# Learning Sparse Features in Granular Space for Multi-View Face Detection

Chang HUANG[1], Haizhou AI[1], Yuan LI[1]and Shihong LAO[2]
[1] Computer Science and Technology Department, Tsinghua University, Beijing 100084, China
[2] Sensing and Control Technology Laboratory, Omron Corporation, Kyoto 619-0283, Japan
E-mail: ahz@mail.tsinghua.edu.cn

## Abstract

*In this paper, a novel sparse feature set is introduced into the Adaboost learning framework for multi-view face detection (MVFD), and a learning algorithm based on heuristic search is developed to select sparse features in granular space. Compared with Haar-like features, sparse features are more generic and powerful to characterize multi-view face pattern that is more diverse and asymmetric than frontal face pattern. In order to cut down search space to a manageable size, we propose a multi-scaled search algorithm that is about 6 times faster than brute-force search. With this method, a MVFD system is implemented that covers face pose changes over +/-45° rotation in plane (RIP) and +/-90° rotation off plane (ROP). Experiments over well-know test set are reported to show its high performance in both accuracy and speed.*

## 1. Introduction

After a decade of efforts, face detection research [1] has achieved great advances, which has already found its ways in many real world applications including biometrics, visual surveillance, human-computer interaction, and even for the auto-focus of digital camera/video devices. Among those efforts, significant works include Rowley's ANN [2], Schneiderman's Bayesian decision rule [3], Viola's Boosted Cascade [4], etc. The break through is due to Viola's Boosted Cascade Framework whose remarkable performance owes to the fast speed of Haar-like feature calculation based on the integral image, the high accuracy of boosted strong classifiers, and the asymmetric decision making of the cascade structure.

Although frontal face detection is somewhat mature, Multi-View Face Detection (MVFD) is still a challenging problem that needs more attention since faces in images or videos of real world are seldom frontal. In the past few years, many derivatives of Viola's work have been proposed for MVFD. For comparison of those Viola's derivates, the hierarchy of Viola's detector is listed in Table 1. For the detector structure it is extended by Li's pyramid structure [5], Jones's decision tree [6] and Huang's WFS tree in [7]. On the level of learning strong classifiers, the discrete Adaboost was replaced by real Adaboost [8], Gentle Boost [9], and boosting chain learning [10]. For weak classifiers, the simple threshold-type function is replaced by finer partition of the feature space such as piece-wise functions [11] or joint binarizations of Haar-like features [12]. As for the feature level in particular, there are works of Lienhart's extended Haar-like feature set [13], Liu's Kullback-Leibler Boosting [14], Baluja's pair-wise points [15], Wang's RNDA algorithm [16] and Abramson's control point [17].

**Table 1.** Hierarchy of Viola & Jones' detector

| Level | Method |
|---|---|
| Detector Structure | Cascade |
| Strong Classifier | Adaboost |
| Weak Classifier | Threshold-type |
| Feature | Haar-like |

In this paper, we will focus our work on the feature level to pursue more fundamental and more effective features for MVFD to meet the rigor requirement of practical applications.

As we know for the feature level Viola's work uses Haar-like rectangle features of different types for the description of the complex frontal face pattern in a redundant manner. However, when the framework was extended to deal with multi-view faces whose pattern is more diverse and asymmetric, those compact and symmetric Haar-like features tend to be not so adaptable as they were with frontal faces due to the mismatch of their inherent characteristics, which has greatly hindered the further improvement of the performance of MVFD systems. This defect of Haar-like features has been pointed out by many people even for frontal face pattern and so far there are three approaches to overcome this problem though some of them are developed for frontal face detection problem.

The first approach is to specify a larger Haar-like feature set to include asymmetries, such as in [13] Lienhart proposed an additional rotated integral image that can fast calculate 45-degree rotated Haar-like features, and in [5] Li constructed looser and relatively shiftable rectangle features to cater to profile face pattern. However this method is limited in practice by its computational complexity and memory cost because of its brute force nature in search.

