INVISIBLEPIANO: ENABLING GESTURE-BASED RIGHT-HAND MONOPHONIC PIANO PERFORMANCE USING A SMARTPHONE CAMERA VIA STATISTICAL MACHINE TRANSLATION

Chihiro MITA (g2120539@is.ocha.ac.jp) [1], **Kento WATANABE** (kento.watanabe@aist.go.jp) [2], **Takayuki NAKATSUKA** (takayuki.nakatsuka@aist.go.jp) [2], **Tomoyasu NAKANO** (t.nakano@aist.go.jp) [2], **Masataka GOTO** (m.goto@aist.go.jp) [2], and **Takayuki ITOH** (itot@is.ocha.ac.jp) [1]

[1] **Ochanomizu University**, Japan
[2] **National Institute of Advanced Industrial Science and Technology (AIST)**, Japan

ABSTRACT

This paper proposes InvisiblePiano, an interactive system that enables gesture-based right-hand monophonic piano performance without an actual piano keyboard. The system captures the user's hand movements using an existing hand tracking method, without requiring a physical interface or virtual visual guides such as VR overlays. To achieve musically natural interaction, InvisiblePiano incorporates a pitch estimation model that infers pitch sequences (i.e., individual MIDI note numbers) directly from the tracking results. This model combines an N-gram pitch language model with a translation model based on key press changes and fingering, enabling the mapping of piano-like hand gestures to structured musical output. Experiments using known melodies to evaluate the pitch estimation model achieved a pitch accuracy of 98.43%. A user study with 18 participants confirmed the system's intuitiveness and usability. These results suggest that InvisiblePiano enables accessible and highly accurate gesture-based right-hand monophonic piano performance without an actual piano keyboard or visual guidance.

1. INTRODUCTION

Recent advances in Virtual Reality (VR) and gesture interfaces have enabled piano playing without an actual piano keyboard. Current systems [1–3] map virtual keys onto space and assign pitches based on their positions. Performers visually recognize the presented keys and control the pitch by aligning their hands with the corresponding positions. These approaches require performers to rely on the absolute positions of keys, which requires visual presentation of keyboard layouts and limits accessibility. In contrast, this study proposes a method that estimates pitch sequences based on hand movements and fingering, instead of relying on position-based mappings. This approach removes the need for visual keyboard presentation while maintaining performance accuracy in gesture-based piano interaction.

Copyright: © 2026. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

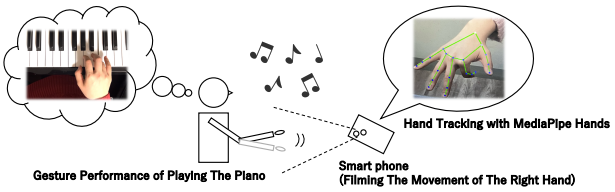


Figure 1. Conceptual diagram of InvisiblePiano, showing gesture-based pitch estimation from camera input.

Previous research has attempted to achieve high accessibility by developing a deep learning-based hand gesture system for air piano playing [4]. However, because this system relies on supervised learning to map diverse hand shapes to specific pitches, it requires a large annotated dataset to capture variations in hand size, finger positions, and movement patterns. In addition, such deep learning models often suffer from interpretability issues, meaning that the reasoning behind their predictions is difficult to explain, which makes error analysis and user feedback challenging. Playing without visually recognizing the keys presents challenges in handling mistouches. Previous mistouch correction systems [5, 6] enable accurate performance without visual guidance, but their reliance on MIDI-compatible hardware limits accessibility.

To address these challenges, this study proposes a probabilistic model based on Statistical Machine Translation (SMT) and integrates it with real-time hand tracking technology (MediaPipe Hands [7]) using only a smartphone camera to develop InvisiblePiano, a portable virtual piano system (Figure 1) that enables users to play music using only hand gestures. The current implementation focuses on right-hand, monophonic (single-note) playing. This model estimates pitch sequences (i.e., sequences of MIDI note numbers) by taking into account not only key press positions but also fingering information and preceding pitch context. This approach aims to enable accessible gesture-based piano performance that does not rely on visual keyboard perception, but still achieves high accuracy (see Sections 3.1 for details).

To evaluate the effectiveness of the pitch estimation model, we (1) analyzed pitch estimation accuracy through quantitative offline experiments and (2) conducted a user study with 18 participants to evaluate whether the system supports vision-independent performance. These results demonstrate that accurate piano performance and pitch estimation are

possible even without an actual piano keyboard.

The main contributions of this research are as follows: (a) We developed InvisiblePiano, a gesture-based right-hand monophonic piano system using only a smartphone camera, without an actual keyboard or visual guidance, improving portability and accessibility. (b) We propose an SMT model for estimating pitch sequences from key press positions and fingering changes. Combined with an existing hand tracking system, this model enables high-accuracy gesture-based right-hand monophonic piano performance. (c) We quantitatively evaluated pitch estimation accuracy in a non-real-time environment and demonstrated the effectiveness of vision-independent performance support through user experiments with 18 participants using InvisiblePiano.

