Real-Time Multi-View Face Detection

ZhenQiu Zhang[1*], *Long Zhu*[2], *Stan Z. Li*[2], *HongJiang Zhang*[2]
1. Institute of Automation, Chinese Academy of Science, Beijing, China
2. Microsoft Research Asia, Beijing Sigma Center, Beijing 100080, China

Abstract

In this paper, we present a detector-pyramid architecture for real-time multi-view face detection. Using a coarse to fine strategy, the full view is partitioned into finer and finer views. Each face detector in the pyramid detects faces of its respective view range. Its training is performed by using a new meta booting learning algorithm. This results in the first real-time multi-view face detection system which runs at 5 frames per second for 320x240 image sequence.

1. Introduction

Statistics show that approximately 75% of the faces in home photos are non-frontal [1], and therefore ability to deal with multi-view faces is important for many face-related applications. Multi-view face detection has been a challenging problem. The challenge is firstly due to large amount of variation and complexity brought about by the changes in facial appearance, lighting and expression [17]. Changes in facial view (pose) further complicate the situation because the distribution of multi-view faces in a feature space is more dispersed and more complicated than that of frontal faces.

The learning based approach has been most effective for face detection. Sung and Poggio [14] divided the frontal face image space and non-face image space each into several probability clusters. PCA is performed on each cluster so that face/non-face classification is performed in terms of both the Mahalanobis distance from the cluster center in the PCA space and the Euclidean distance from the PCA space are used as features. Rowley et al [9] presented a face detection system based on retinally connected neural networks. The input to the NN is the preprocessed image pixel values directly. Post-processing of the neural networks are performed by either ANDing/ORing the outputs or using an additional neural network to arbitrate between the outputs. Osuna el al [5] applied the support vector machines algorithm to train an NN to classify face and non-face patterns. Yang el al [3] uses a network of linear units. The SNoW learning architecture is specifically tailored for learning in the presence of a very large number of features.

Recently, Viola and Jones [19] propose a very fast approach for frontal face detection. Simple Haar-like feature are extracted, face/non-face classification is done by using a cascade of successively more complex classifiers which are trained by using AdaBoost [24] learning algorithm. The cascade structure is supported by an argument made in [4] that cascading classifiers is a better approach than multi-expert methods like voting and stacking.

Over past years, progress has been made for non-frontal faces detection and recognition. Feraud et al [11] adopt the view-based representation for face detection, and use an array of 5 detectors with each detector responsible for one view. Wiskott et al [16] build elastic bunch graph templates for multi-view face detection and recognition. Gong and colleagues [21] study the trajectories of faces in linear PCA feature spaces as they rotate, and use kernel support vector machines (SVMs) for multi-pose face detection and pose estimation. Huang et al [13] use an SVM to classify three facial poses at -33.75, 0, +33.75 degrees.

To deal with complexity due to multi-view, a natural treatment is to divide face images into several subsets according to the facial view and model each view subspace respectively [2], by which explicit 3D modeling is avoided.

The system of Schneiderman and Kanade is claimed to be the first one in the world for multi-view face detection [10]. The algorithm consists of an array of 5 face detectors each of which is specialized for a specific pose of face and accommodates small amount of variation around the designated pose. A detector classifies a sub-window into face/non-face based on statistics of products of histograms computed from examples of the respective view. The results from all the detectors are merged such that they are spatially consistent. The detector is claimed to be the first algorithm in the world for multi-view face detection. However, it is very slow and takes 1 min to work on a 320x240 image over only 4 octaves of candidate size [10].

In this paper, we present a novel framework for real-time multi-view face detection. A detector-pyramid architecture is designed to detect multi-view faces efficiently. The detector-pyramid adopts an integrated strategy of coarse-to-fine view decomposition [18,19], and simple-to-complex face/nonface classification Viola and Jones [19]; a sub-window is processed from the top to bottom of the pyramid by a sequence of increasingly more complex face/non-face classifiers designed for increasingly finer ranges of facial view. The detector-pyramid goes beyond the straightforward view decomposition method [2] in that using the coarse-to-fine and simple-to-complex strategy, a vast number of

---

* The work presented in the paper was carried out at Microsoft Research Asia.

nonface sub-windows can be discarded very quickly with very little lose of face sub-windows. This is very important for fast face detection because only a tiny proportion of sub-windows are of faces.

