PAPER

Accurate Human Detection by Appearance and Motion

Shaopeng Tang,† *Non Member,* Satoshi Goto,† *Fellow*

Summary
In this paper, a human detection method is developed. An appearance based detector and a motion based detector are proposed respectively. A multi scale block histogram of template feature (MB-HOT) is used to detect human by appearance. It integrates gray value information and gradient value information, and represents relationship of three blocks. Experiment on INRIA dataset shows that this feature is more discriminative than other features, such as histogram of orientation gradient (HOG). A motion based feature is also proposed to capture the relative motion of human body. This feature is calculated in optical flow domain and experimental result in our dataset shows that this feature outperforms other motion based features. The detection responses obtained by two features are combined to reduce the false detection. Graphic process unit (GPU) based implementation is proposed to accelerate the calculation of two features, and make it suitable for real time application.

Key words:
Human detection, multi scale block histogram of template, Graphics process unit

1. Introduction

Human detection technique is very important for many applications, such as content analysis, surveillance system and intelligent vehicles. It is a challenging task because of many difficulties. The poses of people are different; the illumination is changing. Especially, people have different clothes. So some very powerful feature in face detection such as skin color cannot be used in human detection. Besides, real time requirement is difficult to achieve for many applications, especially for intelligent vehicle. Pedestrian accident becomes one the largest traffic-related injuries. Automatic pedestrian detection method can reduce this accident efficiently.

Robust and real time human detector should be developed. Recent years, lots of research work has been focused on this field. Learning based human detection method shows excellent performance for this purpose when compared with other method such as template matching and codebook method. Discriminative features are extracted from human appearance and human motion to represent the characteristic of human, to separate the pedestrian from background and other object. For learning based method, people focus on two research directions: looking for more discriminative feature or finding more powerful training method.

Texture, gradient or motion information can be used for extracting feature. Lots of features [1-4] use gradient as clue more or less, and obtain good performance. Histogram of orientation gradient (HOG) [1] is developed from scale invariant feature transform (SIFT) [5], and it is one of most popular feature for human detection. It can represent the gradient characteristic of the human body, especially the shape of head-shoulder, leg and so on. The computation complexity is less than covariance matrix in [3]. In [2] HOG is combined with Boosting method, to reduce the detection time. Texture information can also be used in human detection. Some local binary pattern (LBP) based features [6-7] show discriminative ability in human detection, after modifying the original definition of LBP. The motion of human is different from the motion caused by other objects. Some detectors [8-10] are constructed by using motion information. The combination of appearance based detector and motion based detector can detect the standing people and moving people efficiently from videos [10].

Up to now, the computation complexity is still 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 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 [11-12] 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. Cascade-Rejection structure is used in [2-3, 13]. Only positive samples have to pass all the cascades. For intelligent vehicles, some road subtraction and clustering method can be used to estimate the position and scale of pedestrian [14]. This reduces detection windows more or less, but these methods are a little time consuming.

Recently, Graphics process unit (GPU) shows high computation ability, especially for parallel calculation. The concept of General Purpose GPU is proposed, focusing on solve 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 [15-16] are proposed to solve the computation problem. The GPU based implementation for feature extraction and classification can obtain high speed for detection. New

Manuscript received January xx, 20xx.
Manuscript revised March xx, 20xx.
† The author is with NTT , Musashino-shi, 180-8585,
†† The author is with IEICE Office, Minato-ku, Tokyo, 105-0011 Japan.

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

The purpose of this paper is to propose a robust and real time human detection method. The main contribution includes: a multi scale block histogram of template feature is proposed. Texture information and gradient information are encoded in this feature and it can reflect the relationship of three blocks. GPU based implementation is also proposed to accelerate the detection process. A GPU friendly feature in optical flow domain is also proposed to represent the motion of human, especially the relative motion. This method integrates texture information, gradient information and motion information. GPU friendly algorithm design accelerates detection process, and makes it suitable for practical application. The workflow of our detection method can be seen in Fig. 1.

