Respectful Cameras: Detecting Visual Markers in Real-Time to Address Privacy Concerns

Jeremy Schiff, Marci Meingast, Deirdre K. Mulligan,
Shankar Sastry, and Ken Goldberg

Abstract To address privacy concerns regarding digital video surveillance cameras, we propose a practical, real-time approach that preserves the ability to observe actions while obscuring individual identities. In the Respectful Cameras system, people who wish to remain anonymous wear colored markers such as hats or vests. The system automatically tracks these markers using statistical learning and classification to infer the location and size of each face. It obscures faces with solid ellipsoidal overlays, while minimizing the overlay area to maximize the remaining observable region of the scene. Our approach uses a visual color-tracker based on a nine dimensional color-space using a Probabilistic Adaptive Boosting (AdaBoost) classifier with axis-aligned hyperplanes as weak hypotheses. We then use Sampling Importance Resampling (SIR) Particle Filtering to incorporate interframe temporal information. Because our system explicitly tracks markers, our system is well-suited for applications with dynamic backgrounds or where the camera can move (e.g. under remote control). We present experiments illustrating the performance of our system in both indoor and outdoor settings, with occlusions, multiple crossing targets, lighting changes, and observation by a moving robotic camera. Results suggest that our implementation can track markers and keep false negative rates below 2%.

Jeremy Schiff
Department of EECS, University of California Berkeley, e-mail: jschiff@eecs.berkeley.edu

Marci Meingast
Department of EECS, University of California Berkeley, e-mail: marci@eecs.berkeley.edu

Deirdre K. Mulligan
Faculty of the School of Information, University of California Berkeley,
e-mail: dmulligan@law.berkeley.edu

Shankar Sastry
Faculty of the Department of EECS, University of California Berkeley,
e-mail: sastry@eecs.berkeley.edu

Ken Goldberg
Faculty of Departments of EECS and IEOR, University of California Berkeley,
e-mail: goldberg@berkeley.edu

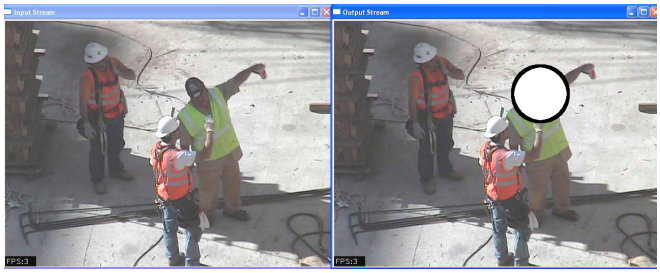

Fig. 1 A sample video frame is on left. The system has been trained to track green vests such as the one worn by the man with the outstretched arm. The system output is shown in the frame on the right, where an elliptical overlay hides the face of this man. The remainder of the scene including faces of workers not wearing green vests, remain visible. Note how the system successfully covers the face even when the vest is subjected to a shadow and a partial occlusion. Please visit "http://goldberg.berkeley.edu/RespectfulCameras" for more examples including video sequences.

1 Introduction

Since September 11, 2001, security concerns have led to increasing adoption of surveillance systems, raising concerns about "visual privacy" in public places. New technologies allow for the capture of significantly more detailed information than the human eye can perceive. Surveillance technologies are additionally empowered by digital recording, allowing footage to be stored indefinitely, or processed and combined with additional data sources to identify and track individuals across time and physical spaces. Robotic cameras can be servoed to capture high resolution images over a wide field of view. For example, the Panasonic KX-HCM280 pan-tilt-zoom camera costs under $750 and has a built-in web-server and a 21x optical zoom (500 Mpixels per steradian). The applications of these surveillance technologies extends beyond security, to industrial applications such as traffic monitoring and research applications such as observing public behavior.

McCahill et al. estimate that there are approximately 4 million public surveillance cameras deployed in the UK [35]. The U.S. has also deployed a number of camera systems in cities such as New York and Chicago for public monitoring [4, 37, 36]. Deployments of such large-scale government-run security systems, in conjunction with numerous smaller-scale private applications, raise fundamental privacy concerns.

In this chapter, we explore an approach we call "Respectful Cameras", that allows monitoring of activity but hides the faces of people who choose to wear recognizeable markers such as hats or vests that are made available. The Respectful Cameras system allows human actions to be observable so that people can monitor what is going on (ie, at a construction site or airport terminal) for security or public relations purposes. We envision such a system being made widely available, as these markers would be cheap, unobtrusive, and easily mass-produced. For example, we could provide inexpensive hats of a particular color or pattern at the border of the

space where cameras are present, similar to the respectful hats or leg-coverings that are made available at the entrance of churches and synagogues.

Existing face and people tracking methods have difficulty tracking in real-time under moving backgrounds, changing lighting conditions, partial occlusions and across facial orientations. We investigate a new approach that uses markers worn by individuals to simplify the quality of face or person detection required for obscuring individual identity, providing a method for individuals to opt-out of observation. These markers provide a visual cue for our system by having the features of the marker such as color, size, and shape to be distinguishable from the background. We use the location of the marker to infer the location of the faces of individuals who wish to "opt-out" of observation.

Recent advances in computer processing have made our algorithms utilizing AdaBoost and Particle Filtering feasible for real-time applications. Our approach learns a visual marker's color-model with Adaptive Boosting (AdaBoost), uses the model to detect a marker in a single image, and finally, applies Particle Filtering to integrate temporal information.

2 Related Work

Protecting the privacy of individuals has become increasingly important as cameras become more ubiquitous and have greater capabilities, particularly resolution and zoom. Examples of this interest includes The National Science Foundation (NSF) funding of TRUST [1], a research center for security and privacy, and privacy has been the subject of recent symposia such as Unblinking [2].

Changes in surveillance ubiquity and capabilities raise questions about the fair balance of police power (the inherent authority of a government to impose restrictions on private rights for the sake of public welfare, order, and security) to monitor public places versus the citizens' freedom to pass through public spaces without fear of government monitoring. According to Gavison, a loss of privacy occurs through visual surveillance by the extent we are known by others and subject to their attention [22]. He discusses our expectation that our actions are observable only by those we see around us, and thus we can judge how we should act. Nissenbaum describes how the high-resolution and zooming capabilities of cameras applied to visual surveillance also violate the contextual expectations of how people will be perceived in public [38]. This places the burden upon an individual to conduct himself or herself as if every move could be recorded and archived. Finally, it should be noted that it is not just surveillance that threatens privacy, but also the ability to be identified [49].

Researchers such as Hampapur et al. have analyzed the system architecture requirements of surveillance systems and built such a system [25]. Chen et al. built a system for people detection, tracking and recognition [11] using gait analysis, face detection and face recognition. As visual privacy is a significant concern of such systems, a number of researchers have explored how to integrate privacy systems

into these surveillance systems [48, 56, 12, 10, 18]. Others such as Chinomi et al. have evaluated different methods for obscuring people in video data [13]. Google has started experimenting with automated face blurring in Google Street View [23].

