# Marker-removal Networks to Collect Precise 3D Hand Data for RGB-based Estimation and its Application in Piano

Erwin Wu
Tokyo Institute of Technology
Tokyo, Japan
wu.e.aa@m.titech.ac.jp

Hayato Nishioka
Sony Computer Science Laboratory
Tokyo, Japan
nishioka@csl.sony.co.jp

Shinichi Furuya
Sony Computer Science Laboratory
Tokyo, Japan
furuya@csl.sony.co.jp

Hideki Koike
Tokyo Institute of Technology
Tokyo, Japan
koike@c.titech.ac.jp

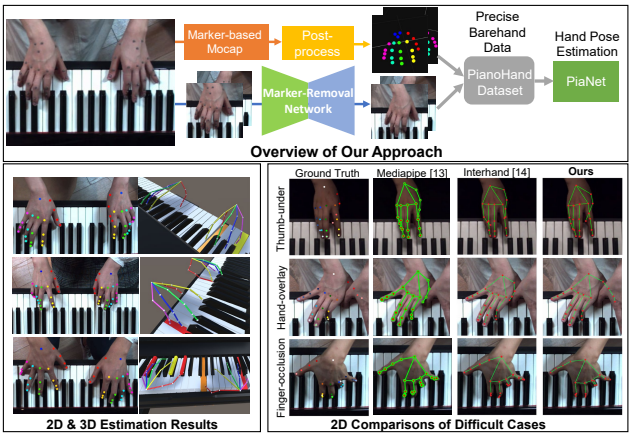


Figure 1. System overview and examples of estimated hand poses.

## Abstract

*Hand pose analysis is a key step to understanding dexterous hand performances of many high-level skills, such as playing the piano. Currently, most accurate hand tracking systems are using fabric-/marker-based sensing that potentially disturbs users' performance. On the other hand, markerless computer vision-based methods rely on a precise bare-hand dataset for training, which is difficult to obtain. In this paper, we collect a large-scale high precision 3D hand pose dataset with a small workload using a marker-removal network (MR-Net). The proposed MR-Net translates the marked-hand images to realistic bare-hand images, and the corresponding 3D postures are captured by a motion capture thus few manual annotations are required. A baseline estimation network PiaNet is introduced and we report the accuracy of various metrics together with a blind qualitative test to show the practical effect.*

## 1. Introduction

Hand motion analysis is one of the most essential techniques for characterizing human behavior, elucidating its underlying mechanism, and acquiring specific skills. Currently, most of the high precision hand tracking methods are either glove-based [2, 26], or marker-based [3], which attach special sensors or markers to the hand. However, these are difficult to equip and too bulky for dexterous skills[20]. On the other hand, many computer vision-based methods utilize convolutional neural networks (CNN) to estimate 3D hand motion from images, however, these methods rely on a robust dataset with high-quality ground truth, which is currently insufficient in many specific areas.

Some predecessors [15, 22] make great efforts in collecting large-scale data with ground truth, but to achieve this, they set up studios with hundreds of cameras for bootstrapping and hire a large number of people to perform manual annotations, which is both times- and cost-consuming. On the other hand, synthetic methods [35, 17] using 3D simulations are introduced to reduce the workload of data collection. However, the domain gap between artificial data and a real one is still enormous for precise hand pose estimations.

Most models trained with these general-purpose datasets are robust across activities but less accurate on some dexterous skills. One representative example is piano playing [9, 8] that includes very unique and awkward hand poses and heavy self-occlusions (such as a thumb-under or hand-overlay motion in Fig 1.), which is in a class by the pianists and therefore does not exist in any current dataset, to the best of our knowledge. Nevertheless, a degree-perfect hand motion analysis is required to reconstruct the performance and to provide feedback, which is difficult for conventional general methods.

In this paper, taking piano playing as an example, we introduce a novel method to collect precise 3D hand pose datasets by means of a marker-removal network to remove the reflective markers from the images. Instead of a studio with hundreds of cameras, the proposed method only requires a marker-based motion capture (MoCap) system (consisting of 12 high-speed cameras) and removes the markers which cause extra features. Inspired by various image-to-image translation tasks such as denoising network [11] or generative adversarial networks (GANs) [17, 34], we develop an encoder-decoder network using automatic marker-synthesizing to translate the markers on the images, called Marker-removal network (MR-net). As a result, realistic bare-hand images with accurate 3D data can be obtained. The collected data are further processed with interpolation and joint re-targeting. For the dataset, we invite 21 experienced pianists to perform different tasks consisting of various hand motions in two different studios. A total number of 2.5M images are collected.

