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

Jeremy Schiff[§], Marci Meingast[§], Deirdre K. Mulligan[‡], Shankar Sastry[§], Ken Goldberg[§†]
[§] Dept. of EECS, [‡] School of Law, [†] Dept. of IEOR
{jschiff@eecs|marci@eecs|dmulligan@law|sastry@eecs|goldberg@ieor}.berkeley.edu
University of California, Berkeley
Berkeley CA, 94720–1777, USA

***Abstract*— To address privacy concerns with digital video surveillance cameras, we propose a practical, real-time approach that preserves the ability to observe actions while obscuring individual identities. In our proposed Respectful Cameras system, people who wish to remain anonymous agree to wear colored markers such as a hat or vest. The system automatically tracks these markers using statistical learning and classification to infer the location and size of each face and then inserts elliptical overlays. The objective is to cover the face of each individual wearing a marker, while minimizing the overlay area to allow observation of actions in the scene. Our approach incorporates a visual color-tracker based on a 9 dimensional color-space. We train Probabilistic AdaBoost to find axis-aligned hyperplanes as classifiers. We then use Particle Filtering to incorporate interframe temporal information. We present experiments illustrating the performance of our system in both indoor and outdoor settings, where occlusions, multiple crossing targets, and lighting changes occur. Results suggest that the Respectful Camera system can reduce false negative rates to acceptable levels (under 2%).**

I. Introduction

The increasing prevalence and ever-improving capabilities of digital surveillance cameras introduce new concerns for visual privacy of individuals in public places. Advances in camera technologies allow for the remote observation of individuals beyond the mere recording of presence in an observed area without the individual's knowledge; instead, it changes the nature of vision itself. Robotic cameras can be servoed to observe high resolution images over a wide field of view. For example, the Panasonic KX-HCM280 pan-tilt-zoom camera costs under $1000 with a built-in web-server and a 21x optical zoom (500 Mpixels per steradian). 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. Such cameras are also quickly becoming affordable for commercial use, causing faster proliferation. Their applications extend beyond security, to industrial applications such as traffic monitoring and research applications such as observing public behavior. Their increased observational power enables data gathering about individuals far beyond the capabilities of perceptible human observers, and poses new challenges to individuals' sense of privacy in public.

*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).

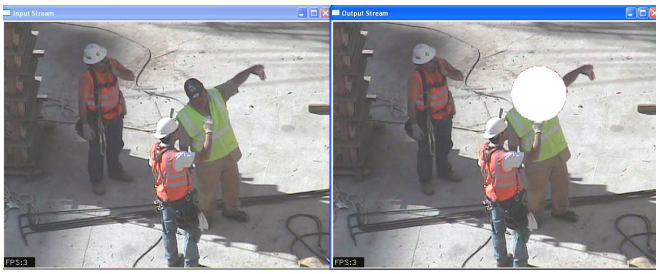

Fig. 1. Sample image frame input on left image, with output regions overlayed on right image. In this sample, the construction workers wearing green vests as markers are made anonymous, while faces of the other construction workers remain visible.

McCahill et al. estimated approximately 4 million cameras deployed in the UK [1]. The U.S. has also deployed a number of camera systems in cities such as New York and Chicago for public monitoring [2], [3], [4]. Deployments of such large-scale government-run security systems, in conjunction with smaller-scale private applications, raises fundamental privacy concerns which must be addressed. In this paper we consider the problem of automatically obscuring faces to assist in visual privacy enforcement. Our objective is to develop "Respectful Cameras."

We investigate a new approach for visual privacy that uses markers worn by individuals to simplify the level of robust person detection required for obscuring individual identity, providing a method for individuals to conceivably opt-out of observation. We would like the false negative rate (where we fail to obscure the face of an individual who wishes for privacy) to be under 2%. Existing face tracking methods have difficulty tracking faces in real-time under moving backgrounds, changing lighting conditions and partial occlusions. These markers provide a visual cue for our system by having a color that is distinct 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.

Our approach provides some level of visual privacy by hiding an individual's identity by obscuring their face with a colored ellipse, while allowing observation of his or

her actions. 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.