The ability to find the faces or full bodies of people is necessary for automated visual privacy. Applicable methods include face detection [9, 51, 50, 8], face tracking [15, 42], people detection [52], and people tracking [39, 53]. Unfortunately, these methods have difficulty detecting and tracking in real-time for domains with partial occlusions, arbitrary pose, changing backgrounds, and changing lighting conditions [30, 55]. Alternatively, Background Subtraction methods, such as Gaussian Mixture Models [31], can be applied to obscure foreground objects. These methods are generally not designed to disambiguate between different moving objects so that we could obscure only the people of interest and leaving other objects such as moving cars visible. However, work such as that of Senior et al. [47] uses object models as a post-processing step to disambiguate between objects and compensate for occlusions. Many Background Subtraction methods will eventually determine that a stationary person is part of the background, which will cause the stationary person to become unobscured. Also, these systems struggle with domains using robotic cameras, as the system will not have a background-model for the new region, which may contain people to be obscured. In contrast, because our system explicitly tracks markers, it is well adapted for scenes observed by robotic cameras.

Approaches to object detection employ statistical classification methods including AdaBoost [51, 40], Neural Networks [17], and Support Vector Machines (SVMs) [41]. Rather than using the person as the feature, we track a visual marker worn by the individual, and use a form of AdaBoost [20] to track the color of that feature. AdaBoost is a supervised learning approach that creates a strong statistical classifier from labeled data and a set of weak hypotheses, which poorly classify the labeled data. Rather than conventional AdaBoost that provides a binary label, we use Probabilistic AdaBoost [21, 32], which provides the probability of an input's label that we use in our Particle Filter.

When using AdaBoost for detecting objects, either pixel-based or region-based features can be used. Pixel-based approaches such as ours use a set of features for each pixel in the image, while region-based use features defined over a group of pixels. Typical region-based approaches explore applying Haar wavelet features to pixel regions [51, 7]. Avidan describes the use of a pixel-based method [6] where each pixel's initial feature vector contains the RGB values, as well as two histograms of oriented gradients similar to those used in Scale Invariant Feature Transform (SIFT) features [34]. These SIFT features are commonly used for problems such as the correspondence between images or in object detection. Rather than incorporating gradient information, our pixel-based approach uses multiple color-spaces as our feature vector.

The research community has also investigated the use of skin color as a mechanism for face detection. One common approach is verifying if each color satisfies a priori constraints [33], another is performing machine learning [16] to model face color. Others use multivariate Gaussians [54, 3] and Gaussians Mixture Models [24] to represent the face colors. Most approaches use a color space other than RGB

such as Hue, Saturation, Value (HSV) or YCrCB, but in contrast to our work, few use more than a single colorspace. It is also common to use pixel color as a pre-processing step to prune out image regions before using a more powerful intensity-based face detector [54]. As these systems evolve and are used in industry, it is important that these systems are trained over a wide variety of races and skin-tones so that the system does not work for some, and not for others.

Sharing our motivations of robust detection, the Augmented Reality community also simplifies object detection with visual markers for tracking and calibrating. Zhang et al. compared many of these methods [57]. Kohtake et al. applied visual markers to simplify object classification to ease the User Interaction problem of taking data stored in one digital device and moving it to another by pointing and selecting physical objects via an "infostick" [29].

Tracking can be used with object detection to enhance robustness. One common method of tracking, Particle Filtering [5, 44], is used to probabilistically estimate the state of a system, in our case, the location of a visual marker, via indirect observations, such as a set of video images. Particle Filtering provides a probabilistic framework for integrating information from the past into the current estimation. Particle Filtering is non-parametric, representing the distributions via a set of many samples, as opposed to parametric approaches that represent distributions with a small set of parameters. For instance, Kalman Filtering [27], represents distributions by a mean and a variance parameter. We choose Particle Filtering because our observation model is non-Gaussian, and thus methods like Kalman Filtering will perform poorly.

Perhaps closest to our approach, both Okuma et al. [39] and Lei et al. [32] also use a probabilistic AdaBoost formulation with Particle Filtering [32]. However, both assume a classifier per tracked-object (region-based), rather than classifier per-pixel. As our markers use pixel-based color, we don't need to classify at multiple scales, and we can explicitly model shape to help with robustness to partial obstructions. Okuma's group applies their approach of dynamic weighting between a Particle Filter and an AdaBoost Object Detector to tracking hockey players. Rather than weighting, our approach directly integrates AdaBoost into the Particle Filter's observation model. Lei et al. uses a similar approach to ours, and performs face and car tracking. However, unlike Lei, our formulation can track multiple objects simultaneously.

A preliminary version of chapter appeared as a paper in IROS 2007 [46].

3 System Input

Our system relies on visual markers worn by individuals who wish to have their face obscured. Our input is the sequence of images from a video stream. Let i be the frame number in this sequence. Each image consists of a pixel array where each pixel has a red, green, and blue (RGB) component.

4 Assumptions

We use the location and form-factor of each visual marker as a proxy for a location and form-factor of a corresponding human head. Thus, we assume there is an offset between the marker's location and the estimated face's location. Similarly, we assume the face's size will be proportional to the size of the visual marker. Intuitively, this means that as the marker's size shrinks, the face will shrink proportionally.

We make the following additional assumptions:

- Whenever a person's face is visible, then the visual marker worn by that person is visible.
- In each frame, all visible markers have a minimum number of visible, adjacent pixels

5 System Output

Our objective is to place solid ellipses to obscure the face of each individual wearing a marker, while minimizing the overlay area to allow observation of actions in the scene.

For each frame in the input stream, the system outputs a set of axis-aligned elliptical regions. These regions should completely cover all faces of people in the input image who are wearing markers. The *i*th output image has a set of elliptical regions E_i associated with it. Each element in E_i is defined by a center-point, denoted by an x and y position, and major and minor axis r_x and r_y:

$$E_i = \{(x, y, r_x, r_y)\} \tag{1}$$

The *i*th output video frame is the same as the *i*th input frame with the corresponding regions E_i obscured via solid ellipses.

Failing to detect a marker when one is present (false negative) is worse than placing an ellipse where there is no face (false positive).

6 Three Phases of System

Our solution consists of three phases: (A) offline learning of a statistical classifier for markers, (B) online marker detection and (C) online marker tracking.

6.1 Phase A: Offline Training of the Marker Classifier

We train a classifier offline, which we then use in the two online phases. For classification, we use the statistical classifier, AdaBoost, which performs supervised learning on labeled data.

6.1.1 Input and Output

A human "supervisor" provides the AdaBoost algorithm with two sets of samples as input, one for pixels colors corresponding to the marker T_+ and one for pixels colors corresponding to the background T_-. Each element of the set has a red value r, a green value g, a blue value b and the number of samples with that color m. Thus, the set of colors corresponding to marker pixels is

$$T_+ = \{(r,g,b,m)\} \tag{2}$$

and the sample set of pixels that correspond to background colors

$$T_- = \{(r,g,b,m)\} \tag{3}$$

As we are using a color-based method, the representative frames must expose the system across all possible illuminations. This includes maximum illumination, minimal illumination, the object under a shadow, and any potential hue effects caused by lighting phenomena such as a sunset. We discuss the AdaBoost formulation in more detail in Section 7.1.

We use a Probabilistic AdaBoost formulation that produces a strong classifier $\eta : \{0,\ldots,255\}^3 \mapsto [0,1]$. This classifier provides our output, a prediction of the probability that the RGB color of any pixel corresponds to the marker color.

