Video-Erkennung mit Smartphones und Smartwatches

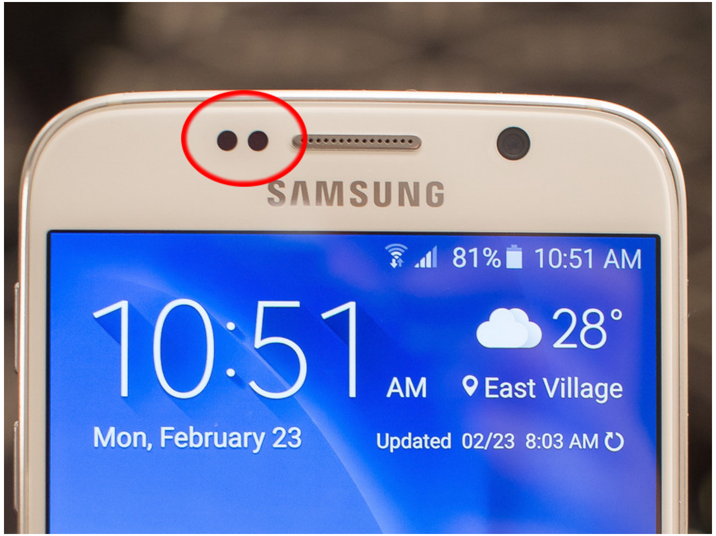
Lorenz Schwittmann, Viktor Matkovic,
Matthäus Wander, Torben Weis

Kasseler Informatik-Kolloquium
7. Juli 2016

Idea

Goal

- Recognize video playing in user's proximity
- Purpose: context awareness
- Using ambient light sensors
  - Built into smartphones and smartwatches
  - Original purpose: adapt screen brightness and user interface to user's ambient light



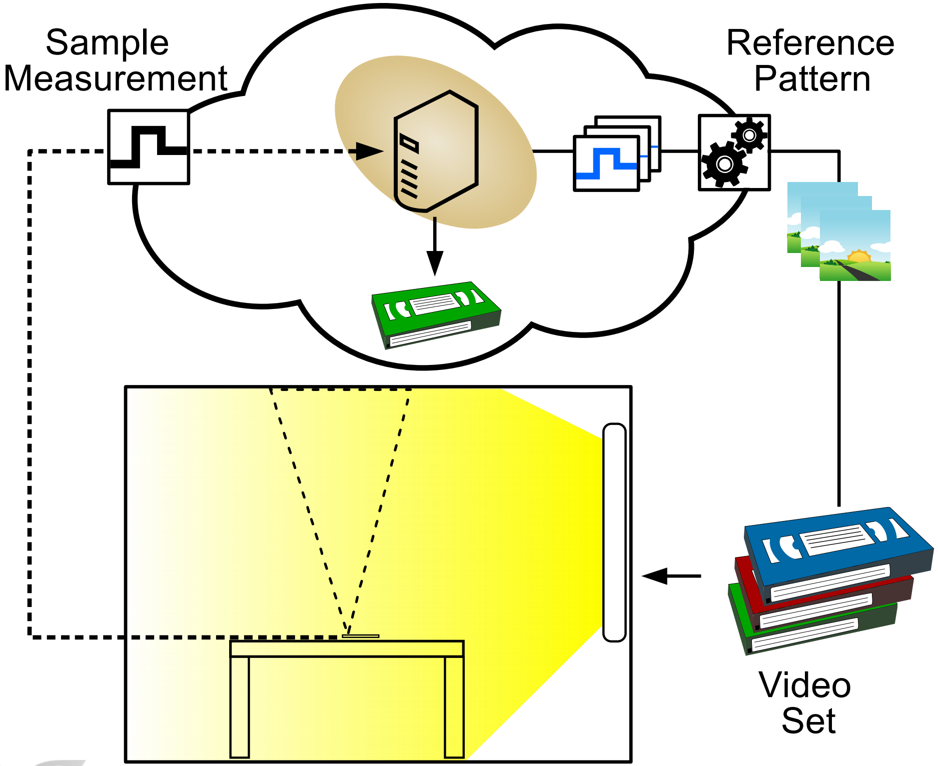
Use Cases

- Smart devices can react to videos playing
- No disturbance in final minutes of a movie
- Provide background information (cast, director, etc.), even to specific scenes
- Learn user's interests
  - Targeted advertisements

Why Ambient Light Sensor?