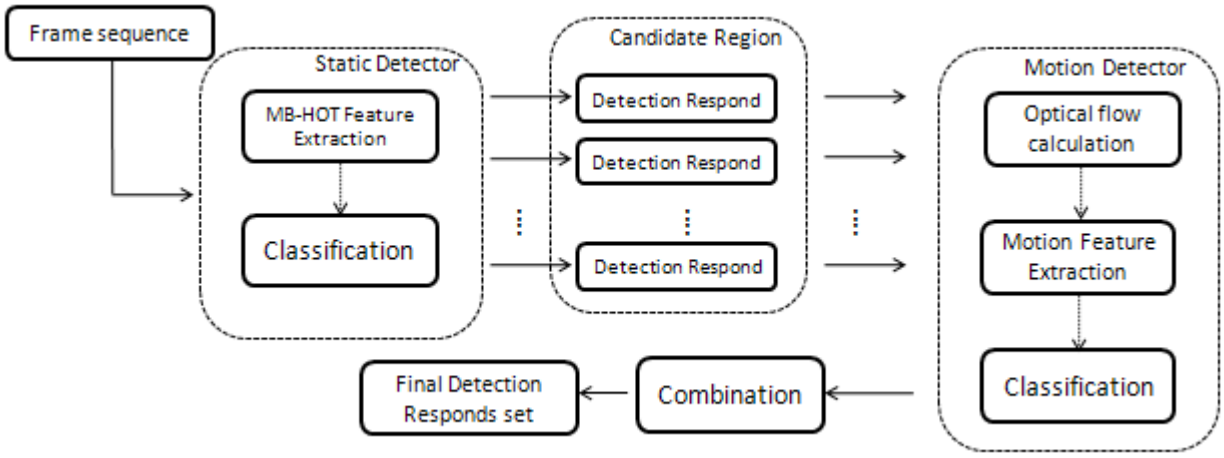

Fig.1 The work flow of our method

The paper is organized as follow. In section 2, some related work about feature extraction is provided. In section 3, we give the definition of MB-HOT feature and its GPU based implementation. In section 4, a motion based feature is given in optical flow field. Experimental results are provided in section 5.

2. Related Work

Appearance based human detection methods can be separated into three groups.

First group of methods are based learning. [1-4, 7, 13, 17-19] Local features are extracted from training dataset and a classifier can be obtained by using some training method, such support vector machine [20], or some boosting method [21]. HOG feature in [1] is one of the most efficient features recently. The orientation of gradient is divided into several bins, and the sum of magnitude for each bin is calculated as feature. SVM is used for training. [2] combines HOG feature with Adaboost method in [13]. [15-16] give the GPU based implementation of HOG feature. Covariance matrix [3] is another powerful feature for human detection. For each pixel, location and intensity derivatives can be obtained. An eight dimensional vector can be used to represent a pixel. A covariance matrix can be calculated by using all vectors of pixels in a region. Each covariance matrix can be treated as a point in Riemannian manifold. Logitboost training method in [21] is used in Riemannian manifold to get classifier. Edgelet feature is proposed in [18]. Different from [4], only horizontal and vertical direction are considered, the predefined edgelet represent the shape of human body well, and makes it more discriminative. Haar-like feature is proposed in [13], integral image is calculated for acceleration and Cascade structure is used, which is widely used in many object detection methods. Local Receptive Fields (LRF) feature [17] and Haar wavelets feature [19] are also suitable features for human detection. Learning based method is efficient for human detection. Lots of training images should be prepared. The more training images it uses the more accurate result would be obtain. Training and detecting are time consuming. Although some acceleration methods can be taken, it is very hard to get acceptable detection time while keep detection rate.

The second group is based on codebook, a set of appearance patches of different parts of object. Appearance patches are extracted from images containing the same object. Some clustering methods are used to construct codebook. In detection process, similar patches in codebook are used to replace the corresponding part in new image and give information about object position. [22-26] are codebook based human detectors. How to find

interesting points to extract patches is very important for this method. The scale is another problem. The size of patches should be carefully selected.

The third group is based on the chamfer matching. They use human template to find the marching regions in edge map of input image. [27-28] are based on this method. [28] proposes a direct template matching approach for global shape-based human detection. [27] is developed form this method and use hierarchical template to reduce the detection time and solve the occlusion problem to some extent. This method is popular in rigid object detection. Although it can also be used in non-rigid object detection such as human detection, it may not give a good result when there are too many edge clusters in edge map.

