Digital Object Identifier: 10.1109/PERCOM.2016.7456511

URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7456511

Video Recognition using Ambient Light Sensors

Lorenz Schwittmann, Viktor Matkovic, Matthäus Wander and Torben Weis
Distributed Systems Group
University of Duisburg-Essen

***Abstract*—We present a method for recognizing a video that is playing on a TV screen by sampling the ambient light sensor of a user's smartphone. This improves situation awareness in pervasive systems because the phone can determine what the user is currently watching on TV. Our method works even if the phone has no direct line of sight to the TV screen, since ambient light reflected from walls is sufficient. Our evaluation shows that a 100% recognition ratio of the current TV channel is possible by sampling a sequence of 15 to 120 seconds length, depending on more or less favorable measuring conditions. In addition, we evaluated the recognition ratio when the user is watching video-on-demand, which exhibits a large set of possible videos. Recognizing professional YouTube videos resulted in a 92% recognition ratio; amateur videos were recognized correctly with 60% because these videos have fewer cuts. Our method focuses on detecting the time difference between video cuts because the light emitted by the screen changes instantly with most cuts and this is easily measurable with the ambient light sensor. Using the ambient light sensor instead of the camera greatly benefits energy consumption, bandwidth usage and raises less privacy concerns. Hence, it is feasible to run the measurement in the background for a longer time without draining the battery and without sending camera shots to a remote server for analysis.**

I. Introduction

Pervasive Computing Systems need to understand the context of users to assist them in a meaningful way. A large body of research has been published aiming at detecting activity and environment of a user based on sensors [1] embedded in mobile devices. In this paper we show that it is possible to detect which television show or movie a person is watching by utilizing the ambient light sensor of smartphones and other mobile devices. We demonstrate that this is possible from apps as well as from a web page that the user is currently looking at. This information can be used in various ways. For example, apps on the mobile device could offer users additional information about the current TV show because the app knows what the user is currently looking at. In addition it helps with situation awareness to understand whether the user is watching an action movie, a TV cooking show or just a commercial. In the case of a movie, he probably does not want to be disturbed, especially in the final minutes of the show. Hence, a pervasive computing system could mute the phone and refrain from playing sounds for every incoming chat message.

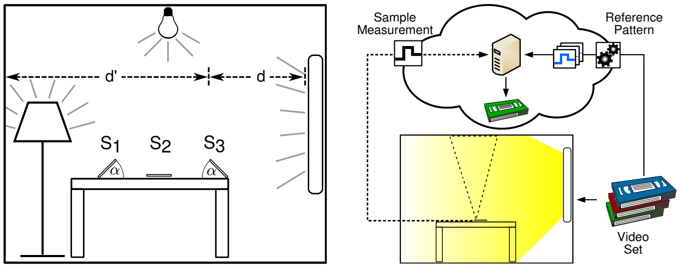
The cameras embedded in modern smartphones might seem like the better sensor to detect what a user is looking at. However, this has technical and privacy limitations. First, running the camera all the time and analyzing the video directly on the phone would result in a substantial battery drain. Sending the video snapshots over the network to analyze it in on a server could help with the battery problem, but most people would hesitate to send pictures of their current surrounding to the cloud for privacy reasons, especially if the gain of this privacy breach is just better situation awareness. Second, the camera is not guaranteed to be pointed directly at the screen, which renders video frame recognition infeasible.

Our approach works by analyzing the ambient light that can stem from reflections of the wall if a mobile device is not pointed at the TV screen. The light level is collected with the ambient light sensor, which is more effective for this task than the camera. The data retrieved by sampling the ambient light sensor has a much smaller footprint than a video stream. Hence, sending this data to a server does not result in a huge CPU or bandwidth usage. Furthermore the ambient light is less of a privacy problem compared to tapping the camera. Apps and web sites do not need special permissions to access the ambient light sensor on most platforms. This is important because each additional permission increases the likelihood that users reject an app for appearing privacy-invasive. A limitation of using the ambient light sensor is, however, the accuracy in bright ambient light. In a room flooded with sunlight the light sensor will not detect sufficient illumination delta, but since TV watching is a typical activity in dim light, our system works when it is most relevant. We have conducted tests with a large body of TV shows and YouTube Videos in various settings and the recognition ratio is surprisingly good considering that ambient light has little entropy compared to a video stream of a camera.

From a privacy point of view one could argue that a system inferring the currently running TV show without user consent violates the user's privacy. In this paper we analyze which frequency and sensor resolution are sufficient for our approach. A privacy-aware mobile device could use this information to limit the entropy of the ambient light sensor and offer better data only after the user gave permissions. The World Wide Web Consortium (W3C) is currently working on a standard API to expose sensors to web sites [2]. Hence, now is the right time to analyze which context data we can infer based on these sensors. Based on this it is possible to determine whether a sensor can be exposed to apps and web sites without permissions, or whether it should be limited for the sake of privacy.

II. Background

The human eye can only detect a certain range of the electromagnetic radiation, which we interpret as color with different intensities. Photometric units consider this imbalance of sensibilities and valuate the visible color spectrum through a weighted function to accommodate the human eye. Ambient light sensors use the photometric unit Lux, which is implicitly based on the standard 1931 CIE photopic luminosity weighted function $V(\Lambda)$ [3]. The luminosity function is analogous to



(a) Experimental setup. Distance and orientation to screen vary, $\alpha = 45°$.

(b) Data flow in overall system.

Fig. 1: Measurement and analysis setup.

the Y tristimulus value from the CIE XYZ color space which is the standard reference for other colorspaces such as sRGB.

Lux is the SI-unit for characterizing ambient light as perceived by humans and is used by light sensors in mobile devices, on which our method is based on. Light sensors approximate Lux values by applying an empirical function to physical measurements of photodiodes. The accuracy is limited as a trade-off with power consumption and manufacturing costs, and the error is non-linear [4]. Hence, a robust method must cope with inaccuracies between different sensor implementations.

