

Collaborative acquisition of multi-view face images in real-time using a wireless camera network

S. Parupati, R. Bakkanagiri, S. Sankar, V. Kulathumani
Dept. of Computer Science and Electrical Engineering
West Virginia University

Email: {sparupat,rbakkana,ssankar}@mix.wvu.edu, vinod.kulathumani@mail.wvu.edu

Abstract—In order to support real-time face recognition using a wireless camera network, we design a data acquisition service to quickly and reliably acquire face images of human subjects from multiple views and to simultaneously index each acquired image into its corresponding pose. In comparison with detection of frontal faces, the detection of non-frontal faces with unknown pose is a much more challenging problem that involves significant image processing. In this paper, we describe a collaborative approach in which multi-view camera geometry and inter-camera communication is utilized at run time to significantly reduce the required processing time. By doing so, we are able to achieve a high capture rate for both frontal and non-frontal faces and at the same time maintain a high detection accuracy. We implement our face acquisition system on a 1.6 GHz Intel Atom Processor based embedded camera network and show that we can reliably acquire frontal faces at 11 fps and non-frontal faces at 10 fps on images captured at a resolution of 640 by 480 pixels.

I. INTRODUCTION

Face recognition systems have evolved significantly into a reliable mechanism for establishing identity of individuals in the context of applications such as access control and criminal identification. But while face recognition systems have traditionally been applied for identification from previously acquired photographs and videos, it is now becoming increasingly important to apply these systems for human identification in real-time. In order to support such real-time face recognition systems, we have designed and implemented a camera network based data acquisition system that collaboratively acquires multi-view face images of a human subject. We have specifically considered a *choke-point* scenario where a network of cameras are deployed over small critical access regions such as entrances, lobbies, walkways etc. in public places for human identification. Potential applications for such a system include raising an alert when criminals are detected entering critical places such as airports and shopping malls, raising an alert when lost individuals are detected and automatic recognition and registration of individuals at international port of entry.

While distributed camera networks can potentially improve the accuracy of human identification by offering multiple views of a subject, in order to be scalable and to be applicable in real-time identification they pose a design challenge in terms of the trade-off required between local processing and centralized computation. On the one hand local processing is needed prior to transmitting all the acquired data to a fusion center so that the network and the fusion center are not overloaded with too much data. On the other hand, it is also crucial to acquire as many images as possible at the individual cameras so as to improve the chances of accurate recognition. This is because a subject could be constantly moving his or her head and there could be a small duration when the face pose with respect to a camera is favorable for identification and it

is desirable to capture this frame. Therefore, it is important to keep computational overhead low at the distributed cameras so that the data can be processed at a high frame rate. Hence there is a requirement for a distributed service with low processing overhead that can quickly and reliably acquire data that is relevant for face recognition while suppressing data that is not useful for recognition. Our approach in this paper is to use the distributed cameras to perform face detection and transmit only the region containing faces detected in each frame to the base station.

Generally, frontal face images are the most suitable ones for reliable face recognition. However, in unconstrained environments it is not always possible to obtain enough high quality frontal face images required for accurate recognition. Under such circumstances, non-frontal face images acquired from a camera network can be used to improve the confidence of face recognition from frontal faces. Often profile (side view) face images contain tattoos, moles and special markers that are useful in human identification [1]. Recent studies have shown that profile views and partial profile views can be used for reliable face recognition with high accuracy [2], [3], [4], [5]. That being said, acquiring multi-view face images is a challenging task in terms of computational overhead especially due to their diversity [6]. Typically, separate face detectors are trained for each pose that are then sequentially or hierarchically applied on each frame to detect a face [7], [8], [9], [10], [11], [6]. Alternatively, a pose classifier is first applied through a sliding window of different sizes (that could fit a face) and then the appropriate face detector for that pose is used to detect the presence of a face [7]. Both of these approaches involve significant image processing and unsuitable when we would like to maximize the number of faces acquired for recognition.