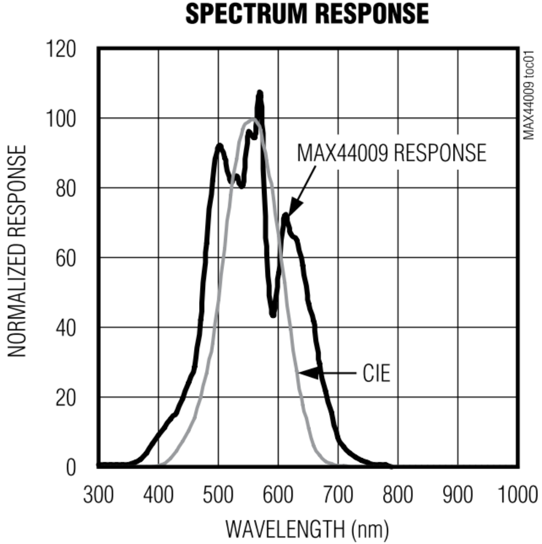
- Audio/video fingerprinting is possible with microphone or camera
- Battery
  - Ambient light sensor consumes less energy
  - Less CPU/data for processing light values than camera images
- Privacy
  - Audio/video recording is privacy-invasive
  - User permission not required to access the ambient light sensor

System Model



Illuminance

- Ambient light sensor measures illuminance in Lux
  - Brightness as perceived by the human eye
- Light sensors use (typically two) photodiodes
  - Empirical Lux approximation
  - Error is non-linear (i.e. less accurate for large Lux values)



Mapping Video Streams to Illuminance

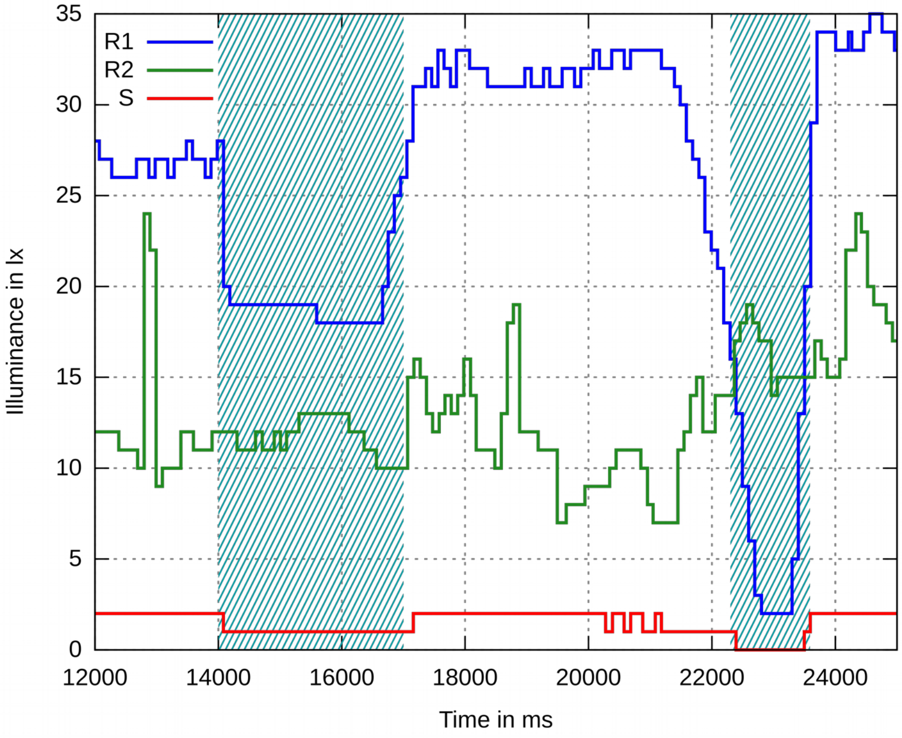
- Transform each video frame to an illuminance value
- Map sRGB color spectrum to CIE XYZ color space