6.2 Phase B: Online Static Marker Detector

For static detection, each frame is processed independently. This phase can be used on it's own to determine marker locations, or can be used as input to a dynamic method such as what we describe in Phase C.

6.2.1 Input and Ouput

The Marker Detector uses as input the model generated from AdaBoost, as well as a single frame from the video stream.

We can use the marker detector without tracking to infer the locations of faces. This would produce for the ith image, a set of regions E_i as defined in Section 5,

to obscure each face. We represent the state of each marker in the ith image as a bounded rectangle. We denote the set of rectangular bounding regions for each marker in image i as R_i. Each rectangular region is represented by a center-point, denoted by an x and y position, and it's size, denoted by a width Δx and a height Δy:

$$R_i = \{(x, y, \Delta x, \Delta y)\} \tag{4}$$

There is an explicit mapping between the size and location of the bounding regions R_i and the sizes and locations of elliptical overlays E_i. The rectangles in R_i are restricted by the assumptions described in Section 4, but have the flexibility to change its shape as the marker moves around the observed space. When used as a component of a marker tracker, the detector supplies the same set of rectangles for initializing the tracker, but also determines for each pixel the probability $P(I_i(u,v))$ that each pixel (u,v) corresponds to the visual marker in image I_i.

6.3 Phase C: Online Dynamic Marker Tracker

The dynamic marker tracker uses temporal information to improve the Online Detector. We do this by using information from the Static Detector along with a Particle Filter for our temporal model.

6.3.1 Input and Output

The dynamic marker tracker uses both the classifier determined in the training phase and output from the static image recognition phase. We process a single frame per iteration of our Particle Filter. Let the time between the previous frame and the ith frame be $t_i \in \mathbb{R}_+$, and the ith image be I_i. We discuss Particle Filtering in more depth in Section 9.1, but it requires three models as input: a prior distribution, a transition model, and an observation model. We use the Static Marker Detector to initialize a Particle Filter for each newly-detected marker. We also use the probabilities $P(I_i(u,v))$, supplied by the Static Marker Detector, to determine the posterior distribution of a marker location for each Particle Filter, given all previously seen images.

The output for the ith frame is also the set of regions E_i as defined in Section 5.

7 Phase A: Offline Training of the Marker Classifier

To train the system, a human "supervisor" left-clicks on pixels in a sample video to add them to the set T_+, and similarly right-clicks to add pixels to set T_-.

In this phase, we use the two sets T_+ and T_- to generate a strong classifier η, which assigns the probability that any pixel's color corresponds to the marker, providing $P(I_i(u,v))$. Learning algorithms are designed to generalize from limited amounts of data. For instance, with the AdaBoost algorithm, we needed a thousand labeled training samples. Also, as our classification algorithm is linear in the number of dimensions of our dataset (9 in our formulation) and number of hyperplanes used as weak hypotheses in the model (20 in our experiments), we can evaluate this classifier in realtime.

7.1 Review of AdaBoost

AdaBoost uses a set of labeled data to learn a classifier. This classifier will predict a label for any new data. AdaBoost constructs a strong classifier from a set of weak hypotheses.

Let X be a feature space, $Y \in \{-1,1\}$ be an observation space and $\{h : X \to Y\}$ be a set of weak hypotheses. AdaBoost's objective is to determine a strong classifier $H : X \mapsto Y$ by learning a linear function of weak hypotheses that predicts Y given X. At each iteration from $t = (1 \ldots T)$, AdaBoost incrementally adds a new weak hypothesis h_t to the strong classifier H:

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{5}$$

and

$$H(x) = \text{sign}\,(f(x)) \tag{6}$$

Let $\eta(x) = P(Y = 1|X = x)$, and define AdaBoost's loss function $\phi(x) = e^{-x}$. The objective of AdaBoost is to minimize the expected loss or

$$E(\phi(yH(x))) = \inf_H [\eta(x)\phi(H(x)) + (1-\eta(x))\phi(-H(x))] \tag{7}$$

This is an approximation to the optimal Bayes Risk, minimizing $E[l(H(X),Y)]$ with loss function

$$l(\widehat{Y},Y) = \begin{cases} 1 \text{ if } \widehat{Y} \neq Y \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

To determine this function, we use a set of training data $\{(x_i,y_i)|x_i \in X, y_i \in Y\}$ sampled from the underlying distribution.

AdaBoost is an iterative algorithm where at each step, it integrates a new weak hypothesis into the current strong classifier, and can use any weak hypothesis with error less than 50%. However, we use the greedy heuristic where at each iteration, we select a weak hypothesis that minimizes the number of incorrectly labeled data points [51]. We use a standard AdaBoost classifier of this form in Phase B, trained on our labeled data.

7.1.1 Recasting Adaboost to Estimate Probabilities

Typically, as described in [45], AdaBoost predicts the most likely label that an input will have. Friedman et. al describe how to modify the AdaBoost algorithm to produce a probability distribution, rather than a binary classification[21]. The modified strong classifier determines the probability that an input corresponds to a label of 1 (as opposed to -1) and is defined by

$$\eta(x) = P(Y = 1|X = x) = \frac{e^{2f(x)}}{1 + e^{2f(x)}} \tag{9}$$

We use this probabilistic formulation to determine the probability that each pixel corresponds to a marker, as $\eta(x) = P(I_i(u,v))$, in Phase C.

7.2 Determining Marker Pixels

We begin by applying Gaussian blur with standard deviation σ_I to the image, which enhances robustness to noise by integrating information from nearby pixels. We use these blurred pixels for T_+ and T_-. We then project our 3 dimensional RGB color space into the two additional color spaces, Hue, Saturation, Value (HSV) [19] and LAB [26] color-spaces. HSV performs well over varying lighting conditions because Value changes over varied lighting intensities, while Hue and Saturation do not. LAB is designed to model how humans see color, being perceptually linear, and is particularly well suited for determining specularities. This projection of RGB from T_+ and T_- into the nine-dimensional RGBHSVLAB color space is the input to AdaBoost.

For weak hypotheses, we use axis-aligned hyperplanes (also referred to as decision stumps),each of which divides the space along one of the nine dimensions into two sets. These hyperplanes also have a +/- direction, where all 9-dimensional tuples that are in the direction and above the hyperplane are labeled as visual marker pixels, and all other tuples are non-marker pixels. The hyperplane separating dimension d at a threshold j is described by:

$$h_{d,j}(X) = \begin{cases} 1 & \text{if } X[d] \geq j \\ -1 & \text{otherwise} \end{cases} \tag{10}$$

Our set of weak hypotheses also includes the complement of these hyperplanes $\overline{h_{d,j}}(X) = -h_{d,j}(X)$. By projecting the initial RGB space into the additional HSV and LAB spaces, we provide more classification flexibility as we have more weak classifiers. For the weak hypothesis, AdaBoost chooses the dimension and threshold at each round that minimizes the remaining error. The algorithm terminates after running for some constant number, n, iterations. There are many ways to set n, for instance splitting the data into a learning set and a validation set. In this technique, learning is applied to the learning set, and the generalization accuracy is evaluated

on the validation set. Thus, we can observe how our model performs not on the data it is exposed to, but at predicting other unobserved data.

