

# EINSATZ VON MOBILE DEVICES MIT FREIHAND-GESTENERKENNUNG ALS PRÄSENTATIONSPLATTFORM IM PROFESSIONELLEN UMFELD

*Thomas Rehm, Saskia Schreiber*

Technische Hochschule Mittelhessen  
Friedberg, Hessen

## ZUSAMMENFASSUNG

In dieser Arbeit soll untersucht werden, inwieweit sich diese Geräte durch ein mobiles Endgerät mit Gestenerkennungssoftware ersetzen lassen und ob die Eingliederung in bestehende Infrastruktur (Beamer / SmartTV) praktikabel ist.

**Schlagwörter**— mobiles Endgerät, Freihandgesten, Gestensteuerung, Präsentation

## 1. EINFÜHRUNG

Seit der Ablösung des Overhead-Projektors durch Laptop und Beamer kamen viele Dienste und Technologien auf den Markt, um das Erstellen von Präsentationen zu vereinfachen. Die eingesetzte Hardware hat sich dagegen nicht wesentlich verändert: Immer noch sind Laptop und Maus (oder separate Fernbedienung) in der Regel fester Bestandteil der Ausrüstung.

Durch die steigende Nutzung und Verbreitung von mobilen Endgeräten sowie vernetzten Ausgabe-Geräten (Beamer/SmartTV) stellt sich jedoch die Frage, ob man daraus nicht ein einfacheres Setup schaffen kann. Einfacher durch reduzierte Infrastruktur: Mobilgerät statt Laptop, kabellose Verbindungen statt direkter Verkabelung; einfacher aber auch mit dem Wegfall separater Steuerungsgeräte durch intuitive Gestensteuerung.

Somit ist diese Arbeit keine Vorstellung einer neuen Technologie. Sie untersucht stattdessen, ob sich mittlerweile etablierte Technologien zu einem neuartigen Konstrukt verbinden lassen, das die Nachfolge klassischer Präsentations-Setups antreten kann.

## 2. STAND DER TECHNIK

### 2.1. Stationäre Geräte zur Gestensteuerung

Die Nutzung von Freihandgesten als Eingabemedium ist an sich kein neues Konzept.

Ein sehr prominentes Beispiel dafür ist die Steuerungshardware Kinect, mit der sich Windows-Geräte und die Spielekonsole Xbox 360 komplett ohne Controller bedienen lassen. Dank eines umfangreichen SDKs können Entwickler mit der

Kinect nicht nur Spiele steuern, sondern die unterschiedlichsten Applikationen mit Körperbewegungen in Verbindung bringen. Da der Benutzer in der Regel frontal vor dem Gerät steht, lassen sich Bewegungen des ganzen Körpers nachverfolgen.

Die Kinect fällt in die Kategorie der stationären Systeme, die alle dank ordentlicher Rechenleistung eine große Genauigkeit bei der Interpretation von Bewegungen erlauben. Für die Steuerung von Präsentationen, wo im Prinzip nur links-rechts-Bewegungen der Hand eine Rolle spielen, wird dies gar nicht benötigt. Dafür fällt der Nachteil der stationären Geräte stark ins Gewicht: Sie sind für den spontanen Einsatz außer Haus zu unflexibel, der Transport und Aufbau zu kompliziert für den fachfremden Präsentierenden.

### 2.2. Gestensteuerung auf Mobilgeräten: Samsung Gesture SDK

Gestensteuerung auf Mobilgeräten wiederum hat andere Voraussetzungen und daher auch andere Ziele. Die Genauigkeit ist durch geringere Prozessorleistungen limitiert, die Aufnahmemöglichkeiten von Gerät zu Gerät unterschiedlich. Hier sticht das *Gesture SDK* von Samsung [1] heraus, welches als einheitliche Schnittstelle auf einigen mobilen Geräten dieses Herstellers vorinstalliert war und von App-Entwicklern genutzt werden konnte. Die unterstützten Geräte brachten dafür einen eigenen Sensor mit. Das Projekt wurde im Dezember 2014 eingestellt.

