Violin Pedagogy for Finger and Bow Placement using Augmented Reality

François de Sorbier and Hiroyuki Shiino and Hideo Saito
Graduate School of Science and Technology, Keio University, Yokohama, Japan
E-mail: {fdesorbi,shiino,saito}@hvrl.ics.keio.ac.jp

***Abstract*—Beginners need a long time before being able to play correctly violin. Learning the bowing technique appears to be a difficult task and retains most of the attention of beginners. Besides this point, the finger placement is also an important part of the learning but often under estimated. One difficulty is that the fingerboard of the violin does not have frets. In this on-going work, we present a marker-less augmented reality system that advises the novice players about their fingering and bowing. We display in real-time the virtual frets by tracking the violin with a depth camera. We also capture and recognize the note currently played to direct the placement of the bow on the strings.**

I. Introduction

The violin as presented in Fig. 1 is one of the most beautiful, but also one of the most difficult instrument. It is made of two parts: a body with the strings and a bow. Unlike the guitar, the body's fingerboard doesn't show frets to make easier the finger placement. It is still possible to stick guides directly on the violin[1] but changing the appearance of the violin is not always appreciated. Novice violinists also have to learn the difficult bowing techniques to be able to play correctly and smoothly a sound. For instance, some studies state that it can take about 700 hours to master basics of violin bowing [1].

Recent years, various methods have been introduced for assisting the learning of such string instruments. Among them, few have focused on violin pedagogy because tracking the movements of the player and of the violin are still difficult issues. All these approaches can be categorized into wearable computing or image-based processing.

In the first category, researches have investigated the use of vibrotactile feedbacks for the improvement of bowing [2]. The novice player is required to wear a special suit with several vibrators positioned on the sleeves. Those vibrators stimulate the bowing or violin hand for keeping a correct position according to a pre-calibrated trajectory. However, this category of systems does not fit our requirement since we believe it is intrusive and not practical. Moreover, those systems may be difficult to apply for teaching the position of the fingers on the strings.

The second category is mainly focusing on tracking the violin's fingerboard thanks to a colour camera. The goal is to display the virtual frets, i.e. the position where the string should be pressed. A common approach is to attach a marker [3] onto the instrument and compute the relative position of the virtual elements. Motokawa and Saito [4] presented such use of markers for an augmented reality based guitar pedagogy system and guided the fingers placement by displaying a virtual hand. Though, the markers are usually not robust against occlusions and are also difficult to place on thin surfaces like a bow. An alternative is to use feature points detection [5] for which keypoints are computed based on a difference of Gaussian. However, since the violin has a poor texture the number of keypoints may be very small and may also failed under illumination changes. Finally, Löchtefeld *et al.* [6] proposed also an augmented reality system for guitar fingering. They added a projector phone attached on the head of the guitar and directly displayed virtual information on the neck. Repeating such process for a violin might be difficult since the head of the violin is smaller. Also, it might change the weight of the violin which is important because the way of holding a guitar and violin are different.

Fig. 1. Parts of the violin.

In our knowledge, no actual work focusing on both bow and finger placement, and not intrusive from the player's viewpoint has been proposed.

In this paper, we introduce a new marker-free system to advice violin's novice players based on a depth camera. The depth-camera has the advantage to remove the constraint of markers because we can directly analyse the depth data

[1] http://www.fretlessfingerguides.com/

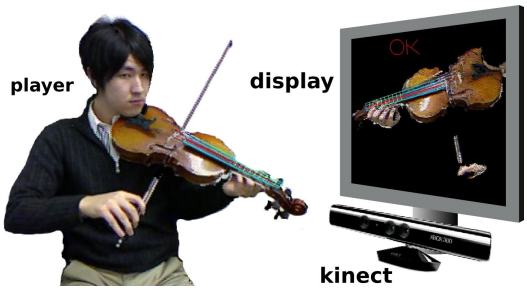

Fig. 2. Overview of our violin pedagogy system.

for estimating the pose of the violin. With this information, our system advises the correct location where to press the strings and to pull the bow. The remainder of this paper is structured as follows: The first section describes the structure of our capture and tracking system. The second section details our violin pedagogy interface. Finally, we present several quantitative results since no user study was conducted for the moment.

