Scalable Mobile Image Recognition for Real-Time Video Annotation

Philipp Fleck*
Graz University of Technology
AR4 GmbH

Clemens Arth†
Graz University of Technology
AR4 GmbH

Dieter Schmalstieg‡
Graz University of Technology

ABSTRACT

Traditional AR frameworks for gaming and advertising focus on tracking 2D static targets. This limits the plausible use of this solutions to certain application cases like brochures or posters, but deprives their use for dynamically changing 2D targets, such as video walls or electronic billboards used in advertising.
In this demo, we show how to use a rapid, fully mobile image recognition system to introduce AR in videos playing on TV sets or other dynamic screens, *without* the need to alter or modify the content for trackability. Our approach uses a scalable and fully mobile concept, which requires a database with a very small memory footprint on mobiles for a video or even a collection of videos.
The feasibility of the approach is demonstrated on over 16 hours of video from a popular TV series, indexing into the video and giving accurate time codes and full 6DOF tracking for AR augmentations.

Keywords: Computer Vision, Augmented Reality, Mobile AR, Video Annotation, Image Recognition

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented and virtual realities; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking; J.7 [Computer Applications]: Computers in Other Systems—Real time

1 INTRODUCTION

Placing advertisements in public environments is a multi-billion dollar business. Traditional advertising is based on placing large billboards over highways, next to streets, or just in shop-floor windows. Due to the digital disruption of our everyday's life, the trend in advertising goes from placing big static posters and large billboards to installing fully digital screens and flexible and interactive displays. This gives rise to new and interesting opportunities using Augmented Reality (AR) to bring the actual advertising content to life and to engage the observer.
Augmented Reality frameworks like VUFORIA™, or KUDAN™focus on tracking 2D static targets robustly, but are neither designed for large-scale image recognition nor for handling large target databases on mobiles. Using cloud-based recognition is prohibitive as latency basically kills the AR experience. Other approaches from the AR community are based on modifying the video content in advance to introduce features for tracking, but due to the degrade of video quality this is prohibitive either.
Our concept is based on the direct recognition of video frames and a fast indexing and efficient storage method. We process hours of video and compress the information into databases with a very small memory footprint, at the same time still maintaining real-time indexing performance and 6DOF trackability of the frames on mobiles. In this demo we show how our system performs on an entire season of a popular TV series, which resembles more than 16 hours of video and an overall amount of 60GB of frame data, compressed into a mobile database of just around 100MB.

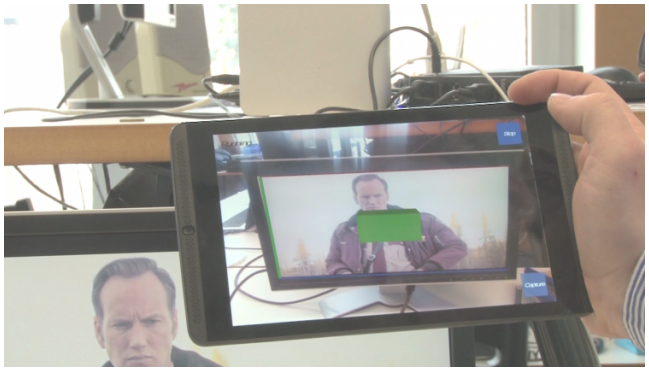

Figure 1: AR annotation on a video streaming on a TV set. The database for the video has a small footprint and is fully contained on the mobile device. Upon registration, an appropriate augmentation is added to the respective screen.

2 RELATED WORK

W.r.t.our approach, relevant work on image recognition is mainly based on the concept of local distinctive features, including SIFT by Lowe [5] or SURF by Bay *et al*. [2], amongst other approaches elaboratively described and evaluated by Mikolajczyk *et al*. [7, 9]. Rapid histogram comparison was investigated in the late 90s already by Arya *et al*. [1], however, the basis for a plethora of work used nowadays is an efficient tree-based method to rapidly index into a huge database by Nister [8]. The other group of related work concerns the use of mobiles and the use of imperceptible markers. Woo *et al*. [10] describes a method to place imperceptible barcode markers into videos, while Celozzi *et al*. [3] give a method using the overlay of beamer images is discussed. Sampaio *et al*. [6] describes a method based on random dot markers embedded into the live image in an imperceptible way.
Our approach is similar in nature to [8], but relies on other features and uses adaptions to the indexing scheme to enable full 6-DOF tracking in AR, improving on the ideas in [4]. Moreover as opposed to all other approaches mentioned above, our approach is still fully mobile and real-time, but does not require the alteration of the video content.

