# FACENET: TRACKING PEOPLE AND ACQUIRING CANONICAL FACE IMAGES IN A WIRELESS CAMERA SENSOR NETWORK

*Kyle Heath*

Stanford University
Department of Electrical Engineering
Stanford, CA 94305

*Leonidas Guibas*

Stanford University
Department of Computer Science
Stanford, CA 94305

## ABSTRACT

Wireless camera sensors networks are an emerging sensing technology that could enable new applications in security, transportation, and healthcare. Tracking and identifying moving objects is a fundamental visual surveillance task and methods that respect the energy and bandwidth constraints of a wireless sensor network are needed. This paper proposes a real-time algorithm for tracking people and acquiring canonical frontal face images in this setting.

Clusters of sensors collaborate to track individuals and groups of people moving in an indoor environment. The sensors are tasked to retrieve the best frontal face image for each tracked individual according to a score based on face size and orientation. The method exploits information about the target's trajectory to retrieve an approximate best frontal face image in an energy efficient manner. Frontal face images acquired by this algorithm are suitable for standard face recognition algorithms and would be valuable for identity management.

To evaluate this approach on real data, a prototype surveillance system called FaceNet was developed. FaceNet displays a compact summary of human activity by overlaying a floorplan diagram with the 2D trajectories and a face image for each track. A simple benchmark of FaceNet shows the amount of data transmitted from the sensors can be reduced by 97 percent compared to a naive centralized streaming architecture and has the potential to significantly reduce the energy used by wireless nodes for this type of surveillance task.

***Index Terms***— Wireless Camera Sensor Networks, Sensor Tasking, Tracking

## 1. INTRODUCTION

Today, networks of cameras are used to monitor transportation systems, businesses, and other important spaces. Wireless sensor technology could make such networks far more pervasive by greatly reducing the cost of deployment and operation. Networks of many cameras are particularly well suited for cluttered real-world environments like buildings and urban areas. In such environments, only a small fraction of the region of interest is visible from any single vantage point because of occlusions. Using many simple camera sensors instead of a few high quality cameras increases the probability that an object of interest can be observed with a useful and unoccluded view.

Yet the sensor network paradigm of distributed sensing, processing, and storage requires very different architectures and algorithms than those used in conventional systems. In most conventional architectures, video is streamed from a sensor to a central location where it can be monitored and stored. In the wireless sensor networks paradigm, these tasks are done in a distributed fashion and close to the source of the data. For battery powered wireless sensors, the dominant design constraints are energy and the bandwidth of the wireless channel. Simply broadcasting a constant stream of high bit-rate data would quickly exhaust a sensor's energy and saturate the network's communication capacity. In this situation, data should be stored locally and processed to create high-level representations that are cheap to communicate.

This paper describes a method for tracking people in 2D world coordinates and acquiring canonical frontal face images that fits the sensor network paradigm. Frontal face images are particularly desireable features for tracking and identity management because they are largely invariant to day-to-day changes in appearance. Our primary contribution is to show how sensing the trajectories of moving objects can be exploited to acquire high quality canonical views while conserving node energy.

The paper is organized as follows. Section 2 describes a lightweight method for visually tracking people in an indoor environment. Section 3 describes a sensor tasking algorithm for energy efficient canonical face image retrieval. Section 4 gives a description of the FaceNet system and some experimental results. In section 5, related work is discussed and section 6 concludes with a discussion of ideas for further investigation.

1-4244-1354-0/07/$25.00 

## 2. LIGHTWEIGHT PEOPLE TRACKING

### 2.1. Overview

Here we describe a method for fusing data from multiple cameras to estimate the 2D trajectory of people moving in the observed space. The method is lightweight in the sense that only simple image processing steps are required on the resource constrained camera sensor. In the following, we assume that the cameras intrinsic and pose parameters have been measured as part of the network deployment process. Several methods for calibrating a network of cameras have been proposed in the literature and particularly applicable work includes [1] and [2]. Furthermore we assume that during the initialization stage, cameras use their calibration information to discover clusters of other cameras that observe the same region of space and elect a cluster leader.