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

II. Related Work

Protecting privacy of individuals has become increasingly important as cameras become more ubiquitous and have greater capabilities, such as better resolution. The National Science Foundation (NSF) has sponsored TRUST [5], a new center for security and privacy, and privacy has been the subject of recent symposia such as [6].

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 [7]. 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 violates the contextual expectations of how people will be perceived in public [8]. 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 [9].

In order to provide automated privacy, the ability to find the faces or full bodies of people is necessary. Applicable methods include face detection [10], [11], [12], face tracking [13], [14], people detection [15], and people tracking [16], [17]. Unfortunately, these methods have difficulty detecting and tracking in real-time while being robust enough to address privacy under partial occlusions, and changing lighting conditions. Alternatively, motion detection methods such as Gaussian Mixture Models [18] can be applied, however they require time to learn a background model, during which, they cannot distinguish moving objects from the background.

Approaches to object detection employ statistical classification methods including AdaBoost [11], Neural Networks [19], and Support Vector Machines [20]. Rather than using the person as the feature, we track a visual marker worn by the individual, and use a form of AdaBoost [21] 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 of weak hypotheses, which poorly classify the labeled data. Rather than conventional AdaBoost that provides a binary label, we use Probabilistic AdaBoost [22], [23], which provides the probability of an input's label, and we use it in our Particle Filter formulation.

When using AdaBoost for detecting objects, there is a choice between using pixel-based and region-based features (we selected pixel-based). Pixel-based approaches use a set of features for each pixel in the image, while region-based usees features defined over a group of pixels. For region-based, typical approaches examined applying Haar wavelet features to pixel regions [11], [24]. Avidan describes the use of a pixel-based method [25] 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 [26]. These SIFT features are commonly used for problems such as the correspondence between images. Rather than incorporating gradient information, our pixel-based approach uses multiple color-spaces as our feature vector.

Sharing our motivations of robust (low false negative and false positive) detection, the Augmented Reality community also simplifies object detection with visual markers for tracking and calibrating. Zhang et al. compared many of these methods [27]. 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" [28].

After applying an object detection method, tracking can be employed to enhance robustness. Particle Filtering 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. Unlike Kalman Filtering [29], Particle Filtering is non-parametric, representing the distributions via a set of samples, rather than through a small set of parameters (for instance means and standard deviations for Gaussians used in Kalman Filtering). We choose to use Particle Filtering because our observation model is non-gaussian, and thus methods such as Kalman Filtering cannot be applied.

Perhaps closest to our approach, both Okuma et al. [30] and Lei et al. [23] also use a probabilistic AdaBoost formulation with Particle Filtering [23]. 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 dynamically weights between a Particle Filter and an AdaBoost Object Detector, and applies

their work to tracking hockey players. Instead of weighting, our approach directly applies AdaBoost into the Particle Filter's observation model. Lei et al. has a similar approach to us, and applies their work to tracking a face and a car. However, unlike Lei, we also describe how to apply Particle Filtering for tracking multiple objects simultaneously.

III. System Input

Our input is the sequence of images from a video stream. Let i be the frame number in this sequence. Each image consists of an pixel-array where each pixel has a red, green, and blue (RGB) component. Our system relies on a visual marker worn by an individual who wishes to have his or her face obscured.

IV. Assumptions

We use the visual marker's locations as a proxy for the location of the human head. Thus, we assume that the face's location will always be at a relative offset from the marker. Similarly, we assume the face's size will be a scaled size of the visual marker.

If a person's face is unobscured for a single frame, the person's identity will be known for many subsequent frames. While false positives make it impossible to see portions of the scene which the user may wish to observe, it does not reduce the privacy of those being viewed. Thus, we assume that false negatives are far less acceptable than false positives.

Our system makes a the following additional assumptions:

- Whenever a person's face is visible, then the visual marker worn by that person is visible
- All visible markers have a minimum number of visible, adjacent pixels
- There is a range of possible ratios between the height and width of a visible marker's bounding box
- The marker color is distinguishable from the background

V. System Output

The objective is to cover 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 output is a set of axis-aligned elliptical regions. These regions should completely cover all faces of people in the input image who are wearing markers. An elliptical region for ith output image is defined by a center-point, denoted by an x and y position, an x-axis aligned radius r_x and a y-axis aligned radius r_y:

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

The ith output video frame is the same as the ith input frame with the corresponding regions E_i obscured via a colored ellipse.