Rather than using the learning set and validation set directly, we went through a calibration process where we recorded video, exposing the marker to all areas in the viewing range of the camera and over varied lighting conditions and marker orientations. We went through the process of collecting an initial set of 100 marker pixels and 100 background pixels, and varied n from 5 to 50. If there were still patches of the background or marker that were misclassified, we would add these mistakes to the model, and repeat the learning. For instance, if a portion of the background was misclassified as the marker, we would add some of those pixels to the background set, and repeat the learning. This iterative approach provided an intuitive way for the user to distinguish between the marker and the background.

8 Phase B: Online Static Marker Detector

This section describes our marker detection algorithm, using only the current frame. Once we have the strong classifier from AdaBoost, we apply the following steps: (1) Apply the same Gaussian blur to the RGB image as we did for training. (2) Classify each pixel in the image (3) Cluster marker pixels (4) Select all clusters that satisfy the constraints.

8.1 Clustering of pixels

To determine the pixels to marker correspondence, we apply the connected-component technique [43]. The connected component algorithm is applied to an image with two types of pixels, foreground and background pixels, or in our case, marker and non-marker. Connected component will recursively group adjacent foreground pixels into the same cluster. We assign a unique group id to each cluster's pixels. Thus, a connected group of marker pixels would be determined to correspond to the same marker as they have the same group id. This yields a set of marker pixels for each visual marker in the frame.

To remove false positives, we enforce additional constraints about each identified marker cluster. We verify there are at least ζ_1 pixels in the cluster, and that ratio of width (Δx) to height (Δy) falls within a specified range from ζ_2 to ζ_3:

$$\zeta_2 \leq \frac{\Delta x}{\Delta y} \leq \zeta_3 \tag{11}$$

Requiring at least ζ_1 pixels per cluster prunes out false positives from small areas of color in the background that do not correspond to the marker. We also found that in some backgrounds, there would be long vertical or horizontal strips similar to our

marker color, which would be incorrectly labeled as a marker as it was the same color. However, our markers have a bounded ratio between width and height. Using this knowledge helps remove these false positives.

9 Phase C: Online Dynamic Marker Tracker

We use Particle Filtering to incorporate temporal information into our models, improving robustness to partial occlusions. Particle Filtering requires a probability distribution for the likelihood of the state given the current, indirect observations. We provide this Observation Model by extending the theory from our Static Marker Detector from Phase B, using the probability that each pixel corresponds to a marker that is provided by the Probabilistic Adaboost formulation described in Section 7.1.1.

9.1 Review of SIR Particle Filtering

While there are many types of Particle Filters, we use the Sampling Importance Resampling (SIR) Filter as described in [5, 44]. It is a non-parametric method for performing state estimation of Dynamic Bayes Nets (DBNs) over discrete time. The state at the time of frame i is represented as a random variable Z_i with instantiation z_i and the evidence of the hidden state E_i with instantiation e_i. There are three distributions needed for SIR Particle Filtering: the prior probability distribution of the object's state $P(Z_0)$, the transition model $P(Z_i|Z_{i-1})$, and the observation model $P(E_i|Z_i)$. The prior describes the initial distribution of the object's state. The transition model describes the distribution of the object's state at the next iteration, given the current object state. Lastly, the observation model describes the distribution of observations resulting from a specific object's state. Particle Filtering uses a vector of samples of the state, or "particles," that are distributed proportionally to the likelihood of all previous observations $P(Z_i|E_{0:i})$. At each iteration, each particle is advanced according to the transition model, and then assigned a probability according to its likelihood using the observation model. After all particles have a new likelihood, they are resampled with replacement using the relative probabilities determined via the observation model. This results in a distribution of new particles which have integrated all previous observations and are distributed according to their likelihood. The more samples that are within a specific state, the more likely that state is the actual state of the indirectly observed object.

9.2 Marker Tracking

In this section, we define our transition models and our observation models for our Particle Filter. We also describe how to track multiple markers simultaneously.

9.2.1 Marker Model

The state of a marker is defined with respect to the image plane and is represented by a 6 tuple of a bounding box's center x and y positions, the height and width of the bounding box, orientation, and speed. As can be seen in Figure 2 this yields:

$$z = (x, y, \Delta x, \Delta y, \theta, s) \tag{12}$$

We model the marker in image coordinates, rather than world coordinates to improve the speed of our algorithms.

9.2.2 Transition Model

The transition model describes the likelihood of the marker being in a new state, given its state at the previous iteration, or $P(Z_i|Z_{i-1} = z_{i-1})$. Our model adds Gaussian noise to the speed, orientation, bounding-box width, and bounding box height and determines the new x and y position via Euler integration. Let $W \sim N(0,1)$ be a sample from a Gaussian with mean zero and standard deviation of one. The mean μ and standard deviation σ for each portion of our model are set a priori. Formally:

$$\begin{aligned}
x_i &= x_{i-1} + s_i \cdot \cos(\theta_i) \cdot t_i \\
y_i &= y_{i-1} + s_i \cdot \sin(\theta_i) \cdot t_i \\
\Delta x_i &= \Delta x_{i-1} + \sqrt{t_i} \cdot (\sigma_{\Delta x} \cdot W + \mu_{\Delta x}) \\
\Delta y_i &= \Delta y_{i-1} + \sqrt{t_i} \cdot (\sigma_{\Delta y} \cdot W + \mu_{\Delta y}) \\
s_i &= \max(0, \min(s_{\max}, s_{i-1} + \sqrt{t_i} \cdot \sigma_s \cdot W)) \\
\theta_i &= \theta_{i-1} + \sigma_\theta \cdot \sqrt{t_i} \cdot W
\end{aligned} \tag{13}$$

At each iteration, we enforce the width and height constraints for each particle described in Section 8.1. The sample from the Gaussian, after being scaled by μ and σ, must be rescaled according to $\sqrt{t_i}$ (as defined in Section 6.3.1) to compensate for changing frame rates.

9.2.3 Observation Model

The observation model describes the distribution of the marker's state given an image, but our formulation gives a probability per pixel, rather than per marker state. We use an objective function as a proxy for the observation model, which has a prob-

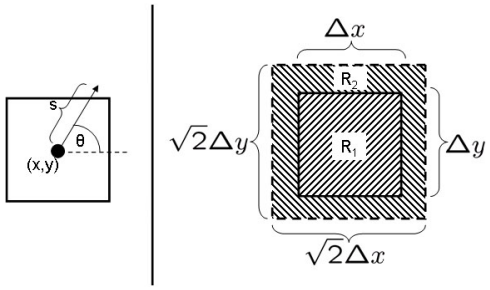

Fig. 2 Illustrates the state of a single bounding box (left) and the probability mask used for the Particle Filter's observation model (right).