We devise simple image features for efficient face/nonface classification. These features are extensions of those used in [19] for frontal face detection in that the former is more suitable to cater to non-symmetry of non-frontal faces.

Every detector in the pyramid is learned from face/nonface examples using a new learning algorithm called FloatBoost [22]. FloatBoost incorporates the idea of Floating Search [18] into AdaBoost to solve the non-monotonicity problem encountered in the sequential algorithm of AdaBoost.

While the Viola-Jones detector [19] is the first real-time frontal face detector and Schneiderman-Kanade detector is the first (non real-time) multi-view face detector, the algorithm presented in this paper results in the first real-time multi-view face detection system which runs at 5 frames per second for 320x240 image sequence on a conventional 700 MHz Pentium III PC.

The rest of the paper is organized as follows: Section 2 introduces the detector-pyramid architecture for multi-view face detection. The design and training of individual detector are presented in section 3 and 4. Method to arbitrate among nine view channels is presented in section 5.Section 6 provides the experimental results and conclusion is drawn in section 7.

2. Detector-Pyramid Architecture

The present multi-view face detection system is distinguished from previous systems in its ability to detect multi-view faces in real-time. It is designed based on the following thoughts: While it is extremely difficult to distinguish multi-view faces from non-face images clearly using a single classifier, it is less difficult to classify between frontal faces and non-faces and also less difficult to do between multi-view faces and part of non-faces. Therefore, narrowing down the range of view will make face detection easier and more accurate for that view.

On the other hand, a vast number of sub-windows (e.g. 70,401 square sub-windows can result from the scan of a 320x240 image, from the size of 20x20 pixels to 240x240 for the size increment factor of 1.25) result from scan of the input image; among these only a tiny proportion (say, up to a few dozens) of them are faces. It can save the computation tremendously if a sequence of detectors of increasing complexity and face/non-face discriminating power are applied to quickly discard non-faces at the earliest possible stage using the simplest possible features.

The detector-pyramid architecture (see Figure 1) is motivated by the above reasons. It adopts the coarse to fine (top-down in the pyramid) strategy [18,19] in that the full range of facial view is partitioned into increasingly narrower ranges, and thereby the whole face space is partitioned into increasingly smaller subspaces. Also it adopts the simple-to-complex strategy (Viola-Jones detector [19]) in that the earlier ones are simpler and so are able to reject a vast number of non-face sub-windows q1uickly whereas the ones in the later stage are more complex and involved and spend more time to scrutinize only a relatively tiny number of remaining sub-windows.

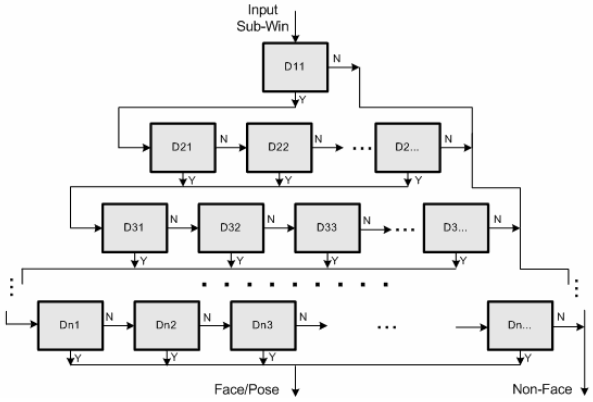

Figure 1: Detector-pyramid.

Our current implementation consists of three levels. The first level consists of a single detector, responsible for the full range of [-90,90] degree (0 degree being the frontal; view). There are three detectors in the second level, responsible for the three view ranges [-90, -40], [-30,+30], [+40,+90], respectively. The third level consists of 9 detectors of [-90, -80],[-70,-60], …, [60,70], [80,90] degrees. Therefore, there are a total of 13 detectors.