2. RELATED WORK

2.1 Virtual Keyboard Systems

Virtual keyboard systems have emerged alongside advances in VR and gesture interfaces. However, most current systems require visual feedback, haptic devices, or complex setups to accurately detect key press locations [2,3,8]. Efforts have been made toward lightweight solutions [9], which recognize keyboards drawn on paper, and [10], which use a transparent keyboard pattern board with a monocular camera. However, these solutions still rely on the performer's visual reference to physical or visual keyboard aids. In contrast, our system allows the performer to play without looking at the keyboard or the corresponding visual guides. Using only hand gesture input from a smartphone camera, our system provides a simpler and significantly lighter virtual piano environment.

2.2 Mistouch Correction

Existing research on mistouch correction focuses primarily on improving performance accuracy on physical MIDI keyboards or hardware. For example, the Transformer-based Polytune [5] and the recurrent neural network-based model of Marinov et al. [6] achieve high precision in detecting and correcting errors in the MIDI input. Importantly, all of these systems assume the presence of an actual piano keyboard. To enhance accessibility, our research focuses on an essentially different field: building a virtual performance environment that does not assume physical keystrokes but instead estimates musical intervals solely from camera-captured hand gestures.

2.3 Gesture-Based Performance

Systems like MuGeVI [11] use hand gestures to play, which differs from traditional instrument performance techniques. Balaji et al. [12] also proposed an air piano system that maps notes to fingers, preventing the player from freely playing a wide range of pitches. Gusita et al. [4] achieved real-time estimation without mapping notes to fingers using a simple neural network to map relative finger positions to pitches, but their range was limited to ten notes. Importantly, deep learning models often require large annotated datasets and extensive computational resources, and their predictions can be difficult to interpret. Our approach addresses these limitations by using an SMT-based probabilistic model. Instead of assigning each finger to a fixed MIDI pitch, the proposed model estimates pitch sequences from traditional piano-like hand gestures by modeling relative hand movements as pitch transitions. This model is interpretable, requires a small training dataset, and is capable of using hand gestures similar to those used to play an acoustic piano.

3. METHODOLOGY

We describe the design and implementation of the InvisiblePiano system, while the details of the pitch sequence estimation model based on SMT are presented in Section 4.

3.1 InvisiblePiano System Overview

We developed InvisiblePiano, piano performance system that enables gesture-based right-handed monophonic playing using only a smartphone camera. This eliminates the need for an actual piano keyboard or dedicated hardware. Figure 2 shows the overall computational architecture. The system enables free-space performance, independent of physical contact or visual surface references. Figure 1 illustrates this conceptual functionality. This approach offers advantages over traditional virtual keyboard systems that rely on visual feedback, tactile feedback, or dedicated sensing hardware, significantly improving portability and accessibility on general-purpose mobile devices. InvisiblePiano uses MediaPipe Hands [7] for real-time hand tracking. The system detects key presses from skeletal hand coordinate information and obtains key presses and fingerings.

3.1.1 Key Press Detection and Feature Extraction

The system determines key press states using a relative y-coordinate threshold based on finger positions when the hand is at rest. The smartphone camera was positioned in front of the performer, capturing the hand from the fingertip side. The coordinate system follows the MediaPipe Hands image coordinate convention, where the x-axis increases horizontally and the y-axis increases downward. A press event is registered when the y-coordinate of a fingertip landmark (indexed by MediaPipe Hands [7]) rises above the baseline threshold, and a release event when it falls below it. This approach ensures robust detection across different hand sizes. Furthermore, the tone length (rhythm) is determined based on the key press (note-on) and release (note-off) events. The duration of each note is calculated as the time difference between these two events. Once a press is detected, the system records the start and end x-coordinates of the key press, denoted as $x_{n_{\text{start}}}$ and $x_{n_{\text{stop}}}$, along with the finger number f_n, where $n \in \mathbb{N}^+$ denotes the sequence of discrete key press events. These values are used to compute two inputs for the pitch estimation model (Section 4): the horizontal displacement between consecutive key presses, denoted by Δg_n, and the finger transition pair, denoted by F_n as follows:

$$\Delta g_n = x_{n_{\text{start}}} - x_{n-1_{\text{stop}}}, \tag{1}$$

$$F_n = (f_n, f_{n-1}). \tag{2}$$

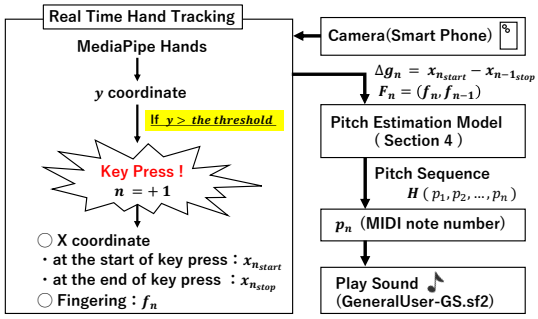


Figure 2. Architecture of InvisiblePiano, including modules for hand tracking, pitch estimation, and MIDI output.

Here, Δg_n represents the horizontal displacement between the current key press and the previous key release, while F_n denotes the ordered pair of finger numbers used for the current and previous key presses. The MIDI pitch at step n is represented by p_n, and the initial pitch p_0 is assumed to be known. Note that while the coordinates are derived from the pixel space of the camera image, the displacement Δg_n is normalized by a configurable reference width (W), corresponding to the size of a virtual white key. This normalization ensures consistent performance across different camera resolutions and screen sizes.

