Enhanced AR Learning for Guitar Beginners with Meta 2 and Camera-based Tracking

Yicheng Li
ycli@stanford.edu

Zhenzhi Xia
zxia3@stanford.edu

June 2018

1 Introduction

We aim to create an enhanced learning experience for guitar beginners utilizing Meta 2 and web-cam-based tracking technique. We build the virtual environment displaying a 1:1 size guitar, music sheet and chords on top of the real world so that guitar players can still work on the real guitar so that the players are able to save substantial amount of time and effort from reading guitar tablatures or gazing back and forth between the guitar frets and paper sheet music.

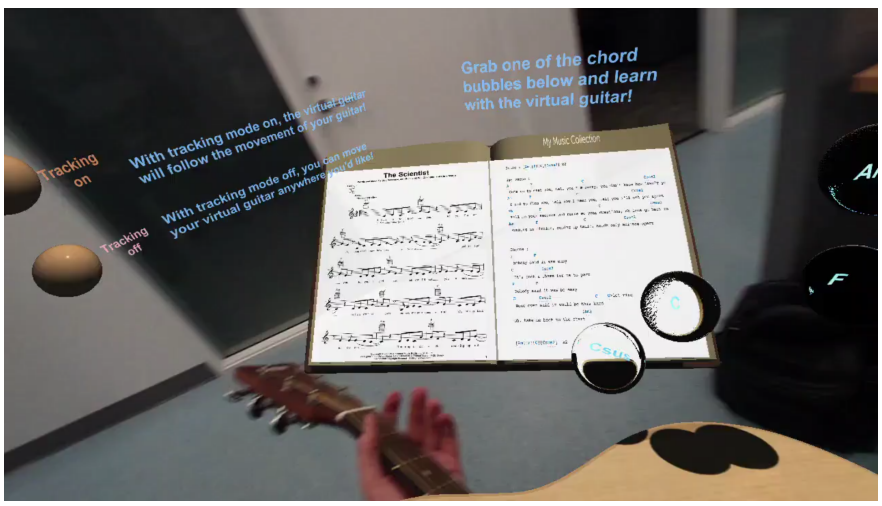


Figure 1: The virtual learning environment built on top of the real world

2 How to make Meta work

First, in order to run Meta SDK, a computer with Windows 10 and a powerful graphics card is required. The Meta SDK should be installed and used with an admin account. Any account that's not admin will not have the permission to run some part of the Meta SDK and therefore not be able to control Meta. Some machines, for example the machine connected to HTC Vive (as of June 2018) can open the Meta SDK, but the processing is too slow, which makes it impossible to develop on it.

Once the Meta SDK is installed with admin account, the first program to be run is "Meta Headset Setup". This will ask the developer to plug in Meta 2. The developer should strictly follow the instructions (i.e.

connect everything in the specified order). If the software says the HDMI port or USB port is not working, simply try another port. Once the headset setup process is complete, the developer should be able to see augmented display. Meta will ask the user to rotate head for environment mapping.

If the headset setup process is successful but there's nothing displayed on the headset, the developer should try unplug everything and start over again.

3 Camera-based Tracking

We want the displayed virtual guitar to follow the movement of the real guitar, which the user is holding. That way, if the user moves around, he/she can still see the virtual guitar. We use camera-based tracking. Specifically, we first stick a custom marker on the guitar, and then use a laptop to capture images of the guitar and compute the 6-DOF pose. The marker is a smartphone displaying four color squares in red, green, blue and white, respectively, as shown in the following figure.

(a) Marker. Yellow was replaced by white.

(b) Demo for camera-based tracking.

Figure 2: camera-based tracking

The mass center of each color square is one feature point. Altogether, we have 4 feature points, and for each of them we know their location on the guitar's coordinate system. We threshold out each color square in the captured image. For the red, green and blue rectangles, we convert the RGB image to HSV space, and look for pixels in the corresponding range. For the white rectangle, we convert the RGB image to grayscale and look for bright pixels. This process gives multiple candidate regions/shapes for each color we are tracking. We apply the constraint that candidate regions should have four edges, and a proper area. Regions that are too small or too large are discarded. We then compute the mean 2D position (global center) of all candidate regions, and select the candidate rectangle closest to the global center in each color channel. This is the constraint that the four color rectangles should be next to each other. In this way, the mass center of each color rectangle in the captured image is determined. Then the problem of tracking the guitar reduces to a Perspective-N-Points problem, and we solve it using the homography method. The computed translation and rotation vectors (6 values in total) are sent to the computer running Meta through a simple socket and used to update the position of the virtual guitar (in Unity). To make the tracking more robust, the average of 5 consecutive estimations of the pose, instead of individual estimations, is sent to Unity. The computation is not expensive and can be done in real time. Thus the laptop serves as a tracker. Note that this can be implemented using a raspberry pi with a simple camera. Our method proves to be effective and low-latency without using any complex computer-vision algorithm or specially designed hardware (i.e. a lighthouse).

The user can also turn off the tracking by grabbing a controller sphere displayed near the virtual guitar. When tracking is turned off, the virtual guitar's position is governed by Meta's "hologram locking"

system. The user can then move the virtual guitar to arbitrary positions or rotate the virtual guitar with his/her hands. Sometimes, Meta's locking isn't very stable, and the virtual guitar could drift away from its original position. In this case, our tracking system serves as a good complement because the homography method does not have drift.

4 Hand Interaction

4.1 Chord Selection

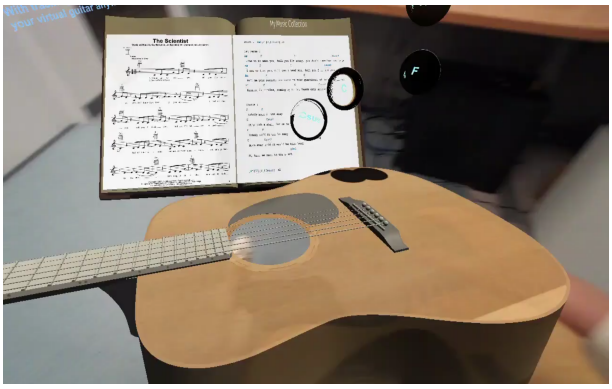
Instead of the traditional guitar tab, the players will now see the virtual bubbles that indicate the chord position,as well as the texts that gives the name of the chord, while in the regular guitar tab, the index of string, fret and fingering could be confusing from time to time. In addition, benefiting from the virtual display of a sheet music offered by Meta2, players don't have to keep raising their heads up and down while learning new chords or new music.

We utilize Meta's hand tracking system to support chord selection. 4 spheres are displayed in the AR environment, each representing a different chord. The user can simply use his/her hand to grab the sphere corresponding to the chord he/she wants to learn. The indication on the virtual guitar will change accordingly. This is achieved by setting only the relevant bubbles active (3 bubbles for C chord, for instance) in the callback function registered for "Disengage" status determined by Meta's hand tracking system.

4.2 Music Sheet Interaction

While the music sheet is initialized at a comfortable position, the user can control it in three ways: 1. use one hand to grab and move the music sheet; 2. use two hands to scale the music sheet; 3. use two hands to rotate the music sheet. All of these interactions are implemented using Meta's hand tracking API.

(a) chord position visualization



(b) virtual music sheet

Figure 3: Interaction with the virtual scenes

5 Future Work

Due to the time limit more features could have been realized. These include music sheet turning with hand gestures, feedback system to players whether they play the right chord and real-time auto-play of chord visualization with selected music. Furthermore, a virtual hand could be displayed on top of the virtual guitar, showing players the optimal fingering.