VI. Three Phases of System

Our solution consists of three phases: (A) learning a color-model for the marker with AdaBoost, (B) identifying the marker in a single image, and (C) using Particle Filtering to integrate temporal information for improved performance.

A. Offline Training of the Marker Classifier

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

1) Input: A human "supervisor" provides the AdaBoost algorithm with two sets of samples, 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 of marker pixels is

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

and the sample set of pixels that correspond background colors

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

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

2) Output: We use a Probabilistic AdaBoost formulation that produces a strong-classifier $H' : \{0, \ldots, 255\}^3 \mapsto [0, 1]$. This classifier predicts the probability that the RGB color of any pixel corresponds to the marker.

B. Run-Time Static Marker Detector

For static detection, each frame is processed independently.

1) Input: The Marker Detector uses as input the model generated from the AdaBoost classifier, as well as a single frame from the video stream.

2) Output: We can use the marker detector without tracking, to determine the location of faces. This would produce for the ith image, a region E_i as defined in Section V. However, if used with the marker tracker, this phase produces a set of rectangles bounding the corresponding markers. A bounded-rectangle on the ith image is defined by a center-point, denoted by an x and y position, a width Δx and a height Δy:

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

This rectangle is restricted by the assumptions described in Section IV.

C. Run-Time Dynamic Marker Tracker

The dynamic marker tracker uses temporal information to improve the Run-time Detector.

1) Input: The dynamic marker tracker uses both the classifier determined in the training phase and output from the static image recognition phase. Because we use Particle Filtering, we process a frame per iteration. 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 IX-A, but it requires three models as input:

a prior distribution, a transition model, and an observation model. We use the image detection system to initialize a Particle Filter for each newly-detected marker. We use the probabilistic classifier to determine the posterior distribution of a hat location for each Particle Filter, given all previously seen images.

2) Output: The output for the ith frame is also the region R_i as defined in Section V.

VII. 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 H', which assigns the probability that any pixel's color corresponds to the marker. Learning algorithms can use far less data than just determining the probability that each color corresponds to the visual marker. In our experiences, AdaBoost works well using a thousand labeled samples, while getting 10 samples for each color to generate the probability explicitly would require $10 \times 256^3 \approx 170$ million samples.

A. 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 $G = \{h : X \to Y\}$ be a set of weak hypotheses. AdaBoost's objective is to determine a function $H : X \mapsto Y$ by learning a linear function of elements from G that predicts Y given X. AdaBoost is an iterative algorithm where at each step, it integrates a new weak-hypothesis into the current strong-classifier.

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

$$\mathbb{E}(\phi(yf(x))) = \inf_f [\eta\phi(f(x)) + (1-\eta)\phi(-f(x))]$$

This is an approximation to the optimal Bayes Risk, minimizing $\mathbb{E}[l(f(X), Y)]$ with loss function

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

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.

In general, AdaBoost can use any weak-hypothesis with error less than 50%. However we use the greedy heuristic where at each iteration, we select a weak hypotheses that minimizes the number of incorrectly labeled data points [11].

1) Recasting Adaboost to Estimate Probabilities: Typically, as described in [31], AdaBoost predicts the most likely label that an input will have. If we let $\beta(x) = \sum_{t=1}^{T} \alpha_t h_t(X)$, then the typical strong classifier is binary and defined to be $H(x) = \text{sign}(\beta(x))$. Friedman et. al describes how to modify the AdaBoost algorithm to provide a probability instead [22]. The strong classifier determines that probability that an input corresponds to a label of 1 (as opposed to -1) is

$$H'(x) = \frac{e^{2\beta(x)}}{1 + e^{2\beta(x)}}$$

B. 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) [32] and LAB [33] 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 more 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 which bisect each of the 9 dimensions. 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 bisecting dimension d at a threshold j is described by:

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

We also include the complement of this hyperplane into our set of weak hypotheses $\overline{h_{d,j}}(X) = -h_{d,j}(X)$. We provide more classification flexibility to our simplistic weak classifiers by projecting the initial RGB space into the additional HSV and LAB spaces. For the weak learner, 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.

