Visually Fingerprinting Humans without Face Recognition

He Wang
University of Illinois at Urbana-Champaign

Xuan Bao
Samsung Research America

Romit Roy Choudhury
University of Illinois at Urbana-Champaign

Srihari Nelakuditi
University of South Carolina

ABSTRACT

This paper develops techniques using which humans can be visually recognized. While face recognition would be one approach to this problem, we believe that it may not be always possible to see a person's face. Our technique is complementary to face recognition, and exploits the intuition that human motion patterns and clothing colors can together encode several bits of information. Treating this information as a "temporary fingerprint", it may be feasible to recognize an individual with reasonable consistency, while allowing her to *turn off* the fingerprint at will.

One application of visual fingerprints relates to augmented reality, in which an individual looks at other people through her camera-enabled glass (e.g., Google Glass) and views information about them. Another application is in privacy-preserving pictures – Alice should be able to broadcast her "temporary fingerprint" to all cameras in the vicinity along with a privacy preference, saying "remove me". If a stranger's video happens to include Alice, the device can recognize her fingerprint in the video and erase her completely. This paper develops the core visual fingerprinting engine – *InSight* – on the platform of Android smartphones and a backend server running MATLAB and OpenCV. Results from real world experiments show that 12 individuals can be discriminated with 90% accuracy using 6 seconds of video/motion observations. Video based emulation confirms scalability up to 40 users.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

Keywords

Visual fingerprinting; smartphones; augmented reality; matching

MobiSys'15, May 18–22, 2015, Florence, Italy.

http://dx.doi.org/10.1145/2742647.2742671.

Figure 1: Example social event: Alice views people's posts displayed in her Google Glass. The whole operation orchestrated by the *InSight* server running in the cloud.

1. INTRODUCTION

Imagine a near future where humans are carrying smartphones and wearing camera-embedded glasses, such as the Google Glass. This paper intends to recognize a human by looking at him from any angle, even when his face is not visible. For instance, Alice may look at people around her in a social gathering and see the names of each individual – like a virtual name tag – suitably overlaid on her Google Glass display. Where revealing names is undesirable, only a short message could be posted. People at the airport could post "looking to share a cab ", students in a startup event could post "seeking a co-founder", and Alice could view each individual's posts above their heads (Figure 1). In general, the ability to differentiate individuals visually could enable human-centric augmented reality [21, 30].

Face recognition [29, 34] is a possible approach to the above problem. However, faces are not always visible. Moreover, many people express discomfort releasing their profile pictures to the cloud, given that it can become a permanent identifier for de-anonymizing other content in the web [7]. Ideally, what is necessary is a *temporary* visual identifier, that can be activated at will, and will identify the individual momentarily but not later.

This paper pursues the intuition that human motion patterns and visual appearance (e.g., clothing colors) can together serve as a temporary visual fingerprint. The key idea is simple. Consider Alice looking at an individual X through her Google Glass (or smartphone camera). The *InSight* server could request Al-

ice to upload a short video snippet of that individual, and use the frame sequence in this video to extract a motion fingerprint of X, denoted by V_X^{Alice}. This motion fingerprint is essentially a string of micro-activities such as walking direction, stepping frequency, stopping, turning, etc., extracted from the video. The server can simultaneously request sensor data (e.g., accelerometer, gyroscope, compass) from people around Alice, and extract a similar motion fingerprint from it. Let M_i denote this sensor-based motion fingerprint for user i. By matching V_X^{Alice} against M_i of each user i, the server can find the strongest match, say for $i = Bob$. The server can convey to Alice that she is looking at Bob and display Bob's message (e.g., "looking for interns") on her glass.

Generalizing, visual fingerprints may not only be from motion patterns, but also from clothing colors, body structure, etc. If Alice recognizes Bob through motion fingerprints, she can extract Bob's clothing features and update a database inside the *InSight* server. In the steady state, the database would cache clothing fingerprints for different individuals. When John looks at Bob later, his Glass only needs to send an image of Bob. The server can extract the clothing fingerprint from the image sent by John and match against pre-computed clothing fingerprints, ultimately notifying John that he is looking at Bob. In summary, we believe that a person's non-facial visual appearance can serve as an identifier. There is evidence of this opportunity given that humans can often recognize other humans without looking at their faces. This paper demonstrates that (wearable) cameras and smartphones can together achieve the same.

Realizing the above idea presents a number of challenges. Extracting fingerprints from sensors and videos can be non-trivial, even though a variety of tools are available in the signal processing and computer vision literature. The fingerprints need to be general for scalability across individuals, while being adequately discriminating for identification. Moreover, fingerprint matching must be done across incompatible dimensions (sensor and vision) requiring the system to cope with normalization issues, dynamic ranges, depth, perspectives, etc. Even for matching clothing fingerprints, challenges emerge due to lighting conditions, wrinkles, and various view angles – the front and back of a dress may have different colors and patterns. Finally, the system needs to support incremental deployment (i.e., not everyone may run *InSight*) while bandwidth and energy overheads should be minimal.

While developing a robust system is challenging, we find that the rich diversity in human behavior offers hope. People walk/turn/pause at different time instants, even when they are walking in groups – observed long enough, their motion sequence should begin to become unique. Encouraged by this opportunity of uniqueness, we adopt a "digital" approach to processing the information. Put differently, we express fingerprints as *strings* defined on a pre-specified motion alphabet. An example fingerprint could be $EEEOOR$..., where E, O, and R correspond to the actions of *walkEast*, *noMotion*, and *turnAround*, respectively. Such motion alphabets are extracted from both sensors and videos, allowing *InSight* to employ string matching algorithms for comparing fingerprints.

InSight translates these ideas into a functional system using Android Galaxy phones and videos taken from Google Glasses. We have not attained real-time operations yet – the server runs on MATLAB with links to OpenCV and machine learning libraries, and returns the result within ten seconds. Evaluations from real world demonstrate the ability to discriminate 12 individuals with 90% accuracy, using 6 seconds of video/motion observations. Video based emulation shows the ability to scale the technique to the order of 40 people. The main contributions may be summarized below.

- *Identifying the possibility that human clothing colors and motion patterns could serve as temporary fingerprints, complementing face recognition.* Using these fingerprints as new degrees of freedom for human-centric visual applications.
- *Quantifying the viability and accuracy of fingerprinting with real-world human behavior.* Building a fully functional prototype and demonstrating promise through micro-benchmarks, real-user evaluation, and larger scale video simulation.

The subsequent sections will expand on these contributions beginning with an overview of *InSight*, followed by detailed system design. However, we first discuss a few potential applications.

2. APPLICATIONS

The goal of this paper is to develop the core visual fingerprinting primitives, with the hope that they will enable new use-cases or aid known applications. We briefly discuss a few possibilities here different from the augmented reality application described above.

(1) Privacy Preserving Pictures/Videos (PPP)

The proliferation of cameras, and wearables cameras in recent years, has raised various discussions on privacy. Many citizens have expressed discomfort at the thought of being included in a video or a picture taken by a stranger, even if it was legally done in a public place. Today's solution is to hastily move out of the camera's field of view when it is clear that a picture is being taken. Of course, this is not always possible because people are often unaware that videos or pictures are being taken around them. Sensitized by this, many privacy conscious users have wished for a capability to express their privacy preferences, such as "please remove me from the video".

Now, assume Alice is taking a video, and Bob, present in the field of view, intends to be removed from Alice's video. Bob can share his motion fingerprint with the server. When Alice takes the video and sends it to the server, visual fingerprints can be computed for every individual in the video, and compared against Bob's. A match suggests Bob is indeed in the video – *InSight* can then remove Bob by replacing him with background imagery in the video. Authors in [6] recently proposed using QR codes on clothing as a means of expressing privacy preferences – we believe *InSight* is a more usable solution.

(2) Visual Addressing and Communication

A grocery store is in conversation with us regarding the applicability of *InSight* for customer localization and communication. The idea is to use wide-angle surveillance cameras mounted on high ceilings to observe the top view of moving customers (each customer visible to the camera as a small blob). Consider the case where Alice is shopping in the store and her smartphone records her motion fingerprint, M_{Alice}. The surveillance camera could also compute her motion fingerprint from the video frames, V_{Alice}. Of course, it might take longer since the motion

alphabet (from a top view) will be limited — the camera would only detect `moving`, `paused`, and `moving direction`. However, even these few bits of information should be adequate to disambiguate customers in the scale of a minute (no two customers move/pause in lock step for that long).

Now, it should be possible for the camera to send a message to Alice, by including V_{Alice} as an address inside the message (like a virtual MAC address). When Alice's device receives the message, a comparison between M_{Alice} and V_{Alice} would indicate that the message is meant for her. Thus, the grocery store can now establish a communication channel with Alice, sending her location-based product information (since the camera exactly knows Alice's location). Even Alice can ask questions such as "where can I find brown rice" and the store can respond with "ahead on your right, at the bottom shelf". Observe that this *visual address-based communication* offers privacy since Alice need not reveal her (permanent) MAC/device IDs. Moreover, she can turn off her motion sensors at will, terminating all interaction with the store.

While the above applications may not be the best, they define the design landscape reasonably well for us to build a generic visual fingerprinting system. The most suitable applications, we hope, will emerge in time.

3. SYSTEM OVERVIEW

This section presents a functional overview of *InSight*; the technical components will follow in Section 4. For preserving context, we use the augmented reality application as the central theme in the rest of the paper, although the techniques extend (with minor modifications) to the other applications.

