PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces

Robert Templeman,[†‡] Mohammed Korayem,[†] David Crandall,[†] Apu Kapadia[†]

[†]School of Informatics and Computing
Indiana University Bloomington
{retemple, mkorayem, djcran, kapadia}@indiana.edu

[‡]Naval Surface Warfare Center, Crane Division
robert.templeman@navy.mil

***Abstract*—Cameras are now commonplace in our social and computing landscapes and embedded into consumer devices like smartphones and tablets. A new generation of wearable devices (such as Google Glass) will soon make 'first-person' cameras nearly ubiquitous, capturing vast amounts of imagery without deliberate human action. 'Lifelogging' devices and applications will record and share images from people's daily lives with their social networks. These devices that automatically capture images in the background raise serious privacy concerns, since they are likely to capture deeply private information. Users of these devices need ways to identify and prevent the sharing of sensitive images.**

As a first step, we introduce PlaceAvoider, a technique for owners of first-person cameras to 'blacklist' sensitive spaces (like bathrooms and bedrooms). PlaceAvoider recognizes images captured in these spaces and flags them for review before the images are made available to applications. PlaceAvoider performs novel image analysis using both fine-grained image features (like specific objects) and coarse-grained, scene-level features (like colors and textures) to classify where a photo was taken. PlaceAvoider combines these features in a probabilistic framework that jointly labels streams of images in order to improve accuracy. We test the technique on five realistic first-person image datasets and show it is robust to blurriness, motion, and occlusion.

I. Introduction

Cameras have become commonplace in consumer devices like laptops and mobile phones, and nascent wearable devices such as Google Glass,[1] Narrative Clip,[2] and Autographer[3] are poised to make them ubiquitous (Figure 1). These wearable devices allow applications to capture photos and other sensor data continuously (e.g., every 30 seconds on the Narrative Clip), recording a user's environment from a first-person perspective. Inspired by the Microsoft SenseCam project [24], these devices are also ushering in a new paradigm of 'lifelogging' applications that allow people to document their daily lives and share first-person camera footage with their social networks. Lifelogging cameras allow consumers to photograph unexpected moments that would otherwise have been missed, and enable safety and health applications like documenting law enforcement's interactions with the public and helping dementia patients to recall memories.

Fig. 1. Wearable camera devices. *Clockwise from top left:* Narrative Clip takes photos every 30 seconds; Autographer has a wide-angle camera and various sensors; Google Glass features a camera, heads-up display, and wireless connectivity. (Photos by Narrative, Gizmodo, and Google.)

However, with these innovative and promising applications come troubling privacy and legal risks [1]. First-person cameras are likely to capture deeply personal and sensitive information about both their owners and others in their environment. Even if a user were to disable the camera or to screen photos carefully before sharing them, malware could take and transmit photos surreptitiously; work on visual malware for smartphones has already demonstrated this threat [52]. As first-person devices become more popular and capture ever greater numbers of photos, people's privacy will be at even greater risk. At a collection interval of 30 seconds, the Narrative Clip can collect thousands of images per day — manually reviewing this bulk of imagery is clearly not feasible. Usable, fine-grained controls are needed to help people regulate how images are used by applications.

A potential solution to this problem is to create algorithms that automatically detect sensitive imagery and take appropriate action. For instance, trusted firmware on the devices

[1]Google Glass: http://www.google.com/glass/start/
[2]Narrative (formerly known as Memoto): http://getnarrative.com
[3]Autographer: http://autographer.com

Permission to freely reproduce all or part of this paper for noncommercial purposes is granted provided that copies bear this notice and the full citation on the first page. Reproduction for commercial purposes is strictly prohibited without the prior written consent of the Internet Society, the first-named author (for reproduction of an entire paper only), and the author's employer if the paper was prepared within the scope of employment.
NDSS '14, 23-26 February 2014, San Diego, CA, USA
Copyright 2014 Internet Society, ISBN 1-891562-35-5
http://dx.doi.org/doi-info-to-be-provided-later

bathroom personal office lab common area

Fig. 2. Sample first-person images from our datasets. Note the blur and poor composition, and the visual similarity of these four images despite being taken in spaces with very different levels of potential privacy risk.

could scan for private content and alert the user when an application is about to capture a potentially sensitive photo. Unfortunately, automatically determining whether a photo contains private information is difficult, due both to the computer vision challenges of scene recognition (especially in blurry and poorly composed first-person images), and the fact that deciding whether a photo is sensitive often requires subtle and context-specific reasoning (Figure 2).

Nevertheless, in this work we take an initial step towards this goal, studying whether computer vision algorithms can be combined with (minimal) human interaction to identify some classes of potentially sensitive images. In particular, we assume here that certain locations in a person's everyday space may be sensitive enough that they should generally not be photographed: for instance, a professor may want to record photos in classrooms and labs but avoid recording photos in the bathroom and in his or her office (due to sensitive student records), while at home the kitchen and living room might be harmless but bedroom photos should be suppressed.

In this paper we propose an approach called "PlaceAvoider", which allows owners of first-person cameras to *blacklist sensitive spaces*. We first ask users to photograph sensitive spaces (e.g., bathrooms, bedrooms, home offices), allowing our system to build visual models of rooms that should not be captured. PlaceAvoider then recognizes later images taken in these areas and flags them for further review by the user. PlaceAvoider can be invoked at the operating system level to provide warnings *before* photos are delivered to applications, thus thwarting visual malware and withholding sensitive photos from applications in general.

PlaceAvoider complements existing location services, using them to reduce the computational effort made when classifying images. For example, GPS can be used to identify the building in which the device is located, but it is typically not accurate enough to identify a specific room, because GPS signals are not reliably available indoors. Even if a reliable indoor location service existed, it would pinpoint where a camera is, not what it is looking at (e.g., when the camera is in a hallway, but capturing a nearby bathroom).

Research challenges. This work addresses several research challenges in order to make PlaceAvoider possible. First, we need an approach to recognize rooms using visual analysis with reasonable computational performance (either locally on the device or on a secure remote cloud). Second, many (or most) images taken from first-person cameras are blurry and poorly composed, where the already difficult problems of visual recognition are even more challenging. Third, rooms change appearance over time due to dynamic scenes (e.g., moving objects) as well as variations in illumination and occlusions from other people and objects. Finally, photos from 'other' spaces (i.e., spaces that are not blacklisted) may form a large fraction of images, and false positives must be kept low to reduce the burden on the owner.

Our Contributions. Our specific contributions are:

1) **Presenting PlaceAvoider**, a framework that identifies images taken in sensitive areas to enable fine-grained permissions on camera resources and photo files;
2) **Recognizing images of a space** by using a novel combination of computer vision techniques to look for distinctive 'visual landmarks' of the enrolled spaces and global features of the room such as color patterns;
3) **Analyzing photo streams** to improve the accuracy of indoor place recognition by labeling sequences of images jointly, using (weak) temporal constraints on human motion in a probabilistic framework;
4) **Implementing and evaluating PlaceAvoider** using first-person images from five different environments, showing that photos from sensitive spaces can be found with high probability even in the presence of occlusion or images taken from non-enrolled spaces.

The remainder of the paper describes these contributions in detail. Section II describes our architecture, constraints, and concept of operation, while Section III describes our image classification techniques. Section IV reports our evaluation on several first-person datasets. We discuss the implications of our results in Section V before surveying related work in Section VI and concluding in Section VII.

II. Our Approach

Our goal is a system that allows users to define context-based fine-grained policies to control the sharing of their images from smartphones and first-person cameras. We start by describing our privacy goals.

A. Privacy goals and adversary model

The increasing presence of cameras in electronic devices means that cameras are now more likely to enter sensitive spaces, where the cost of image leaks may be high. Our work aims to protect the privacy of users in two ways.

First, we assume that users will want to share some of their first-person photos with social and professional contacts but will need help managing and filtering the huge collections of images that their devices collect. Their social contacts are not 'adversaries' in the traditional sense (where attackers actively try to obtain sensitive photos), but inadvertent sharing of certain images can nevertheless cause embarrassment (e.g., photos with nudity) and have social or professional consequences. Thus it is important to help users identify potentially sensitive images before they are shared.

Second, malicious applications (such as Trojan applications) that have access to a device's camera may seek to

actively capture sensitive images in the background. For example, visual malware such as PlaceRaider [52] may be used to surveil sensitive spaces like offices or to blackmail victims by capturing nude photographs in their bedroom. We assume such applications have been installed (either unwittingly or as a Trojan application) with the requisite permissions for the camera and Internet access, but that the operating system has not been compromised.

B. System model

We consider a model in which sensitive photos are identified by analyzing the image content in conjunction with contextual information such as GPS location and time, i.e., *where* and *when* the photo was taken. To make image analysis for privacy leaks tractable, we focus on fine-grained control of images based on the *physical spaces* captured within the images. Our approach could withhold sensitive images from applications until they are reviewed by the owner of the camera, or it could tag images with metadata to be used by trusted (e.g., lifelogging) applications to assist the owner in analyzing their image collections.