In normal operation, sensors in a cluster enter a low-power state where they take turns watching the common observation space. When a moving object is detected, nodes in the cluster are alerted and begin sending a compact summary of foreground regions detected in their view to a cluster leader. The cluster leader calculates an approximate 2D visual hull using the foreground regions and known camera geometry. A Bayesian filter is applied to the noisy visual hull data to produce an occupancy probability map. Peaks in the occupancy probability map are tracked across time to estimate the trajectories of moving targets. The approach requires only very simple processing at the camera sensors and greatly reduces the amount of data that must be transmitted across the wireless network compared to transmitting raw or even compressed image data.

### 2.2. Compact vector-based foreground image

Each camera generates a compact vector representation of the foreground region in its view as follows. A simple Gaussian background model is used to create a foreground probability image for the current camera frame. This image is thresholded at different levels to generate a set of foreground confidence masks. The outer contours of each confidence mask are simplified to polygons with a small number of vertices. This set of polygons is then used as a compact vector representation of the foreground likelihood image. Using more than two confidence masks allows the system to avoid making noisy binary decisions about the presences or absence of a foreground region that might introduce large errors in the visual hull calculation that will follow. Additionally, the degree of polygon simplification can be tuned to achieve the best trade-off between the faithfulness and the compactness of the foreground representation. This can be seen as a generalization of the compact scan-line representation used in [3] which removes the tight restrictions on horizontal camera placement.

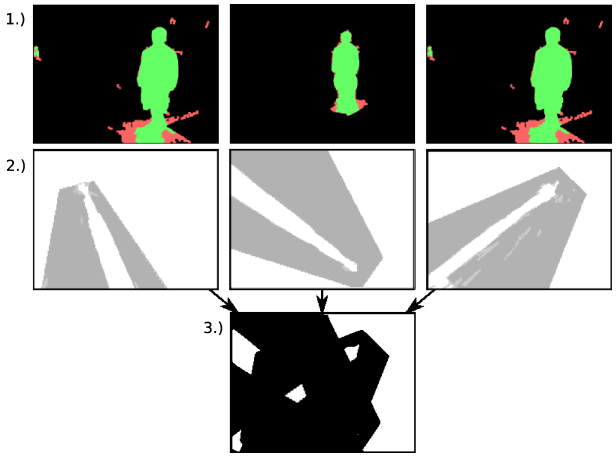


**Fig. 1**. Calculation of the 2D visual hull by 1.) forming a compact foreground probability image and 2.) projecting the foreground image from the camera image plane onto the ground plane and 3.) multiplying projected images

### 2.3. Occupancy map data fusion

The vector-based foreground images from multiple cameras are fused at the cluster leader by calculating the visual hull of the observed objects. Since we are interested in the location of the objects on the ground plane, a 2D slice of the 3D visual hull parallel to the ground plane is sufficient. As described in detail in [3] and [4], the calculation of this 2D slice can be performed by projecting the foreground probability images from the camera's image plane onto a plane parallel to the ground and then multiplying these projected images. This process is illustrated in Figure 1.

The resulting visual hull can not be used directly for tracking objects because it may contain connected components that do not correspond to any object in reality. These extra regions are called phantoms because they tend to appear and disappear from thin air. These are discussed in more detail in [3]. By considering the visual hull to be a noisy measurement of the space actually occupied by people, a Bayesian filter with a simple human motion model can be applied to estimate the occupied space. Using a simple random walk model of motion, the probability of location at the next time step is distributed as a Gaussian centered at the current location with variance proportional to the object's expected speed. The result of this simple filtering step is an occupancy probability map whose peaks correspond to space that is likely to be occupied by a person.

### 2.4. Extraction of tracks

The occupancy probability map is thresholded to extract blobs that are likely to be occupied by a person. These blobs are tracked across time using a Kalman filter with a simple nearest neighbor heuristic for data association. Examples of tracks

generated with this lightweight tracking algorithm are shown in Figure 3.

