HoloLens AR – Using Vuforia-based marker tracking together with text recognition in an assembly scenario

Sebastian Knopp[1]
Institute for Machine Tools and Production Processes, Chemnitz University of Technology

Philipp Klimant[1]
Institute for Machine Tools and Production Processes, Chemnitz University of Technology

Robert Schaffrath[2]
Fraunhofer Institute for Machine Tools and Forming Technology IWU

Eric Voigt[2]
Fraunhofer Institute for Machine Tools and Forming Technology IWU

Rayk Fritzsche[2]
Fraunhofer Institute for Machine Tools and Forming Technology IWU

Christoph Allmacher[1]
Institute for Machine Tools and Production Processes, Chemnitz University of Technology

ABSTRACT

Many real world applications for Microsoft HoloLens*-based applications suffer the problem of reliably recognizing and identifying movable objects within an environment. While the HoloLens is perfectly able to discern already known rooms, it still has troubles with reflecting surfaces or identically shaped objects. Using dedicated recognition libraries for each task poses the issue of shared resource access in the rather controlled HoloLens environment- In this poster we present a solution for scenario with hard to track objects and similarly shaped objects for an electrical cabinet assembly task, where the reflective cabinet is tagged with a marker and the prefabricated cables are differentiated by text-based labels.

Keywords: HoloLens, AR, Vuforia, Text recognition, Application showcase.

Index Terms: Applied computing → Industry and manufacturing; Human-centered computing → Augmented Reality; Computer systems organization → Distributed architectures

1 INTRODUCTION

Every Augmented Reality (AR) application is dependent on reliably recognizing its surroundings and specific objects within the environment, by utilizing limited and preferably portable hardware resources. The HoloLens itself is equipped with a HD video camera, a time of flight depth camera and four environment-understanding cameras [1]. With those, it can recognize an already known environment, localize itself in that environment and project virtual objects into the field of view of the user, fitting this environment. However differentiating multiple objects of similar or identical shape, like interchangeable or mass produced parts, require a third party library to identify additional information labelled on these parts. While there are various image-processing libraries available, which range from image detection over object detection to text recognition, due to different system and architecture requirements, not all of them can run on the hardware of the HoloLens itself. In cases where different libraries are required to run simultaneously, race conditions with access right to hardware resources like the video camera will hinder a continuous process, as soon as one library monopolizes a resource. Additionally image processing is a rather computation-heavy process that might exceed the hardware specifications of the HoloLens.
For these reasons, it can be necessary to have some of the processing tasks done remotely on a more capable CPU.

[1] name.surname@mb.tu-chemnitz.de
[2] name.surname@iwu.fraunhofer.de

* https://www.microsoft.com/en-us/hololens

2 REMOTE INTEGRATION OF IMAGE PROCESSING LIBRARIES

Due to the limited computational power of the HoloLens and similar head-mounted AR glasses it is often imperative to run complicated algorithms on hardware that is more potent. For access to a remote system, the HoloLens offers three possible ways of communication. Cable-based (USB), Bluetooth and WLAN. While the connection via USB-cable offers enough bandwidth with short latency it is always limited by the length of the cable and defeats the purpose of a wearable, mobile device. Therefor we do not propose it as a viable solution for the problem.
While the low power-usage of Bluetooth 4.1 LE is appealing given the HoloLens is only fitted with a 16.5Wh battery, the low data rate of one to three Mbit/s poses an unnecessary restriction on naturally lager bandwidth requirements while sending image streams. In terms of development, the Bluetooth feature is focused on the connection of peripheral input devices. [1]
Therefore, this work targeted the Wireless Local Area Network (WLAN - 802.11ac) communication built in the HoloLens. The standardized communication via a WLAN allows for implementation of the image-processing library on any system connected to the same network or even throughout the internet. This allows for more flexibility in hardware and architecture requirements during the development of the library.

3 REFERENCE IMPLEMENTATION

3.1 Use Case

A practical application of image processing in a HoloLens application would be a manual assemble task of a larger number of similarly looking and shaped objects in a mass production environment. In our project “Automatischer Schaltschrankbau (ASSB)” *(engl. Automatic manufacturing of an electrical cabinet)* the task is the assembly of an electrical cabinet where all components are already prefabricated, e.g. all cables have been cut to a fitting length and differentiated by text-based labels, according to the layout plan. For the identification of the type of electrical cabinet, we chose a QR-code based image marker. During the assembly, the user wearing the HoloLens holds the label of the next cable into the field of view of the HoloLens’ main video camera and in return is presented with a graphical representation of where to place and connect the respective cable. The combination of mass production of standardized or prefabricated parts with automatic detection and usage indication of these parts during the manual assembly, tackles the complexity issues of a larger product variety in assembly systems [2]

