Cascade Classifier Using Combination of Histograms of Oriented Gradients for Rapid Pedestrian Detection

Wenhui Li
College of Computer Science and Technology, Jilin University, Changchun 130012, China
Email: liwenhui2050@163.com

Yifeng Lin
College of Computer Science and Technology, Jilin University, Changchun 130012, China
Email: linyifeng_jlu@yahoo.cn

Bo Fu and Mingyu Sun
College of Computer Science and Technology, Jilin University, Changchun 130012, China
Email: {fubocloud@163.com, sunmingyu370@sohu.com}

Wenting Wu
CSR Qishuyan Locomotive & Rolling Stock Technology Reaching Institute co., Ltd, Changzhou 213011, China
Email: wuwenting@foxmail.com

***Abstract*—Accurate and efficient human detection has become an important area for research in computer vision. In order to solve problems in the past human detection algorithms such as features with fixed sizes, fixed positions and fixed number, the human detection based on united Hogs algorithm was proposed. This algorithm can dynamically generate the features closer to human body contours. Basically maintaining the detection speed, the detection accuracy was improved by our algorithm. We demonstrate that comparing with human detection algorithm using haar, traditional hogs and a hog with variable-size blocks, our algorithm is better in both of detection rate and false positive rate.**

***Index Terms*—pedestrian detection, hog, adaboost, weak classifier, cascade classifier**

I. INTRODUCTION

In recent years, with the development of image recognition, object detection in video sequences and 2D images has made a series of success. Such as, in the study of human face detection, Viola and Jones [1] proposed the algorithm of rectangular features with cascade boosting, which made face detection faster and more accurate. After the great success in face detection technology, human detection has become a hot issue on computer vision [2]. Useful information on human detection is got mainly from body shapes and body parts. Relevant algorithms in human detection have been proposed, which are mainly divided into two categories: methods based on various parts of the body (VPB, for short) and methods based on single detection window. Reference [3] gave a description of them in detail.

For all parts of the body methods, their main purpose is to deal with various changes in the human body, such as the changes in light and body postures. In these methods, each part of the body is usually detected respectively, and calculus of their geometry probabilities is done to detect the overall human body. In 2001, Ioffe and Forsyth [4] described VPB as horizontal projections of vertical cylinders, and proposed the incremental combination method, which efficiently combined these parts into the whole body. In the same year, Belongie et al [5] proposed to use the context of relations between VPB to describe the human body. Mikolajczyk et al [6] featured the local orientations and their relationships to describe VPB in 2004.

In 2005, Felzenswalb and Huttenlocher [7] used Gaussian derivative models with different directions and scales to define VPB, and elasticly assembled them into one human body. Also in 2005, Wu et al [8] proposed the Edgelet method based on silhouette, which was good when the body was blocked, but it was complex in feature extraction. Mikolajczyk and Uemura [9] extracted a large number of low-dimension features in local motion and shape, and unsupervised combined clusters to build vocabulary tree and vocabulary forest. They extracted features from the test sequence to access the tree and

Nation Nature Science Foundation: No 60873147; Science and Technology Development Program of Jilin Province: No 20060527

Corresponding Author: Yifeng Lin (1982—): Male, Ph D. His major research interests include detection of human face and human body, supervised and semi-supervised learning, image classification and image retrieval.

doi:10.4304/jsw.8.1.71-77

voted the target type and location, depend on probability. This method could be used to locate and recognize the target efficiently. In local features representation, a large number of interest points in space and time are required to be extracted in order to maintain the performance of the detection.

Preprocessing is often required, such as camera motion compensation and extracting interest points in foreground areas (interest areas) by machine learning. Literature [10] discussed the preprocessing in extracting local shapes and target features.

Human action (such as gestures and human behavior) has been wildly used in human-computer interaction. Human detection is preprocessed in some algorithms for human action detection [11], in which human action is detected after its body detection. Therefore, human action detection and recognition provides a new idea in human detection [12]. Human action is non-rigid. There are two issues involved in human action detection, which are how to represent human action and how to classify them. These also become two main steps in human action detection [13].

