Multi-view Face Detection Based on Position Estimation over Multi-camera Surveillance System

Ching-chun Huang[a] , Jay Chou[b] , Jia-hau Shiu[b] and Sheng-Jyh Wang[b]

[a]Dept. of Electrical Engineering, National Kaohsiung University of Applied Sciences, 415 Chien Kung Road, Kaohsiung, Taiwan, R.O.C

[b]Dept. of Electronics Engineering, National Chiao Tung University, 1001 Ta-Hsueh Road Hsinchu, Taiwan, R.O.C

ABSTRACT

In this paper, we propose a multi-view face detection system that locates head positions and indicates the direction of each face in 3-D space over a multi-camera surveillance system. To locate 3-D head positions, conventional methods relied on face detection in 2-D images and projected the face regions back to 3-D space for correspondence. However, the inevitable false face detection and rejection usually degrades the system performance. Instead, our system searches for the heads and face directions over the 3-D space using a sliding cube. Each searched 3-D cube is projected onto the 2-D camera views to determine the existence and direction of human faces. Moreover, a pre-process to estimate the locations of candidate targets is illustrated to speed-up the searching process over the 3-D space. In summary, our proposed method can efficiently fuse multi-camera information and suppress the ambiguity caused by detection errors. Our evaluation shows that the proposed approach can efficiently indicate the head position and face direction on real video sequences even under serious occlusion.

Keywords: Face detection, Multi-view face detection, Multi-camera surveillance system, Image information fusion

1. INTRODUCTION

Up to now, a lot of algorithms have already been proposed to solve the face detecting problem. The most popular approach is the training-based approach which collects lots of face data to construct a database for training. With the face database, a suitable classifier is learned to detect faces with high detection rate and low false alarm rate. For example, Viola and Jones[1] proposed the Adaboosting detection algorithm which is fast, robust and reliable to detecting frontal faces in 2-D images. Nowadays, several algorithms with similar structures have been proposed to improve the accuracy of detection based on Adaboosting detection algorithm. However, there still exist many difficulties in face detection, one of which is the detection of non-frontal faces. For non-frontal face detection, there appear view-dependent deformation and variation. Hence, these frontal face classifiers usually cannot be directly applied to non-frontal face detection.

In many applications, such as visual surveillance system, human faces in the captured images may not be upright and frontal. In these cases, the detection of faces becomes much more complicated. These non-frontal faces usually contain less information and present more diversity. This fact makes non-frontal detection a lot more sensitive to noise, background, illumination, and facial model.

A few methods for multi-view face detection have been proposed in recent years. They could be roughly divided as single-camera systems and multi-camera systems. For single-camera systems, Huang et al.'s[2] method provided an important reference. In their system, they proposed a method to construct a rotation invariant multi-view face detector. Their method was composed of a Width-First-Search tree detector structure, a Vector Boosting algorithm for learning strong classifiers, a domain-partition-based learning method, sparse features in granular space, and a heuristic search for sparse feature selection. Their system can detect multi-view faces with low computational complexity and high detection accuracy. However, the detection task may fail in some cases, such as low-resolution faces, inter-object occlusions, and incomplete human faces in images. It is also difficult for the method to detect the back side of human heads. Apparently, non-frontal face detection based on a single view of observation would be very difficult. The use of multiple cameras may somewhat relief the difficulties in non-frontal face detection.

Multimedia on Mobile Devices 2012; and Multimedia Content Access: Algorithms and Systems VI, edited by
Reiner Creutzburg, David Akopian, Cees G. M. Snoek, Nicu Sebe, Lyndon S. Kennedy, Proc. of SPIE-IS&T Electronic
Imaging, Vol. 8304, 830411 · © 2012 SPIE-IS&T · CCC code: 0277-786X/12/$18 · doi: 10.1117/12.908167

Multi-camera systems could provide us more information about the scenarios we concern. For a multi-camera surveillance system, more than one camera is installed within a surveillance zone. These cameras are installed at different locations, capture more information of the targets, and help users to monitor the targets in a more precise and efficient way. Huang and Wang[3] proposed an efficient way to fuse 2-D foreground detection result from multi-camera views. In their system, they adopted a probabilistic method to label multiple targets based on a Markov network. Zhang et al.[4] presented a system that integrates temporal and spatial information to build a multi-camera multi-view face detection system inside a room. By integrating temporal and spatial information with the dynamic programming approach, they aimed to detect the face of the lecturer for a lecture scenario inside an appropriately equipped smart room. In their approach, the multi-view face detector was implemented based on the FloatBoost approach[5]. However, those methods usually detect the face regions on 2-D images and then project the regions back to 3-D space for locating the 3-D positions. The inevitable false face detection and rejection may degrade the system accuracy even under a multi-camera surveillance environment.

In our system, we aim to establish multi-view face detection for an intelligent multi-camera surveillance system. Here, we plan to accomplish a system that is capable of detecting all targets' faces within the surveillance zone and is able to indicate the direction of each face inside a surveillance zone. Unlike most frameworks doing detection in 2-D images, our goal is to do this job in 3-D space since it would help us to well use the 3-D geometry knowledge such as the size of human face, the rough height of a human head above the ground plane, and etc. In detail, our system searches for the targets over the 3-D space using a sliding cube. Each searched 3-D cube is projected onto the 2-D camera views to determine the existence and direction of human faces. With the 3-D geometry prior, we could detect faces on 2-D images without trying different scales of patch sizes if comparing with many previous methods. Moreover, our approach can efficiently combine 2-D information from different camera views and suppress the ambiguity caused by 2-D detection errors. By fusing information form multi-camera views, we can infer the location of faces and their directions.

The rest of this paper is organized as follows. In Sec. 2, we present the main idea of the proposed framework. In Sec. 3, we explain how we estimate the locations of candidate targets on the 3-D ground plane. In Sec. 4, we detail our Multi-view face detection framework for locating the head positions and extracting the face directions. Experimental results and discussions are presented in Sec. 5. Last, Sec. 6 concludes this paper.