II. Architecture of our system

Our system needs to track in real time the violin without the help of any markers, while the virtual advices are displayed on a screen. An overview of the components of the system is depicted in Fig. 2. In this configuration, the player does not have to carry any device and the musical instrument is not altered by markers. We based our solution on the Kinect [2], a camera that captures in real-time a colour image from a given scene with its corresponding depth information. Knowing the intrinsic parameters of the depth camera (focal length f and principal point $\mathbf{p}$) and assuming square pixels, it is possible to convert the 2-D *plus* depth raw data into a 3-D point cloud using this transformation:

$$\mathbf{X} = \begin{bmatrix} \frac{1.0}{f} & 0 & -\mathbf{p}_x \\ 0 & \frac{1.0}{f} & -\mathbf{p}_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}. \quad (1)$$

where $\mathbf{X}$ is a 3-D point and $\mathbf{x}$ the 2-D *plus* depth input.

However, the raw depth data from Kinect are often noisy and implies that some 3-D points might be inaccurate. A bilateral filter [7] can then be applied to reduce the noise while preserving the edges,

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|I_p - I_q|) I_q \quad (2)$$

where $G_{\sigma_x} = e^{-x^2\sigma^{-2}}$ and W_p is the normalizing constant.

A. Overview

Advising the player about the finger placement requires adding the virtual frets on the capture of the violin currently displayed on the screen. Our approach is based on a 3-D reconstruction of the violin. First, this model is used to manually set the position of the virtual frets in a known referential. Second, we are able to compute the rigid transformation between it and the current captured point cloud. We can then know where to display the virtual frets by using this method.

For each frame, we evaluate the rigid transformation (rotation and translation) between the point cloud from the depth camera and the model, using the classic Iterative Closest Point algorithm [8]. We had the choice between a frame-to-frame or frame-to-model comparison methods (registration). In the first case, we cumulate all the transformations estimated between two consecutive frames. But this approach is prone to error propagation along the tracking while an error with the frame-to-model registration is taken into account only for one frame. For this reason, we preferred the frame-to-model tracking approach.

ICP is also known to be a time consuming algorithm especially if the number of data to compare is high. Conversely, if we decrease too much the number of points for better performance then the estimation of the rigid transformation might be inaccurate. We decided to reduce the complexity of the computation by dividing the pre-computed model into multiple sub-models. Each sub-model is then defined with enough point for the registration while the computation time is reduced.

B. Creation of the violin model

We based the construction of the violin's model on several sub-model captured from slightly different viewpoints. The benefit of this approach is that each sub-model is described with enough points for robust pose estimation during the tracking. During this off-line stage, we capture and segment the violin based on its main colour and the depth information. Details about this segmentation will be given in the following section. All the sub-models are mainly captured with parts of the front face of the violin because it is the most often observed and most important area during the tracking phase. We store only a small number of sub-models that we identify according to its plane equation obtained during the segmentation.

Though, we need to be sure that the sub-models stored in the database are enough different. We then compare the normal

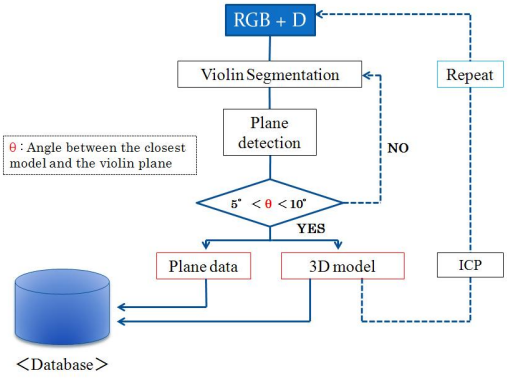

Fig. 3. Pipeline for the capture and the storage of the sub-models.

[2]http://www.microsoft.com/en-us/kinectforwindows/

Fig. 4. Several results of our violin segmentation approach. The body and the fingerboard are correctly extracted.

