
HoloKeys - An Augmented Reality Application for Learning the Piano

Dominik Hackl
University of Applied Sciences
Upper Austria
4232 Hagenberg/Austria
Email: dominikhackl@gmx.at

Christoph Anthes
University of Applied Sciences
Upper Austria
4232 Hagenberg/Austria
Email: christoph.anthes@fh-hagenberg.at

Abstract—This paper describes the design and the implementation approach of a piano training application. HoloKeys is an Augmented Reality tool which is capable to superimpose the keys to be played on a real piano. Musical pieces are loaded as MIDI files, interpreted and can be displayed in two different ways. This prototype provides many possibilities for extension which can make it a powerful teaching tool.

I. INTRODUCTION

Augmented Reality (AR), described by Azuma as a technology where the user sees 'the real world, with virtual objects superimposed upon or composited with the real world' [1], has become a hot topic in the recent years. The application areas are wide spread and range far beyond simple advertisements and virtual manuals from advanced training to sophisticated remote collaboration scenarios. Using AR to train musical instruments has a long tradition in the field but because the rapid development in AR Head-Mounted-Displays (HMDs) this application area has gained new attention.

We present HoloKeys, a prototypical implementation of an AR training tool for learning the piano. HoloKeys runs on an HMD which the user is wearing while sitting in front of a physical piano. The application indicates notes that are supposed to be played by displaying virtual keys superimposing the physical keyboard with two different approaches. Acquiring the musical data dynamically by loading and processing MIDI (Musical Instrument Digital Interface) files, the application is fully agnostic considering the musical pieces to be trained. To achieve the required precision for the augmentations on the piano, the application was implemented using fiducial marker tracking. Since this application is a prototype, an extensive collection of possible enhancements and prospects for the future is given.

A. Outline

The remainder of this paper is structured as follows: The next chapter provides an overview of the related work in music teaching applications. Chapter III will introduce the conceptual design of the application describing the architecture and the user interface. Implementation details are provided in Chapter

IV. Finally conclusions are drawn and an outlook into the future work is given.

II. RELATED WORK

Music education has a long tradition in the field of AR. In an early approach Cheng and Robinson provided a visual sheet music overlay displayed planar in the visual field of the user. The display of the augmentations is triggered when he looks at the hands. The type of sheet is depending on which hand he looks. The augmentation is not registered (meaning it is not directly spatially interconnected) to a real object opposed to the approach presented in this publication. An HMD is used for display [2]. Cakmakci et al. augmented the information which string to pull on a guitar with the intention to reduce cognitive discontinuities compared to the traditional way of learning an instrument. They were the first to provide information on the interaction to be taken in an immediate way on an instrument [3]. The registration of the guitar and the virtual hand is implemented with the help of fiducial markers.

In order to avoid the use of fiducial markers on the piano Huang et al. use their knowledge on the application domain and track the keys of the piano for pose estimation with the help of natural feature recognition [4]. Unfortunately they provide no details on the display used, but the frame-rate of 15 frames per second, implies that it has not been developed for a head-tracked system.

Chow et al. focus on the educational level of AR piano teaching showing that with the help of augmentations and gamification components the motivation and interest in learning the piano could be increased. They provided a system illustrating the notes to be played by lines approaching the keys. Their findings also indicate that notation literacy does not increase using their system of illustration [5]. We use a similar approach for the augmentations of the notes to be played but rely on an optical see-through HMD instead of a video-based HMD.

Opposed to this visualisation approach Torres-Fernandez et al. introduce a virtual character which illustrates how well the piano player has performed. To interpret the played music they

compare the input from a MIDI keyboard with an initially loaded MIDI file [6]. A similar analysis was suggested and implemented earlier by Barakonyi and Schmalstieg [7]. They make use of fiducials for tracking and a desktop AR system equipped with a webcam and a traditional screen.

In terms of visualisation Weing et al. demonstrate a system in the area of Spatial Augmented Reality where they project the keys to be pressed directly on the piano. Different modes show for example the current and the next keys to be pressed. If a wrong key is pressed it is highlighted in red to provide feedback to the user [8].

Zhang et al. use a completely virtual keyboard and track the hand of the user with fiducial markers and the finger positions with a self-developed data glove. Their approach targets the rehabilitation of the motor function of stroke survivors rather than teaching the piano [9].

Compared to these existing and presented approaches our system is unique in terms of used display technology.

III. CONCEPTUAL DESIGN

The following chapter gives an overview of the application's hardware and software components and explains how the individual parts interact with each other.

A. Architecture Overview

The application's setup is illustrated in Fig. 1 and consists of the following two hardware components.