VIII. Run-Time Static Marker Detector

This section describes a 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) Cluster marker pixels using the connected component method. (3) Select all clusters that satisfy certain constraints to be locations of markers.

A. Clustering of pixels

To determine which pixels correspond to which markers, we apply the connected-component technique [34]. We iterate through all pixels that have been classified as markers, and assign the cluster for that pixel (as defined by connected-component) with a unique group-id. This yields a set of marker pixels for each visual marker in the frame.

To remove false positives, we verify there are at least c pixels in the cluster, and that ratio of width (Δx) to height (Δy) falls within a specified range from a to b: Formally

$$a \leq \frac{\Delta x}{\Delta y} \leq b$$

IX. Run-Time Dynamic Marker Tracker

We use Particle Filtering to incorporate temporal information into our models improving robustness to partial occlusions. As Particle Filtering requires probability distributions for how likely the state is given indirect observations, we describe a pixel-based Probabilistic AdaBoost formulation, which can be adapted for such purposes.

A. Review of SIR Particle Filtering

While there are many versions of Particle Filters, we use the Sampling Importance Resampling (SIR) Filter as described in [35], [36]. It is a non-parametric (sampling-based) method for performing state estimation of Dynamic Bayes Nets (DBNs) over discrete time. The state at iteration i is represented as a random variable χ_i with instantiation χ_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 $P(\chi_0)$, the transition model $P(\chi_i|\chi_{i-1})$, and the observation model $P(E_i|\chi_i)$. The prior describes the distribution of the object's state at the beginning of inference. The transition model describes, the distribution of the object's state at the next iteration, given the current state of the object. 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 according to the likelihood of all previous observations $P(\chi_i|E_{0:i})$. At each iteration, each particle is advanced according to a 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.

B. Marker Tracking

Particle Filtering uses three models: a prior distribution, transition model, and observation model.

1) Marker Model: The state of a marker is defined with respect to the image plane and is represented by an axis-aligned bounding box and a velocity. This results in a 6 tuple of the 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:

$$\chi = (x, y, \Delta x, \Delta y, \theta, s)$$

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

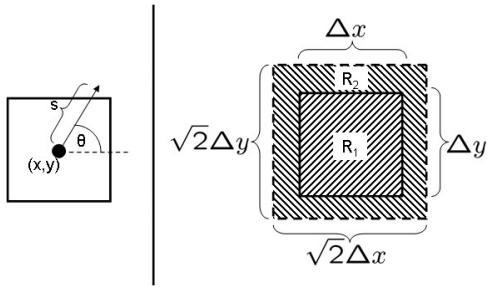

Fig. 2. Illustrates the state of a single bounding box (left) and the probability mask used for the Particle Filter's observation model (right).

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 $P(\chi_i|\chi_{i-1} = \chi_{i-1})$. 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:

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

At each iteration, we also enforce that the width and height constraints for each particle described in Section VIII-A. The sample from the gaussian (after being scaled by μ and σ) must be rescaled according to $\sqrt{t_i}$ in order to compensate for non-constant frame rates.

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 probability 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 $\sqrt{2}\Delta x \times \sqrt{2}\Delta y$. The $\sqrt{2}$ scaling factor makes the areas of R_1 and R_2 be equal. Let $P(I_i(u, v))$ be the probability that the pixel at u, v corresponds to a marker, as is provided by AdaBoost. Then:

$$R_1 = \left\{ (u,v) \;\middle|\; \begin{array}{l} x - \frac{\Delta x}{2} \leq u \leq x + \frac{\Delta x}{2}, \\ y - \frac{\Delta y}{2} \leq v \leq y + \frac{\Delta y}{2} \end{array} \right\}$$

$$R_2 = \left\{ (u,v) \;\middle|\; \begin{array}{l} x - \frac{\Delta x}{2} \leq \frac{u}{\sqrt{2}} \leq x + \frac{\Delta x}{2}, \\ y - \frac{\Delta y}{2} \leq \frac{v}{\sqrt{2}} \leq y + \frac{\Delta y}{2}, \\ (u,v) \notin R_1 \end{array} \right\}$$

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

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

Our final metric used as our observation model is:

$$P(\chi|E_t = e_t) = (1 - P_1)P_1 + P_1P_2$$

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.