A. Summary of contributions

In order to support robust and real-time face recognition, we design a network service for multi-view face image acquisition. Our service uses the geometry of the multi-camera network to collaboratively acquire both frontal and non-frontal face images in real-time while maintaining a high sampling rate. An overview of our approach is as follows. We first train face detectors based on Haar-like features [12], [13] for each pose class that is required to be detected. We then run a frontal-face detector on each camera in the network. Whenever a frontal face has been detected on any camera, say C_f , it sends a notification to other cameras which then narrow down their search to the region surrounding the epipolar line corresponding to the point where the frontal face is detected in C_f . By applying a pose-specific face detector on this much smaller region in the image, the cameras are able to quickly

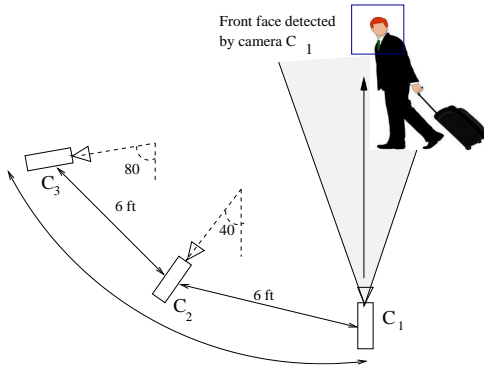


Fig. 1. Our experimental deployment of 3 cameras. The cameras are deployed along an arc of radius 10 feet with a separation of 6 feet between the cameras along the arc as shown. The angles made by the principal axes of cameras C_2 and C_3 with that of camera C_1 are 40° and 80° respectively. The cameras are deployed on tripods at a height of 7 feet from the ground. All cameras run a frontal face detector. When a frontal face is detected on any camera, a notification is broadcast to other cameras.

extract non-frontal face images and simultaneously index these faces into the corresponding face pose.

Thus, we utilize the multi-view camera geometry and inter-camera communication to reduce the amount of image processing required for multi-view face detection. Using this, we are able to process an image for detecting non-frontal faces at almost the same rate as for frontal faces. At the same time, by narrowing down the potential regions in an image for non-frontal face detection, we significantly improve the reliability of non-frontal face detection. Our system is easy to setup, does not require camera calibration and only depends on fundamental matrices of transformation between camera pairs.

B. Outline of the paper

In Section 2, we describe our multi-view face acquisition system and present effective buffer management strategies to maintain frame synchronization between the cameras in the presence of network delays and unequal processing times on the cameras. In Section 3, we evaluate the performance of our multi-view face acquisition system using experiments on a 3 node embedded camera network. We state related work in Section 4 and conclude in Section 5.

II. SYSTEM DESCRIPTION

In this section, we describe our network service for collaborative multi-view face acquisition. We divide our presentation into 3 parts: (1) system model, (2) software implementation, and (3) buffer management for frame synchronization.

A. System model

Our system for collaborative face acquisition consists of a network of N_c cameras with overlapping views that are all focused on a critical region such as entrances to public places and narrow corridors or walkways. The allowable distances between the cameras and the height of deployment will depend on the parameters of the cameras used for the system. For example if pan, tilt and zoom cameras are used, the cameras could be physically distant from the region and set to focus on the critical region. If fixed focal length cameras with low resolution are used, they will have to be closer together. However, we do require that the cameras are able to acquire facial images that are either frontal view or side view (and not top view).

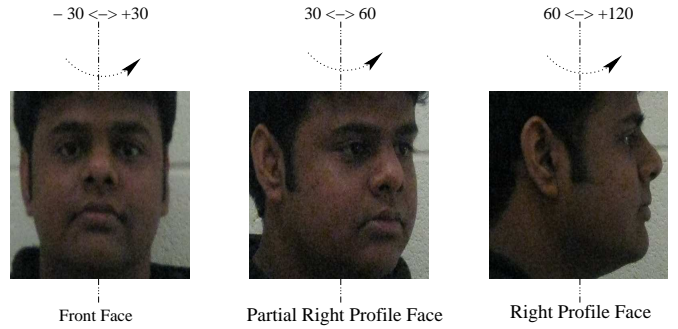


Fig. 2. We classify faces into front, profile and partial profile based on the yaw angles

For our specific experimental setting, we use a network of 3 cameras located along an arc of radius 10 feet. The cameras are deployed on tripods at a height of 7 feet from the ground with their principal axes parallel to the horizontal plane and with a separation of 6 feet between the cameras along the arc as shown in Fig. 1. The angles made by the principal axes of cameras C_2 and C_3 with that of camera C_1 are 40° and 80° respectively. The relative orientations between the cameras (the angles between principal axes of each pair of cameras) are assumed to be known. We assume that a clock synchronization algorithm is running on the nodes but we note that the clocks of any two nodes may not be in perfect synchrony. Let t_s denote the maximum clock synchronization error between any pair of cameras. This implies that the local clocks of any two cameras can be at most t_s apart.