Finally, to demonstrate the benefit of our dataset, a baseline called PiaNet is trained to regress root-relative 3D hand joint positions. Three quantitative experiments are performed to compare our network in different conditions. Besides conventional quantitative metrics such as PCK, MPJPE, and MPJPA, a qualitative study is also performed by inviting experienced pianists for a blind test to rank the estimated 3D hand pose from online piano videos. In summary, this paper illustrates a whole pipeline of how to collect a precise posture dataset using the proposed MR-Net and how to use the data to train a strong baseline that outperforms other SOTA by using one specific scenario as a representative – piano playing. Our contributions can be concluded as follows:

- A marker-removal network for translating marked-hand to bare-hand images results in few domain gaps compared to real-world data and can be applied to many other marker-based motion captures.
- Using the MR-Net, we collect the PianoHand2.5M, which is the first large-scale precise 3D hand image dataset for piano scenarios from professional pianists.
- A strong baseline PiaNet for 3D hand pose estimation in piano playing, which outperforms some state-of-the-art methods in various metrics.
- A qualitative study indicates that PiaNet achieves good results in practical use such as online videos.

## 2. Related Work

### 2.1. RGB-based 3D Hand Pose Dataset

Compared with the large number of depth-based hand pose datasets [30, 27, 4, 25], existing RGB-based datasets [32, 35, 17, 23, 18, 29] have a very limited number of frames and subjects because obtaining accurate 3D annotation from RGB images is difficult. Stereo Tracking Benchmark (STB) [32] is one of the most commonly used datasets to report single RGB-based hand pose estimation, but the number of frames (18K) and subjects (1) is very limited. Rendered Hand Pose (RHP) Dataset [35] contains 44K images of two isolated hands, the images are synthesized by animating 3D human models and have a large domain gap from the real one. Mueller et al. [17] tried to reduce the domain gap of synthesized data by using GAN, however, GAN-based methods might introduce unnatural artifacts to the data.

For RGB-based datasets with a large number of images, Simon et al. [22] proposed a hand dataset (680K) using the CMU Panoptic studio, which consists of humans performing different tasks and interacting with each other. Moon et al. [15] captured a large-scale (2.6M) two hands interaction dataset with different lights and camera angles. However, these two methods both utilized hundreds of depth cameras which are almost not possible to reproduce. Also, a large number of cameras can make synchronizations very difficult, thus not suitable for capturing with a fast shutter time, which is required for fast motion like piano playing.

Recently, Zimmermann et al. [36] collected a single hand pose and mesh dataset named FreiHand with only 8 RGB cameras, and they utilized a semi-automatic method for annotation. Although manual verification was still required which might involve human error, their work showed a direction for capturing using a relatively simple setup.

The above-mentioned RGB-based datasets focus on obtaining general hand poses instead of a specified application. In terms of piano, there is a piano fingering dataset published by Nakamura et al. [19] but no hand posture information is included. The only public dataset (to the best of our knowledge) which includes piano hand motions is the previously mentioned CMU Panoptic HandDB [22], however, the number of piano images is very limited.

### 2.2. RGB-based 3D Hand Pose Estimation

Many works [35, 5, 31, 17, 33, 13] are trying to estimate hand poses from RGB image sequences. Zimmermann and Brox [35] are the first who tried to train a CNN-based model that estimates 3D joint position directly from an RGB image. Ge et al [5] directly regressed a hand mesh from RGB images using a GraphCNN [1], but the training requires special hand meshes as ground truth which is difficult to obtain.

For real-time estimation, Mediapipe [14, 31] from Google provided a very easy-to-use API to access which enabled on-device real-time hand pose estimation, however, their estimation result is 2.5D instead of 3D position, which cannot calculate the 3D joint angle. Mueller et al. [17] employed CycleGAN [34] for bridging the domain gap to generate realistic synthetic data. Zhou et al. [33] trained a lightweight inverse-kinematics network that enhances the regression of angle-based hand poses. Liu et al. [13] ex-

tended the work of Moon et al. [15] and achieved a light network that better extracts global features from a single image.

These works show the maturity of the current hand pose estimation technologies. However, as mentioned before, all these networks require a precise and robust dataset to achieve good performance.

