Assembling LEGO set with augmented reality instructions

Project report
Object Recognition and Computer vision - MVA

Othman Sbai, Pierre-Alain Langlois

December 20, 2016

1 Introduction

The advent of augmented reality devices such as Microsoft Hololens, Sony SmartEyeglass or Google Glass and others have made possible many interesting applications that augment the visual experience of the user with 3D holograms that are blended on his reality. Applications range from interior decoration and design, gaming but also increasing productivity in businesses by enhancing the real world and giving birth to broader imagination.

Among the applications of this new promising technology still in development, one can be interested providing instructions for people to help them accomplish tasks either with human supervision or by annotating reality. In fact, as presented in [1], we can extract from the tutorial videos available online instructions for performing many tasks such as changing car tires, assembling furniture and also performing Cardiopulmonary rescucitation. These instructions can be efficiently provided to user with a 3D augmented reality device in the form of holograms and world annotations that are much comprehensible than paper instructions. An example of this world annotation is illustrated in the following figure 1.

2 Approach of the project

As a simplified approach to the problem providing holographic instructions as described before, we set to work on the assembly of a LEGO set using a Microsoft Hololens device to visualize the created 3D holograms and assessing the created experience.

2.1 Lego set description

In coordination with other group of students working on a similar project topic, we decided to choose the LEGO set named "Blue racer" (ref 31027) shown in figure 2

3 Introduction to Hololens device

3.1 Working with hololens

Microsoft Hololens is not a cheap platform to work on as it cost at least 3000$. Thankfully, there is the possibility of working on an emulator for hololens on Visual Studio. This requires a certain setup:

- Visual studio 2015 and Windows 10 Pro or Education (in order to support virtualization)
- Unity game engine

There are several interesting and good tutorials to start working with hololens emulator. The process of building an app for hololens requires:

- First creating and setting up the scene in Unity (camera position, background)

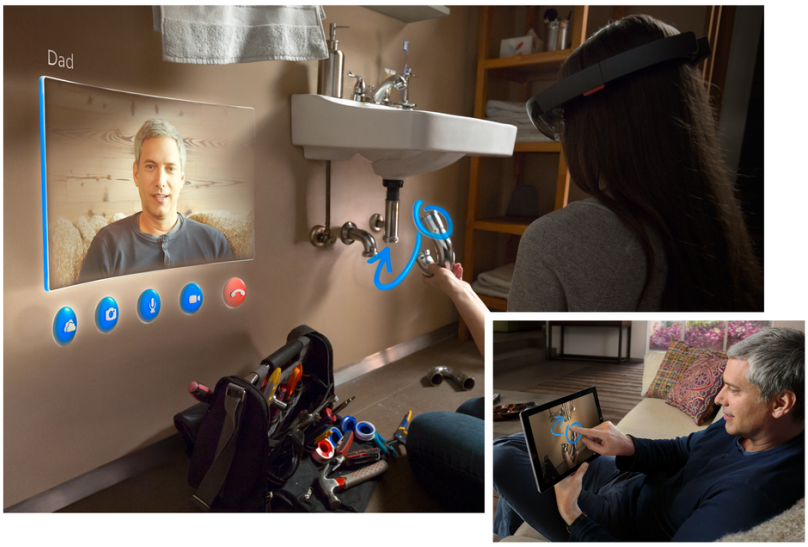

Figure 1: Annotating the world with Microsoft Hololens

Figure 2: Lego set chosen for this project

- Setting some build settings, and building the solution for visual studio
- Modifying the C# scripts
- Running the app on the emulator

Modifying the scene requires rebuilding the solution in Unity and reloading it in VS while modifying only the scripts requires only running the app in VS.

3.2 About Hololens device - input/output description

A hologram is an object made of light and sound, Hololens device renders 3D objects in precise locations of the world. It has a dynamic sensor-driven understanding of the world and updates the holograms depending on user head movement. In order to make the created holograms look realistic, the Hololens does a *spatial mapping* of the surrounding environment of the user which dynamically updates mapping of surrounding objects so that holograms can live relatively to the real world 3D objects.