Our proposed system has three elements: a *privacy policy* to indicate private spaces, an *image classifier* to flag sensitive images, and a *policy enforcement mechanism* to determine how sensitive images are handled by the receiving applications. For instance, Figure 3 illustrates how PlaceAvoider allows fine-grained control of a camera based on context-based policy. We now briefly describe these three components:

- ***Privacy policy.*** In this work, a policy is a set of blacklisted spaces — we use the term *blacklisted* generally to refer to any space that we want to label (i.e., a blacklisted space can vary with respect to its sensitivity). Each space in the policy includes a geospatial location (e.g., latitude and longitude), enrollment images or a visual model of the space, a string identifier (e.g., 'bathroom'), and the action to be taken by PlaceAvoider (e.g., which application(s) can access the image). In addition, a sensitivity value can be given to trade-off between conservative and liberal blacklisting when the image analysis is not very certain.
- ***Image classifier.*** The image classifier builds models of enrolled spaces, and then classifies new images according to where they were taken. The classifier must deal with significant image noise, including motion blur, poor composition, and occlusions (caused by people and objects added to a space). The classifier can process individual images, or jointly process image sequences in order to improve accuracy. As illustrated in Figure 3, this classification step could be outsourced to an *off-board image classifier* such as a cloud service (akin to cloud-based speech-to-text translation offered by Android[4] and Apple iOS[5]). We discuss trade-offs between on- and off-board processing in Section IV-E.
- ***Policy enforcement.*** We assume two possible policy enforcement mechanisms. User policies can specify that sensitive photos must be blocked from applications, in which case users can review these photos before they are delivered to the application, or users can allow access to trusted applications that make use of metadata supplied by the image classifier. The policy enforcement mechanism delivers photos accordingly, either to the reviewing interface or to the trusted applications.

We anticipate two types of scenarios that PlaceAvoider must handle. The first scenario is when the user can practically enroll all possible spaces in the structure, like in a home with a dozen rooms. We call these *closed locales*; for these places, our classifier can assign each photo into one of these n rooms using an n-way classifier. The second scenario is for *open locales* — buildings with a large number of spaces for which it is not feasible to enroll every space. This is a more challenging case in which we also need to identify photos taken in none of the n classes. We evaluate PlaceAvoider under both scenarios in Section IV.

C. Usage scenario

PlaceAvoider addresses the following usage scenario. Mary wears a sensor-enabled lifelogging device so that she can record her activities throughout the day and capture moments that would otherwise be hard to photograph (like interactions with her infant). However, she is concerned about the camera taking photos in sensitive areas. She decides to set a PlaceAvoider policy. She has five rooms in her apartment and enrolls them by taking pictures of each space as prompted by PlaceAvoider. She asserts that she does not want photos taken in her bathroom or bedroom. She sets a similar policy at work. She spends most of her time in her office, a lab, and a conference room. She enrolls these spaces, deeming the lab a sensitive room.

Soon afterwards she is working in the lab and receives an alert on her smartphone indicating that an application is attempting to take a photo in a sensitive space. She confirms the alert, wondering why her exercise-monitoring app is attempting to take surreptitious photos and decides to uninstall the app. Later that evening, she downloads the photos from her lifelogging camera. The PlaceAvoider system neatly organizes her photos temporally and spatially, flagging the images that were taken in sensitive spaces.

III. Image classification

Having described our system architecture, adversarial model, and usage scenario, we now turn to the challenge of automatically recognizing where a photo was taken within an indoor space based on its visual content. As described above, we assume that GPS has provided a coarse position, so our goal here is to classify image content amongst a relatively small number of possible rooms within a known structure. While there is much work on scene and place recognition [56], [37], we are not aware of work that has considered fine-grained indoor localization in images from first-person devices.

We first consider how to classify single images, using two complementary recognition techniques. We then show how to improve results by jointly classifying image sequences, taking

[4]Voice Search: http://www.google.com/insidesearch/features/voicesearch/
[5]Siri: http://www.apple.com/ios/siri/

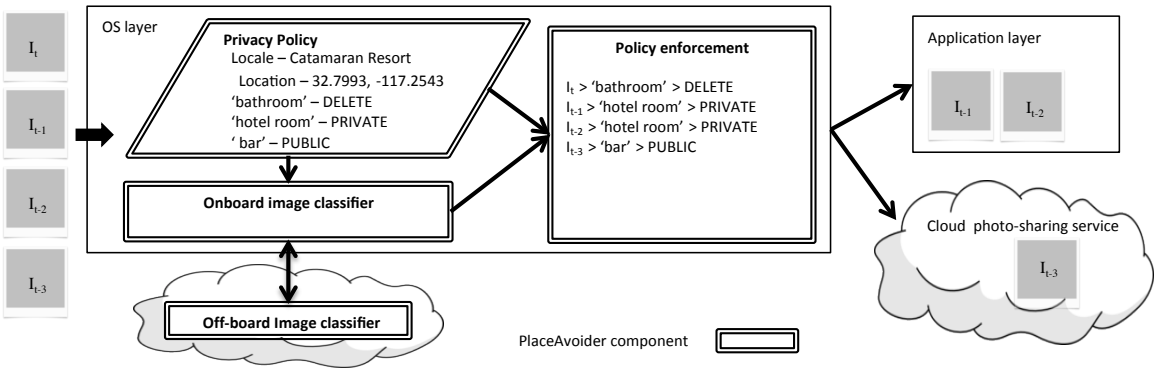
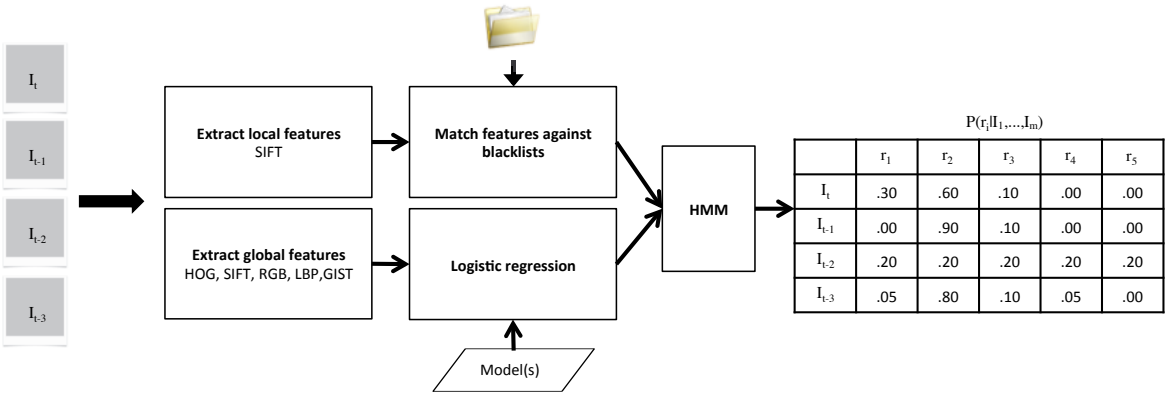


Fig. 3. An abstract depiction of PlaceAvoider enforcing a fine-grained camera privacy policy. Our model leverages cloud computation to perform compute-intensive tasks. Cloud-based implementations of PlaceAvoider could also enforce privacy preferences for photo sharing sites.



	r_1	r_2	r_3	r_4	r_5
I_t	.30	.60	.10	.00	.00
I_{t-1}	.00	.90	.10	.00	.00
I_{t-2}	.20	.20	.20	.20	.20
I_{t-3}	.05	.80	.10	.05	.00

Fig. 4. The PlaceAvoider classifier works on streams of images extracting local and global features. Single image classification feeds the HMM which outputs room labels and marginal distributions.

advantage of temporal constraints on human motion. Figure 4 depicts the classifier architecture used in PlaceAvoider.

A. Classifying individual images

We employ two complementary methods for classifying images. The first is based on a concept from computer vision called 'local invariant features', in which distinctive image points (like corners) are detected and encoded as high-dimensional vectors that are insensitive to image transformations (changes in illumination, viewpoint, zoom, etc.). The second approach relies on global, scene-level image features, capturing broad color and texture patterns. These approaches are complementary: local features work well for images with distinctive objects, but fail when images are blurry or more generic, while global features model the overall 'look' of a room but are less useful for close-ups of individual objects.

We take machine-learning approaches to both the local- and global-level techniques. We thus require training data in the form of images taken in each class (each room of the building); this training data is produced during the enrollment phase, when PlaceAvoider prompts the user to take images that cover the space of interest. Unlike most prior work (Section VI), our training images do not require rigid organization, extensive interaction, or specialized equipment.

More formally, we assume that we have a small set $\mathcal{R} = \{r_1, ..., r_n\}$ of possible locations (kitchen, living room, etc.), and for each room r_i we have a set $\mathcal{I}_i$ of training images. Given a new image I, our goal is to assign it one of the labels in $\mathcal{R}$.