ability of 1 if the bounding box tightly bounds a rectangular region of pixels with high probability. Let bounding box R_1 be the marker's state and bounding box R_2 have the same midpoint as R_1 but have size $\sqrt{2}\Delta x \times \sqrt{2}\Delta y$. R_1 and R_2 are disjoint. The $\sqrt{2}$ scaling factor makes the areas of R_1 and R_2 be equal. Then:

$$R_1 = \left\{ (u,v) \,\middle|\, |x-u| \leq \tfrac{\Delta x}{2}, |y-v| \leq \tfrac{\Delta y}{2} \right\} \tag{14}$$

$$R_2 = \left\{ (u,v) \,\middle|\, |x-u| \leq \tfrac{\Delta x}{\sqrt{2}}, |y-v| \leq \tfrac{\Delta y}{\sqrt{2}}, (u,v) \notin R_1 \right\} \tag{15}$$

$$P_1(Z_i = z_i | I_i) = \frac{1}{\Delta x \Delta y} \left(\sum_{(u,v) \in R_1} P(I_i(u,v)) \right) \tag{16}$$

$$P_2(Z_i = z_i | I_i) = \frac{1}{2\Delta x \Delta y} \left(\begin{array}{l} \sum_{(u,v) \in R_1} P(I_i(u,v)) + \\ \sum_{(u,v) \in R_2} 1 - P(I_i(u,v)) \end{array} \right) \tag{17}$$

Our final metric used as our observation model is:

$$P(Z | E_t = e_t) = (1 - P_1)P_1 + P_1 P_2 \tag{18}$$

This metric has the essential property that there is an optimal size for the bounding box, as opposed to many other metrics which quickly degenerate into determining the marker region to consist of all the pixels in the image or just a single pixel. For intuition, assume the projection of the visual marker produces a rectangular region. If a particle's bounding region is too large, its objective function will be lowered in region R_1, while if it is too small, then the objective function would be lowered in region R_2. This function yields a probability of 1 for a tight bounding box around a rectangular projection of the marker, yields the probability of 0 for a bounding box with no pixels inside that correspond to the marker, and gracefully interpolates in between (according to the confidence in R_1). We illustrate the two areas in Figure 2.

9.2.4 Multiple-Object Filtering

Our formulation uses one Particle Filter per tracked marker. To use multiple filters, we must address the problems of: (1) markers appearing, (2) markers disappearing and (3) multiple filters tracking the same marker. We make no assumptions about where markers can be obstructed in the scene.

For markers appearing, we use the output of the Marker Detection algorithm to determine potential regions of new markers. We use an intersection over minimum (IOM) metric, also known as the Dice Measure [14], defined for two regions R_1 and R_2 is:

$$IOM(R_1,R_2) = \frac{\text{Area}(R_1 \cap R_2)}{\min(\text{Area}(R_1),\text{Area}(R_2))} \tag{19}$$

If a Marker Detection algorithm has an IOM of more than a specified overlap γ_1 with any of Particle Filter's most likely location, then a Particle Filter is already tracking this marker. If no such filter exists, we create a new marker at this region's location by creating a new Particle Filter with the location and size of the detection region. We choose an orientation uniformly at random from 0 to 2π, and choose speed from 0 to the maximum speed $s_{\max}$ that is chosen a priori.

To address disappearing markers, we require that the probability of the state of at least one particle for a filter exceeds γ_2, otherwise the filter is no longer confident about the marker's location, and is deleted.

Multiple Particle Filters can become entangled and both track the same marker. If the IOM between two Particle Filters' exceeds the same threshold as appearing filters γ_3, we remove the filter that was created most recently. We remove the most recent to maximize the duration a Particle Filter tracks its marker.

10 Experiments

We ran two sets of experiments to evaluate performance. We experimented in our lab where we could control lighting conditions and we could explicitly setup pathological examples. We then monitor performance on video from a construction site as we vary model parameters. All tests involved video from a Panasonic KX-HCM280 robotic camera, transmitting an mJPEG stream of 640x480 images. We ran all experiments on a Pentium(R) CPU 3.4 GHZ.

Currently, the system has not been optimized, and we could easily extend our formulation to incorporate parallelism. The rate that we can process frames is about 3 frames per second, which is approximately 3x slower than the maximum incoming frame rate of 10fps.

For both the Lab Scenario and Construction Site, we trained the AdaBoost algorithm on 2 one-minute video sequences specific to the environment, using the method described in Section 6.1, exposing the system to many potential backgrounds, location and orientations of the visual markers, and over all lighting conditions that the experimental data experiences. After training AdaBoost, we used the

same sequences to calibrate our parameters. In our experiments we used:

$\sigma_{\Delta x}$	$= 25$ pixels	$\sigma_{\Delta y}$	$= 25$ pixels
σ_s	$= 100$ pixels	σ_θ	$= \frac{3}{2}\pi$ radians
$\mu_{\Delta x}$	$= 12.5$ pixels	$\mu_{\Delta y}$	$= 12.5$ pixels
$s_{\max}$	$= 300$ pixels	σ_I	$= 3.7 pixels$
ζ_1	$= 300$ pixels	ζ_2	$= \frac{1}{5}$
ζ_3	$= 5$	γ_1	$= 0.2$
γ_2	$= 0.4$	γ_3	$= 0.2$
# Particles	$= 2000$	# Weak Hypotheses	$= 20$

We define an image to be a false negative if any part of any face is visible and to be a false positive if there is an obscuring region in E_i that does not touch any of the faces in image i. These metrics are independent of the number of people in the scene.

To determine the statistics for each experiment, we processed each video with our system and then went through each image, frame by frame, and hand labeled each for false positives and false negatives. We then went through the same sequence twice more to help ensure quality results. This required approximately 30 seconds per frame, or nearly 60 hours of manual labeling for the experiments presented.

10.1 Lab Scenario Experiments

Within the lab, where we can control for lighting changes, we explore scenarios that challenge our system. Our marker is a yellow construction hat, and we assume the face is at the bottom-center of the bounding box and the same size as the hat. We evaluate how the system performs when 1) there are lighting conditions that the system never was trained on, and 2) two individuals (and their respective markers) cross. Lab experiments were run on 51 seconds of data acquired at 10 frames per second (fps). We summarize our results in the following table:

Lab Scenario Experiments

Experiment	# Frames	Correct	FPs	FNs	FP+FNs
Lighting	255	96.5%	0.0%	3.5%	0.0%
Crossing	453	96.9%	0.0%	3.1%	0.0%

Table 1 This tables shows the performance of in-lab experiments. To evaluate the system, we place each frame into the category of correctly obscuring all faces without extraneous ellipses, being a false negative but not false positive, being a false negative but no a false positive, and being both a false negative and false positive. We denote false negatives with FN and false positives with FP.

Fig. 3 Example of a False Negative with the Respectful Cameras System: A sample image frame input on left image, with output regions overlayed on right image. This sample illustrates where an intense light from a flashlight induced a specularity, causing the classifier to lose track of the hat. As a result, the right image has no solid white ellipses overlaid on the face as it should.

Fig. 4 Example of Crossing Markers and the Respectful Cameras System: A sample image frame input on left image, with output regions overlayed on right image. This sample illustrates tracking during a crossing, showing how the Particle Filter grows to accommodate both hats.