Consider a university-organized matchmaking event for startups, where variety of students and faculty come with the goal of forming teams. Upon entering, users check-in with the *InSight* server and specify their rough locations. Later, say Alice looks at an individual X through her Google Glass (or smartphone camera) and requests to identify the individual. During the initial bootstrap phase, the *InSight* server running in the cloud asks Alice's glass to record and upload a video snippet of that individual. The server also requests sensor data (accelerometer, gyroscope, and compass) from smartphones of all people around Alice (Figure 2). Once the data arrives, the server processes these sensor data and extracts motion fingerprints, M_i, for each user i. This motion fingerprint is essentially a feature vector, where features include `isStanding`, `isWalking`, `walking-direction`, `step-duration`, `phase`, `isRotating`, `pause-timings`, etc. The server also analyzes the video snippet from Alice and computes a similar motion fingerprint, but from the consecutive frames of the video. Let V_X^{Alice} denote this video-based motion fingerprint. By matching V_X^{Alice} against all values of M_i, *InSight* finds the strongest match, say for $i = Bob$. If this matching score is greater than a confidence threshold, the server conveys to Alice that she is looking at Bob, and displays the message Bob intends to share with others, e.g., "PhD student in CS, looking for CEO for mobile analytics startup".

Now, once *InSight* recognizes Bob in the video snippet, it extracts Bob's color fingerprint, C_{Bob}, and updates a fingerprint database. This fingerprint essentially captures color features and patterns from Bob's clothing. Thus, in the steady state, when everyone's color fingerprint is registered in the database, users no longer need to update videos or sensor readings. If John looks at Bob later (see Figure 3), his Glass only needs to send to the server an image of Bob. The server extracts the color fingerprint from John's image, C_X^{John} and matches against all C_i in the database (the candidate set can be trimmed using John's rough location). Assuming C_X^{John} matches best with $C_{i=Bob}$, and the matching score is above a threshold, the server informs John that he is looking at Bob. Of course, these color fingerprints are valid in the time scale of events – for another event next day, people will be wearing different clothes, and the color database will have to be re-populated.

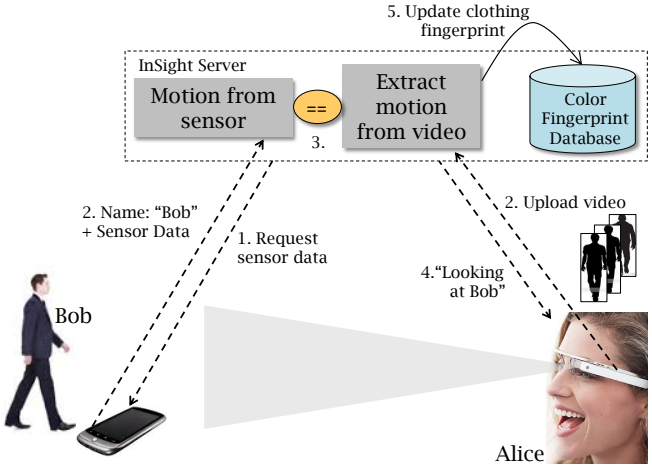

Figure 2: Motion fingerprint based matching during initialization and for new *InSight* users.

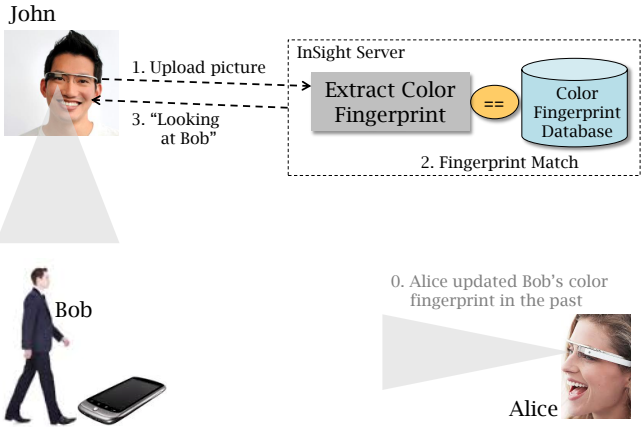

Figure 3: Color fingerprint matching in steady state.

We make two observations about the properties of motion and color fingerprinting.

1. A short window of sensor data is mostly adequate to compute a discriminating motion fingerprint for an individual. This is due to the inherent diversity of human motion, i.e., people's micro-motions are not likely to be synchronous for long durations, and the first instance of "asynchrony" can be used to tell them apart.

2. As mentioned earlier, once Alice identifies Bob using motion fingerprints, Bob's color fingerprint, C_{Bob}^{Alice}, is added to the fingerprint database ($C_{Bob} = C_{Bob}^{Alice}$). When John

identifies Bob later, perhaps from a different angle, Bob's fingerprint is further refined ($C_{Bob} = C_{Bob} \cup C_{Bob}^{John}$).

In general, people's color fingerprints increasingly become complete over time, which improves the accuracy of recognition, which further completes the fingerprint. Thus, *motion fingerprints are only necessary to "register" a person for the first time. Once* InSight *has bootstrapped, color profiles become effective, and motion fingerprints are used only to boost matching confidence, if necessary.* The details on what constitutes color and motion fingerprints are presented in Section 4.

Matching Motion Fingerprints

Motion–fingerprint matching at the *InSight* server is non-trivial because the fingerprints are in different domains – M_{Bob} is obtained from accelerometer/gyroscope/compass readings, while V_X^{Alice} is extracted from video frames. To bring compatibility, we propose to translate all motion fingerprints into a common *semantic alphabet*, where example alphabets are "walking north", "rotating", "pausing" (see example alphabets in Table 1). Thus, both M_{Bob} and V_X^{Alice} are represented as strings on this alphabet, and fingerprint matching boils down to string matching. As an example, say Bob and Neil walk northward for 3 time units, and then Bob pauses while Neil continues walking for one more unit, and then Neil pauses too. Their respective strings will then be $M_{Bob} = NNNOO...$ and $M_{Neil} = NNNNO...$, and it would be possible to tell them apart at the fourth time unit. Specifically, if Alice is looking at Bob, then *InSight* will compute $V_X^{Alice} = NNNOO...$, which is expected to match better with M_{Bob}. In our actual implementation, the motion alphabet is far more sophisticated, including different directions of walking, duration and phase of walking steps, rotations, etc. In fact, for the above case, *InSight* will analyze the duration and phase of walking steps, and unless Bob and Neil are well synchronized in their footsteps, they will be separated within 4 time units.

Table 1: A few examples of motion alphabet. *InSight* uses much more fine grained alphabets.

O	stationary, paused
N	walking north
S	walking south
E	walking east
W	waling west
R	rotating
U	undetermined motion

4. SYSTEM DESIGN

This section describes the extraction of motion and color fingerprints from sensor and video data, followed by fingerprint matching schemes. The techniques borrow from literature where suitable, with appropriate adaptions to this specific cross-sensor application.

4.1 Extracting motion from sensor data

We begin with a description of how sensor readings from Bob's smartphone are translated into a motion string. The key motion alphabets that make up this string are derived from: (1) rotation, (2) walking, (3) walking step duration, (4) walking step phase, and (5) walking direction.

• **Rotation Detection**: Since the phone can be in an unknown orientation, we cannot rely on any single axis of the gyroscope (x, y, or z) to properly detect rotation. Therefore, we first project the rotation rate vector $r = (r_x, r_y, r_z)$ on to gravity vector $g = (g_x, g_y, g_z)$ in the phone's coordinate system, i.e., $r_{gravity} = (r_x g_x + r_y g_y + r_z g_z)/\sqrt{g_x^2 + g_y^2 + g_z^2}$. Then, the rotation angle around gravity is $\alpha_{gravity} = \int r_{gravity} dt$. If $\alpha_{gravity}$ exceeds a certain threshold, the user's motion is labeled as rotating (denoted by R in the motion alphabet). The alphabet R is essentially a binary indicator and its natural to ask: *why not extract more bits of information from rotation?* This is because detecting various degrees of human rotation from videos is a difficult problem.

• **Walking Detection**: The act of human walking manifests on the accelerometer with periodic high/low impulses as well as some rotation around the gravity axis. However, due to movements of the phone inside the pocket, and certain unusual gait patterns, simple threshold based schemes were inadequate to recognize walking patterns. Instead, we employ a bagged decision tree model and train it with two features, namely (1) standard deviation on the magnitude of accelerometers, and (2) rotation around gravity. We omit details in the interest of space, but found this to offer high consistency (as evident in Section 5.1).

• **Walking Step Duration**: Given that people walk with varying speeds, the duration of walking steps can be a useful discriminator. To this end, we first detect if the user is walking; if so, we apply a standard Kalman filter on the accelerometer data to first smooth out the reading and accurately identify the local maxima. These local peaks – shown in Figure 4 – are actually the time points when the human feet land on the ground. The peaks closer to the taller raw impulses correspond to the leg that carries the phone. The step duration, denoted T_{step}, may be computed as half of the time window between two consecutive peaks. Note that T_{step} can be computed without prior knowledge of which pocket the phone is in. Even if the phone is in the shirt pocket, or held in the hand, such peaks are visible, and T_{step} can be computed. We have not evaluated the case of backpacks and jacket pockets.

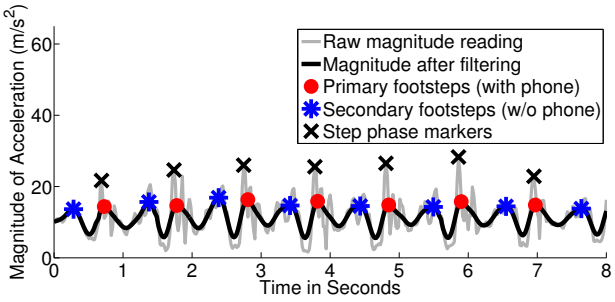

Figure 4: Walking: magnitude of accelerometer readings (smoothed) with step marker while user is walking.