The second approach is to utilize complicated features. For example, Liu developed KL-Boosting to unify Haar-like features into a more discriminating KL feature [14], and Wang [16] adopted RNDA algorithm for the construction of complex optimal feature set without geometric constraints as Haar-like features. This kind of methods usually get small feature set for the final classifier with good generalization ability, but suffer from more computational load due to lot of floating operations involved in each feature calculation.

Oppositely, the third approach is to bring in even more simple and free definition of basic features, e.g. Baluja's pair-wise point [15] and Abramson's control points [17]. Both defined their features directly on the basis of pixels, which is the most fundamental element of an image, while the normalization of mean and standard deviation is discarded to avoid too much computational load. This change yields a double-edged sword, that is, the feature extraction becomes extremely fast as well as the point-based feature tends to be powerless for complicated task.

In this paper, we focus our work on the feature level by managing to learn sparse features in granular space for MVFD. On other levels in table 1, we follow the works in [11], that is, the piece-wise function for weak classifier, Real Adaboost algorithm for strong classifier, cascade structure for detector and view-based MVFD. In our approach, an expressive feature set that is defined as the linear combination of several granular features of different positions and scales, which we call sparse features, is introduced. Those sparse features can be calculated as efficiently as Haar-like features meanwhile they are capable to achieve higher discrimination power so as to guarantee better performance of MVFD system.

The remaining parts of this paper are organized as follow: section 2 introduces the sparse feature in granular space, section 3 describes the learning algorithm, section 4 gives the experiments and section 5 comes to the conclusion.

## 2. Definition of Sparse Feature in Granular Space

The sparse feature in our system is defined to be the linear combination of several granules as follows:

$$F(\boldsymbol{\pi}) = \sum_i \alpha_i p_i(\boldsymbol{\pi}; x, y, s), \quad \alpha_i \in \{-1, +1\} \tag{1}$$

where $\boldsymbol{\pi}$ is the gray level data of the input image and $p_i$ is a granule of the sparse feature. A granule is specified by 3 parameters: x-offset $x$, y-offset $y$, and scale $s$. For instance, $p_i(\boldsymbol{\pi}; x, y, s)$ indicates that the size of granule is $n$ by $n$, where $n$ is the $s$-th power of 2, and its left-top corner is at position $(x, y)$ in the reference window (Figure 1). For a $24 \times 24$ reference window, there are $\sum_s (24 - 2^s + 1)^2$ different granules in total, which are over-completed for the 576-dimensional space. This redundant granular feature set allows us to build up compact and powerful sparse features. In addition, to avoid floating operations, the combining coefficient $\alpha_i$ is restricted to be binary values, which further evades integer multiplications for computation efficiency.

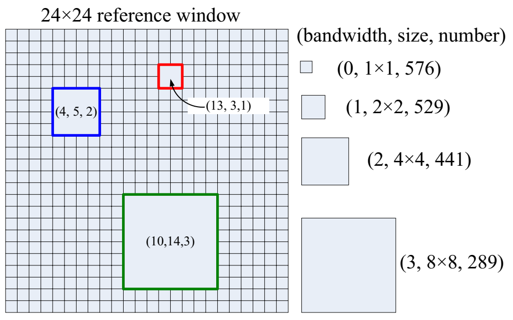

**Figure 1.** Sparse feature and its granules

In order to extract sparse feature values efficiently, a pyramid of bitmaps is constructed, in which each layer (bitmap) corresponds to the granules of a certain scale. We called it granular space, that is, in fact a special multi-scale space of the original image. Moreover, the integral image is still retained for the normalization to improve the robustness of the system.

Apparently, the sparse feature set is more diverse than Haar-like feature set and more compact than KL feature set or RNDA based feature set, which makes it possible to characterize multi-view face patterns in a more efficient way. In the next section, we will propose a novel learning algorithm for sparse feature based on heuristic search strategy.