## 3. TASKING FOR CANONICAL FACE IMAGE CAPTURE AND LAZY RETRIEVAL

The most direct method of finding the best canonical face image for each individual moving in the observed space might be to transmit the video data to a central processing node which can track the individuals in world coordinates and inspect all the data for this best image. Clearly transmitting the video stream from the sensor is not feasible in this context since this would easily exceed the bandwidth and energy available to a node. On the other extreme, each node can locally search for face images in its own view. This approach wastes a lot of energy because only a small fraction of views can contain a good quality face image of an individual at a given time. Furthermore this method does not provide associations of face images to individuals. It is not clear how a node can know when the global sensing task of acquiring a canonical face image for each individual has been completed so that it may return to a low power state.

The proposed method described below avoids both problems by fusing some lightweight data from multiple sensors in realtime to observe global properties of the objects. This information can be used to continuously task a small subset of the cameras to cache images in local storage. The locations of the best views are determined and the actual image data can be retrieved in a lazy fashion.

First we need to define what makes one face image better than another. Clearly a large image is more useful than a smaller one for the purpose of recognition. Also an image from a consistent frontal viewpoint is desireable because it can be used with standard face recognition algorithms like EigenFaces [5]. Thus an image in which the person is more directly facing the camera is better. Though estimating head orientation from images has been studied in the literature [6], we use a simpler method using information already available from the tracker. We observe that people typically face the direction in which they are walking and simply use the direction of motion as a cheap but noisy approximation of head orientation.

To formally quantify the notion of view quality, a function $S$ is introduced to score the view from camera $C_i$ of the face of person $P_j$ at time $t$ as follows. Let $\mathbf{v}_j$ be the velocity of person $P_j$ at time $t$ and $\widehat{\mathbf{v}}_j$ be the unit vector in the direction of $\mathbf{v}_j$. Let $\mathbf{d}_{ij}$ be a unit vector pointing from the person $P_j$ towards camera $C_i$. Also let $f_{occ} = \frac{\theta_{occ}}{\theta_{FOV}}$ be the fraction of the field of view of camera $C_i$ occluded by the head of person $P_j$ as illustrated in Figure 2. The view scoring function is defined as:

$$S(C_i, P_j, t) = \begin{cases} (\widehat{\mathbf{v}}_j \cdot \mathbf{d}_{ij}) f_{occ} & \text{if ScoreValid} \\ 0 & \text{otherwise} \end{cases}$$

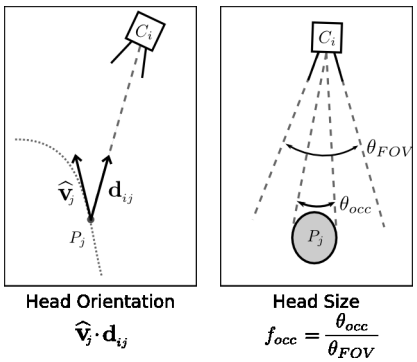


**Fig. 2**. The camera-to-person view quality score $S(C_i, P_j, t)$ has two components. The variables of the head orientation component and head size components are illustrated on the left and right respectively. On the left, the person is moving along the curved path and the head orientation is approximated by the tangent to the path. On the right, $f_{occ}$ is the fraction of the camera's field of view subtended by the person's head.

The $\mathrm{ScoreValid}$ condition is satisfied when the tracked person is moving faster than some minimum speed and there is no other person between person $P_j$ and camera $C_i$ that can occlude the view. The minimum speed condition is needed because the approximation of head orientation by motion is reasonable only when the person is walking. The score consists of the following two terms. The first term is maximized when the direction to the camera and the direction of travel are the same. When the head orientation assumption holds, this gives higher scores to direct frontal face images than to others. The second term is largest when the face region fills the entire camera field of view and thus gives preference to images in which the face appears larger.