In some of the existing human action detection algorithms [12], landmarks are extracted after human body or part of the body is tracked. Using these landmarks, human action can be detected and recognized. Optical flow is used to detect human action in other algorithms. A new optical flow descriptor based on median motion compensation was proposed to describe human action and interest areas in literature [14]. In this algorithm, the shape descriptor Dh and the optical flow descriptor Df were combined to be a joint hog-flow descriptor. This method depended on the map of foreground division or the likelihood map of appearance, and it could be used in moving cameras and dynamic background. This method could not only effectively calculate optical flow caused by motion in the background, but also correct optical flow in the foreground, so this optical flow descriptor was robust to dynamic and complex background.

However, human detection algorithms using Hog, whichever Dalal and Triggs [15] or Zhu et al [2] proposed, are static in the stage of feature extraction, and the number of features obtained is limited greatly by the size of training samples.

II. Fast Human Detection Using a Cascade of Hogs Framework

The core of human detection based on *Histograms of Oriented Gradients (*Hog) is the statistical gradient feature extraction. First, the input color image is transformed to a gray image, and a random detection window is obtained from it. Gradient of each pixel within the window is calculated, which makes such a detection window to be a gradient map with the same size. In order to speed computing, the range of gradient is divided into nine directions, and the gradient integration map at each direction is calculated corresponding to the gradient map, which makes such a detection window to be the gradient integration maps at nine directions with the same size.

The detection window can be divided into several overlapping blocks, and reference [9] proved a more accurate detection when blocks were overlapping. Each block can be divided into smaller sub-blocks (cell). Gradients at nine directions in one cell are calculated through gradient integration maps. The numbers of these gradients can constitute a 9-dimension vector. If one block is divided into four cells with equal size, a 36-dimension feature vector will be got in a block. The feature vector will be normalized in its block, which will make its length to be 1. Reference [9] proved a more accurate detection when normalization happened. The 36D vector featured by the block is reduced to one value with one dimension, which will be trained to be a weak classifier using linear SVM. In order to improve the detection speed, the weak classifiers are trained into a few strong classifiers by AdaBoost algorithm, which will finally be combined to be a cascade classifier for human detection. The fast human detection algorithm using Hog in this paper is shown in Fig. 1:

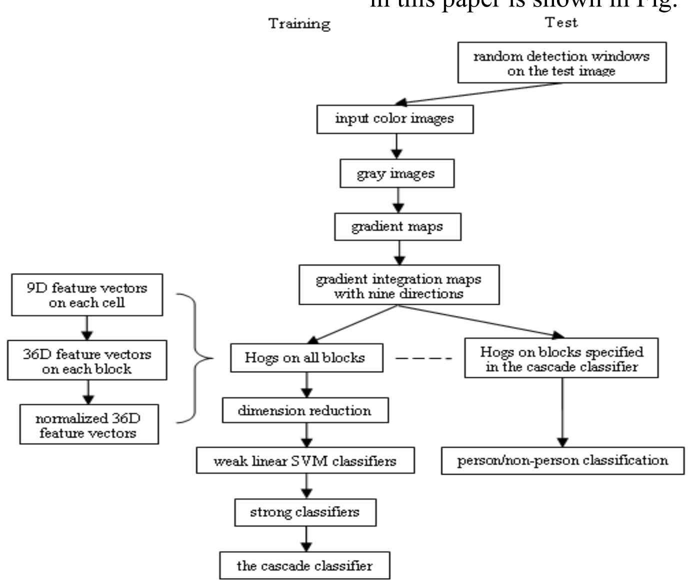


Figure 1. Fast human detection algorithm using histograms of oriented gradients

There are two obvious shortcomings in the algorithm proposed by Zhu et al [2]:

a). the fixed number of features, and

b). a large number of redundant operations in training of weak classifiers and strong classifiers because of the poor separability caused by untrained initial features.

To solve these problems, this paper proposes a pedestrian detection method based on the combination of Hogs (We call it “united Hogs”). First of all, the brief introduction to the irregular rectangles is needed, which is critical in our method.

III. A Novel Representation of Irregular Rectangles

A. Vertex-vector Representation of Rectangles

In order to improve computational efficiency, we describe and track an irregular rectangle with *Vertex-Vector Representation* (VVR for short). A clockwise scan of the outer boundary of an irregular rectangle starts from the vertex at the left-top of this rectangle. The rectangle will be represented by vertices obtained in this proceed. There is a digit between the vertices, which means scanning direction and should be 0, 1, 2 and 3, when scanning right, scanning down, scanning left and scanning up, respectively. This digit also means containing direction of the edge composed of two vertices in VVR. That is, it means which side of this edge is contained within the rectangle. In the case of clockwise scan, scanning right, scanning down, scanning left and scanning up indicates containing down, containing left, containing up and containing right. Edges expressed by VVR are vector edges, using bold, such as $\mathbf{P_2P_3}$, directions of which are digits between two vertices. Fig. 2 shows VVR of a regular rectangle and an irregular rectangle.