III. System Model

We assume a user with a mobile device watches a video on a dedicated screen, e.g., television or computer screen. The user is indoors in dim light and the ambient light sensor in the phone records diffuse light emitted by the screen. The sensor is not obstructed, for example the mobile device is held by the user or lying on a table face up.

Fig. 1a shows our setup with different orientation scenarios that we consider in this paper: *facing user* (S_1), *resting on table* (S_2) and *facing screen* (S_3). Distance to screen (d) and distance to the back wall (d') are also variables in our setup. d' is relevant for scenario S_1 because there is no direct sight to the screen and the light sensor relies on diffuse light reflected from the wall. We do not consider ceiling height as a variable of our system but assume standard residential values.

IV. Approach

Our system is shown in Fig. 1b. The objective is to determine the video being played in the user's proximity from a set of known videos, e.g., currently running televion shows or on-demand video content. The ambient light sensor readings from the user's mobile device are transmitted to an external server. The server has reference patterns available, which have either been calculated from the RGB color information of the known video frames or measured in a reference setting. The server can identify the video played by comparing the Lux values retrieved from the mobile device with its reference patterns. Though these patterns do not necessarily match the mobile device's measured values exactly, we introduce a measure that works reliably despite varying light environments, different color reproduction of screens and different sensor calibrations.

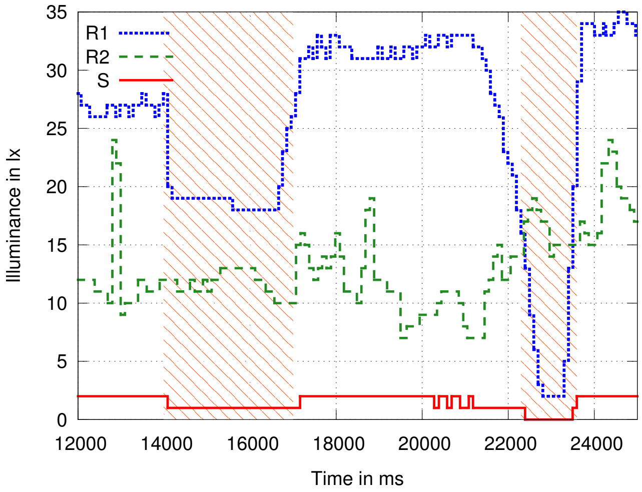


Fig. 2: Brightness comparison of two reference patterns (R1, R2) and one sample measurement (S).

A. *Observing Similarities*

The brightness values collected from a sensor measurement or calculated from video frames are discrete functions of time. Fig. 2 shows three ambient light sensor measurements. The measurements cover disparate ranges of illuminance, which is the result of different light environments. R1 and R2 have been collected under ideal conditions ($d = 1m$, S_3) with the phone facing the screen, while S has been collected in a more realistic setup with the phone resting on the table ($d = 2m$, S_2).

To illustrate the similarity of measurements, major changes of illuminance of S are indicated in the hatched area in Fig. 2. As we can see, S and R1 behave alike; whenever S shows a variance, R1 does so too – although on a higher magnitude. This is not the case for S and R2. Instead there are some hints indicating different videos caused these measurements: neither causes any of R2's significant peaks a rise of S nor has S's drop at 22 300 ms any influence on R2. Based on this, one can conclude—correctly—that S and R1 represent the same video.

B. *Correlation Analysis*

Given the above observations, we will now formalize illuminance correlations. We define a measurement $A \subset \mathbb{N} \times \mathbb{N}$ as a set of tuples containing timestamps and Lux values. For two measurements A, B we define t_i as the duplicate free ordered sequence of points in time where a value exists for either A or B, i.e.

$$t_i := \begin{cases} \inf\{x_1 | (x_1, x_2) \in (A \cup B)\} & \text{if } i = 1 \\ \inf\{x_1 | (x_1, x_2) \in (A \cup B) \wedge x_1 > t_{i-1}\} & \text{if } i > 1 \end{cases}$$

The samples do not have to share data points at the same points in time and as the example in Fig. 2 indicates, this is rather infrequent in reality. However, to compare both measurements we have to interpolate values between data points. We refrained from using linear interpolation because this led to poor identification rates in preliminary evaluations. The reason for this is the ambient light sensor resolution. Consider the intervall between 14 000 ms and 17 100 ms in Fig. 2: S would rise with linear interpolation, while the illuminance remains relatively constant for R1. Correlating a rising with a constant interval would lead to a weaker correlation coefficient than comparing two equally formed functions. Instead we interpolate values between data points by using a step function, i.e. we assume that between data points values remain constant. Formally, we define $f_X(u)$ as the illuminance of measurement X at time u as

$$\begin{aligned} f_X(u) &:= y_2 | (y_1, y_2) \in X \land (y_1 = g_X(u)) \\ g_X(u) &:= \inf\{x_1 | (x_1, x_2) \in X \land x_1 \geq u\}. \end{aligned}$$

Using these definitions, we apply Pearson's weighted correlation coefficient to determine the similarity of measurements. The fundamental idea is to correlate $f_A(t_i)$ with $f_B(t_i)$ for every element in t_i weighted with the time in between. The weighted average $\mathrm{m}(X,t)$ of a measurement X and the weighted covariance $\mathrm{cov}(A,B,t)$ are defined by

$$\begin{aligned} \mathrm{m}(X,t) &:= \frac{\sum_i (t_{i+1} - t_i) f_X(t_i)}{\sum_i (t_{i+1} - t_i)} \\ \mathrm{cov}(A,B,t) &:= \frac{1}{\sum_i (t_{i+1} - t_i)} \sum_i \Big((t_{i+1} - t_i) \\ &\qquad (f_A(t_i) - \mathrm{m}(A,t))(f_B(t_i) - \mathrm{m}(B,t)) \Big). \end{aligned}$$

Finally the weighted correlation coefficient $\mathrm{corr}(A,B,t)$ is defined by

$$\mathrm{corr}(A,B,t) := \frac{\mathrm{cov}(A,B,t)}{\sqrt{\mathrm{cov}(A,A,t) \cdot \mathrm{cov}(B,B,t)}}.$$

A potential source of interference in the above method is a time offset of the measurements. Even if the mobile device and reference measurement are recorded at the same time, they might still be shifted by a constant amount of time due to an imprecise clock on the mobile device, network latency or content propagation delay. To compensate for such a shift, we probe various offsets $o \pm 2000$ ms with 20 ms step width and choose the o with the maximum correlation coefficient.