### 2.3. Handwave

Relevant ist das Projekt Handwave - ein Projekt im medizinischen Bereich, durchgeführt von Studenten der Universität Washington, Seattle, aus dem Jahr 2015 [2]. Die resultierende App hat das Ziel, Medizinern aus hygienischen Gründen Interaktion ohne Berührung ihres Mobilgerätes zu ermöglichen. Sie ermöglicht beispielsweise, mit horizontalen und vertikalen Gesten durch Instruktionen zu navigieren. Die eingesetzte Software *OpenCV* [3] muss dafür separat auf dem Gerät installiert oder mitgeliefert werden und hat den Nachteil, dass es große Kompatibilitätsprobleme mit neueren Android-Versionen gibt. Dafür kommt die App oh-

ne zusätzliche Sensoren aus, sie nutzt die Frontkamera des Gerätes zum Auswerten der Gesten. Aus diesem Grund spielen die Ergebnisse des Projektes auch für diese Arbeit eine wichtige Rolle, da sie die Machbarkeit kameragesteuerter Gestenerkennung auf Mobilgeräten demonstriert.

## 2.4. Screen Mirroring und Wireless Connections

Damit der Einsatz eines Mobilgerätes als Präsentationsmedium überhaupt Sinn macht, sollte dieses auch mobil bleiben, also möglichst ohne Kabel mit dem Beamer/TV kommunizieren können. Glücklicherweise gibt es in der Zeit der SmartTVs einige Lösungsansätze, die meisten bereits im Produktiveinsatz.

In *Screencast in the Wild: Performance and Limitations* [4] werden u.A. die verbreiteten Protokolle *AirPlay*, *Chromecast* und *Miracast* gegenübergestellt und auf ihre Performance getestet. Dass hier *Miracast* sehr gut abschneidet, bedeutet im Umkehrschluss auch eine gute Prognose für die direkte Anbindung an SmartTVs und Set-Top-Boxes, da diese häufig damit arbeiten (vgl. Abschnitt 4.5).

## 3. APP-KONZEPT ANDROIDGESTURECONTROL

Nachfolgend sind die Kern-Funktionen gelistet, durch die sich die Proof-of-Concept-App von ähnlichen Lösungen abgrenzen soll:

- **Freihand-Foliensteuerung:** Erkennung von Hand-Gesten um eine innovative Freihand-Bedienung während der Präsentation zu gewährleisten
- **User Interface:** Button zum ein-/ausschalten der Gestenerkennung
- **Bestehendes Präsentations-Ökosystem:** Nutzen einer bestehenden Web-App für Präsentationen, um Decks im Browser zu editieren/zu verwalten
- **Native App:** Implementierung einer nativen App zu besserer Bedienbarkeit und Einbindung nativer Features (Vollbild, Kamera-Support)
- **Reduzierung der benötigten Geräte:** Lediglich ein Mobilgerät wird benötigt, es müssen keine zusätzlichen Geräte zur Gestenerkennung angeschafft und transportiert werden
- **Universelle Portierbarkeit:** Einsatz von Webtechnologien für erleichterte Portierbarkeit (Android/iOS)

## 4. IMPLEMENTIERUNG

In der Realisierung werden bestehende Techniken kombiniert, um das innovative und intuitive Bedienkonzept zu erreichen. Die Technologien funktionieren gut in den Umgebungen, für

die sie entwickelt wurden. Die Herausforderung besteht darin, in der Kombination einen funktionierenden Prototypen zu realisieren.