3.1.2 Real-Time Pitch Feedback and Implementation

The estimated pitch p_n is played in real time using FluidSynth with the GeneralUser-GS.sf2 SoundFont. Furthermore, our system uses an SMT model to predict note sequences from hand movements; the details of this pitch sequence estimation model are described in Section 4. Because SMT uses a lightweight probabilistic model instead of complex deep learning computations, the system achieves low computational cost and minimal latency, which are essential for real-time performance. While the core capability of the system is free-space performance, as illustrated in Figure 1, Experiment 2 (Section 5) employed a static paper keyboard reference placed on a desk. This experimental choice was made solely to provide a consistent tactile anchor for participants, ensuring performance consistency and simplifying experimental control of hand position, without compromising the reliance of the core technology on relative hand gesture tracking.

4. PITCH ESTIMATION MODEL FOR INVISIBLEPIANO

For InvisiblePiano, we propose a probabilistic model to estimate a pitch sequence $H = (p_0, p_1, \cdots, p_T)$ where T is the length of the performance (the number of detected key press events). This estimation problem is formulated as a sequence-to-sequence translation task inspired by SMT.

We first represent a hand gesture sequence as a combination of two complementary components: key press change information G and fingering information F. Given this representation, the ultimate goal of the model is to estimate the most probable pitch sequence $\hat{H}$ conditioned on the observed hand gesture components G and F:

$$\hat{H} = \underset{H}{\operatorname{argmax}}\, P(H \mid G, F). \tag{3}$$

Due to the complexity of calculating the full sequence probability directly, this estimation is performed by maximizing the cumulative score of local transitions. This is achieved via a sequential beam search algorithm, where the log-probability is approximated by the sum of local scores $\mathrm{Score}(p_n)$. The overall optimization is approximated as:

$$\hat{p}_n = \underset{p_n}{\operatorname{argmax}} \log \mathrm{Score}(p_n). \tag{4}$$

Here, $\mathrm{Score}(p_n)$ is the local score calculated at each time step based on the previous context. To focus on the core challenges of gesture-based pitch estimation, this study targets monophonic right-hand melodies on white keys, with each pitch transition constrained to within one octave.

4.1 Local Score: Integration of Components

The local score, $\mathrm{Score}(p_n)$, represents the score of a candidate pitch p_n at the time step n given both the gesture-related observations and the musical context. Following the framework of SMT, this score is computed by integrating two probabilistic components: the **Translation Model (TM)** (Section 4.2), which evaluates the likelihood of the observed hand gesture and fingering given a candidate pitch transition, and the **Pitch Language Model (PLM)** (Section 4.3), which captures the prior probability of the pitch based on the preceding pitch context. Formally, the local score is defined as:

$$\mathrm{Score}(p_n) = \alpha \underbrace{P(\Delta p_n)}_{\text{TM}} + \beta \underbrace{P(p_n \mid p_{n-N}^{n-1})}_{\text{PLM}}, \tag{5}$$

where α and β are treated as hyperparameters in our model.

4.2 Translation Model (TM)

TM provides the gesture-based likelihood component used in calculating the local score $\mathrm{Score}(p_n)$. It estimates the probability of G and F given a candidate pitch transition. G is represented by the horizontal key press coordinate and its displacement $\Delta g_n = x_{n_{\text{start}}} - x_{n-1_{\text{stop}}}$. F is denoted by the finger index $f_n \in \{1, 2, 3, 4, 5\}$, and its 2-gram fingering information is expressed as $F_n = (f_n, f_{n-1})$. $\Delta p_n = p_n - p_{n-1}$ represents the relative pitch transition. The prior probability $P(\Delta p_n)$ is assumed to be uniform across all possible pitch changes and can therefore be omitted (i.e., treated as a constant). Accordingly, the local likelihood is defined as follows:

$$\begin{aligned} P(\Delta p_n) &:= P(\Delta p_n \mid \Delta g_n, F_n) \\ &\approx P(\Delta g_n, F_n \mid \Delta p_n). \end{aligned} \tag{6}$$

In implementation, this joint likelihood is decomposed into two components: the Action Model and the Fingering Model. Assuming that key press displacement (Δg_n) and fingering information (F_n) are conditionally independent given pitch transition (Δp_n), the joint likelihood is approximated as:

$$P(\Delta g_n, F_n \mid \Delta p_n) \approx \underbrace{P(\Delta g_n \mid \Delta p_n)}_{\text{Action Model}} \cdot \underbrace{P(F_n \mid \Delta p_n)}_{\text{Fingering Model}}. \tag{7}$$

Here, $P(\Delta g_n \mid \Delta p_n)$ represents the Action Model, and $P(F_n \mid \Delta p_n)$ represents the Fingering Model.