The cameras are connected wirelessly to a base-station where the transmitted face images are collected and may be used for identification. Our goal is to acquire face images corresponding to the following poses: front, left (or right) profile, partial left (or partial right) profile. We use the yaw angle (that measures the rotation of a face image along the vertical axis) to define front, profile and partial profile faces (illustrated in Fig. 2). We define a face image of a subject acquired by a camera to be frontal if the yaw angle made by the subject's pose ranges from -30° to 30° . We define a face image of a subject acquired by a camera to be partial left (partial right) profile if the yaw angle made by the subject's pose ranges from -30° to -60° (30° to 60°). We define a face image of a subject acquired by a camera to be left (right) profile if the yaw angle made by the subject's pose ranges from -60° to -120° (60° to 120°). We have used the term *side face* to denote any non-frontal pose of the face.

B. Software implementation

The operations performed on the embedded cameras are divided into the following four threads. The implementation has been illustrated with a pseudo-code in Fig. 3.

1) *Capture*:: Images of the scene are captured on the cameras at F fps (frames per second) by the *capture* thread. Let $t_f = \frac{1}{F}$. We program the individual cameras to capture images whenever the local clock value is a multiple of t_f . The timestamp of frame x is defined as the time of capture of frame x and denoted as $t(x)$. Also, we call frames x and y captured in two different cameras to be synchronous if $t(x) = t(y)$. Each image acquired by the *capture* thread is stored in buffer B_{ff} along with its timestamp. Let $|B_{ff}|$ denote the maximum number of such images that can be stored in buffer B_{ff} .

Note that it may not be possible to capture an image at

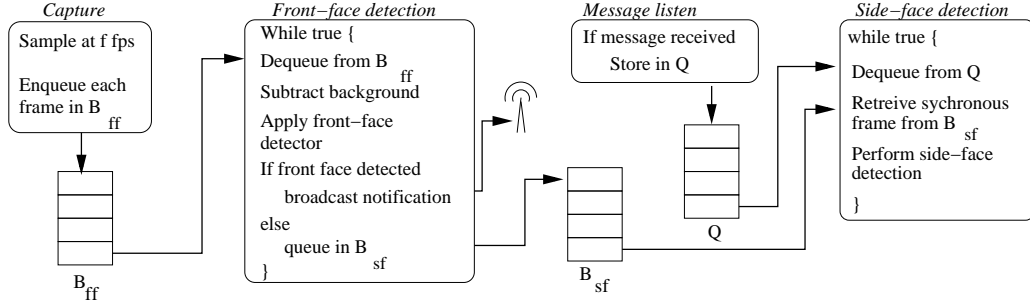


Fig. 3. Pseudo-code for operations on each embedded camera. each node executes 4 threads: capture, frontal face detection, message listening and side-face detection. The capture thread samples images at F fps and queues them in B_{ff} . The frontal face detection thread dequeues frames from B_{ff} and applies frontal face detector on background subtracted images. If a face is detected, a notification is broadcast to other cameras, otherwise the background subtracted frame is stored in B_{sf} . The message listening thread queues any incoming message into Q . The side-face detection thread dequeues messages from Q , retrieves the synchronous frame corresponding to the message from B_{sf} and performs the side-face detection procedure.

precise intervals of t_f time units, but we still encode the timestamp of an acquired image to be the closest multiple of t_f from the time of capture. Also, recall that there could be a maximum clock synchronization error of t_s time units between cameras. Due to both these reasons, images acquired by different cameras in the network with the same timestamp may not correspond to the same global time.