### 4.1. Reveal.js + gesture.js

Die Basis für die Gestenerkennung bildet das *gesture.js* Skript, das von Entwickler William Wu für das Präsentations-Framework *Reveal.js* [5] von Hakim El Hattab entwickelt wurde. *Gesture.js* lädt mittels WebRTC einen Webcam-Video-Stream und gibt diesen in einer Canvas-Fläche aus (nähere Informationen in der Demo: [6]). Per JavaScript werden Bildinformationen aus dem Canvas-Objekt extrahiert und manipuliert. Hier setzt der Algorithmus zur Gestenerkennung an und erkennt die Handbewegung vor einem ruhigen Hintergrund. Der Algorithmus ruft dann die passende *Reveal.js*-Funktion auf, um zur jeweilig passenden Bewegungsrichtung durch die Folien zu navigieren.

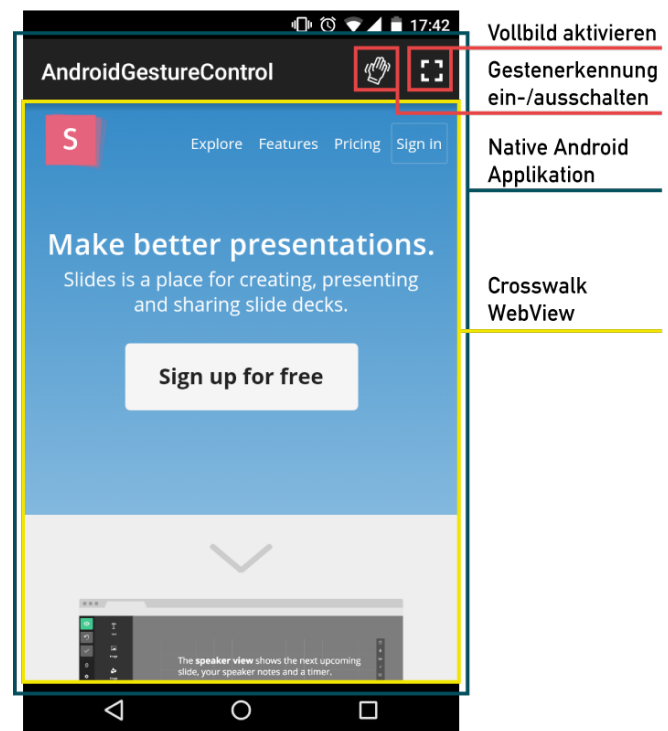


Fig. 1. Screenshot App mit Slides.com-Startseite.

### 4.2. Slides.com

Das bereits erwähnte *Reveal.js*-Framework ist unter der freien MIT Lizenz verfügbar. Mit *slides.com* hat der Entwickler Hakim El Hattab eine Freemium-Plattform bereitgestellt, die eine Web-App für die Erstellung der Decks/Präsentationen im *Reveal.js* Style und Format realisiert. Durch den Einsatz des Services wird ein bewährtes Tool zur Erstellung und Verwaltung der Decks eingesetzt und der Nutzer kann sich sowohl

in der App als auch am Computer in sein Slides.com-Konto einloggen.

#### 4.3. Native Android-App mit WebView

Um Slides.com und die Gestenerkennung per JavaScript in einer nativen Android-App kombinieren zu können, wird die Web-App in einem WebView aufgerufen und das modifizierte *gesture.js*-Skript injiziert. Es handelt sich somit um eine *WebView Application* [7]. Da der von Android bereitgestellte WebView stark eingeschränkt ist und keinen direkten Zugriff auf die Kamera des Geräts zulässt, wird der alternative WebView des *Crosswalk Project* [8] eingesetzt. Dieser ist ein In-App Browser mit vollem Funktionsumfang des aktuellsten Chromium-Browsers mit der Rendering-Engine *Blink* [9]. Durch den Crosswalk-WebView ist das direkte Ansprechen der Kamera mittels WebRTC API möglich.