(deduced from the plane equation) of the current candidate with the normal of the sub-models already stored. If the normal is enough different then we include this violin's point cloud as a new sub-model. In our current implementation, we store 25 sub-models with the related plane equations for an efficient retrieval during the tracking. An overview of this stage is presented in Fig. 3.

C. Violin segmentation

Once the sub-models have been generated, we can start the tracking of the violin for each input frames. For each depth map, we start by applying the bilateral filter from equation (2) for reducing the noise from the raw data. The filtered depth map with the corresponding colour information is then segmented searching for the brown colour of the violin. However, the result of this segmentation will remain incomplete because of the specular reflections, occlusions and difference of colour on the surface. For instance, the fingerboard has yet to be included in the segmentation. In our approach, we take advantage of the depth information to complete this colour-based segmentation.

Our solution is to add one more stage after this colour segmentation but based on the depth information. We start by computing a plane equation describing the body of the violin. From the first segmentation, we can get the 3-D values corresponding to the extracted colour pixels. Those 3-D points are then used to estimate a plane equation that we minimized with RANSAC. Knowing the classic dimensions of the violin, we define a bounding box positioned at gravity centre of all the points belonging to this plane. The plane equation is also used to align this bounding box. All the points included to this volume are then conserved to represent the violin of the current frame. Several visual results of the segmentation of the violin are presented in Fig. 4.

D. tracking

The goal of tracking stage, as depicted in Fig. 5, is to estimate the pose of the violin. We compare the 3-D point cloud segmented from the current frame with one of the sub-models previously stored in the database. The choice of the sub-model is done by comparing the normals defined with the body of the violin. The transformation associated with these normals also helps to find out a good initial solution for the ICP algorithm.

During the tracking, even if some 3-D points that do not belong to the violin are included in the segmentation, like from the player's hands or the bow, then the pose estimation will still be correct. With the ICP algorithm those points that do not fit one of the sub-models will be considered as outliers and removed for the pose estimation.

III. The violin pedagogy

In this section, we explain our approach about our choices for the violin pedagogy using augmented reality. Besides the explanations about the virtual advices, we will also introduce how we capture and use the analysis of the sound played by the violinist.

A. Virtual information

In the previous section, we described our approach to compute the rigid transformation between the current capture of the violin and one sub-model from the database. We also explained that for each sub-model we stored the rigid transformation toward the first one stored in the database. This information is important since the virtual information is manually set based on the first model. So, using these transformations, we can convert the virtual guides to the current view of the violin or vice versa. We finally retained to transform the captured violin into the referential of the virtual guides. This choice has the advantage to display the virtual information always at the same position on the screen while we adapt the current captured violin to it. By doing so, the player don't have to search the advices on the screen since it will always be at the same position.

The goal of our pedagogic support system is to advice the violinists about their fingering, i.e. where they should press the strings on the finger board for playing the correct note. Since the violin does not have fret, we display virtual frets for advising the correct placement on the fingerboard as depicted in Fig. 6. However, knowing the position of the frets does not help the novice player to know which of the four strings has to be pressed. For this reason, we emphasised the strings to make them easily visible and added a red point at the intersection of the string and the fret that have to be pressed.

B. Sound analysis

By visualizing the virtual frets and strings, the player can know where to press the fingerboard to play a note. However, it

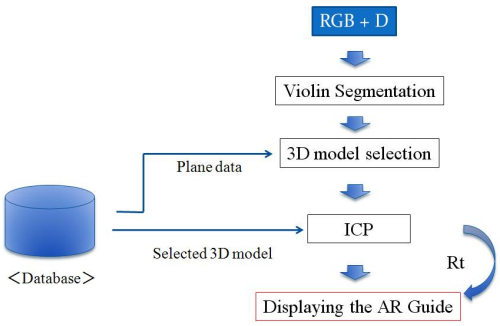

Fig. 5. Pipeline during the violin tracking stage.

