LabDesignAR: Configuring multi-camera motion capture systems in augmented reality

Citation for the original published paper (version of record):
Baytas, M., Yantaç, A., Fjeld, M. (2017). LabDesignAR: Configuring multi-camera motion capture systems in augmented reality. Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST, Part F131944. http://dx.doi.org/10.1145/3139131.3141778

(article starts on next page)

Check for updates

LabDesignAR: Configuring Multi-Camera Motion Capture Systems in Augmented Reality

Mehmet Aydın Baytaş
Qualisys AB
Gothenburg, Sweden
Koç University
Istanbul, Turkey
mbaytas@ku.edu.tr

Asım Evren Yantaç
KUAR
Koç University
Istanbul, Turkey
eyantac@ku.edu.tr

Morten Fjeld
t2i Interaction Laboratory
Chalmers University of Technology
Gothenburg, Sweden
fjeld@chalmers.se

ABSTRACT

We present LabDesignAR, an augmented reality application to support the planning, setup, and reconfiguration of marker-based motion capture systems with multiple cameras. LabDesignAR runs on the Microsoft HoloLens and allows the user to place an arbitrary number of virtual "holographic" motion capture cameras into an arbitrary space, in situ. The holographic cameras can be arbitrarily positioned, and different lens configurations can be selected to visualize the resulting fields of view and their intersections. LabDesignAR also demonstrates a hybrid natural gestural interaction technique, implemented through a fusion of the vision-based hand tracking capabilities of an augmented reality headset and instrumented gesture recognition with an electromyography armband. The source code for LabDesignAR and its supporting components can be found online.

CCS CONCEPTS

• **Human-centered computing → Mixed / augmented reality**;
• **Applied computing → Computer-aided design**;

KEYWORDS

LabDesignAR, motion capture, augmented reality, HoloLens, gestural interaction, natural interaction

ACM Reference format:
Mehmet Aydın Baytaş, Asım Evren Yantaç, and Morten Fjeld. 2017. LabDesignAR: Configuring Multi-Camera Motion Capture Systems in Augmented Reality. In *Proceedings of VRST '17, Gothenburg, Sweden, November 8–10, 2017,* 2 pages.
https://doi.org/10.1145/3139131.3141778

1 INTRODUCTION

Marker-based motion capture with multiple cameras is a technology that enables precise position tracking for a variety of applications. These include animating characters for movies and games; sports performance analysis, the diagnosis and treatment of musculoskeletal and neurological disorders, and closed-loop control of mechatronic systems, among others. Motion capture can also power immersive virtual and augmented reality (VR/AR) systems such as "free-roaming" gaming setups, where precise tracking is required for multiple bodies over large distances [Bloem 2017].

During motion capture measurements, physical markers (e.g. reflective spheres) are attached to the subject, which must always be visible to at least two cameras. ('Missing' markers will cause 'gaps' in the data, which must be corrected during post-processing.) Ensuring adequate coverage is important when planning a new motion capture setup, or when reconfiguring an existing system (cf. [Allen and Welch 2005]). Care must be taken to choose the correct hardware and positioning, taking into account the expected movement patterns and the constraints of the space. For this purpose, recently, vendors of motion capture equipment provide interactive web applications that let users build a basic model of their capture environment and experiment with camera placement[1]. Designs created with these applications can serve as blueprints for subsequent sales and installation processes. Moreover, cameras may be moved, added, or removed after the initial setup. This also can be laborious, since cameras can be difficult to reach (e.g. high or underwater) and must be secured tightly after each adjustment.

To address the challenges in planning and reconfiguring multi-camera motion capture systems, we designed and developed LabDesignAR – an AR application to support the selection, placement, setup, and reconfiguration of marker-based motion capture systems with multiple cameras. LabDesignAR runs on the Microsoft HoloLens and allows the user to place an arbitrary number of "holographic" motion capture cameras into a real, arbitrary space. The cameras can be arbitrarily moved and rotated, and different lens configurations can be selected to visualize the resulting frusta and capture volume coverage.

LabDesignAR demonstrates an AR-based solution to challenges in a field that serves to enable VR/AR applications. It also demonstrates a 'hybrid' technique for natural gestural interaction [Yin and Davis 2010], where the user's hands are tracked using an electromyography (EMG) sensor worn on the forearm, along with the depth cameras on the AR headset. The application and its source code are freely availabe online[2], under a permissive license. The components used in the application to enable networking between a motion capture system[3], an EMG armband[4], and an AR headset[5] are also similarly open-sourced.