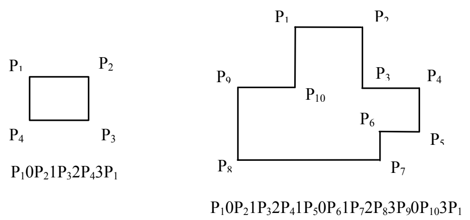


Figure 2. VVR of a regular rectangle and an irregular rectangle

B. The Intersection of Two Regular Rectangles

Fig. 3 shows the four situations in the intersection of two regular rectangles. The intersection detection relies on this: whether any one of the four vertices of one rectangle is in the range of four vertices coordinates of another or not. That is, two regular rectangles will be intersected, if the following is correct:

R_1: P_1 (x_1, y_1), P_2 (x_2, y_2), P_1 and P_2 are the vertices at the Diagonal of one rectangle (the same below)

R_2: P_3 (x_3, y_3), P_4 (x_4, y_4)

$$\text{If } x_i \in [x_3, x_4] \text{ and } y_i \in [y_3, y_4],\ i=1, 2; \tag{1}$$

$$\text{or } x_i \in [x_1, x_2] \text{ and } y_i \in [y_1, y_2],\ i=3, 4. \tag{2}$$

Note that situation d in Fig. 3 does not exist when two rectangles has the same size and different locations. Formula (1) and (2) should be executed at the same time for intersection test in situation d in Fig. 3.

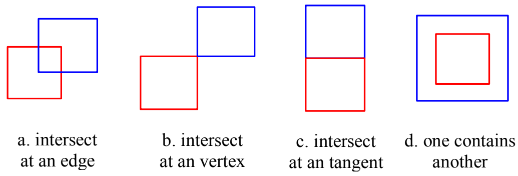


Figure 3. The intersection of two regular rectangles

C. The Intersection of Two Irregular Rectangles

Fig. 4 shows three situations in the intersection of two irregular rectangles. Irregular rectangles combined in our paper are in different locations, so situation that one irregular rectangle is contained by another is not considered.

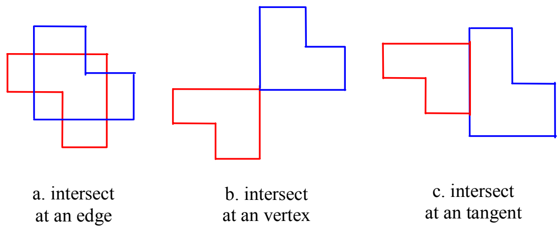


Figure 4. The intersection of two irregular rectangles

There are a lot of intersection detection methods on irregular polygons. Because of VVRs of irregular rectangles in our paper, the intersection of irregular rectangles becomes the intersection of the edges of rectangles. The edge from an odd vertex to an even vertex in VVR is horizontal and the digit between them is even; the edge from an even vertex to an odd vertex is vertical and the digit between them is odd. For example, in the rectangle at the right side of Fig. 2, $\mathbf{P_5P_6}$ is horizontal and $\mathbf{P_{10}P_1}$ is vertical. The intersection of two horizontal (vertical) edges means the part coincidence of them, while the intersection of one horizontal edge and another vertical edge brings a point, taking advantage of which, we improve the intersection detection method on irregular rectangles as follows:

N_1, N_2: the number of vertices of two irregular rectangles

P^1_i: the i^{th} vertices in VVR of the first irregular rectangle, (x^1_i, y^1_i) is its coordinates

d^1_i: the i^{th} direction in VVR of the first irregular rectangle

P^2_j: the j^{th} point in VVR of the second irregular rectangle, (x^2_j, y^2_j) is its coordinates