978-1-7281-4765-9/19/$31.00 ©2019 IEEE
DOI 10.1109/ISMAR-Adjunct.2019.00030

3.2 User Interface Concept

The user interface concept is centred on the hands free interaction principle [3]. The goal is to require as little as possible user input to keep the focus on efficiently solving the task. For that purpose, a menu was designed and placed next to the currently worked on cabinet to provide information about the part recognized most recently. Further information like reached progress and hints for less obvious steps can be displayed there. The only break with the hands free design is the confirmation of successful completion of a step, which cannot be determined automatically with current hardware. While this confirmation could also have been designed hands free, by looking at the confirmation button with the centred cursor for a specific amount of time, we decided against this option, because it disturbed the workflow in an unnatural way, at a time where the workers hand are not occupied anyway [4]. Therefore, the usage of the click-gesture already implemented in the HoloLens was deemed the more efficient input interaction. Furthermore, the manual step confirmation can be used to gather statistical data about the assembly process.
During steps, the search for known parts in the user's field of view is conducted automatically without further user input.

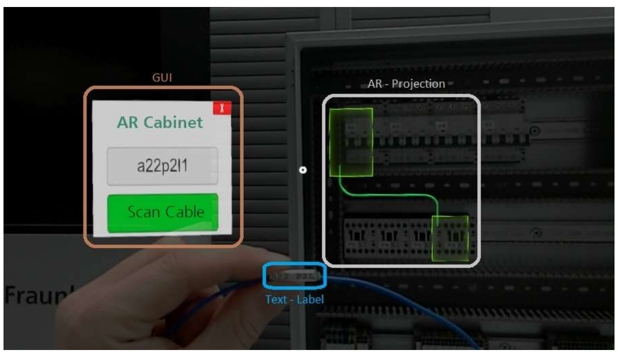

Figure 1: View through the HoloLens after recognizing a text-label on a cable

3.3 HoloLens Integration of Remote Cable Recognition Library

The only viable development and runtime environment for the HoloLens is Unity and in that the widest spread library for marker tracking with HoloLens support is Vuforia. This library handles the pose calculation based on the marker with extended SLAM tracking after the marker left the field of view. All spatial positioning of 3D-object projection is handled within the library and the unity environment. While Vuforia easily works as a plug-and-play library, it also monopolizes the access right to the main video camera. The only options to get the current picture of this camera for further image processing is either shut down Vuforia for the time of the image capture or to request the image directly from the tracking library. We decided to work with the image provided by Vuforia to avoid loss of position during the frequent switches between native and library-based pose tracking. The provided image has a resolution of 896 x 504 pixels in the format of RGB888 with a total size of 1,3MB. This image is sent for further processing to a remote server via an http-PUT request. With a http-GET request the last result will be retrieved. For the networking implementation, UnityWebRequest was used. However, this feature can cause the bug of providing an error code for successfully received answers on a GET-request. Until this issue is fixed it would be preferable to rely on an own socket implementation.

In case a cable has been recognized, no further analysis queries will be sent to avoid unnecessary power consumption and accidental recognition of a different label entering the field of view and thus causing an augmentation of the wrong step.
Using the http standard allows for any number of server implementation in any given architecture as long as it has a contention to the same network or the internet and supports http. In our case we implemented a lightweight http-server in C++ and imported the text recognition library. As soon as a picture was transmitted successfully to the server, it will be analysed by a machine-learning algorithm for lines in the picture forming the shape of letters in a combination of previously taught, alphanumeric strings. The result of the last analysis can be requested from this server with a GET-request at any given time. Older results are deemed irrelevant and will be discarded after each new analysis. The result is either "nothing found" or the identifier and label-text that was recognized.
Even though we achieved a significant time reduction of the image processing, compared to running the entire analysis on the HoloLens itself, in future implementations a compression of the image stream will further reduce communication time and with that increase the efficiency of the system.

4 Conclusion

We presented a solution to transfer calculation-heavy algorithms like image processing away from the limited hardware of an see through HMD like the HoloLens to a more powerful, remote computer that is not fixed on the x86 architecture. This enables the option to use the HoloLens in a very flexible productive environment together with modern technologies like machine learning. This way some weaknesses of modern see-through-HMDs can be compensated by additional remote hardware without compromising the flexibility and mobility of the HMD. That adds to the possibilities of industrial HoloLens use collected in [5].

Acknowledgments

European Union
Europe funds Saxony.
EFRE
European Regional Development Fund

5 References

[1] Taylor, Allen G. "HoloLens Hardware." Develop Microsoft HoloLens Apps Now. Apress, Berkeley, CA, 2016. 153-159.

[2] Hu, S. J., et al. "Product variety and manufacturing complexity in assembly systems and supply chains." CIRP Annals-Manufacturing Technology 57.1 2008. 45-48.

[3] Witt, Hendrik, Tom Nicolai, and Holger Kenn. "Designing a wearable user interface for hands-free interaction in maintenance applications." Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on. IEEE, 2006.

[4] Haller, Michael, ed. "Emerging technologies of augmented reality: Interfaces and design. Igi Global, 2006. 34-42

[5] Aleksy, Markus, et al. "Utilizing HoloLens to Support Industrial Service Processes." 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA). IEEE, 2018.