## 3. Heuristic Search for Sparse Feature Selection

Having defined the feature set for the Adaboost algorithm, the remaining work to build a MVFD system is to select proper sparse features in granular space during Adaboost learning procedure to construct a strong classifier. Since the sparse feature set is defined as the combination of a number of granules (we choose 8 at maximum in our experiments), all possible combinations can not be enumerated due to its exponential complexity in nature. Therefore the conventional brute-force method that searches the whole feature space exhaustively as [4] is impossible. In order to solve this problem, we propose here a heuristic search algorithm, which efficiently selects a proper sparse feature for the training of the piece-wise function ([11]) in each round of Adaboost algorithm.

## 3.1 Heuristic Search

Heuristic search as it is stated in classical artificial intelligence theory has the general form shown in Figure 2. It starts with an empty closed list and a non-empty open list whose elements are used as promising initial seeds. The search process forwards by means of expanding a certain element in open list, and the closed list is used to keep those elements having been expanded to avoid duplicated expansion.

- Initialize:
  - Open list $OL = \{f_1, f_2, ..., f_n\}$, closed list $CL = \{\}$
- Loop:
  - Calculate fitness of each feature in *OL*.
  - Select the one with the highest fitness, denoting it as $f^*$
  - $OL = OL - \{f^*\}$, $CL = CL \bigcup \{f^*\}$.
  - Expand $f^*$ to construct a finite feature set $F^*$
  - $OL = OL \bigcup (F^* - CL)$
- Output:
  - Output the best feature in $OL \bigcup CL$

**Figure 2.** Heuristic search for feature selection

Applying the above general heuristic search to sparse feature selection in the granular space, we have to deal with three key issues: how to initialize seeds in open list, how to evaluate the fitness for feature selection and how to expand a sparse feature in a proper way.

## 3.2 Seeds Initialization

The initialization of seeds in open list is the first step of heuristic search. In practice, we enumerate a redundant Haar-like feature set under the constraints of sparse feature, that is, a feature unit is square rather than rectangle, and select some of the best ones as the initial seeds for heuristic search. The constraints make the enumerated Haar-like features in our approach is much less than those in [4] (only about twenty thousand).

The similarity between two sparse features can be quickly evaluated by the distance between them. The distance is defined as the maximum of minimum of granule to granule distances as equation 2, and the granule to granule distance is the Euclidian distance in a 3D space that takes their sizes as the third dimension as equation 3. With this measurement, the selected seeds could be kept a certain distance away from each other so as to avoid duplicated initialization and expansions in the later stage of heuristic search.

$$\varphi(F_1, F_2) = \max\left(\max_{p_i \in F_1}\left(\min_{p_j \in F_2}\left(\phi(p_i, p_j)\right)\right), \max_{p_j \in F_2}\left(\min_{p_i \in F_1}\left(\phi(p_i, p_j)\right)\right)\right) \quad (2)$$

$$\begin{aligned}\phi(p_1(x_1, y_1, b_1), p_2(x_2, y_2, b_2)) &= \Big[\left(x_1 + 2^{b_1 - 1} - x_2 - 2^{b_2 - 1}\right)^2 \\ &\quad + \left(y_1 + 2^{b_1 - 1} - y_2 - 2^{b_2 - 1}\right)^2 + \left(2^{b_1} - 2^{b_2}\right)^2\Big]^{1/2}\end{aligned} \quad (3)$$

## 3.3 Fitness Evaluation

Fitness evaluation is used to select the most potential sparse feature in open list for further expansion to acquire better features. Three aspects must be concerned in the fitness evaluation of a sparse feature: complexity, age, and discriminability.