• **Walking Phase**: Even when two users are walking at the same speed, i.e., T_{step} is identical, their exact step timings may be out of phase. Thus, the phase of walking can also be a component of the motion fingerprint of a person. To this end, we use the peak around the bigger jerk as the step phase marker, shown in Figure 4. If different users exhibit these peaks at different times, they may be separated so long as time is appropriately synchronized among devices. If the accuracy of synchronization is upper bounded by, say δ, then the phase differences can be measured in that granularity. We show later that our synchronization is in

the order of $\frac{T_{step}}{3}$, permitting us to create 3 buckets of users with respect to their walking phases.

• **Walking direction**: *InSight* intends to leverage the user's walking direction, with some granularity, as an attribute of her motion. However, walking direction estimation is challenging given that the phone is in an unknown orientation on the user's body. The problem is difficult because the act of walking imposes various kinds of vertical, horizontal, and sideward forces on the smartphone (to stabilize the body), and there is a narrow window during which the acceleration on the phone is most dominantly along the user's heading direction. This narrow time window is actually when the user's leg swings (or rotates) forward, captured by the cross product of rotation axis, $\mathscr{R}_{axis}$ and gravity g. More precisely, the user's heading vector, $H = \mathscr{R}_{axis} \times g$, illustrated in Figure 5. Given the rotation matrix $\mathscr{R}$ from gyroscope, $\mathscr{R}_{axis}$ is the null space of $(\mathscr{R} - I)$ (I is the identity matrix) and rotation angle $\mathscr{R}_\theta = arccos(\frac{trace(\mathscr{R})-1}{2})$. In our system, we make $\mathscr{R}_\theta$ always positive and the sign of $\mathscr{R}_{axis}$ is determined by the right-hand rule. When the leg carrying the phone (called the primary leg) swings, the $\mathscr{R}_{axis} \times g$ points to user's heading direction; the same cross product points in the opposite direction when the secondary leg swings.

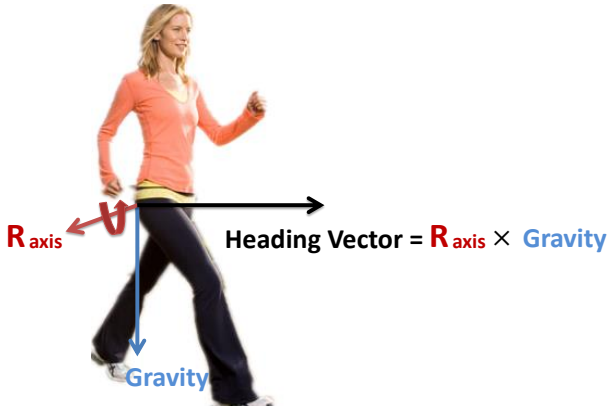

Figure 5: Heading vector = cross product of rotation axis and gravity.

Importantly, this cross product must be computed when the leg is in full swing, otherwise, the sensor data can be polluted with noise (especially when the leg slows down and strikes the ground). To avoid such noise, *InSight* chooses a period of 30% of T_{step} starting from the time when the secondary leg strikes the ground (i.e., when the primary leg is about to swing). $\mathscr{R}_{axis}$ is derived from this time window. Figure 6(b) shows the $H(t) = \mathscr{R}_{axis}(t) \times g(t)$, where $\mathscr{R}_{axis}(t)$ is the rotation axis during $[t, t+30\%T_{step}]$. Clearly, $H(t)$ alternates its direction as the user swings its primary and secondary legs alternately.

The heading vector is in the phone's coordinate system and needs to be interpreted in the global magnetic reference frame (i.e., with respect to North). For this, we project the magnetometer reading to the horizontal plane [32], and record the angle between this projected vector and the heading vector. Figure 6 shows the results. Vertical dotted lines are the moments when the heading vector is recorded. As expected, heading direction alternates roughly 180° between primary and secondary steps. At the moments of heading vector recordings, the direction matches the ground truth as it corresponds to the window when the primary leg swings. Upon close scrutiny, we find that the estimated walking directions are always slightly higher than

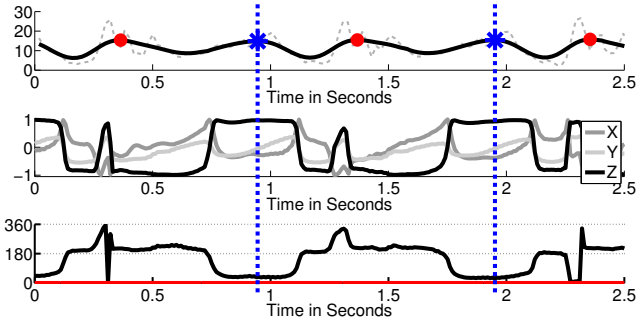

Figure 6: Walking direction: (top) Magnitude of acceleration while user is walking. (middle) Heading direction. Dotted lines are the moments when heading vector is calculated; (bottom) Estimated walking direction (ground truth of 0° in red). As expected, they match at the primary steps and differ by 180° at the secondary steps (with heading direction opposite of the primary).

the ground truth. This is because human leg motions are often slightly diagonal, and the actual walking direction is an average of two diagonals from two steps, resulting in forward locomotion. We use a calibration factor of 35° to compensate for this effect. The final walking direction is divided into 8 directions, in the granularity of 45°, which creates 8 motion alphabets. The following subsection describes how the same motion alphabets are also derived from videos (of Bob) captured from (Alice's) Google Glass camera.

4.2 Extracting motion from video

Given that a video can contain multiple moving individuals, *InSight*'s first step is to mark and track every person in the video. This calls for enveloping each individual with a bounding box. Of course, for motion related insights (such as walking period, phase, etc.) lower regions of the bounding box needs to be processed to extract alphabets. This section describes each of these steps systematically.

• **Detection and Tracking**: To detect and track humans in the video, we borrow existing techniques from computer vision [11, 12] and modify it per *InSight*'s needs. Using the borrowed techniques, Figures 7 (a)-(e) show the accuracy of placing a bounding box around each person, along with a confidence score. To cope with false positives, we only consider the boxes with a confidence score above 50. Now, to track people across the video frames, we again employ a Kalman filter [19]. Formally, the state of each *tracker* is denoted by $\mathbf{s}_k = [x_k, y_k, v_{x_k}, v_{y_k}]^T$, where $\{x_k, y_k\}$ and $\{v_{x_k}, v_{y_k}\}$ are the position and speed of the person in the 2D frame k. Observation z is the center of the bounding box obtained from the pedestrian detector. We use the following state and observation equations.

$$s_{k+1} = Fs_k + \omega_k \quad (1)$$

$$z_k = Hs_k + u_k \quad (2)$$

where

$$F = \begin{pmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

dt in F is the duration between adjacent frames. $\omega_k \sim N(0, Q_k)$ and $u_k \sim N(0, R_k)$ are state model noise and measurement noise

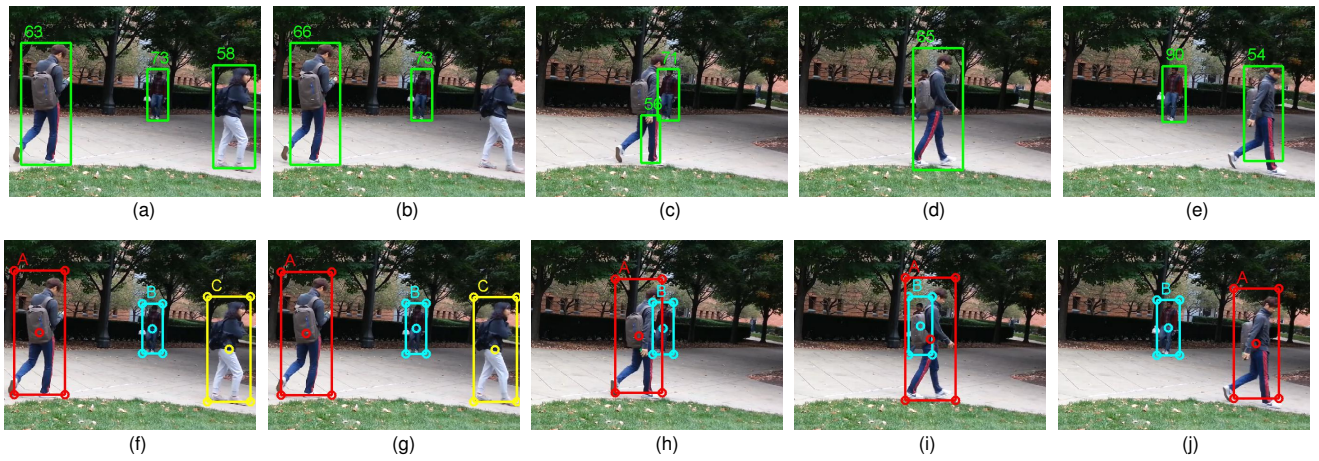

Figure 7: Pedestrian detection and tracking: (a)~(e) are with pedestrian detection; (f)~(j) are with tracking. (b) shows a temporal miss. (c) shows a temporally wrong detection. (d) shows a miss caused by occlusion. As we can see from (f) ~ (j), these errors are fixed by tracking. (f) ~ (j) also show people can be tracked properly after temporal encounters.

respectively. Q_k and R_k are proportional to the person's size. Q_k is also inversely proportional to the number of tracked frames.