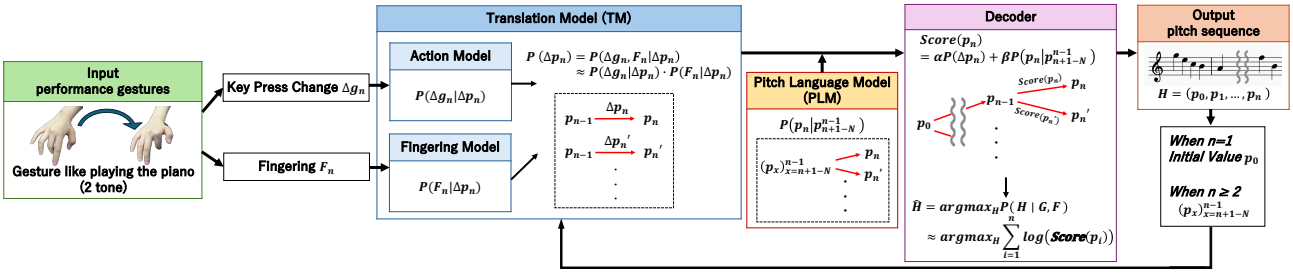


Figure 3. Overview of the pitch estimation model based on SMT (n: time step, N: context length of the N-gram model).

4.2.1 *Action Model*

The action model $P(\Delta g_n \mid \Delta p_n)$ evaluates the probability that a key press displacement Δg_n is observed given a pitch transition Δp_n. Following previous research on automatic fingering estimation using HMMs [13], this model is defined using a Gaussian distribution. The value $d(\Delta p_n)$ represents the displacement, specifically the distance between the center of the white key corresponding to p_{n-1} and the center of the white key corresponding to p_n. $\mu_{d(\Delta p_n)}$ denotes the mean of this displacement $d(\Delta p_n)$ and $\sigma_{d(\Delta p_n)}$ denotes the standard deviation of $d(\Delta p_n)$. Using these definitions, the action model is written as:

$$P(\Delta g_n \mid \Delta p_n) = \mathcal{N}(\Delta g_n \mid \mu_{d(\Delta p_n)}, \sigma^2_{d(\Delta p_n)}). \quad (8)$$

The parameters $\mu_{d(\Delta p_n)}$ and $\sigma_{d(\Delta p_n)}$ are determined by the conditional pitch transition Δp_n. We apply a clipping operation to Δg_n, assuming no sudden large pitch jumps. W represents the width of the white key. We restrict Δg_n to the interval $[-7W, 7W]$ using the following clipping function:

$$\Delta g_n^{\text{clip}} = \max(-7W, \min(\Delta g_n, 7W)). \quad (9)$$

This procedure ensures numerical robustness by limiting the influence of extreme values while maintaining the overall behavior of the hand gesture signal within a reasonable range. Furthermore, considering that predicting key positions for larger pitch transitions can be difficult in situations where visual confirmation is unavailable, the standard deviation $\sigma_{d(\Delta p_n)}$ of the action model $P(\Delta g_n \mid \Delta p_n)$ is dynamically set based on the amount of displacement $d(\Delta p_n)$ as follows:

$$\sigma_{d(\Delta p_n)} = \begin{cases} 0.5W & \text{if } |d(\Delta p_n)| \leq W, \\ 1.0W & \text{if } W < |d(\Delta p_n)| \leq 4W, \\ 1.5W & \text{if } 4W < |d(\Delta p_n)| \leq 7W. \end{cases} \quad (10)$$

This dynamic setting models the increasing uncertainty (variance) associated with larger displacements in a visually unconstrained environment. Figure 4 shows an example of this distribution. Note that W is a parameter that can be freely set depending on the simulated or actual playing environment.

4.2.2 *Fingering Model*

To consider structural constraints of the hand, the probability distribution of pitch transition Δp_n for each 2-gram fingering information $F_n = (f_n, f_{n-1})$ is constructed as the fingering model. We consider all 25 possible fingering information, and estimate the probabilities from occurrence frequencies in training data. Since the model constrains each pitch transition to within one octave, different categorical distributions are defined depending on the sign of Δg_n. When $\Delta g_n \geq 0$, $\Delta p_n \in \{0, 1, \ldots, 12\}$; when $\Delta g_n < 0$, $\Delta p_n \in \{0, -1, \ldots, -12\}$.

4.3 Pitch Language Model (PLM)

The Pitch Language Model (PLM) is an N-gram model that predicts the probability of each pitch based on the preceding pitches, ensuring musically plausible sequences. This is achieved using a context-dependent N-gram model, $P(p_n \mid p_{n-N}^{n-1})$, which is constructed from the training data. The PLM is crucial for suppressing musically unnatural or improbable pitch sequences that might otherwise be favored by relying solely on the local hand gesture likelihood (TM).

4.4 Real-Time Decoding with TM-PLM Score Integration

Because the system operates in real time, the decoding algorithm must determine the pitch to be played at each step n without waiting for the entire sequence. At each step, we calculate the local score $\text{Score}(p_n)$ as a weighted sum of the log-probabilities from the TM and PLM models (Equation 4). Therefore, we use beam search to retain the most promising partial paths based on the combined score and immediately output the pitch with the highest probability at the current step as the played note.

5. EVALUATION OF PITCH ESTIMATION MODEL

The primary research question addressed in this experiment was: *RQ1: How accurately can the proposed model estimate the pitch sequence?* To evaluate this, we conducted simulations without real-time constraints, aiming to quantitatively assess model accuracy. We did not perform direct comparisons with existing systems [4–6], as they differ fundamentally in input modalities and task definitions. The main metric was the sequence match rate between the estimated pitch sequence and the ground truth.