$$\text{Fitness}(F_i) = \psi(complexity, age, discriminability) \quad (4)$$

The complexity is characterized by the number of granules adopted in the sparse feature. In general a sparse feature with low complexity should be preferred since it is more efficient to compute and intuitively it may have lower structural risk. The age of a sparse feature was initialized to be 0 when it was added into the open list and is increased by 1 for each loop. Since the expansions in heuristic search is essentially constructing a branch of a search graph whose nodes correspond to sparse features, the parameter, age, can be utilized to balance the width and depth of the graph. In our experiments, the expansion of elder features is encouraged. In this way, the search tree keeps to be wide enough to alleviate the local minimum problem. At last, if adopting piece-wise function based weak classifiers as [11], the discriminability of a sparse feature is defined as:

$$D(F_i) = 1.0 - Z = 1.0 - 2\sum_j \sqrt{W_+^j W_-^j} \quad (5)$$

where $W_+^j$ is the total weight of positive samples that fall into the $j$-th bin while $W_-^j$ is that of negative samples. The second term of equation 5 is actually equivalent to the normalization factor in Adaboost that holds the upper bound of training error if adopting piece-wise functions [8][11]. Consequently, a sparse feature $F_i$ that gets higher $D(F_i)$ is more discriminative in Adaboost framework.

How to integrate these three parameters still remains an open question, and we adopt an empirical function for the overall evaluation of feature $F_i$ as

$$\text{Fitness}(F_i) = D(F_i) \times 0.985^c \times \left(0.5e^{-0.3a} + 0.5\right) \quad (6)$$

where $c$ is the adopted granule number and $a$ is the age

### 3.4 Expansion Operators

To expand a selected sparse feature, we developed three operators: add, remove and refine. An illustration is shown in Figure 3 as an example.

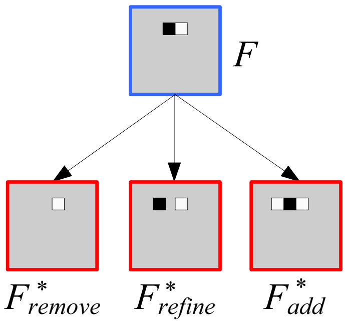

**Figure 3.** Three types of expansion operators (White square is the positive granule while black one is negative. *Remove* is to delete a current granule in *F*，*add* is to insert a new one, and *refine* is to replace a current one with one of its neighbors)

Denoting the original feature as *F*, its granule set as *P*, a granule in *P* and its corresponding coefficient as $p_{in}$ and $\alpha_{in}$, another granule that does not belong to *P* and its corresponding coefficient as $p_{out}$ and $\alpha_{out}$, and the neighboring granule set of $p_{in}$ as $Nb(p_{in})$, these three operators can be formulized as follow.

$$F_{add} = F + \alpha_{out} p_{out}, \quad p_{out} \notin P, \alpha_{out} = \{-1, +1\} \tag{7}$$

$$F_{remove} = F - \alpha_{in} p_{in}, \quad p_{in} \in P \tag{8}$$

$$F_{refine} = F - \alpha_{in} p_{in} + \alpha_{out} p_{out}, \quad p_{in} \in P, p_{out} \in Nb(p_{in}), \alpha_{out} = \alpha_{in} \tag{9}$$

Since there are many candidates of $p_{in}$ and $p_{out}$ in the reference window, in fact each expansion operator builds up a set of new sparse features, and only the best one (i.e. with the highest fitness) is chosen to be the final expanded result of this operator.

### 3.5 Multi-Scaled Search

Since the add operator and refine operator could generate thousands, or even tens of thousands of new features, how to choose a proper feature from such an enumerated set is a crucial problem. If adopting the conventional brute-force approach, the computational load is exactly the product of sparse feature space size and the training sample space size because each sparse feature has to be computed on every training sample (Figure 4). As a result, the feature selection procedure will become too time-consuming ever before the search graph is expanded enough to find a good sparse feature.