Local features. Our local feature classifier represents each enrolled space as a collection of distinctive local feature descriptors that are invariant to variations like pose and illumination changes. We use the Scale Invariant Feature Transform (SIFT) [38] to produce these features. Briefly summarized, SIFT finds corners and other points in an image that are likely to withstand image transformations and analyzes the distribution of gradient orientations within small neighborhoods of each corner. It identifies a local orientation and scale and then encodes the gradient distributions as a 128-dimensional invariant descriptor vector for each point.

To build a model of room $r_i \in \mathcal{R}$, we extract SIFT features for each training image, producing a set of 128-dimensional vectors for each room (where each image contributes hundreds or thousands of vectors depending on its content). The individual feature lists are concatenated into one list, ignoring the spatial position of the feature, yielding a set M_i for each room r_i.

To classify a test image I, we again find SIFT features. Our task now is to compare this set of SIFT features to each model M_i, finding the one that is most similar. We could simply

count the number of 'matching' points in each set (for some definition of "matching"), but this yields poor results, because many image features exist in multiple rooms of a house. For instance, consistent architectural or design elements may reside throughout a home, or similar objects may exist throughout the offices of a building. We thus match images to models based on the number of *distinctive* local features that they have in common.

In particular, we define a scoring function S that evaluates a similarity between a test image I and a given set of SIFT features M_i corresponding to the model of room r_i,

$$S(I, r_i) = \sum_{s \in I} \mathbb{1}\left(\frac{\min_{s' \in M_i} ||s - s'||}{\min_{s' \in M_{-i}} ||s - s'||} < \tau\right), \quad (1)$$

where M_{-i} is the set of features in all rooms except r_i, i.e. $M_{-i} = \cup_{r_j \in \mathcal{R} - \{r_i\}} M_j$, $\mathbb{1}(\cdot)$ is an indicator function that is 1 if its parameter is true and 0 otherwise, $||\cdot||$ denotes the L2 (Euclidean) vector norm, and τ is a threshold. Intuitively, given a feature in a test image, this scoring function finds the distance to the closest feature in a given model, as well as the distance to the closest feature in the *other* models, and counts it only if the former is significantly smaller than the latter. Thus this technique ignores common features, counting only those that are distinctive to a particular room. The minimizations in Equation (1) can be computationally intensive for large sets since the vectors have high dimensionality. We consider how to do them more efficiently in Section IV.

To perform classification for image I, we simply choose the room with the highest score, $\arg\max_{r_i \in \mathcal{R}} S(I, r_i)$, although we consider an alternative probabilistic interpretation in Section III-B.

Global features. Unfortunately, many first-person images do not have many distinctive features (e.g., blurry photos, photos of walls, etc.), causing local feature matching to fail since there are few features to match. We thus also use global, scene-level features that try to learn the general properties of a room, like its color and texture patterns. These features can give meaningful hypotheses even for blurry and otherwise relatively featureless images. Instead of predefining a single global feature type, we instead compute a variety of features of different types and with different trade-offs, and let the machine learning algorithm decide which of them are valuable for a given classification task. In particular, we use:

1) *RGB color histogram,* a simple 256-bin histogram of intensities over each of the three RGB color channels, which yields a 768-dimensional feature vector. This is a very simple feature that simply measures the overall color distribution of an image.
2) *Color-informed Local Binary Pattern (LBP),* which converts each 9×9 pixel neighborhood into an 8-bit binary number by thresholding the 8 outer pixels by the value at the center. We build a 256-bin histogram over these LBP values, both on the grayscale image and on each RGB channel, to produce a 1024-dimensional vector [30]. This feature produces a simple representation of an image's overall texture patterns.
3) *GIST,* which captures the coarse texture and layout of a scene by applying a Gabor filter bank and spatially down-sampling the resulting responses [41], [13]. Our variant produces a 1536-dimensional feature vector.
4) *Bags of SIFT,* which vector-quantize SIFT features from the image into one of 2000 'visual words' (selected by running k-means on a training dataset). Each image is represented as a single 2000-dimensional histogram over this visual vocabulary [56], [37]. This feature characterizes an image in terms of its most distinctive points (like corners).
5) *Dense bags of SIFT* are similar, except that they are extracted on a dense grid instead of at corner points and the SIFT features are extracted on each HSV color plane and then combined into 384-dimensional descriptors. We encode weak spatial configuration information by computing histograms (with a 300-word vocabulary) within coarse buckets at three spatial resolutions (1×1, 2×2, and 4×4 grid, for a total of $1 + 4 + 16 = 21$ histograms) yielding a 300 $\times$ 21 = 6,300-dimensional vector [56]. This feature characterizes an image in terms of both the presence and spatial location of distinctive points in the image.
6) *Bags of HOG* computes Histograms of Oriented Gradients (HOG) [11] at each position of a dense grid, vector-quantizes into a vocabulary of 300 words, and computes histograms at the same spatial resolutions as with dense SIFT, yielding a 6,300-dimensional vector. HOG features capture the orientation distribution of gradients in local neighborhoods across the image.

Once we extract features from labeled enrollment images, we learn classifiers using the LibLinear L2-regularized logistic regression technique [17].

B. Classifying photo streams

The first-person camera devices that we consider here often take pictures at regular intervals, producing temporally ordered streams of photos. These sequences provide valuable contextual information because of constraints on human motion: if image I_i is taken in a given room, it is likely that I_{i+1} is also taken in that room. We thus developed an approach to jointly label sequences of photos in order to use temporal features as (weak) evidence in the classification.

We use a probabilistic framework to combine this evidence in a principled way. Given a set of photos $I_1, I_2, ..., I_m$ ordered with increasing timestamp and taken at roughly regular intervals, we want to infer a room label $l_i \in \mathcal{R}$ for each image I_i. By Bayes' Law, the probability of a given image sequence having a given label sequence is,

$$P(l_1, ..., l_m | I_1, ..., I_m) \propto P(I_1, ..., I_m | l_1, ..., l_m) P(l_1, ..., l_m),$$

where we ignore the denominator of Bayes' Law because the sequence is fixed (given to us by the camera). If we assume that the visual appearance of an image is conditionally independent from the appearance of other images given its room label, and if we assume that the prior distribution over room label depends only on the label of the preceding image (the Markov assumption), we can rewrite this probability as,

$$P(l_1 ... l_m | I_1 ... I_m) \propto P(l_0) \prod_{i=2}^{m} P(l_i | l_{i-1}) \prod_{i=1}^{m} P(I_i | l_i). \quad (2)$$

The first factor $P(l_1)$ is the prior probability of the first room label. We assume here that this is a uniform distribution, and thus it is ignored. The second factor models the probability of a given sequence of room labels and should capture the fact that humans are much more likely to stay in a room for several frames than to jump randomly from one room to the next. In this paper we use a very simple prior model,

$$P(l_i|l_{i-1}) = \begin{cases} \alpha, & \text{if } l_i \neq l_{i-1}, \\ 1-(n-1)\alpha, & \text{otherwise,} \end{cases}$$

where n is the number of classes (rooms) and α is a small constant (we use 0.01). Intuitively, this means that transitions from one room to another have much lower probability than staying in the same room. This prior model could be strengthened depending on contextual information about a place — e.g. it may be impossible to travel from the kitchen to the bedroom without passing through the living room first — but we do not consider that possibility in this paper.

The third factor in Equation (2) models the likelihood that a given image was taken in a given room. Intuitively, these likelihoods are produced by the local and global classifiers described in Section III-A, but we need to 'convert' their outputs into probabilities. Again from Bayes' Law,

$$P(I_i|l_i) = \frac{P(l_i|I_i)P(I_i)}{P(l_i)}.$$

We again ignore $P(I_i)$ (since I_i is observed and hence constant) and assume that the prior over rooms $P(l_i)$ is a uniform distribution, so it is sufficient to model $P(l_i|I_i)$. For the global classifiers, we use LibLinear's routines for producing a probability distribution $P_G(l_i|I_i)$ from the output of a multi-class classifier based on the relative distances to the class-separating hyperplanes [17]. For the local features, we introduce a simple probabilistic model. Equation (1) defined a score $S(I, r_i)$ between a given image I and a room r_i, in particular counting the number of distinctive image features in r_i that match I. This matching process is, of course, not perfect; the score will occasionally count a feature point as matching a room when it really does not. Suppose that the probability that any given feature match is correct is β, and is independent of the other features in the image. Now the probability that an image was taken in a room according to the local feature scores follows a binomial distribution,

$$P_L(l_i|I_i) \quad \propto \quad \binom{N}{S(I,l_i)} \beta^{S(I,l_i)}(1-\beta)^{N-S(I,l_i)}$$

where N is the total number of matches across all classes,

$$N = \sum_{r_i \in \mathcal{R}} S(I, r_i).$$

We set $\beta = 0.9$ in this paper; the system is not very sensitive to this parameter unless it is set close to 0.5 (implying that correct matches are no more likely than chance) or to 1 (indicating that matching is perfect).

To produce the final probability $P(I_i|l_i)$, we multiply together $P_L(I_i|l_i)$ and $P_G(I_i|l_i)$, treating local and global features as if they were independent evidence.