d^2_j: the j^{th} direction in VVR of the second irregular rectangle

for (int i=1; i<N_1+1; i++)//each edge of N_1

{

if (d^1_i==1||d^1_i==2)//exchange the coordinates of P^1_i //and $P^1_{i\%N1+1}$ when scanning down and scanning //left to make sure the coordinates of P^1_i are //always smaller than the coordinates of $P^1_{i\%N1+1}$ // (the same below)

```
        {
           exchange (x^1_i, x^1_{i%N1+1});
           exchange (y^1_i, y^1_{i%N1+1});
        }
        for (int j=1; j<N_2+1; j++)
        {
            if (d^2_j==1||d^2_j==2)
            {
               exchange (x^2_j, x^2_{j%N2+1});
               exchange (y^2_j, y^2_{j%N2+1});
            }
            if ( (x^1_{i%N1+1}>=x^2_j)&& (x^1_i<=x^2_{j%N2+1})&& (y^1_{i%N1+1}>=y^2_j)&& (y^1_i<=y^2_{j%N2+1}))
              return true; //the coordinates of the first are in
              //the range of the coordinates of the second,
              //which means they are intersected
        }
    }
```

Obviously, algorithm B is a special case of algorithm C.

IV. Fast Pedestrian Detection with United Hogs

Fast Pedestrian Detection with Multi-Hogs is as follows:

f: the expression of irregular rectangular which a feature or a weak classifier is based on

n: the layer where a feature or a weak classifier is

Set Q: store the expression (f, n) of new features generated in each layer

Set T: store the expression (f, n) of new features generated in all layers

Set T*: store the expression (f, n) of new useful features in all layers

1) Take 1/10 (floor) pixels of the smaller one in length and width of the training images as the initial side size of blocks (aspect ratio is 1: 1); take the step as half of the side size of blocks; scan training images and obtain the initial features; n=1. Make sure the smallest size of a block is 2*2 pixels and the smallest step is 1 pixel.

2) Train these features to weak linear SVM classifiers, and add their expression (f, n) into Q and T; select *useful* weak classifiers from Q as initial ones in the first layer, and add their expressions (f, n) into T*; empty Q.

3) When the current layer n=1, use algorithm B and otherwise use algorithm C to determine the intersection of weak classifiers; n=n+1.

4) Use the algorithm on the generation of VVR of the new rectangle in the intersection of two irregular rectangles to generate the expressions (f, n) of new weak classifiers obtained by combinations in step 3; judge on the same irregular rectangle for each new weak classifier: compare f with all the elements that have n!=1 in T. If a feature does not have the same irregular rectangle as others, then use the algorithm on the generation of inner points in an irregular rectangle to calculate Hog values, train this feature to a weak linear SVM classifier and add its expression (f, n) to Q and T.

5) Select *useful* weak classifiers from Q as ones in the current layer, add their expressions (f, n) into T* and empty Q.

6) Repeat the steps 3, 4 and 5. The number of repetition is the number of layers, which depends and generally from 5 to 10 times. T* is the set of weak classifiers in these training samples.

After combination, the detection rate and false positive rate of weak classifiers have improved. The shape of Human body should give priority to meet these *advanced* weak classifiers, which therefore will be selected in the early stages of cascade classifier. As long as they satisfy the certain requirements in the certain stage of cascade classifier, each of them can be a strong classifier. It shows that a weak classifier after combination is actually a strong classifier in a sense. The more effective features are obtained by combining weak classifiers constantly. The method of feature integration in our paper is totally different from Adaboost algorithm and is a more thorough application of the machine learning in features extraction.

V. Experiments and Results

The original MIT pedestrian dataset contains 509 training images and 200 test images, with the backgrounds of cities, simple pedestrian gestures and only two perspectives. The algorithm proposed by Dalal and Triggs was almost perfect in this test set, so they formed a more challenging set INRIA with new data. This dataset contains 1805 pedestrian images in 64*128 resolutions with a variety of postures, very complicated backgrounds and rapid changes of illumination. All experiments in our paper are done on this test set.

We define a few statistics that are often used in human detection:

False Positive (FP): the number of false detection in negative samples

False Negative (FN): the number of false detection in positive samples

True Positive (TP): the number of true detection in positive samples

True Negative (TN): the number of true detection in negative samples

Detection rate (or recall)=the number of true detection in positive samples / total number of positive samples=TP/ (TP+FN)

False positive rate=the number of false detection in negative samples / total number of negative samples=FP/ (FP+TN)

Detection precision=the number of true detection in positive samples / the number of samples that are detected to positive=TP/ (TP+FP)

False detection rate=1- Detection precision= the number of false detection in negative samples / the number of samples that are detected to positive=FP/ (TP+FP)

miss rate=1- recall=the number of false detection in positive samples / total number of positive samples=FN/ (TP+FN)

FPPW (false positive per window)=the number of false detection in negative samples windows / total number of negative samples windows. FPPW is equivalent to false positive rate.