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 [37], defined for two regions R_1 and R_2 is:

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

If a Marker Detection algorithm has an IOM of more than a specified overlap γ 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 2 π, and speed is randomly chosen between 0 and a maximum speed that is chosen a priori.

To handle disappearing markers, if the maximum probability affiliated with a filter is below γ_1 , then 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_2, we remove the filter that was created most recently. We remove the most recent to preserve the association between a marker and its corresponding Particle Filter for as long as possible.

X. 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.

Fig. 3. Sample image frame input on left image, with output regions overlayed on right image. This sample illustrates where the intense light induced a specularity, causing the classifier to lose track of the hat.

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 2x slower than the incoming frame rate.

For both setups, we trained on 2 one-minute video sequences using the method described in Section VI-A, exposing the system to many potential backgrounds, location and orientations of the visual markers, and over all lighting conditions that the experimental data experiences.

An image has a false negative if any part of any face is visible and has a false positive if there is a obscuring region that touches no face. These metrics are independent of the number of people in the scene. To evaluate the system, we place each frame into the category of correctly obscuring all faces, 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. For the tables, let FN be false negatives and FP be false positives.

A. 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 directly below (centered at the bottom-middle 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%

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

Fig. 4. 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.

Fig. 5. 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.

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. 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.

2) Crossing: 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 flexibility in our transition model, namely the biasses to the bounding-region controlled via $\mu_{\Delta x}$ and $\mu_{\Delta y}$. At the following frame in Figure 5, the system successfully segments what it believed to be a single hat in the previous frame into two two hats by creating a new Particle Filter.

B. Construction Site Experiments

The construction site data was collected from footage recorded at the CITRIS construction site at the University of Berkeley, California, under Human Subjects Protocol #2006-7-1. For the construction site, our marker is a green construction vest and we assume the face is directly above (centered at the top-middle 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 and the Static Marker Detector. All experiments were run on data acquired at 6 fps. This diminished 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. 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%

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. All 9 is superior in both reducing false positives and false negatives.

2) Particle Filtered Data: In this test, we evaluated performance between a non-Particle Filtered approach, where we just use each frame independently, and using Particle Filtering. We can see that the system dramatically reduces the number of false-positives, while inducing slightly more false-negatives. There were two extra false-negatives induced 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 portion of the face that is visible due to the partially cropped shirt is only 8 pixels wide.

XI. Conclusion and Future Work

We have presented the Respectful Cameras visual privacy system which tracks visual markers to robustly infer the location of individuals wishing to remain anonymous. We present 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.

In future work, we will experiment with different markers to identify preferred colors or patterns. It may be possible to build a Respectful Cameras method directly into the 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.

To obtain our experimental data, videos, or to get updates about this project, please visit: http://www.cs.berkeley.edu/∼jschiff/RespectfulCameras.

XII. Acknowledgements

Thanks to Ambuj Tewari for assisting in formulating the Probabilistic AdaBoost and Jen King for her help with

Fig. 6. Sample image frame input on left image, with output regions overlayed on right image. This sample illustrates how without Particle Filtering, 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, yielding a single, large ellipse.

relating this work to policy and law. Panasonic Inc. donated the cameras for our experiments.

REFERENCES

[1] M. McCahill and C. Norris, "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.

[2] M. Anderson, "Picture this: Aldermen caught on camera," *Chicago Sun-Times*, Jan. ,14 2006.

[3] NYCLU, "NYCLU report documents rapid proliferation of video surveillance cameras, calls for public oversight to prevent abuses," December 13, 2006. [Online]. Available: http://www.nyclu.org/whoswatching_pr_121306.html

[4] M. T. Moore, "Cities opening more video surveillance eyes," *USA Today*, July 18, 2005.