The model in Equation (2) is a Hidden Markov Model (HMM), and fast linear-time algorithms exist to perform inference. In this paper we use the HMM to perform two different types of inference, depending on the application (as described in Section IV). We may wish to find the most likely room label l_i^* for each image I_i given all evidence from the entire image sequence,

$$l_1^*, ..., l_m^* = \arg \max_{l_1,...,l_m} P(l_1, ..., l_m^*|I_1, ..., I_m),$$

which can be solved efficiently using the Viterbi algorithm [29]. In other applications, we may wish to compute the marginal distribution $P(l_i|I_1, ..., I_m)$ — i.e., the probability that a given image has a given label, based on all evidence from the entire image sequence — which can be found using the forward-backward algorithm [29]. The latter approach gives a measure of confidence; a peaky marginal distribution indicates that the classifiers and HMM are confident, while a flat distribution reflects greater uncertainty.

IV. Evaluation

We conducted several experiments to measure the accuracy and performance of PlaceAvoider on a variety of datasets and scenarios. We first describe our first-person image datasets (Section IV-A) and then evaluate the performance of local and global classifiers on single images (Section IV-B) before evaluating the combined features and joint stream classification in Section IV-C. We evaluate the accuracy in a retrieval setting (Section IV-D) and report computational performance in Section IV-E.

A. Evaluation datasets

We are not aware of any existing dataset of first-person imagery suitable for our study, so we collected five new datasets in a variety of indoor spaces. For each dataset, we first collected enrollment (training) photos that were deliberately taken by a human who tried to take a sufficient number of photos to cover each room. For each dataset, we took between three and five rounds of enrollment images at different times of the day to capture some temporal variation (e.g., changes in illumination and in the scene itself). The number of enrollment images per space (the sum over all rounds) varied from 37 to 147, depending on the size of room and the user.

Collecting these images is simple and only took a few minutes. We then collected stream (test) datasets in which the person wore a first-person camera as they moved around the building. Because Google Glass and other devices were not yet commercially available, we simulated them with a smartphone worn on a lanyard around the person's neck. These smartphones ran an app that took photos at a fixed interval (approximately three seconds), and collection durations ranged from about 15 minutes to one hour.

Our datasets consisted of three home and two workplace environments, each with five rooms (classes):

- ***House 1,*** a well-organized family home with three bedrooms, bathroom, and study;
- ***House 2,*** a sparsely-furnished single person's home, with garage, bedroom, office, bathroom, and living room;

- ***House 3,*** a somewhat more cluttered family home with two bedrooms, a living room, kitchen, and garage;
- ***Workplace 1,*** a modern university building with common area, conference room, bathroom, lab, and kitchen;
- ***Workplace 2,*** an older university building with a common area, conference room, bathroom, lab, and office.

The datasets were collected independently by four of the authors.[6] The authors simulated various daily chores during the stream collection, with the aim of obtaining realistic coverage across various rooms. For example, in Workplace 2 the author obtained a cup of coffee, picked up printed material, spoke with an administrative assistant, and visited the conference room and common areas. In House 1, the author simulated activities like visiting the bathroom, working in the study, reading, and organizing. In House 2, the author performed household chores with a high degree of movement, including cleaning, folding and organizing clothes, moving objects from room to room, etc. Table I presents detailed statistics on the datasets.

B. Single image classification

Local features. We begin by evaluating the classifier based on local features described in Section III-A. In addition to presenting raw classification accuracy statistics, we also test the effect of various parameters on the accuracy of this approach. To do this without overfitting to our test dataset, all results in this section use the enrollment photos for both training and testing, using a cross-validation approach. In particular, if a dataset has r rounds of enrollment photos, we train r classifiers, in each case using $r-1$ rounds as training images and the other round as the test images, and then averaging the accuracies together. This methodology simulates a closed locale, as defined in Section II-B, where each photo is known to have been taken in one of the enrolled spaces and the task is to classify amongst them. We discuss the evaluation of open locales in Section IV-D.

Table II presents results of n-way classification for each of the five datasets (where $n = 5$ since there are five rooms in each dataset). The classification accuracies range across the datasets, from a high of 98.4% accuracy for House 1 to 76.2% for House 2. This is not surprising, given that House 2 is sparsely decorated with relatively few feature points for the local classifier to use. We compare these results to a baseline that simply chooses the largest class; even for House 2, our classifier beats this baseline by over 2.5 times.

For images with few interest-point descriptors, like blurry photos or photos of walls and other textureless surfaces, the local classifier has little information with which to make a decision. Table II shows the average number of distinctive features per image across the three datasets. When there are no features to match, or multiple rooms have the same (small) number of feature matches, the classifier resorts to a random guess amongst these rooms. The table shows the number of images for which this happened, as well as the number of images for which there were no matches at all (so that the classifier resorted to 5-way random guessing).

The local feature classifier requires a threshold τ to determine whether a feature match is distinctive (Equation (1) in Section III-A). Intuitively, the larger the value of this threshold, the more feature points are considered during matching, but these points are less distinctive; for smaller values the matched feature points are much more accurate, but eventually become so few that there are many ties and most of the classifier's decisions are random guesses. We empirically found minimal sensitivity for τ between 0.3–0.6. For the experiments in this paper we select a value in the middle of this range, $\tau = 0.45$.

To test the effect of image resolution on accuracy of the local classifier, Table II also presents correct classification rates on images sub-sampled to 1 MegaPixel (MP). This sub-sampling also has the effect of decreasing the number of detected SIFT feature points, since SIFT uses heuristics based on image size to determine how many points to produce. Surprisingly, performance on the lower-resolution images either equals or beats that of the high-resolution image on all five datasets. This suggests that the limiting factor on performance is not image resolution but perhaps image quality; all of our images were taken indoors without a flash and include significant blur and sensor noise. Decreasing image resolution to 1MP thus does not decrease performance and in fact may help to reduce noise.

Global features. As we discussed in Section II, a problem with the local classifier is that it fails on images with few distinctive points, because there are few feature matches and the classifier must resort to random guessing. Our global features are designed to address this problem by building models of general scene-level characteristics instead of local-level features. Table III compares classification performance of our six global features, using the same evaluation criteria as with the local features — 5-way classification using cross validation on the enrollment set. For the datasets with relatively few features, like the sparsely-decorated House 2, the best global features outperform the local features (78.8% vs. 76.2% for House 2, and 93.9% vs. 84.0% for Workspace 1), but for the other sets the local features still dominate. In the next section we combine these features together with temporal reasoning in order to improve accuracy. Since the two bags-of-SIFT and the bags-of-HOG features outperform the other global techniques by a significant margin for most datasets, we elected to use only these three in PlaceAvoider.

C. Temporal stream classification

We next evaluate the probabilistic joint image stream labeling technique proposed in Section III-B. For this experiment, we used all of the enrollment photos for training and used the photo streams for testing. We performed inference on the Hidden Markov Model (HMM) by using the Viterbi algorithm to find the most likely sequence of states given evidence from the entire image stream.

Table IV shows the results of this step. When classifying single images, the global and local classifiers perform roughly the same, except for the sparsely-decorated House 2 where global features outperform local features by almost eight

[6]The authors collected these datasets to avoid the difficulties associated with enlisting participants to photograph their own sensitive spaces. We plan to run user studies with recruited participants in the future.

TABLE I. SUMMARY OF OUR DATASETS. ALL DATASETS HAVE FIVE ROOMS (CLASSES). MAJORITY-CLASS BASELINES ARE SHOWN. FOR HOUSE 3, THREE ROUNDS WERE TAKEN WITH AN HTC AMAZE PHONE, ONE WITH A DIGITAL SLR CAMERA, AND ONE WITH A SAMSUNG GT-S5360L PHONE.

	Enrollment photos						Test photo streams			
Dataset	Device	Native resolution	# of images	# of rounds	Mean images/room	Baseline accuracy	Device	Native resolution	# of images	Baseline accuracy
House 1	iPhone 4S	8MP	184	3	61	22.8%	iPhone 4S	8MP	323	29.8%
House 2	iPhone 5	8MP	248	3	83	29.9%	iPhone 5	8MP	629	31.0%
House 3	(see caption)	3-6MP	255	5	85	30.2%	HTC Amaze	6MP	464	20.9%
Workplace 1	Motorola EVO	5MP	733	3	244	24.4%	HTC Amaze	6MP	511	32.1%
Workplace 2	HTC Amaze	6MP	323	5	108	25.4%	HTC Amaze	6MP	457	28.9%

TABLE II. LOCAL FEATURE CLASSIFIER TRAINED AND TESTED ON ENROLLMENT IMAGES USING CROSS-VALIDATION.