Accumulated reject rate=the number of windows that were rejected before the current stage (including the current stage) in the cascade classifier / total number of windows that were rejected in the cascade classifier

Sparse scan: scan 800 detection windows in each image

Dense scan: scan 12800 detection windows in each image

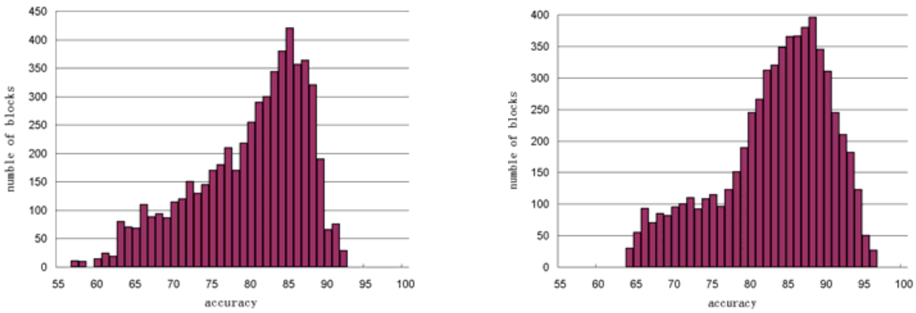
As mentioned earlier, for training images with 64*128 pixels, we used 6*6 pixels as the initial size of blocks and 3 pixels step. Therefore there was a total of 20*41=820 initial blocks in the first layer, which then were combined to generate new features in 9 layers. The useful weak classifiers were trained to the effective strong AdaBoost classifiers which constituted the cascade classifier with fmax=0.7 and dmin=0.998 in each stage. The cascade classifier was trained with the algorithm in reference [2] about 9 days in a PC with the performance of 1.4GHz CPU and 512MB RAM. Test samples were in INRIA.

For test images with 64*128 in INRIA, if we mark the weak classifiers, of which average detection rates are greater than 50% and average false positive rates are less than 45%, as the useful weak classifiers, Table I shows that the proposed algorithm in this paper is better than the algorithm by Zhu et al [2] in weak classifiers. The performance of strong classifiers and the cascade classifier is guaranteed by reduction in the number of weak classifiers and their usefulness, which also improve the detection rate and reduce the training and testing time.

TABLE I.
THE COMPARISON IN WEAK CLASSIFIERS BETWEEN OUR METHOD AND ZHU ET AL METHOD

	Total number of weak classifiers	The number of useful weak classifiers	useful weak classifiers proportion	the average detection rate of useful weak classifiers	the average false positive rate of useful weak classifiers
Fast human detection using hogs with variable-size blocks by Zhu et al	5031	947	18.82%	66.10%	30.71%
Human detection using united hogs in our paper	3787	3787	100.00%	72.33%	26.95%

Fig. 5 shows classification accuracy of blocks in the algorithm by Zhu et al [2] and our algorithm with the same false rate. Blocks here are just trained to the weak linear SVM classifiers, and do not constitute strong classifiers or even cascade classifier. It can be shown that the features from our algorithm contain more information and higher accuracy of human detection.



(a) (b)

Figure 5. Classification accuracy of (a)5031 blocks in Zhu et al algorithm and (b)3787 blocks in our algorithm

Table II shows relevant information of cascade classifiers after their training in human detection using Haar, Hogs with variable-size blocks in L1 normalization and united Hogs in L2 normalization in a PC with the performance of 1.4GHz CPU and 512MB RAM. Accumulated reject rates in these three algorithms are shown in Fig. 6:

- There are 30 stages in the cascade classifier in human detection using Hogs with variable-size blocks. Accumulated reject rate will not reach 90% until the fourth stage.
- There are 24 stages in the cascade classifier in our algorithm. The third stage can make accumulated reject rate reach 90%.
- There are 14 stages in the cascade classifier in human detection using Haar. Accumulated reject rate will not reach 90% until the tenth stage.

Haar is unstable for human detection, and detection windows that were rejected by strong classifiers at each stage are basically even. So, accumulated reject rate curve of the cascade classifier in human detection using Haar does not work obviously in early stages as the other two curves. Its other properties are also the worst except its time at training and testing. As can be seen from Table II, in the cascade classifier in our algorithm, the average number of SVM weak classifiers per stage, the average number of weak classifiers for detecting per window, the detection rate and false positive rate in INRIA are optimal. Although the algorithm proposed in our paper improves the detection rate, it is difficult to accurately compare system overheads in training and testing between our method and Zhu et al method. Here are several reasons:

1) Algorithm B and algorithm C are executed frequently in generation of weak classifiers in our algorithm, which greatly increases the training and testing time. However, initial blocks that are selected regularly reduce system overheads.