### 2.3. Piano-related Hand Pose Research

There are quite a few researches [16, 3, 8, 26, 10] dealing with hand postures that are related to piano playing. Moryossef et al. [16] extracted fingering information from public videos and MIDI files. Furuya et al. [3] used a motion capture system to record the 3D hand kinematics for teaching musically untrained individuals to practice a simple and short melody. Johnson et al. [8] are the first to detect hand postures for piano instructions, but they utilized a depth camera which was less common and cannot be applied to recorded or online videos. Moreover, the low resolution and speed of a depth camera can hardly ensure high-precision piano instruction. Takahashi et al. [26] utilized a soft exoskeleton glove to support a novice to play a musical excerpt, of which a coach hand-pose dataset is required for instruction. Reversely, Liang et al. [12] made use of real-time hand pose estimation to develop a virtual piano application using a depth camera. All these researches show the necessity of a precise professional pianist hand pose dataset, which also indicates future applications for our work.

## 3. Methodology

In this section, the whole process from data capture to post-processing is introduced to show a clear procedure for collecting precise bare-hand data.

As mentioned before, a marker-based MoCap can obtain high-precision 3D hand poses and is relatively easy to set up, while CV-based methods require a precise dataset of bare-hand images. Therefore, for precise ground truth, we employ a well-calibrated commercial MoCap system to capture accurate 3D hand pose data and aligned marked-hand images. After that, the markers on the hands are removed by our proposed marker-removal network, which results in a bare-hand image with its 3D hand pose ground truth. Finally, the data is further processed with automatic interpolation, and a solver using forward kinematics is built to re-target the marker positions to real joint positions, no manual annotation is required in the whole process.

### 3.1. Data Capturing Environment

*PianoHand2.5M* is captured in two different locations, a green-back laboratory environment with an 88-key electric piano (Kawai ES-110) and a studio with an 88-key grand piano (Kawai GE-20). Both are equipped with an Optitrack[1] MoCap system, where eleven Optitrack Prime 13W monochrome IR cameras and one Optitrack Prime Color FS RGB camera are well-calibrated and synchronized for capturing. All cameras capture at 240 frames-per-second (FPS) while the exposure time is set to 4 ms, and the image resolution is $1920 \times 1080$ (1080p). The placement of the camera is slightly different for the 2 conditions due to the difference in the shape of the piano (can be seen in the right figures in Fig.2), but the RGB reference camera is fixed to the top-middle of the piano. The final mean re-projection error of both setups is similar, ranging from 0.19 to 0.28 mm (approx. pixel root mean square error is 0.24-0.35). The markers used on the hands are the Optitrack hemisphere 4 mm facial reflective markers, which are very tiny and thus relatively easy to remove. For each hand, twenty markers are placed on each joint and fingertip and three additional markers are placed close to the wrist as a triangle to obtain the wrist rotation (as shown in Fig.2). In total, 46 markers are placed on both hands of the pianists. Besides the marker information, a MIDI of each play is also recorded and synchronized, for the grand piano scene, we also record a real-time depth of each key (a max. 10 mm keystroke) by using an IR-sensor embedded behind each key of the piano. The whole data-capturing procedure is approved by the local IRB department.

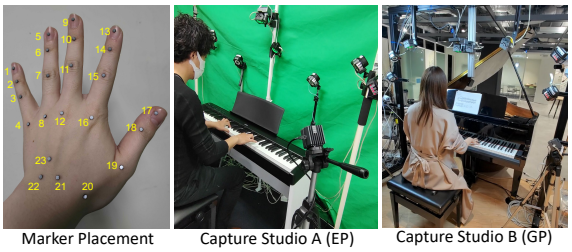


Figure 2. (Left) Hand with 23 markers. (Right and Middle) The two studios for capturing: One with an electric piano (EP), and another with a grand piano (GP).

### 3.2. Generation of Data

For training a convolutional neural network, the captured raw image data which have markers on the hands may affect the training because the markers are providing extra features. Three methods for translating marked-hand to bare-hand are introduced to be compared with raw data (markers unremoved). Additionally, the 3D tracking information of the markers also needs to be pre-processed to serve as ground truth for the training. Note that, in all descriptions about coordinates here, the $x$-axis indicates horizontal movements (positive for the right direction facing the piano), $y$-axis indicates vertical movements (positive for the up direction), and $z$-axis stands for the depth. The origin $O$ is set to the center of the 88-key edge (between the key E4 and F4), the same level as the key surface.