For a sub-window, if it is rejected by detector at the top level, it will be seen as non-face region and will not be processed by later levels. If it goes through first level, it will be processed by second level. If any detector in second level classifies it as face, it will be processed by last level, or it will be rejected as non-face. There are much more detectors on the bottom of our framework, and it help us focus our attention on those possible face region, while paying much less time on impossible face region. At the last level, each detector only dues with 20 degree ranges of view and each detector has high detection rate for that view. This pyramid-like framework makes our system have both high detection rate and rapid detection speed for multi-view face detection.

The full-view detector in the implementation is able to reject about 50% of non-face sub-windows scanned in the performing stage, while retaining 99% of training face examples in the training stage. Only retained sub-windows possibly containing faces are further processed in the subsequent levels of finer detectors. The results from the detectors in the bottom level are merged to make a final decision regarding the input sub-window.

3. Design of Individual Detectors

The high speed and detection rate of the algorithm depend not only on the detector-pyramid architecture but also individual detector. Each detector classifies a sub-window into face/non-face. Two types of simple features,

which are block differences similar to steerable filters, are computed as shown in Figure 2. Each such feature has a scalar value which can be computed very efficiently from the summed-area table [6] or integral image [19]. These features are non-symmetrical to cater to nonsymmetrical characteristics of non-frontal faces. They have more degrees of freedom than those of [19] in their configurations: 6 (x, y, delta x, delta y, dx, dy) in the two block features and 7 (x, y, delta x, delta y, dx, dx', dy) in the three block features. There are a total number of 102,979 two-block features for a sub-window of size 20x20 pixels. There are a total number of 188,366 three-block features (with some restrict to their freedom).

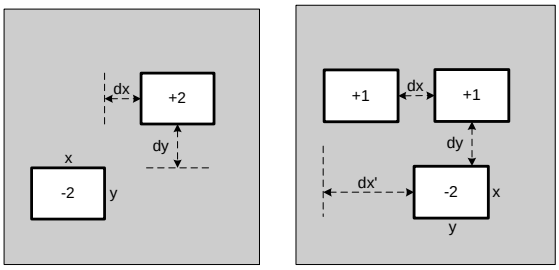

Figure 2: The two types of simple Harr wavelet like features defined in a sub-window. The rectangles are of size x by y and are at distances of (dx,dy) apart. Each feature takes a value calculated by the weighted +-1,2 sum of the pixels in the rectangles.

A face/nonface strong classifier is constructed based on a number of weak classifiers where a weak classifier performs face/non-face classification using a different single feature, e.g. by thresholding the scalar value of the feature according the face/non-face histograms of the feature. A detector can be one or a cascade of such face/nonface strong classifiers, as in [19].

4. Training of Individual Detectors

How to choose a good combination of weak classifiers from tens of thousands of features to construct a powerful detector is a challenging problem of feature selection [8][15] and classifier design. We have devised a new boosting algorithm, called FloatBoost [22], for learning face detectors for the detector-pyramid. Similar or better performance than AdaBoost is achieved with fewer weak classifiers.

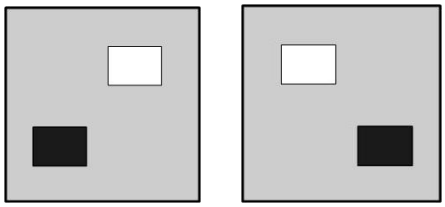

Figure 3: Mirroring feature. On the left is a feature learned for a left view detector. On the right is the corresponding feature mirrored for the right view counterpart.

The detectors in the pyramid are trained separately, using different training sets. An individual detector is responsible for one view, with possible partial overlapping with its neighboring detectors. Due to the symmetry of faces, we need to train side view detectors for one-side only, and mirror the trained models for the other side. For one feature used in left-side view, we mirror its structure (See Figure 3) to construct a new feature used for right-side view. Each left-side view feature is mirrored by this way, and these new features are combined to construct right side view detectors.

