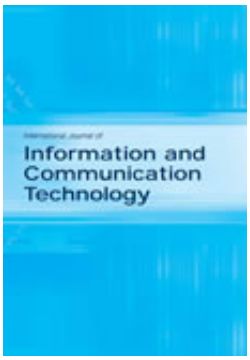


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Violin fingering teaching algorithm based on augmented reality and motion capture technology

Mengyingyi Lei

DOI: 10.1504/IJICT.2025.10071987

Article History:

Received:	06 May 2025
Last revised:	23 May 2025
Accepted:	23 May 2025
Published online:	16 July 2025

Violin fingering teaching algorithm based on augmented reality and motion capture technology

Mengyingyi Lei

Xi'an Peihua University,
Xi'an 710125, China
Email: leimengyingyi@163.com

Abstract: Violin fingering techniques change rapidly, making it difficult to correct incorrect fingering in real time during instruction. To address this issue, this paper first employs a multi-head attention mechanism (MAM) and multi-scale dilated convolutional neural networks (DCNN) for hand fingering motion capture. Since hand movement occurs during performance, an augmented reality (AR)-based hand pose estimation module is designed. The pose from orthography and scaling with iterations (POSIT) algorithm, optimised using the Gauss-Newton method, is used to estimate relatively precise camera-based fingering poses. Finally, cosine similarity is used to compare virtual and real finger techniques, and corrections are made based on the features of the target finger technique to improve teaching effectiveness. Experimental results show that the proposed algorithm achieves a correction accuracy rate 3.66%–13.13% higher than the baseline algorithm, laying a foundation for improving violin finger technique instruction.

Keywords: violin fingering instruction; motion capture; augmented reality; multi-scale dilated convolution; POSIT algorithm.

Reference to this paper should be made as follows: Lei, M. (2025) 'Violin fingering teaching algorithm based on augmented reality and motion capture technology', *Int. J. Information and Communication Technology*, Vol. 26, No. 26, pp.17–32.

Biographical notes: Mengyingyi Lei received her Master's degree from the Guangxi Arts University in 2015. She is currently a Lecturer in the Xi'an Peihua University. Her research interests include machine learning, augmented reality and artistic education.

1 Introduction

The violin, a treasure among string instruments, occupies a pivotal position in classical and modern music composition thanks to its rich expressiveness and profound artistic depth. However, learning to play the violin is a complex and challenging process, especially when it comes to mastering fingering techniques, which often requires learners to invest a great deal of time and effort in repeated practice and correction (Goldie, 2015). Traditional violin fingering instruction primarily relies on verbal guidance from teachers, demonstration performances, and students' visual observation and imitation. While this teaching method has a long history, it has obvious limitations in terms of teaching efficiency, personalised guidance, and learning experience (Peinan and

Pattananon, 2022). As the information technique rapidly growing, the emergence of augmented reality (AR) and motion capture technology has brought revolutionary opportunities for violin fingering instruction (Aykut and Taş, 2023). How to apply these two technologies to violin fingering instruction to help learners quickly identify and correct incorrect movements and improve learning efficiency is a research topic with practical value (D'Amato et al., 2020).

In violin fingering instruction, relying on traditional verbal guidance, teachers often find it difficult to clearly explain abstract concepts, key points, and complex 3D structures that are difficult to understand (Akdeniz, 2015). AR technology allows users to experience a new environment where reality and virtual scenes are seamlessly integrated, opening up new possibilities for violin fingering instruction. Campo et al. (2023a) designed a virtual instrument that students can operate collaboratively, using an optical multi-touch screen as a multi-user input device to simulate percussion performance. Since camera-based motion tracking has become a popular supporting technology for gesture-based human-computer interaction, Wang (2024) proposed a gesture-controlled virtual violin, whose internal sound engine can be continuously controlled through various complex gestures. Similarly, the virtual instrument system developed by Fonteles and Rodrigues (2021) also uses the real-time posture detected by the Kinect device to interact with musical instruments. Rosa-Pujazón et al. (2015) used the Kinect device to implement virtual percussion instruments such as drums. Feng (2023) conducted research on gesture recognition in violin performance. They estimated the position of each joint and predicted the tapping of the fingertips, creating a virtual violin system that could simulate violin performance by tapping fingers on any surface, but required a physical instrument to operate.

In AR-based violin fingering instruction, accurately capturing real-time finger position data, comparing it with standard fingering models, and correcting learners' fingering is a key issue. Dalmazzo and Ramírez (2019) used the number and direction of fingertips as gesture features and combined them with a decision tree classifier to perform fingering recognition, but the recognition accuracy was not high. Nakamura et al. (2020) established a two-dimensional gesture coordinate system based on the main direction of gestures and used the spatial distribution characteristics of gesture coordinate points to perform preliminary recognition of piano fingering. They then used the dynamic time warping (DTW) method to identify the final fingering. Gao and Li (2023) extracted two types of features from gesture images, namely the histogram of orientation gradients (HOG) and local binary patterns (LBP), and then fused these features. They combined this to complete gesture recognition.

Deep learning-based finger recognition methods use automatic feature extraction, which has gradually improved the accuracy of gesture recognition. Su and Liang (2002) used data gloves to capture hand acceleration and bending angle data, then preprocessed the signals and fed them into a recurrent neural network (RNN) for classification. Pigou et al. (2018) proposed a spatio-temporal dual-stream convolution architecture that can extract both the spatial dimensional features of fingering images and the temporal dimensional features contained in video sequences, thereby improving recognition performance. Sun et al. (2020) jointly used improved convolutional neural network (CNN) and support vector machine (SVM) to extract features and classify segmented hand images, but the recognition error was relatively high. Liu and Fu (2023) used CNN and bidirectional long short-term memory networks (BiLSTM) to learn sequential data,

and after completing the classification, converted the prediction results into specified commands for application in AR teaching.