V. Implementation

The implementation of our approach consists of the illuminance measurement on the mobile device and a server-side analysis. We have implemented the measurement part as website for Android devices and as native app for Windows Phone 8.1.

A. Client-side Measurement

The website reads the light sensor via the *Ambient Light Events* API, which is work in progress by the W3C [2]. As of today, Firefox is the only browser that already implements the draft specification. Our website accesses the JavaScript API by registering for *devicelight* events. The event is fired upon registration and whenever the light level changes. Our implementation sends the collected Lux values together with a timestamp to our web server for correlation analysis. We tested the website successfully with Firefox for Android 40.0.3 on Nexus 5, Nexus 7 and Samsung GT I9023.

Firefox is not available for Windows Phone, thus we implemented a native Windows Phone app with the event-based *Windows.Devices.Sensors.LightSensor* API. The implementation is analogous to the website and works succesfully on a Lumia 520.

We could not test our approach on iOS devices, because the ambient light sensor is currently not exposed to iOS applications (cf. Section VIII).

B. Server-side Reference Patterns

Our approach requires video reference patterns for comparison with our measured light values. In this section we present an efficient approach to acquire the reference patterns analytically without resorting to manual measurements.

Given an sRGB video frame, we can convert the sRGB color spectrum to the CIE XYZ color space (see Section II) by linear transformation. Y in this case represents the illuminance in the XYZ color space. We convert an sRGB value $\vec{x}_{RGB}$ to Y as follows:

$$h(\vec{x}_{RGB}) = \vec{x}_{RGB} \cdot \begin{pmatrix} 0.2126 \\ 0.7152 \\ 0.0722 \end{pmatrix} = Y$$

We define a video frame I to be a matrix comprised by $M \times N$ pixels, each with an sRGB color value. We then calculate the average luminance for an RGB encoded video frame using h:

$$f(I^{M \times N}) = \frac{1}{M \cdot N} \sum_{x=0}^{M} \sum_{y=0}^{N} h(I_{x,y})$$

VI. Evaluation

We evaluated our approach in various scenarios. Unless otherwise noted, the samples were recorded with a Nexus 5 on Android 5.1.1 bearing an APDS-9930 ambient light sensor.

A. TV Channel Recognition

To evaluate our method we selected 20 real world TV clips representing 20 different TV channels from 7 different television genres (advertisement, concerts, news, series, sports, talk shows, traditional animation). Each video was clipped to 300 seconds and measured with $d = 2m, S_1$ (sample) and $d = 1m, S_3$ (reference video).

For every sample measurement we calculated the correlation coefficient with every reference measurement. For example, Fig. 3 shows correlations between one advertisement and every reference measurement. The sample correlated highly with the corresponding reference measurement and can be clearly told apart from other candidates. Even videos of the same genre—advertisements (**AD**) in this case—have a significantly lower correlation coefficient. We conclude that there are no genre-specific illuminance changes.

The confidence in the recognized video can be expressed as the distance between the corresponding reference measurement

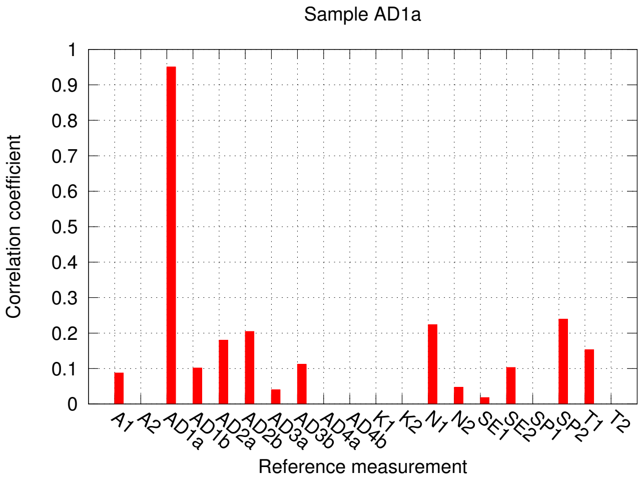


Fig. 3: Correlation to various reference measurements.

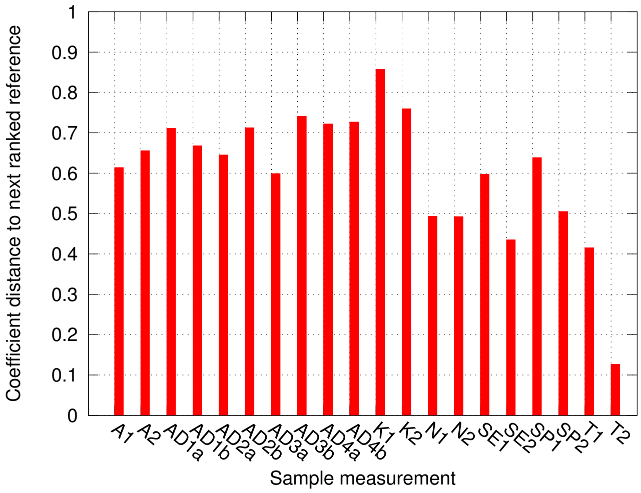


Fig. 4: Distance to next ranked reference measurement for every sample measurement.

(0.95) and the next ranked reference measurement (0.23). The higher this value, the lower the probability of a mismatch. To give an overview for the whole video set, we display this difference for every sample measurement in Fig. 4. All videos have been recognized correctly. The lowest confidence value (**T2**) originates from a political talk show featuring primarily frontal shots of the discussants. This lack of tracking shots or cuts complicates recognition, showing the limit of our method. Yet we were able to correctly identify every video in this set.