10.1.1 Lighting

In this setup, there is a single person, who walks past a flashlight aimed at the hat during two different lighting conditions. We experiment with all lights being on, and half of the lab lights on. In the brighter situation, the flashlight does not cause the system to lose track of the hat. However, in the less bright situation, the hat gets washed out with a specularity and we fail to detect the hat during this lighting problem. An explanation for why the specularity only was visible in the less bright situation is that our camera dynamically modifies the brightness of the image depending on the scene it observes. Thus, in the darker scene, the specularity would have been perceived to be brighter than in the brighter scene. We show one of the failing frames in Figure 3. In general, the system performs well at interpolating between observed lighting conditions, but fails if the lighting is dramatically brighter or darker than the range of lighting conditions observed during training.

Fig. 5 Example of the Respectful Cameras System: A sample image frame input on left image, with output regions overlayed on right image. This sample illustrates tracking after a crossing (one frame after Figure 4), showing how the system successfully creates a second filter to best model the current scene.

10.1.2 Crossing Markers

In this test, two people cross paths multiple times, at different speeds. Figure 4 shows how the system merges the two hats into a single-classified hat when they are connected, while still covering both faces. We are able to accomplish this via the biasses in our transition model, $\mu_{\Delta x}$ and $\mu_{\Delta y}$, which reduces false-negatives when multiple faces are in a scene. At the following frame in Figure 5, the system successfully segments what it determined to be a single hat in the previous frame into two two hats by creating a new Particle Filter.

10.2 Construction Site Experiments

The construction site data was collected from footage recorded in March, 2007 at the CITRIS construction site at the University of Berkeley, California, under UCB Human Subjects Protocol #2006-7-1[1]. Because we needed to respect the privacy of the construction workers and restrictions in the protocol, we limited our data acquisition to a one-week period. The video sequence presented contains a number of difficult challenges, particularly partial obstructions of the marker, significant changes in the background due to the tracking of the moving person with a robotic camera, and lighting differences including sharp changes from shadows. Also, the system observes areas of the scene it was not trained on, as the robotic camera moved 10 times to track the construction worker as he walked throughout the construction site. For the construction site, our marker is a green construction vest and we assume the face is located at the top-center of the vest, as we show in Figure 1. We first evaluate the performance of the system as we use different color-spaces used for input to AdaBoost. We then evaluate the differences in performance between the Particle Filtered approach from Phase C and the Static Marker Detector from Phase

[1] To contact UCB Human Subjects, refer to http://cphs.berkeley.edu/content/contact.htm.

B. All experiments were run on data acquired at 6 fps, simulating that the system can process at this speed, rather than the current capability of 3fps. This diminished recording speed (the max is 10 fps) was caused by requiring us to view the video stream to move the camera to follow a person during recording, while having the system store a secondary video stream to disk for later experimentation. The data suggests that our system can perform with a 6.0% false positive rate, and 1.2% false negative rate for this real-world application. We summarize our results over a 76 second (331 frame) video sequence from a typical day at the construction site in the following table:

Construction Site Experiments

Experiment	% Correct	FPs	FNs	FP+FNs
Only RGB	19.4%	68.6%	5.1%	6.9%
Only HSV	86.1%	11.5%	1.2%	1.2%
Only LAB	84.3%	10.9%	3.6%	1.2%
All 9 (RGB+HSV+LAB)	93.4%	5.4%	0.6%	0.6%
Static Marker Detector	82.8%	16.3%	0.0%	0.9%
Dynamic Marker Tracker	93.4%	5.4%	0.6%	0.6%

Table 2 This tables shows the performance of experiments at the CITRIS construction site. To evaluate the system, we place each frame into the category of correctly obscuring all faces without extraneous ellipses, being a false negative but not false positive, being a false negative but no a false positive, and being both a false negative and false positive. We denote false negatives with FN and false positives with FP.

10.2.1 Color Models

In this test, we investigate how our system performs by using different color spaces, specifically because we are only using simple axis-aligned hyperplanes as our weak hypotheses. We compare the algorithm's performance when just using RGB, just HSV, just LAB, and then the "All 9" dimensional color space of RGB+HSV+LAB. We determined that using all nine is superior in both false positive and false negative rates. This data suggests that color-spaces that explicitly decouple the brightness from the color (LAB and HSV) perform better than those that do not (RGB). This is probably exacerbated by our choice of weak hypotheses that decouple each dimension.

10.2.2 Particle Filtered Data

In this test, we evaluated performance between a non-Particle Filtered approach, where we just use each frame independently as described in Phase B, and using Particle Filtering as described in Phase C. We can see that the system significantly reduces the number of false-positives from 17.2% to 6.0%, while inducing slightly more false-negatives from 0.9% to 1.2%. There were two extra false-negatives in-

Fig. 6 Sample image frame input on left image, with output regions overlayed on right image. This sample illustrates how without Particle Filtering, even with the nine dimensional color-space, partial occlusions segment the visual marker, resulting in multiple small ellipses.

Fig. 7 Sample image frame input on left image, with output regions overlayed on right image. This sample illustrates how Particle Filtering overcomes partial occlusions, when using the nine dimensional color-space, yielding a single large ellipse.

duced by the Particle Filter, one from the shirt being cropped at the bottom of the scene, and one where the previous frame experienced extreme motion blur. We were very strict with our definitions of false-negatives as the face's visible region due to the partially cropped shirt is only 8 pixels wide.

Examples of input and output data can be found in Figures 8, 9, 10, 11, and 12.

11 Machine Learning Discussion

There are a number of other machine learning techniques other than AdaBoost that can be explored. Other machine-learning based classification approaches include K-Nearest Neighbors, Neural Networks and Support Vector Machines. AdaBoost was a good fit to our problem as our formulation has a fairly low dimensional space (our problem only has nine dimensions). Also, as we wanted our system to run in real-time, we wanted a lightweight classification approach. We explicitly chose weak hypotheses that were very fast to evaluate, and axis-aligned hyperplanes require only a lookup for the particular dimension, and a comparison with each selected weak hypothesis. This approach is similar to Viola and Jones' [51] motivation of using simple features that are fast to evaluation. A significant advantage of using

the greedy selection method for the next weak hypothesis is that the formulation can be thought of as performing a form of feature selection aswell. If the H dimension in HSV has poor prediction performance, the weak hypotheses associated with H will not be chosen, making the system more robust. This is possible because our weak hypotheses treat each color dimension independently. Feature selection methods such as wrapping [28] have shown to improve classification performance. The motivation of feature reduction approaches follows the principle Occam's Razor. We are searching for the "simplest" way to predict the classification. This AdaBoost formulation implicitly uses feature selection in the step which chooses the next weak hypothesis according to the greedy heuristic in [51]. Exploration of a very high dimensional space (like 10,000 dimensions) is computationally very difficult for our approach, as it would require we explore all weak hypotheses, for each step of the weak hypothesis selection process.

Additionally, we project the RGB space into the HSV and LAB spaces. The original images are provided in RGB, giving us the features in that color-space with no needed computation. As discussed earlier, HSV is more robust to changing lighting conditions, and LAB is better at handling speculiarities and is designed to be perceptually linear. As shown in our experiments, using this redundant formulation as well as the implicit feature reduction of AdaBoost, effectively utilizes the additional colorspaces. If all of the best predictive powers were just in the RGB colorspace, then the hyperplanes for HSV and LAB would never be selected. We found the chosen weak hypotheses spanned all of the colorspaces. Projecting into the other two color spaces gives us 3x the features to explore at each step, improving performance, as we show in our experiments. We also chose to use colorspace projections rather than other types of projections as we know they were designed to give robust properties such as attempts at being invariant to lighting conditions.