In summary, existing violin fingering teaching algorithms do not fully consider the dynamic changes in fingering, resulting in low accuracy in correcting students' incorrect fingering during teaching. To address the above issues, this paper proposes a violin fingering teaching algorithm based on AR and motion capture technology. The main innovative modules of this algorithm are summarised as follows.

1 Designed a violin dynamic fingering motion capture module based on MAM and multi-scale DCNN. By constructing a spatial connection map of hand movements through a spatial attention mechanism and utilising temporal attention to learn the temporal relationships between joints, the depth and effectiveness of feature extraction were significantly improved. To avoid noise interference, a multi-scale DCNN was used to flexibly capture the spatial relationships between finger movements, significantly improving recognition performance.

2 To locate the position of the hands during performance, an AR-based hand pose estimation module was designed. The pose from orthography and scaling with iteration (POSIT) algorithm and point-to-point data were used to estimate the relative finger positions with high accuracy, and the Gauss-Newton method was used to optimise the distance between control points to obtain more accurate virtual finger positions.

3 Based on violin dynamic fingering motion capture and hand pose estimation, we designed a fingering motion comparison method that combines feature description based on relative two-dimensional vectors between hand joints and cosine similarity. For incorrect joints, this paper calculated the corrected positions based on the features of the target fingering.

4 Simulation experiments and visualisation analysis were conducted on real datasets. The results showed that the top-1 and top-5 correction accuracy rates of the proposed algorithm were 94.18% and 97.36%, respectively, which were better than the baseline algorithm and could significantly improve the teaching effect of violin fingering.

2 Relevant technologies

2.1 Definition and classification of violin fingering

The arrangement of fingers and the order in which they alternate when playing an instrument, as well as the notation of this arrangement and order in the score, is called fingering (Kruger and Jacobs, 2020). Proper fingering is one of the playing techniques that can make violin playing as close to perfect as possible. It helps us express musical emotions accurately and even more creatively.

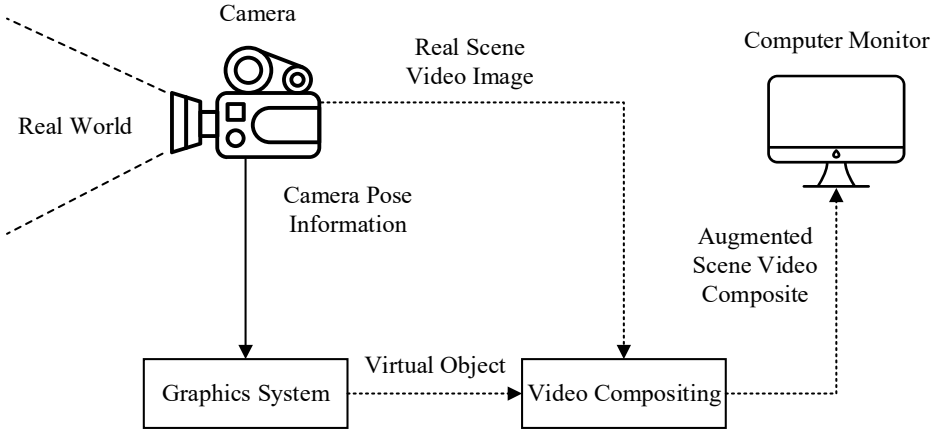
Finger techniques, like bowing techniques, can also be classified into different categories. Based on the structure of the finger technique itself, finger techniques can be divided into 'fixed fingering' and 'non-fixed fingering' (Kinoshita and Obata, 2009). Fixed fingering techniques play an irreplaceable role in consolidating hand positions and pitch concepts, as well as improving memory, but they are somewhat mechanical in

nature. 'Non-fixed fingering techniques,' on the other hand, can be further divided into 'extended fingering techniques' and 'dense fingering techniques.' From a stylistic perspective, violin fingering techniques can be categorised into classical fingering, jazz fingering, and other techniques. In summary, there are numerous classifications of violin fingering techniques, and the appropriate classification should be selected based on the specific context.

2.2 AR technology

The principle of AR is to apply virtual information generated by a computer through simulation to the real world. In this way, the two types of information complement each other, thereby achieving enhancement of the real world (Yılmaz and Göktaş, 2018). A typical AR system structure is shown in Figure 1, which mainly consists of interactive devices such as virtual scene generation units, displays, and tracking and positioning devices. Among these, the virtual scene generation unit is used for peripheral management of scene models; the display is primarily used for real-time transmission of signals that fuse the real world with virtual objects; the head-mounted tracking and positioning device is used to track the coordinates and field of view of objects in the real world; and the interaction device is primarily used for input and output of environmental control signals and sensory signals.

Figure 1 The typical AR system structure



AR and virtual reality both belong to the category of immersive technology, both aim to provide users with a richer, more vivid and immersive experience, breaking the limitations of traditional two-dimensional interaction, allowing users to interact with digital content in a new way. Through AR technology, violin fingering teaching will realise the change from abstract symbols to figurative interaction, significantly reducing the threshold for beginners.

2.3 Dilated convolution

CNN is a deep learning model specifically designed to process data with a grid structure (such as images, videos, time series, etc.). In the CNN, DCNN is a technique that expands the receptive field by inserting gaps in the convolution kernels (Wang et al., 2019). Their core advantage lies in their ability to capture broader contextual information without increasing computational complexity or the number of parameters, thereby improving network performance.

Unlike standard convolution, DCNN can achieve zero-padding without affecting the original data. Adjusting the hole ratio (*ar*) can change the size of the hole convolution, where the hole ratio is the number of zeros filled in the adjacent parameters of the convolution kernel. Standard convolution is actually the convolution of DCNN when *ar* is 1. The definition of DCNN is shown below.

$$y(i,\ j) = \sum_{h=1}^{H}\sum_{w=1}^{W} x(i + ar \times h,\ j + ar \times w) \times w(h,\ w) \tag{1}$$

where H is the length of the input feature map, W is the width of the input feature map, $x(i, j)$ is the pixel value at position (i, j) in the input image, and $y(i, j)$ is the output after the dilated convolution.