In general given the two highest coefficients c_1, c_2 we suggest using their difference as part of a metric for confidence in the recognized video: If the highest correlation coefficient is higher than a certain threshold ($c_1 > X$) and the next ranked video has a correlation coefficient significantly below this ($c_1 - c_2 > Y$) we consider a video to be recognized correctly. According to our evaluation $X = 0.7$ and $Y = 0.2$ are decent empirical thresholds.

	Q_1	Q_2	Q_3
YouTube Popular	7	27	46
YouTube Professional	23	32	43
TV	29	43	49

TABLE I: Distribution of sensor readings per minute of analyzed video sets.

B. *YouTube Recognition*

Video on demand services provide a much larger set of potential videos than regular TV. To evaluate video matching abilities on this scale we composed a set of videos by crawling YouTube.

1) Popular Videos: We downloaded 1526 unique videos, which were categorized as most popular YouTube videos in 81 regions. The record length is 60 seconds in this analysis, we thus omitted videos shorter than 60 seconds, which yielded 1180 video clips. For each video, we recorded a reference measurement ($d = 1\text{m}, S_3$) and a sample measurement ($d = 2\text{m}, S_1$).

60% of the video clips could be identified correctly. Compared to the previous scenario the recognition ratio is rather low. Spot-checking the set revealed this is due to seldom changes in illuminance caused by certain videos types characteristic for YouTube: *a*) Freeze frame videos, i.e. audio only *b*) Freeze frames with occasional text fade-in *c*) Single person speaking to a fixed camera. Since most productions are conducted by non-professionals, cuts are rarely used in the remaining cases.

2) Professional Productions: To evaluate professional video on demand content, we downloaded 200 videos from channels run by public broadcasting organizations, from which 149 remained after filtering for a length of at least 60 seconds. Reference and sample measurements were conducted with the same parameters and in the same setups as before.

This time 92% of the videos could be identified correctly. One could argue that this higher ratio occurs due to choosing a smaller set of videos. However, reducing the previous YouTube video set to 149 randomly chosen clips yielded in a recognition ratio of just 69% and therefore contradicts this possibility. This stresses the suitability of our method for professional real world productions as they appear on charged video on demand services.

Table I shows the distribution of data points in quartiles for each video set. Although the median (Q_2) does not differ significantly for professional and popular videos, 25% of popular productions have just 7 or less data points. This corresponds to recognition ratios, which is above 50% for all video sets but below 75% for popular YouTube videos. We conclude that a sufficient amount of illuminance changes is crucial for a successful identification.

C. *Environment*

We studied how different environmental settings such as distance and orientation towards screen and ambient light affect video recognition.

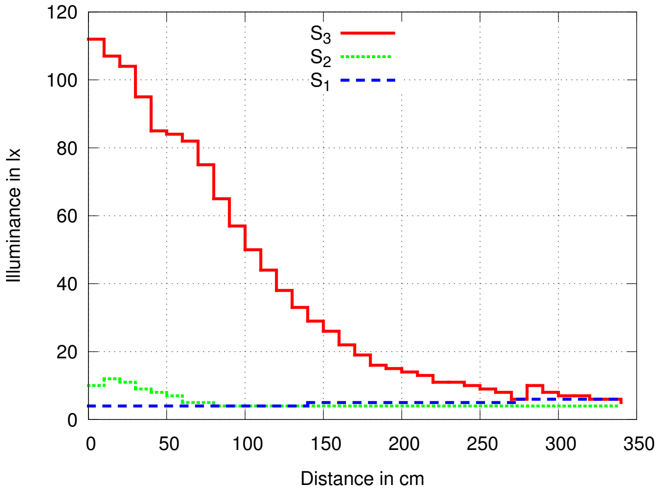


Fig. 5: Recorded illuminance of a white screen depending on distance.

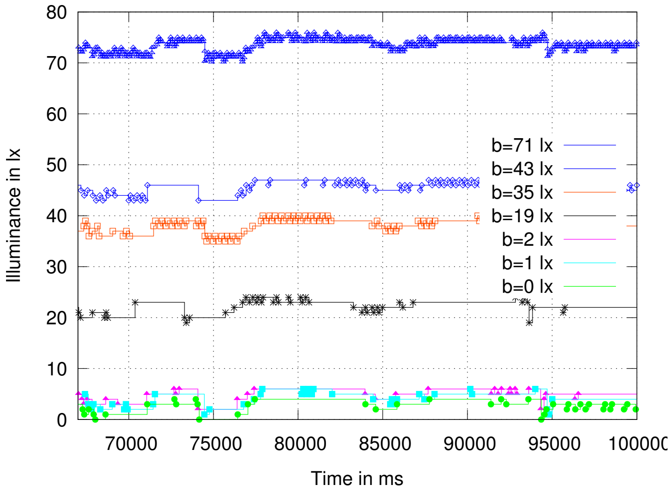


Fig. 6: Impact of environmental light on sensor readings.

1) Range and Orientation: Using the video set and its reference measurement from Section VI-A we performed 9 series of measurements by combining each scenario described in Section III with distances from $d = 1$m to $d = 3$m.

Although lower illuminance was measured in less favorable conditions (cf. Fig. 2) all videos could be recognized. However, we discovered that distance to screen does not necessarily have a negative impact on recorded illuminance. In scenario S_1 our measurement did yield better results at $d = 3$m than $d = 2$m. To understand this effect, we recorded illuminance of a white screen depending on scenario and distance to screen. In Fig. 5, for S_2 and S_3 illuminance roughly decreases as distance increases. For S_1, this effect is reversed: the higher the distance, the more illuminance has been recorded. Though this happens on a low scale, it has an impact considering that some measurements only consist of 0 and 1 lx. Since such measurements could be recognized correctly (e.g. **T1** in Section VI-A) the measured values are not noise but caused by actual video frames. We conclude that this effect is caused by the white wall situated opposite of the screen. As d' decreases, more reflected light is measured.