1) *The Piano*: The core component is a physical piano which is used for the actual playing. Underneath the piano keyboard which is usually made of 88 keys a fiducial marker is placed which is used by the application for tracking. The keys of a regular piano are standardized in size which makes the application fully independent considering the type of piano. In case a keyboard is used the key width can be adjusted.

2) *The Head-Mounted-Display*: The user sits in front of the piano and wears an HMD on which the application runs. Through the HMD the user sees augmentations in the form of highlighted keys on top of the real keyboard. The HMD also handles tracking by recognizing the image marker with the help of computer vision algorithms. The HMD therefore keeps track of the player's position and displays the augmentations accordingly. Additionally, the HMD is responsible for sound output of the music to be played. This gives the user an impression on how the piece is supposed to sound and makes it easier to play along with it.

B. Interface

In order to manage different settings and control the playback, a simple user interface was implemented. The originally two-dimensional UI is placed inside the 3D scene using world-stabilized coordinates. Considering the usually static setup of the application with the user sitting in front of the piano, the world-stabilized menu is a reasonable approach [10]. User input works through gaze-based interaction combined with gestures.

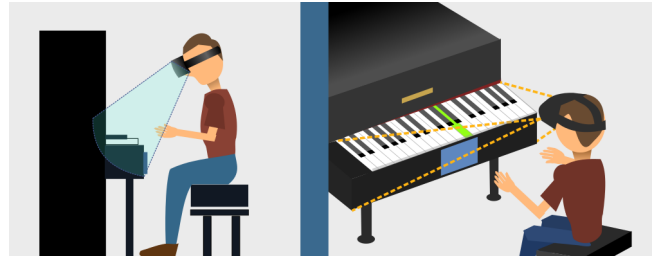


Fig. 1. Illustration of the conceptual design. The user, sitting in front of the piano and wearing an HMD, looks down at the keyboard. When there are notes to be played the respective key is highlighted. Underneath the keyboard there is an image marker which is used for tracking.

1) *The Main Menu*: The initial scene of the application is the main menu. There the user can select the musical piece to play as well as the desired playback speed. By pressing the start button the application will switch to playback mode and begin visualizing and playing the musical piece.

2) *Playback Mode*: In playback mode the user sees the augmentations of the keys to be played superimposing the physical keyboard. Additionally a timeline shows the current playback position and gives the user the option to jump to different positions inside the piece. With the pause button the user is able to interrupt the playback or return to the main menu.

3) *Calibration Mode*: In calibration mode the application displays an augmentation of only one key, the middle C. The user can adjust the position of the marker until the virtual key perfectly fits the real one. This is useful to setup the optimal position of the marker on the piano. Additionally the user can also adjust the pitch of the virtual piano sound in calibration mode because this does not necessarily match with the real piano. Playback volume can be adjusted in the HMD.

C. Display of Augmentations

Generally the HMD displays an augmentation of a bright green key to indicate that the actual key on that position has to be pressed. Two different approaches as seen in Fig. 2 were tested and both have their advantages and disadvantages concerning predictability and Field Of View (FOV) limitations.

1) *The Instant Approach*: The moment a key is supposed to be pressed it becomes highlighted. Once it is supposed to be released it switches back to normal. This way the user can more or less observe the playing of the piece in real-time, comparable to watch the fingers of an actual pianist. While this approach can be useful for advanced players, it is hardly possible to learn a new piece or even to play along with it, because the player has no way of predicting the next notes. Still, observing this looks great and could be used for showcase purposes (self-playing piano), as the limited FOV is also less of a problem there.

2) *The Beatmania Approach*: Note objects are created far in the distance and from there start moving towards the particular keys. As soon as the virtual object reaches the real key, the note should be played. With this approach, which became

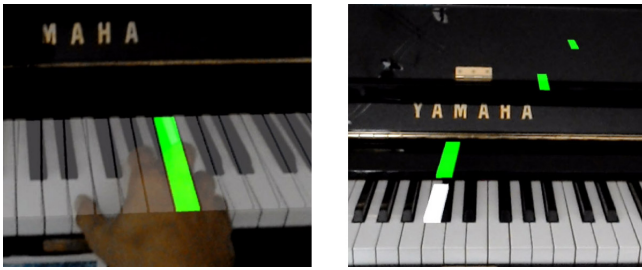


Fig. 2. Comparing the two tested approaches. Left: The Instant Approach. Right: The Beatmania Approach.

popular with the game 'Beatmania' [11] and is still used in many music rhythm games today, the user can anticipate the upcoming notes and prepare accordingly. When learning a piano piece the musician's brain utilizes its 'muscle memory' and fine motor skills rather than memorizing each individual note [12]. Therefore learning a piece with the Beatmania approach should be equally efficient than learning it from sheet music, especially for beginners.

IV. IMPLEMENTATION

This chapter goes into detail regarding the concrete implementation of HoloKeys. It starts with a brief overview of used hardware and software tools followed by an in-depth description of the two main development tasks, visualization and MIDI processing.