3 Violin fingering motion capture based on multi-head attention and multi-scale dilated convolutions

3.1 Temporal and spatial characteristics of violin fingering

Due to the rapid dynamic changes in violin fingering, traditional methods find it difficult to extract the dynamic characteristics of fingering. To address this issue, this paper uses MAM to encode the spatial relationships between key joints, constructing a fully connected graph to accurately capture the complex dependencies between joints. At the same time, parallel multi-scale DCNNs with different expansion rates are introduced to effectively capture multiple time information, thereby improving the model's perception of dynamic changes. The overall module for violin fingering motion capture is shown in Figure 2.

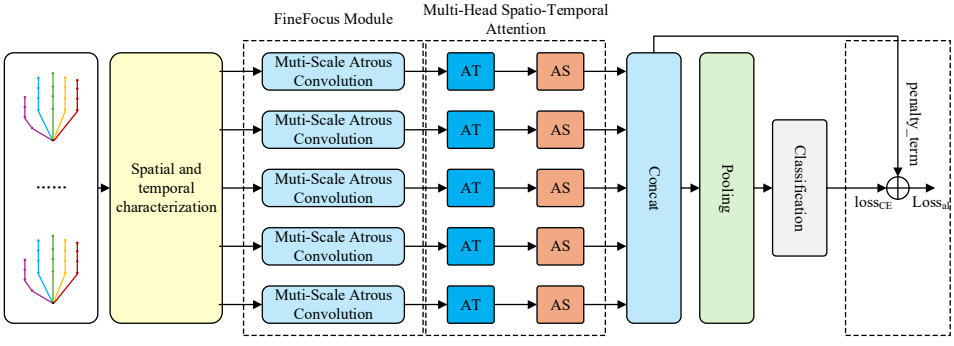
To accurately capture the fingering movements of the violin, this paper applies position encoding to the joints of the hand so that the model can better utilise the relationships between joints, as shown in equation (2) and equation (3). The purpose of positional encoding is to inject signals of positional information into the model input, which is particularly important for processing sequential data.

$$E_{pos,2i} = \sin\left(\frac{pos}{10,000^{\frac{2i}{d}}}\right) \tag{2}$$

$$E_{pos,2i+1} = \cos\left(\frac{pos}{10,000^{\frac{2i}{d}}}\right) \tag{3}$$

where d is the encoding feature dimension; pos is the position index of the hand in the sequence; i is the feature dimension index. Each frame t in the gesture data contains N joint position information, represented as $P_t = [p_t^1, p_t^2, ..., p_t^N]$, where p_t^j is the three-dimensional coordinate vector of the j^{th} joint in frame t. Assume that at each time step t, each joint p_t^j generates a position encoding vector E, which is then combined with the three-dimensional coordinates of the joint to form joint feature F_t^j as $F_t^j = [p_t^j; E]$.

Figure 2 The overall module for violin fingering motion capture (see online version for colours)



3.2 *Enhancement of violin fingering features based on spatio-temporal attention*

After obtaining the characteristics of finger techniques in violin playing, this paper designed spatial attention (AS) and temporal attention (AT) to extract spatial and temporal information of finger techniques, respectively. AS first takes F_t^j as input, updates it, and encodes spatial information. Then, the updated node features are input into AT to further learn time information; finally, the results are averaged and aggregated into a vector, which is used as the feature representation for classification. AS first calculates the scaling dot product between the query vectors and key vectors of nodes within the same time step; then, the results are normalised using the softmax function, as shown in equation (4) and equation (5).

$$s_{jk}^h = \frac{Q_{t,j}^h \cdot K_{t,k}^h}{\sqrt{d_{\text{model}}/h}} \tag{4}$$

$$\alpha_{jk}^h = \text{softmax}\left(s_{jk}^h\right) \tag{5}$$

where d_{model} is the size of the model's hidden level; s_{jk}^h represents the scaling point product between nodes p_t^j and p_t^k; $Q_{t,j}^h$ and $K_{t,k}^h$ are represented by the query vector and key vector between the hand joint nodes, respectively, α_{jk}^h is the attention weight between p_t^j and p_t^k.

Each AS head generates a weighted skeleton diagram representing the specific spatial structure of the hand. Then, the AS features of node p_t^j are weighted and averaged with the value vector, and the results are merged as shown below.

$$\overline{F}_{t,j}^{h} = \sum_{j=1}^{N} \left(\alpha_{jk} \cdot V_{t,k}^{h} \right) \tag{6}$$

$$\tilde{F}_t^j = Concate\left[\overline{F}_{t,j}^{1}, \overline{F}_{t,j}^{2}, ..., \overline{F}_{t,j}^{H} \right] \tag{7}$$

where H is the number of attention heads; $\overline{F}_{t,j}^{h}$ is the weighted average of the AS weights and value vector $V_{t,k}^{h}$ of p_t^j; $\tilde{F}_t^j$ is the weighted average of multiple heads combined to form the final AS feature representation p_t^j.

For the AT part, the output node features of AS are used as input, and the attention mechanism described above is used for encoding. Given a time step T and the number of hand joints J, perform time masking t_{mask} and spatial masking s_{mask} as shown below.

$$t_{mask}[i, j] = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases} \tag{8}$$

$$s_{mask}[i, j] = 1 - t_{mask}[i, j] \tag{9}$$

To ensure the transmission of information from the same hand joint at different times, select an appropriate mask for constraint based on the time or space domain, as shown below.

$$\alpha = \begin{cases} \alpha \times t_{mask} + (1 - t_{mask}) \times \eta \\ \alpha \times s_{mask} + (1 - s_{mask}) \times \eta \end{cases} \tag{10}$$

3.3 Feature focusing and recognition results output based on multi-scale hole convolution

Due to noise or errors in weight distribution, attention distribution may not be accurate. To address this issue, a fine-focusing module based on multi-scale DCNN was designed before AS. By expanding the receptive field, the network can capture multi-scale spatial dependencies without significantly increasing the number of parameters. DCNN uses different convolution kernel sizes and dilation rates to adapt to different scale features in dynamic fingerings.