		Native-sized images				Downsampled images (1MP)			
Dataset	Baseline accuracy	Classification accuracy	Mean # of features	# of images with ties	# of images with 5-way tie	Classification accuracy	Mean # of features	# of images with ties	# of images with 5-way tie
House 1	22.8%	98.4%	297	2	0	**98.4%**	249	0	0
House 2	29.9%	76.2%	209	27	8	**77.4%**	66	50	21
House 3	30.2%	95.7%	59	12	5	**96.9%**	352	2	0
Workplace 1	24.4%	84.0%	33	115	45	**86.8%**	31	133	52
Workplace 2	25.4%	92.9%	104	15	6	**93.5%**	44	39	17
Average	26.5%	89.4%	—	—	—	**90.6%**	—	—	—

TABLE III. GLOBAL FEATURE CLASSIFIER TRAINED AND TESTED ON ENROLLMENT IMAGES USING CROSS-VALIDATION.

Dataset	Baseline accuracy	Bags of SIFT	Dense bags of SIFT	Bags of HOG	LBP	GIST	RGB histogram
House 1	22.8%	**89.1%**	81.4%	82.7%	41.6%	71.9%	57.4%
House 2	29.9%	49.7%	**78.8%**	78.7%	52.8%	64.8%	47.9%
House 3	32%	**89.4%**	68.9%	66.2%	51.9%	65.5%	57.4%
Workplace 1	24.4%	83.2%	**93.9%**	88.8%	76.2%	85.1%	79.8%
Workplace 2	25.4%	73.8%	83.1%	**83.2%**	67.5%	72.2%	55.0%
Average	26.5%	77.0%	**81.2%**	79.9%	58.0%	71.9%	59.5%

percentage points. On average, the classifiers outperform a majority baseline classifier by almost 2.5 times. The HMM provides a further and relatively dramatic accuracy improvement, improving average accuracy from 64.7% to 81.9% for local features, and from 64.3% to 74.8% for global features. Combining the two types of features together with the HMM yields the best performance with an average accuracy of 89.8%, or over 3.1 times the baseline.

Figure 5 shows some sample images from the House 2 stream, including a random assortment of correctly and incorrectly classified images. We can speculate on the cause of some of the misclassifications. When images are collected looking through windows or doors such that little of an enrolled space is captured in the image, the classifier confidence is intuitively reduced (see panels 1, 4, and 7 of the misclassified examples in Figure 5). Similarly, high degrees of occlusion in images will frustrate classification attempts (panel 3 of the misclassified examples demonstrates this).

Human interaction. An advantage of our probabilistic approach is that it can naturally incorporate additional evidence, if available. For instance, a lifelogging application or the device operating system could ask the user to help label ambiguous images. We simulated a simple version of this approach by having the HMM identify the least confident of its estimated labels (i.e., the image with the lowest maximum marginal probability). We then forced that image to take on the true label by modifying $P(l_i|I)$ in Equation (2) to be 1 for the correct label and 0 for the incorrect labels, and reran inference. We repeated this process 10 times, simulating PlaceAvoider asking the user to label 10 images. The last column of Table IV presents the results, showing a further increase in performance over the fully automatic algorithm, achieving over 90% accuracy for four of the datasets, and 95–100% accuracy for three of them. An additional enhancement would be to update the visual models themselves based on these new labeled images, but we leave this for future work.

Online inference. Note that our HMM assumes that the entire photo stream is available — i.e., in labeling a given image, the classifier can see images in the future as well as in the past. This scenario is reasonable for photo-sharing, lifelogging, and other applications that are tolerant to delay. For applications that require online, real-time decisions, the HMM can be modified to look only at the past (by running only the forward pass of the Forward-Backward Algorithm), but at a reduced accuracy: average HMM performance across the five datasets falls from 89.8% to 82.6% in this case.

Impact of scene occlusion. First-person cameras capture dynamic scenes with moving objects and people, and this often causes large portions of a scene to be occluded by foreground subjects in the photos. These occlusions increase the difficulty of indoor place recognition, but we expect them to be commonplace — in fact, potential occlusions may be the basis for defining a room as *sensitive* in a privacy policy. (For

TABLE IV. CLASSIFICATION OF TEST STREAMS BY THE SINGLE IMAGE CLASSIFIERS AND VARIATIONS OF THE HMM.

		Single image classifier		Joint stream classifier			
Dataset	Baseline accuracy	Local features	Global features	Local features	Global features	Local+global features	Local+global+ human interaction
House 1	29.8%	**52.9%**	48.3%	**89.2%**	64.0%	**89.2%**	**95.0%**
House 2	31.0%	41.8%	**49.1%**	55.0%	56.4%	**74.6%**	**76.8%**
House 3	20.9%	**81.5%**	80.0%	97.4%	86.9%	**98.7%**	**99.8%**
Workplace 1	32.1%	**75.9%**	74.6%	75.5%	**89.2%**	87.7%	**91.0%**
Workplace 2	28.9%	**71.6%**	69.4%	92.3%	81.2%	**98.7%**	**100.0%**
Average	28.5%	64.7%	64.3%	81.9%	74.8%	**89.8%**	**92.5%**

Fig. 5. Some sample classification results from the House 2 stream, showing correctly classified (top) and incorrectly classified (bottom) images.

TABLE V. EFFECT OF IMAGE OCCLUSION ON CLASSIFICATION ACCURACY, ON OUR SYNTHETIC DATASET BASED ON WORKSPACE 2.

	Single image		Photo streams	
% of occluded images	Local classifier	Global classifier	Local+ global	Local+global+ interaction
0	71.6%	69.4%	98.7%	100.0%
10	67.6%	68.7%	98.9%	100.0%
20	67.2%	68.3%	99.6%	100.0%
30	64.6%	69.8%	99.8%	100.0%
100	68.0%	69.8%	98.5%	100.0%

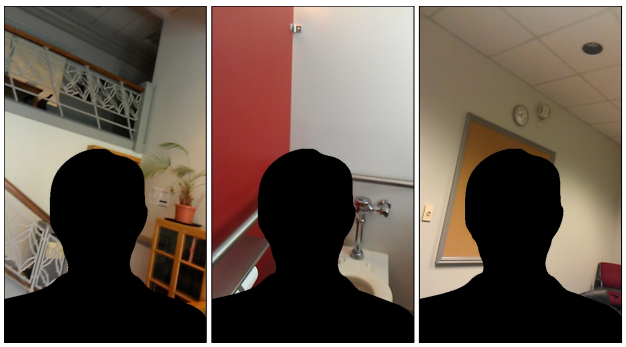


Fig. 6. Sample images from our dataset with synthetic occlusions.

instance, empty bathrooms are usually innocuous but photos of people in the bathroom may cause concern.)

While our streams include some incidental occlusions, we wanted to measure the effect that more frequent occlusions would have on classifier accuracy. To do this, we generated a dataset with simulated occlusions, superimposing a human silhouette (which blocked about 30% of the image) on varying fractions of the images. Figure 6 shows examples of our simulated images, and Table V presents accuracies on these images on the Workspace 2 dataset. (We chose this dataset because it had relatively high performance with both types of individual features and the stream classifier.) We observe that local feature performance declines as more images are occluded, while the accuracies of the global features and HMM are relatively stable, decreasing by less than a percentage point. We save more extensive investigations of occlusion by real objects and people for future work.

D. Retrieving private images

The experiments so far have cast our problem as one of image classification: given an image known to have been taken in one of n rooms, identify the correct room. The main goal of PlaceAvoider, however, is not necessarily to identify the exact room, but to filter out images taken from some subset of potentially private rooms. This is an image retrieval problem: given a stream of images, we wish to retrieve the private ones, so that they can be filtered out. Since our classification algorithms are imperfect, the user could provide confidence thresholds to select between a highly conservative or a highly selective filter, depending on their preferences and the degree of sensitivity of the spaces.

The top row of Figure 7 shows precision-recall curves for retrieving private images from each of our five datasets.

To generate these, we conducted five retrieval tasks for each dataset, one for each room, and then averaged the resulting P-R curves together. For the local and global features we used the maximum value (across classes) of $P_L(l_i|I)$ and $P_G(l_i|I)$, respectively, as the free parameter (confidence), and for the HMM we used the maximum marginal (across classes) of $P(l_i|I_1, ..., I_m)$ computed by the Forward-Backward algorithm. We see that for House 1, House 3, and Workspace 2 we can achieve 100% recall at greater than 70% precision, meaning that all private images could be identified while removing only 30% of the harmless images. For Workspace 1 we can achieve about 90% precision and recall, whereas for the very difficult House 2, about 40% precision is possible at 90% recall.

The above results reflect the closed scenario, where we assume that the user has enrolled all possible rooms in the space. As a preliminary evaluation of the open locale scenario, we created synthetic streams in which we inserted randomly chosen segments of streams from other datasets, such that about 20% of the images in these noisy streams were in the 'other class' category. This can be interpreted as a user collecting 80% of their images in spaces that are enrolled, which is arguably reasonable. In practice, the distribution of time spent amongst spaces in a building will likely be an individual function. The bottom row of Figure 7 shows the precision-recall curves in this case. While retrieval accuracy degrades somewhat compared to the original streams, in three of the datasets (House 3 and the two Workspaces) we still observe nearly 100% recall at greater than 80% precision. We posit that for the vast amounts of photos obtained in lifelogging applications, such precision values are reasonable as they still leave a large fraction of harmless images for sharing. The blocked photos can always be reviewed manually to identify such false classifications. While these results are promising, evaluation on larger-scale, realistic first-person datasets will be needed to characterize performance in real-world open locale scenarios.

