Towards User Perspective Augmented Reality for Public Displays

Jens Grubert[1], Hartmut Seichter[2], Dieter Schmalstieg[3]
[1,3]Graz University of Technology, [2]University of Applied Sciences Schmalkalden

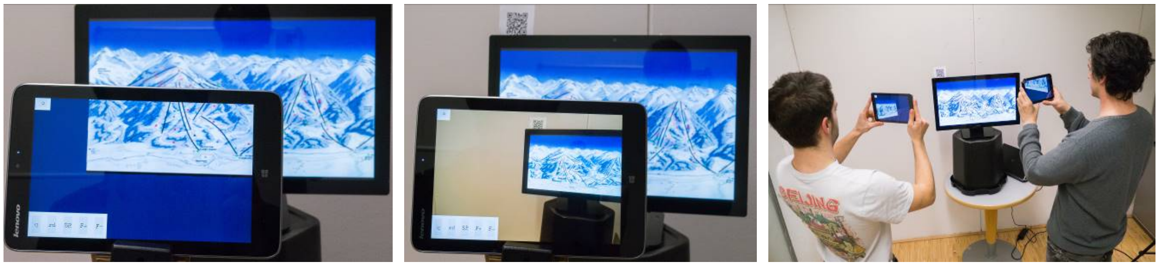

Figure 1: (Left) Simultaneous 3D head tracking and natural feature tracking enable user perspective augmented reality for digital display content. (Middle) Device perspective rendering usually found in handheld augmented reality devices. (Right) The system requires only access to a remote screencast and is otherwise self-contained, making it suitable for multiple users.

Abstract

We work towards ad-hoc augmentation of public displays on handheld devices, supporting user perspective rendering of display content. Our prototype system only requires access to a screencast of the public display, which can be easily provided through common streaming platforms and is otherwise self-contained. Hence, it easily scales to multiple users.

Keywords: user perspective rendering; public displays; augmented reality

Index Terms: H.5.2 [User Interfaces]: Graphical User Interfaces

1 Introduction

Letting users interact with public displays through handheld devices is compelling for public-private display interaction. When interacting with these displays through augmented reality (AR), user perspective rendering (UPR) can be beneficial, specifically for use cases where the surrounding visual context is important for interaction (e.g., public maps).

UPR approaches for general scenes rely on distorting the video feed of the back-facing camera [4] or using coarse 3D reconstructions [1]. Both approaches can suffer from visual artefacts, as the acquisition of real world data through cameras or reconstruction is imperfect. By comparison, UPR relative to known digital content on large screens can deliver high visual quality. However, previous systems have required considerable infrastructure for interaction with large digital screens. For example, Spindler et al. demonstrated UPR interaction on a tabletop system albeit requiring external tracking systems and projectors [3]. Baur et. al demonstrated tracking of screen content using handheld devices [2] but required a dedicated external tracking server which was limited in the supported number of concurrent users and which is hard to deploy in real-world settings. In contrast, we contribute the first system that allows user perspective AR for public displays, showing dynamically changing content at interactive frame rates without the need for additional infrastructure. Our prototype goes beyond previous work as it only needs access to a remote screencast of the display content and is otherwise self-contained. Hence, it naturally scales to multiple users.

2 Implementation

We implemented our system as depicted in Figure 2. It runs self-contained on mobile devices with dual camera access.To enable interaction between a mobile device and a situated screen, the system needs access to a screencast (e.g., via a the Real-Time Streaming Protocol) and information about the physical extent of the screen. This is communicated to the client via a QR code attached to the bezel of the public display. In our prototypical implementation we employed the FFMPEG software to capture and wirelessly screencast content at a resolution of 1920x1080 pixel via MPEG transport stream.

For tracking the screen content we employ a proprietary natural feature tracker (NFT). Depending on the computational capabilities of the mobile device, there are several choices for re-initialization when the display content changes. The simplest approach re-initializes the tracking system for every frame. However, this is wasteful as it ignores frame-to-frame coherence. A more efficient approach uses fast image differencing and updates only parts of the NFT model that have significantly changed. Generation of a new or updated NFT model is computationally expensive and introduces unwanted latency, potentially resulting in jerky motion or intermittent loss of tracking. The generation process can therefore be masked by interleaving it with tracking the current model, and amortizing the generation of the new NFT model over several frames.

For tracking the user's face, we combine a 2D deformable face tracker with a solver for the perspective-n-point problem. In a first step, 2D image points of facial landmarks are estimated using deformable model fitting. For the second step, we use a rigid 3D model which is mapped to selected image points of the 2D model (eyes, nostrils, temples).

[1]jg@jensgrubert.de, [2]seichter@fh-sm.de, [3]dieter@icg.tugraz.at

Author's version.