2. OVERVIEW

2.1 System Overview

In this paper, the whole system operates on an environment where a surveillance zone is monitored by multiple cameras. The main goal of our face detection framework is to locate human heads and detect the face directions. As in Figure 1, our system includes two steps – (1) 3-D position estimation and (2) multi-view face detection framework.

For the first step, we detect the locations of candidate targets. Here, we fused multi-view foreground detection results in order to identify the positions of candidate targets on the 3-D ground plane. The goal of this step is to filter out most impossible positions in the 3-D space and to speed up the searching process in the second step. For the second step, we aim to locate the optimal head position and determine the face direction of each target in the 3-D space. Here, we search the face with a 3-D cube within the possible subspaces, which is determined by the first step. Next, the 3-D cube at a possible face location is projected onto the 2-D camera views to get projected image regions. These image regions are verified by using eight pre-trained face detectors, which correspond to eight face views, in order to measure the probabilities for different face views. The measured probabilities from multiple images are finally fused in a systematic manner. Based on the fused probabilities, the finding of head positions and face directions are finally formulated as an optimization problem and solved in a unified way. In the following sections, we will explain the details of each functional block in our system flow.

3. 3-D POSITION ESTIMATION

3.1 Background Subtraction on a Single Camera

To identify the location of candidate targets on the 3-D ground plane, we need to detect the foreground regions which is accomplished by taking the difference between the current image and the reference background in a pixel-wise manner. Here, we model the reference background based on the Gaussian mixture model (GMM) approach[6]. An example of the foreground detection result is shown in Figure. 2. Note that the foreground regions are neither perfectly silhouetted nor well connected due to the influence of noise, variation of illuminations, and shadows.

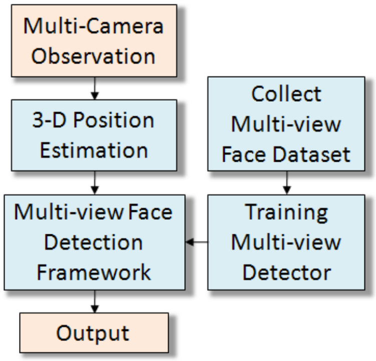

Multi-Camera Observation
3-D Position Estimation
Collect Multi-view Face Dataset
Multi-view Face Detection Framework
Training Multi-view Detector
Output

Figure. 1: Flow chart of the proposed multi-view face detection framework.

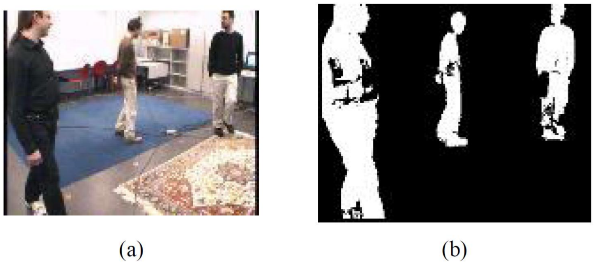

(a) (b)

Figure. 2: (a) The original image. (b) The result after background subtraction.

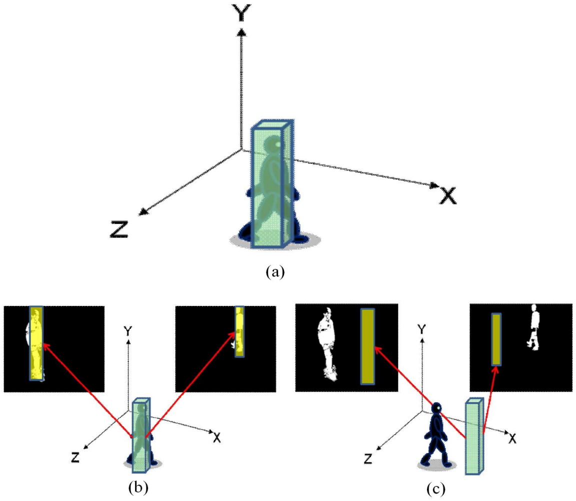


Figure. 3: (a) The illustration of the model-driven approach for information fusion. (b) A pillar at a true location generates larger overlapped regions. (c) A pillar at a wrong location generates smaller overlapped regions.

3.2 Information Fusion

By fusing the foreground regions from multi-camera views, we could determine the 3-D ground positions of candidate targets in a probabilistic manner. Here, we apply the model-driven approach proposed by Huang and Wang[3] to fuse 2-D information. By constructing a probability map named as Target Detection Probability (TDP), we could represent the probability of having a moving target at a ground location. In Figure. 3, we illustrate the concept of model-driven approach for information fusion. Here, as shown in Figure. 3(a), we use a pillar model to represent a human standing at a location on the ground plane. By projecting the pillar model at a location onto all 2-D images and calculate the overlapped area of the foreground region and the projection region, we could estimate the value of TDP at that 3-D location. Basically, the larger the overlapped region is, the more likely a target is standing at that location in the 3-D space. If the assumed 3-D location is incorrect, then the overlapped regions would be small. Based on this concept, we calculate the probability at every position in this surveillance zone and establish the TDP map by formulating the TDP as

$$G(X) \equiv p(X \mid F_1, \cdots, F_N) \sim p(X)p(F_1, \cdots, F_N \mid X) \tag{1}$$

In (1), X represents a location (x_1,x_2) on the ground plane. N is the number of static cameras in the multi-camera system. F_i denotes the foreground image of the ith camera view. Assume the size of camera views is M_s x $N_{s\,.}$ The point $(m\ ,n)$, which is in the range of $0\leq m \leq(M_s\text{-}1)$ and $0\leq n\leq(N_s\text{-}1)$, denotes the coordinates of a pixel on the foreground image. Then F_i is defined as

$$F_i(m,n) = \begin{cases} 1 & (m,n) \in \textit{foreground regions} \\ 0 & (m,n) \notin \textit{foreground regions} \end{cases} . \tag{2}$$