[5] "Trust: Team for research in ubiquitous secure technology." [Online]. Available: http://www.truststc.org/

[6] "Unblinking: New perspectives on visual privacy in the 21st century." [Online]. Available: http://www.law.berkeley.edu/institutes/bclt/events/unblinking/unblink.html

[7] R. Gavison, "Privacy and the limits of the law," *89 Yale L.J.*, pp. 421–471, 1980.

[8] H. F. Nissenbaum, "Privacy as contextual integrity," *Washington Law Review*, vol. 79, no. 1, 2004.

[9] R. Shaw, "Recognition markets and visual privacy," in *UnBlinking: New Perspectives on Visual Privacy in the 21st Century*, November 2006.

[10] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proc. of IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 1991, pp. 586–591.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. of IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 01, p. 511, 2001.

[12] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," *Proc. of IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 236–243, 2005.

[13] F. Dornaika and J. Ahlberg, "Fast and reliable active appearance model search for 3-d face tracking," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 34, no. 4, pp. 1838–1853, 2004.

[14] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proceedings of ECCV*, 2002, pp. 661–675.

[15] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," *iccv*, vol. 1, pp. 90–97, 2005.

[16] K. Okumna, A. Taleghani, N. de Freitas, J. Little, and D. Lowe, "A boosted particle filter: Multitarget detection and tracking." in *Proc. of Conf. European Conference on Computer Vision (ECCV)*, 2004.

[17] B. Wu and R. Nevatia, "Tracking of multiple, partially occluded humans based on static body part detection," *cvpr*, vol. 1, pp. 951–958, 2006.

[18] D.-S. Lee, "Effective gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, pp. 827– 832, May 2005.

[19] R. Feraud, O. J. Bernier, J.-E. Viallet, and M. Collobert, "A fast and accurate face detector based on neural networks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 42–53, 2001.

[20] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection," *cvpr*, vol. 00, p. 130, 1997.

[21] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[22] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," 1998. [Online]. Available: citeseer.ist.psu.edu/friedman98additive.html

[23] Y. Lei, X. Ding, and S. Wang, "Adaboost tracker embedded in adaptive particle filtering," in *Proceedings of International Conference on Pattern Recognition (ICPR)*, vol. 4, Aug. 2006, pp. 939–943.

[24] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, and T. Koehler, "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information," in *IEEE Proceedings of Intelligent Vehicles Symposium*, June 2005, pp. 255–260.

[25] S. Avidan, "Spatialboost: Adding spatial reasoning to adaboost." [Online]. Available: citeseer.ist.psu.edu/avidan06spatialboost.html

[26] D. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, vol. 20, 2003, pp. 91–110. [Online]. Available: citeseer.ist.psu.edu/lowe04distinctive.html

[27] X. Zhang, S. Fronz, and N. Navab, "Visual marker detection and decoding in ar systems: a comparative study," in *International Symposium on Mixed and Augmented Reality*, 2002, pp. 97–106.

[28] N. Kohtake, J. Rekimoto, and Y. Anzai, "InfoStick: An interaction device for inter-appliance computing," *Lecture Notes in Computer Science*, vol. 1707, pp. 246–258, 1999. [Online]. Available: citeseer.ist.psu.edu/kohtake99infostick.html

[29] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the American Society of Mechanical Engineers, Journal of Basic Engineering*, pp. 35–46, March 1960.

[30] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe, "A boosted particle filter: Multitarget detection and tracking," 2004. [Online]. Available: citeseer.ist.psu.edu/okuma04boosted.html

[31] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Computational Learing Theory*, pp. 80–91, 1998.

[32] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics Principles and Practice*. NY: AW, 1990.

[33] A. K. Jain, *Fundamentals of digital image processing*. Prentice Hall International, 1989.

[34] A. Rosenfeld, "Connectivity in digital pictures," *J. ACM*, vol. 17, no. 1, pp. 146–160, 1970.

[35] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[36] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Pearson Education, 1995.

[37] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.