Recently, there are also many motion based detectors [8-10]. They are suitable for detecting human from video. [8] uses the optical flow as feature for training directly. Support vector machine is used for getting a classifier. This method can get acceptable detection result in outdoor scene. [9-10] also use optical flow. But instead of using flow directly, they extract characteristic feature from optical flow field. This not only reduces the length of feature, but makes the result more precise. In [9], Principal Component Analysis coefficient is extracted from optical flow field for feature. Adaboost method is used for training, which makes it possible for real time detection. [8-9] can only detect moving human from video. In [10], oriented histogram of flow and appearance is extracted as feature for detection. This feature combines flow information and appearance information together, so it can detect the moving and standing human from video efficiently. Some old method can be seen in Gavrila's survey [29].

3. MB-HOT Feature

3.1 Definition

In our previous work, histogram of template feature (HOT) is proposed. Eight templates are given in Fig.2. They are 3 pixels combination in a 3×3 region.

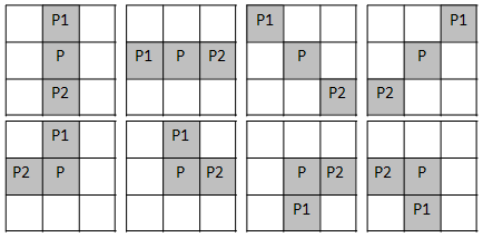

Fig.2 8 templates in HOT feature

A formula is given:

$$I(P) > I(P1) \,\&\, \&\, I(P) > I(P2) \qquad (1)$$

For each template, if the intensity value of P is greater than the other two, it is regarded that the 3×3 region meets this template. An 8 bins histogram can be calculated for detection window. Each bin corresponds to one template. The value of each bin is the number of 3×3 regions meeting corresponding template in detection window. So for formula (1), this detection window can be represented by 8-dimensional vector. Other 3 formulas (2), (3), (4) are also used, so 32-dimensional vector is extracted as feature for this detection window.

$$I(P) + I(P1) + I(P2) >= \arg\max_i \{I(P_i) + I(P1_i) + I(P2_i)\} \qquad (2)$$

$$Mag(P) > Mag(P1) \,\&\, \&\, Mag(P) > Mag(P2) \qquad (3)$$

$$Mag(P) + Mag(P1) + Mag(P2) >= \arg\max_i \{Mag(P_i) + Mag(P1_i) + Mag(P2_i)\} \qquad (4)$$

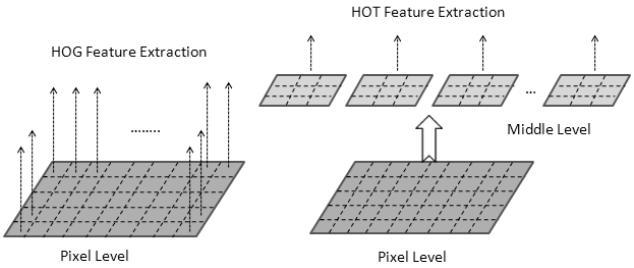

Fig.3 Feature extraction from different level

The main advantage of HOT feature is that this feature is extracted from middle level, which is different from HOG feature that is extracted from pixel level. See Fig.3. For HOG feature, the orientation and magnitude of gradient are calculated for each pixel, and HOG feature is obtained by using all these gradient information. HOT feature is calculated from middle level. The basic unit of this level is 3×3 region. So, HOT feature is more macrostructures than feature extracted from pixel level.

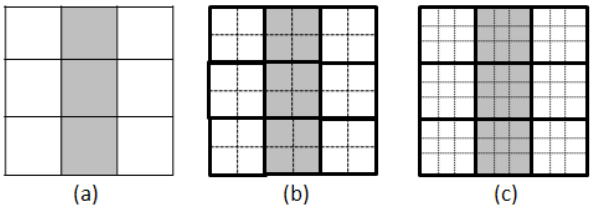

Fig.4 One template used in MB-HOT feature. s =1 for (a), s =2 for (b) and s =3 for (c). The value of each block the average value of all pixels in this block

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 Fig.4. The template is defined in 3×3 block region. The value of each block is the average value of all pixels in this block. So in 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 original HOT feature. In 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 $F = \{f_1, f_2, ...f_i...\}$.