Moreover, given the location X, we assume the foreground images are conditionally independent of each other. Also, we assume the prior $p(X)$ is uniform distributed that indicates the equal possibility of finding a moving person at X. Therefore, (1) can be rewritten as

$$p(X)p(F_1, \cdots, F_N \mid X) = p(X)\prod_{i=1}^{N} p(F_i \mid X). \tag{3}$$

On the other hand, to formulate $p(F_i|X)$, we approximate a moving target at the ground position X as a rectangular pillar. The height H and radius R of the pillar are modeled as independent Gaussian random variables, with their Gaussian priors $p(H)$ and $p(R)$ being pre-trained via training samples. Based on the pre-calibrated projection matrix of the ith camera and a sample pair (H,R), we project the pillar onto the ith camera view to get the projected image M_i. Here we define M_i on the ith camera view as

$$M_i(m,n \mid H,R,X) = \begin{cases} 1 & \textit{if}\,(m,n) \in \textit{projected regions} \\ 0 & \textit{if}\,(m,n) \notin \textit{projected regions} \end{cases} . \tag{4}$$

The expectation of the overlapped region of M_i and F_i with perspective normalization offers a reasonable estimate about $p(F_i|X)$. That is, we define $p(F_i|X)$ as

$$p(F_i \mid X) = \iint \Omega_i(H,R,X)p(H)p(R)dHdR, \tag{5}$$

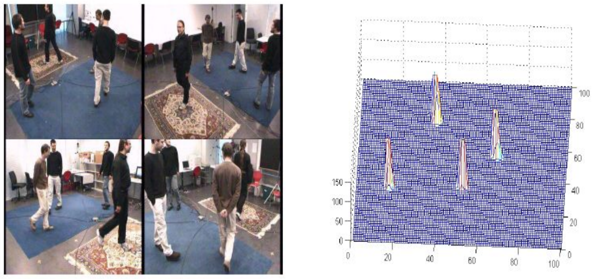
where the normalized overlap correlation, Ω_i, is defined as

$$\Omega_i(H,R,X) \equiv \frac{\iint F_i(m,n)M_i(m,n \mid H,R,X)dmdn}{\iint M_i(m,n \mid H,R,X)dmdn} \tag{6}$$

Based on (3) and (5), the TDP distribution could be calculated. An example of TDP distribution is shown in Figure. 4. Here, we may find the TDP is composed of many clusters; each cluster indicates a candidate target on the ground. Therefore, the candidate targets can be identified by some clustering algorithms, such as Mean-Shift clustering. After clustering, we can extract the number of candidate targets N_T inside the current surveillance zone and estimate the ground location X_i for the ith target by finding its corresponding cluster centers. Please refer to the paper[3] for more details.

4. MULTI-VIEW FACE DETECTION FRAMEWORK

After the position estimation, we identify the ground locations X of detected candidate targets on the 3-D ground. However, the head location of each candidate is still unknown. Even so, the extracted locations are useful for speeding up head finding. In next subsections, we aim to search head positions and determine the face directions. We will formulate the problem and explain our multi-view face detection framework.



(a) (b)

Figure. 4: (a) Input images. (b) The TDP of four moving targets in the surveillance zone.

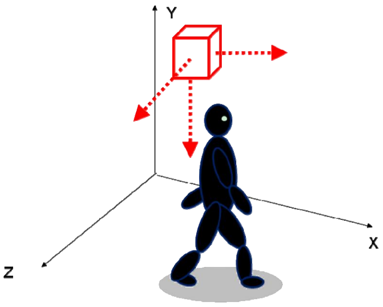


Figure. 5: The sliding cube in 3-D space.

4.1 Finding Target Heads and Face Directions

Many detection algorithms are already proposed for multi-view face detection. Most of these methods based on some learning approaches to train suitable detectors. After the training process, the trained detectors are able to detect specific object based on the sliding window approach in the image. However, because of some reasons, this sliding window approach in 2-D image may not be suitable for our application. First, it would not be easy to train a high accuracy detector for all face views. Second, from time to time, we need to search all scales to detect faces with different sizes including very small faces in the image. These small faces are usually too small to correctly identify. Also, the heavy searching time is unwelcome. Third, there could be some occlusions in the scene. Sometimes, we may need to detect faces that are incompletely observed in the image view. Due to the above reasons, detecting face directly in 2-D images usually generates many inevitable false detection and false rejection.

A multi-camera system may provide us more information about the scene and could theoretically decrease the false positive rate and increase the detection rate. However, the performance of the multi-camera system depends heavily on the way we utilize the 3-D geometric information. A conventional way is to detect faces in each 2-D image and then the 2-D detection results are projected back to the 3-D space for the final decision. Strictly speaking, this intuitive way is too ideal to be used in practical applications. This approach could work only when the detection rate of the 2-D face detector is high enough and the false alarm is low. Otherwise, the presence of plentiful false positives and false negatives would make the inference in the 3-D space very complicated and mistake-prone. The wrong information comes from 2-D images would accumulate and generate a lot of ambiguous results in the 3-D space.