2) *Frontal face detection*:: The *frontal face detection* thread dequeues the oldest frame from B_{ff} to detect if a frontal face exists in the image. To detect frontal faces, we first perform background subtraction on an acquired frame by modeling the background based on median filtering over a set of frames and applying a threshold based differencing with respect to this image. Next, we apply morphological filtering and connected component grouping to eliminate noise regions and estimate rectangular blobs where a face image could be present. We then apply an OpenCV implementation of the Haar Cascade based face detector [12] in each of the estimated foreground blobs. If a frontal face has been detected in frame x , a notification message $M(c(x), t(x), w(x))$ is broadcast to all other cameras in the system, specifying the frame time $t(x)$, the location of the center of the face $c(x)$ and the width $w(x)$ of the bounding square around the detected face. If a frontal face is not detected, the background subtracted image is stored in buffer B_{sf} . Let $|B_{sf}|$ denote the maximum number of such background subtracted images that can be stored in buffer B_{sf} .

3) *Message listening*:: When a message $M(c(x), t(x), w(x))$ is received from another camera, the message is simply queued in buffer Q with maximum number of elements denoted by $|Q|$.

4) *Side face detection*:: The *side face detection* thread retrieves messages from Q one at a time. The timestamp $t(x)$ in the message M is used to retrieve the frame y in the buffer B_{sf} whose timestamp $t(y)$ is equal to the time $t(x)$ corresponding to frame x . Frame y is then removed from the buffer B_{sf} and a side-face detection procedure is run on frame y .

We use epipolar geometry to detect the side face in the frame y : if two cameras C_1 and C_2 observe the same scene point $W(X, Y, Z)$ and if the image point corresponding to W in C_1 is $P_1(x, y)$, then the image point $P_2(x, y)$ corresponding to W in C_2 must lie on the epipolar line corresponding to $P_1(x, y)$ [14]. The fundamental matrix is an algebraic representation of this epipolar geometry. Given the fundamental matrix F_{12} between cameras C_1 and C_2 , the relation between

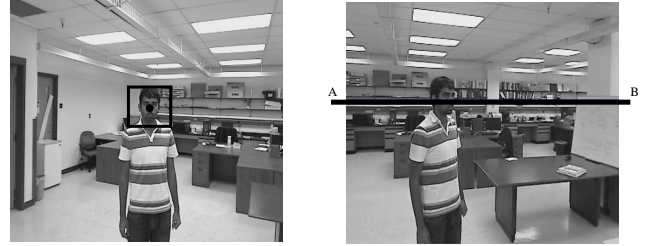


Fig. 4. Using camera network geometry for side face detection. When a frontal face is detected on any camera, a notification is broadcast to other cameras specifying the center of the detected face. Epipolar geometry is used to project this point to a corresponding epipolar line (shown as AB) in the other cameras. The other cameras apply a pose-specific side-face detector in a small region surrounding the segment of the epipolar line that intersects with the background subtracted image.

P_1 and P_2 is given by the following equation:

$$P_2' F P_1 = 0 \quad (1)$$

The epipolar line corresponding to point P_1 in camera C_2 is described by:

$$l_2 = F P_1 \quad (2)$$

For our experimental setting, we compute the fundamental matrices between each pair of cameras offline by using SIFT features for finding corresponding points, estimating the matrices using the normalized 8-point algorithm and then using RANSAC [15] to remove outliers from the detected keypoints [14]. We provide the fundamental matrices to the cameras before the experiments. Using the fundamental matrix F_{rs} we project the point $c(x)$ (the centroid of the frontal face detected) to a corresponding epipolar line in the synchronous frame y of the camera receiving the message (Fig. 4). We then determine the segment of the epipolar line that intersects with the background subtracted image retrieved from B_{sf} and extract a square block of size $W \times W$ pixels around the center of this segment, where W is set to be equal to the width of the detected frontal face image. Based on the relative camera orientations, we determine the expected pose of a side face and apply the side-face detector corresponding to the particular pose class on the extracted square block. For our experiments, we have trained face detectors for the left partial profile and left profile faces using an OpenCV implementation of the method described in [13]. To detect right partial profile and right profile faces, we apply the same detectors on the mirror images of the block.

C. Buffer management

Let t_{nd} denote the maximum network delay incurred between transmission and reception of a notification message. Let t_{ff} and t_{sf} denote the processing times for frontal face detection and side face detection respectively. In this section, we analyze requirements on $|B_{sf}|$ to ensure that synchronous frames always exist in the respective buffer B_{sf} of a camera when a notification message is being processed. We also analyze impact of t_{nd} , t_s , t_{ff} , t_{sf} and t_f on the expected number of frontal face and side face images that can be processed. For ease of presentation, we divide our analysis into the following cases.