When processing a new video frame, we employ the method in [8] to associate each bounding box with a tracker. In the association process, we leverage the observation that position, direction of speed, and size, are not likely to change significantly in adjacent frames; in fact, the higher the speed of the user, the larger the chance that the direction of speed remains the same. Thus, after association, each target's Kalman filter goes one iteration further. Unlike [8], where a single particle filter is used to track the center of the target, we enrich each tracker with four corners of the bounding box. Each corner is also passed through a Kalman filter as described above. The bounding box thus filtered is later utilized for estimating walking direction and step phase detection.

Figures 7(a)-(e) show the results from pedestrian detection. Although reasonable, it fails to detect a person in Figure 7(b), falsely detects one in Figure 7(c), and suffers from occlusion in Figure 7(d). However, Figures 7(f)-(j) show the efficacy of applying tracking. Also, Figures 7(f) and 7(j) show that people can be tracked properly even after temporal encounters.

• **IsWalking and Walking Direction**: To detect whether the target person is walking, we use the speed of the bounding box. Since speed is one of the states in the Kalman filter, we obtain it from the tracking process described earlier. Figure 8 shows 2D speed of the center of the bounding box – higher speeds denoted with a longer arrow. People in Figures 8(a)-(e) are walking and the person in Figure 8(f) is standing in place. If a person's movement has significant speed, then we can detect walking by the mean of 2D speed, s_{xy}, normalized by that person's height, during a predefined period as given by Equation 3.

$$s_{xy} = \frac{1}{L_P} \sum_{i \in P} \frac{\sqrt{(c_x^i)^2 + (c_y^i)^2}}{h^i} \tag{3}$$

where P is a collection of frames in the period; L_P is the cardinality of P; c_x^i and c_y^i are the horizontal and vertical components of the target's center speed at frame i; and h^i is the target's height at frame i. However, if the person is walking mainly along the line perpendicular to the camera's plane (Figures 8(d) and (e)), s_{xy} can be as small as the case where the person is just standing. To cope with this scenario, we use a term s_z to estimate the target's motion along the z axis.

$$s_z = \alpha_z \frac{h^{L_P} - h^1}{mean\{h^i : i \in P\}} \tag{4}$$

where α_z is a calibration factor such that $|s_z|$ and s_{xy} are roughly similar if the target's speed is the same whether moving in 2D plane or along the z axis. When the combined value of $s_{xy} + |s_z|$ of a target is above a certain threshold, we mark that person as walking.

We calculate the user's walking direction with respect to the camera's facing direction. The relative direction is quantized to 8 bins with centers at 0°, 45°, 90°,..., 315°. To classify the direction, we train a model with bagged decision tree with features s_x (Equation 5), s_z, and d_z, where d_z is the slope of linear regression of heights. Intuitively, the bounding box's speed and increase/decrease of its height together help in determining the user's relative direction.

$$s_x = \frac{1}{L_P} \sum_{i \in P} \frac{c_x^i}{h^i} \tag{5}$$

• **Step Duration and Phase**: To detect the duration and phase of walking steps, we choose the lower region of the bounding box obtained from the tracker. We represent the motion in this region through Space-Time Interest Points, which essentially captures fast changes in video pixels. This technique is borrowed from [13] and we omit the details in the interest of space. As an abstraction, the technique marks the fast-changing spots on the video (Figures 9(a)-(c)) with brighter spots indicating faster movements. Clearly, the distribution of spots and their brightness vary while the user is walking. We define a feature, F_{center}, which captures the rhythm of this alteration. Figure 9(d) shows a typical path of F_{center} – the peaks are essentially the step phase, while the time separation between the peaks is the step duration. The technique becomes unreliable when the user is walking towards or away from the camera, in which case, we refrain from extracting step duration and phase.

• **Rotation**: The technique of Space-Time Interest points can be applied for detecting rotation as well. The key observation is that

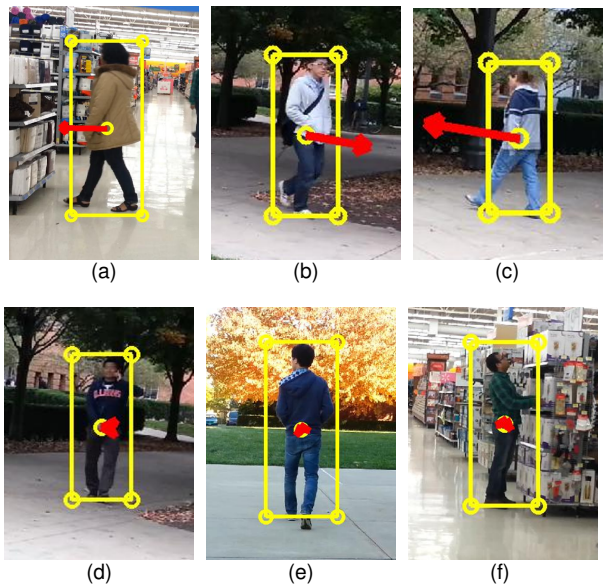

Figure 8: Center speed in 2D plane: (a)-(c) have significant speed, and (d)-(f) have insignificant speed.

rotation causes fast changing spots to be scattered over the entire bounding box, while other activities cause the spots to be confined to a relatively smaller region. Figure 10 shows an example where the spots are confined to the hands when the user is moving them; in contrast, rotation causes the spots to cover the entire body. Formally, we define $X = \{x_i\}$ and $Y = \{y_i\}, (i = 1, 2, .., N)$ as the X and Y coordinates of the centroid of the spots. We define the following four features – $f_1 = max\{X\} - min\{X\}$, $f_2 = max\{Y\} - min\{Y\}$, $f_3 = var\{X\}$, and $f_4 = var\{Y\}$ – and train a bagged decision tree to classify the spot distribution. The output is a binary answer: rotation or not.

• **Unknown**: *InSight* extracts a motion alphabet from each time unit (one second). However, confusions arise when the user transitions from one action to another within that second (e.g., stationary user starts walking). We conservatively deem these time units as an "unknown". The motion string thus contains motion alphabets and unknowns interspersed with each other.

4.3 Matching Motion Strings

We now describe how motion strings obtained from sensors are matched with motion strings extracted from videos. First, since the walking direction estimated from sensors is with respect to global north, we map it to one of the 8 quantized directions relative to our camera-facing direction. Then, *InSight* computes the "distance" between the two strings, similar in spirit to *edit distance*.

Specifically, denote the motion strings based on the video and sensor data as V and M respectively, and their length L (note that the string lengths are same). For each position i in the strings, we compare $V(i)$ and $M(i)$. The difference between them is 0 only if: (1) $V(i)$ and $M(i)$ are identical or (2) $V(i)$ and $M(i)$ both correspond to walking in identical or adjacent directions, their step durations are within a threshold ratio, and their step phase marker of $M(i)$ falls into a range calculated from step phase makers of $V(i)$. Otherwise, their difference is recorded as 1. Let D be the total difference across the length of the strings. Then we define string similarity as $(1 - \frac{D}{L})$. When comparing a video string of

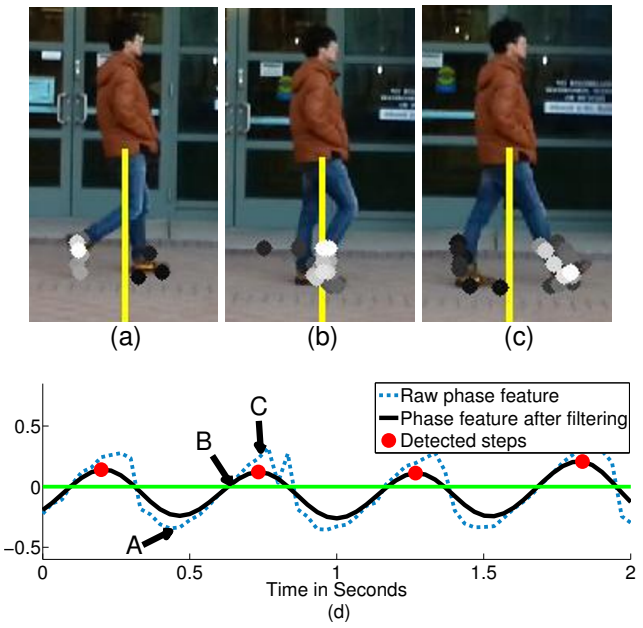

Figure 9: Walking phase and duration: (a)~(c) Centers of detected cuboids; (d) Walking phase feature changes over time and detected steps. Points A, B and C in (d) correspond to the instances in (a), (b) and (c) respectively.

a person X against multiple people's sensor strings, we pick the one (say Bob's) with the single highest similarity (if any). Only if this highest similarity is above a certain threshold, then X is identified as Bob, otherwise we declare the recognition as "unsure".

4.4 Extracting color fingerprint

Once Bob is identified based on his motion, the server extracts and adds his color fingerprint to its repository. Thereafter, server may be able to recognize Bob from an image without seeking video and sensor data, saving both latency and bandwidth. While various visual features of Bob can be considered to form his color fingerprint, clothing is an obvious choice as a temporary fingerprint. Therefore, in this paper, we extract the features of Bob's clothing and use them as Bob's color fingerprint. As a first step in getting the fingerprint, we detect the clothing area in the per-

Figure 10: Scattering of motion spots: (a) rubbing hands, (b) putting hands into pocket, (c) rotating

son's image. Then, we use a well known technique, namely spatiograms, to extract a color fingerprint.