Instead of searching and detecting the targets in 2-D images and then combining the outcome of each camera view in the 3-D space, in our system, we try to search and detect the targets in the 3-D space. This approach is like an extension from the 2-D sliding window approach to a 3-D sliding cube approach. In Figure. 5, we illustrate this concept. We now slide a cube in the 3-D space and determine 3-D head locations and face directions. However, we do not have the reconstructed 3-D scene for the 3-D based detection. In practical situations, what we have are the observations of 2-D images. Also, the reconstruction of the 3-D scene from all 2-D images is not reliable due to the limited number of cameras and the insufficient information from 2-D images. Hence, in our approach, we directly look for supports in the

2-D images. Here, we well utilize the geometric connection between 3-D space and 2D images according to the process of camera calibration beforehand. Based on the prior knowledge of 3-D geometric space, we generate hypotheses of head locations and face directions in the 3-D domain and base on the observed data from multiple 2-D images to make the final decisions. In each hypothesis, we assume the target face is at a specific location and direction in the 3-D space and confirm this hypothesis in its corresponding 2-D image regions. Figure. 6 shows an example of this process. In this example, a person is assumed to walk in a 3-D surveillance zone and a cube is sliding in the 3-D space to find where the head and what face direction of this person are. If the cube is slid to a suitable location that contains the human's head, the corresponding regions in 2-D images will fit the face portion with a proper face view. On the contrary, if the cube is at a place without any person's head, then the corresponding region will map to the background region and will not fit the face portion in each camera view.

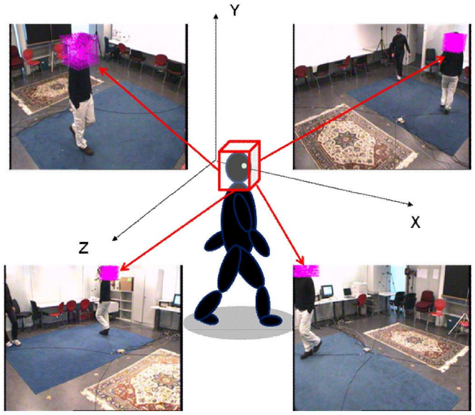


Figure. 6: A 3-D sliding cube approach for the finding of human heads and face directions.

4.2 Problem Formulation

Based on all the discussion in the previous sections, we now formulate our system goal as an optimization problem in a unified manner. Until now, we have detected N_T candidate targets and their ground locations $\{X_i\}_{i=1\sim N_T}$. Here, we use T_i to represent the ID of the ith target. For each target, we still need to determine its head location and face orientation. As mentioned before, we use a sliding cube approach to find the optimal location (l*) and orientation (h^*) of the target T_i. Here, we define the optimal solution as

$$(h^*_{T_i}, l^*_{T_i}) = \arg \max_{h\in H \;\; l\in L|_{T_i}} p(h,l \mid D, I, X_i, C). \qquad (7)$$

To numerically analyze this optimization problem in (7), we uniformly quantize the solution spaces of 3-D head positions and face orientations and denote the spaces as L and H respectively. In our system, the interesting 3-D space L, bounded by the surveillance zone and a user-defined height 200cm, is divided into 100x100x50 cubes. The orientation space H, ranging from zero to 360 degree, is divided into eight face directions. We also define other notations in (7) as below:

(D): The set of eight image-based face classifiers pre-trained for different face orientations.

(I): The set of multi-camera image views.

(X_i): The ground location of the candidate target T_i.

(C): Camera layout and geometry information.

($L|_{T_i}$): The possible 3-D head positions of candidate target T_i.

Here, $L|_{T_i}$ needs to be detailed. In our system, X_{N_T} indicates the set of estimated ground positions of the N_T detected targets in the 3-D space and could be utilized to reduce the solution space of the head locations. We Assume $X_i = \{x_i, y_i,$

$0\}$, where (x_i, y_i) is the ground position of the ith detected target. If we know the mean of human height is z_0 beforehand, we can reduce the interesting 3-D head positions of the target T_i, and define the reduced space $L|_{T_i}$ as

$$L|_{T_i} = \left\{ (x,y,z) \middle| \begin{array}{c} x_i - s/2 \le x \le x_i + s/2 \\ y_i - s/2 \le y \le y_i + s/2 \\ z_0 - s/2 \le z \le z_0 + s/2 \end{array} \right\}. \tag{8}$$

In (8), s defines a search range and is determined by the average size of a 3-D head. In our system, we set s *as* three times of the average size in order to account for the uncertainty. Also in (8), the average 3-D human height z_0 is obtained through statistical training. In Figure. 7, we illustrate the reduced position space given the plane location (x_0, y_0, 0) of the first detected target.

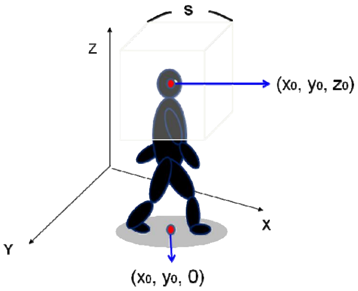


Figure. 7 : The reduced position space given a detected target location (x_0, y_0, 0).

To solve (7), we still need to define the calculation of $p(h,l|D,I,X_i,C)$. In detail, for each hypothesis (h,l) in the 3-D space, we project a cube at 3-D location l onto 2-D images to locate the focused patches and also generate the expected face orientations in different camera views based on h, I, and C. Here, we use Equation (9) to expresses the 3-D to 2-D projection process, where the function B(.) projects the 3-D cube and generates the expected face direction; $I_{n,ED,l}$ indicates the projected image patch with the expected face direction ED_n in the nth camera views of our four-camera system.

$$I_{n,ED,l} = B(l,h \mid I,C) \quad \forall l \in L|_{T_i} \quad n = 1,2,3,4 \tag{9}$$

For each camera view, we based on the expected face direction ED to select the corresponding face classifier from the classifier set D. By feeding the image patch $I_{n,ED,l}$ into the selected classifier, we could evaluate the likelihood $p_{n,l,h}$ of the hypothesis (h,l) based on information form this camera view. This process could be defined as

$$p_{n,l,h} = D(I_{n,ED,l}; ED_n) \qquad n = 1,2,3,4. \tag{10}$$

Note that ED_n determines one of the eight pre-trained face classifiers from the classifier set D. By combining the likelihoods from all camera views, we then define $p(h,l|D,I,X_i,C)$ as

$$p(h,l \mid D,I,X_i,C) = \prod_{n=1}^{4} p_{n,l,h} \tag{11}$$

Finally, we exhaustively search the solution spaces H and $L|_{T_i}$ in order to determine the optimal head location (l*) and face orientation (h^*) for target T_i in (7). Thanks for the pre-process of 3-D position estimation step introduced in Sec. 3, the solution space $L|_{T_i}$ is greatly reduced and the searching process is speeded up. Moreover, based on the automatically extracted target number N_T, we know the number of targets we need to search. Unlike many conventional face detection methods, no more lots of detected windows around a face region but only N_T face window with suitable scales are detected. In the next subsection, we would like to introduce how we train the classifier set D.