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

- Y represents relative luminance
- Calculate average illuminance of a frame

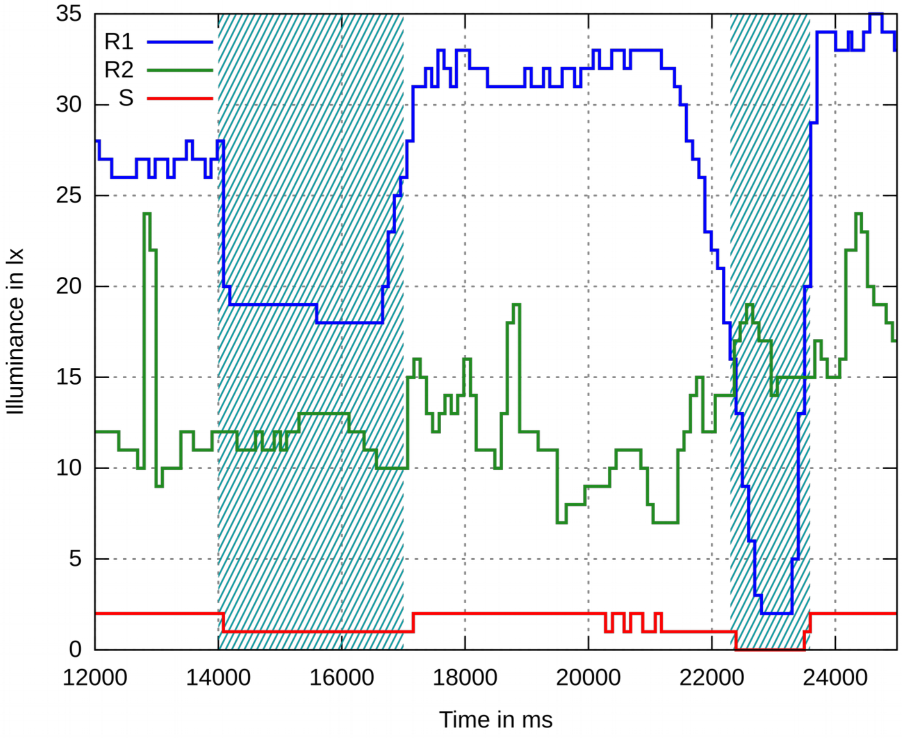
Signal Correlation

- Input:
  - *n* reference patterns/TV channels (R1, R2)
  - 1 sample measurement (S)
- Output:
  - Most likely reference pattern



Signal Correlation II

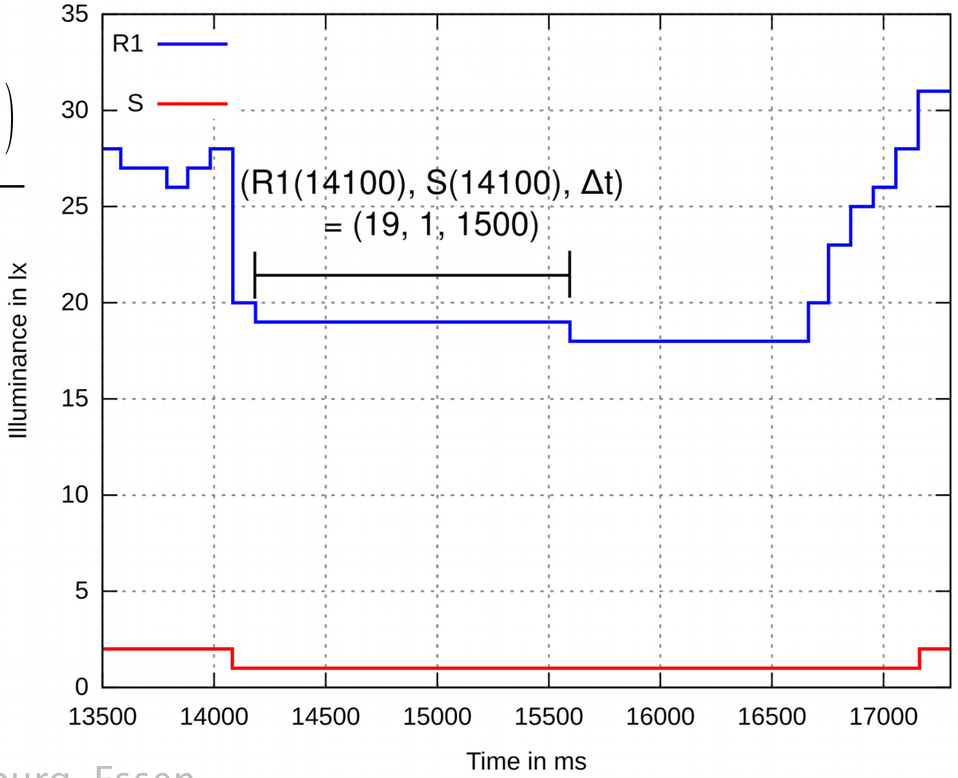
- Challenges:
  baseline and scaling
- We use Pearson's weighted correlation coefficient