5. Arbitrate among Individual Outputs

In our framework, we have nine channels at the last layer; each channel represents one facial view. To arbitrate among these nine detectors we use some heuristic methods.

Firstly, we combine the output of some view ranges into one class. After combination, nine channels of view are converted to five channels (left profile, left half-profile, frontal, right half-profile and right profile). For example, we combine $[-90^o, -60^o]$ as left half-profile. Then, we arbitrate outputs within these five view poses. We use Rowly's heuristic method. We clean-up outputs of each detector. See figure 4, A is the last output of front face channel, and only frontal faces are detected by this channel. B is the last output of half-profile channel. This channel in fact includes two channels: right half-profile channel and left half-profile channel. Some frontal faces will be detected by this channel because half-profile detectors will detect part of frontal face as half-profile face (See Figure 5-B). C represents the last output of profile channel, and this channel includes two channels: right profile, left profile too.

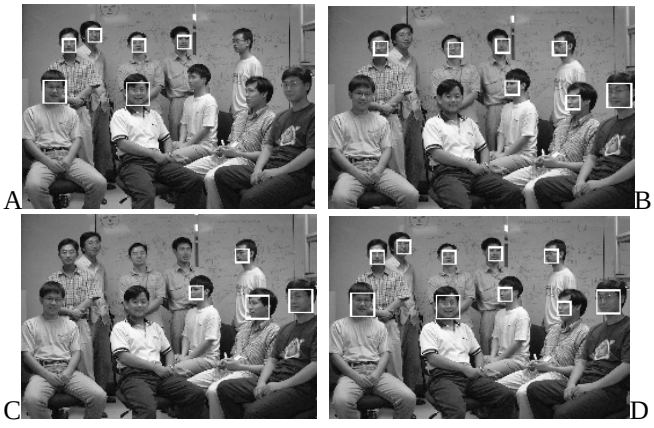

Figure 4: Output of fontal (A), half-side (B), and full-side (C) view channels, and the final result (D) after post-processing.

To arbitrate among five channels, we present a novel heuristic method. In practice, we find half-profile detectors and profile detectors often detect part of the frontal face as half-profile or profile face. So we prescribe that if a particular location is identified as a frontal face, then all other locations detected by profile or half profile face detectors which overlap it are likely to be errors, and can

therefore be eliminated. Similarly, if a particular location is identified as half-profile face, then all other locations detected by profile face detectors are eliminated. D is the last output arbitrating among five channels (see figure 5). We can find that some faces in B (detected by half-profile channel) overlap part of faces in A (detected by frontal channel). We identify these faces in B which overlaps with faces in A as errors, and eliminate them in the last output (in D). By the same, half-profile and profile channels have some overlaps too, and we eliminate faces detected by profile channel, which overlap with faces by half-profile channel.

6. Experimental Results

This section describes the final face detection system including training data preparation, training procedure, and the performance comparison with previous view-based multi-view face detection system.

6.1 Training Data Set

More than 6,000 face samples are collected by cropping from various sources (mostly from video). The view is in the range of $[-90^o, 90^o]$ with -90^o representing the left-side view and 0^o representing the frontal view. A total number of about 25,000 multi-view face images are generating from the 6,000 samples by artificially shifting or rotation. In our system, we partition multi-view face space into smaller and smaller (top-down in the pyramid) subspaces of narrower view ranges. At the top layer, there is only one detector. So all face sample are grouped into one class. At the second layer, there are three detectors, and face samples are grouped into three view classes (frontal, left-profile and right-profile). Face samples labeled with $-20^o, -10^o, 0^o, 10^o, 20^o$ are grouped as frontal faces, those with $[-90^o, -30^o]$ are grouped as left-profile face and the faces with $[30^o, 90^o]$ are grouped as right-profile faces. At the third layer, there are nine detectors, and face samples are grouped into nine view classes of [-90, -80], [-70, -60], …, [80,90] degrees.