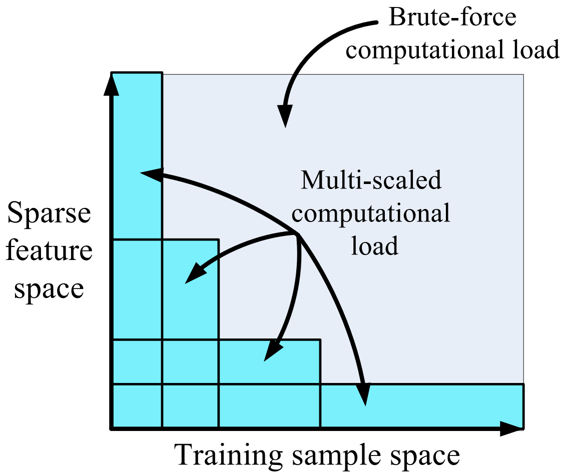
Brute-force computational load
Multi-scaled computational load
Sparse feature space
Training sample space

**Figure 4.** Multi-scaled selection vs. Brute-force selection on computational load

However, for the feature selection task, we could firstly adopt only a small part of the training samples to evaluate all features and reject some of the features, and then adopt more samples to reject some of the remaining features again, and so on (Figure 4). Based on this idea, we developed a fast feature selection algorithm in the multi-scaled sample space as Figure 5.

```
● Initialize:
  – Sparse feature set $F = \{f_1, f_2, ..., f_m\}$
  – Training sample set $S = \{s_1, s_2, ..., s_n\}$
  – Scale ratio $k$
● Preprocess:
  – Partition training samples into a series of nested sub sets:
      $S_0 \subset S_1 \subset \cdots \subset S_k = S, \quad |S_{i+1}| = 2|S_i|$
  – Let $F_0 = F$
● For ($i = 0$; $i < k+1$; $i$++)
  – Calculate fitness value of each feature in $F_i$ using sample set $S_i$
  – Divide $F_i$ into two parts with equal size:
      $F_i^+ \cup F_i^- = F, \ F_i^+ \cap F_i^- = \varnothing, \ |F_i^+| = |F_i^-|$
      $\forall f^+ \in F_i^+, \forall f^- \in F_i^-, \text{Fitness}(f^+) \ge \text{Fitness}(f^-)$
  – Let $F_{i+1} = F_i^+$
● Output:
  – Select the best one in $F_k$,
      $f^* = \arg\min_{f \in F_k} (\text{Fitness}(f))$
```

**Figure 5.** Algorithm of fast feature selection in multi-scaled sample space

After the evaluation in each round, the feature set is reduced by a half (the better half remains). In this way, the feature set size is decreasing rapidly while the utilized sample set size is increasing. This fast feature selection algorithm works on the basis that only strong features can survive until the end. Supposing originally the sample set size is $n$ and the feature set size is $m$, the brute-force approach requires $m \times n$ times of calculation. On the other hand, in the fast feature selection algorithm, the number of calculation becomes

$$T_{fast} = \sum_{i=0}^{k} |S_i||F_i| = 2^{-k}(k+1)mn \qquad (10)$$

Typically we choose $k$ = 7, then $T_{fast}$ = 0.0625$mn$. The computation load is significantly reduced compared to brute-force approach. In experiments, due to other additional computation cost, the weak learning process of multi-scaled sample space approach is about 6 times faster than that of brute-force approach, which results a training load of our MVFD system comparable to that of Haar-like feature based system.

## 4. Experiment on MVFD

Following Wu's parallel cascade framework in [11], we divide the faces into five categories according to the left-right ROP angle, which are left full profile, left half profile, frontal, right half profile and right full profile, covering [-90°, -50°], [-50°, -20°], [-20°, +20°], [+20°, +50°] and [+50°, +90°] respectively. For each category, the range of RIP [-45°, 45°] is partitioned into 3 sub categories. In this way, we manage to create a view-based MVFD that covers RIP [-45°, +45°] and ROP [-90°, +90°], which contains 15 different cascade detectors. Due to the horizontal symmetry between left and right faces, only 8 cascades have to be trained, and the other 7 ones can be got by the horizontal flip of the trained cascades (Figure 6). Each cascade detector contains 1382 weak classifiers in 18 layers. The main difference between ours and [11] is in the feature level in Table 1, that is, sparse features rather than Haar-like features.

For the training of different views, we collected about 30000 frontal faces, 25000 half profile ones and 20000 profile ones, normalized them by the labeled eyes and mouth. Some of them are shown in Figure 7.