2) Regardless of training or testing, the algorithm on the generation of inner points in an irregular rectangle has to be needed to calculate the feature vector of a block generated, which increases overheads comparing with the traditional gradient integration maps in regular rectangles.

However, gradient integration maps in 9 directions doesn't have to be calculated in our algorithm, and feature vectors are reduced from 36-dimension to 9-dimension, which also save time to some extent when dimensions are reduced to the feature value.

3) L1 normalization is good for the rapid calculation of gradient integration maps, so the feature vector of every block is normalized in L1 in Zhu et al method, while L2 normalization is used in our method.

4) The average number of SVM weak classifiers per detection window is reduced by the cascade classifier trained in our algorithm. But it is meaningless to compare only this property, because the complexity of a single weak classifier is increased in our algorithm, which is different from simple calculation of feature vectors by gradient integration maps in Zhu et al method.

Results in Table II show the conclusion of these discussions: in our algorithm, the training and testing time have increased to some extent, and the difference in the time that is spent in detecting each frame between sparse scan and dense scan has also increased. Especially when the sparse scan, speed in Zhu et al algorithm can achieve 33FPS, and speed in our algorithm can only reach 12FPS, which reduces by half and cannot strictly meet the real-time requirement.

TABLE II.
THE COMPARISION IN CASCADE CLASSIFIERS IN THESE THREE ALGORITHMS

	Human detection using Haar	Human detection using hogs with variable-size blocks in L1 normalization	Human detection using united hogs in L2 normalization
The number of SVM weak classifiers in first layer	18	4	2
The number of SVM weak classifiers in second layer	21	4	2
The number of SVM weak classifiers in third layer	24	4	3
The number of SVM weak classifiers in forth layer	30	4	3
The number of SVM weak classifiers in fifth layer	39	6	5
Total number of SVM weak classifiers	760	811	527
The number of layers	14	30	24
The average number of SVM weak classifiers per layer	54.29	27.03	21.96
The lowest Detection rate ruled in each layer	90.00%	99.75%	99.80%
The highest false positive rate ruled in each layer	70.00%	70.00%	70.00%
Detection rate in INRIA	52.49%	93.46%	95.78%
False positive rate in INRIA	0.001572	0.000015	0.000015
The average number of weak classifiers per detection window	20.0	4.8	3.1
The training time in the same condition	6 days	7.5 days	8.8 days
Average time spent in detecting each frame in sparse scan	12ms	30ms	84ms
Average time spent in detecting each frame in dense scan	60ms	128ms	520ms

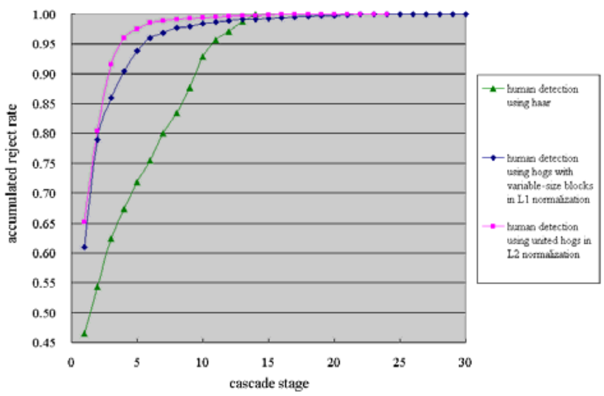


Figure 6. Accumulated reject rates of these three algorithms

VI. SUMMARY

We propose the human detection based on united Hogs algorithm. This algorithm can dynamically generate the features closer to the human body contours. Basically maintaining the detection speed, the detection rate of our algorithm is increased by 2.75% and 4.03% than the fast human detection based on Hog with variable-size blocks algorithm.

Since rectangles generated in the combination are irregular, it is difficult to use integration maps to speed up calculations and therefore the detection speed is severely affected, which is the further research in the next paper.

ACKNOWLEDGMENT

The authors wish to thank Ying Wang and Hongyin Ni, who spent a lot of time on programming and testing. This work was supported in part by a grant from Nation Nature Science Foundation No 60873147, and Science and Technology Development Program of Jilin Province No 20060527.