Figure 5: Multi-view face examples

6.2 Training phase

There are 13 detectors in our system, but we only need train eight detectors. The right view detectors at the second and third levels can be constructed by mirroring features used in left view detectors. This method saves about half training time for our system. These detectors are trained separately, using their own training data. Non-face images used for training these detectors are collected from 12,000 images which don't contain face.

Every detector can be a cascade of strong classifiers and this guarantees high detection speed. At the top level, the detector is trained using all the faces from -90^o to 90^o. It has a cascade of three strong classifiers structure. The number of features in these three strong classifiers is 5, 13 and 20 respectively. It can reject about 50% non-face training data, while retaining 99% face train data in training stage.

At second level, there are three detectors, each of which is trained to detect part range of the full-view faces. Training faces are separated into three classes to train these detectors. At this level, each detector has a cascade of six strong classifiers structure. In our system, this level can totally rejects about 97% non-face training data which go through top level, and retain 98% face train data in training stage.

At bottom level, face training data is separated into nine classes. At this level, each detector is a cascade of about twenty strong classifiers structure. Each detector has a detection rate of about 94%, and achieves a false positive rate of about 4×10^{-6}.

6.3 Detection Results

The final detector is scanned across the image at multiple scales and locations. Scaling is achieved by scaling the detectors themselves, rather than scaling the image. This process makes sense because the features can be evaluated at any scale with the same cost. We scale the detectors using a factor of 1.25. In Figure 4, the image is 320 by 240 pixel size. There are a total of 70,401 sub-windows to be verified in this image. The full-view detector at the top level needs 110 ms to process all these sub-windows. About 40% sub-windows from test image are rejected by this coarse classifier, and only 41,114 sub-windows can pass through this classifier. At the second level, there are three detectors. They totally need 77 ms to process all the rest sub-windows. About 97% sub-windows of the 41,114 sub-windows are rejected by this level, and only 1298 sub-windows pass through this level. At the third level, there are nine detectors. They process all these 1298 sub-windows. But they only need 15 ms to do it, because most sub-windows are rejected at first and second levels. The timing is summarized in Table 1.

Level	First	Second	Third	Total
Time	110ms	77ms	15ms	202ms

Table 1: Times needed for each level to run the 320*240 image.

Because spend 15 ms is needed for the third level, so it will not affect the efficiency much of the whole system if we partition multi-view face space into smaller subspaces of narrower view ranges at the third level. That it to say (now we have nine detectors on the third level), if we decompose multi-view face space into smaller subspaces (for example:

19 view ranges), this system will still has high detection speed, but the detection rate will probably be increased.

Method	View-based	Detector-Pyramid
Time	976ms	202ms

Table 2: Comparison between the view-based and detector-pyramid architecture in speed for multi-view face detection.

If we had not adopted the pyramid-like framework presented in this paper, we can apply all these nine detectors at the third level directly on all sub-windows without coarse classification at the top and second levels. This method will (we call it view-based) cost much time for multi-view face detection (see Table 2).

Our system is tested on CMU profile face test set. This test set consists of 208 images with 441 faces of which 347 were profile views from various news web sites. These images were not restricted in terms of subject matter or background scenery. They were collected from various news web sites. The database can be downloaded at http://vasc.ri.cmu.edu/idb/html/face/profile_images/index.html. We present some results shown in Fig 6. We also provide a video clip showing multi-view face detection at http://research.microsoft.com/~szli/Demos.

Figure 6: Examples of Detection Results

7. Conclusions

In this paper, we have presented a detector-pyramid architecture for multi-view face detection. Using a coarse-to-fine and simple-to-complex scheme, our system solves the problem effectively and efficiently by discarding most of non-face sub-windows using the simplest possible features at the earliest possible stage. This leads to the first real-time multi-view face detection system.

Given this framework demonstrates good performance in multi-view face detection, we stress that the underlying architecture is fairly general and can be applied to other appearance based object detection problem.