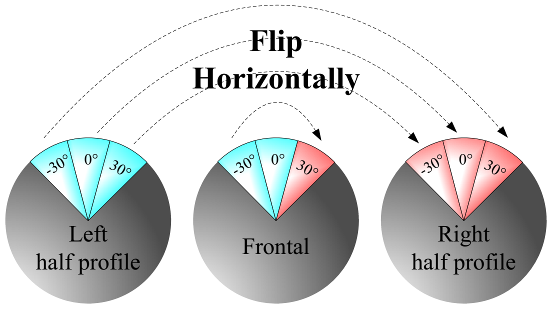

**Figure 6.** Construction of view-based MVFD (full profile views are omitted here)

**Figure 7.** Some training samples

To compare with the previous works ([3][7][11]) on MVFD, we test on the CMU profile set. The results are shown in Table 2 and Figure 8, in which our sparse feature based MVFD system achieves the best performance. The first sparse feature found by the heuristic search for each of the 8 trained cascades is given in Figure 9. Some detection results of our MVFD are given in Figure 10.

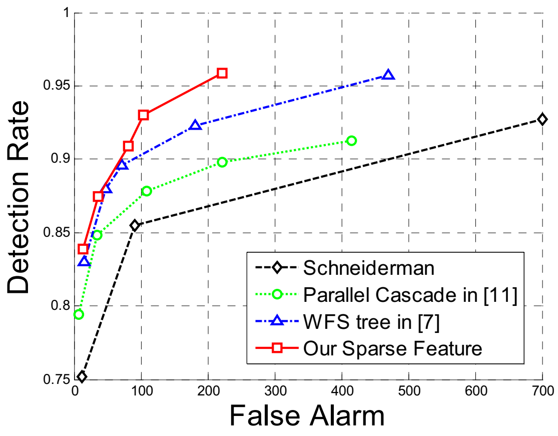

**Figure 8.** ROC curves on CMU profile testing set

**Table 2.** MVFD results on CMU profile face test set. (Totally 208 images, 441 faces)

| Methods | detection rate (false alarms) | | | | |
|---|---|---|---|---|---|
| Schneiderman [3] | 75.2% (12) | 85.5% (91) | 92.7% (700) | - | |
| Wu's Parallel Cascade [11] | 79.4% (8) | 84.8% (34) | 87.8% (109) | 89.8% (221) | 91.3% (415) |
| Huang's WFS tree [7] | 83.0% (16) | 88.0% (48) | 89.6% (72) | 92.3% (181) | 95.7% (470) |
| Our Sparse Feature | 83.9% (14) | 87.5% (36) | 90.9% (82) | 93.0% (104) | 95.9% (222) |

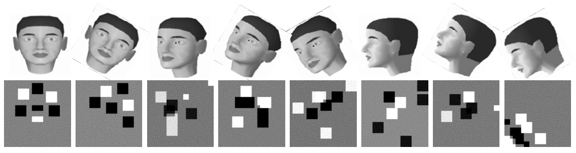
**Figure 9.** First sparse feature in each of the eight trained cascades

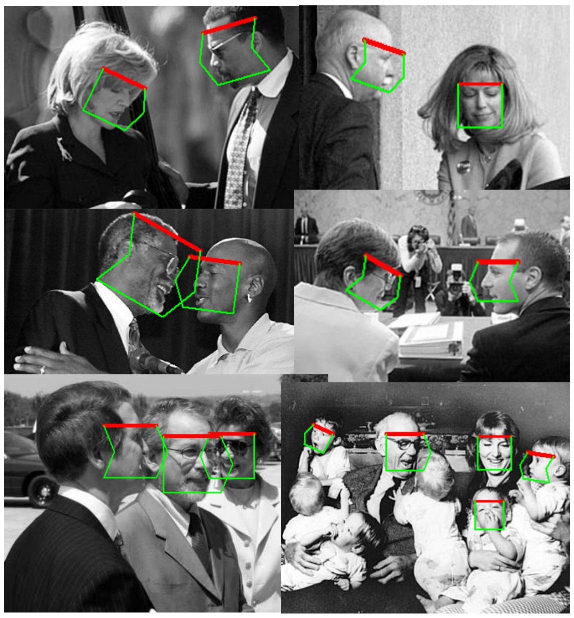
**Figure 10.** Detection results on CMU profile set

On a Pentium4 2.8GHz PC, our MVFD takes about 90ms in QVGA size (320×240) image averagely.