4.3 Modified Face Detection Algorithm

In our system, we train our 2-D face classifiers based on the procedure of Adaboosting training algorithm[7]. Here, we select Viola and Jones's face detection method[8-9] with modification because of its robustness and efficiency. In detail, we use the Adaboost algorithm to train a set of classifiers based on Haar-like features. Also, we apply the "integral image" for faster and easier computations. Finally, the cascade structure used in [8-9] is also applied in our system. In Figure. 8(a), we show the cascade structure. This cascade structure can quickly discard most non-face regions of a tested image at the earlier stages so that more efforts can be spent on face-like regions in the later stages.

The major difference between Viola and Jones's method and ours is that we modified the exclusion strategy of the cascade structure. In Figure. 8, we show the distinction between our structure and the original one. In our system, our goal is to design classifiers to output multi-level likelihood values instead of a false-true decision. Here, we found that if a tested image patch could pass more stages in the cascade structure, the patch is more likely to be a face region. Based on the finding, we proposed a modified cascade structure as shown in Figure. 8(b) to fulfill our system objective. Comparing the two structures, the original one excludes all rejected sub-windows into one class. However, in the modified structure, we further classify these rejected sub-windows into several subgroups, depending on the number of passed classifiers before rejection. As shown in Figure. 8(a), the original structure dumps all the rejected sub-windows to one rejection class. Hence, there are only two possible outcomes: face or non-face. There is no distinction between the sub-windows rejected by the first classifier and the sub-windows rejected by the last classifier. In contrast, in the modified structure, we separate sub-windows into several classes, with each class assigned a likelihood value depending on the number of classifiers the sub-window has passed through. As a sub-window has passed through more classifiers, it is more likely to contain a face-like region. If a sub-window passes through all classifiers, it gets the largest value of likelihood, as shown in Figure. 8(b). Finally, we independently train eight detectors for eight different viewing angles. Figure. 9(a) illustrates an example of the eight different viewpoints. For each viewpoint, we use the modified structure to train a face detector.

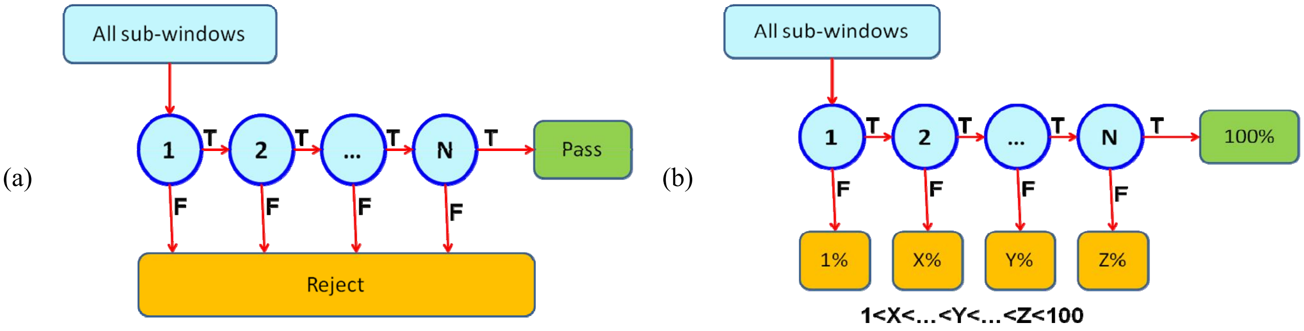
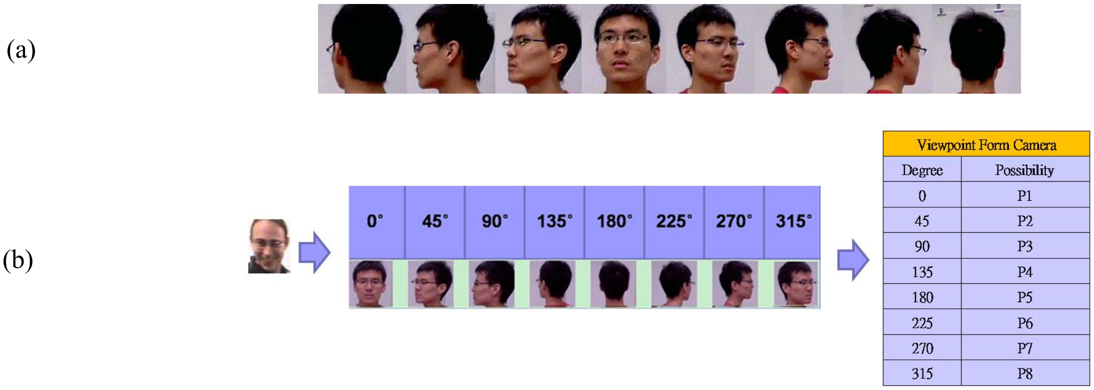


Figure. 8: (a) Original cascade structure of Viola and Jones's training algorithm. (b) Our modified structure.