HoloLens includes a world-facing camera mounted on the front of the device which will be our input for 2D object recognition and localisation (i.e the current LEGO assembly status and localisation of objects of interest). The captured frames provide RGB data as well as the location of the camera in the world, and the perspective projection of the camera.

3.3 Other inputs of the hololens

Hololens provides several interesting inputs that we can use in this project. The gaze is a form of input used in holographic apps in order to target object and act on them.

3.4 Mapping from 2D coord in hololens RGB camera to 3D coord in Unity scene

In order to put holograms in the correct place in the 3D scene in Unity and thus rendered in the right 3D location in user's vision, we need to map the pixel position on the RGB frame obtained from the hololens to the 3D coordinate in Unity Scene (or at least the ray)

See the section: "Pixel to Application-specified coordinate system" to find the 3D location of the hologram in the application coordinate system.

https://developer.microsoft.com/en-us/windows/holographic/locatable_camera

We have chosen the topic : *A.3 Instructions for assembling simple lego objects.*

The advent of augmented reality tools such as Microsoft Hololens have made possible many interesting applications that augment the visual experience of the user providing relevant informations and distractions. Across the web, one can find lots of tutorial videos for performing a certain task be it assemble a furniture, prepare a meal, change tire in addition to DIY videos. These explanations can really be enhanced through augmented reality technology by overlaying instructions in the view of Hololens for example in order to adapt the tutorial to the real world's configuration.

We propose to tackle the problem of providing instructions for assembling simple LEGO set. This simple game-related problem is a good start in manipulating and recognizing 3D objects from a head-mounted camera. Our goal is to create an assistant that is able to recognize the state of the LEGO mounting problem and suggest the next step by blending virtual movement on the reality perceived by the player thanks to Hololens.

We will first implement a system that allows to recognize and locate the pieces in the frame in real time [3]. This system will provide simple informations to help the user assembling the pieces.

If the system works well enough, we will try to add the pose estimation in the process in order to give more precise informations about the pieces orientation [4].

Some researchers already worked on the problem of LEGO brick identification and retrieval in realtime from 2D images[2]. In our case, we will generate data directly from the hololens (both RGB and geometrical data), since we have only a few lego pieces to recognize. If needed, we will also use data augmentation techniques.

4 Plan of work

- GOAL: Implement an interactive assistant that helps solving/assembling a LEGO set. Providing relevant statistics to evaluate the performance of the method.
- We assume the knowledge of the set of sequences required for the assembly, this can be either supplied by LEGO from the manual, or can be deduced from tutorial videos, as was done in the paper [1].
- From a head mounted camera, recognize LEGO part to be moved and its destination and display a hint of the movement overlaid on the hololens.
- If the previous work is good enough, we will try to add support for pose estimation and more advanced instructions.

5 Operational organization

5.1 Group members

- Othman Sbai (MVA & École des Ponts ParisTech)
- Pierre-Alain Langlois (MVA & École des Ponts ParisTech)

5.2 Plans for work sharing

Pierre-Alain will be focused on the work regarding the object detection (including segmentation) and the pose estimation task. Othman will be focused on the tracking constraints, and the implementation and experiments on the hololens device. Both of us will also perform testing on each other implementations in order to make the produced code more robust.

References

[1] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised Learning from Narrated Instruction Videos. *arXiv:1506.09215 [cs]*, June 2015. arXiv: 1506.09215.

[2] Leendert Botha, Lynette Van Zijl, and Mcelory Hoffmann. *Realtime LEGO Brick Image Retrieval with Cellular Automata*. July 2009.

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *arXiv:1506.02640 [cs]*, June 2015. arXiv: 1506.02640.

[4] Paul Wohlhart and Vincent Lepetit. Learning Descriptors for Object Recognition and 3d Pose Estimation. *arXiv:1502.05908 [cs]*, February 2015. arXiv: 1502.05908.