E. Computational performance

Our current version of PlaceAvoider is a proof of concept, implemented on general purpose workstations with a mixture of unoptimized C++, Matlab, Python, and R code. This code takes on average 18.421 seconds to process an image on a 2.6GHz Xeon server. This performance may be reasonable for cloud-based applications with offline computation, but ill-suited for realtime use. We suspect that this running time could be improved significantly through simple optimizations (like re-writing all code in C++). We now discuss algorithmic refinements to improve the usability of PlaceAvoider with mobile devices.

Decreasing local feature matching time. A disadvantage of our local feature classifier is that the minimizations in Equation (1) can be computationally intensive, requiring several seconds per image. However, there is inevitable redundancy between enrollment images, because each of our enrollment datasets consists of several rounds of photo collection and there is spatial overlap between images. We developed four techniques for reducing the runtime and space requirements of the classification algorithm by attempting to remove the redundancy from these room models:

TABLE VI. AVERAGE SINGLE IMAGE CLASSIFICATION TIMES, INCLUDING BOTH FEATURE EXTRACTION AND CLASSIFICATION FOR HOUSE 3 ENROLLMENT IMAGES.

Classifier	Time (s)	Accuracy
Global dense bags of SIFT	14.325	68.9%
Local SIFT	2.517	95.7%
Global bags of HOG	1.110	66.2%
Global LBP	1.107	51.9%
Local SIFT (10% blacklists)	0.996	55.2%
Global bags of SIFT	0.469	89.4%
Global GIST	0.247	65.5%
Global RGB histogram	0.063	57.4%

- *K-means* performs k-means clustering on the set of SIFT features in a room model, representing the room simply as the set of cluster centroids;
- *Locality sensitive hashing* is a dimensionality-reduction technique that attempts to preserve spatial relationships between the hashed and unhashed points [21]. We run LSH and then collapse each hash bin having multiple points into a single descriptor;
- *Approximate Nearest Neighbors* scans through the set of descriptors, iteratively collapsing together the two closest descriptors until a specified number of target descriptors is reached. ANN [2] is used to make this process efficient;
- *Random sub-sampling* simply chooses a subset of the SIFT descriptors at random.

We evaluated these four model reduction techniques by setting their parameters such that each one reduced the number of descriptors by a factor of five on a subset of our Workspace 2 dataset. We found that of these techniques, ANN suffered the worst accuracy (73.7% compared to 93.2% with the full model), followed by k-means (87.3%) and hashing (87.3%). Surprisingly, random subsampling actually worked the best of these techniques (87.9%).

Classifier running times. Table VI presents the average running time for our various local and global feature classifiers, including the local classifier with full room models and one randomly subsampled to 10%. As a point of reference, we also show the accuracy of each individual feature type in classifying images from the House 3 cross-validation dataset. We observe that most of the computation time is due to one feature, global Dense Bags of SIFT, due to the fact that it has to compute SIFT descriptors along a dense image grid. The other features show a general trade-off between accuracy and running time: local feature matching performs best (95.7%) but requires the most time (2.5 seconds), whereas RGB histograms require only a few milliseconds per image, but the accuracy is quite low (56.3%).

Once the local and global classifiers are done, stream classification using HMMs is very fast, taking about 0.077 seconds to classify an entire stream or about 0.1 milliseconds per image. HMM inference takes asymptotic time linear in the number of images and quadratic in the number of rooms.

A lightweight classifier. From the results in Table VI we hypothesized that we could build a lightweight classifier that

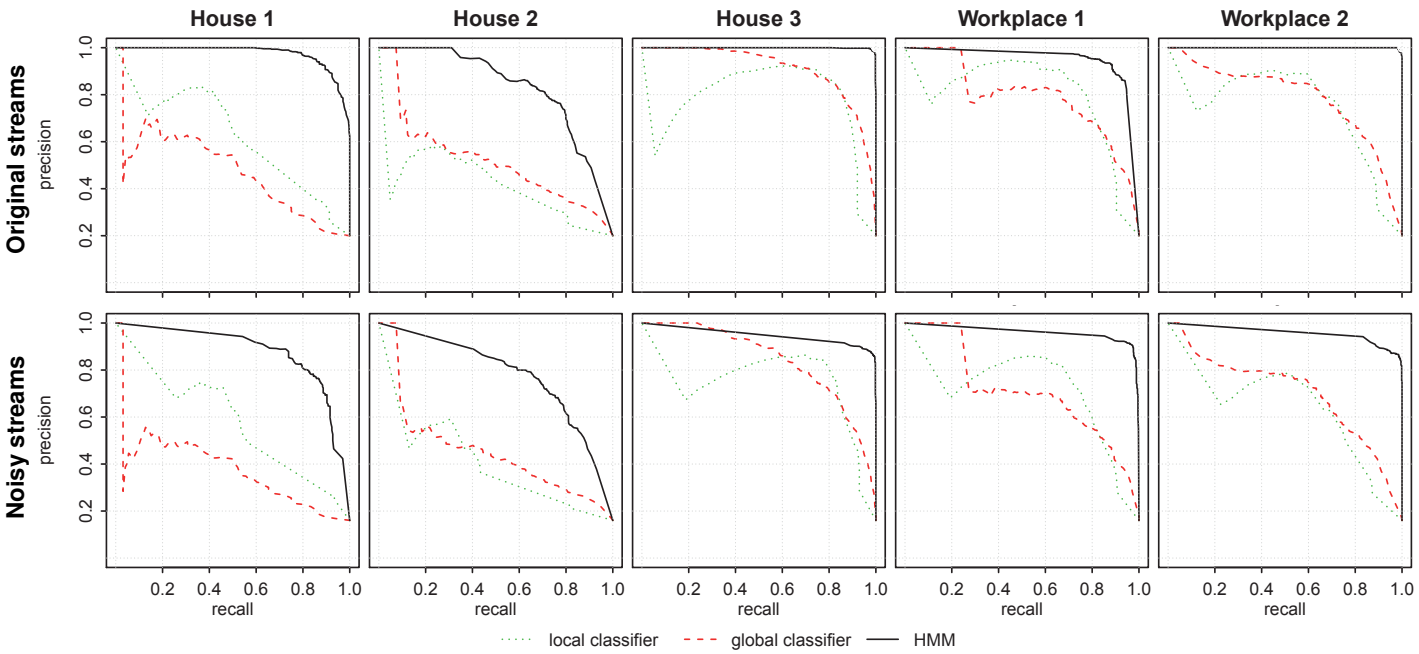


Fig. 7. *Top:* Precision-Recall curves for retrieving images from a given room, averaged over the five rooms, for each of our five datasets. This represents image retrieval in the closed locale scenario. *Bottom:* P-R curves for retrieving images from our noisy dataset with images from other rooms injected into the stream, simulating the open locale scenario.

works nearly as well as the full classifier. We thus built a simple classifier consisting of one local and one global feature, chosen by their low computation demand compared to accuracy: local subsampled features and global bags of SIFT. We tested this lightweight classifier on the Workspace 2 and House 3 datasets, finding that it reduces the average classification accuracy from about 99.9% to 90.5%, but reduces the computation time by over an order of magnitude (from 18.421 seconds to 1.465 seconds per image). Such a lightweight classifier could likely be run on a mobile device.

V. Discussion

Cloud vs. local computation. As discussed in Section IV-E, limited mobile computation resources may impact the performance of PlaceAvoider's image classification. While lightweight classifiers (suitable for the mobile device) provide reasonable accuracy, maximal performance for realtime applications could be realized by outsourcing computation to the cloud, similar to apps on mobile devices today. This requires network connectivity and sufficient bandwidth resources for image uploads. As shown in Table II, our classifiers perform well with down-sampled images, thus reducing bandwidth requirements.

An implementation of PlaceAvoider may default to classification on the cloud, but utilize a lightweight onboard classifier (Section IV) during periods of wireless unavailability. Cloud-based photo-sharing sites could also integrate the full version of PlaceAvoider to aid in labeling of images and content based image retrieval.

Privacy of other people. PlaceAvoider offers users a degree of useful control over images collected in their personal spaces, but this control is limited only to the person's own camera. People's concerns of imaging privacy are often due to the presence of cameras that are not their own. To mitigate these concerns, we would ideally allow people to specify policies for other cameras, even when they are owned by other people. For example, people may share their policies with other users (e.g., social or professional contacts), or use a central repository for such sharing (e.g., Bob enrolls his office in PlaceAvoider to prevent Mary from taking a photo in that space with her phone). Such protocols for surveillance cameras, a more tractable problem, have been proposed [4], [48]. A major challenge with such an approach is that people may not want to share their stream and/or enrollment photos due to privacy concerns, and even sharing abstract models may reveal some private details. Halderman et. al. address this concern in their system that requires unanimous consent amongst bystanders in a privacy-preserving way [22]. Policies may be enforced using models based on secure SIFT (based on features extracted from homomorphically encrypted images), which has been shown to perform well [25].