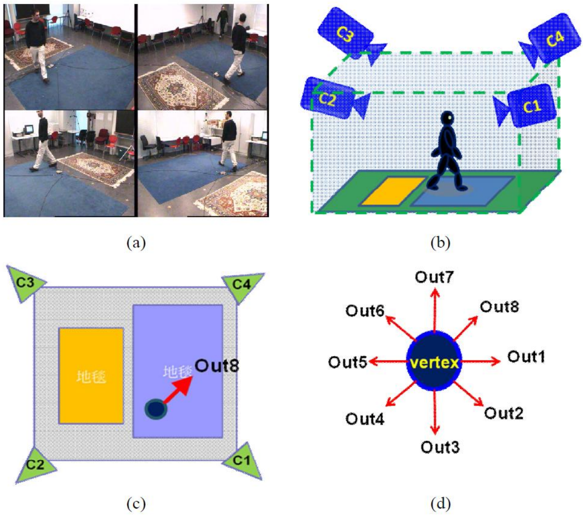
Viewpoint Form Camera	
Degree	Possibility
0	P1
45	P2
90	P3
135	P4
180	P5
225	P6
270	P7
315	P8

Figure. 9: (a) Eight views of the same person. (b) An illustration of an image patch verified by eight face classifiers for eight views.

5. EXPERIMENTS RESULTS

5.1 Environment Setup

In this subsection, we define the input and the output of our system. We also clearly define our scenario and experiment setup. In our experiments, the surveillance room is an indoor rectangular space with four cameras mounted at the four corners of the room. Each camera was calibrated beforehand and was set up with an appropriate viewing angle to clearly monitor the people in the room. In Figure. 10, we clearly illustrate our system setup. Figure. 10(a) shows the image views of four cameras and Figure. 10(b) shows the stereogram of the environment. In Figure. 10(c), we show a bird-eye view of the experiment environment. Also, we partition the orientation space of the scene into eight directions as illustrated in Figure. 10(d). This definition will help us to distinguish face directions in an easier way.



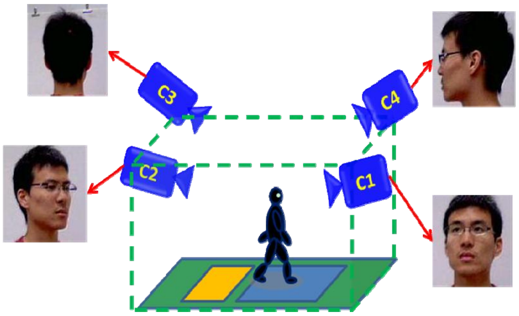
(a) (b) (c) (d)

Figure.10: (a) Image views of four cameras. (b) 3-D stereogram of the surveillance zone. (c) Bird-eye view of the surveillance zone. (d) Eight face directions.

Our system aims to find the face positions in all camera views and to indicate the direction of the detected faces in a bird-eye-view style. For example, in Figure. 10, there is a person walking in the surveillance room with his head approximately facing toward 315 degrees. According to our eight-orientation diagram in Figure. 10(d), the direction of his face approximately belongs to Outcome 8. In Figure. 10(c), we show the position of this man on the ground plane and the direction of his face in a bird-eye view style.

In our system, the geometry layout of four cameras is well learned and utilized for our system inference. As we can see in Figure. 11, there exists a coherence relationship among different camera views. For example, if a camera has captured a frontal face view, then the opposite camera must have captured the back side of the head, as illustrated in Figure. 11.

(a)



(b)

Hypothesis In 3-D Space	Corresponding Viewpoint In 2-D Images			
	Camera 1	Camera 2	Camera 3	Camera 4
H1	45	135	225	315
H2	0	90	180	270
H3	315	45	135	225
H4	270	0	90	180
H5	225	315	45	135
H6	180	270	0	90
H7	135	225	315	45
H8	90	180	270	0

Figure.11: (a) Coherence of different views (b) The learned relationships of the face directions among four cameras in our experiment environment.

5.2 Multi-view Face Dataset

In our method, we trained eight classifiers for eight different views around the head. The eight classifiers for different face views are 0˚, 45˚, 90˚, 135˚, 180˚, 225˚, 270˚, 315˚. For each direction, 400 training images are collected. These 400 images were captured from 50 students in NCTU, with 42 male students and 8 female students. For each student, we took pictures under two different backgrounds. For each background, 4 images are collected for each direction. Hence, there are totally 3200 (3200=50*4*8*2) images in our database. Figure. 12 shows some example images for each view.

We also evaluated the performance of our eight face detectors. Figure. 13(a) shows the detection rate of the eight different detectors. Figure. 13(b) shows the false alarm of the eight detectors. As we can, all these detectors achieve reasonable performance. Moreover, as expected, the front-view detector has better performance than the others.

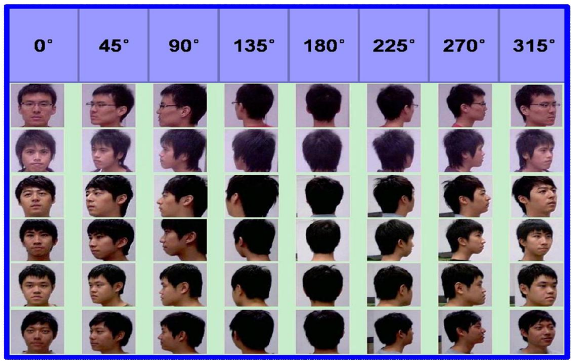
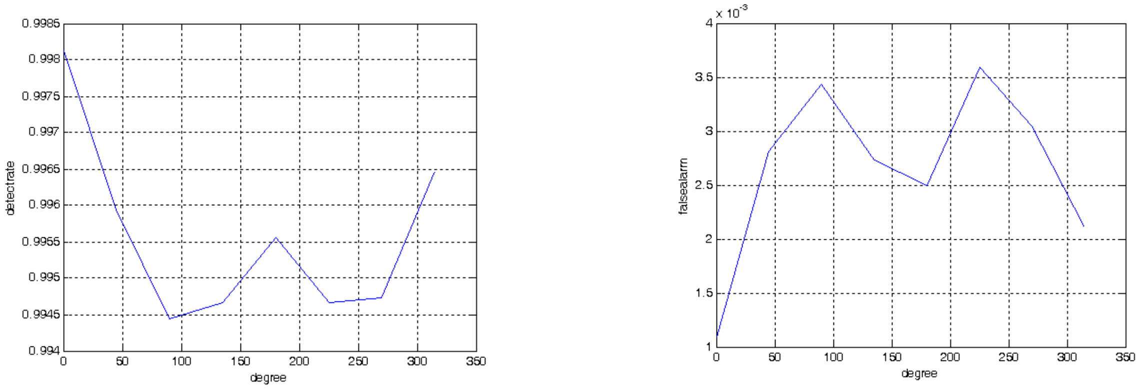