Given this view scoring function, the tasking algorithm works as follows. After the cluster leader begins tracking a new person, it starts a face acquisition process for that track. The process monitors the trajectory of the target and calculates a score according to the above metric for each camera. At a regular interval, the cameras with the highest scores are commanded to cache their video stream in local storage. The process monitors the trajectory during a period of observation until the object leaves, or until it is requested to provide the current best face image. The process then examines the history of scores and selects the camera and instant of time associated with the highest score (i.e. a likely best canonical image). The process requests that the camera extracts the portion of this best shot from its cached video and verify the existence of a frontal face image. The camera sensor does this by running the Viola-Jones face detection algorithm [7] on the small patch that should contain the head of the tracked person (which is easily calculated from the known camera geometry). This face detection operation is performed infrequently

for verification of the predicted face location and is not run continuously on full images to search for a face. By verifying the predicted face location with the face detector, the algorithm can detect when the head orientation approximation was wrong and try again. The FaceNet camera tasking algorithm is outlined in Algorithm 1.

**Algorithm 1** Sensor Tasking for Face Image Retrieval

**while** $P_j$ is tracked **do**
  $\text{bestCamId} \leftarrow \arg\max_i S(C_i, P_j, t)$
  $\text{bestCamScore} \leftarrow S(C_{\text{bestCamId}}, P_j, t)$
  **if** bestCamScore $>$ taskingThreshold **then**
    command $C_{\text{bestCamId}}$ to cache images locally
    taskingHistory[t].score = bestCamScore
    taskingHistory[t].camId = bestCamId
  **end if**
  $t \leftarrow t + 1$
  waitUntilTime(t)
**end while**
**while** no face image for $P_j$ **do**
  $t_{\text{best}} = \text{findTimeOfBestScore(taskingHistory)}$
  $\text{bestCamId} = \text{taskingHistory}[t_{\text{best}}].\text{camId}$
  checkForFaceInCachedImage($C_{\text{bestCamId}}, t_{\text{best}}$)
**end while**

## 4. RESULTS

### 4.1. Experimental Setup

To evaluate this approach on real data, we built a camera network called FaceNet from off-the-shelf hardware. The FaceNet deployment consists of sixteen web cameras connected to eight shuttle PCs that communicate over an ethernet network. The cameras are deployed horizontally at eye level on the walls of a 6m x 7m room and arranged as depicted in Figure 3. This approach was used to facilitate fast development and testing and to provide a platform where a wide range of resource constraints (memory, bandwidth) can be simulated. Intel's OpenCV library was used for image processing operations on the camera nodes and the open source ICE middleware was used for distributed object communication.

In this simple configuration, all cameras observe the same space and thus form a single logical cluster. One of the PC's serves as the cluster leader for data fusion and tasking. The FaceNet GUI application displays a summary of human activity in the form of trajectories and an associated face image for each tracked person. The GUI display shown in Figure 3 shows two people being tracked in real-time.

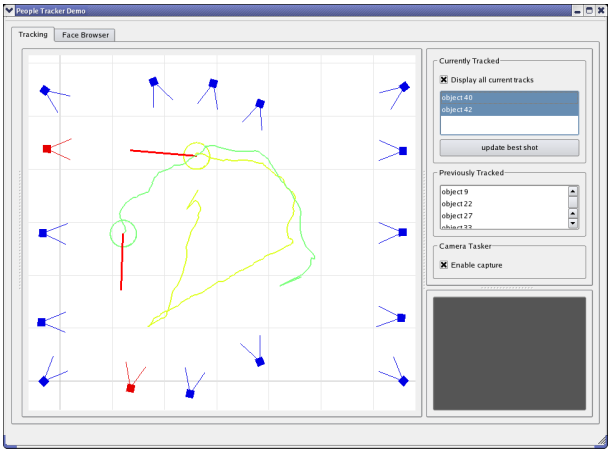


**Fig. 3**. Plan view diagram of the single-room FaceNet deployment - In this image, two people are tracked in real time using the lightweight visual hull tracking algorithm and two cameras have been tasked to cache images.