Fig. 6. Several views of the violin augmented with the virtual frets and the emphasized strings. A red line and a red dot respectively denote the string and the fret that have to be pressed.

is still difficult to understand if the note played was correct or not, or if the position of the bow on the strings is also correct. We then added a microphone to capture the sound played. Our system analyse this sound using a real-time spectrum analyser[3] based on a wavelet transformation to analyse the violins sound and to evaluate the accuracy of the pitch[4] in cent unit when it is compared to a given note.

We propose three approaches to define the reference note used for the comparison. The first one asks the violinist to play a specific note. The system knowing the goal pitch can then evaluate the correctness of the note. The second one asks the player to select a scale. The system will then ask to play the notes from this scale. In the last approach, the player plays the note of his choice that the system try to recognize. When the result is displayed, the player can then check if the note played was the expected one or not.

Figure 7 shows the virtual information displayed when evaluating the accuracy of the pitch from the note currently played. If the user plays at the correct pitch, an "OK" mark is displayed. Otherwise, If the pitch is too low or too high, then the "Low" or "High" marks appear. In this latest case, a green arrow is also displayed to show to the player the direction where the bow has to be moved to get the correct pitch.

[3]http://www.fmod.org/

[4]frequency of a note

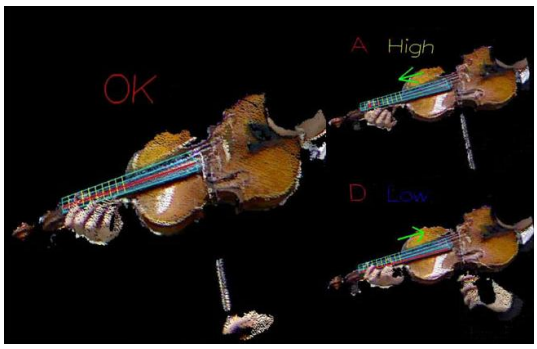

Fig. 7. The pitch of the note currently played is analysed for advising the position of the bow on the strings. If the position is not correct a message and an arrow are displayed to indicate the correct direction where to move the bow.

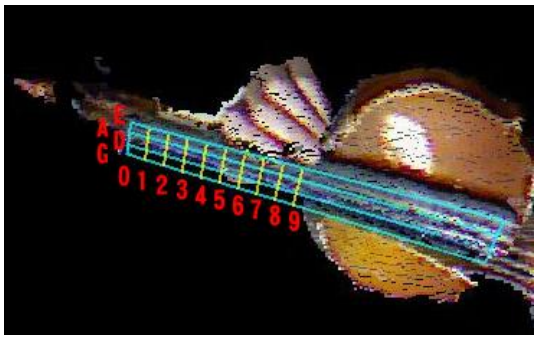

Fig. 9. Numbering of the frets used for the evaluation of the pitch.

IV. RESULTS

Our experiments were performed on an Intel Core2 DUO 2.80GHz PC. We used Kinect with the SDK OpenNI[5] at a resolution of 640×480. We measured an average computational time of 21ms that is suitable for a real-time application. The violinist performing the experiments was a confirmed player.

For this experiment, we first evaluated the accuracy of our tracking approach based on the ICP algorithm. We compared it with the AR-Toolkit marker tracking. We added four markers on the body of the violin and pre-computed the sub-models included it. During the online phase, we computed the rigid transformation between the first sub-model and the segmented violin using our approach and using the marker-based approach. Considering the marker-based transformation as the ground truth, we got the results presented in table I. Except for few frames where the tracking appears to be wrong (line of maximum error), the average errors are low which confirm that our marker-less tracking gives good results. The origin of the remaining errors can come from the noisy depth values from Kinect. Another possibility might be that the sub-models have also been captured with errors. For this last point a possible improvement might be to include KinectFusion [9] in our future implementation.

TABLE I
EVALUATION OF OUR TRACKING COMPARED TO THE GROUND TRUTH. IT SHOWS THE RIGID TRANSFORMATION MATRIX DECOMPOSED IN THREE ROTATIONS AN ONE TRANSLATION.