Pearson's Weighted Correlation Coefficient

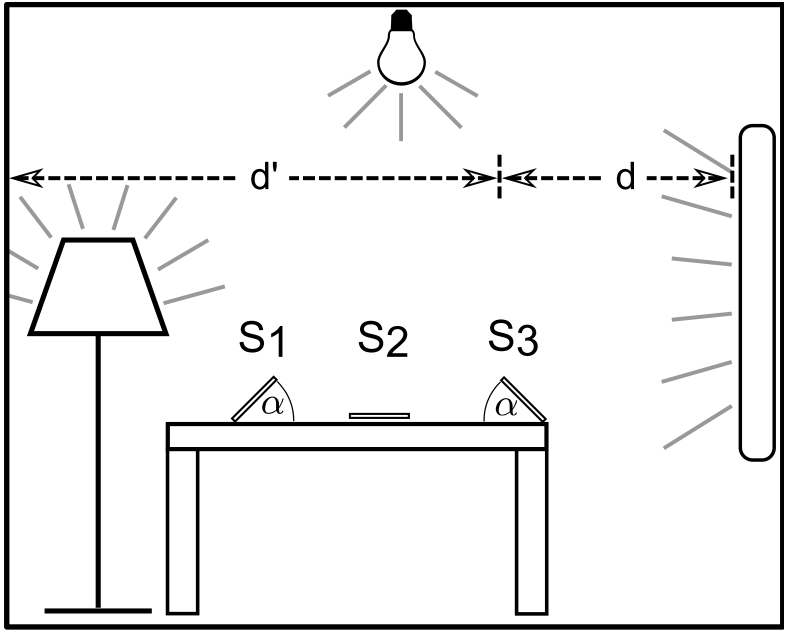
$$cov(A,B,t) := \frac{\sum_i \left((t_{i+1} - t_i)(f_A(t_i) - m(A,t))(f_B(t_i) - m(B,t)) \right)}{\sum_i (t_{i+1} - t_i)}$$

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



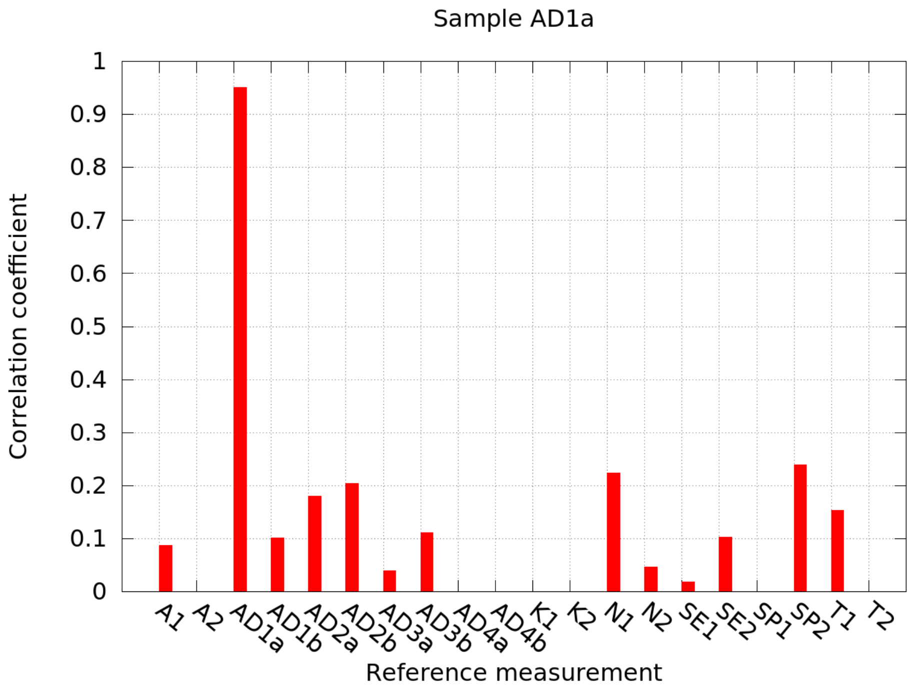
Evaluation Setup

- Variables of evaluation setup include
  - Distance to screen
  - Orientation towards screen
  - Preexisting ambient light