Clothing area detection: We extract color fingerprints when target is in near-front or near-back view. Calvin Upper-body Detector [14] model is trained to return bounding-boxes fitting the head and upper half of the torso. Figures 11(a) and 11(b) show the detected upper-body; Figure 11(c) shows that the detector doesn't fire in other views. We then use a pose estimation model [33] trained on Buffy dataset [15], which returns with joints such as neck, shoulder, etc. The neck and shoulder joints (red lines in Figure 11) help crop the upper-torso area as the clothing area. Then, we apply spatiograms on the cropped image to extract the target's color fingerprint.

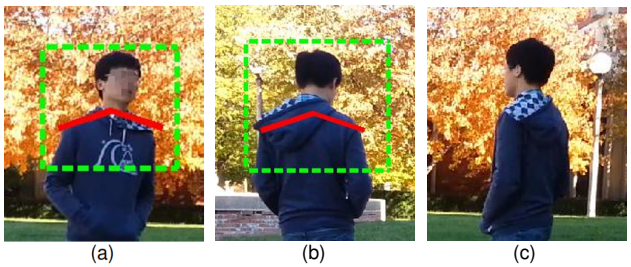

Figure 11: View detection and shoulder estimation. Green dashed boxes show the detected upper-body. The upper-body detector can detect upper-bodies in either near-front view (a) or near-back view (b), but it cannot detect upper-bodies in other views (c). Red solid lines are shoulder extracted by pose estimation.

Spatiograms: Spatiograms are essentially color histograms with spatial distributions encoded in its structure. Put differently, while basic color histograms only capture the relative frequency of each color, spatiograms capture how these colors are distributed in 2D space. The second order of spatiogram can be represented as [5]:

$$h_I(b) = < n_b, \mu_b, \sigma_b >, \quad b = 1,2,3,\cdots,B,$$

where B is the number of color bins, n_b is the number of pixels whose value falls in the b^{th} bin, and μ_b and σ_b are the mean vector and covariance matrices of the coordinates of those pixels respectively. Through such a representation, a white over red stripe can be distinguished from a red over white stripe, even if the number of red and white pixels are identical in both. Also, to cope with various viewing distances, we normalize the spatial information with respect to the *shoulder width* so that all the spatial representation is relative to the captured body size in each photo. Finally, to decouple lighting conditions from the colors, we convert the pixels from RGB to HSV, and quantize them into $B = 5x1x2$ bins.

4.5 Color Fingerprint Matching

When John views Bob through his glass – either from the front or the back – *InSight* again crops out a region around Bob's upper body, and applies the same fingerprinting operations on this image. These fingerprints – one in the repository and another from John – are now ready for matching. Our matching algorithm first computes the spatiogram similarity between each person in John's view with Bob's fingerprint in the repository. Denote the spatiograms to be compared as $S = \{n, \mu, \sigma\}$ and $S' = \{n', \mu', \sigma'\}$, both having B color bins. We define the similarity measure as in [9]:

$$\rho = \sum_{b=1}^{B} \sqrt{n_b n_b'}\, 8\pi |\Sigma_b \Sigma_b'|^{1/4}\, \mathcal{N}(\mu_b; \mu_b', 2(\Sigma_b + \Sigma_b'))$$

Essentially, the similarity decreases (following a Gaussian function) with increasing difference between the colors and their spatial locations. Fingerprints are considered to match if ρ is greater than a certain threshold.

When motion information (video and sensor data) is available in addition to clothing fingerprints, *InSight* server utilizes them both to make the recognition more robust. First, it computes the ρ value for each video frame that captures target's near-front or near-back view, and calculates the mean $\bar{\rho}$. It deems clothing similarity as 1, if $\bar{\rho}$ is above a certain threshold and 0 otherwise. Next, it will compute the overall similarity as the average of motion similarity and color similarity. Then, it will pick a person with the single highest overall similarity. If this person's overall similarity is above a certain threshold, then *InSight* returns the person's name, and unsure otherwise.

5. EVALUATION

This section is organized in 3 parts: **(1) Micro-benchmarks** to evaluate the accuracy with which motion alphabets can be detected from each second of video and sensor data. **(2) Scenario with real users** to evaluate *InSight*'s ability to discriminate individuals through motion/visual fingerprint comparison. **(3) Video simulation** to evaluate scalability across large number of users. The experiment design and details are presented under each of the 3 sub-sections.

5.1 Micro-benchmark (for Motion Alphabets)

Experiment Design:

Motion alphabets define the atomic operation in *InSight* – this section evaluates whether each second in videos and sensor data can be reliably converted to motion alphabets. For this, we recruited 12 volunteers, gave each of them a Samsung Android phone running an *InSight* client, and asked each of them to cover various actions, such as walking, rotations, taking turns, standing still, etc., as well as some upper body movements like checking emails, stretching, etc in an area of $20mx15m$ outside our building. During the entire experiment, a designated observer video recorded each volunteer's motion patterns separately. The experiment was performed in 4 sessions and in total 80 minutes of sensor and video data were collected (and each frame manually labeled for ground truth). These videos were then examined frame by frame and manually labeled with ground truth (i.e., each second of the video tagged with one of the motion alphabets, such as walking or not, walking direction, starting and ending frame of each step, rotating or not, etc.).

Results

Walking detection: We use a 3-fold cross-validation to evaluate the accuracy of walk-detection. Recall that walking detection with sensors is based on bagged decision trees, while for videos, we used a calibration factor α_z and a motion threshold. We set $\alpha_z = 4$ and motion threshold of 0.5. Figure 12 shows the confusion matrix – evidently, the detector is highly accurate for both sensors and videos, with mis-detection not above 0.6% and 1.5% respectively.

Step duration: We painstakingly computed the ground truth for step duration, i.e., the start and end time of each step. This enables comparison with estimates made from video and sensor

	No	Yes
No	99.9%	0.1%
Yes	0.6%	99.4%

	No	Yes
No	98.9%	1.1%
Yes	1.5%	98.5%

Figure 12: Detecting walking vs. not-walking: (a) from sensors; (b) from vision.

data. Figure 13 plots the CDF of the relative error for both dimensions of information. The error distribution with videos exhibits a staircase function since the ground truth was marked in the units of video frames. Overall, the relative error is less than 8% in more than 85% instances for both video and sensor data.

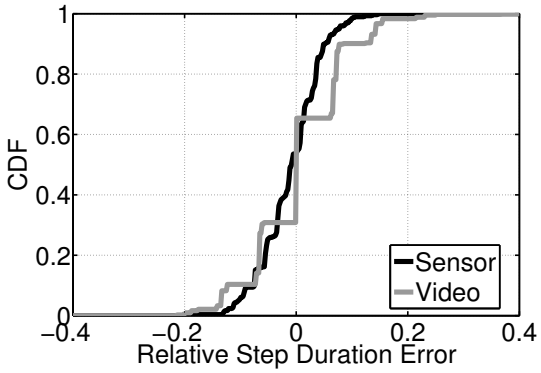

Figure 13: Step duration estimation error compared to ground truth.

Walking direction: Using sensor data, we compute the walking direction w.r.t. global north and plot the relative error in Figure 14. Evidently, the error is not high and confined mostly to $\pm 45^{\circ}$ around the true value. We also estimate the walking direction from the videos and classify them into one of 8 classes – Figure 15 reports the confusion matrix. Classification accuracy (at 45° granularity) is consistently high, and the slight confusion is mostly with adjacent angular directions.

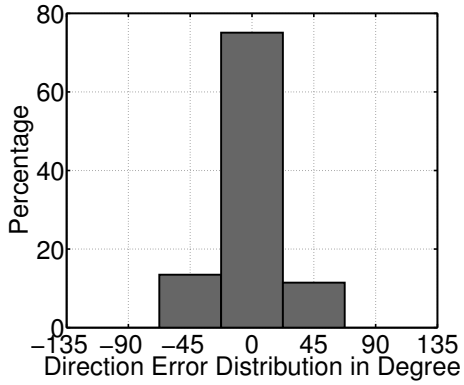

Figure 14: Walking direction estimation: relative error with sensors.

Step Phase: Recall that step phase of two individuals is the difference between the time points at which their respective feet strike the ground. Figure 16(a) shows the histogram of the difference in the phase (normalized by T_{step}), estimated from the sensor data

	0	45	90	135	180	225	270	315
0	99.6%	0.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
45	1.0%	99.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
90	0.0%	0.9%	97.4%	1.7%	0.0%	0.0%	0.0%	0.0%
135	0.0%	0.0%	0.0%	99.0%	1.0%	0.0%	0.0%	0.0%
180	0.0%	0.0%	0.0%	1.7%	98.3%	0.0%	0.0%	0.0%
225	0.0%	0.0%	0.0%	0.0%	0.4%	99.2%	0.4%	0.0%
270	0.0%	0.0%	0.0%	0.0%	0.0%	2.5%	93.4%	4.1%
315	0.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.5%	99.0%

Figure 15: Walking direction estimation: confusion matrix for quantized directions based on video.

and compared against ground truth. Figure 16(b) shows the same histogram but computed from video data. However, for recognizing individuals in *InSight*, what matters is the video-sensor phase offset (i.e., the difference in phase computed between video and sensor data). Figure 16(c) shows this difference. The graph suggests that the resolution at which step phases can be discriminated is around $0.3 \times T_{step}$ (T_{step} is the step duration) – two individuals that are different by less than this value will appear to be in lock-step.

Rotation: Recall that, rotation is detected from videos by training a bagged decision tree and classifying the identified spot distribution (Figure 10). For extracting rotation from sensor data, we compute the rotation angle around gravity, $\alpha_{gravity}$, and apply a threshold of 15°. Figure 17 reports the a confusion matrix – the high accuracy confirms reliable rotation detection.