DCNN consists of five branches, each of which extracts features of specific scales through convolution. The fifth branch extracts global features through global average pooling. Finally, all features are concatenated and unified through convolution for dimension reduction and fusion, generating the final output as shown below, where BN is the normalisation operation, the dilation rate d is generated from the input features as shown in equation (12), $P(x)$ is the pooling operation, and W is the dilation rate value.

$$\begin{cases} Y_{1\times1} = ReLU\left(BN\left(W_{1\times1} * X\right)\right) \\ Y_d = ReLU\left(BN\left(W_{3\times1,d} * X_{dil=d}\right)\right) \\ Y_G = ReLU\left(BN\left(W_{1\times1} * GlobalAvgPool(X)\right)\right) \\ Y = ReLU\left(BN\left(W_{1\times1} * Concat\left(Y_{1\times1}, Y_6, Y_{12}, Y_{18}, Y_G\right)\right)\right) \end{cases} \tag{11}$$

$$d = \sigma\left(W \cdot P(x)\right) \tag{12}$$

After processing the output feature map through the network, pooling is performed to form a vector for result prediction. Meanwhile, a SoftMax layer is added to the end of the model to calculate the probability value of each fingering category. The category with the highest probability value is the final prediction result.

4 Violin fingering teaching algorithm based on AR and motion capture technology

4.1 Finger position estimation for violin playing based on improved AR algorithm

To improve the teaching effectiveness of violin fingering, this paper designs a violin fingering teaching algorithm based on AR and motion capture technology, as shown in Figure 3. First, RGB video clips of violin performances were captured using the Azure Kinect DK (Servi et al., 2024) device, and the performer's hand skeleton sequence was extracted using the device's SDK. Finally, the extracted hand skeleton sequence is input into the aforementioned finger movement capture module to obtain finger recognition results. When specific finger movements are recognised, positional changes occur during the performance. To locate the position of the character's hands, a hand pose estimation module based on an improved AR algorithm is designed to obtain estimation results for comparison and analysis. This enables comparison and correction of students' finger movements, thereby improving teaching effectiveness.

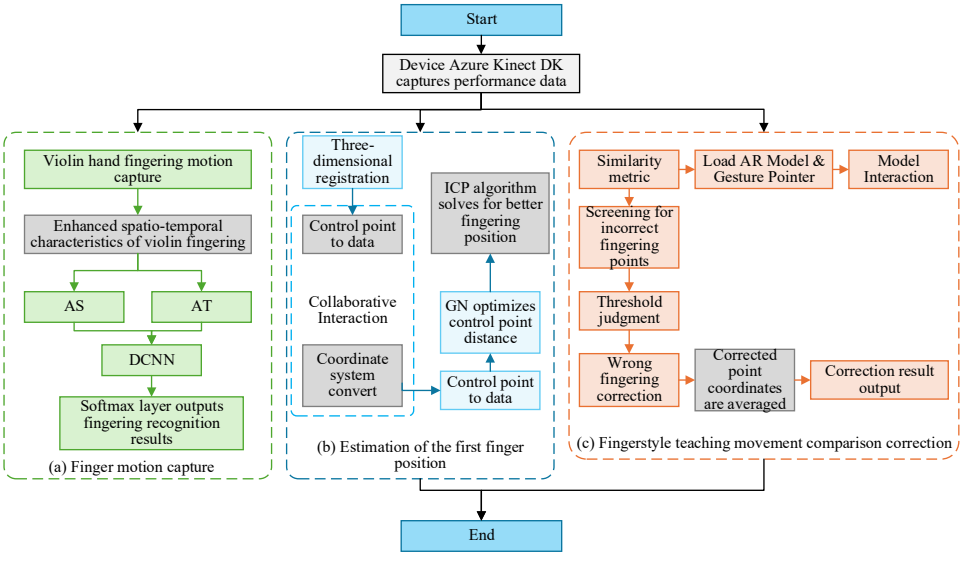
The POSIT algorithm is a commonly used method for solving camera pose (Bhakar and Bhatt, 2020), but in violin playing, when the left hand presses the strings, some fingers are often obscured by the neck, causing POSIT to misjudge the fingering. For this purpose, the POSIT algorithm and point-to-data estimation are used to estimate the relative camera pose with high accuracy. Finally, the Gaussian Newton method (GN) is used to optimise the distances between control points to obtain a more accurate finger pose.

First, the initial pose of the camera is solved using the control point information in the original POSIT algorithm. The correspondence between the control points and the coordinate system is shown in Figure 4. First, assume that the three-dimensional point of the hand is P_i and its normalised coordinates are represented by $m_i = (x_i^m, y_i^m, 1)$. Let u_i represent the pixel coordinates of P_i, which are known information, and let K represent the internal parameter matrix of the camera. Based on the camera model, the normalised coordinates $m_i = K^{-1}u_i$ can be calculated. From the pinhole camera model, we can see that P_i and its normalised coordinates, as well as the optical centre O_c, are on the same

straight line. Therefore, the cross product of the vectors represented by m_i and P_i^c is $\vec{0}$, as shown below.

$$\overrightarrow{m_i} \times \overrightarrow{P_i^c} = \vec{0} \tag{13}$$

Figure 3 The flow of violin fingering teaching algorithm based on AR and motion capture (see online version for colours)