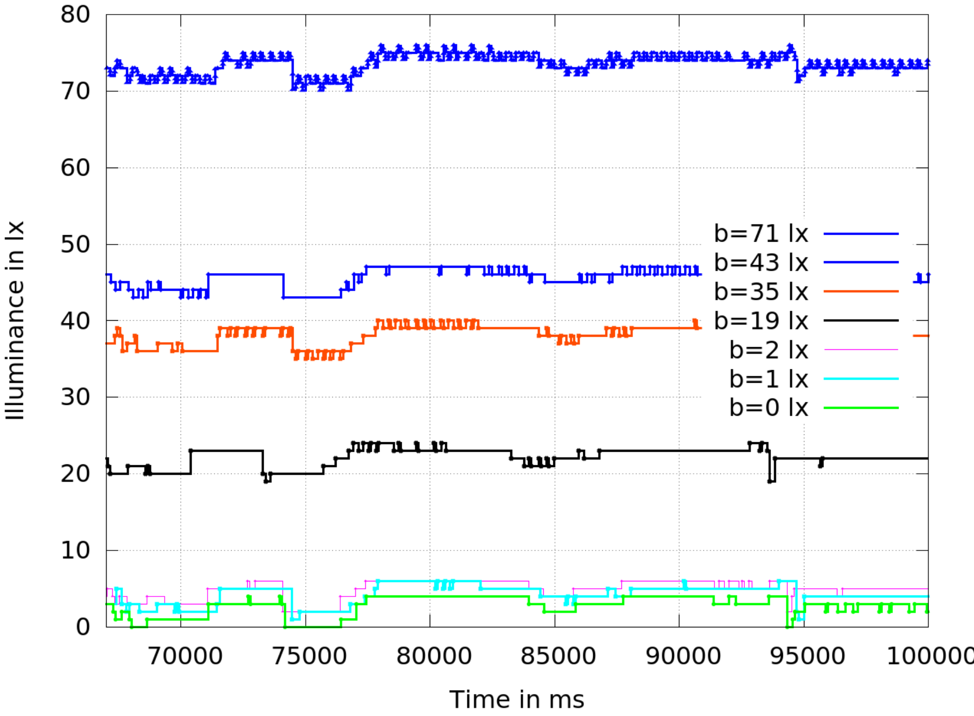
Evaluation: Genre Recognition

- 20 TV channels, 7 genres
- 20 reference patterns and sample measurements
- Phone faces wall, room is dark
- Recognized every video



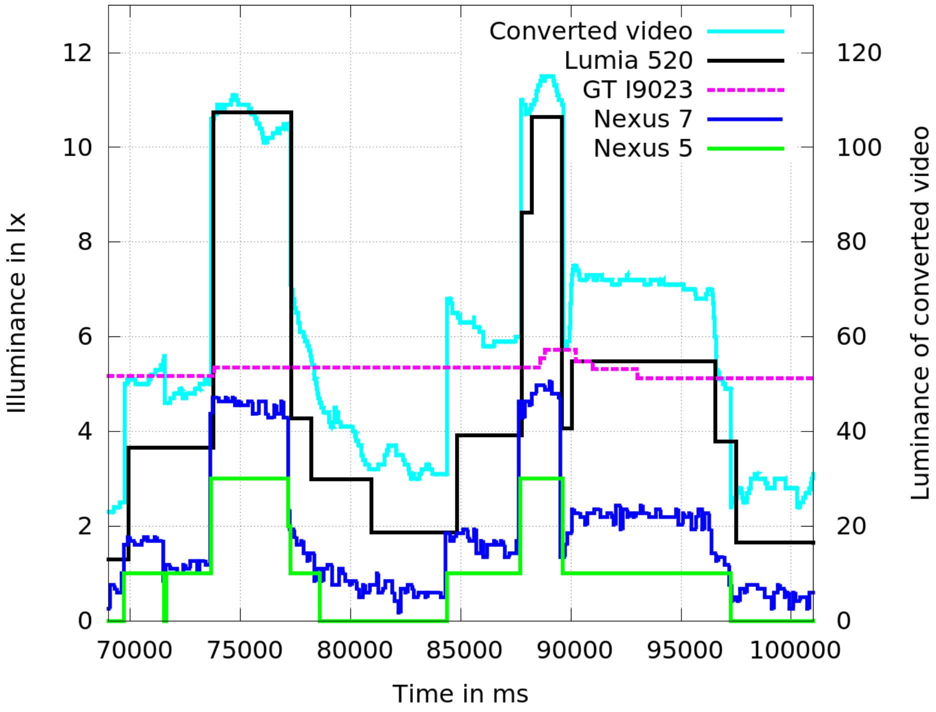
Evaluation: Preexisting Ambient Light

- Turn on room lights
- Re-evaluated TV channel set
- No significant change in recognition ratio
- Light is additive, recognition depends on sensor quality