	No	Yes
No	97.8%	2.2%
Yes	1.3%	98.7%

	No	Yes
No	98.9%	1.1%
Yes	1.2%	98.8%

Figure 17: Rotating or not: (a) by sensors; (b) by vision.

5.2 Real User Scenario

The above evaluation shows the consistent accuracy of detecting motion alphabets from both video and sensor data. This subsection extracts motion strings from individuals and examines the discriminative abilities in them. The overall performance depends on 2 factors: (1) the inherent diversity in human motion patterns, and (2) effectiveness of *InSight*'s fingerprint design and matching schemes.

Experiment Design

We again conduct experiments with the help of the same 12 volunteers as before. But, unlike the micro-benchmark setting, in this set of experiments, the volunteers' motion and behavior were completely natural. They were allowed to naturally move around or pause, and do as they pleased. They were also not

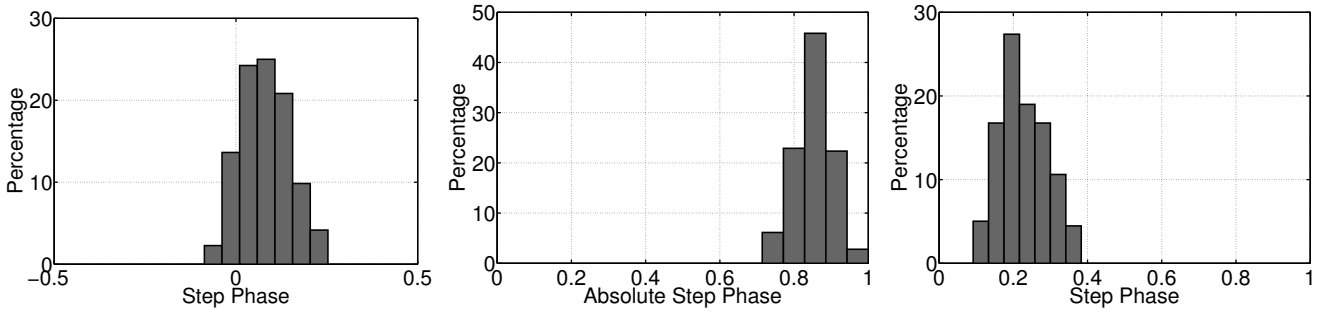

Figure 16: Step phase marker distribution: (a) sensor phase markers with respect to ground truth; (b) video phase markers with respect to ground truth; (c) sensor phase markers with respect to video phase markers.

instructed about clothing – they came to the experiment wearing the same clothes that they wore to school that morning.

We have not developed a real-time version of *InSight* – our current evaluation is offline and structured as follows. We pretend that the observer requests to recognize one of the volunteers at a random time t. Starting at time t in our data set, we crop out a 10 second video of that volunteer, as well as a 10 second sensor stream from all 12 smartphones. A motion string derived from this video is then matched against all the motion strings derived from the sensors. The string matching algorithm either returns a matching smartphone (or volunteer), or returns *unsure* if there is no single highest score or the score is below a threshold (set to 0.95). We repeat this experiment 100 times with different request times.

Results

Figure 18 shows the recognition accuracy. To understand the contribution of different motion alphabets towards overall human recognition, we evaluate various combinations of alphabets. Figure 18(a) starts with the simplest one – walking or not. In other words, for each second of the video and sensor data, a volunteer's motion is categorized as walking or not walking. Then, for increasing string lengths, we plot the performance of the matching scheme. For increasing time durations (on the x-axis), we show the fraction of people correctly recognized, incorrectly recognized, and unsure. For instance, from the first 7 seconds of video and sensor data, we could correctly recognize 2% of the cases and the rest were unsure (none incorrectly recognized). As we consider longer durations of motion, recognition performance improves, reaching 7% with 10 seconds of motion. This improvement is expected since motion of two individuals diverges over time making them more distinguishable. Of course, the distinguishability is not high in this case, since *walking* alone, is hardly a strong discriminator.

To improve over a binary walking indicator, we include *walking direction* in the motion alphabet (quantized to 8 classes) and present its performance in Figure 18(b). Understandably, walking direction helps improve the recognition, up to 30%. Similarly, Figure 18(c) and Figure 18(d) indicate that considering step phase and duration together along with walking direction can recognize individuals correctly in 50% of the cases. Figure 18(e) shows that when rotating or not is further added to the alphabet (in addition to walking direction/phase/duration), a person can be recognized in close to 72% cases with 10 seconds of observations.

Finally, clothing fingerprints can also be used in combination with motion based alphabets for recognizing humans. Figure 18(f) shows that when clothing fingerprint is used in conjunction with motion, 6 seconds of observation are sufficient to recognize a person with 90% probability. Overall, these results demonstrate that motion and clothing together help recognize individuals accurately, and recognition performance improves over time.

5.3 Video Simulation (for Scale)

Experiment Design

Recruiting large number of volunteers and bringing them for multiple social experiments proved more difficult than we imagined. Still, to gain insights on *InSight*'s scalability to higher user density and different settings, we resort to an approximation. Our key idea is to record video of people in public places, and even though we do not have sensor data from them, we will *synthesize sensor data by injecting statistical error into ground truth observations*. For this we execute the following steps:

- record videos of people in public places, such as university cafes, grocery stores, busy street intersections.
- extract the video-based motion fingerprints for each user in the video, denoted V_i,
- also manually extract ground truth for each user from the video, denoted T_i,
- inject errors into the ground truth based on past error distributions, observed when extracting motion alphabets from sensors (i.e., $M_i = T_i + Error$),
- compare the video based motion strings to the synthetic sensor based strings (i.e., $V_x == M_i$?).

Our earlier evaluation in Section 5.1 demonstrated that sensor based strings achieve high accuracy, so this synthetic sensor strings should well mimmic reality. The results should be a faithful approximation of *InSight*'s performance at scale.

The motion alphabets are synthesized as follows. To determine whether a person is walking or not, we use the confusion matrix shown in Figure 12(a). Specifically, if a person is walking as per the ground truth, then she is marked as walking with 99.4% probability and as not-walking with 0.6% probability; when she is not walking, her motion is marked as walking and not-walking with 0.1% and 99.9% probability respectively. We follow the same for deciding rotating or not, using the rotation confusion matrix in

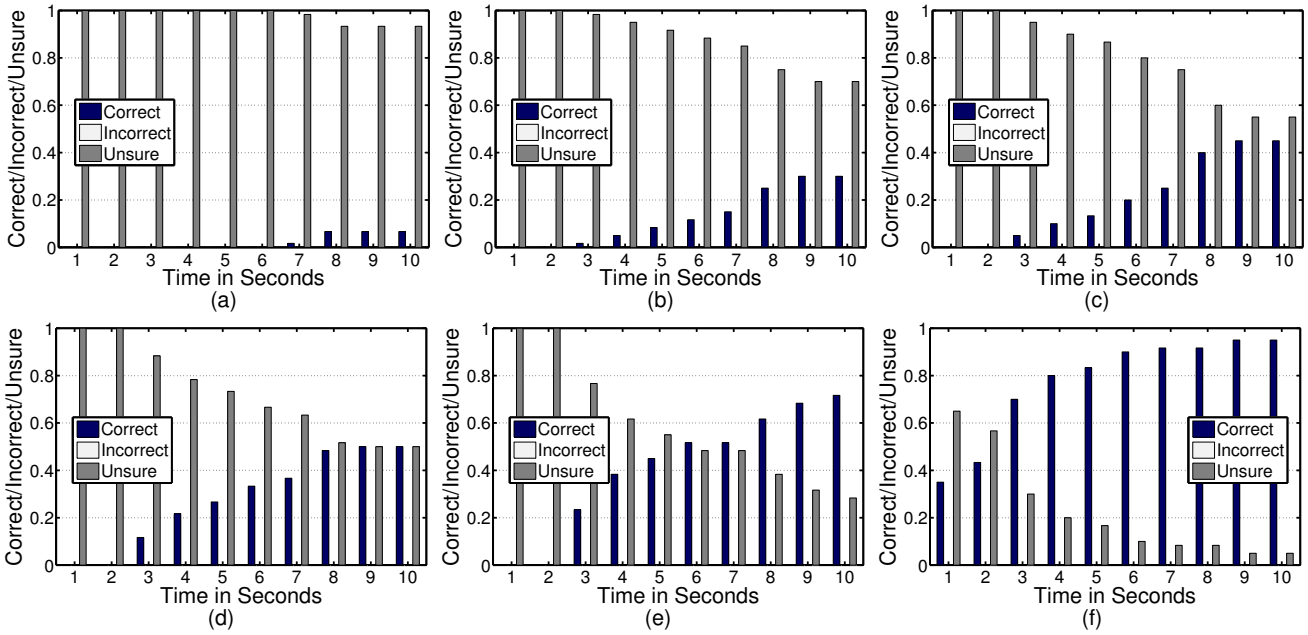

Figure 18: Recognition performance with motion alphabets and color fingerprints: (a) Walking or not only; (b) Walking direction only; (c) Walking direction with walking phase; (d) Walking direction with step phase and duration; (e) All motion alphabets including rotation; (f) All motion alphabets along with color fingerprints.

Figure 17(a). To obtain the step duration, we add a random error to the ground truth, following the distribution in Figure 13. For step direction, we do the same using the distribution from Figure 14. For determining the step phase, we add a random shift according to the step phase distribution in Figure 16 (a). The random shift varies from −9.1% of T_{step} to 25.3% of T_{step}. Now, since the error distributions were from a 12 member evaluation, the variance is likely to be smaller compared to a larger population. To account for such situations, we added additional errors to ensure the data is not optimistic.

