image sequences," in *CVPR*, 2007.

[19] D.M.Gavrila and V.Philomin, "Real-time object detection for smart vehicles," 1999, pp. 87-93.

[20] Z.Lin*, et al.*, "Hierarchical part-template matching for human detection and segmentation," in *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[21] LibSVM [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[22] INRIA Dataset [Online]. Available: http://lear.inrialpes.fr/data

[23] Z.Wei*, et al.*, "Real-time accurate object detection using multiple resolutions," in *ICCV*, 2007.

[24] Shaopeng.T and Satoshi.G, "Histogram of template for human detection," in *ICASSP*, Dallas, 2010.

[25] S.Hedvig, "Detecting human motion with support vector machines," presented at the ICPR, 2004.

[26] G.Dhiraj and C.Tsuhan, "Real-time pedestrian detection using eigenflow," presented at the ICIP, 2007.

[27] N.Dalal*, et al.*, "Human detection using oriented histograms of flow and appearance," presented at the ECCV, 2006.

[28] A.S.Ogal and Y.Aloimonos, "Shape and the stereo correspondence problem," *International Journal of Computer Vision,* vol. 65, pp. 147-162, Dec. 2005.

[29] C.Zach*, et al.*, "A Duality Based Approach for Realtime TV-L1 Optical Flow," presented at the DAGM, 2007.

[30] A.S.Ogal and Y.Aloimonos, "A roadmap to the integration of early visual modules," *IJCV,* vol. 72, pp. 9-25, Apr. 2007.

[31] CAS [Online]. Available: http://www.cbsr.ia.ac.cn/english/index.asp

[32] A.Ess*, et al.*, "Robust multi-person tracking from a mobile platform," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* 2009.

[33] L.Yuan*, et al.*, "Learning to associate: hybridboosted multi-target tracker for crowded scene," presented at the CVPR, 2009.

[34] W.Abd*, et al.*, "Real-time human detection and tracking from mobile vehicles," presented at the IEEE Intelligent Transportation System Conference, 2007.

[35] C.Wojeck*, et al.*, "Sliding-windows for rapid object class localization: A parallel technique," presented at the DAGM, 2008.

[36] Z.li and N.Ramakant, "Efficient Scan-Window Based Object Detection using GPGPU," presented at the CVPR, 2008.

[37] D.Tran and D.Forsyth, "Configuration estimates improve pedestrian finding," in *NIPS*, 2007.

[38] Z. Lin and L. S. Davis, "A pose-invariant descriptor for human detection and segmentation," in *ECCV*, 2008.

[39] R. Okada and S. Soatto, "Relevant feature selection for human pose estimation and localization in cluttered images," in *ECCV*, 2008, pp. 434-445.

[40] T.Hastie*, et al.*, *The elements of statictical learning: data mining, inference and prediction*. New York: Springer-Verlag, 2001.

[41] S. K. Zhou*, et al.*, "Image based regression using boosting method," in *ICCV*, 2005.

[42] A. Bissacco*, et al.*, "Fast human pose estimation using appearance and motion via multi-dimensional boosting regression," in *CVPR*, 2007.