5.1 Data and Configuration

We prepared training data for PLM from "The World of Nursery Rhymes and Nursery Songs" [14]. These included the melodic sections of 392 children's songs, with an average of approximately 200 notes per song. All songs were monophonic, transposed to C major, and excluded notes with accidentals, focusing only on white key pitches. Notes

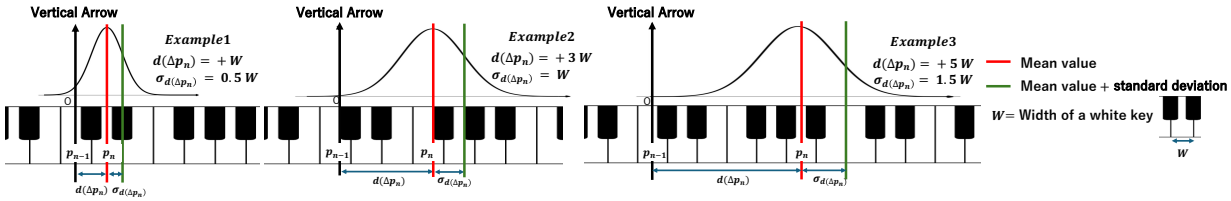


Figure 4. Gaussian distribution representing the probabilistic output of the Action Model.

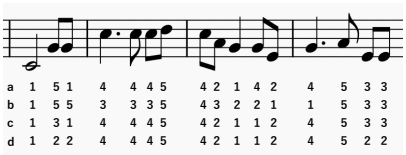


Figure 5. Example fingering for a phrase in "Oborozukiyo" (Taki). a, b, c, d are fingering patterns.

in different octaves were treated as identical. For the PLM, we constructed multiple N-gram models, ranging from 2-grams to 10-grams. Our approach is to use the highest possible model for which a corresponding N-gram exists, rather than backoff smoothing.

We used the PIG dataset [15] to train Fingering Model. This model was adjusted with a heuristic weight to compensate for toward natural fingering:

- If the finger distance $d = f_n - f_{n-1}$ and the key distance $(\Delta g_n/W)$ satisfied $|d - (\Delta g_n/W)| = 0$, the probability was multiplied by a weight of 1.15.
- If $|d-(\Delta g_n/W)| = 1$, the probability was multiplied by a weight of 1.10.

After all, we normalize the distribution to sum to 1.

We prepared test data consisting of 13 pieces categorized as Fully Contained Pieces (5 pieces, 448 notes), Partially Contained Pieces (3 pieces, 326 notes), or Unseen Pieces (5 pieces, 455 notes, classical genre) based on the degree of inclusion in the training data. We then generated test inputs by adding random noise to the key press changes according to score and combining them with fingering patterns provided by four expert performers. We applied the random noise as a continuous value from a uniform distribution, converted to W, based on the following rules:

- If $|\Delta g_n| \leq W$: Range of $-0.5W$ to $+0.5W$ widths.
- If $W < |\Delta g_n| \leq 4W$: Range of $-W$ to $+W$ widths.
- Otherwise: Range of $-1.5W$ to $+1.5W$ widths.

Figure 5 shows an example of the fingering for the test data. In this experiment, we assumed W to be 20 mm. The global hyperparameters were set to $\alpha = 2/3$ and $\beta = 1/3$.

To reduce instability in sequential pitch estimation, we introduced a 15% correction factor that boosts the score of the highest-scoring path at each step. Table 1 compares beam search paths with and without this correction. The 15% factor effectively suppresses frequent path switching, resulting in a more consistent estimated pitch sequence.

5.2 Results and Discussion

The average match rate for Fully Contained Pieces is 98.43%, demonstrating the effectiveness of the proposed model for recovering pitch sequences even with slight key press misalignments when the musical context is known.

Table 1. Comparison of beam search paths with and without a 15% correction factor for an unseen Telemann piano piece. True pitch sequence: [67, 64, 67, 62, 67, 60, 72, 71, 69].

Step	0% correction path	15% correction path
4	[67, 64, 67, 62]	[67, 64, 67, 62]
5	[67, 67, 72, 65, 70]	[67, 67, 72, 65, 70]
6	[67, 64, 67, 64, 69, 69]	[67, 64, 67, 62, 67, 67]
7	[67, 64, 67, 64, 69, 67, 70]	[67, 64, 67, 62, 67, 67, 67]
8	[67, 67, 67, 67, 72, 71, 71, 71]	[67, 64, 67, 62, 67, 67, 67, 67]
9	[67, 67, 67, 67, 72, 71, 71, 69, 69]	[67, 64, 67, 62, 67, 67, 67, 67, 67]

Note: Fingering displacements (mm): [0, -30.86, 28.46, -53.79, 55.98, -96.37, 128.73, -10.91, -26.52]

In contrast, the average match rate for Partially Contained Pieces (32.63%) and the average match rate for Unseen Pieces (18.07%) is low, highlighting a clear challenge in generalizing this model to novel melodic sequences.

5.2.1 *Effect of Fingering Model*

According to Table 2, No Fingering Model (only Action Model) often failed (e.g., Fingering a, Piece 1: 18.75%), demonstrating that relying solely on key press change is insufficient. This confirms that mediating through the intermediate semantic representation of fingering is essential for stable and accurate pitch restoration.

Visualized key press change in successful trials (100%) with Proposed Model (Figure 6) indicates that it successfully absorbs input noise.