R1: $t_{nd} = 0$, $t_s = 0$, $t_{sf} + t_{ff} < t_f$ In this case both the side-face and frontal-face processing can be finished before a new frame is sampled. Moreover the network delay is 0. Hence we only need to set $|B_{ff}| = 1$, $|B_{sf}| = 1$ and $|Q| = 1$. The expected number of frontal (N_{ff}) and non-frontal (N_{sf}) faces that can be detected by a camera over a time T is given by the following equations.

$$N_{ff}^{R1} = \frac{T}{t_f} \quad (3)$$

$$N_{sf}^{R1} = \frac{T}{t_f} \quad (4)$$

R2: $t_{nd} > 0$, $t_s > 0$, $t_{sf} + t_{ff} < t_f$ In this case also, a camera can finish both frontal and side-face processing before a new frame is sampled. But since $t_{nd} > 0$, any message retrieved from Q will have a timestamp that is old by at most $t_{nd} + t_{ff}$ (since it takes at most t_{ff} time for frontal face processing). Moreover, if $t_s > 0$ and the camera detecting the frontal face lags behind any side face processing camera by t_s units, then the message retrieved from Q could have a timestamp that is old by at most $t_{nd} + t_{ff} + t_s$. Therefore, in order to be able to retrieve a frame from B_{sf} corresponding to the timestamp of the incoming message, the condition on $|B_{sf}|$ is given by the following equation.

$$|B_{sf}| > \frac{t_{nd} + t_{ff} + t_s}{t_f} \quad (5)$$

Note that if the camera detecting the frontal face is ahead of the other cameras by t_s units and if $t_s > t_{nd} + t_{ff}$, then the frame corresponding to the timestamp in the incoming message will not be found in the side face processing cameras. The clocks in the side-face processing cameras in this case have not reached the clock value in the frontal face processing camera. In this case, the image with the most recent timestamp in B_{sf} is used for performing side-face detection. Thus we see that the clock synchronization error between cameras causes the processing of side-faces to occur on frames that are apart by at most t_s time units.

The expected number of frontal (N_{ff}) and non-frontal (N_{sf}) faces that can be detected by a camera over a time T remain unchanged from case *R1* and are given by Eq. 3 and Eq. 4 respectively.

R3: $t_{nd} > 0$, $t_s > 0$, $t_{ff} > t_f$ In this case, the frontal face processing time is greater than t_f . The side-face processing time t_{sf} can be greater or smaller than t_f . However, since each side-face processing camera also runs a frontal face detector, the average time to process each message from Q is $t_{sf} + t_{ff}$ which is greater than t_f . Therefore, new frames will be

captured into B_{ff} and new messages can arrive in Q before the earlier ones are processed. Now, if we assume that the system will always remain active (in other words, there will be a human subject in the scene at all times), then even if we buffer frames in B_{ff} and Q , there will be no *idle* time to process these frames. We can therefore set $|B_{ff}| = 1$, and $|Q| = 1$. Thus the frames cannot be processed at the sampling rate and only the most recently acquired frame and most recently received message are queued in B_{ff} and Q respectively.

We now determine the size of B_{sf} required so that for any frame in which a frontal face has been detected, we can find the corresponding synchronous frame in B_{sf} . We note that $|Q| = 1$ and the maximum time before which this message is retrieved by a camera for detecting non-frontal faces is bounded by $t_{ff} + t_{sf}$. Now from the discussion in case *R2*, we note that the timestamp in the message that is retrieved is old by $t_{nd} + t_{ff} + t_s$. So, the required size of buffer B_{sf} is determined by the following equation.

$$|B_{sf}| > \frac{2t_{ff} + t_{sf} + t_{nd}}{t_f} \quad (6)$$

The expected number of frontal (N_{ff}) and non-frontal (N_{sf}) faces that can be detected by a camera over a time T are given by the following equations.

$$N_{ff}^{R3} = \frac{T}{t_{ff}} \quad (7)$$

$$N_{sf}^{R3} = \frac{T}{t_{ff} + t_{sf}} \quad (8)$$

III. PERFORMANCE EVALUATION

In order to evaluate the performance of our data acquisition system, we implement it on a 3 node embedded camera network (schematics shown in Fig. 1). We assemble an embedded camera using a Logitech 9000 camera, an Intel Atom 1.6 GHz processor based motherboard from Acer and an IEEE 802.11 based wireless card. We consider one human subject in the scene at a time. Each subject stands at a distance of approximately 10 feet from the cameras (close to the center of the arc) facing any one of the 3 cameras. Note that, if the subject is facing camera C_1 as shown in Fig. 1, then the pose estimated by camera C_2 and C_3 are right partial profile and right profile respectively. We have tested the system with 10 different subjects with approximately 15 minutes of data collected for each subject.