In this experiment, the number of granules in each sparse feature is 8 at maximum. Since each granule needs to access memory only once while the basic unit of Haar-like feature, rectangle, requires four times for its four corner points, so the sparse features are computation efficient just as Haar-like features yet they are more powerful to represent more complicated patterns (see Figure 9 for an example). Compared to the KL-features [14] and the unconstrained features based on the RNDA algorithm [16], the sparse features are much more regular and easy to compute. As for the extremely fast feature extraction approaches such as pair-wise points [15] and control points [17], they are somewhat too weak to build a high performance MVFD system..

## 5. Conclusion

In this paper, we define a novel sparse feature set in granular space to handle the problem of diversity and asymmetry existed in multi-view face patterns, and then we propose a heuristic search strategy to find a proper sparse feature for the weak learning of Adaboost. Instead of brute-force approach, a fast selection algorithm based on multi-scaled sample space is developed to significantly speed up the search process of sparse features. Finally, following [11]'s parallel cascade structure, a real-time MVFD system is implemented that has achieved better performance on CMU profile testing set than that of the previous works [3][7][11].

## 6. Acknowledgements

This work is supported mainly by a grant from OMRON Corporation. It is also supported in part by National Science Foundation of China under grant No.60332010.

## 7. Reference

[1] M. H. Yang, D. J. Kriegman and N. Ahuja, "Detecting Faces in Images: A Survey", PAMI 2002, VOL 24, NO.1

[2] H. A. Rowley, "Neural Network-based Human Face Detection", Ph.D. thesis, Carnegie Mellon University, May 1999.

[3] H. Schneiderman and T. Kanade, "A Statistical Method for 3D Object Detection Applied to Faces and Cars". CVPR 2000.

[4] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", CVPR 2001.

[5] S. Z. Li, L. Zhu, Z. Q. Zhang, et al., "Statistical Learning of Multi-View Face Detection". ECCV 2002.

[6] M. Jones and P. Viola, "Fast Multi-view Face Detection". MERL-TR2003-96, July 2003.

[7] C. HUANG, H. Z. AI, Y. Li and S. H. LAO, "Vector Boosting for Rotation Invariant Multi-View Face Detection", ICCV 2005

[8] R. E. Schapire and Y. Singer, "Improved Boosting Algorithms Using Confidence-rated Predictions", Machine Learning, 37, 1999, 297-336.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. Annals of Statistics, 2000, 28: 337~374

[10] R. Xiao, L. Zhu, H. Zhang, Boosting Chain Learning for Object Detection, ICCV 2003.

[11] B. Wu, H. Ai, C. Huang, and S. Lao. "Fast rotation invariant multi-view face detection based on Real AdaBoost". *Proc. of IEEE Conf. on Automatic Face and Gesture Recognition*, pages 79–84, 2004.

[12] T. Mita, T. Kaneko, O. Hori, "Joint Haar-like Features for Face Detection", ICCV 2005

[13] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", *ICIP 2002*

[14] C. Liu and H. Y. Shum, "Kullback-Leibler Boosting", *Proc. Of CVPR*, pages 587–594, 2003.

[15] S. Baluja, M. Sahami and H. A. Rowley, "Efficient Face Orientation Discrimination", ICIP 2004

[16] P. Wang and Q. Ji, "Learning Discriminant Features for Multi-View Face and Eye Detection", CVPR 2005.

[17] Y. Abramson and B. Steux, "YEF* Real-Time Object Detection", International Workshop on Automatic Learning and Real-Time 2005