2) Light Environment: So far we evaluated our method in an almost dark room at night. Although this is a plausible scenario, being able to recognize videos in a half-light environment increases the practical application of our approach. We therefore evaluated how our method performs in a lit room. The basis for our comparison is the video set and its reference measurements from Section VI-A. All samples were performed with $d = 2m$ and in orientation scenario S_2. For each measurement series we increased the ambient light baseline (i.e. with the screen turned off) from $b = 0$ lx to $b = 71$ lx by turning on an additional lamp. The maximum lighting $b = 71$ lx surpasses common living room conditions, which are at 50 lx according to [5].

Fig. 6 shows the illuminance recorded on one sample video in different light environments. Although environmental ambient light causes sensor readings to be shifted up, relatively they remain nearly identical. This is also reflected in the recognition ratios: apart from one video, which was mismatched at $b = 1$ lx and $b = 19$ lx, every video was recognized correctly. We therefore reduced the sampling time to check if there is any effect at all. The results can be seen in Fig. 7. Even in a lit room at night, there is no significant impact on the recognition ratio. We conclude that our method is also applicable if moderate ambient light is present.

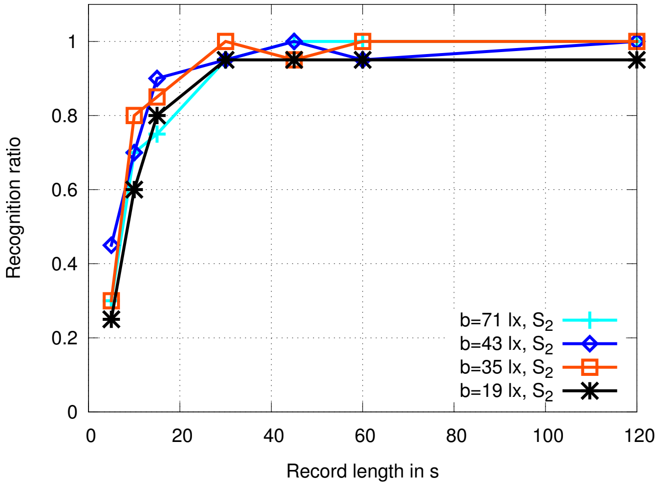


Fig. 7: Recognition ratio depending on light condition and sample time.

This conclusion is supported by the fundamental properties of light. Physically seen, light and therefore also its illuminance is additive. Even with a very high baseline caused by, e.g., direct sunlight, the diffuse light emitted by the screen adds to the total. However, our method is limited by sensor accuracy; while most sensors detect reliably a variation of 3 lx in dim light, they fail to detect the same variation at a baseline of 100 000 lx.

D. Record Length

We now analyze the impact of the length of the measured sample on the recognition ratio. The basis for our comparison is the video set of Section VI-A with $d = 2m$ in all orientation

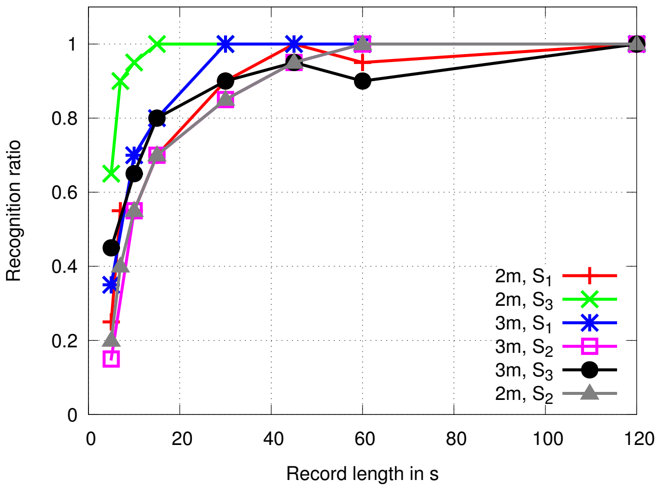


Fig. 8: Recognition ratio depending on setup and sample time.

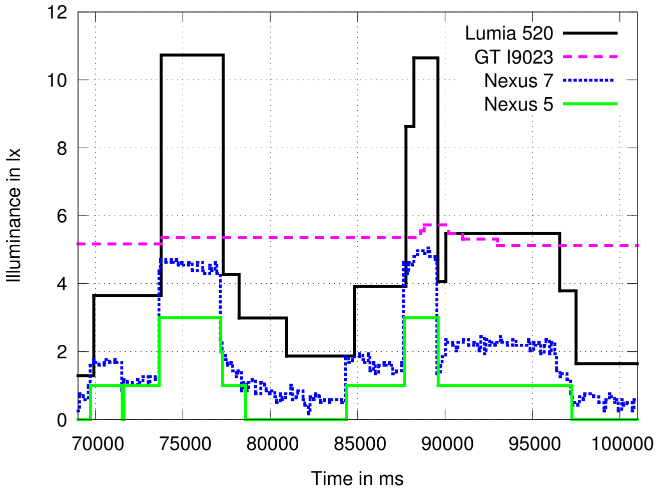


Fig. 9: Recorded illuminance by various devices.

scenarios (sample measurements) and $d = 1$ m, S_3 (reference measurement). As stated in Section VI-C1, all clips could be recognized under these conditions. We reduced the length down to a minimum of 5 seconds and observed the change in this rate.

As shown in Fig. 8, the minimum time required to achieve a perfect match for every video is 60 seconds for S_2 and 120 seconds for S_1. For S_3, 100% recognition ratio is achieved after 15 seconds of record length.

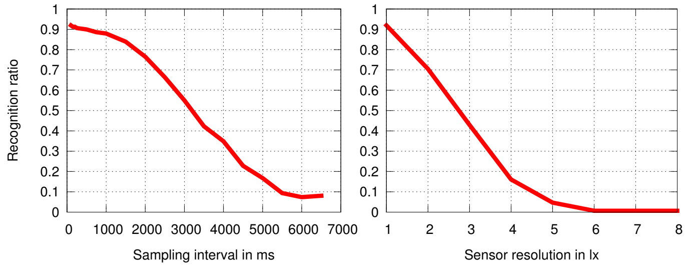
Although the sensor orientation differs in S_2 and S_1, the recognition ratios are comparable for small record lengths. We conclude that in this case diffuse light reflected from the walls of the room is the primary source of information; only a small fraction of direct light actually reaches the sensor. This is different for S_3, where the mobile device is pitched towards the screen. The sensor records direct light from the screen, which results in a higher resolution and therefore a higher recognition ratio. This suggests that a lack of record length can be compensated by increasing the sensor resolution.