When n control points are involved in the calculation, a linear equation system $Mx = 0$ containing more equations can be formed. The size of the M matrix is $2n \times 12$, and the solution of the equation system is contained in the null space of the M matrix, as shown in equation (14), where N is the size of the null space of M; β_i is the eigenvalue of the characteristic vector v_i. Let $u_j = [u_j\ v_j]^T$ represent the pixel coordinates of the j^{th} control point. According to the coordinate system transformation principle, we have equation (15), where f_x and f_y are the pixel points on the x-axis and y-axis, respectively.

$$x = \sum_{i=1}^{N} \beta_i v_i \tag{14}$$

$$s\begin{bmatrix} u_j \\ v_j \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta v_x^{[j]} \\ \beta v_y^{[j]} \\ \beta v_z^{[j]} \end{bmatrix} \tag{15}$$

In the above equation, s represents the scale factor. By rearranging the equation, we can obtain the pixel coordinates of the control points as follows.

$$\begin{cases} u_j = \dfrac{v_x^j}{v_z^j} f_x + u_0 \\ v_j = \dfrac{v_y^j}{v_z^j} f_y + v_0 \end{cases} \tag{16}$$

The initial virtual hand pose for violin playing was obtained through the above calculations, but the accuracy of the initial pose was not optimal, so further optimisation was carried out. During the transformation of the three-dimensional coordinate system, the distances between the four control points remain unchanged. Therefore, an optimised objective function is constructed based on the distances between the control points, as shown in equation (17), where c_j^c is calculated from the three-dimensional coordinates c_j^w in the world coordinate system, the initial pose R, and the current time t.

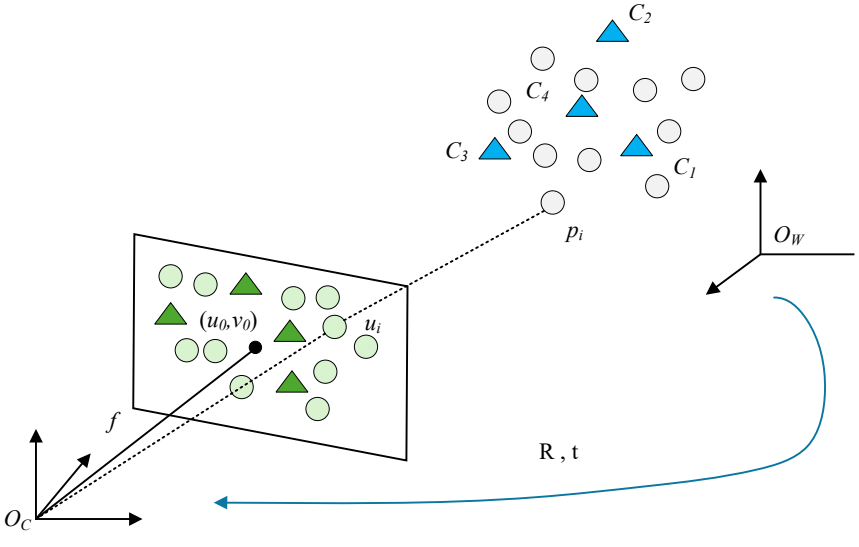
$$c_j^c = Rc_j^w + t \tag{17}$$

The value of β can be solved using equation (14). In the world coordinate system, the relative distance between points c_i and c_j is known. Apply GN optimisation to find the least squares solution, and define the objective function to be optimised as follows.

$$Error(\beta) = \sum_{i<j} \left(\left\| c_i^c - c_j^c \right\|^2 - \left\| c_i^w - c_j^w \right\|^2 \right) \tag{18}$$

where $c_j^c = (\beta v_x^j, \beta v_y^j, \beta v_z^j)$ is the optimisation variable and β is the constrained variable.

Figure 4 Control points versus coordinate system (see online version for colours)



By using β after the convergence of the objective function to calculate the new c_j^c, coordinates of the reference point in the camera system can be calculated, and at this

time, the 3D-3D point pair information is obtained. Finally, the point pair information is substituted into the iterative closest point (ICP) algorithm to re-solve for better violin finger postures R and t.

4.2 Violin fingering technique comparison and correction

After obtaining relatively accurate violinist hand position and posture through AR methods, cosine similarity is used to measure the fingering within the same hand gesture, and a similarity matrix is formed to filter out incorrect fingering points, as shown below.

$$\cos(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1| \times |v_2|} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}} \tag{19}$$

This yields a similarity matrix, as shown in equation (20), where S_{ij} refers to the cosine similarity between vector a_{ij} in the target fingering and vector a'_{ij} in the AR fingering.

$$Sim_t = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{bmatrix} \tag{20}$$

Since the average similarity data for correct points is significantly higher than that for incorrect points, the average similarity can be used to determine whether the fingering is correct. However, due to the varying degrees of error in the violinists' fingering techniques, the average similarity is difficult to measure using a fixed threshold. Therefore, the average value b_i of the average similarity S_{ij} of all points is used, combined with the standard deviation as the threshold for judgment, as shown in equation (21) and equation (22). Finally, data below the threshold are identified as outliers, and their relative differences are recorded and accumulated for the final determination of correct and incorrect fingerings E, as shown in equation (23).

$$b_i = \frac{\sum_{j=1, i \neq j} S_{ij}}{n-1} \tag{21}$$

$$B_t = \frac{\sum_{i=1}^{n} b_i}{n}, S_t = \sqrt{\frac{\sum_{i=1}^{n} (b_i - B_t)^2}{n}} \tag{22}$$

$$E = E + \left| \frac{B_t - S_t - b_i}{B_t - S_t} \right| \tag{23}$$

After determining the correctness of students' violin fingering, correct any incorrect fingering. Correcting incorrect fingering requires using information about the correct positions of the joints in the hand and the correct relative positions to predict the correct positions for the incorrect fingering, thereby achieving fingering correction. Correct techniques can provide accurate joint position information, and correct relative position information can be obtained from the feature descriptions of correct finger techniques. By

averaging the corrected point coordinates calculated from all correct finger techniques, the final corrected point coordinates can be obtained, as shown in equation (24).

$$(x_n, \gamma_n) = \frac{\sum_{i=0}^{n} (x_i, y_i) + L_{in}}{m}, (e_i < k) \tag{24}$$

where x_n is the target fingering, γ_n is the virtual fingering of AR, y_i is the hand joint point of the correct fingering, L_{in} is the fingering sequence to be compared, m is the number of joint points of the correct fingering, k is the threshold set to determine whether the joint points are correct or incorrect, and $e_i < k$ is the relative error of the i^{th} joint point less than the threshold.