### 4.2. Evaluation of Lightweight Visual Hull Tracking

Reliable tracking in the prototype network has been achieved for three or four people interacting in a natural way. In order to resolve and track an individual, this method requires that there exists at least one view in which the individual can be isolated from other foreground objects. In moderately crowded situations where blobs frequently merge and split, identity management techniques as proposed in [8], [9], [10] could be applied to extend short tracks into more useful long tracks. In more crowded situations where the visual hull fails to isolate individuals at all, a different tracking algorithm would be required to track individuals instead of groups of individuals.

### 4.3. Energy savings

To evaluate the energy that could be saved by this "smart camera" method, we compare the performance with a conventional "dumb camera" architecture where no image analysis is done locally. In the comparison architecture, data must sent to a central point for processing to complete the same tracking and face image selection task. To be generous, assume the cameras in the comparison system can aggressively compresses video to a 37.5 KB/s stream and send data only when motion is detected (perhaps when triggered by a passive infrared sensor). To make the comparison independent of the platform, we compare the required communication in bytes since this is the dominate factor in this wireless application.

The table below compares the communication costs of the two methods for the trial shown in Figure 3. In this trial, two people are tracked in realtime using the visual hull tracking algorithm over a period of five minutes. Motion was detected in 76 percent of the video frames during the observation period

leading to an average data rate of 28.5 KB/s from each camera in the centralized processing system. To compare with FaceNet, we calculate the data rate for both image retrieval and tracking tasks (including transmitting the compact foreground representations). Because only 14 small images are transmitted from the FaceNet cameras, the cost of transmitting image data is very small. For this example, the FaceNet method reduces the total amount of data transmitted from the cameras by 97 percent.

| Average data sent per camera | FaceNet | Centralized |
|---|---|---|
| Image data rate (bytes/s) | 7.4 | 28500 |
| Tracking data rate (bytes/s) | 967 | 0 |
| Total data rate (bytes/s) | 974.15 | 28500 |

### 4.4. Evaluation of Tasking Algorithm

To evaluate the proposed sensor tasking algorithm on its own in a challenging real-world setting, we conducted a trial involving 11 people interacting in the FaceNet room for 4 minutes. The setting was a "social mixer" type event where people circulated around the room while talking and eating. Because the performance of the tasking algorithm depends on the accuracy of the tracking data it uses, we will evaluate the tasking algorithm independently from the visual hull tracking algorithm described above. Wide-angle cameras mounted on the ceiling were used to record video that was processed offline to estimate ground truth tracks. These ground truth tracks were provided as input to the tasking algorithm for the following evaluation.

#### *4.4.1. Face Image Acquisition Failure*

The goal of this section is to evaluate how effectively the tasking algorithm finds a canonical face image for each track. To do this, we define some measures of system performance, explain the primary parameters influencing performance, and present some measurements of the effect of these parameters.

To measure performance, we define two kinds of failures. A false positive failure occurs when the system selects an image that is either not a face or is the wrong face. Alternatively, a false negative error occurs when the system fails to find a face image for a track when some camera did in fact have a suitable view. For example, a false positive failure could occur if Sam leans forward to tie his shoe at the wrong moment and the system accidentally acquires the face of Sally who was standing behind him. A false negative might occur if Sam starts walking backwards to defeat the system. In this case, any cameras opposite the direction of motion have a clear view of the face but are never tasked to cache images. Thus the system fails to find a face image when one is potentially available.

Next we describe several important system parameters and their associated performance trade-offs.

**Sensor Memory:** Each camera sensor has a limited amount of memory in which to cache images locally. If the camera is out of memory, it discards an old image to cache a new one. If the amount of video a camera can store is shorter than the duration of a track, useful images of that track may be lost. In this case, the false negative rate could increase and the selected image quality may decrease. This parameter controls a trade-off between sensor hardware cost and performance.

**Tasking Threshold:** For a camera to be tasked to acquire images of a track, the view quality score for a camera-track pair must be above a threshold $\mathrm{taskingThreshold}$ (See Algorithm 1). Setting this value low reduces the likelihood that we fail to cache a potential good image, but increases the fraction of the time a camera sensor must be active and capture images. This parameter controls a trade-off between energy consumption and performance.