12 Conclusion

We have discussed the Respectful Cameras visual privacy system which tracks visual markers to robustly infer the location of individuals wishing to remain anonymous. We discuss a static-image classifier which determines a marker's location using pixel colors and an AdaBoost statistical classifier. We then extended this to marker tracking, using a Particle Filter which uses a Probabilistic AdaBoost algorithm and a marker model which incorporates velocity and interframe information.

It may be possible to build a Respectful Cameras method directly into the camera (akin to the V-chip) so that faces are encrypted at the hardware level and can be decrypted only if a search warrant is obtained.

While a 1.2% false negative rate for the CITRIS construction is encouraging, we would like it to be even lower to better address privacy applications. One encouraging idea is to run the Respectful Cameras system from multiple cameras, and cross reference the results. Assuming that the false negative rates of multiple cameras is independent, with three cameras, the false-negative rate would be 0.0002%.

For related links, press coverage, videos of our experiments, a discussion of practical and political issues, or to get updates about this project, please visit: http://goldberg.berkeley.edu/RespectfulCameras.

Acknowledgements Thanks to Ambuj Tewari for assisting in formulating the Probabilistic AdaBoost and Jen King for her help with relating this work to policy and law. Thanks to Anand Kulkarni, Ephrat Bitton and Vincent Duindam for their edits and suggestions. Thanks to Panasonic Inc. for donating the cameras for our experiments.

This work was partially funded by the a TRUST Grant under NSF CCF-0424422, with additional support from Cisco, HP, IBM, Intel, Microsoft, Symmantec, Telecom Italia and United Technologies. This work was also partially supported by NSF Award 0535218, and by UC Berkeleys Center for Information Technology Research in the Interest of Society (CITRIS).

Fig. 8 Input frames from the in-lab crossing experiment

Fig. 9 Output frames showing the Particle Filter with the nine-dimensional colorspace performs well under dynamic obstructions

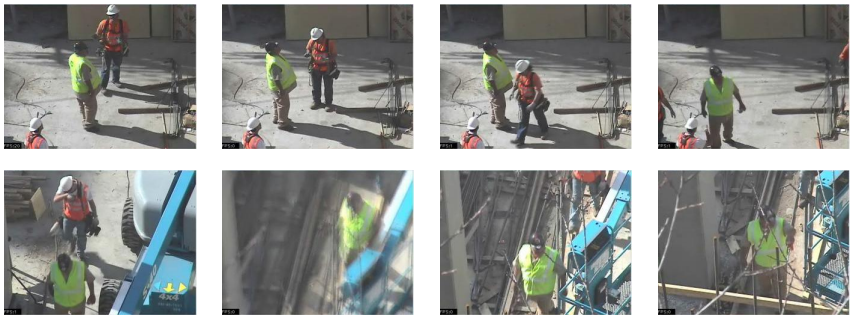

Fig. 10 Input frames from the CITRIS construction site

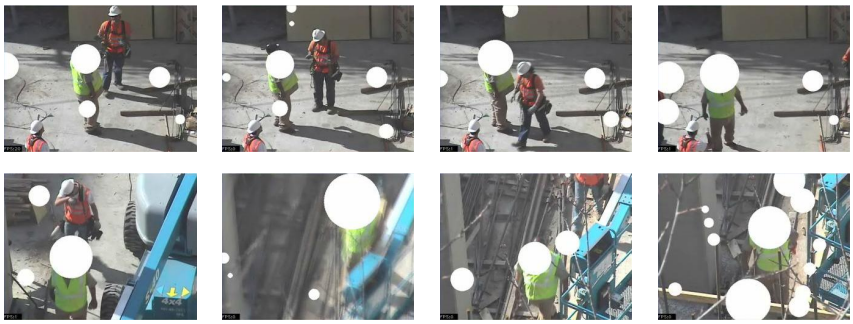

Fig. 11 Output frames showing using only the RGB colorspace is insufficient

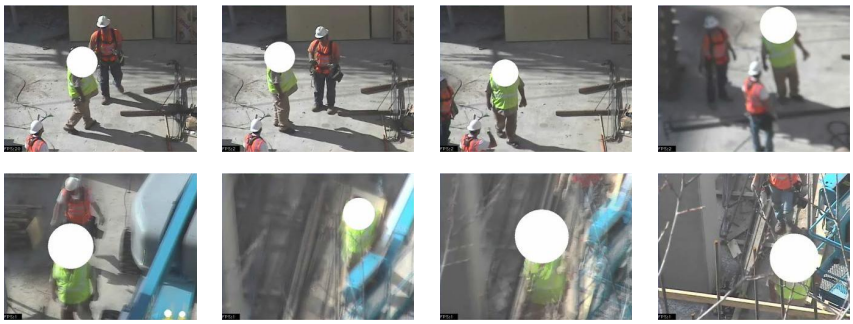

Fig. 12 Output frames using the full nine-dimension colorspace, showing better performance

References

1. TRUST: Team for research in ubiquitous secure technology. URL http://www.truststc.org/
2. Unblinking: New perspectives on visual privacy in the 21st century. URL http://www.law.berkeley.edu/institutes/bclt/events/unblinking/unblink.html
3. Amine, A., Ghouzali, S., Rziza, M.: Face detection in still color images using skin color information. In: Proceedings of International Symposium on Communications, Control, and Signal Processing (ISCCSP) (2006)
4. Anderson, M.: Picture this: Aldermen caught on camera. Chicago Sun-Times (2006)
5. Arulampalam, S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on Particle Filters for on-line non-linear/non-Gaussian Bayesian tracking. IEEE Transactions of Signal Processing **50**(2), 174–188 (2002)
6. Avidan, S.: Spatialboost: Adding spatial reasoning to AdaBoost. In: Proceedings of European Conference on Computer Vision, pp. 386–396 (2006)
7. Bahlmann, C., Zhu, Y., Ramesh, V., Pellkofer, M., Koehler, T.: A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In: IEEE Proceedings of Intelligent Vehicles Symposium, pp. 255–260 (2005)
8. Bourdev, L., Brandt, J.: Robust object detection via soft cascade. Proceedings of IEEE Conference Computer Vision and Pattern Recognition (CVPR) **2**, 236–243 (2005)
9. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library, 1st edn. O'Reilly (2008)
10. Brassil, J.: Using mobile communications to assert privacy from video surveillance. Proceedings of the IEEE International Parallel and Distributed Processing Symposium pp. 8 pp.– (2005)
11. Chen, D., Bharusha, A., Wactlar, H.: People identification across ambient camera networks. In: International Conference on Multimedia Ambient Intelligence, Media and Sensing (AIMS) (2007)
12. Chen, D., Chang, Y., Yan, R., Yang, J.: Tools for protecting the privacy of specific individuals in video. EURASIP Journal of Applied Signal Processing **2007**(1), 107–107 (2007)
13. Chinomi, K., Nitta, N., Ito, Y., Babaguchi, N.: PriSurv: Privacy protected video surveillance system using adaptive visual abstraction. In: S. Satoh, F. Nack, M. Etoh (eds.) MMM, *Lecture Notes in Computer Science*, vol. 4903, pp. 144–154. Springer (2008)
14. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology **26**(3), 297–302 (1945)
15. Dornaika, F., Ahlberg, J.: Fast and reliable active appearance model search for 3-D face tracking. IEEE Transactions of Systems, Man and Cybernetics, Part B **34**(4), 1838–1853 (2004)
16. Fang, J., Qiu, G.: A colour histogram based approach to human face detection. International Conference on Visual Information Engineering (VIE) pp. 133–136 (2003)
17. Feraud, R., Bernier, O.J., Viallet, J.E., Collobert, M.: A fast and accurate face detector based on neural networks. IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI) **23**(1), 42–53 (2001)
18. Fidaleo, D.A., Nguyen, H.A., Trivedi, M.: The networked sensor tapestry (NeST): a privacy enhanced software architecture for interactive analysis of data in video-sensor networks. In: Proceedings of ACM Workshop on Video Surveillance & Sensor Networks (VSSN), pp. 46–53. ACM Press, New York, NY, USA (2004)
19. Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F.: Computer Graphics Principles and Practice. AW, NY (1990)
20. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Computer and System Sciences **55**(1), 119–139 (1997)
21. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Annals of Statistics **28**(2), 337–407 (2000)
22. Gavison, R.: Privacy and the limits of the law. 89 Yale L.J. pp. 421–471 (1980)
23. Google Inc.: Privacy FAQ. URL http://www.google.com/privacy_faq.html#toc-street-view-images