5 Experimental results and analyses

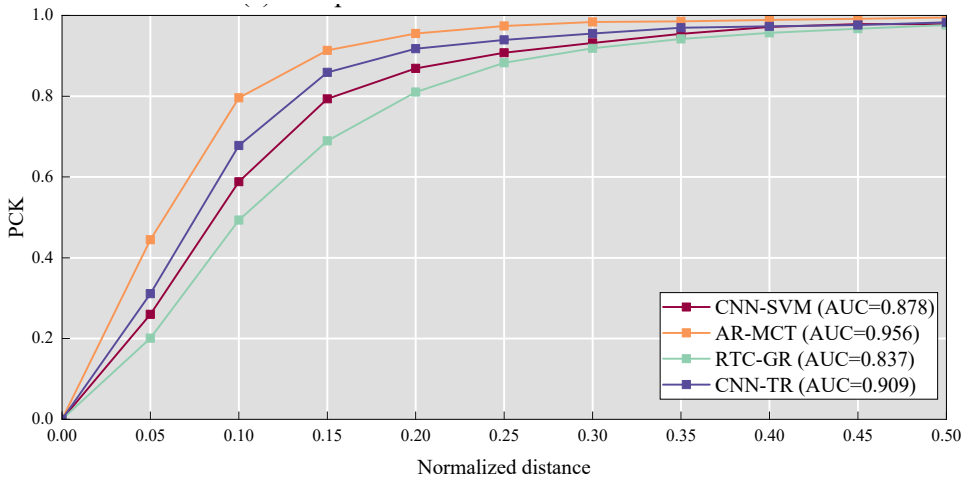
This paper uses the violin fingering teaching dataset in reference (Campo et al., 2023b), which contains synchronised audio, hand movements, and 16,357 accurately annotated fingerings, including nine categories such as classical fingering, jazz fingering, and ethnic fingering. The training set and test set ratio is 8: 2. Extract detailed fingerings for violin playing from audio, including the string being played with the bow, finger numbers, and finger positions. The experiment is based on the PyTorch deep learning framework, using Python 3.7 as the programming language, with a hardware environment consisting of an NVIDIA GeForce GTX 1660 Ti 6GB graphics card, an Intel Core i7-8700 6-core CPU, and 16GB DDR4 memory. The experiment used the Adam optimisation algorithm, with the iteration batch size set to 16, the learning rate set to 0.0001, and the maximum training rounds set to 200.

The top-1 and top-5 correction accuracy rates of the proposed algorithm AR-MCT are compared with those of the baseline algorithms RTC-GR (Pigou et al., 2018), CNN-SVM (Sun et al., 2020), and CNN-TR (Liu and Fu, 2023) as shown in Table 1. The top-1 and top-5 correction accuracy rates of AR-MCT are 94.18% and 97.36%, respectively, which are 13.13% and 11.89% higher than those of RTC-GR, 7.81% and 7.64% higher than those of CNN-SVM, and 3.66% and 4.79% higher than those of CNN-TR. AR-MCT not only utilises MAM and DCNN to accurately recognise violin fingering, but also uses the recognition results for AR hand fingering pose estimation, greatly improving the accuracy of correction for incorrect violin fingering movements and laying a foundation for improving the effectiveness of violin fingering instruction.

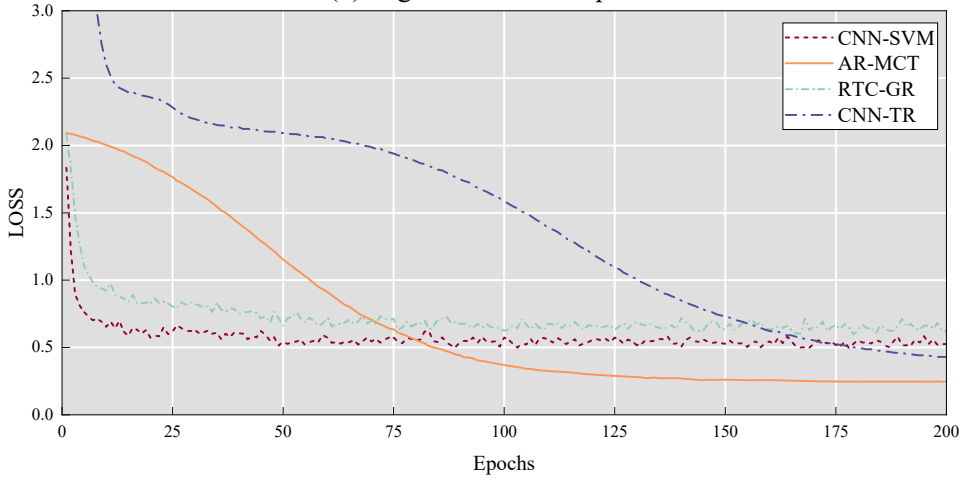
Table 1 Comparison of finger correction accuracy between different teaching algorithms

Algorithm	*top-1*	*top-5*
RTC-GR	81.05%	85.47%
CNN-SVM	86.37%	89.72%
CNN-TR	90.52%	92.57%
AR-MCT	94.18%	97.36%

Figure 5 Performance results for the estimation of fingering joint points of the hand, (a) comparison PCK curve and AUC value (b) algorithm loss comparison (see online version for colours)



(a)