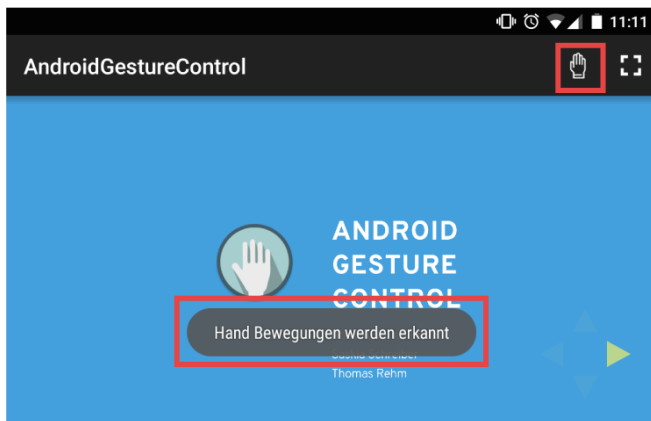


Fig. 2. Screenshot App mit aktivierter Gestenerkennung.

#### 4.4. Benutzerinteraktion

Wichtig für ein intuitives Erlebnis ist auch die Bedienung der App, die mit wenigen Buttons ihre Funktionalität bereitstellt. So befinden sich neben dem WebView mit den Bedienelementen der Web-App noch je ein Button für die Aktivierung des Vollbild-Modus und für das Ein- und Ausschalten der Gestenerkennung (siehe Fig. 1). Wird die Gestensteuerung vom Nutzer aktiviert, ändert sich das Piktogramm und dem Nutzer wird in einer kurzen Nachricht mitgeteilt, dass jetzt die Bewegungen erkannt werden (siehe Fig. 2).

#### 4.5. Präsentieren mit AndroidGestureControl

Die eigentliche Ausgabe und Präsentation des Bildes erfolgt mittels Bildschirmübertragung des Android-Bildschirms an einen Beamer oder Fernseher. Die Übertragung kann per Kabel via *MHL (Mobile High-Definition Link)* [10] direkt an den HDMI-Anschluss des Beamers/TVs oder per WLAN an einen

passenden Empfänger mit HDMI-Anschluss übertragen werden. Dabei wird ein HDMI-Stick eingesetzt, der ein passendes Übertragungsprotokoll wie *Google Cast* [11], *Miracast* [12] oder *Apple AirPlay* [13] unterstützt. Einige SmartTVs unterstützen die genannten Protokolle bereits und benötigen keinen zusätzlichen Stick. Die drahtlose Übertragung des Bildschirms bietet dem Präsentierenden eine höhere Freiheit als die kabelgebundene Lösung. Die genutzten Geräte müssen lediglich über das gleiche drahtlose Netzwerk verbunden sein. Teilweise benötigen die Geräte einen Internetzugang, um die Daten zu übertragen; in jedem Fall muss die Präsentation zuvor per Internet aufgerufen werden.

### 5. ZUSAMMENFASSUNG UND BEWERTUNG DER ERGEBNISSE

Im Rahmen dieser Arbeit wurden verschiedene Technologien kombiniert, um das gesetzte Ziel zu erreichen: Gestensteuerung über Kamerainformationen auf der einen und Screen Mirroring auf der anderen Seite. Gestensteuerung auf Mobilgeräten einzusetzen ist offensichtlich kein Problem mehr und dank der Nutzung von Webtechnologien auch nicht auf Android beschränkt. Die Rechenleistung der Geräte reicht dafür aus, auch wenn die Genauigkeit der im Prototyp eingesetzten Algorithmen noch feiner sein könnte. Da die Umrechnung der Bilddaten in Gesten jedoch nicht das Kernthema darstellt, ist das Ziel aus Sicht der Autoren dennoch voll erreicht worden. Weitere Themen ergeben sich natürlich: Device-übergreifende Performancetests könnten beispielsweise aufzeigen, wo noch Spielraum für mehr Datentransfer und somit genauere Gestenerkennung ist. Bei der Auswahl des Protokolles zur Verbindung der Geräte wurden sowohl Google Cast als auch Miracast über Set-Top-Boxen getestet. Beide Varianten funktionieren, haben aber Einschränkungen in Handhabung und Stabilität. Eine direkte Nutzung der Protokolle durch SmartTVs ist damit ein wichtiges Kriterium für die Alltagstauglichkeit des vorgestellten Systems. Das wiederum setzt voraus, dass in den Präsentationsräumen überhaupt ein SmartTV zur Verfügung steht. Entsprechende Statistiken[14] zeigen, dass diese zumindest in privaten Haushalten immer mehr an Bedeutung gewinnen.

