A Research Framework for Visually Linking Mobile Services to TV Content

Matthias Baldauf
FTW Telecommunications
Research Center Vienna
Donau-City-Strasse 1
1220 Vienna, Austria
baldauf@ftw.at

Ulrich Lehner
FTW Telecommunications
Research Center Vienna
Donau-City-Strasse 1
1220 Vienna, Austria
lehner@ftw.at

Peter Fröhlich
FTW Telecommunications
Research Center Vienna
Donau-City-Strasse 1
1220 Vienna, Austria
froehlich@ftw.at

Proceedings of TVUX-2013: Workshop on Exploring and Enhancing the User Experience for TV at ACM CHI 2013,
27 April 2013, Paris, France.

Abstract
Mobile devices have established as 'second screens' used simultaneously while watching traditional broadcasted television. By identifying the currently running telecast, they are able to provide accompanying information and interactive services. One novel way to achieve this synchronization is the visual recognition by pointing a smartphone camera towards the television set. To facilitate the investigation of respective open research questions, we present our ongoing work on an experimentation framework for visually linking mobile services to TV content. We describe its management platform, recognition engine and mobile application prototype and further outline interesting research aspects we plan to address with the described framework.

Author Keywords
TV synchronization; mobile interaction; video recognition

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces: *Interaction Styles, Prototyping*; I.5.4 [Pattern Recognition]: Applications: *Computer vision*

General Terms
Experimentation, Human Factors

Figure 1. Example *VideoSurf*: Accessing background information and mobile services accompanying the currently watched telecast by pointing the smartphone towards the television set.

Introduction

According to a recent study by *Nielsen*[1], up to 86% of TV viewers with smartphones use them simultaneously while watching TV. Broadcasting companies, TV set manufacturers and various start-ups recognized this trend towards the so-called 'second screen' and have started with respective offers to enhance the TV viewing experience by delivering background information or even interactive services via mobile applications while the seamless synchronization between the TV content and the mobile application is a crucial element. One promising approach is pointing the smartphone towards the television set to visually recognize the telecast through the built-in camera as depicted in Figure 1.

Since the implementation of necessary recognition platforms is not trivial, there are no such frameworks publicly available for research purposes. In this paper, we outline our current work towards a respective experimentation framework to facilitate the investigation of both open technical and human-computer interaction (HCI) issues.

Device/TV synchronization

One very simple, yet less user-friendly approach for synchronizing a mobile device with the current telecast are explicit 'check-ins' as for example applied by *GetGlue*: users are asked to specify what TV show they are watching. While this method could be cumbersome (e.g. to specify a specific episode of a specific season of a TV show), it further provides no means to recognize specific moments within a show. Another explicit method is the usage of Internet-enabled set-top boxes which let mobile devices query the currently watched channel. However, industry fragmentation prevents this approach from becoming a universal solution. Other synchronization approaches rely upon the adaptation of the original TV broadcasting signal. Examples include the integration of well-known visual markers such as QR codes in the video stream (e.g. [2]). A technically more advanced approach is the encoding of identifiers or even the digital content itself into the broadcasting audio signal and its decoding at the viewer's site through special hardware (e.g. [5]).

Most sophisticated synchronization approaches do not require any adaptations of the TV broadcasting signal. Solutions based on the recognition of audio fingerprints include *Shazaam*, *Umami*, and *Intonow*. A vision-based example is *VideoSurf*: Similar to the framework presented in this paper, *VideoSurf* recognizes TV content by pointing the smartphone camera towards the TV set. However, no respective framework is publicly available for research purposes.

Experimentation framework

The current prototype of our experimentation framework consists of three major parts: the management platform, the recognition service, and a mobile application prototype. The management platform and the recognition service operate on the same image database.

Management Platform

Our web-based platform for managing image targets is implemented using *Java Server Pages* (JSP) on a *Tomcat* server. Its user interface (depicted in Figure 2) allows easily uploading new images and browsing through the collection of existing ones. We make use of latest HTML5 features to make user interactions more

[1] http://blog.nielsen.com/nielsenwire/?p=31338

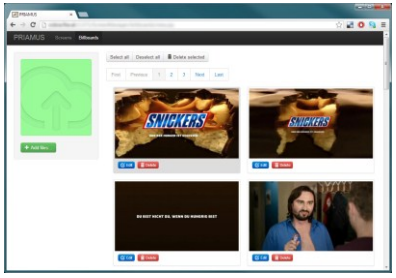

Figure 2. Overview of submitted and annotated video frames and the drag'n'drop field for manually uploading new images.