3 DEMO DESCRIPTION

The core concept of our solution is to use a rapid indexing engine on a large database of images to index in real-time and to provide tracking information for the respective image for further use of AR augmentations (see Fig. 3 for an overview of the framework). This works for arbitrary images, however, we demonstrate that this scheme even works for huge collections of frames of videos, as described in the following.
To create the database for videos, we first extract the keyframes and extract local features for each individual keyframe. In videos

*e-mail: philipp.fleck@icg.tugraz.at
†e-mail: arth@icg.tugraz.at
‡e-mail:schmalstieg@icg.tugraz.at

Figure 2: Exemplary screenshots running the recognition and tracking on frames from a 16-hour video of a popular TV series. On the top of each frame, the label of the frame in the database is denoted in pink. The cube is augmented to prove the trackability of the frames.

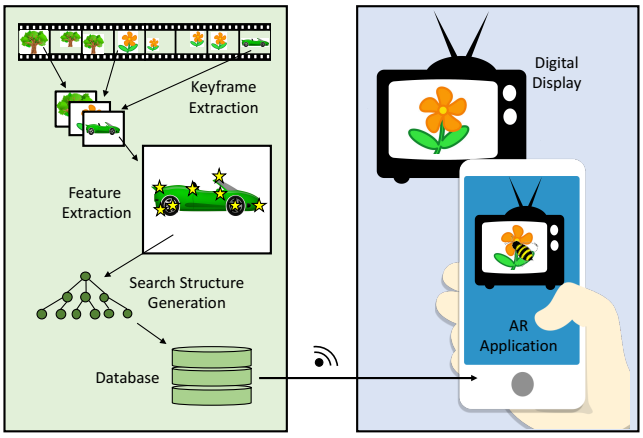

Figure 3: Application Concept. On the left, the keyframe extraction and database creation is illustrated including feature extraction and search structure generation. The database is transferred to an AR app, which can recognize frames from the live video and annotate AR content.

# Images	Size of Database [bytes]	# of Features
21	673 k	approx. 9.8 k
100	3 M	approx. 44.8 k
8000	128 M	approx. 3.6 M
10000	143 M	approx. 4.5 M

Table 1: Database size for a number of images used in evaluations ([k]=10^3, [M]=10^6).

using standard compression techniques like H264, keyframes are enumerated as the starting frames of individual shots. From the features extracted from all keyframes, we create a database using some ideas from [4]. The database itself is very lightweight, as indicated in the exemplary results in Tab. 1 and easily fits on mobiles, even for very large numbers of images. The database size does not increase linearly with the number of images, as the internal structure is tree-based.

For mobile application, the database is deployed on Android or iOS phones as part of a Unity3D application. During runtime, the actual live image acquired with the mobile device camera is recognized in the database and the tracking is initialized based on the matched features. Note that the time for recognition is constant for each individual frame given a certain database size. For example, it is about 30*ms* for a 10k database on a Samsung Galaxy S6 or an Apple iPhone 6S.

As the video content changes over time, the tracker is updating the feature tracks to maintain tracking during a shot, until a new keyframe arises. This leads to tracking of frames across shots, even if not each individual frame of a video is indexed. Some exemplary snapshots of the system running on a 16-hour video from a popular TV series is shown in Fig. 2.

4 Application in Advertising

A simple possibility to employ this technology in non-AR advertising is to distribute coupons as visitors engage with some digital screen. In shopping malls videos displayed usually have a length of some minutes and run in looping mode. Databases of small size can easily be distributed to visitors within mobile apps, like news articles are fetched in newspaper apps.

As the video can be tracked as well, more evolved options for AR include personalized advertising, *e.g.* recoloring objects in the videos based on personal preferences. However, a challenging but very interesting idea is to enable the user of becoming an actor in a video shot, which was originally shot in 3D, using an avatar acquired with some 3D sensing device like Kinect.

References

[1] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, Nov. 1998.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *CVIU*, 110(3):346–359, June 2008.

[3] C. Celozzi, F. Lamberti, G. Paravati, and A. Sanna. Enabling human-machine interaction in projected virtual environments through camera tracking of imperceptible markers. *Int. J. Hum. Comput. Interaction*, 29(8):549–561, 2013.

[4] A. Hartl, D. Schmalstieg, and G. Reitmayr. Client-side mobile visual search. In *VISAPP*, pages 125–132, 2014.

[5] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, Nov. 2004.

[6] S. Luiz, Y. Yoshio, Y. Goshiro, T. Takafumi, S. Christian, and K. Hirokazu. Detection of imperceptible on-screen markers with unsynchronized cameras. *IPSJ SIG Notes. CVIM*, 2015(64):1–4, jan 2015.

[7] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 65(1):43–72, 2005.

[8] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

[9] C. Schmid and K. Mikolajczyk. A performance evaluation of local descriptors. *ICPR*, pages 257–263, 2003.

[10] G. Woo, A. Lippman, and R. Raskar. Vrcodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter. In *ISMAR*, pages 59–64, 2012.