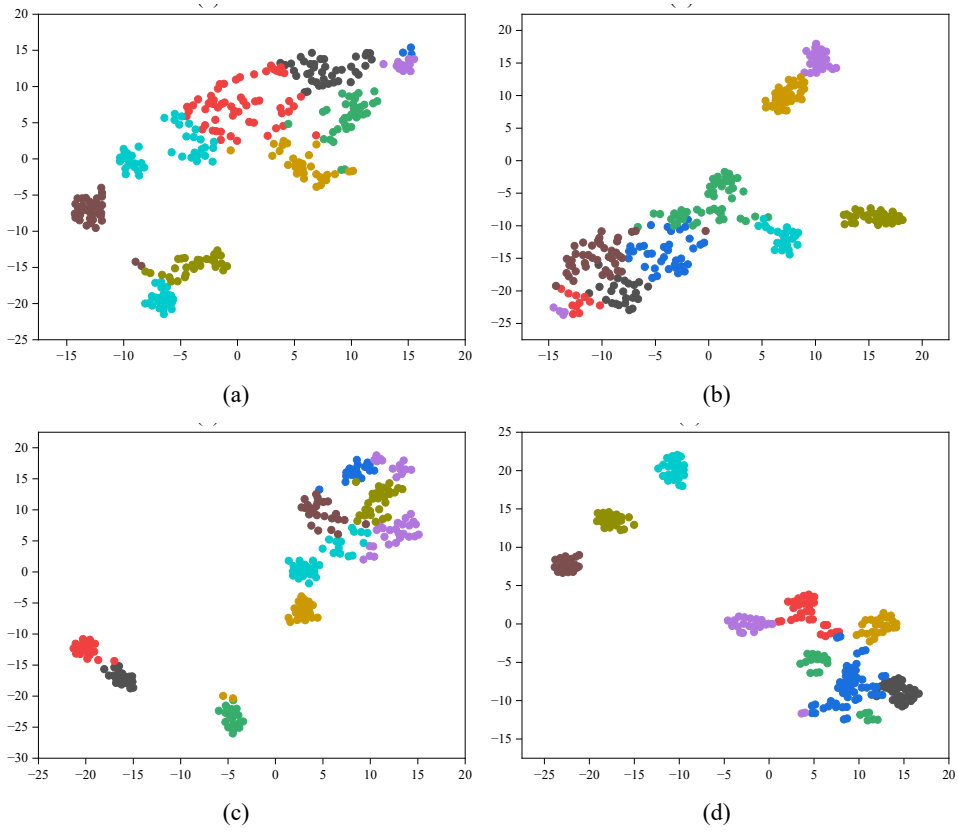
(b)

The PCK curves and AUC values of the correct estimation ratio (PCK) for hand finger joint nodes, as well as the estimation loss comparisons between AR-MCT and RTC-GR, DL-VBA, and CNN-TR, are shown in Figure 5. As shown in Figure 5(a), the horizontal axis represents the normalised distance, with threshold points set at (0, 0.05, …,0.5) for the test settings. The vertical axis denotes the corresponding PCK values. The PCK values of AR-MCT are higher than those of other algorithms, and the AUC is also the largest. When the threshold is 0.15, the PCK value of AR-MCT has reached 91.5%, while RTC-GR, CNN-SVM, and CNN-TR have reached 70.2%, 78.9%, and 83.2%, respectively. Although RTC-GR extracts spatio-temporal features of finger movements through a spatio-temporal dual-stream convolution structure, it does not enhance key features, resulting in poor motion estimation performance. Although CNN-SVM utilises

CNN and SVM to extract features and classify hand fingering images, it does not consider the dynamic spatiotemporal characteristics of violin fingering movements. CNN-TR recognises hand fingering images through CNN-BiLSTM, but its ability to perceive dynamic changes in fingering movements is weak.

The loss values for the four algorithms' finger movement estimation are shown in Figure 5(b). AR-MCT achieved a significant reduction in the first 15 epochs, demonstrating that the algorithm can quickly capture and adapt to key features in the training dataset. As training continued until the 120-th epoch, despite a period of slight fluctuations and adjustments, it ultimately showed a strong convergence trend and stabilised at a relatively low level.

Figure 6 Visualisation of the recognition effect of violin fingering movements, (a) RTC-GR (b) CNN-SVM (c) CNN-TR (d) AR-MCT (see online version for colours)



In addition, this paper also visually demonstrates the recognition performance of each algorithm for the nine types of violin fingering actions in the dataset, as shown in Figure 6. The RTC-GR algorithm identifies various types of finger movements in a relatively scattered manner, and different types of finger movements may be confused. The recognition performance of the CNN-SVM algorithm is superior to that of RTC-GR. It can accurately recognise some finger movement types, but there are still cases of confusion between different finger movement types. When compared with the CNN-TR algorithm, only a few types of finger movements remain unrecognised. The AR-MCT

algorithm achieves the best recognition performance, clearly identifying various types of finger movements and demonstrating the overall best recognition performance.

6 Conclusions

Correct fingering is one of the key factors for successful violin performance. However, the rapid dynamic changes in violin fingering make it challenging for teachers to promptly correct students' incorrect fingering during instruction. To this end, this paper first designs a violin dynamic fingering motion capture module, adopts a multi-head attention mechanism, and utilises multi-scale hollow convolution to enhance the capture of complex dependencies between hand joints, significantly improving recognition performance. After identifying specific finger positions, the hands move during the performance. To locate the position of the hands, an AR-based hand pose estimation module was designed. The POSIT algorithm and point-to-point data are used to estimate the relative position of the fingers in relation to the camera, and the GN method is used to optimise the distance between control points to obtain more accurate virtual finger positions. Finally, cosine similarity is used to compare the virtual fingering with the actual fingering. For incorrect skeletal points, fingering correction is performed based on the feature description of the target fingering, thereby improving the teaching effectiveness. The experimental results indicate that the PCK value of the proposed algorithm reaches 91.5%, enabling accurate correction of incorrect fingering in violin fingering instruction.