	SR_x(deg)	R_y(deg)	R_z(deg)	T(mm)
Minimum error	0.12	0.25	0.20	0.22
Maximum error	13.29	8.27	7.89	32.1
Average error	3.07	2.69	2.78	7.20

We also evaluated the accuracy of the virtual frets position and numbered like as depicted in Fig. 9. Each fret has a corresponding pitch, so by pressing the strings at a fret level, we expect to obtain a similar pitch. For this experiment, we measured the correctness of the pitch when a skilled player (to ensure a correct manipulation of the bow) was pressing the string on the virtual frets. Table II presents the results of

[5]www.openni.org

Fig. 8. Different visual results of the overlaying of a precomputed sub-model from the database onto the current captured image of the violin. Note that the occlusions with the bow and the hands do not interfere with the correct estimation of the rigid transformation.

this experiment for each fret, where a difference of pitch closes from zero means that the accuracy is good. These results show that the position of the frets is almost correct. Once again, the cause of the difference might be the noise from the depth data.

TABLE II
MEASURE OF THE DIFFERENCE ON EACH FRET BETWEEN THE CURRENT PITCH AND THE EXPECTED PITCH IN CENT UNIT

Fret number	1	2	3	4	5
Difference of pitch	11.1	14.1	12.0	12.4	13.4
Fret number	6	7	8	9	Average
Difference of pitch	15.8	12.8	13.9	19.2	13.8

V. CONCLUSIONS

We have presented our on-going work about a marker-free augmented reality system that helps novice violinists during their learning. Thanks to a depth camera, we are able to help virtually the player on his fingering by overlaying the virtual frets and emphasising the strings of the violin. We also proposed to capture and analyse the sound for advising the player about the correct position of the bow on the strings. The advantage of our system is that all the stages are performed in real-time, and also, the tracking is robust against occlusions that often occur while playing a musical instrument. Currently, our system is applied to a violin but can be easily extend to other string instruments without fret like the cello.

Currently, we are modifying the system to integrate a wireless see-through HMD which might improve the visual quality and immersion of the novice player. At that time, we will try to confirm our choice by conducting a user evaluation about the visualization and the pedagogy parts. We are also trying to apply our method on GPU for faster performances, especially for the ICP algorithm. We will be able to analyse more information like the position of the bowing of the player and then give more advices.

REFERENCES

[1] J. Konczak, H. vander Velden, L. Jaeger, *Learning to play the violin: motor control by freezing, not freeing degrees of freedom by freezing,* Journal of motor behavior, vol.41, no.3, pp. 243-252, 2009.

[2] J. van der Linden, E. Schoonderwaldt, J. Bird, R. Johnson, *MusicJacket: Combining Motion Capture and Vibrotactile Feedback to Teach Violin Bowing,* IEEE Transactions on Instrumentation and Measurement, vol.60, no.1, pp. 104-113, January 2011.

[3] Y. Motokawa, H. Saito, *Support system for guitar playing using augmented reality display,* In Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 243-244, 2006.

[4] H. Kato, M. Billinghurst, *Marker tracking and HMD calibration for a video-based augmented reality conferencing system,* In Proceedings of the 2nd International Workshop on Augmented Reality, pp. 85-94, 1999.

[5] D. Lowe, *Object recognition from local scale-invariant features,* Proceedings of the International Conference on Computer Vision, pp. 1150-1157, 1999.

[6] M. Löchtefeld, S. Gehring, R. Jung, A. Krger, *Using mobile projection to support guitar learning,* In Proceedings of the 11th international conference on Smart graphics (SG'11), pp. 103-114, 2011.

[7] C. Tomasi, R. Manduchi, *Bilateral filtering for gray and color images,* Sixth International Conference on Computer Vision, pp.839-846, 1998.

[8] S. Rusinkiewicz, M. Levoy, *Efficient variants of the ICP algorithm,* Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, pp.145-152, 2001.

[9] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, A. Fitzgibbon, *KinectFusion: Real-Time Dense Surface Mapping and Tracking,* Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pp.127-136, 2011.