Evaluation: Devices

- Recognition ratios of devices (TV channel set):
  - Nexus 5 & 7 (2013): 100%
  - Lumia 520 (2013): 90%
  - GT I9023 (2011): 60%



Evaluation: YouTube Video Recognition

- 1180 popular YouTube videos, 60 seconds each. Distance 2m, phone faces wall.
  - Recognition ratio 60%. Reason: seldom changes in illuminance
  1) Freeze frame videos, i.e. audio only
  2) Freeze frames with occasional text fade-in
  3) Single person speaking to a fixed camera
- 149 professional YouTube videos, same setup
  - Recognition ratio 92%

Number of Sensor Readings per Minute

	Recognition	Q(0.25)	Q(0.5)	Q(0.75)
YouTube Popular	60% (N=1180)	7	27	46
YouTube Professional	92% (N=149)	23	32	43
TV	100% (N=20)	29	43	49

- Professional videos have more cuts and light changes
- More light changes $\Rightarrow$ better recognition

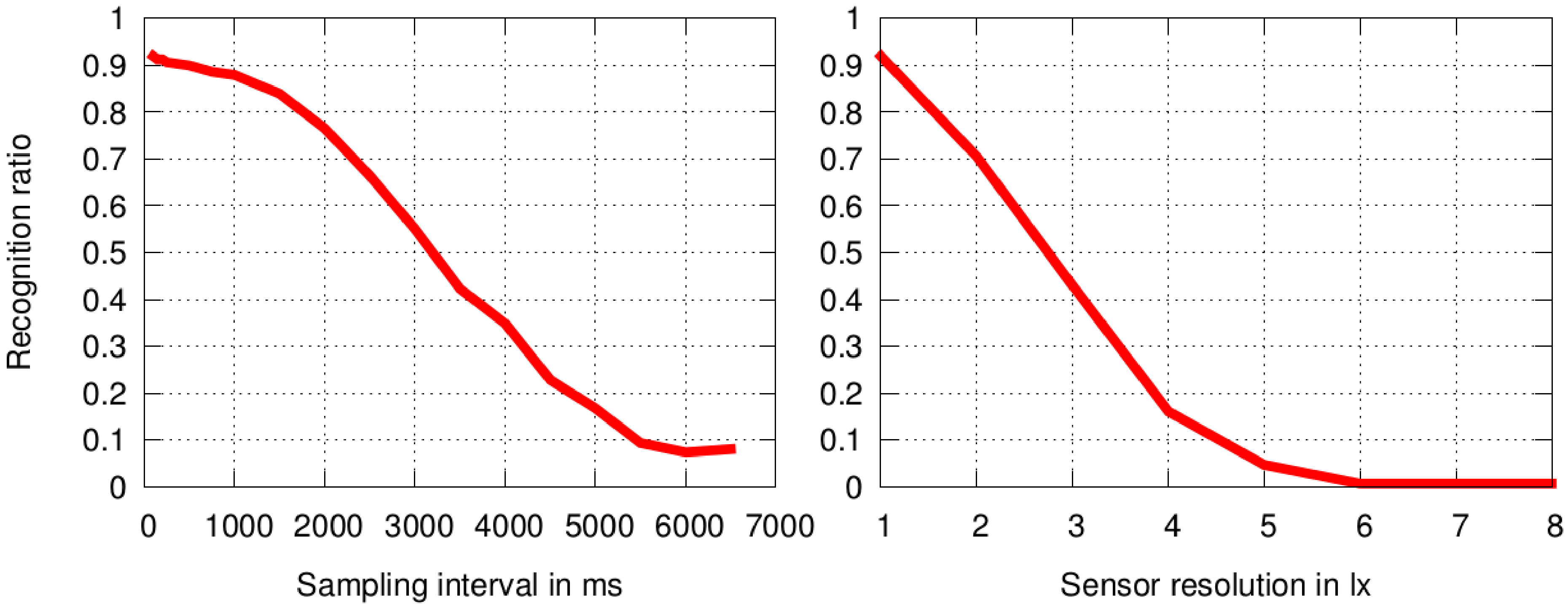
Platform Support for Ambient Light Sensing