E. Devices

Ambient light sensors approximate illuminance using empirical functions (cf. Section II). To rule out a sensor or device bias, we conducted measurements with a device set consisting of Nexus 5, Samsung GT I9023, Nexus 7 (2013) and Lumia 520. To provide an overview of these devices, Fig. 9 shows illuminance recorded by these devices in scenario $d = 2m, S_3$.

As we can see, the devices' sensor readings vary both in magnitude and frequency. The frequency is the result of sensor quality and sampling frequency of the operating system. The deviation of magnitude is caused by sensor-specific calibrations and accuracy as mentioned in Section II. Since our method correlates changes of illuminance, the absolute sensor reading is of lesser importance. However, there is a lower bound: If the sensor resolution is too low to detect changes in illuminance, there will be too few values for the correlation analysis and our method will fail.

An example for this effect is visible in the interval from 70 000 ms to 80 000 ms in Fig. 9. All devices except GT I9023 show a peak in illuminance followed by a local minimum. Since the effect is not limited to this particular measurement excerpt, it also has an impact on the recognition ratio: Given the video set from Section VI-A the readings from Nexus 5 and Nexus 7 yielded a 100% recognition ratio. Measurements from Lumia 520 caused a ratio of 90% and I9023 of 60%.



(a) Sensor sampling rate (b) Sensor granularity

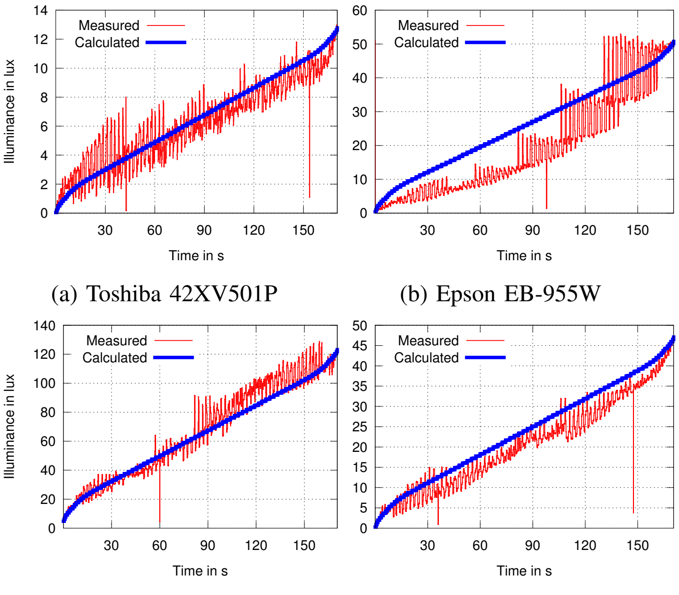
Fig. 10: Effect of sensor quality on recognition ratio.

One explanation for this variance in sensor performance is the age. GT I9023 was released 2011 while all other devices were released 2013 and might therefore profit from technology advancements.

F. Sensor Limits

As we identified in the previous section, the sensor resolution is crucial for our approach. To analyze this dependency in a systematic manner, we truncated the sensor readings collected in Section VI-B2 and observed the impact on the recognition ratio.

1) Sampling Rate: First, we truncated the sampling rate, i.e. the minimum time between two sensor readings. Fig. 10a shows that reducing the sampling rate has a non-linear negative impact on the recognition ratio. Although recognition decreases sharply for sampling intervals $> 2000ms$, we do not consider this to be problematic since it has little real world impact: All ambient light sensors examined in this paper provide sampling rates of $\ll 1000$ ms, which yields recognition ratios of at least 88%.



(a) Toshiba 42XV501P

(b) Epson EB-955W

(c) Samsung UE65F6470

(d) Toshiba 47WLG66

Fig. 11: Analytic vs. measured illuminance of colors.

2) Sensor Granularity: We emulated a low sensor resolution by rounding measured illuminance down to multiples of the emulated granularity. The impact on recognition ratio can be seen in Fig. 10b. As before, this has a negative impact on recognition ratio, though the effect is stronger: the recognition ratio falls below 43% when truncating the sensor resolution to $\geq$ 3 lx. The reason for this is that ambient light delta is in this order of magnitude, as discussed in Section VI-C.

G. Reference Illuminance

One way for obtaining illuminance values is to analyze videos (cf. Section V-B). To evaluate suitability of this approach, we compared analytically derived illuminance with actual measured illuminance. For this, we evenly sampled the RGB color space and sorted the resulting colors by their calculated illuminance. We displayed these colors on three screens and one projector and recorded resulting illuminance on a Nexus 7 (2013). We chose this tablet since it bears the best sensor in our device set. The results are presented in Fig. 11. Since our method implicitly normalizes input sequences only relative changes are relevant. We therefore linearly transformed the calculated illuminance to fit the measured data.

As we can see, there are deviations from our calculated values depending on the TV screen. This occurs due to device-specific color rendering. To evaluate whether our method is capable to succeed despite color deviations we used the video set from Section VI-A and recorded one minute long measurements. These measurements were compared with analytically derived values.

The recognition ratios were 80% for 11a and 11d, 95% for 11b and 100% for 11c. These ratios imply that recognition ratio does not correlate with color accuracy: the projector in 11b has the largest color deviation but ranks second in recognition ratio. Instead, illuminance is determining. The brightest display devices have the highest recognition ratios. This is reasonable because for a constant sensor resolution a larger spectrum of illuminance can be represented by more sensor values leading to a more accurate representation.

We conclude that analytical derived illuminance provides a suitable estimation of actual measured illuminance.

VII. Feasibility

So far we have evaluated our method in various scenarios. In this section we will discuss real world challenges for our approach.