Fig.5 shows that all pixels meeting formula (3) when the templates in Fig.4 are used.

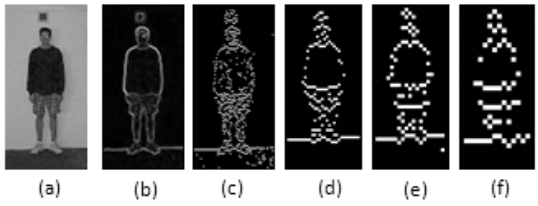

Fig.5 (a) original image; (b) gradient image; (c) for formula (3), all pixels meeting template 1 in Fig.3, when s =1; (d) when s =2; (e) when s =3; (f) when s =4.

The feature extracted from different scale templates contains more shape information of human body than that only one scale is used. In Fig.6, we give the histogram of pixels meeting 8 templates, for formula (1), (2), (3), (4) respectively. So it contains 32 bins. Fig.6 (a), (b), (c), (d) show the histogram by using different scale templates.

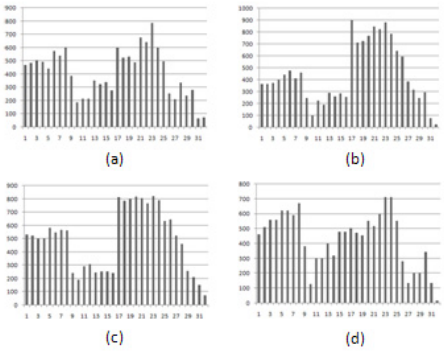

Fig.6 histograms generated by using different scale templates. They are extracted from Fig.4 (a). s =1 for (a), s =2 for (b), s =3 for (c), s =4 for (d). They show different statistically property.

3.2 GPU based Implementation

Due to the huge computation complex, learning based detection method is very hard to be implemented real time. In CPU based implementation, there are some acceleration methods, such as integral image [2, 13], and cascade rejection method [2-3, 7, 13]. There methods reduce detection time, but for practical application, they still need be improved.

Recently, GPU shows high parallel computation ability. Especially after the CUDA computation framework is proposed by NVIDIA. GTX285 and CUDA are used in our method to get real time detector. The flow can be seen in Fig.7.

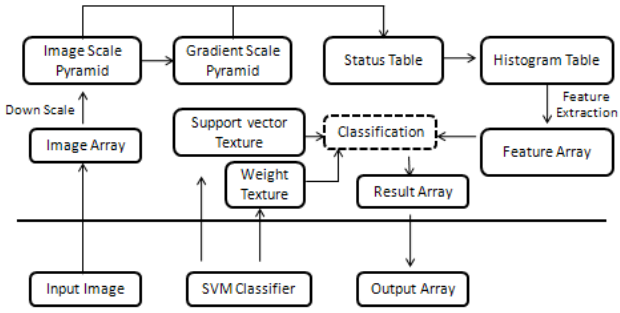

Fig.7 Workflow of GPU based MB-HOT detector

Down Scale: in order to detect human in different size, a scale pyramid is built. For each level of scale pyramid, a sub pyramid is built for MB-HOT feature calculation. See Fig.8. Linear interpolation is used. A thread block contains 16×16 threads and one thread corresponds to one pixel in scaled image.

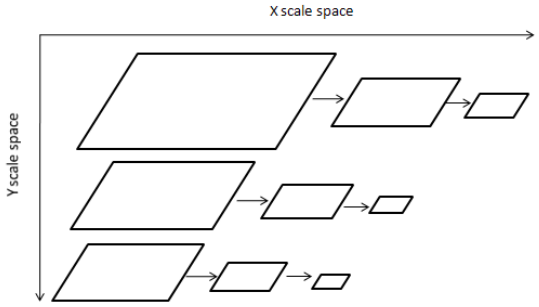

Fig.8 Scale pyramid. Y scale space is for detecting human varying in size. X scale space is for feature calculation.

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 magnitude value of one pixel. 18×18 pixels are copied to shared memory for each thread block, to get coalesced access.

Status Table Calculation: for each pixel in scale pyramid, a status table is calculated. Status table is a 32 bit value. See Fig.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.