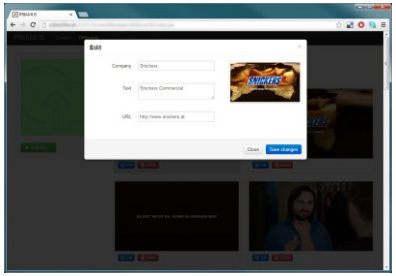

Figure 3. Dialog for annotating video frames with meta information such as a corresponding URL.

comfortable. For example, a set of multiple new target images can be easily contributed by drag'n'dropping the images onto the respective green upload area. Each of the uploaded images then can be annotated by hitting the respective edit buttons. Currently, the meta data that can be assigned to an image include a company name, some arbitrary text as well as a corresponding URL (see Figure 3). After upload, image files are renamed using a universally unique identifier (UUID) and stored in the local file system; the meta information is persistently stored using a *MS SQL* database. For processing video files we currently apply a simple script which extracts video frames at custom intervals and uploads them with passed meta data in a batch (done for the TV commercial in Figure 2).

Recognition Service

We realized the image recognition engine using *Microsoft's .net* framework and *EmguCV*, a *C#* wrapper for the popular computer vision library *OpenCV*. Via a simple restful API service, an image in JPG format can be submitted to the engine using an HTTP post request.

The actual image recognition is based on the SURF algorithm [1] to find unique image parts, so-called interest points, and describe their neighborhood in form of a feature vector with a length of 64 values. To 'recognize an image', the feature vector of a submitted image needs to be compared with the ones of the uploaded images. For this task we apply a fast approximate nearest neighbor search using the FLANN algorithm [4]. The descriptors of the uploaded image targets are held in one joint matrix from which the nearest neighbor search index in form of a kd-tree is constructed through the respective *EmguCV* methods. Each time a new target image is uploaded we calculate the descriptors and update this search index. In analogy, the respective vectors are removed from the index when a target image is deleted. For an image to be recognized we calculate its feature vectors and perform the nearest neighbor search for each one. By applying Lowe's distance ratio optimization [3], we consider only strong matches and count these matches for each of the target images from the database. For the target image with the most matches, we then try to calculate the homography, i.e. the transformation matrix to the submitted image. If this matrix can be determined, we assume that this target is the correct match. The service returns the target id and the associated meta data in JSON format or an error code in case no match could be found.

Mobile application

The framework's current mobile prototype is an Android application. It shows the built-in camera viewfinder and an 'Identify' button. Upon press of this button the application takes a set of camera snapshots in a specific time interval, applies a 30% JPG compression to reduce transmission times and submits them to the recognition service via Wifi or 3G. The number of snapshots, the time as well as further parameters such as the zoom level can be easily changed through a configuration menu for experimentation. As soon as a first hit is detected, i.e. the service returns a valid JSON string, the application displays the fetched meta data to communicate the successful recognition.

Research questions

Our experimentation framework supports the research of manifold issues of visual smartphone synchronization with TV content, both from the perspective of human-computer interaction and the technical point of view.

Technical questions address the improvement of the TV content recognition to reduce waiting times and thus deliver an enhanced user experience. How many frames need to be captured by the mobile device to guarantee a fast, yet reliable recognition? This is closely related to the question on how and at which granularity to extract suitable image targets from the videos to be annotated. How can a preprocessing of the captured frames on the mobile device enhance the overall recognition? Which level of compression can be applied to the search images without impeding the recognition? What is the best compromise between compression and transmission times? Or should even entire video sequences be captured and sent to (an updated version of) the recognition service? May some of the workload be shifted to 'Smart TV' sets in form of dedicated apps? Can the visual synchronization approach also be applied for TV live events instead of prerecorded shows?

HCI research should shed light on user-related issues of the visual synchronization approach. What are acceptable pointing (i.e. recording) and recognition times? What are the advantages of this approach over audio-based recognition which obviously works transparently without any user interaction? While most available second screen aps are restricted to accompanying information, what novel interactive services are enabled through the presented visual synchronization approach? For example, could we realize games to be directly played on the TV content such as hitting some targets? How can available digital content or even interactive services be communicated in a sophisticated markerless setup?

Conclusions and outlook

In this paper, we presented our ongoing work towards an experimentation framework for linking mobile services to broadcasted video content. While first tests proved the general feasibility and functionality of the described system, we will continue optimizing the framework to improve its robustness and performance and address the outlined technical research questions. Further, we plan to design and conduct respective user studies to explore promising services and their user-acceptance. In the longer term, we mean to make the framework available to the HCI research community.

Acknowledgements

This work has been carried out within the project PRIAMUS financed by FFG and A1. The Competence Center FTW Forschungszentrum Telekommunikation Wien GmbH is funded within the program COMET by BMVIT, BMWA, and the City of Vienna. The COMET program is managed by the FFG.

References

[1] Bay, H., Ess, A., Tuytelaars, T., and van Gool, L. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding*, 110, 3, 346-359, 2008.

[2] FOX Broadcasting Company. FOX CODES. http://www.fox.com/qrcodes/

[3] Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60, 2, 91-110, 2004.

[4] Muja, M. and Lowe, D.G. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. *Proc. VISSAPP'09*, 331-340, 2009.

[5] Photeon Technologies. Media Transmission. http://photeon.com/applications.html