A. Automation

Since measuring reference patters for all videos is a cumbersome task, the server can analyze each video entirely in software and map it to a brightness scale (cf. Section V-B). By abolishing the need for physical measurements, this increases the mass scale practicability of our method.

To compare the quality of such calculated reference patterns we used the second YouTube video set with its sample measurements from Section VI-B. We applied our method on these measurements and the calculated patterns. This time a recognition ratio of 92% could be reached, which is identical to the recognition ratio yielded by a comparison with reference measurements. This suggests that both reference measurements and calculated reference patterns are equally suited for our approach.

B. Network Load

Compared to streaming camera footage to a server, our approach requires a significant lower amount of data. Since a video is typically played at 25 FPS a sensor sampling rate below 40 ms does not yield in better results. If we assume this as an upper bound and represent a sensor reading with two bytes, our method will require at most 3,000 bytes for one minute of video footage. Although illuminance *can* change for every frame this is seldom in real world settings (cf. Fig. 2 and Fig. 6). We therefore conclude that our approach is feasible with respect to network load.

C. Server-side Load

Given a sensor measurement, the server has to a) maintain its reference set with currently running tv shows and b) compute correlation with this set.

The first step requires the server to sample all TV channels at a specific rate and calculate illuminance using the method shown in Section V-B. As shown in Fig. 10a, it is sufficient to sample every 250ms to achieve a high recognition ratio. Also, a video stream with low resolution suffices for illuminance determination easing CPU requirements. Our prototype is implemented in a browser environment and requires 10%-20% CPU usage on consumer grade hardware for real-time analysis. We therefore deem it feasible to analyze a large amount of TV channels using dedicated servers.

For the second step, our offset approach consumes most of the CPU time since it increases the load by a factor of 100. Without this, our unoptimized proof of concept Python prototype requires 2 ms on a commodity laptop for correlating two videos from the set used in Section VI-B2. However, our offset approach can be implemented in an efficient manner by finding the maximum correlation coefficient using a heuristic. Furthermore, once the content propagation delay has been determined for a specific user this information can be reused. In this case, determining the offset boils down to compensating jitter.

VIII. Privacy Considerations

Our approach utilizes the ambient light sensor whose data is less privacy-invasive than, e.g., camera or microphone recordings. Nevertheless, learning about the user's context bears the risk of violating his or her privacy, in particular when collecting data without consent or when using the data in ways unexpected by the user. In this section, we discuss countermeasures and survey the permissions required for reading the ambient light sensor on various platforms.

A. Truncation

The purpose of the ambient light sensor is to dim the screen to ensure visibility in bright daylight while conserving battery and avoiding eye fatigue at night. Adjusting the screen brightness is handled by the operating system. The sensor Lux values are exposed to applications to give them the opportunity of adjusting the appearance of the user interface, e.g., switch to a darker theme. The operating system could limit the potential for context inference without consent by truncating the sensor readings.

We have shown in Section VI-F that our approach is still feasible with a sensor sampling interval of 1 seconds. Truncating the sensor sampling interval to 5 seconds makes our approach infeasible, but also makes the sensor readings useless for mobile devices, whose light conditions change quickly when taking the device out of the pocket or opening the protective cover. Another dimension for truncation is the sensor resolution. Our approach becomes infeasible when the sensor readings are rounded to levels of 5 lx, yet this resolution is more than adequate for adjusting the screen brightness of user interfaces. In fact, rounding to three or five different illuminance levels should suffice for most applications. The W3C Media Queries [6] for example provide a three-level (*dim, normal, washed*) ambient light reading to applications. The additional advantage of a qualitative classification is that devices can use different illuminance thresholds to account for technological characteristics—e-ink displays are for example better readable in sunlight than LCDs.

B. Permissions

Running on Firefox for Android, our web implementation does not need to request user permission when accessing the ambient light API. The light event is fired on the active tab only and not on background tabs, iframes or when the screen is turned off. This behavior has been specified in the API draft [2], which also recommends to consider an indication to the user when the sensor is active and to allow turning the sensor off. Enabling the user to notice and control the sensor readings is a worthy idea, but not trivial to achieve given the magnitude of active sensors besides the light sensor on today's mobile devices and the limited space on the screen for visual indicators. Requesting user permission before sensor access is not intended in the API draft, unlike, e.g., in the geolocation web API [7].

Sensor data can be collected in background when implemented as native app. Android exposes the ambient light sensor as Lux value since Android 1.5 [8]. Background sensing is possible even when the screen is locked or turned off. There is no permission required and no user indication of an active sensor. A similar case is Windows Phone 8.1, which allows apps to access the sensed Lux value without permission or user indication. Background sensing can be implemented via the *DeviceUseTrigger* API.

iOS does not provide the ambient light sensor readings to applications. There is an unsupported, private API in iOS, but third-party apps are denied access to private APIs in the operating system.

IX. Related Work

Our approach allows to infer context about the high-level activity of the user (watching TV) from a low-level sensor reading (ambient light sensor). The fundamental difference to image-based video fingerprinting [9]–[11] or audio fingerprinting techniques [12], [13] is that we do not need to use the camera or microphone, which has advantages in terms of power consumption, CPU and network usage, usability and user privacy.

Utilization of the ambient light sensor for context-aware computing has had little attention in the literature compared to other sensors. Ravi and Iftode [14] suggested to use the light sensor for fingerprinting room lighting conditions for the purpose of indoor localization. Li et al. [15] suggested to use visible light communication for indoor localization: a light sensor attached to a smartphone receives location beacons, which are broadcasted by modulating white light-emitting diodes (LED) in the room. Visible light communication allows for accurate sub-meter localization but the LED modulation requires a more sophisticated light sensor with high-frequency sampling.

Several researchers suggested to extract ambient color and illuminance from camera images for the purpose of indoor localization [16]–[19]. The illuminance feature could be gathered with the ambient light sensor in today's mobile devices. Color sensing is available on a few devices such as Samsung Galaxy S3 with CM36651 sensor, though most devices measure the illuminance only.