**Number of Cameras:** Using more cameras can increase the diversity of viewpoints and increases the likelihood of capturing a good face image when there are many occluders. This parameter controls a trade-off between deployment cost and system performance.

Figures 4, 5, and 6 show the effect of these parameters on the failure rates for the trial with 11 people described above. In general, the false positive failures are independent of these parameters. False positive failures were generally caused by errors in the estimated location of the target which caused the predicted face region to accidentally include the face of someone behind the target individual. Figure 4 indicates that increasing the number of images that a camera can store beyond 75 frames does not significantly improve the performance. With the cameras operating at 15 FPS, this corresponds to a capacity of 5 seconds of video (corresponding to the duration of many tracks). Figure 5 indicates that the tasking threshold can be increased from 0.01 up to 0.05 to save power before performance begins to degrade. To evaluate the effect of the number of cameras on performance, the tasking algorithm was run using several subsets of the full 16 cameras. The results shown in Figure 6 indicate that relatively good performance can be obtained using just 5 of the 16 cameras in this trial. The successful use of a much smaller number of cameras is likely due to the high mobility of the targets in this trial.

### 4.5. Face image quality

Figure 7 shows a series of face images acquired by FaceNet and sorted by their predicted view quality score. In general, it seems that the the view quality prediction made by the scor-

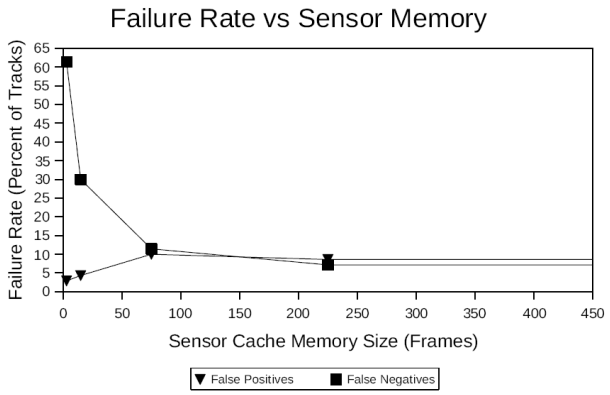


**Fig. 4**. False positive and negative failure rates as a function of the number of frames a sensor can store

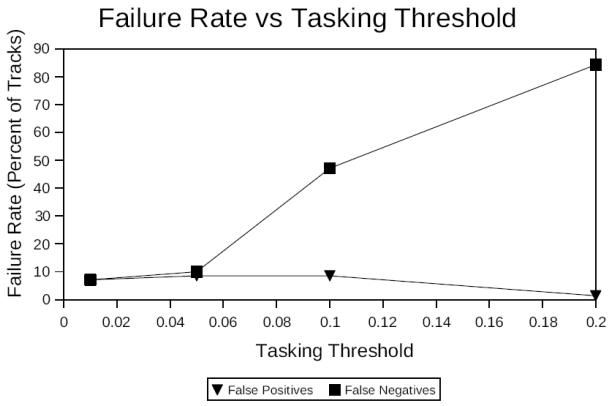


**Fig. 5**. False positive and negative failure rates as a function of the tasking threshold

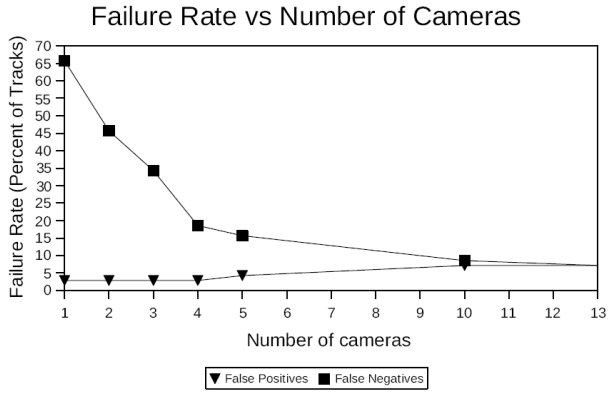


**Fig. 6**. False positive and negative failure rates as a function of the number of cameras


**Fig. 7**. A series of retrieved face images sorted by their predicted view quality score

ing function does correspond to an intuitive sense of image quality. However the last two images show that differences in face orientation are not necessarily reflected in the score (because of the simple approximation used).