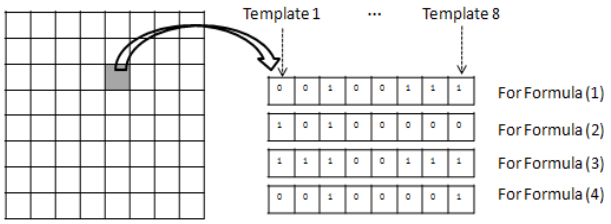

Fig.9 32 bit status table.

Histogram Table Calculation: the input of this part is status table images and the output is a set of 32-bin histograms. One histogram shows the number of pixels meeting different templates in an 8×8 pixels region. A thread block contains 32 threads. One thread computes the value of each bin respectively.

Feature Calculation: calculate a feature for a 64×128 detection window. It contains 7×15 blocks in the first level of X scale space, 3×7 blocks in the second level and so on. A block contains 16×16 pixels. The stride between 2 blocks is 8 pixels. A thread block contains 32 threads. It calculates a 32 dimensional vector for a 16×16 pixels region. A thread is used to compute one value of this 32-d vector. Four histograms are copied from 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 blocks 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 [30]. Classifier is a set of support vectors. These support vectors are stored in texture memory before detection process. The method in [15] is used. Each thread block is responsible for one detection window and each thread computes weighted sums corresponding to each column of the window.

4. Motion based Feature

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

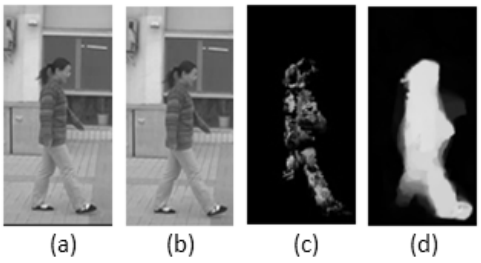

Fig.10 (a) Last frame; (b) Current frame; (c) Optical flow by [31]; (d) Optical flow by [32].

First we focus on how to calculate dense optical flow, which is popular method to estimate motion. Optical flow calculation is a pixel correspondence problem. Given a pixel in the first image, look for a nearby pixel in the second frame. It is supposed that the brightness is constant and motion is small. So displacement vector can be obtained. But if there is great displacements and changing illumination, the result is not good. The method in [31, 33] is used in our approach. The result can be seen in Fig.10 (c).

For feature extraction, our feature is based on histogram of templates feature, but gives some improvement. 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 (1) for example. The formula (1) is replaced by formula (5). The same improvement is applied to formulas (2), (3) and (4), which make our feature suitable to deal with 2 dimensional values.

$$I(P_x) > I(P1_x)\,\&\,\&\,I(P_x) > I(P2_x)\,\&\,\& \\ I(P_y) > I(P1_y)\,\&\,\&\,I(P_y) > I(P2_y) \tag{5}$$

For pixel P, $I(P_x)$ denotes the x (horizontal) components of optical flow; $I(P_y)$ denotes the y (vertical) components 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 block is divided into 9 cells. The optical flow value of pixel in 8 outer cells is subtracted by value of corresponding pixel in central cell. And the histogram of template feature is extracted from 8 outer cells. The motivation is that if the person's limb width is approximately the same size as the cell size, it can capture relative displacements of limbs to the background and nearby limbs [10]. Another reason is that if the camera

is moved smoothly, the subtraction operation can reduce the effect caused by moving camera.

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 optical flow. In our experiment, the optical flow of each frame is pre calculated by using method in [31, 33]. In [32], a GPU based optical flow calculation method is proposed. The result can be seen in Fig10 (d). We will integrate it in our detection framework for future work.

5. Experiment

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

The comparison is done on INRIA dataset [34]. 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 training dataset and testing dataset. 1208 human annotations and 1218 non-human images are used for training stage, and 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 [34].

Re-sample strategy is used in our experiment. 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.

Different from strategy in [1], Color normalization is not used in our feature. Although according to [1], using RGB information for gradient calculation will improve the performance by 1.5% at 10^{-4} false positives per window (FPPW). Only gray value is used for MB-HOT feature.

The comparison result can be seen in Fig. 11. The data of other features are copied from respective papers[1, 3, 35].