### 6. LITERATUR

- [1] SAMSUNG Developers, "Gesture by SAMSUNG Developers," 2014. [Online]. Available: <http://developer.samsung.com/resources/gesture>. Accessed: 2015-10-07
- [2] N. Dell, K. D'Silva, and G. Borriello, "Mobile Touch-Free Interaction for Global Health," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '15. New York, NY,

USA: ACM, 2015, pp. 15–20. [Online]. Available: <http://doi.acm.org/10.1145/2699343.2699355>

223468/umfrage/prognose-der-smart-tv-entwicklung-nach-anteil-der-tv-geraete-in-deutschland/. Accessed: 2015-10-07

- [3] Itseez, “OpenCV Website.” [Online]. Available: <http://opencv.org/>. Accessed: 2015-10-07
- [4] C.-F. Hsu, D.-Y. Chen, C.-Y. Huang, C.-H. Hsu, and K.-T. Chen, “ScreenCast in the Wild: Performance and Limitations,” in *Proceedings of the ACM International Conference on Multimedia*, ser. MM ’14. New York, NY, USA: ACM, 2014, pp. 813–816. [Online]. Available: <http://doi.acm.org/10.1145/2647868.2654991>
- [5] H. E. Hattab, “Reveal.js,” 2015. [Online]. Available: <https://github.com/hakimel/reveal.js/>. Accessed: 2015-10-07
- [6] W. Wu, “Webcam-based gesture recognition with reveal.js,” 2013. [Online]. Available: <http://revealjs.herokuapp.com/#/>. Accessed: 2015-10-07
- [7] P. Hazarika, C. P. Rahul Raj, and S. Tolety, “Recommendations for Webview Based Mobile Applications on Android,” in *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*, 2014, pp. 1589–1592.
- [8] Intel Corporation, “Crosswalk - build world class hybrid apps,” 2015. [Online]. Available: <https://crosswalk-project.org/>. Accessed: 2015-10-07
- [9] Chromium Org., “Blink - The Chromium Projects.” [Online]. Available: <http://www.chromium.org/blink>. Accessed: 2015-10-07
- [10] MHL Consortium, “MHL® – What is MHL? How Do I Use It?” [Online]. Available: <http://www.mhlconsortium.org/consumer.aspx#whatIsMHL>. Accessed: 2015-10-07
- [11] Google Developers, “Google Cast,” 2015. [Online]. Available: <https://developers.google.com/cast/>. Accessed: 2015-10-07
- [12] M. A. M, P. RaoT, and D. N. G. S, “EXTENDED DISPLAY MODE USING ANDROID MIRACAST,” *IJITR*, vol. 0, no. 0, 2015. [Online]. Available: <http://www.ijitr.com/index.php/ojs/article/view/555>
- [13] A. Support, “Mit AirPlay drahtlos Inhalte von einem iPhone, iPad oder iPod touch streamen,” 2015. [Online]. Available: <https://support.apple.com/de-de/HT204289>. Accessed: 2015-10-07
- [14] BLM, “Prognose zur Anzahl der Haushalte mit mind. einem an das Internet angeschlossene TV-Gerät von 2011 bis 2016 (in Millionen),” 2012. [Online]. Available: <http://de.statista.com/statistik/daten/studie/>