A. Used Technologies

The application was developed for tablet devices as well as the HoloLens. The tablet approach is mainly used for demonstration purposes, rather than actual training.

1) Hardware:

- HoloLens¹

The HoloLens as a current AR HMD provides good sensory support as well as spatial audio and stereoscopic display capabilities. Its main disadvantage the limited FOV poses an issue to the applicability of this use case.

2) Software: To allow cross-platform and cross-device development the following set of tools and libraries was used.

- Unity²

Unity is traditionally a game engine which has found wide adoption in the whole domain of Mixed Reality [13]. It allows scene setup and provides scripting capabilities. The applications developed with Unity can easily be deployed on a multitude of target platforms including iOS and Android devices as well as UWP (Universal Windows Platform) devices.

- Vuforia³

The Augmented Reality part of the project is based on Vuforia, an AR tracking library which perfectly integrates

with Unity. Vuforia supports several different tracking methods ranging from recognizing plain images to complex objects. With a specific setup, Vuforia can also be used on the HoloLens.

- C# Synth Project and MIDI Support⁴

The C# Synth Project is an open-source library which is used for processing MIDI data and synthesizing it to audio data. MIDI is an industry standard for interconnection between musical instruments and digital devices. Its file format represents musical information like notes values, volume and tempo. Although MIDI is a complex format, it is still the most popular and commonly used format to store musical data. For piano pieces the format is usually sufficient because only one channel is required to store a series of notes and tempo changes.

B. Visualization and Tracking

The application's visuals consist of a Unity 3D scene which renders the virtual keys, combined with Vuforia's tracking abilities to provide the information on where to render the keys.

1) *Vuforia's image target:* For this application tracking via fiducial marker and image target was used. The image target in Unity is a planar object in 3D space which is associated with a set of 2D images. These images represent the markers that are placed somewhere in the real world. Once the camera recognizes a marker the application can trace back the position of the HMD and can therefore project all augmented objects accordingly.

2) *Tracking setup:* Marker images and other tracking settings can be configured in Vuforia's web interface. This configuration with all related assets is then compiled into a Unity package that can be imported into Unity after that. In Unity two components of Vuforia, ARCamera and ImageTarget, are used. Subordinate objects of the ImageTarget become affected by the marker-related projection.

3) *Generating the keyboard:* In order to display the currently played keys, first an entire virtual keyboard is displayed half-transparently superimposing the real one. A script takes care of automatically generating all 88 key objects. One base key object is placed in the scene and aligned at around 90 degrees relative to the ImageTarget. This registration has to match with the real world relation between marker and piano keyboard. All other keys are then generated as duplicates of the base object with respective offset and color (black or white).

C. Audio and MIDI Playback

The two core components of the C# Synth Project library are the MidiSequencer which handles loading and processing MIDI data and the MidiStreamSynthesizer which handles the actual audio playback.

¹<https://www.microsoft.com/en-us/hololens>

²<https://unity3d.com/>

³<https://www.vuforia.com/>

⁴<https://csharpsynthproject.codeplex.com/>

1) *Handling key actions:* During playback the MidiSequencer fires two events that are relevant for this application: MidiNoteOn and MidiNoteOff. These two events are respectively fired when the playback of a note is triggered or terminated and therefore indicate exactly the time when a key is pressed and released. In the implementations of these two event handlers the MIDI code of the affected note is passed as a parameter. The only operation is to map this MIDI code to our according key object and set its material color to either green (in NoteOn) or the default color (in NoteOff).

2) *Combining the audio sources:* The MidiStreamSynthesizer creates actual audio data based on the sequencer's input. To make sure that this audio data is actually redirected to Unity's audio source, the special method OnAudioFilterRead has to be implemented. This method supports direct writing into the audio buffer and therefore redirect the contents of the StreamSynthesizer to Unity's audio source.

V. CONCLUSION

As a prototype the application serves well, but due to the limited FOV, which will most likely increase in the next years with the following generations of AR hardware, its real world usage could be doubted. Furthermore, an evaluation of the different augmentation methods would be useful. Especially when trying out a few more possible approaches, a user test could find out which of the methods are most likely to work in a real-world scenario. A more in-depth study of musical augmentation methods would also be useful for teaching other instruments or even in completely different areas of music.

A. Future Work - The Virtual Piano Teacher

A long-term vision could be the creation of a full-featured virtual piano teacher using AR. Especially early-stage piano learning contains many tasks that could be implemented with AR technologies like the one explained in this paper combined with gamification elements.