24. Greenspan, H., Goldberger, J., Eshet, I.: Mixture model for face-color modeling and segmentation. Pattern Recognition Letters **22**(14), 1525–1536 (2001)
25. Hampapur, A., Borger, S., Brown, L., Carlson, C., Connell, J., Lu, M., Senior, A., Reddy, V., Shu, C., Tian, Y.: S3: The IBM smart surveillance system: From transactional systems to observational systems. Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on **4**, IV–1385–IV–1388 (2007)
26. Jain, A.K.: Fundamentals of digital image processing. Prentice Hall International (1989)
27. Kalman, R.: A new approach to linear filtering and prediction problems. Transactions of the American Society of Mechanical Engineers, Journal of Basic Engineering pp. 35–46 (1960)
28. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence **97**(1-2), 273–324 (1997)
29. Kohtake, N., Rekimoto, J., Anzai, Y.: InfoStick: An interaction device for inter-appliance computing. Lecture Notes in Computer Science **1707**, 246–258 (1999)
30. Kong, S.G., Heo, J., Abidi, B.R., Paik, J., Abidi, M.A.: Recent advances in visual and infrared face recognition: a review. Transactions of Computer Vision and Image Understanding (CVIU) **97**(1), 103–135 (2005)
31. Lee, D.S.: Effective gaussian mixture learning for video background subtraction. IEEE Transactions of Pattern Analysis and Machine Intelligence **27**, 827– 832 (2005)
32. Lei, Y., Ding, X., Wang, S.: AdaBoost tracker embedded in adaptive Particle Filtering. In: Proceedings of International Conference on Pattern Recognition (ICPR), vol. 4, pp. 939–943 (2006)
33. Lin, H.J., Yen, S.H., Yeh, J.P., Lin, M.J.: Face detection based on skin color segmentation and SVM classification. Secure System Integration and Reliability Improvement (SSIRI) pp. 230–231 (2008)
34. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. In: International Journal of Computer Vision, vol. 20, pp. 91–110 (2003)
35. McCahill, M., Norris, C.: From cameras to control rooms: the mediation of the image by CCTV operatives. CCTV and Social Control: The politics and practice of video surviellance-European and global perspectives (2004)
36. Moore, M.T.: Cities opening more video surveillance eyes. USA Today (2005)
37. New York Civil Liberties Union (NYCLU): Report documents rapid proliferation of video surveillance cameras, calls for public oversight to prevent abuses (2006). URL http://www.nyclu.org/whoswatching_pr_121306.html
38. Nissenbaum, H.F.: Privacy as contextual integrity. Washington Law Review **79**(1) (2004)
39. Okuna, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A boosted Particle Filter: Multitarget detection and tracking. In: Proceedings of Conference European Conference on Computer Vision (ECCV) (2004)
40. Opelt, A., Fussenegger, M., Pinz, A., Auer, P.: Weak hypotheses and boosting for generic object detection and recognition. In: Proceedings of Conference European Conference on Computer Vision (ECCV), pp. 71–84 (2004)
41. Osuna, E., Freund, R., Girosi, F.: Training support vector machines: an application to face detection. Proceedings of IEEE Conference Computer Vision and Pattern Recognition (CVPR) pp. 130–136 (1997)
42. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Proceedings of European Conference on Computer Vision (ECCV), pp. 661–675 (2002)
43. Rosenfeld, A.: Connectivity in digital pictures. J. ACM **17**(1), 146–160 (1970)
44. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Pearson Education (1995)
45. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Computational Learing Theory pp. 80–91 (1998)
46. Schiff, J., Meingast, M., Mulligan, D.K., Sastry, S., Goldberg, K.: Respectful cameras: Detecting visual markers in real-time to address privacy concerns. In: International Conference on Intelligent Robots and Systems (IROS), pp. 971–978 (2007)
47. Senior, A., Hampapur, A., Tian, Y., Brown, L., Pankanti, S., Bolle, R.: Appearance models for occlusion handling. Journal of Image and Vision Computing (IVC) **24**(11), 1233–1243 (2006)

48. Senior, A., Pankanti, S., Hampapur, A., Brown, L., Tian, Y.L., Ekin, A., Connell, J., Shu, C.F., Lu, M.: Enabling video privacy through computer vision. IEEE Security & Privacy **3**(3), 50–57 (2005)
49. Shaw, R.: Recognition markets and visual privacy. In: UnBlinking: New Perspectives on Visual Privacy in the 21st Century (2006)
50. Turk, M., Pentland, A.: Face recognition using Eigenfaces. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition (CVPR), pp. 586–591 (1991)
51. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) **01**, 511 (2001)
52. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. IEEE International Conference on Computer Vision (ICCV) **1**, 90–97 (2005)
53. Wu, B., Nevatia, R.: Tracking of multiple, partially occluded humans based on static body part detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) **1**, 951–958 (2006)
54. Wu, Y.W., Ai, X.Y.: Face detection in color images using AdaBoost algorithm based on skin color information. In: International Workshop on Knowledge Discovery and Data Mining (WKDD), pp. 339–342 (2008)
55. Yang, M., Kriegman, D., Ahuja, N.: Detecting faces in images: A survey. IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI) **24**(1), 34–58 (2002)
56. Zhang, W., ching S. Cheung, S., Chen, M.: Hiding privacy information in video surveillance system. Proceedings of IEEE International Conference on Image Processing (ICIP) **3**, II–868–71 (2005)
57. Zhang, X., Fronz, S., Navab, N.: Visual marker detection and decoding in AR systems: a comparative study. In: International Symposium on Mixed and Augmented Reality, pp. 97–106 (2002)