Improving image classification. While PlaceAvoider generally performed well in our evaluation, it does not yield perfect classification accuracy and thus has much room for improvement. For example, we investigated the lower performance of datasets for House 1 and House 2, and found a high negative correlation between classifier performance and the variance in the quantity of extracted SIFT features among spaces in the dataset; classifier bias is induced towards rooms that have more SIFT features. Minimizing these bias effects amongst enrolled spaces should significantly improve overall classifier performance. Our local feature extraction uses grayscale images, as is standard practice; integrating color information may improve performance significantly. Finally, we employed

no conditioning, noise-reduction, filtering, or other processing of images before feature extraction, and pre-processing steps could improve classifier performance. We leave these areas of improvement for future work.

Leveraging other characteristics. The image classification techniques we use offer reasonable performance to analyze large streams of images. More sophisticated analysis is possible and could offer improvements to PlaceAvoider. For example, people could enroll specific objects in a room, and these could be used to identify sensitive spaces (e.g., if a particular art object or high-end electronics device is detected in an image). While our enrollment process is not burdensome, the system would be improved by bootstrapping available images to eliminate the collection of separate enrollment images.

Other semantic, scene-level analyses could offer better identification of sensitive images, even in areas that have not been enrolled, using scene classification algorithms [56], [41], [33]. For instance, we could build systems that try to estimate a general *type* of room, like kitchen, bathroom, etc., based on general models of these spaces (i.e. what these rooms typically look like). While this general scene categorization would be desirable, computer vision work has shown that recognizing specific targets is much more accurate than recognizing categories of objects; e.g., it is much easier to build a specific model of *your* bathroom than a general model to recognize *any* bathroom. Another possibility is to analyze the poses and activities of people in the scene to provide additional evidence to the classifier, using work on people and pose recognition [11], [14]; photos showing people in distress, in compromising poses, or wearing little clothing could be flagged as sensitive. We leave such an exploration to future work.

VI. Related Work

Lifelogging issues and privacy. Allen [1] and Cheng et al. [8] demonstrate that there is a maelstrom of legal issues related to lifelogging, many of which are privacy related. Specifically, Allen discusses how in the United States, cloud-stored life logs are not afforded 4th and 14th Amendment protections, and this raises the importance of controlling *which* information to log. The expert opinions on lifelogging privacy issues were validated by a user study that was performed to measure perceptions of lifelogging [28]. They found that users want control over the data that is collected and stored, thus motivating the need for technologies like PlaceAvoider. The existing work that seeks to preserve privacy for lifelogging is notably limited. Chaudhari et al. [7] offer a protocol to detect and obfuscate faces in lifelogged video streams. Interestingly, the bulk of cited work on lifelogging was framed with then-current technology. Current lifelogging devices and smartphone lifelogging apps (like Saga[7]) are much more advanced in their collection capabilities while not addressing many privacy concerns.

Camera permissions. Systems like PlaceRaider [52] demonstrate the need for controls on the use of cameras on smartphones and problems that can stem from coarse-grained permissions. The inadequacy of coarse-grained permission systems for sensitive resources has been well documented. Bugiel et al. include a survey of least-privilege-preserving approaches in their XMandroid paper [5] and propose a defense system to prevent privilege escalation. Privilege escalation is an orthogonal problem and has been addressed by systems that automatically prevent installing programs based on permissions [16], [12] or that monitor inter-app communications [19], [6] among other approaches.

Systems have been proposed that can enforce fine-grained permission policies including Apex [40], Porscha [42], and CRePE [10]. Similarly, labeled images can be tracked with mechanisms like TaintDroid [15] and Paranoid Android [44]. PlaceAvoider differs from these systems in that it can dynamically assess the sensitivity of sensor-data content.

Roesner et al. [46] implement a system where the enumeration of fine-grained policy rules is not necessary, instead electing to capture user intent at the time of resource use. This approach helps in applications where users deliberately tap a button to take a photograph and are explicitly aware of the specific photo being taken. Our work addresses precisely the opposite scenario where photos are taken in the background and thus intention-based access control does not provide a suitable defense.

Imaging defenses. There have been very few systems analogous to PlaceAvoider that seek to control the collection of imagery. Truong et al. [53] describe a third-party system where offending CCD or CMOS cameras in a space can be detected and disabled via a directed pulsing light. While this system provides an interesting and useful way to prevent the use of cameras, it requires specialized and dedicated infrastructure to be installed in *each* sensitive space. PlaceAvoider allows similar functionality to be integrated within the camera.

The DARKLY system [26] presents a novel approach to add a privacy-protection layer to systems where untrusted applications have access to camera resources. DARKLY integrates OpenCV within device middleware to control the type and amount of image content available to applications. This approach applies the principle of least privilege to image information, albeit in a different manner than PlaceAvoider. For example, a policy may exist that permits an application only to have access to the number of faces detected in any image. Regardless of context, when invoking the camera with DARKLY, this application would receive only select parameterized image information (e.g., the number of detected faces). PlaceAvoider, however, enforces policies based on image context derived from image content. While solving different problems, DARKLY and PlaceAvoider could potentially be combined — e.g., analysis by PlaceAvoider could inform transformations applied by DARKLY.

Inferring location from images. Inferring location or user activity from smartphone sensors is an active research area. CenceMe [39] uses ambient audio and movement information to infer activity and conversation type, but simply uses the GPS service for location — recorded images are not used for classification. CrowdSense@Place [9] *does* use computer vision techniques (alongside processing of recorded audio) to classify location amongst one of seven general categories (e.g., home, workplace, shops) — this system was not evaluated for its ability to perform the specific scene recognition that

[7]Saga: http://www.getsaga.com

PlaceAvoider performs but this approach would be useful to identify general types of locations where privacy risks are high.

Much of this work is in the computer vision domain for robotics applications. Robot topological localization techniques often require specialized cameras that are incompatible with form factors used by phones and lifelogging devices. Se et al. use a Triclops stereo vision camera for their localization techniques with robots [49], [50]. Similarly, Ulrich and Nourbakhsh use a specialized 360-degree panoramic camera that operates in a fixed plane [54].

Even in the absence of such specialized cameras, localization techniques for robot applications often leverage other conditions that cannot be assumed for our use cases. Ledwich and Williams offer a system that imposes strict constraints on the training images that are unrealistic in the applications that we propose [34]. Kosecka and Li propose a system [31] that uses contiguous streams for training along with precision odometry (instrumentation that measures distanced traveled over time). Similarly, Jensfelt et al. developed a localization system [27] that requires odometry or other dead-reckoning sensors. While sensor arrangements on mobile devices are increasing in sophistication and capability, these localization solutions from the robotics domain are not directly applicable given the dynamics of movement for mobile devices.

Recent work has studied geo-location in consumer images, although most of this work has been limited to highly photographed outdoor landmarks where thousands or millions of training images can be downloaded from the web [37], [51], [20], [36]. An abstraction of absolute camera location seeks to classify images based on the type of scene (e.g., indoors vs. outdoors). Oliva and Torralba label scenes according to the 'gist' of the image by analyzing the distribution of spatial image frequencies [41]. Subsequent work seeks finer granularity by classifying the type of scene at a high level (e.g., living room vs. bedroom) [56], [45]. The majority of work has considered well-composed, deliberately-taken images, although some very recent papers in the computer vision literature have considered first-person video. This work includes selecting important moments from raw first-person video [35], jointly recognizing and modeling common objects [18], inferring the camera owner's actions from object interaction [43], and even using first-person video to collect psychological data about people's visual systems in naturalistic environments [3]. None of this work considers privacy issues as we do here, although in future work we plan to leverage some of these approaches to assign semantic labels that have privacy meanings.

Indoor localization and positioning. The computational expense of inferring camera location with computer vision approaches applied to images may be mitigated partly through localization and positioning methods to reduce search spaces. A comprehensive survey of localization and positioning approaches is outlined by Hightower [23]. Most of these systems require external infrastructure (e.g., audio or electromagnetic beacons) or a dense constellation of cooperating devices [47], and *a priori* knowledge of the environment (e.g., maps) is often required. Some approaches rely less on infrastructure and operate in a peer-to-peer ad hoc manner. Kourogi [32] developed a system that requires no infrastructure, but uses sensors that are much more sophisticated than what is available in consumer mobile devices. Woodman et al. developed a system [55] that performs effective localization, but requires a sensor array affixed to an individual's foot. As discussed in Section I, camera location and the location of image content is not necessarily the same; the PlaceAvoider classifier is necessary to enforce privacy policies based on image content.