1) Use Cases:

- Learning notes and the piano keyboard
Simple exercises or games to recognize the note names and match it with the proper keys could really increase the early-stage learning rate. For beginners the note names could be augmented on top of every key until they become familiar with it.
- Learning easy to intermediate musical pieces
Especially for smaller pieces the AR learning approach could surpass traditional learning by music sheets. Beginners who are not used to reading music yet, would still be able to learn pieces quickly on their own. Additionally a lot more useful information like fingering, expression and dynamics could be displayed during playback.
- Technical exercises
The importance of regular technical exercises for piano students is huge but generally underestimated and disliked. With the introduction of AR and gamification, a whole lot of enjoyable and still pianistically valuable exercises could be realized. By adding some sort of level

system, the student would be even more aware of his progress and more likely to remain motivated.

- Dictionary of chords, scales etc.
A very useful utility not only for beginners but also for advanced pianists would be a piano dictionary. The player could look up all possible chords and scales and would be able to see them highlighted right on top of his keyboard. Especially for jazz piano where complex chords and scales are common, this technology would be of great service.
- 2) *Further Improvements:*
 - Using music sheets as markers
The use of music sheets, perhaps in the form of a special music book, as fiducial markers could eliminate the need for additional markers placed on the piano. It could not only automatically detect the musical piece to be played but also indicate, when to turn the sheets or even highlight musical attributes on the sheets.
 - Checking the learning performance
Real-time feedback of the user's playing could greatly contribute to the learning experience. This could be achieved on the one hand by using MIDI keyboards to directly receive the MIDI input of pressed keys or on the other hand by recording and deconstructing the audio data. The first approach would be technologically straight-forward but would limit the application to electronic keyboard instruments while the second approach would be more flexible but complicated to implement and perhaps inaccurate [14].

The possibilities of the virtual piano teacher are enormous but all are based on the core concept of the technique explained in this paper. As soon as there are improvements in AR hardware, especially concerning FOV, virtual piano teachers can be implemented and actually start to become a helpful tool.

REFERENCES

- [1] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, August 1997.
- [2] L.-T. Cheng and J. Robinson, "Personal contextual awareness through visual focus," *IEEE Intelligent Systems*, vol. 16, no. 3, pp. 16–20, 2001.
- [3] O. Cakmakci, F. Brard, and J. Coutaz, "An augmented reality based learning assistant for electric bass guitar," in *10th International Conference on Human-Computer Interaction*, 2003.
- [4] F. Huang, Y. Zhou, Y. Yu, Z. Wang, and S. Du, "Piano AR: A markerless augmented reality based piano teaching system," in *Third International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2011.
- [5] J. Chow, H. Feng, R. Amor, and B. C. Wunsche, "Music education using augmented reality with a head mounted display," in *Fourteenth Australasian User Interface Conference (AUIC2013)*. Melbourne, Australia: ACM, Jan. 2013, pp. 73–79.
- [6] C. A. T. Fernandez, P. Paliyawan, and C. C. Yin, "Piano learning application with feedback provided by an ar virtual character," in *5th Global Conference on Consumer Electronics*. Kyoto, Japan: IEEE, Oct. 2016.
- [7] I. Barakonyi and D. Schmalstieg, "Augmented reality agents in the development pipeline of computer entertainment," in *4th international conference on Entertainment Computing (ICEC'05)*. Sanda, Japan: Springer, Sep. 2005, pp. 345–356.

- [8] M. Weing, A. Rhlig, K. Rogers, J. Gugenheimer, F. Schaub, B. Knings, E. Rukzio, and M. Weber, “Pi.a.n.o.: Enhancing instrument learning via interactive projected augmentation,” in *Conference on Pervasive and ubiquitous computing adjunct publication (UbiComp13)*. Zurich, Switzerland: ACM, Sep. 2013, pp. 75–78.
- [9] D. Zhang, Y. Shen, S. Ong, and A. Nee, “An affordable augmented reality based rehabilitation system for hand motions,” in *International Conference on Cyberworlds (CW '10)*. Singapore, Singapore: IEEE, Oct. 2010.
- [10] M. Billinghurst and H. Kato, “Collaborative mixed reality,” in *International Symposium on Mixed Reality (ISMR '99)*. Springer, 1999, pp. 261–284.
- [11] S. Steinberg, *Music Games Rock*. P3: Power Play Publishing, 2011. [Online]. Available: <http://www.musicgamesrock.com/>
- [12] R. Shusterman, “Muscle memory and the somaesthetic pathologies of everyday life,” *Human Movement*, vol. 12, no. 1, pp. 4–15, 2011.
- [13] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, “Augmented reality: A class of displays on the reality-virtuality continuum,” *Presence: Telemanipulator and Telepresence Technologies*, vol. 2351, pp. 282–292, 1994.
- [14] S. Dixon, “On the computer recognition of solo piano music,” in *Proceedings of Australasian computer music conference*, 2000, pp. 31–37.