- **Android**: user permission not required, background sensing possible
- **Android Wear**: must request permission to keep device awake, but not for light sensing
- **iOS**: does not expose ambient light sensor to applications
- **Windows Phone**: user permission not required, background sensing possible
- **Web**: user permission not required, sensing on active tab only (Firefox for Android)

Privacy Considerations

- Apps and websites can implement video recognition without user consent
  - “I know what you watched last summer”
- Potential countermeasures:
  - Require user permission (cumbersome with a dozen of sensors)
  - Truncate sensor data (sampling interval or sensor resolution)
  - High-level information instead of Lux, e.g. “dim, normal, washed” (W3C Media Queries Draft)

Recognition with Truncated Sensor Data



Conclusion

- Approach for recognizing videos playing in user's proximity
- Recognition ratio depends on illuminance changes and sensor quality (precision and resolution)
  - TV 100%, YouTube professional 92%, YouTube popular 62%
- Seemingly harmless sensor data may violate privacy
  - Truncate sensor resolution to prevent video recognition

PerCom 2016 Paper

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**

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.

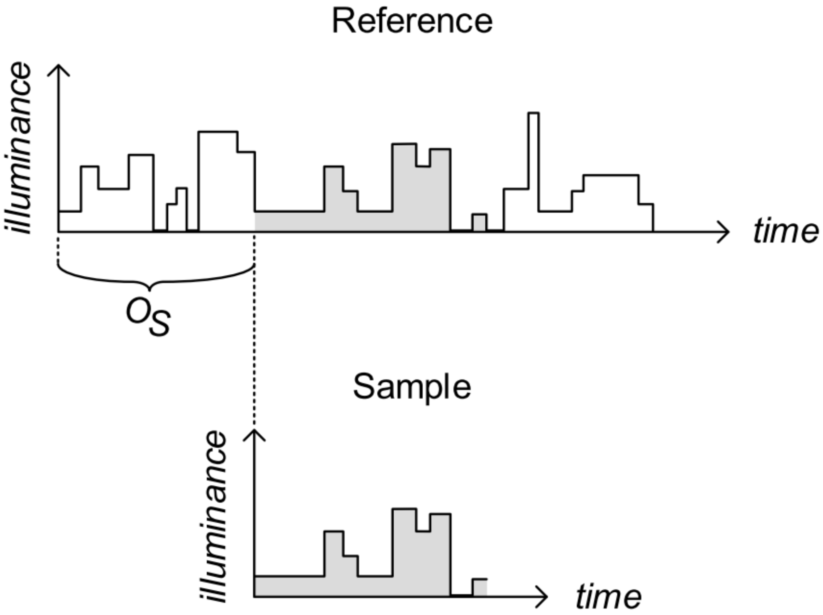
Our approach works by analyzing the ambient light that can stem from reflections of the wall if a mobile device is not pointed at the TV screen. The light level is collected with the ambient light sensor, which is more effective for this task than the camera. The data retrieved by sampling the ambient 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

Feasibility: Server-side Load

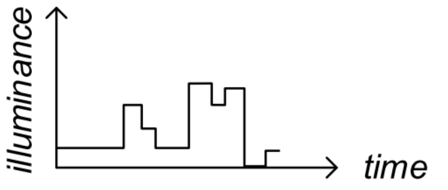
- Video conversion: real-time, 10%-20% CPU usage in prototype HTML5 application (Intel Core i7-3520M)
- Correlating 2 videos (60 seconds): 2ms
- Network load: 40 ms (=25 FPS) sampling interval yields at most 3 KBytes/minute

Challenge: Find Offset of Current Video Clip

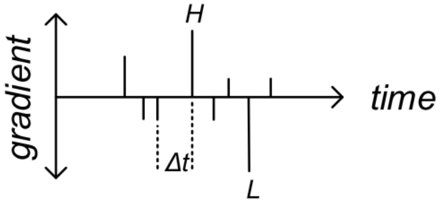
- Exhaustive search
  - Adjust offset O_S few ms
  - Re-run correlation
  - Suitable for live TV
- Find probable anchors
  - Look for matching edges
  - Suitable for Video-on-Demand