VII. Conclusion

We believe that as cameras become more pervasive and as the background collection of imagery becomes more popular, people's privacy is put at increasingly greater risk. We have presented an approach for detecting potentially sensitive images taken from first-person cameras in the face of motion, blur, and occlusion, by recognizing physical areas where sensitive images are likely to be captured. Owners of cameras can review images from these sensitive regions to avoid privacy leaks. We believe this is an important first step in this increasingly important area of privacy research.

Our results are promising and may be good enough for some applications, but our classifier accuracies are likely insufficient for others, and the problem of highly accurate indoor visual place classification from first-person imagery remains open. We plan to continue investigating computer vision techniques that estimate meanings of images to better identify potentially sensitive photo content and situations. We also plan to investigate privacy concerns of *bystanders* — the people being captured within the images — because as devices like Google Glass become more common in society, bystanders need ways to actively protect their own privacy.

Acknowledgment

This material is based upon work supported by the National Science Foundation under grants CNS-1016603, CNS-1252697 and IIS-1253549. This work was also partially funded by the Office of the Vice Provost of Research at Indiana University Bloomington through the Faculty Research Support Program. We thank the anonymous reviewers for their valuable comments and John McCurley for his editorial help.

References

[1] A. Allen, "Dredging up the past: Lifelogging, memory, and surveillance," *The University of Chicago Law Review*, pp. 47–74, 2008.

[2] S. Arya and D. Mount, "Approximate nearest neighbor queries in fixed dimensions," in *ACM Symposium on Discrete Algorithms*, 1993.

[3] S. Bambach, D. Crandall, and C. Yu, "Understanding embodied visual attention in child-parent interaction," in *Joint IEEE International Conference on Development and Learning and and on Epigenetic Robots*, 2013.

[4] J. Brassil, "Technical challenges in location-aware video surveillance privacy," in *Protecting Privacy in Video Surveillance*. Springer, 2009, pp. 91–113.

[5] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, and A.-R. Sadeghi, "XManDroid: A new Android evolution to mitigate privilege escalation attacks," Technische Universität Darmstadt, Technical Report TR-2011-04, Apr. 2011.

[6] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, A.-R. Sadeghi, and B. Shastry, "Towards taming privilege-escalation attacks on Android," in *19th Annual Network & Distributed System Security Symposium (NDSS)*, Feb. 2012.

[7] J. Chaudhari, S. Cheung, and M. Venkatesh, "Privacy protection for life-log video," in *IEEE Workshop on Signal Processing Applications for Public Security and Forensics*, 2007, pp. 1–5.

[8] W. Cheng, L. Golubchik, and D. Kay, "Total recall: are privacy changes inevitable?" in *ACM Workshop on Continuous Archival and Retrieval of Personal Experiences*, 2004, pp. 86–92.

[9] Y. Chon, N. Lane, F. Li, H. Cha, and F. Zhao, "Automatically characterizing places with opportunistic crowdsensing using smartphones," in *ACM Conference on Ubiquitous Computing*, 2012, pp. 481–490.

[10] M. Conti, V. T. N. Nguyen, and B. Crispo, "Crepe: context-related policy enforcement for Android," in *International Conference on Information Security*, 2011, pp. 331–345.

[11] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[12] M. Dietz, S. Shekhar, Y. Pisetsky, A. Shu, and D. S. Wallach, "Quire: Lightweight provenance for smart phone operating systems," *CoRR*, vol. abs/1102.2445, 2011.

[13] M. Douze, H. Jegou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for web-scale image search," in *ACM International Conference on Image and Video Retrieval*, 2009.

[14] K. Duan, D. Batra, and D. Crandall, "A Multi-layer Composite Model for Human Pose Estimation," in *British Machine Vision Conference*, 2012.

[15] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones," in *USENIX Conference on Operating Systems Design and Implementation*, 2010, pp. 1–6.

[16] W. Enck, M. Ongtang, and P. McDaniel, "Mitigating Android software misuse before it happens," Pennsylvania State University, Tech. Rep. NAS-TR-0094-2008, 2008.

[17] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[18] A. Fathi, X. Ren, and J. Rehg, "Learning to recognize objects in egocentric activities," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[19] A. P. Felt, H. J. Wang, A. Moshchuk, S. Hanna, and E. Chin, "Permission re-delegation: attacks and defenses," in *Proceedings of the USENIX Conference on Security*, 2011, pp. 22–22.

[20] J.-M. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, and S. Lazebnik, "Building Rome on a Cloudless Day," in *European Conference on Computer Vision*, 2010.

[21] A. Gionis, P. Indyk, R. Motwani *et al.*, "Similarity search in high dimensions via hashing," in *IEEE International Conference on Very Large Data Bases*, vol. 99, 1999, pp. 518–529.

[22] J. Halderman, B. Waters, and E. Felten, "Privacy management for portable recording devices," in *ACM Workshop on Privacy in the Electronic Society*, 2004, pp. 16–24.

[23] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, Aug. 2001.

[24] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood, "Sensecam: a retrospective memory aid," in *ACM Conference on Ubiquiotous Computing*, 2006.

[25] C. Hsu, C. Lu, and S. Pei, "Homomorphic encryption-based secure sift for privacy-preserving feature extraction," in *IS&T/SPIE Electronic Imaging*, 2011.

[26] S. Jana, A. Narayanan, and V. Shmatikov, "A Scanner Darkly: Protecting user privacy from perceptual applications," in *34th IEEE Symposium on Security and Privacy*, 2013.

[27] P. Jensfelt, D. Kragic, J. Folkesson, and M. Bjorkman, "A framework for vision based bearing only 3d slam," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1944–1950.

[28] T. Karkkainen, T. Vaittinen, and K. Vaananen-Vainio-Mattila, "I don't mind being logged, but want to remain in control: a field study of mobile activity and context logging," in *SIGCHI Conference on Human Factors in Computing Systems*, 2010, pp. 163–172.

[29] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[30] M. Korayem, A. Mohamed, D. Crandall, and R. Yampolskiy, "Solving avatar captchas automatically," in *Advanced Machine Learning Technologies and Applications*, 2012.

[31] J. Kosecká and F. Li, "Vision based topological markov localization," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 2. IEEE, 2004, pp. 1481–1486.

[32] M. Kourogi and T. Kurata, "Personal positioning based on walking locomotion analysis with self-contained sensors and a wearable camera," in *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2003, pp. 103–112.

[33] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[34] L. Ledwich and S. Williams, "Reduced SIFT features for image retrieval and indoor localisation," in *Australian Conference on Robotics and Automation*, 2004.

[35] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[36] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm, "Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs," in *European Conference on Computer Vision*, 2008, pp. 427–440.

[37] Y. Li, D. Crandall, and D. P. Huttenlocher, "Landmark Classification in Large-scale Image Collections," in *IEEE International Conference on Computer Vision*, 2009.

[38] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[39] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *ACM Conference on Embedded Network Sensor Systems*, 2008, pp. 337–350.

[40] M. Nauman, S. Khan, and X. Zhang, "Apex: Extending Android permission model and enforcement with user-defined runtime constraints," in *ACM Symposium on Information, Computer and Communications Security*, 2010, pp. 328–332.

[41] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[42] M. Ongtang, K. Butler, and P. McDaniel, "Porscha: Policy oriented secure content handling in Android," in *Annual Computer Security Applications Conference*, 2010, pp. 221–230.

[43] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[44] G. Portokalidis, P. Homburg, K. Anagnostakis, and H. Bos, "Paranoid Android: Versatile protection for smartphones," in *Annual Computer Security Applications Conference*, 2010, pp. 347–356.

[45] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[46] F. Roesner, T. Kohno, A. Moshchuk, B. Parno, H. J. Wang, and C. Cowan, "User-driven access control: Rethinking permission granting in modern operating systems," in *IEEE Symposium on Security and Privacy*, 2012, pp. 224–238.

[47] A. Savvides, C. Han, and M. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *International Conference on Mobile Computing and Networking*, 2001, pp. 166–179.

[48] J. Schiff, M. Meingast, D. Mulligan, S. Sastry, and K. Goldberg, "Respectful cameras: Detecting visual markers in real-time to address privacy concerns," in *Protecting Privacy in Video Surveillance*. Springer, 2009, pp. 65–89.

[49] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[50] ——, "Vision-based global localization and mapping for mobile robots," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 364–375, 2005.

[51] N. Snavely, S. Seitz, and R. Szeliski, "Modeling the World from Internet

Photo Collections," *International Journal of Computer Vision*, vol. 80, pp. 189–210, 2008.

[52] R. Templeman, Z. Rahman, D. Crandall, and A. Kapadia, "PlaceRaider: Virtual theft in physical spaces with smartphones," in *Network and Distributed System Security Symposium*, 2013.

[53] K. Truong, S. Patel, J. Summet, and G. Abowd, "Preventing camera recording by designing a capture-resistant environment," in *International Conference on Ubiquitous Computing*, 2005, pp. 73–86.

[54] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *IEEE International Conference on Robotics and Automation*, 2000, pp. 1023–1029.

[55] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *International Conference on Ubiquitous Computing*, 2008, pp. 114–123.

[56] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3485–3492.