The public videos were recorded under 3 different scenarios – near a busy area outside the student union during summer (referred to as the "union" video later), at the CS department cafe in the winter (referred to as "cafe" video) and at the entrance of a Target store in the winter (referred as "store" video). Figure 19 show example video frames. People moved in and out of the videos – so at any typical time instant, we observed between 3 to 10 in the view finder. However, for each of the videos, we computed motion fingerprints of all the people across time, and then compare against each other. For instance, the union video included 40 distinct individuals in 5 minutes, and we pretend as if all the 40 people were present at the same time. *InSight* is expected to be able to discriminate each individual accurately from these 40 individuals. The cafe and store had 15 people each due to less churn.

Results

Figure 20 reports results from the "union" video simulations. As a high level summary, *InSight* was able to recognize most of the user by combining motion and clothing. Specifically, since the recording was in summer, people wore colorful clothing, which by itself achieved 50% accuracy among 40 people. Expectedly, motion aided this discrimination, however, given that many users walked often, walking-or-not was not a major discriminator. Walking direction and step duration were helpful, but still not sufficient due to the high density of users (users mostly walked in two dominant direction, one towards the restaurants, and another towards dorms). However, when including walking phase, results improved significantly.

We zoom into the results here. Figure 20(a) shows the clothing confusion matrix across 40 people. Figures 20(b) and (c) plot the recognition performance over time, using motion alone followed by motion+clothing. Figure 20(d) shows the confusion matrix after combining motion and clothing and the similarity threshold set to 0.95. Using clothing alone, *InSight* recognizes 50% of individuals. Using motion alone, *InSight* recognizes 40% of individuals within 10 seconds; with combination of clothing and motion, the recognition performance rises up to 88% within 5 seconds and 90% within 8 seconds.

Figure 21 and 22 report on the results from the "cafe" and "store" video simulations. Since both scenarios were recorded in winter, a majority of the people were wearing dark shades (dominantly black and gray). Thus, in both cases, clothing offered fewer bits of information. However, motion compensated well, especially in the cafe where people walked, paused, turned, etc. With combined clothing and motion, *InSight* achieves 80% accuracy in 6 seconds, 93% in 8 seconds at the cafe, and 87% in 5 seconds, 93% in 10 seconds at the store.

The results above suggest that humans inherently exhibit diversity in their motion and clothing patterns that even low resolution feature vectors offer promise of identification. Where faces are permanent and high entropy fingerprints, these motion strings could serve as a useful alternative, when temporary "visual identification" is of interest, without revealing persistent identities.

6. POINTS OF DISCUSSION

We discuss a number of limitations and untapped opportunities with *InSight*.

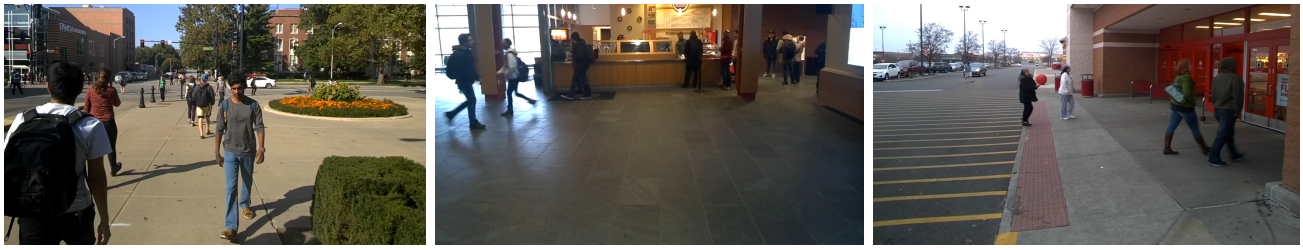

Figure 19: Example frames from (a) outside student union, (b) university cafe and (c) Target store.

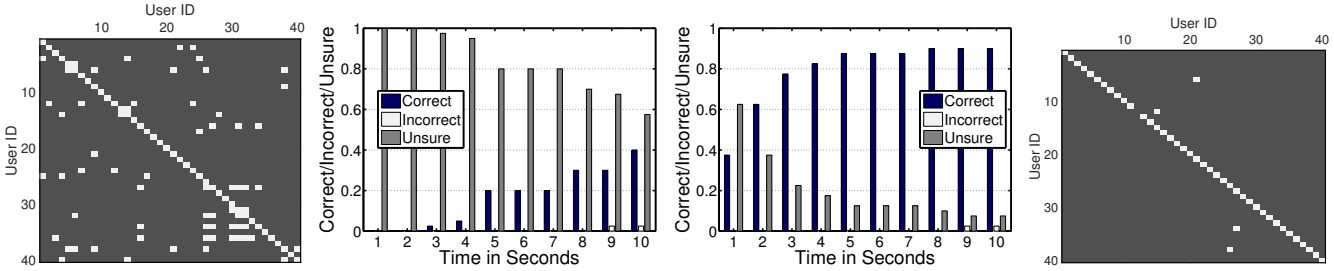

Figure 20: Union video Performance: (a) clothing only confusion matrix; Recognition over time using (b) motion only (c) motion + clothing (d) motion + clothing confusion matrix (after applying a threshold of 0.95 on similarity score).

Coping with Practical Hurdles. While we believe *InSight* is amenable to deployment in the real-world, it needs to be engineered, tested, and fine-tuned for various practical scenarios. Occlusions is perhaps the key limitation at this point. In a crowded environment, an individual is likely to be occluded by others, preventing the computer vision algorithm from carving out a precise bounding box for each person. This can inject confusion in the system, especially if the bounding boxes erroneously include different parts of two or more individuals. Coping with these complications is left to future work. Also, low lighting intensity in certain environments may affect visual clarity and fingerprinting. Further, people may change clothing – put on a scarf or take off their jackets – after their fingerprints have been registered in the database. Of course, motion fingerprints are a dependable fallback to cope with some of these situations, however, their efficacy in the wild remains to be seen.

Real Time Operation. In its current form, *InSight* is an offline system. Running this online will require substantial "heavy lifting" likely to be executed in the cloud [10] or cloudlets [27]. This paper's focus is towards demonstrating the core opportunity – the necessary engineering for an end to end system will depend on the application in question, and is a separate work altogether. Innovative research is yielding intelligent cloud-offloading techniques [10,27], designed explicitly for applications like *InSight*. In view of this, we believe real-time operation will be feasible over time, perhaps requiring porting *InSight* on MAUI-like programming frameworks [10].

Energy. The energy footprint of *InSight* may not be excessive. Motion sensors on smartphones need not consume much energy even after prolonged continuous sensing. The camera on the other hand can be activated only when the user desires to view annotations of the environment (e.g., when Alice wants to learn who a particular person is). This is true even for other applications such as privacy preserving pictures – the camera is again used only during the recording of a video or for taking a picture.

Incremental Deployment. If some users do not run *InSight* on their phones, the fingerprint matching process faces additional challenges. For instance, Alice may be looking at Bob in reality, and even though Bob is not running *InSight*, Bob's clothing fingerprint may match best with Chris (a registered user). To avoid such errors, *InSight* requires that the fingerprint matching threshold be very high, with the hope that motion-based matching would be triggered. Nonetheless, certain scenarios (such as weddings, funerals, uniformed school children) may still derail *InSight*. Perhaps an adaptive process is needed, where the *InSight* server identifies large similarities in clothing patterns and triggers motion based matching more aggressively.

Utilizing all bits of information. Upon receiving a video clip, *InSight* can extract clothing fingerprints for multiple people even if it is unable to map them to their identities. Over time, the repository of unnamed clothing fingerprints increase. Now, as individuals get identified through motion, it may be possible to resolve some of the other clothing fingerprints via the process of elimination. We envisage that such a form of inferencing would be possible due to the global view on the fingerprint data, available at the *InSight* server. We have not tapped this opportunity in this paper.

Peer-to-peer version of *InSight*. While we describe *InSight* as a cloud-based system, it is also possible to realize it over a peer-to-peer model. The sequence of operations, somewhat different from Figure 2, can be as follows. (1) Alice's glass takes a picture of X and broadcasts it, asking "who is the person in the picture"; simultaneously it starts recoding a video snippet. (2) Smartphones in the vicinity receive the image (over Bluetooth or WiFi Direct), extract the color fingerprint, and match it with their self color fingerprint (if they have one); simultaneously each of them activate their motion sensors. (3) These smartphones send their color fingerprint matching score along with their sensor data to Alice's glass. (4) Alice's phone computes the motion matching score, combines with the color score, and ultimately identifies X as Bob. (5) Alice's glass extracts Bob's color fingerprint from the video snippet and unicasts it to Bob's smartphone, which updates its own fingerprint. Of course, such a system assumes that smartphones are capable of executing the compute-heavy algorithms locally, perhaps difficult in today's platforms.

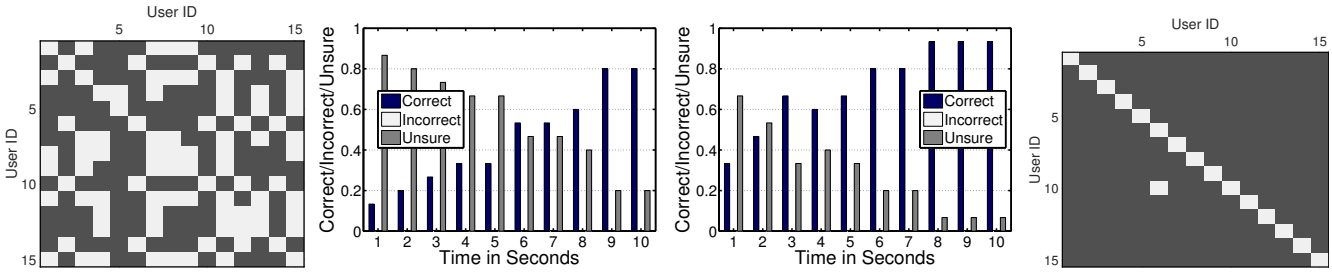

Figure 21: Cafe video simulation: (a) clothing only confusion matrix; Recognition over time using (b) motion only (c) motion + clothing (d) motion + clothing confusion matrix (after applying a threshold of 0.95 on similarity score).