The analysis of lower match rates, particularly in specific trials within the Partially Contained Pieces and Unseen Pieces categories, revealed key contributing factors: transitions across rests (utilized for unusual positional shifts) and complex, specialized finger leaps. This suggests a necessity to expand the Fingering Model to robustly accommodate a wider variety of specialized or genre-atypical gestures. Crucially, the ability of our model to precisely attribute performance degradation to specific interpretable factors (like rest utilization or complex finger leaps) is a significant advantage over complex black-box deep learning models, facilitating clearer pathways for future model improvement.

5.2.2 *Impact of Context Length on Generalization*

Analysis of Partially Contained Pieces revealed that estimation errors were reduced when PLM order was decreased from 10-gram to 2-gram. Table 3 shows the improvement in relative pitch match rate (focusing on accuracy of Δp_n, ignoring absolute drift).

Table 2. Pitch Match Rate (%) in Fully Contained Pieces: Proposed Model vs. No Fingering Model (Underscore indicates higher match rate).

ID	Proposed Model	No Fingering Model
a_1	100.00	18.75
a_2	100.00	21.43
a_3	92.70	23.36
a_4	100.00	100.00
a_5	100.00	100.00
b_1	100.00	3.12
b_2	100.00	8.57
b_3	100.00	11.68
b_4	93.02	27.13
b_5	100.00	100.00
c_1	100.00	100.00
c_2	91.43	42.86
c_3	100.00	100.00
c_4	96.12	100.00
c_5	100.00	26.09
d_1	100.00	6.25
d_2	100.00	22.86
d_3	100.00	100.00
d_4	95.35	100.00
d_5	100.00	26.09

Note: ID = Fingering ID + Piece ID (e.g., a_1: Fingering a, Piece 1).

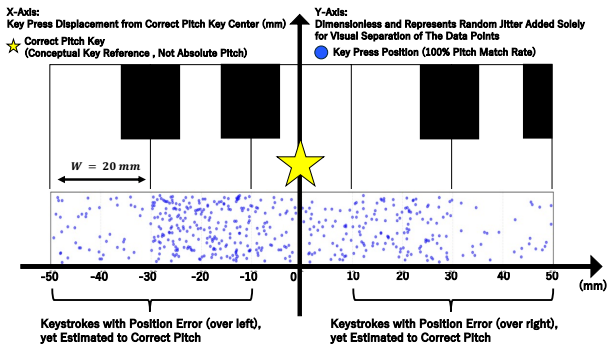


Figure 6. Distribution of key press achieving 100% pitch match. X-axis: key press displacement (mm) relative to the intended key center (0 mm). Y-axis: dimensionless and incorporates jitter for clarity. The scattered data outside the intended key center demonstrates the model's ability to compensate for key press change errors.

This improvement (up to +45.95 pp) with the 2-gram model confirms that overly sparse higher-order N-grams limit generalizability for novel sequences. This suggests that TM correctly provides the directionality, while a less constrained Language Model (2-gram) allows for novel transitions that are still musically plausible.

The low average match rate for Unseen Pieces (18.07%) indicates a failure to generalize to new genres/melodies. Table 4 provides detailed results of the entire sequence of Unseen Pieces (classical genre) by piece. However, accuracy was consistently higher when limited to short initial sequences (e.g., the first 12 notes), suggesting the model operates more stably when contextual prediction is less dominant. Table 5 provides the pitch match rates for the first 12 notes of the Unseen Pieces. We identify three critical areas for improvement: optimizing N-gram integration, expanding the translation model to accommodate wider fingering variation, and incorporating cross-genre learning to achieve better generalization.

Table 3. Comparison of Relative Pitch Match Rate (%) in Partially Contained Pieces (pp indicates percentage point difference/improvement).

ID	10-gram	2-gram	pp
a_1	44.78	53.73	+8.95
a_2	62.76	76.55	+13.79
a_3	19.82	65.77	+45.95
b_1	44.78	64.18	+19.40
b_2	67.59	71.03	+3.44
b_3	21.62	53.15	+31.53
c_1	44.78	61.19	+16.41
c_2	71.03	80.69	+9.66
c_3	28.83	59.46	+30.63
d_1	16.42	55.22	+38.80
d_2	62.07	66.21	+4.14
d_3	28.83	56.76	+27.93

Note: ID = Fingering ID + Piece ID (e.g., a_1: Fingering a, Piece 1).

Table 4. Pitch Match Rate (%) for full sequence of Unseen Pieces.

Piece (ID)	a	b	c	d
1	22.50	32.50	30.00	-
2	11.11	11.11	17.59	-
3	7.50	20.00	6.25	13.75
4	12.50	12.50	23.86	12.50
5	25.42	25.42	16.95	23.73

Table 5. Pitch Match Rate (%) for the first 12 notes of Unseen Pieces, higher match rates compared to Table 4.

Piece (ID)	a	b	c	d
1	58.33	66.67	50.00	–
2	83.33	83.33	83.33	–
3	58.33	41.67	41.67	33.33
4	58.33	58.33	66.67	58.33
5	83.33	83.33	83.33	75.00

6. USER STUDY OF INVISIBLEPIANO

The primary research question addressed in this experiment was: *RQ2: How does InvisiblePiano affect accuracy and user experience without visual guidance of the keyboard, compared to a system that relies solely on key press position?* We conducted a user study approved by the Ochanomizu University Ethics Committee. Participants interacted with InvisiblePiano in conditions where the keyboard could not be visually monitored, and we measured both performance accuracy and subjective user experience.