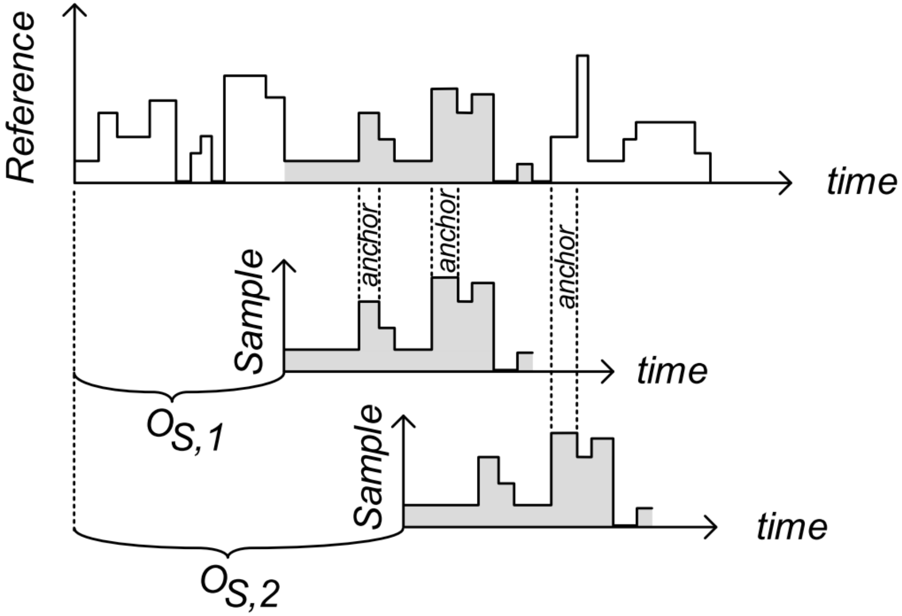
Offset Search: Find Probable Anchors



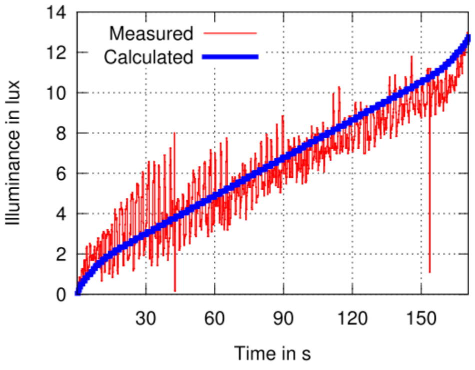
(a) Measurement M.



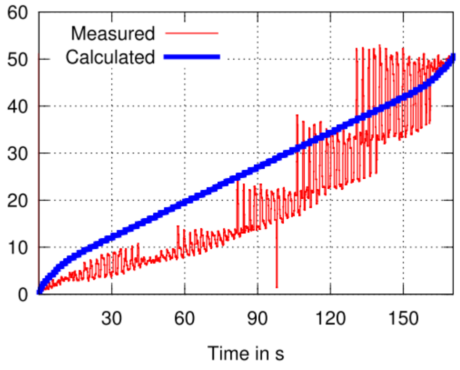
(b) Derivative M'.



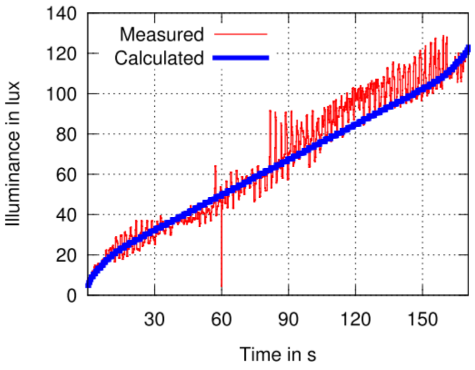
Color Deviation of Screens



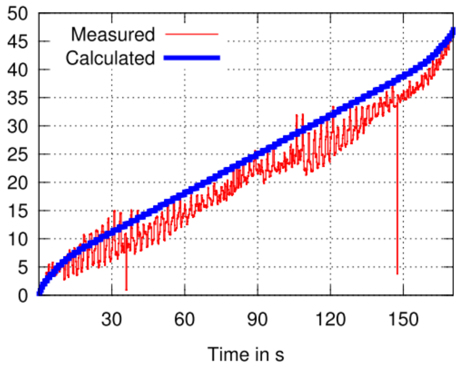
(a) Toshiba 42XV501P



(b) Epson EB-955W



(c) Samsung UE65F6470



(d) Toshiba 47WLG66

Related Work

- Image-based/audio video fingerprinting techniques
  - Higher CPU, network & power consumption than with ALS
- Indoor localization with ambient light or camera:
  - Ravi and Iftode: fingerprint room from ambient light
  - Li et al.: visible light communication with LED modulation
  - Extract ambient color/illuminance from camera
- Spreitzer: infer probable keystrokes from light data