Figure.12: Examples of the multi-view face dataset



(a) (b)

Figure. 13: (a) Detection rate of 8 classifiers. (b) False alarm rate of 8 classifiers.

5.3 Head Localization and Multi-view Face Detection

To understand the process of head localization and multi-view face detection in our system, we show the projected windows under different 3-D location hypotheses in the four camera views. In Figure. 14(a), owing to a correct 3-D location hypothesis, the projected windows match the face regions well, while in Figure 14(b), the projected windows shift away from the correct face regions due to the wrong 3-D location hypothesis. To quantitative compare, we draw the calculated likelihood values under both the correct 3-D location and the wrong 3-D location over eight hypotheses of face orientations as shown in Figure. 14(c). Please note the blue curve, indicating the values under the correct location, is always higher than the green curve, representing the values under the wrong location. Also, the largest likelihood value over the blue curve indicates the optimal hypothesis of face orientation.

We tested our system over the video sequence provided by Fleuret's work [10]. Note the sequence contains more than 2000 frames. To quantitatively evaluate the detection and correspondence performance, false positive rate (FPR) and false negative rate (FNR) are used. In our system, the target detection and correspondence are defined as "correct" when the projected regions of the detected target in all camera views intersect the same individual and the detected face directions match the ground truth. Based on this definition, the calculated FPR and FNR of all tested sequence are 0.065 and 0.023.

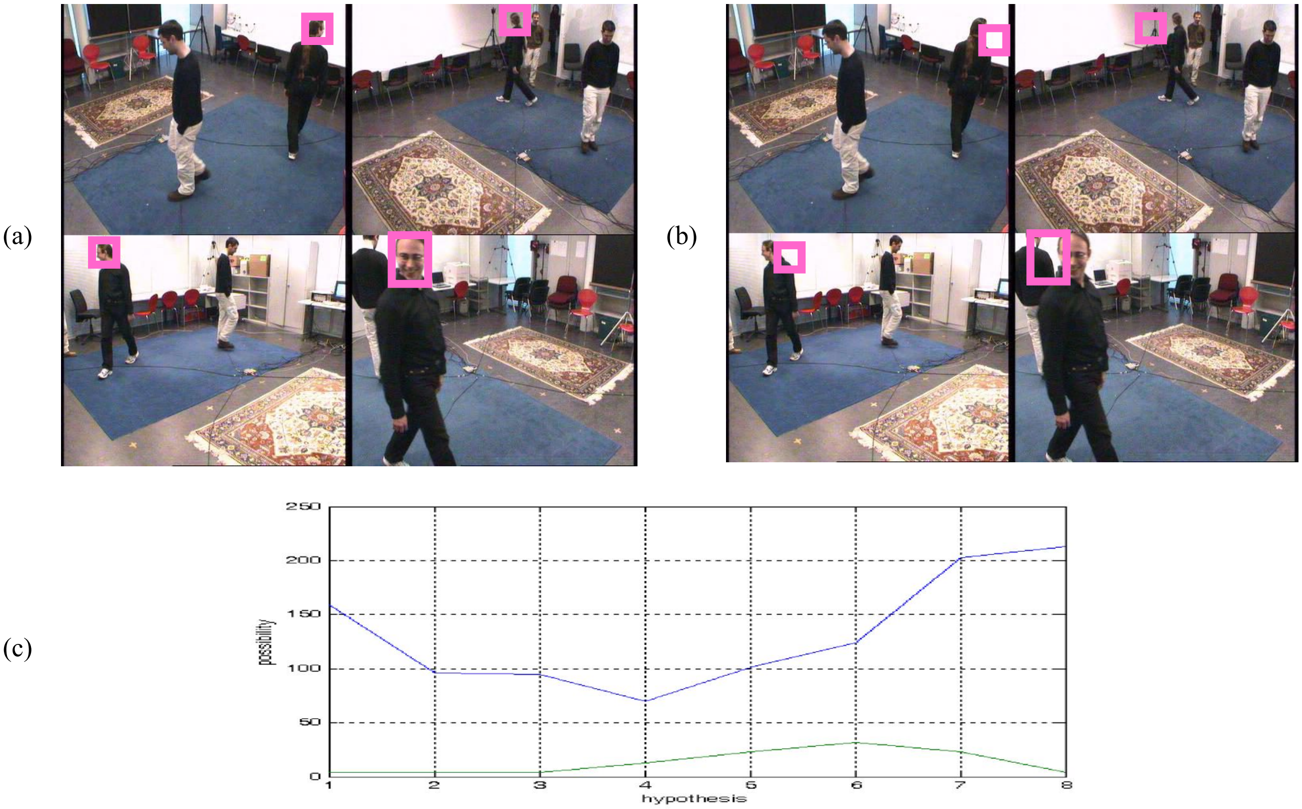


Figure. 14: (a) Detection result at a correct 3-D position. (b) Detection result at an incorrect 3-D position. (c) The blue line corresponds to the likelihood values of eight hypotheses of face orientations at the correct position, and the green line corresponds to he likelihood values of eight hypotheses at the incorrect position.