To evaluate the quality of the returned face image, we compare the selected face image with one chosen by a human. To do this, a volunteer manually inspected all the recorded video of each track from all cameras and picked the best frontal face image. The volunteer was directed to pick the largest frontal face images possible. Figure 8 is a side-by-side comparison of the human and algorithm selected canonical frontal images for 10 samples from the 64 tracks in the trial. In general, the manually selected images are larger than the automatically selected images but have similar head orientations. The larger size is to be expected since the set of images available to the human includes images when the person is not moving which are not considered by the algorithm. It is interesting that in 6 out of the 10 samples, FaceNet chose an image nearly identical to that chosen by a human.

## 5. RELATED WORK

The task of tracking and identifying people from video data has long been a topic of interest in the field of computer vision. The literature on tracking from single and multiple views is large and the VSAM project is just one of many noteable projects in the area of automated visual surveillance [11]. Most classical approaches assume that processing of image data is done in a centralized fashion and that large computational and communication resources are available for the task. With the increasing availability of small and low cost cameras, storage, and processing hardware, there is growing interest in performing some visual surveillance operations in a wireless sensor network. The Panoptes project [12] investigated ways of indexing and searching video data collected by many wireless cameras and studied the power usage of a 802.11 wireless platform. The Meerkats project [13] also developed an 802.11 based wireless platform (the Meerkat node) to study tradeoffs between energy and performance in the context of tasks like sensing moving people.

The visual hull is used in various settings for 3D reconstruction. A probabilistic framework for reconstruction of 3D objects is given in [4] where a camera is modeled as a probabilistic space occupancy sensor along a ray. In our approach, we are satisfied with calculating just a 2D slice of the full 3D

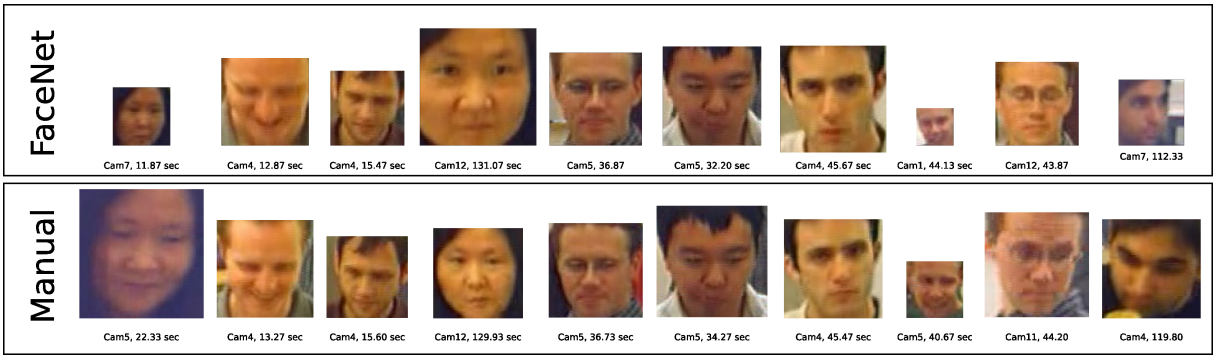


**Fig. 8**. Comparison of manually and automatically selected best frontal face images for 10 tracks.

visual hull. Our approach to tracking people using a 2D visual hull lies on a continuum between the very lightweight approach used in [3] and the approach used in [14]. In [3] a compact "scanline" representation of the foreground is created by summing the columns of a binary thresholded foreground image. However, using the scan-line representation for people tracking imposes restrictions on camera placement and is not well suited for information rich overhead views, for example. In [14] a foreground probability image is computed for each view and this image is then projected onto a plane parallel to the ground. Tracking is done offline by segmenting out coherent "snakes" traced by blobs moving through the space time volume formed by stacking the visual hull images. The advantage of this method is that it pushes back the foreground segmentation into the fusion step which reduces the effect of noisy foreground segmentation. The foreground representation used here represents a variable resolution foreground probability image that can be made nearly as compact as the scan-line method of [3] or a very close approximation to the full probability image used in [14].