Future research can focus on optimising feature extraction of finger movements to further improve the performance of the motion capture module in complex finger movement recognition tasks. Specifically, by introducing a dynamically adjustable multi-scale feature extraction mechanism, the receptive field and expansion rate can be adaptively adjusted according to the complexity of violin fingering to flexibly capture multi-scale features. A feature enhancement strategy is designed for similar fingerings. The feature differences are strengthened through contrastive learning, and the generative adversarial network is utilised to expand the training data and improve the generalization ability.

Declarations

All authors declare that they have no conflicts of interest.

References

Akdeniz, H.B. (2015) 'A comparison of the main features of Suzuki and traditional violin education', *Journal of Literature and Art Studies*, Vol. 5, No. 2, pp.107–113.

Aykut, E. and Taş, S. (2023) 'Gamified violin playing in virtual reality based metaverse environment', *Journal of Emerging Computer Technologies*, Vol. 3, No. 1, pp.7–11.

Bhakar, S. and Bhatt, D.P. (2020) 'Development of augmented reality application to detect the 3D marker through POSIT algorithm', *Journal of Discrete Mathematical Sciences and Cryptography*, Vol. 23, No. 2, pp.365–372.

Campo, A., Michałko, A., Van Kerrebroeck, B. and Leman, M. (2023a) 'Dataset for the assessment of presence and performance in an augmented reality environment for motor imitation learning: a case-study on violinists', *Data in Brief*, Vol. 51, p.109663.

Campo, A., Michałko, A., Van Kerrebroeck, B., Stajic, B., Pokric, M. and Leman, M. (2023b) 'The assessment of presence and performance in an AR environment for motor imitation learning: a case-study on violinists', *Computers in Human Behavior*, Vol. 146, pp.22–31.

D'Amato, V., Volta, E., Oneto, L., Volpe, G., Camurri, A. and Anguita, D. (2020) 'Understanding violin players' skill level based on motion capture: a data-driven perspective', *Cognitive Computation*, Vol. 12, pp.1356–1369.

Dalmazzo, D. and Ramírez, R. (2019) 'Bowing gestures classification in violin performance: a machine learning approach', *Frontiers in Psychology*, Vol. 10, pp.34–41.

Feng, Y. (2023) 'Design and research of music teaching system based on virtual reality system in the context of education informatization', *Plos One*, Vol. 18, No. 10, pp.1–16.

Fonteles, J.H. and Rodrigues, M.A.F. (2021) 'User experience in a kinect-based conducting system for visualization of musical structure', *Entertainment Computing*, Vol. 37, pp.10–17.

Gao, H. and Li, C. (2023) 'Automated violin bowing gesture recognition using FMCW-radar and machine learning', *IEEE Sensors Journal*, Vol. 23, No. 9, pp.9262–9270.

Goldie, S. (2015) 'Features: why the order of finger introduction matters in beginning string instruction', *American String Teacher*, Vol. 65, No. 2, pp.34–37.

Kinoshita, H. and Obata, S. (2009) 'Left hand finger force in violin playing: tempo, loudness, and finger differences', *The Journal of the Acoustical Society of America*, Vol. 126, No. 1, pp.388–395.

Kruger, A. and Jacobs, J. (2020) 'Playing technique classification for bowed string instruments from raw audio', *Journal of New Music Research*, Vol. 49, No. 4, pp.320–333.

Liu, J. and Fu, F. (2023) 'Convolutional neural network model by deep learning and teaching robot in keyboard musical instrument teaching', *Plos One*, Vol. 18, No. 10, pp.21–30.

Nakamura, E., Saito, Y. and Yoshii, K. (2020) 'Statistical learning and estimation of piano fingering', *Information Sciences*, Vol. 517, pp.68–85.

Peinan, Z. and Pattananon, N. (2022) 'The comparison of violin teaching method', *ASEAN Journal of Religious and Cultural Research*, Vol. 5, No. 1, pp.27–32.

Pigou, L., Van Den Oord, A., Dieleman, S., Van Herreweghe, M. and Dambre, J. (2018) 'Beyond temporal pooling: recurrence and temporal convolutions for gesture recognition in video', *International Journal of Computer Vision*, Vol. 126, pp.430–439.

Rosa-Pujazón, A., Barbancho, I., Tardón, L.J. and Barbancho, A.M. (2015) 'A virtual reality drumkit simulator system with a Kinect device', *International Journal of Creative Interfaces and Computer Graphics (IJCICG)*, Vol. 6, No. 1, pp.72–86.

Servi, M., Profili, A., Furferi, R. and Volpe, Y. (2024) 'Comparative evaluation of Intel RealSense D415, D435i, D455 and Microsoft Azure Kinect DK sensors for 3D vision applications', *IEEE Access*, Vol. 12, pp.111311–111321.

Su, A.W. and Liang, S.-F. (2002) 'A class of physical modeling recurrent networks for analysis/synthesis of plucked string instruments', *IEEE Transactions on Neural Networks*, Vol. 13, No. 5, pp. 1137-1148.

Sun, S.-W., Liu, B.-Y. and Chang, P.-C. (2020) 'Deep learning-based violin bowing action recognition', *Sensors*, Vol. 20, No. 20, p.5732.

Wang, Y. (2024) 'China's use of virtual and augmented reality music simulators for teaching music', *Asia Pacific Education Review*, Vol. 10, pp.1–10.

Wang, Y., Wang, G., Chen, C. and Pan, Z. (2019) 'Multi-scale dilated convolution of convolutional neural network for image denoising', *Multimedia Tools and Applications*, Vol. 78, pp.19945–19960.

Yılmaz, R.M. and Göktaş, Y. (2018) 'Using augmented reality technology in education', *Cukurova University Faculty of Education Journal*, Vol. 47, No. 2, pp.510–537.