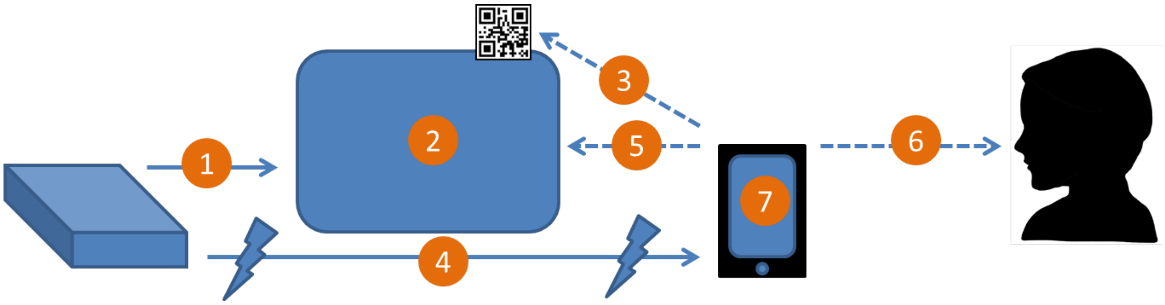

Figure 2: Overview of the system workflow. (1) A content source such as a PC sends a video signal to a public display. (2) The public display shows the corresponding image. (3) The handheld device decodes a QR code to determine the screencast channel. (4) A screencast hardware or software multicasts the video signal to a wireless network. (5) The handheld device tracks the location of the public display with the back-facing camera (6) and the location of the user's face with the front-facing camera. (7)

Rendering the screen content from the user's perspective is done in two stages. In the first stage, the screencast image is written to an OpenGL texture buffer and texture mapped to a quad, which is positioned according to the tracking data of the back-facing camera. In the second stage, this scene is rendered to texture from a camera position estimated by the 3D head tracker, while only considering the translation and assuming an orientation towards the screen center. In the main rendering pass, we use projective texture mapping from the same viewpoint to map the previously captured scene onto a screen-aligned quad, which has the extent of the mobile device screen.

Approximated user-perspective rendering can be achieved given the approach described above. However, to ensure the best registration quality, further adjustments should be considered. First, the intrinsics (i.e. focal length, principal point and skew) and the relative poses of the back and front facing cameras should be estimated. The relative poses of the cameras towards each other and the screen center can either be estimated manually by measuring the offsets between the camera centers and the screen center or more accurately by multi-camera calibration. Finally, the landmark points on the user's face should be calibrated, so that the distance from device to head can be correctly measured, and the focal length of the virtual camera used in rendering can be scaled accordingly. In practice, we use the interpupillary distance as a scale parameter. If the interpupillary distance is unknown, for example, when multiple users share a device, the focal length of the virtual camera must be set manually.

3 SAMPLE APPLICATION

We implemented a prototype application (Figure 1), which resembles public display content typically found in modern skiing resorts. A public display shows the status of ski slopes as well as weather information. Passers-by can initialize interaction by pointing their mobile device towards the QR code attached at the public display. The display visualizes their personal lift rides and the total number of kilometers they have skied (see accompanying video). Screencast via multicast networking allows an arbitrary number of users to interact concurrently with the public display without performance degradation.

4 PERFORMANCE AND IMPROVEMENTS

Our prototypical implementation runs at approximately 13 fps on a Dell XPS12 tablet with Intel i7-3667U processor and 8GB RAM, Intel HD4000 GPU at 1920x1080 pixel resolution. The average runtime performance of individual modules is as follows: NFT detection: 10 ms, NFT patch tracking: 4 ms, 2D face tracking: 17 ms, 3D head pose estimation: 0.12 ms, all of which happen in the scenegraph event traversal. The render traversals take 50 ms.

Our system can be improved in several ways. To increase the robustness of the natural feature tracking system, alternative keypoint descriptors or more iterative predictive tracking approaches can be incorporated. Also, we currently do not employ all facial feature points provisioned by the 2D face tracker. By learning new face models targets at runtime, we could drastically reduce the number of tracked 2D points, thus increasing performance.

Another limitation concerns the physical setup of the front facing cameras in some handheld devices. If the field of view of the camera is too narrow or the camera is placed at a corner of the bezel, the face tracker might not be able to track faces centered in front of screen in typical interaction distances (~30-50 cm).

Moreover, UPR of digital screen content could be combined with real-world content acquired through 3D reconstructions [1] or image-based warping [4] (with the associated visual artifacts). This would allow to extent the spatial scope of interaction beyond the boundaries of the public displays.

5 CONCLUSION

We demonstrated AR interaction with situated displays as a feasible way to enrich the real world at interactive frame rates. Our key observation is that our system is solely dependent on lightweight screencasting and "second screen" channels. All time-critical computations are performed locally on the handheld. Consequently, it overcomes the need for special infrastructure and naturally scales to multiple users. In the future, we want to extend the system for ad-hoc multi-display interaction, enabling rich interaction spaces beyond a single display.

ACKNOWLEDGMENTS

This work was supported by the EU FP7 project MAGELLAN under the grant number ICT-FP7-611526.

REFERENCES

[1] J D. Baricevic, C. Lee, M. Turk, T. Höllerer, & D. Bowman. A hand-held AR magic lens with user-perspective rendering. In Proc. ISMAR 2012, pp. 197-206. IEEE.

[2] Baur, D., Boring, S., & Feiner, S.. Virtual projection: exploring optical projection as a metaphor for multi-device interaction. In Proc.CHI 2012, ACM (2012), 1693-1702.

[3] M. Spindler, W. Büschel & R. Dachselt. Use your head: tangible windows for 3D information spaces in a tabletop environment. Proc. ITS 2012, pp245-254. ACM.

[4] M. Tomioka, S. Ikeda & K. Sato. Approximated user-perspective rendering in tablet-based augmented reality. In ISMAR 2013, pp. 21-28, IEEE.