REFERENCES

[1] P. Viola, and M. J. Jones, "Robust Real-Time Face Detection", International Journal of Computer Vision, vol. 52, no. 2, pp. 137–154, 2004.
[2] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients", In Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition, 2006.
[3] D. M. Gavrila, "The Visual Analysis of Human Movement: A survey", Journal of Computer Vision and Image Understanding, vol. 73, no. 1, pp. 82–98, 1999.
[4] S. Ioffe, and D. Forsyth, "Probabilistic Methods for Finding People", International Journal of Computer Vision, vol. 43, no. 1, pp. 45–68, 2001.
[5] S. Belongie, J. Malik, and J. Puzicha, "Matching Shapes", In Proceeding of the 8th ICCV, Vancouver, Canada, pp. 454–461, 2001.
[6] K. Mikolajczyk, C. Schmid, amd A. Zisserman, "Human Detection based on a Probabilistic Assembly of Robust Part Detectors", In Proceeding of European Conference on Computer Vision, 2004.
[7] P. Felzenszwalb, and D. Huttenlocher, "Pictorial Structures for Object Recognition", International Journal of Computer Vision, vol. 61, no. 1, pp. 55–79, 2005.
[8] Bo Wu, and Ram Nevatia, "Detection of Multiple Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors", In Proceeding of ICCV, 2005.
[9] K. Mikolajczyk, H. Uemura, "Action Recognition with Motion Appearance Vocabulary Forest", In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New Jersey: IEEE CS, pp. 1–8, 2008.
[10] J. Liu, J. Luo, M. Shah, "Recognizing Realistic Actions from Videos in the Wild", In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New Jersey: IEEE CS, pp. 1–8, 2009.
[11] Z. Lin, Z. Jiang, L. S. Davis, "Recognizing Actions by Shape-motion Prototype Trees", In Proceedings of IEEE Conference on Computer Vision, New Jersey: IEEE, pp. 1–8, 2009.
[12] A. Fathi, G. Mori, "Action Recognition by Learning Mid-level Motion Features", In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New Jersey: IEEE CS, pp. 1–8, 2008.
[13] R.Poppe, "A Survey on Vision-based Human Action Recognition", International Journal of Image and Vision Computing, vol. 28, no. 6, pp. 976-990, 2010.
[14] P. Turaga, R. Chellappa, V. S. Subrahmanian et al, "Machine Recognition of Human Activities: A Survey", International Journal of IEEE Transactions Circuits and Systems for Video Technology, vol. 11, no. 8, pp. 1473–1488, 2008.
[15] N. Dalal, and B. Triggs, "Histograms of Oriented Gradients for Human Detection", In Proceeding of Conference on Computer Vision and Pattern Recognition, pp. 886–893, 2005.

Wenhui Li was born in 1961. He is a Professor and doctor supervisor in College of Computer Science and Technology, Jilin University, Changchun, P. R. China. His major research interests include computer vision, image processing, pattern recognition, graphics, virtual reality, CAD and geometric constraint.

Yifeng Lin was born in 1982. He received the B.Sc. in 2005 and M.S. 2007 in College of Computer Science and Technology, Jilin University, Changchun, P. R. China. He has been working on the PhD degree in this college since 2007. His research interests include computer vision, image processing, pattern recognition, machine learning and 3D virtual engine. He especially has been researching in detection of human face and human body, supervised and semi-supervised learning, image classification and image retrieval.

Bo Fu was born in 1983. Since 2009, He has been working on the PhD degree in College of Computer Science and Technology, Jilin University, Changchun, P. R. China. His research interests include computer image processing, image denoising, and pattern recognition and anomaly intrusion within the video region.

Mingyu Sun was born in 1978. Since 2010, He has been working on the PhD degree in College of Computer Science and Technology, Jilin University, Changchun, P. R. China. His research interests include the research and design on 2D and 3D geometric constraint systems. He has a good understanding of geometric constraint and intelligent learning.

Wenting Wu was born in 1984. She received the B.Sc. in 2007 and works in CSR Qishuyan Locomotive & Rolling Stock Technology Reaching Institute co., Ltd, Changzhou, China. Her research interests include infrared flaw detection on vehicles and tracks. He has a good understanding of image denoising, image matching, registration and correction, and image recognition based on priori knowledge.