We also show some detection results in Figure. 15. To clearly present our outputs, we use bounding boxes with different colors to indicate different targets. We also mark the detected face direction onto the bird-eye view of the surveillance zone. In this example, there are two persons in the scene. As shown in the figure, our system can detect faces and identify the face directions even if some serious occlusion occurs or someone is out of image view. In Figure. 15(a), there is an occlusion case in the top-left image and there is a missing person in the lower-right image. For this example, our system can still find the approximate locations of the faces and the face directions, as shown in Figure. 15(b). Another experimental result is illustrated in Figure. 15(c-d).

6. CONCLUSION

In this paper, we present a multi-view face detection system over a multi-camera surveillance system. Through this system, we can detect all target faces in the given images and identify the direction of each face in the 3-D space. Unlike existing approaches whose performance are usually degraded by inter-object occlusion, the proposed system does not directly detect targets over the 2-D image domain nor project the 2-D detection results back to the 3-D space for correspondence. In our system, we search for the targets over small cubes in the 3-D space. Each searched 3-D cube is projected onto the 2-D camera views to determine the existence and the direction of a human face. With this approach, we can efficiently combine 2-D information from different camera views to make a more reliable and robust inference even under the inevitable false face detection and rejection of our 2-D classifiers.

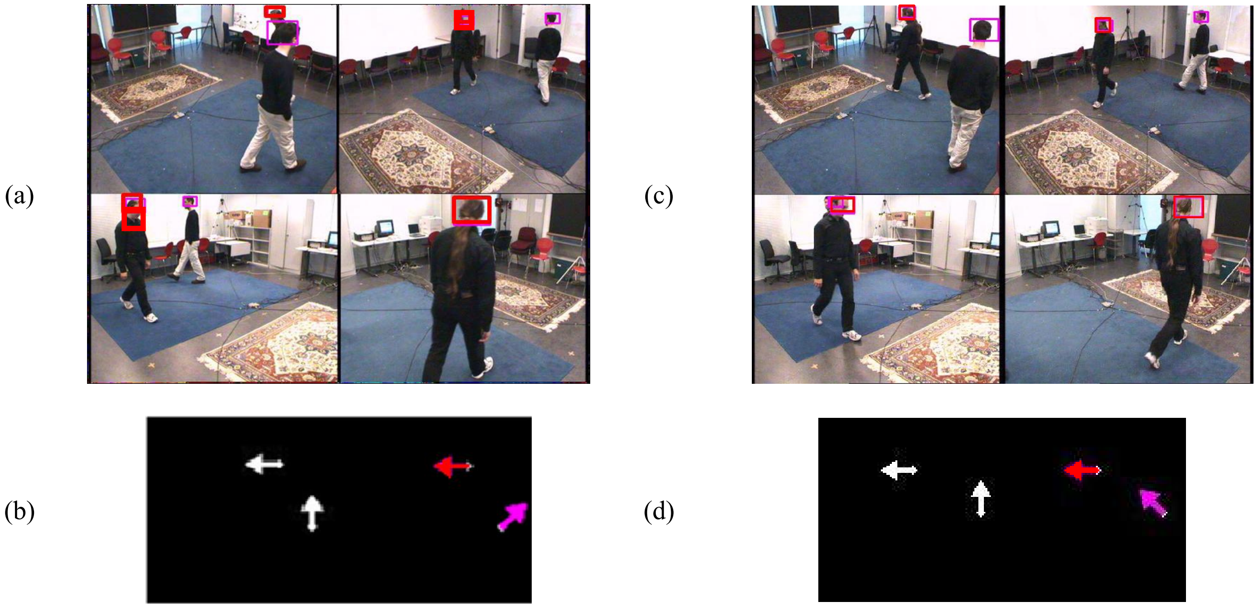


Figure. 15: (a) Multi-view face detection results with inter-object occlusion. (b) The bird-eye view of detected face directions of (a). (c) Another Multi-view face detection results. (d) Detected face directions of (c). Note white arrows indicate the ground truth. Colored arrows indicate our detection results.

ACKNOWLEDGEMENTS

This work was supported by the National Science Council, Taiwan, under the Grants No. NSC100-2218-E-151-007.

REFERENCES

[1] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," IEEE Conference on Computer Vision and Pattern Recognition, 511-518 (2001).

[2] C. Huang, H. Ai, Y. Li, and S. Lao, "Vector Boosting for Rotation Invariant Multi-View Face Detection," IEEE International Conference on Computer Vision, 446-453 (2005).

[3] Ching-Chun Huang and Sheng-Jyh Wang, "Moving Targets Labeling and Correspondence over Multi-Camera Surveillance System Based on Markov Network," IEEE International Conference on Multimedia and Expo, 1258-1261 (2009).

[4] Z. Zhang, G. Potamianos, A.W. Senior, and T.S. Huang, "Joint Face and Head tracking inside Multi-camera Smart Rooms," Signal Image and Video Processing. 1(2), 163-178 (2007).

[5] S. Li, Z. Zhang, L. Zhu, H.-Y. Shum, and H. Zhang, "Floatboost Learning for Classification," International Conference on Neural Information Processing Systems, 993-100 (2002).

[6] P. KaewTraKulPong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection," European Workshop on Advanced Video-based Surveillance Systems, 1-5 (2001).

[7] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," International Conference on Machine Learning, 322-330 (1997).

[8] M. Jones and P. Viola, "Fast Multi-view Face Detection," Technical Report, Mitsubishi Electric Research Laboratories, 1-11 (2003).

[9] P. Viola and M. Jones, "Robust Real-Time Face Detection,"International Journal of Computer Vision. 57(2), 137-154 (2004).

[10] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multi-Camera People Tracking with a Probabilistic Occupancy Map," IEEE Transactions on Pattern Analysis and Machine Intelligence. 30(2), 267-282 (2008).