From Fig11, it can be seen that the Multi-scale HOT outperforms HOG[1] and Multi-resolution HOG[35]. The performance is nearly the same but a little better than COV[3]. Considering that the COV feature uses different training method and variable sub window strategy which can improve the result efficiently, and the computation complexity, we can say that our feature is better than COV feature.

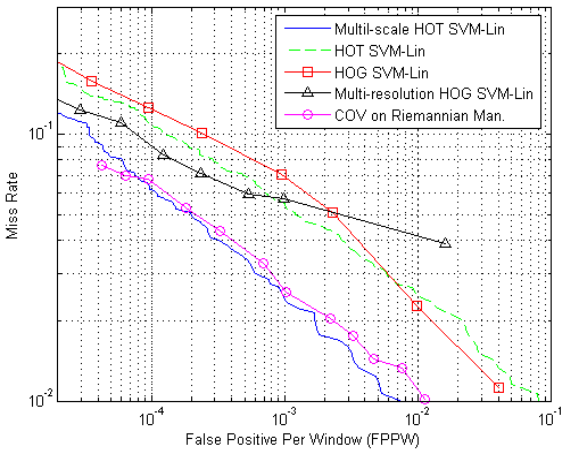

Fig.11 Comparison with other features. The curves are copied from respective papers

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

The CAS dataset [30] is selected for positive samples. This dataset is also used in [9] to get positive samples. It contains six different subsets of 12 people walking in six different directions.

The regions containing human is manually marked and extracted. The corresponding optical flow is calculated by current frame and last frames. The optical flow method in [31, 33] is used. The positive samples are divided into training dataset and testing dataset. The samples in testing dataset are not included in training dataset.

Some videos without human are also selected for negative samples. We manually mark the region containing other objects but not human. The optical flow is extracted.

At last, 3000 positive samples and 3000 negative samples are selected for training, which are scaled into 64 $\times$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.

We compare our feature with [8-10]. In [8] optical flow is used as feature directly. In [9] KLT coefficient is taken as feature. In [10], 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 Fig.12. It can be seen that our feature outperform the other motion based features.

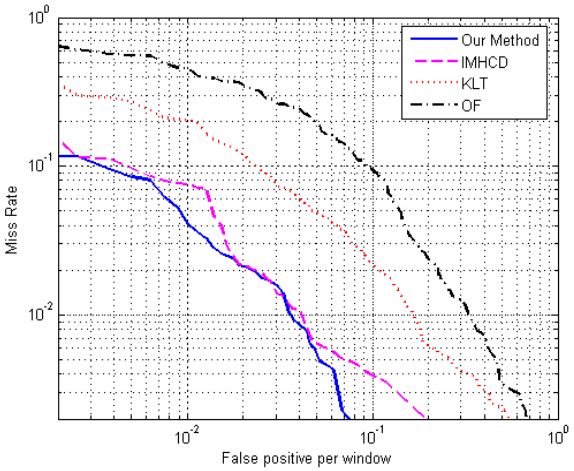

Fig.12 Comparison with IMHCD, OF and KLT feature

In third 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 Fig.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 very simple model to combine the result of 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 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.

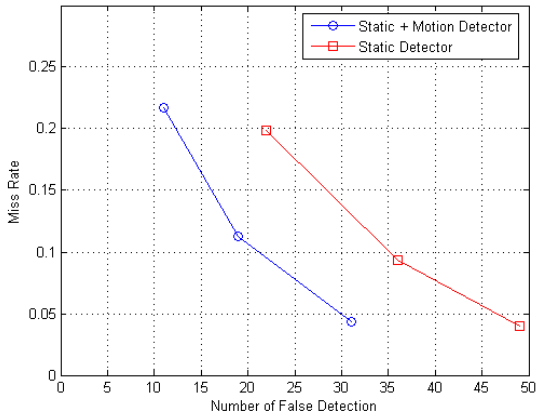

Fig.13 Comparison of static detector and static & motion detector

Some detection results can be seen in Fig.14.

Fig.14 Some detection results of video sequences

In fourth 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 [36] is used to get classifier. These support vectors are obtained offline and are loaded into texture memory before detection stage. The calculation time and memory consumption for each stage can be seen in Table.1. It can be seen that although sliding window strategy is used, MB-HOT feature can meet the real time requirement.