## 6. CONCLUSION

A method has been described for tracking people and acquiring canonical face images which respects the requirements of a network with energy constrained wireless camera sensors. The FaceNet system saves energy by significantly reducing the amount of data transmitted from the wireless sensors. In a benchmark comparing FaceNet with a conventional centralized approach to this surveillance task, the FaceNet approach transmits 97 percent less data. Trajectories tagged with canonical images are a very compact and useful high-level representation which generalizes to other targets of interest (vehicles, animals).

A significant limitation of the lightweight visual hull based tracking approach described above is that it does not scale well to tracking many people in dense configurations. In order to resolve and track an individual, this method requires that there exists at least one view in which the individual can be isolated from other foreground objects. Reliable tracking in the prototype network has been achieved for up to three or four people interacting in a natural way. Future work will investigate extensions and alternative lightweight tracking algorithms that can handle more realistic dense configurations of moving targets.

**Acknowledgments: The authors wish to acknowledge support of ARO grant W911NF-06-1-0275, NSF grants CNS-0435111 and CCF-0634803, and an NSF graduate fellowship.**

## 7. REFERENCES

[1] Tomas Svoboda, Daniel Martinec, and Tomas Pajdla, "A convenient multi-camera self-calibration for virtual environments," Tech. Rep., Swiss Federal Institute of Technology, 2005.

[2] Stanislav Funiak, Carlos E. Guestrin, Mark A. Paskin, and Rahul Sukthankar, "Distributed localization of networked cameras," in *Proceedings of the Fifth International Symposium on Information Processing in Sensor Networks*, 2006.

[3] D.B. Yang, H.H. Gonzalez-Banos, and L.J. Guibas, "Counting people in crowds with a real-time network of simple image sensors," in *Proc. Ninth IEEE International Conference on Computer Vision*, 2003.

[4] Jean-Sebastien Franco and Edmond Boyer, "Fusion of multi-view silhouette cues using a space occupancy grid," in *International Conference on Computer Vision*, 2005.

[5] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1991.

[6] Nicolas Gourier, Daniela Hall, and James L. Crowley, "Estimating face orientation from robust detection of salient facial structures," in *FG Net Workshop on Visual Observation of Deictic Gestures*, 2004.

[7] Paul Viola and Michael Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2002.

[8] L. Guibas J. Shin and F. Zhao, "A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks," in *Proc. of 2nd International Workshop on Information Processing in Sensor Networks*, 2003.

[9] S. Thrun J. Shin, N. Lee and L. Guibas., "Lazy inference on object identities in wireless sensor networks," in *Proc. Fourth International Conference on Information Processing in Sensor Networks*, 2005.

[10] Brad Schumitsch, Sebastian Thrun, Leonidas Guibas, and Kunle Olukotun, "The identity management kalman filter (imkf)," in *Proceedings of Robotics: Science and Systems*, 2006.

[11] Collins, Lipton, Kanade, Fujiyoshi, Duggins, Tsin, Tolliver, Enomoto, and Hasegawa, "A system for video surveillance and monitoring: Vsam final report," Tech. Rep., Robotics Institute, Carnegie Mellon University, 2000.

[12] Wu chi Feng, Ed Kaiser, Wu chang Feng, and Mickael Le Baillif, "Panoptes: Scalable low-power video sensor networking technologies," in *ACM Transactions on Multimedia Computing, Communications and Applications*, 2005.

[13] C. B. Margi, X. Lu, G. Zhang, G. Stanek, R. Manduchi, and K. Obraczka, "Meerkats: A power-aware, self-managing wireless camera network for wide area monitoring," in *Distributed Smart Cameras Workshop - SenSys06*, 2006.

[14] Saad M. Khan and Mubarak Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," in *European Conference on Computer Vision*, 2006.