6.1 Participants

Eighteen participants (women in their 20s) with varying levels of piano experience participated in the task (4 advanced, 7 intermediate, and 7 beginner participants). Hand span measurements confirmed that the participants had a

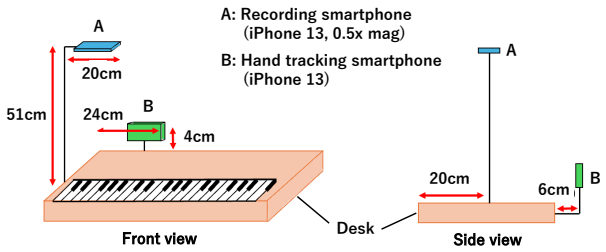


Figure 7. Details of the experimental setup

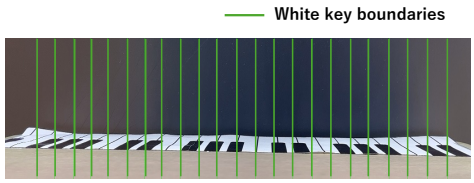


Figure 8. Key range correspondence in the baseline. The green line marks the boundary between the white keys.

sufficiently diverse range of hand spans, covering the 2.5th to 97.5th percentiles of adult male hand span data [16].

6.2 Apparatus

Two smartphones were used for the setup (one for hand tracking and one for recording). Details are shown in Figure 7. Participants were seated at a desk facing a paper keyboard placed on the tabletop, as illustrated in Figure 7. In this experiment, two conditions were set for comparison.

- **Baseline:** A gesture-based piano performance system using MediaPipe Hands [7], where pitch is determined solely by the x-coordinate of the key press position.
- **InvisiblePiano:** A gesture-based piano performance system using MediaPipe Hands [7], where pitch is estimated from key press transitions and fingering using a statistical model, enabling compensation for positional errors.

The baseline condition was intentionally designed without any compensation mechanism in order to isolate the effect of the proposed model. This allows us to evaluate whether performance improvements are attributable to the model-based pitch estimation rather than the positional accuracy of key press. Landmark coordinates (MediaPipe Hands) were obtained in a 1920×1080 pixel image, and the width per white key was set to 66.92 pixel. The keyboard boundaries of Baseline were set based on this width (Figure 8). The weighting parameters α and β of the InvisiblePiano in Eq. 5 were set to $2/3$ and $1/3$, respectively. The N-gram of the Pitch Language Model was set to a maximum of 10.

6.3 Task

The task was to play a passage from "Happy Birthday to You" (11 notes) in C major, a genre outside the model's training data. Figure 9 shows the score for the assigned piece. Before the experiment began, the subjects sat at a desk and quietly practiced moving their fingers while looking at a paper keyboard attached to the desk (W: 2.0 mm). Once the subjects deemed their practice sufficient, they

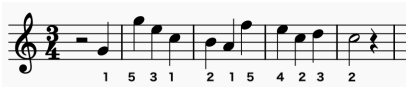


Figure 9. "Happy Birthday to You" in C major, with indicated finger numbers

performed the actual tasks, first the Baseline and then InvisiblePiano, without looking at the paper keyboard and without replaying. This fixed order was intentionally chosen to allow participants to first experience the difficulty of key press errors in the Baseline condition, which lacks compensation, and to use this as a cognitive reference point for evaluating the proposed system. Although a positive learning effect might be expected due to this order, the results showed that key press accuracy in the second (InvisiblePiano) condition was in fact lower than in the Baseline condition (see Table 6), suggesting that the order effect did not lead to an overestimation of performance.

6.4 Measures

Performance accuracy was evaluated using the sequence match rate between the pitch sequence played by participants and the correct pitch sequence from the score. Participants also completed a subjective questionnaire, which asked them which system—Baseline or InvisiblePiano—they found easier to play.

6.5 Results and Discussion

Table 6 summarizes mean (M), median (Md), and standard deviation (SD) for each condition. Across 18 participants, InvisiblePiano achieved substantially higher pitch-match accuracy (M=87.88%, Md=100%) than the baseline (M=61.83%, Md=64%). Re-evaluating the same performances using only key press positions (baseline logic) reduced the match rate to 49.61%, indicating that improvements are due to the compensation mechanism rather than more precise finger placement. A Wilcoxon signed-rank test on the participant-wise pitch-match rates confirmed that this improvement was statistically significant ($p < 0.01$).

The relatively large standard deviation in the corrected output (21.82%) suggests inter-participant variability, consistent with the observation that while most participants benefited from the compensation, a few showed more moderate improvements. Fifteen out of 18 participants rated InvisiblePiano as easier to play (two preferred the baseline, and one rated both equally), supporting the system's usability from a user experience perspective. One beginner commented, "All I had to do was remember which fingers to use and the direction of movement," suggesting that the system affords high accessibility and allows performers to focus on musical intent rather than precise finger placement.