We use the NTP protocol for achieving clock synchronization between nodes and empirically observe a synchronization error of 10 ms. Alternatively we could also use a completely decentralized clock synchronization protocol and recent papers have demonstrated sub-millisecond accuracies [16] with extremely small communication costs and synchronization messages sent only once every 10 seconds. The impact of the clock synchronization error is that images grabbed at two cameras with the same timestamp may not correspond to the same global time. However, we note that with an error of a few milli-seconds, a subject could not have moved much in that time.

We perform our experiments in two environments: one with a lot of clutter in the background (this environment is shown in Fig. 4) and the other one with a relatively plain background. Images are sampled by each camera at 25 fps.

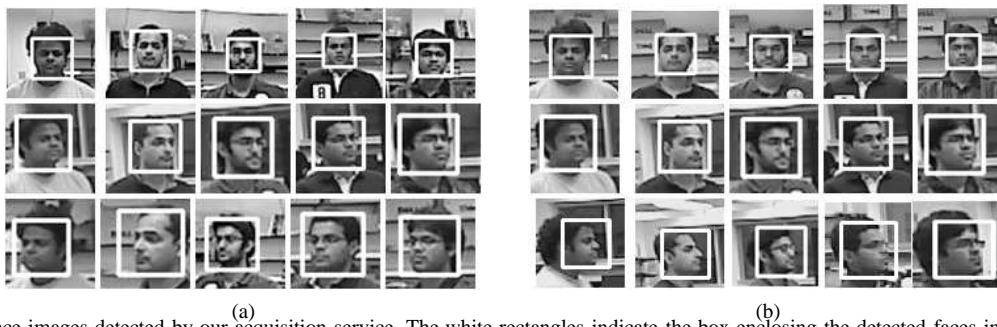


Fig. 5. Example face images detected by our acquisition service. The white rectangles indicate the box enclosing the detected faces in each pose. (a) Images acquired with subjects facing C_2 : (Top) Frontal (Middle) Left partial profile (Bottom) Right partial profile. (b) Images acquired with subjects facing C_1 : (Top) Frontal (Middle) Right partial profile (Bottom) Right profile.

Operation	Time (ms)	Time (ms)
	(clear)	(cluttered)
Image capture and storage	2	2
Background subtraction	2	3
Dilation	2	2
Frontal face detection	75	102
Total t_{ff}	81	109
Total t_{sf}	15	15

TABLE I
PROCESSING TIMES: MULTI-VIEW FACE DETECTION IN CLEAR AND CLUTTERED BACKGROUND

Thus $t_f = 40$ ms. In Table I, we show the average execution times for the different processing modules in our system. In Table II, we show the number of frames that are processed per second for detecting frontal faces and side faces. The frontal face detector is applied on background subtracted regions and sometimes even on spurious blobs detected as the foreground. The side-face detector on the other hand is applied only on a much smaller region that is corroborated by the frontal face detecting camera. We note from Table II that the number of frames processed per second for detecting frontal and side faces conform to the rates shown in Eq. 7 and Eq. 8 respectively when accounting for the operating system overhead. For instance in a clear background we observe that the average value of $t_{ff} = 81$ ms and average value of $t_{sf} = 15$ ms and accordingly 11 frames can be processed per second for detecting frontal faces and 10 frames can be processed per second for detecting side faces in a clear background.