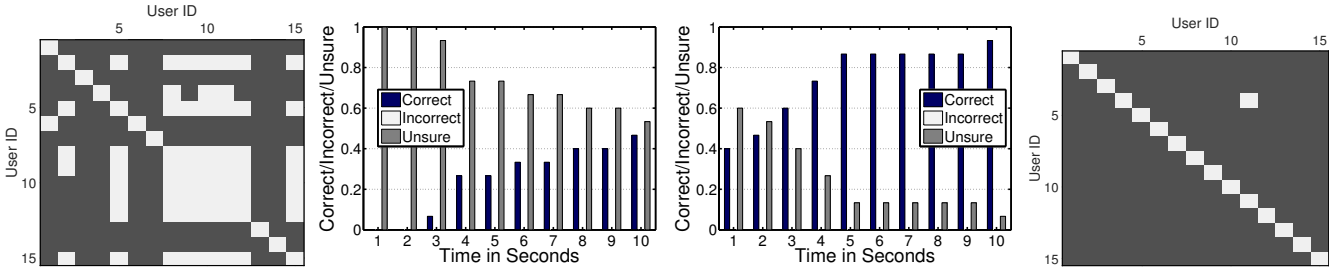

Figure 22: Store video simulation: (a) clothing only confusion matrix; Recognition over time using (b) motion only (c) motion + clothing (d) motion + clothing confusion matrix (after applying a threshold of 0.95 on similarity score).

7. RELATED WORK

There exist several works on activity recognition based on video [13, 26] and sensors [4]. TagSense [23] uses motion as an indication of whether a person is in the picture, but does not need to actually identify each individual. Face recognition and other visual bio-metrics are of course possible alternatives [16, 18, 34], but need the face to be visible (in addition to practical concerns on revealing a permanent identifier). *InSight*, on the other hand, temporarily fingerprints individuals, exposing only soft-biometrics of the user that cannot identify them later. We observe that *InSight* is different from gait analysis [16], used to "fingerprint" individuals. While gait analysis zooms into the intricacies of walking patterns (i.e., how one walks), we use far higher level motion alphabets capturing duration of walks, turns, pauses (none of which is permanent).

The specific applications we have discussed have received research attention in the recent past [3, 17, 20, 22, 25, 28]. Researchers have explored the possibility of looking at objects in a store, using wearable devices and radio-optical beacons [3] and/or RFID based techniques [28]. Such modes of communication are innovative and complementary to *InSight*. Qualcomm Vuforia [1] is a commercial Mobile AR SDK for object recognition and 3D object tracking. Videoguide [2] is a Vuforia app used to animate architecture work in Barcelona museum. Contrary to Vuforia which requires deployment in advance, *InSight* is a training-free system intended for humans.

Privacy preservation in the age of wearable cameras is also witnessing considerable research attention. Authors in [6] suggest a QR code pasted on people's clothing, as an expression of privacy preferences to surrounding cameras. Of course, the QR code may not necessarily be in view of the camera; reading from longer ranges is difficult; with human motion, reading QR codes is hard. PrivateEye [24] proposes to avoid recording objects by drawing a signature shape around it. *InSight*, on the other hand, does not require any form of instrumentation, except that the smartphone should run the app. Natural behavior of humans should exhibit adequate diversity for recognition, in turn useful for for inclusion, exclusion, or communication.

Our workshop paper on *InSight* [31] was an initial exploration into the possibility of using clothing colors as a temporary visual identifier. This paper builds on the workshop version in multiple fronts, including (1) the significant addition of motion information to the notion of visual fingerprinting, (2) a range of techniques to correlate motion from vision and device sensors, (3) the idea of expressing fingerprints as activity–strings and applying string matching algorithms on them, (4) a more complete evaluation through micro-benchmarks and offline evaluation, and finally (5) isolating the notion of visual fingerprinting and discussing its various applications in augmented reality, privacy preserving pictures, and indoor localization.

8. CONCLUSION

This paper pursues a hypothesis that motion patterns and clothing colors may pose as a human fingerprint, adequate to discriminate one individual from others. If successful, such a fingerprint could be effectively used towards human recognition or content announcement in the visual vicinity, and more broadly towards enabling human-centric augmented reality. Pivoted on this vision, we develop a proof of concept – InSight – which correlates motion fingerprint extracted from the video with those extracted from smartphone sensors. Clothing colors offer additional dimensions of correlation, boosting the confidence in recognition. We find promise in this direction, and are committed to building a fuller, real-time, system.

9. ACKNOWLEDGMENTS

We sincerely thank our shepherd Dr. Tam Vu and the anonymous reviewers for their valuable feedback. We are grateful to Google and to NSF for partially funding this research through grant CNS-1430064.

10. REFERENCES

[1] Qualcomm Vuforia. https://www.qualcomm.com/products/vuforia.

[2] Videoguide, Antoni Gaudi Modernist Museum in Barcelona. http://www.casabatllo.es/en/visit/videoguide.

[3] A. Ashok, M. Gruteser, N. Mandayam, J. Silva, M. Varga, and K. Dana. Challenge: mobile optical networks through visual MIMO. In *ACM MobiCom*, pages 105–112, 2010.

[4] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive Computing*, pages 1–17, 2004.

[5] S. T. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. *IEEE CVPR*, 2:1158 – 1163, 2005.

[6] C. Bo, G. Shen, J. Liu, X. Li, Y. Zhang, and F. Zhao. Privacy.tag: Privacy concern expressed and respected. In *ACM SenSys*, pages 163–176, 2014.

[7] K. W. Bowyer. Face recognition technology: security versus privacy. *IEEE Technology and Society Magazine*, 23(1):9–19, 2004.

[8] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1820–1833, 2011.

[9] C. O. Conaire, N. E. O'Connor, and A. F. Smeaton. An improved spatiogram similarity measure for robust object localisation. *IEEE ICASSP*, 1:1069–1072, 2007.

[10] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. MAUI: making smartphones last longer with code offload. In *MobiSys*, pages 49–62, 2010.

[11] P. Dollár. Piotr's Image and Video Matlab Toolbox (PMT). http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.

[12] P. Dollár, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. In *ECCV*, pages 645–659, 2012.

[13] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE VS-PETS*, pages 65–72, 2005.

[14] M. Eichner and V. Ferrari. Calvin upper-body detector. http://groups.inf.ed.ac.uk/calvin/calvin_upperbody_detector/.

[15] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *IEEE CVPR*, pages 1–8, 2008.

[16] J. Han and B. Bhanu. Individual recognition using gait energy image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(2):316–322, 2006.

[17] A. Henrysson and M. Ollila. UMAR: Ubiquitous mobile augmented reality. In *ACM MUM*, pages 41–45, 2004.

[18] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):4–20, 2004.

[19] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[20] A. Mayberry, P. Hu, B. Marlin, C. Salthouse, and D. Ganesan. iShadow: Design of a wearable, real-time mobile gaze tracker. In *MobiSys*, pages 82–94, 2014.

[21] P. Mistry and P. Maes. SixthSense: A wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches*, number 11, 2009.

[22] W. Piekarski and B. Thomas. ARQuake: the outdoor augmented reality gaming system. *Communications of the ACM*, 45(1):36–38, 2002.

[23] C. Qin, X. Bao, R. Roy Choudhury, and S. Nelakuditi. TagSense: A smartphone-based approach to automatic image tagging. In *ACM MobiSys*, pages 1–14, 2011.

[24] N. Raval, A. Srivastava, K. Lebeck, L. Cox, and A. Machanavajjhala. MarkIt: Privacy markers for protecting visual secrets. In *ACM UbiComp Adjunct*, pages 1289–1295, 2014.

[25] J. Rekimoto and Y. Ayatsuka. CyberCode: designing augmented reality environments with visual tags. In *ACM DARE*, pages 1–10, 2000.

[26] N. Robertson and I. Reid. A general method for human activity recognition in video. *Computer Vision and Image Understanding*, 104(2-3):232–248, 2006.

[27] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for VM-based cloudlets in mobile computing. 8(4):14–23, 2009.

[28] R. Tenmoku, M. Kanbara, and N. Yokoya. A wearable augmented reality system using positioning infrastructures and a pedometer. In *IEEE ISWC*, page 110, 2003.

[29] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE CVPR*, pages 586–591, 1991.

[30] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Real-time detection and tracking for augmented reality on mobile phones. *Visualization and Computer Graphics, IEEE Transactions on*, 16(3):355–368, 2010.

[31] H. Wang, X. Bao, R. Roy Choudhury, and S. Nelakuditi. Recognizing humans without face recognition. In *ACM HotMobile*, number 7, 2013.

[32] H. Wang, Z. Wang, G. Shen, F. Li, S. Han, and F. Zhao. Wheelloc: Enabling continuous location service on mobile phone for outdoor scenarios. In *IEEE INFOCOM*, pages 2733–2741, 2013.

[33] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE CVPR*, pages 1385–1392, 2011.

[34] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.