Spreitzer [20] demonstrated a potential side channel attack by exploiting the ambient light sensor: after a training phase, a malicious app could use the ambient light reading to infer information about probable keystrokes, e.g., to guess personal identification numbers. This emphasizes our demand for sensor truncation unless the user consents to ambient light sensing.

Some of these approaches employ machine learning techniques while we compare raw signals. In order to utilize machine learning effectively, a TV channel would have to be identifiable by a certain learnable feature. However, our evaluation in Section VI-A does not show evidence for a genre-specific or channel-specific correlation. Whenever a new show starts on the channel, this invalidates any previously trained feature because the new video frames are unrelated to the previous ones. We thus conclude that machine learning techniques do not match our use case well.

X. Conclusion

We presented an approach for recognizing a video playing in the user's proximity by analyzing the ambient light sensor readings of mobile devices. Our method correlates the characteristic video flickering with reference illuminance values from a set of known videos. These reference patterns can either be obtained by a reference measurement or by calculating them from a video stream. We have tested our approach successfully on several mobile devices in a typical living room scenario. Our evaluation shows that the recognition works well for television channels (100%) and professional YouTube content (92%), yet moderately for amateur YouTube content (60%). The discrepancy is caused by the amount of video cuts or sudden light changes, which are more frequent in professional video productions.

Although our method works best at night, room lights do not have a detrimental effect on the recognition ratio as long as the mobile device is not pointed directly into the light source. We argue that existing ambient light in general can be compensated by a higher sensor resolution. Vital parameters for successful video recognition are the measured record length, sensor resolution and sampling rate. A recognition ratio of about 90% is achieved with 7 seconds samples when having a direct sight from sensor to screen and with 30 seconds samples when pointing the sensor away from the screen (e.g. holding the mobile device). Sensors with higher resolution achieve better recognition ratios with a given record length, where a resolution of 1 lx recognizes 90% of professional videos. The sampling interval is suitable on all tested devices and can be even reduced to 1 second to save battery power without severely degrading the recognition ratio.

The Lux values collected by the ambient light sensor are exposed to applications on Android, Windows Phone and Firefox for mobile devices. Permission for accessing the sensor values is not required, which bears the privacy risk of leaking the user's context without prior consent. Our recommendation to system implementers is to truncate the sensor Lux values to predefined light levels, which suffice to adjust the user interface subject to the ambient light environment. Applications should be required to ask for permission when exact Lux values are needed, e.g., to support the user with contextual information derived from our video recognition approach.

For future work, we suggest to apply our approach to smartwatches or other wearables that integrate even better in the user's environment than smartphones and tablets. The required record length could be decreased and the recognition ratio further increased with improved ambient light sensors. High-resolution RGBW sensors could allow sensing at daylight or recognizing slight changes of illuminance, e.g., in amateur videos.

References

[1] M. Conti, S. K. Das, C. Bisdikian, M. Kumar, L. M. Ni, A. Passarella, G. Roussos, G. Tröster, G. Tsudik, and F. Zambonelli, "Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber–physical convergence," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 2 – 21, 2012.

[2] D. Turner and A. Kostiainen, "Ambient light events," Sep. 2015, W3C Working Draft. [Online]. Available: http://www.w3.org/TR/ambient-light/

[3] Commission Internationale de L'Eclairage, *The Basis of Physical Photometry, 2nd ed. (CIE 018.2-1983)*, Jan. 1983.

[4] C. Schuss, T. Leikanger, B. Eichberger, and T. Rahkonen, "Efficient use of solar chargers with the help of ambient light sensors on smartphones," in *Open Innovations Association (FRUCT16), 2014 16th Conference of.* IEEE, 2014, pp. 79–85.

[5] A. Pears, "Chapter 7: Appliance technologies and scope for emission reduction," in *Strategic Study of Household Energy and Greenhouse Issues*. Australian Greenhouse Office, 1998, p. 61.

[6] F. Rivoal and T. Atkins Jr., "Media queries level 4," May 2015, W3C Editor's Draft. [Online]. Available: https://drafts.csswg.org/mediaqueries/

[7] A. Popescu, "Geolocation API specification," Jul. 2014, W3C Editor's Draft. [Online]. Available: http://dev.w3.org/geo/api/spec-source.html

[8] Sensors overview. Google Inc. [Online]. Available: http://developer.android.com/guide/topics/sensors/sensors_overview.html

[9] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *Recent Advances in Visual Information Systems*. Springer, 2002, pp. 117–128.

[10] D. Kundur and K. Karthik, "Video fingerprinting and encryption principles for digital rights management," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 918–932, 2004.

[11] S. Lee and C. D. Yoo, "Robust video fingerprinting for content-based video identification," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 7, pp. 983–988, 2008.

[12] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system." in *ISMIR*, vol. 2002, 2002, pp. 107–115.

[13] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, "A review of audio fingerprinting," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 41, no. 3, pp. 271–284, 2005.

[14] N. Ravi and L. Iftode, "Fiatlux: Fingerprinting rooms using light intensity," in *Advances in Pervasive Computing: Adjunct Proceedings of the 5th International Conference on Pervasive Computing*, 2007.

[15] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. USENIX Association, 2014, pp. 331–343.

[16] H. Aoki, B. Schiele, and A. Pentlan, "Realtime personal positioning system for a wearable computer," in *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on.* IEEE, 1999, pp. 37–43.

[17] B. Clarkson, K. Mase, and A. Pentland, "Recognizing user context via wearable sensors," in *Wearable Computers, The Fourth International Symposium on*, Oct 2000, pp. 69–75.

[18] N. Ravi, P. Shankar, A. Frankel, A. Elgammal, and L. Iftode, "Indoor localization using camera phones," in *Mobile Computing Systems and Applications, 2006. WMCSA'06. Proceedings. 7th IEEE Workshop on.* IEEE, 2006, pp. 49–49.

[19] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 261–272.

[20] R. Spreitzer, "Pin skimming: Exploiting the ambient-light sensor in mobile devices," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*. New York, NY, USA: ACM, 2014, pp. 51–62.