VRST '17, November 8–10, 2017, Gothenburg, Sweden

ACM ISBN 978-1-4503-5548-3/17/11.
https://doi.org/10.1145/3139131.3141778

[1] E.g.: labdesigner.qualisys.com
[2] github.com/mbaytas/LabDesignAR
[3] github.com/mbaytas/QualisysUDP
[4] github.com/mbaytas/MyoUDP
[5] github.com/mbaytas/HoloLensUDP

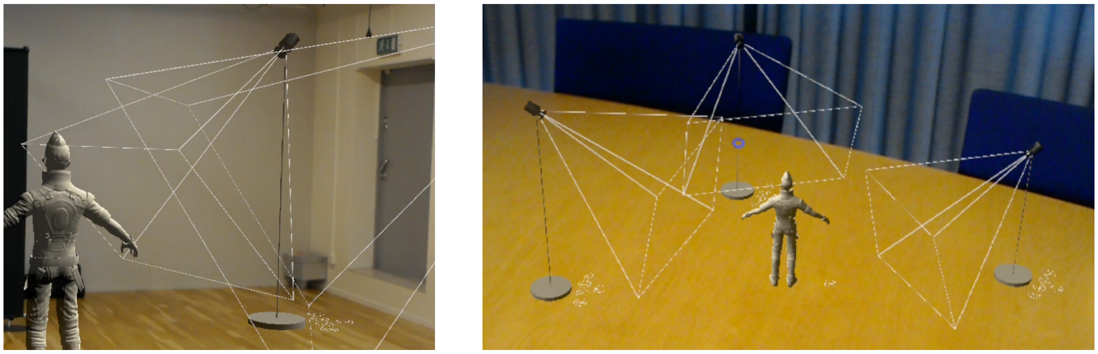

Figure 1: LabDesignAR running in 1:1 scale (left) and in 1:10 scale 'tabletop mode' (right).

2 USING LABDESIGNAR

LabDesignAR is designed principally to enable configuring a prospective motion capture setup in situ during pre-sales, which involves deciding on the number of cameras, selecting the appropriate lenses, and drafting a plan for positioning the cameras. It can also aid reconfiguring an existing motion capture system, which can involve re-positioning, adding, or removing cameras.

LabDesignAR greets the user with a control panel that hovers around the user, and contains buttons for functionality such as adding/removing camera holograms and adjusting the extent of camera frusta. The AR headset scans the space around the user and continuously updates its model of the environment. Camera holograms afford direct manipulation; they have a 'base' that responds to 'tap' gestures and can attach to any horizontal or vertical surface in the scanned environment. The cameras themselves respond to 'tap and hold' gestures for adjusting their height and yaw, as well as a 'fist' gesture (recognized by the EMG armband) that allows users to 'grab' the camera and adjust its roll and pitch with natural forearm movements. For each camera, the geometry of its frustum can be changed by selecting different options for its lens. The options that are currently available in LabDesignAR are based on the Qualisys Miqus camera range[6], but can be expanded to support any camera.

The application also has a 'tabletop mode' where holograms in 1:10 scale can be added to the scene, which can be used to obtain a better overview of the configuration. In both 1:1 and 'tabletop' scales, is also possible to insert a human model into the scene to verify the size of the capture volume covered by the camera frusta.

3 IMPLEMENTATION DETAILS

LabDesigAR has been built using the Unity game engine[7], and runs on the Microsoft HoloLens (Development Edition)[8] hardware with the Microsoft Windows Mixed Reality[9] operating system. We used components from the open source HoloToolkit-Unity project[10] to enable real-time spatial mapping and uninstrumented gestural interaction. We implemented instrumented gestural interaction using a Myo armband[11]. The electromyography sensors on the armband are able to provide the data used to detect a closed fist, while its inertial measurement unit streams arm motion and orientation data. The armband is connected via Bluetooth to seperate host computer that communicates with the HoloLens over Wi-Fi. A Windows application built using the open source MyoSharp C# wrapper over the Myo SDK[12] listens to gesture and orientation data from the armband and streams these to the HoloLens.

4 CONCLUSION AND FUTURE WORK

We presented LabDesignAR, an open source AR application to support the planning, setup, and reconfiguration of marker-based, multi-camera motion capture systems. Source code for the application and its supporting components are available online, as a resource for AR designers and developers.

Our experience in building LabDesignAR highlights opportunities for improvement in AR technology and developer tools. In an attempt to visualize the frusta and coverage of real cameras in an existing motion capture setup, we used the open source Qualisys Real-time SDK[13] to stream information in real time to the headset. However, we could not align holographic frusta with real cameras, due to large and non-linear discrepancies between the tracking systems of the AR and motion capture hardware. We also experimented with calculating and displaying an accurate real-time 3D model of the capture volume that is 'seen' well by the cameras. We are confident that, in the future, AR hardware and development tools will evolve to enable more diverse interactive content.

REFERENCES

B. Danette Allen and Greg Welch. 2005. A General Method for Comparing the Expected Performance of Tracking and Motion Capture Systems. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '05)*. ACM, New York, NY, USA, 201–210. https://doi.org/10.1145/1101616.1101658

Noah Bloem. 2017. Zombies?! Surviving the undead legion at the VR Arcade Amsterdam. *DutchReview* (5 2017). Retrieved August 17, 2017 from https://dutchreview.com/dutch-reviews/places/zombies-surviving-the-undead-legion-at-vr-arcade-amsterdam/.

Ying Yin and Randall Davis. 2010. Toward Natural Interaction in the Real World: Real-time Gesture Recognition. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction (ICMI-MLMI '10)*. ACM, New York, NY, USA, Article 15, 8 pages. https://doi.org/10.1145/1891903.1891924

[6] qualisys.com/cameras/miqus
[7] unity3d.com
[8] microsoft.com/hololens
[9] developer.microsoft.com/windows/mixed-reality
[10] github.com/Microsoft/HoloToolkit-Unity
[11] myo.com
[12] github.com/tayfuzun/MyoSharp
[13] github.com/qualisys/RTClientSDK.Net