The study identified two primary issues: slight output latency caused by computational correction, and hand-gesture recognition errors, such as misidentifying non-pressing fingers as active presses. These findings indicate the need to further strengthen the robustness of the hand-tracking module. Moreover, the compensation mechanism can obscure the performer's awareness of finger-placement inaccuracies, potentially limiting the system's suitability for practice

Table 6. Statistical Summary of Pitch Match Rates (%). Baseline Logic is results based only on key positions for performances using InvisiblePiano.

Condition	Mean	Median	SD
Baseline	61.83	64	18.71
InvisiblePiano	87.88	100	21.82
Baseline Logic	49.61	55	17.11

contexts that require technical skill development. Future improvements should also focus on integrating dynamics and timbre control derived from expressive performance gestures [17].

7. CONCLUSIONS

We propose InvisiblePiano, a gesture-based right-handed monophonic piano performance system using a smartphone camera. We summarize the main insights from Experiments 1 and 2. Experiment 1 addressed RQ1 on pitch sequence estimation, and Experiment 2 addressed RQ2 on performance and user experience without visual keyboard feedback.

In Experiment 1, the proposed model achieved high accuracy for melodies included in the training data, but performance decreased for unfamiliar performers and unseen pieces. This reflects the influence of both the Fingering Model and the Pitch Language Model on training data dependence. Furthermore, accuracy was affected by the context length of the Pitch Language Model, highlighting the need for model improvements to enhance generalization to unseen melodies. In Experiment 2, InvisiblePiano reproduced intended pitches more accurately than a baseline model based solely on key press positions. Participants also reported that InvisiblePiano felt easier and more intuitive, demonstrating accessibility even for beginners.

Future work includes optimizing fingering representation, expanding the pitch language model to diverse musical genres, improving real-time responsiveness and key press detection, and incorporating expressive musical parameters such as dynamics and timbre control. In addition, future research should investigate the extension of the proposed approach to polyphonic performance, which requires simultaneous estimation of multiple finger-pitch correspondences. While current hand-tracking technology can reliably detect both hands, extending the probabilistic framework to model bimanual interactions and left-hand fingering remains a significant challenge. The low performance observed for unseen classical pieces also suggests the need for larger and more diverse training corpora to improve generalization beyond nursery-rhyme-based melodic material.

Acknowledgments

We thank all participants for their time and feedback.

8. REFERENCES

[1] V. Wyk *et al.*, "A multimodal gesture-based virtual interactive piano system using computer vision and a motion controller," in *Proc. ACIRS2017*, 2017, pp. 1–6.

[2] Y. Fu *et al.*, "A haptic feedback glove for virtual piano interaction," *Virtual Reality & Intelligent Hardware*, vol. 7, no. 1, pp. 95–110, 2025.

[3] H. Liang *et al.*, "Barehanded music: Real-time hand interaction for virtual piano," in *Proc. I3D '16*, 2016, pp. 87–94.

[4] C. Guşită *et al.*, "A novelty real-time gesture recognition model for air-hand piano playing using mediapipe," in *Proc. ICCP2024*, 2024, pp. 3–10.

[5] B. S.-H. Chou *et al.*, "Detecting music performance errors with transformers," in *Proc. AAAI-25*, vol. 39, no. 22, 2025, pp. 23 687–23 695.

[6] G. Marinov, "Real-time error correction and performance aid for MIDI instruments," *arXiv preprint arXiv:2011.13122*, 2020.

[7] F. Zhang *et al.*, "MediaPipe Hands: On-device real-time hand tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[8] Y. Feng, "Design and research of music teaching system based on virtual reality system in the context of education informatization," *PLoS ONE*, vol. 18, no. 10, 2023, article e0285331, total 16 pages.

[9] H. Begum *et al.*, "Digital Beethoven — An Android based virtual piano," in *Proc. ICET2017*, 2017, pp. 1–5.

[10] Y. Wang and L. Song, "Virtual piano system based on monocular camera," in *Proc. ICIC 2021*, ser. Lecture Notes in Computer Science, vol. 12837, 2021, pp. 112–121.

[11] Y. Yang *et al.*, "MuGeVI: A multi-functional gesture-controlled virtual instrument," in *Proc. NIME2023*, 2023, pp. 536–541.

[12] A. Balaji *et al.*, "Air piano interaction via flex sensors and computer vision on edge devices," in *Proc. AIIoT*, 2024, pp. 1–6.

[13] Y. Yonebayashi *et al.*, "Automatic decision of piano fingering based on hidden Markov models," in *Proc. IJCAI-07*, 2007, pp. 2915–2921.

[14] World of Nursery Rhymes and Songs, "World of nursery rhymes and songs," https://www.s-pst.info/douyou-syouka/, accessed 26 March 2026.

[15] E. Nakamura *et al.*, "Statistical learning and estimation of piano fingering," *Information Sciences*, vol. 517, pp. 68–85, 2020.

[16] M. Kouchi, "AIST Japanese Hand Measurement Data: List of Measurement Items," https://www.airc.aist.go.jp/dhrt/hand/data/list.html, accessed 26 March 2026.

[17] N. Aroso, "Role of movement and gesture in communicating music expressiveness to an audience: An experiment on dynamics perception after a contemporary percussion performance," *Convergences - Journal of Research and Arts Education*, vol. 13, no. 25, pp. 83–94, 2020.