Table.1 Calculation time and memory consumption for each step (Microsecond / Mega byte)

Image Size	320×240	640×480
Down Scale	0.391/ 0.16	0.984 /0.63
Gradient	0.06 /0.16	0.16 /0.63
Status Table	0.26 /0.63	0.32 /2.52
Histogram	0.225/ 0.31	0.46 /1.26
Feature Calculation	6.2 /11.61	31 /83.32
Classification	2.1 /27.06	16/ 27.06
Overall	9.236 /39.93	48.924 /147.1

MB-HOT can return some candidate regions that may contain human. Then the motion based features are extracted to remove some false detection. The GPU based implementation of motion based feature will be used for verification. The feature extraction time for a 64×128 region is about 0.063ms.

6. Conclusion

In this paper, a robust human detection method is given. MB-HOT feature and motion based feature are proposed respectively. Experiments show that they can get higher detection rate in their own field. The motion based feature can reduce false detections returned by MB-HOT feature efficiently without bring too much computation. The GPU based implementation makes it possible for real time detection. For future work, the GPU based optical flow computation method will be integrated and the frame coherence will be used to make the detection more robust.

Acknowledgments

This research was supported by "Ambient SoC Global COE Program of Waseda University" of the Ministry of Education, Culture, Sports, Science and Technology, Japan, and CREST Program.

References

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

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

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

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

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

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

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

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

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

[15] C.Wojeck, *et al.*, "Sliding-windows for rapid object class localization: A parallel technique," presented at the DAGM, 2008.
[16] Z.li and N.Ramakant, "Efficient Scan-Window Based Object Detection using GPGPU," presented at the CVPR, 2008.
[17] S.Munder and D.M.Gavrila, "An experimental study on pedestrian classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 28, November 2006.
[18] B.Wu and R.Nevatia, "Detection and tracking of Multiple, partially occluded humans by bayesian combination of edgelet based part detectors," *IJCV,* 2007.
[19] P. Papageorious and T. Poggio, "A trainable system for object detection," *IJCV,* vol. 38, pp. 15-33, 2000.
[20] B. Scholkopf and A. Smola. (2002) Learning with kernels support vector machines, regularization, optimization and beyond.
[21] J.Friedman, *et al.*, "Additive logistic regression: a statistical view of boosting," *Ann. Stat.,* vol. 28, pp. 337-407, 2000.
[22] B.Leibe, *et al.*, "Pedestrian detection in crowded scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, 2005, pp. 878-885.
[23] B.Leibe, *et al.*, "Combined object categorization and segmentation with an implicit shape model," in *ECCV*, 2004, pp. 17-32.
[24] S.Agarwal and D.Roth, "Learning a sparse representation for object detection," in *ECCV*, 2002.
[25] E.Seemann, *et al.*, "Towards robust pedestrian detection in crowded image sequences," in *CVPR*, 2007.
[26] M.Andriluka, *et al.*, "People-tracking-by-detection and people-detection-by-tracking," presented at the CVPR, 2008.
[27] 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.
[28] D.M.Gavrila and V.Philomin, "Real-time object detection for smart vehicles," 1999, pp. 87-93.
[29] D.M.Gavrila, "The visual analysis of human movement: A survey," *Computer Vision and Image Understanding,* vol. 73, pp. 82-98, 1999.
[30] CAS [Online]. Available: http://www.cbsr.ia.ac.cn/english/index.asp
[31] A.S.Ogal and Y.Aloimonos, "Shape and the stereo correspondence problem," *International Journal of Computer Vision,* vol. 65, pp. 147-162, Dec. 2005.
[32] C.Zach, *et al.*, "A Duality Based Approach for Realtime TV-L1 Optical Flow," presented at the DAGM, 2007.
[33] A.S.Ogal and Y.Aloimonos, "A roadmap to the integration of early visual modules," *IJCV,* vol. 72, pp. 9-25, Apr. 2007.
[34] INRIA Dataset [Online]. Available: http://lear.inrialpes.fr/data
[35] Z.Wei, *et al.*, "Real-time accurate object detection using multiple resolutions," in *ICCV*, 2007.
[36] LibSVM [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/