Rates per second	Clear	Cluttered
Frontal face processed	11.1	8
Frontal face detected	10.2	6.05
Side-face processed	10	5.5
Side-face detected	9.7	5.2

TABLE II
DETECTION RATES OF FRONTAL AND SIDE FACES USING COLLABORATIVE ACQUISITION SERVICE

The actual number of frontal and side faces detected correspond to the output of the detector itself. The difference between frames processed and faces detected gives a measure of the false negatives for the respective detectors. In a clear background, the number of frontal faces detected per second are almost equal to the number of frames processed per second. All the frontal faces detected are notified to the other

cameras and the number of side faces detected per second in each camera matches the frontal face detection rate. In a cluttered background, the number of missed detections for frontal faces are high and yields a frontal face detection rate of 6 faces per second and as seen in Table II, the side face detecting cameras are able to match this detection rate.

The number of falsely detected side-faces using our acquisition service were negligible (close to 0.1% of the total number of side faces detected in our experiments). By selectively applying the side-face detector on regions corroborated by the frontal face detecting camera, we are able to achieve this low false alarm rate.

The maximum network delay is observed to be 50ms, but we note that this only affects the size of B_{sf} and not the overall face detection rate. We also note that the required buffering is very low (approximately 10 frames). By transmitting only the face images, that are on average 60 by 60 pixels in size, we are able to reduce communication bandwidth by 98% compared with transmitting the entire image (640 by 480 pixels) and by 80% when compared with transmitting the background subtracted image (100 by 200 pixels on average). By performing face detection and simultaneously estimating the pose, we are also able to reduce significant processing time at the fusion center for face recognition.

IV. RELATED WORK

In [12], Viola and Jones have designed a face detector that is suitable for real-time frontal face detection [12]. Their approach utilizes the AdaBoost algorithm to identify a sequence of Haar like features that indicate the presence of a face. Since then other frontal face detection algorithms have been developed, a survey of which is presented in [17]. Approaches for multi-view face detection have generally been of two types. The first approach is to estimate the pose over each sliding window in an image (which may not necessarily have a face) and then applying the pose specific detector [7]. When involving multiple face poses this is a hard problem and moreover false estimates of a face pose will lead to incorrect detection of a face. In the second approach, different view-specific face detectors are applied sequentially or hierarchically to an image [7], [11], [18], [6]. In this paper, we have used an OpenCV [19] implementation of the frontal face detector presented in [12] and trained pose-specific detectors using Haar like features for non-frontal faces as described in [13]. Then, we have used the information about a detected frontal view along with relative camera orientations and the subject location to detect non-frontal faces in other cameras and we observe that our approach decreases overall processing time.

Multi-view camera geometry has been exploited by several recent research efforts to effectively fuse information from different cameras and consequently improve the accuracy in the context of tasks such as object detection, behavior matching, action classification and reliable foreground extraction [20], [21], [22]. By way of contrast, in this paper we have utilized multi-view geometry to improve the computational efficiency of the system by collaborating among the cameras in real-time and reducing the amount of image processing required.

In the context of face recognition, multiple cameras have been used for tracking in an active control mode by which one or more cameras are controlled to yield a dynamic coverage [23], [24]. An example of such a system is the combination of a fixed camera and PTZ camera that is used for close-up tracking of humans and subsequent identification. In our approach instead of continuously tracking an individual at close quarters to eventually get a good view that is suitable for recognition, we rely on redundancy offered by multiple camera views to opportunistically acquire a suitable face image for identification [25].

V. CONCLUSIONS

In this paper, we presented a collaborative multi-view face acquisition service that can be used to support real-time face recognition. Our service can detect and extract face images from different poses and simultaneously identify these poses while maintaining a high sampling rate. We avoid complex image processing and instead use network geometry and communication between the cameras to reduce the processing time. We are able to achieve a non-frontal face detection rate that is almost equal to frontal face detection rate, thus highlighting the advantage over multi-view face detection schemes based on sequentially or hierarchically applying detectors for different poses. Our service is light-weight in terms of processing complexity, has low buffering requirements and is appropriate for implementation on different smart camera platforms [26] resulting in portable and even covert deployments for human recognition.

We note that while we have used face detectors based on Haar-like features in our system, they could be replaced with other pose-specific face detectors as well. Also, while the specific performance numbers for processing rate are platform and algorithm specific, the key observation is that our system can be used to detect non-frontal faces at the same rate as frontal faces (where the rate of processing is determined by the algorithm and the platform).