REFERENCE

[1] A. Kuchinsky, C. Pering, M.L. Creech, D. Freeze, B. Serra and J. Gwizdka. “Consumer Multimedia Organization and Retrieval System”. Proceedings of ACM SIG CHI'99 Conference.

[2] A. P. Pentland, B. Moghaddam and T. Starner. “View-based and modular eigensapces for face recognition”. In CVPR, pages 84-91, 1994.

[3] D. Roth, M.-H. Yang and N.Ahuja. “A SNoW-Based Face Detector”. NIPS'00.

[4] E Alpaydin and C Kaynak. “Cascading Classifiers”. Kykernetika, 34(4), 369-374.

[5] E. Osuna, R. Freund, and F. Girosi. “Training support vector machines: An application to face detection”. In CVPR, pages 130--136, 1997.

[6] F. Crow. “Summed-area tables for texture mapping”. In Processings of SIGGGRAPH, volume 18(3), pages 207-212, 1984.

[7] F. Fleuret and D. Geman. “Coarse-to-fine face detection”. Inter. Journal of Computer Vision, 2001.

[8] G.H. John, R. Kohavi, and K. Pfleger. “Irrelevant features and the subset selection problem”. In Processings of the Eleventh International Conference on Machine Learing, Pages 121-129,1994.

[9] H.A. Rowley, S.Baluja, and T.Kanade. “Neural network-based face detection”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1):23--28, 1998.

[10] H. Schneiderman and T. Kanade. “A statistical method for 3d object detection applied to faces and cars”. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2000.

[11] J. Feraud, O. Bernier, and M. collobert. “A fast and accurate face detector for indexation of face images”. In Proc. Fourth IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 52-59, 1998.

[12] J. Friedman, T. Hastie, and R. Tibshirani. “Logistic regression: a statistical view of boosting”. Technical report, Department of Statistics, Sequoia Hall, Stanford, Univerity, July 1998.

[13] J. Huang, X. Shao, and H. Wechsler. “Face pose discrimination using support vector machines (SVM)”. In

Proceedings of International Conference Pattern Recognition, Brisbane, Queensland, Australia, 1998.
[14] K.K. Sung and T. Poggio. “Example-based learning for view-based human face detection”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1):39--51, 1998.
[15] K. Tieu and P. Viola. “Boosting image retrival”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000.
[16] L. Wiskott, J. Fellous, N. Kruger, and C. V. malsburg. “Face recognition by elastic bunch graph matching”. IEEE Transactions on Pattern Analysis and Machine Intelligence,19(7):775-779, 1997.
[17] M. Bichsel and A.P. Pentland. “Human face recognition and the face image set’s topology”. In Image Understanding, Volume 59, pages 254-261, 1994.
[18] P. Pudil, J. Novovicova, and J. Kittler. “Floating search methods in feature selection”. Pattern Recognition Letters, 1119--1125, 1994.
[19] P. Viola and M.J. Jones, “Robust real-time object detection”, IEEE ICCV Workshop on Statistical and Computational Theories of Vision. Vancouver, Canada. July 13, 2001.
[20] R.E. Schapire and Y. Singer. “Boosting algorithms using confidence-rated predictions”. Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pages 80--91, 1998.
[21] S. Gong, S.McKenna, and J.Collins. “An investigation into face pose distribution”. In Proc. IEEE International Conference on Face and Gesture Recognition, Vermont, 1996.
[22] S.Z. Li, L. Zhu, Z.Q. Zhang, and H.J. Zhang. “Statistical Learning of Multi-View Face Detection”. In Proc. 7th European Conference on Computer Vision. Copenhagen, Demark. May 2002
[23] Y. Amit, D. Geman and K. Wilder. “Joint induction of shape features and tree classifiers”. In IEEE Trans. Pattern Analysis. Mach. Intell, 19, 1300-1305, 1997.
[24] Y. Freund and R.E. Schapire. “A decision-theoretic generalization of online learning and an application to boosting”. In computation Learning Theory: Eurocolt’95, pages 23-37, Springer-Verlag, 1995.