In this paper, we have used the detection of frontal face images in a camera to guide the computation at run-time in other cameras. Alternatively, the detection of patterns or events other than frontal faces can also be used to trigger localized image processing operation in other cameras and improve the computational efficiency of the system. This gives rise to a more generalized use of our proposed framework for collaboration in a camera network.

As a next step, we would like to integrate our multi-view face acquisition framework with multi-view face recognition techniques and quantify the achievable recognition performance [2], [3], [4], [5]. In particular, we would like to integrate our system with sparse encoding based recognition techniques [27] that are especially well-suited for distributed light-weight implementations. We would also like to extend our system to operate with multiple subjects in the scene at any given time

and study system performance.

REFERENCES

- [1] U. Park and A. K. Jain. Face matching and retrieval using soft biometrics. *IEEE Transactions on Information Forensics and Security (TIFS)*, 5(3):406–415, 2010.
- [2] I.A.Kakadiaris, H.Abdelmunim, W.Yang, and T.Theoharis. Profile-based face recognition. In *8th IEEE international Conference on Automatic Face and Gesture Recognition*, pages 1–8, 2008.
- [3] K.W.Cheung, J. Chen, and Y.S. Moon. Pose-tolerant Non-frontal Face Recognition using EBGM. In *2nd IEEE International Conference on Biometrics: Theory, Applications and Systems*, 2008.
- [4] F.Yang, M.Paindavoine, H.Abdi, and D.Arnoult. Fast image mosaicing for panoramic face recognition. *Journal of Multimedia*, 1(2):14–20, 2006.
- [5] B. Bhanu and X. Zhou. Face recognition from face profile using dynamic time warping. In *International Conference on Pattern Recognition*, volume 4, pages 499–502, 2004.
- [6] C. Huang, H. Ai, Y. Li, and S. Lao. High-performance rotation invariant multiview face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):671–686, April 2007.
- [7] M. Jones and P. Viola. Fast multi-view face detection. Technical Report TR2003-96, Mitsubishi Electric Research Laboratories, 2003.
- [8] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994.
- [9] H. A. Rowley and S. Baluja and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(1):22–38, 1998.
- [10] K. Sung and T. Poggio. Example-based learning of view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(1):39–51, 1998.
- [11] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. In *European Conference on Computer Vision*, 2002.
- [12] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.
- [13] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *IEEE ICIP*, 2002.
- [14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [15] A. Fischler and C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 24, June 1981.
- [16] T. Schmid, P. Dutta, and m. Srivastava. High-resolution, low-power time synchronization: an oxymoron no more. In *International Conference on Information Processing in Sensor Networks (IPSN)*, 2010.
- [17] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical Report MSR-TR-2010-66, Microsoft Research, 2010.
- [18] J. Feraud, O. Bernier, and M. Collobert. A fast and accurate face detector for indexation of face images. In *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.
- [19] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [20] S. M. Khan, P. Yan, and M. Shah. A homographic framework for the fusion of multi-view silhouettes. In *IEEE 11th International Conference on Computer Vision (ICCV)*, 2007.
- [21] C. Wu and A. Khalili and H. Aghajan. Multiview activity recognition in smart homes with spatio-temporal features. In *International Conference on Distributed Smart Cameras (ICDSC)*, 2010.
- [22] E.B. Ernis and V. Saligrama and P. M. Jodoin and J. Konrad. Abnormal behavior detection and behavior matching for networked cameras. In *Second ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–10, 2008.
- [23] N. Krahnstoever, Ser-Nam Lim T. Yu, K. Patwardhan, and P. Tu. Collaborative real-time control of active cameras in large scale surveillance systems. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.
- [24] X. Zhou, R. Collins, T. Kanade, and P. Metes. A master-slave system to acquire biometric imagery of humans at distance. In *ACM International Workshop on Video Surveillance*, 2003.
- [25] V. Kulathumani, S. Parupati, A. Ross, and R. Jillela. Collaborative face recognition using a network of embedded cameras. In *Distributed Video Sensor Networks*, pages 373–389. Springer, 2011.
- [26] R. Kleihorst, B. Schueler, A. Danilin, and M. Heijligers. Smart camera mote with high performance vision system. In *ACM SenSys Workshop on Distributed Smart Cameras*, 2006.
- [27] J. Wright and A. Y. Yang and A. Ganesh and S. Sastry and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(2):210–227, 2008.