[1] https://optitrack.com/

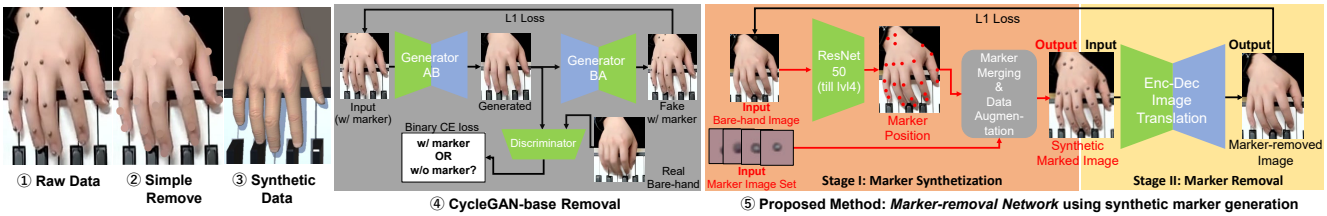


Figure 3. Conventional methods of generating training data (2-3), GAN-based method (4), and the proposed MR-Net (5).

#### 3.2.1 Tracking Data Pre-processing

**Marker labeling and interpolation:** Firstly, the tracked markers are labeled automatically using a greedy matching algorithm, where manual labeling might be performed when the tracking of a marker is lost, however, thanks to the well-set-up camera system, this operation is only required for less than 0.1% of the total data. Next, an automatic cubic spline interpolation is employed to fix the occasional tracking loss with a maximum gap set to 10 frames. If there are still imperfections after this processing, the data are abandoned for a retake (very seldom which only happens twice in the whole capturing session). Finally, square bounding boxes of the hands are automatically annotated based on the center of all joints.

**Hand Solver:** It is obvious that there is an offset between the marker and the attached joint. Different from body pose, where kinematic motions might result in a complicated calculation, the offset of markers and the real joints is a constant value that is close to the thickness of the hand. Before the recording, we ask all the subjects to place their hand tightly on a flat horizontal table of which the height is already measured by the MoCap system, thus the offset of the joints and markers is equal to half of the distance between the markers and the table surface. In the case of the thumb, a tiny rotation in the roll-axis is manually performed to the marker positions to fix the natural thumb rotation. Finally, the wrist rotation can be easily calculated from the plane of the 3 markers placed on the center of the wrist while the wrist position is considered to be the middle of point 21 and 22 in Fig.2 (Left).

What to note here is, the whole data recording tasks last over 30 hours, but the time of manual supervision required for pre-processing and annotation is less than 4 hours (excluding the time for checking the data), which is evidently reduced compared to conventional works.

#### 3.2.2 Synthetic Data (Baseline 3)

For comparison and also as a baseline, a synthetic dataset is created from the captured data. Technically, synthetic hand data is the simplest way to "remove the marker" by reconstructing a hand model from hand pose data. We developed a virtual piano environment and a hand simulator using forward kinematics. For each finger, we assume that $p_0 - p_4$ represent the 3D position of the fingertip, DIP, PIP, MCP, and the center of wrist, while a bone vector $\mathbf{v_i} = p_{i-1} - p_i$. The Euler angle $\alpha_1, \alpha_2, \alpha_{3v}, \alpha_{3h}$ representing the rotation of DIP, PIP, vertical MCP, and horizontal MCP, respectively, can be calculated:

$$\alpha_i = \begin{cases} arccos(\frac{\mathbf{v_i} \cdot \mathbf{v_{i+1}}}{\|\mathbf{v_i}\|\|\mathbf{v_{i+1}}\|}|_{x=0}), & i = 1, 2, 3v \\ arccos(\frac{\mathbf{v_i} \cdot \mathbf{v_{i+1}}}{\|\mathbf{v_i}\|\|\mathbf{v_{i+1}}\|}|_{y=0}), & i = 3h \end{cases} \quad (1)$$

Since the DIP and PIP (IP and MCP for thumb) have only one degree of freedom (DOF), the rotation angle of a joint can be easily calculated from the projection angle between the previous bone vector $v_i$ and the latter bone vector $v_{i+1}$ on the yaw plane (y-z plane). On the other hand, the MCP (CMC for thumb) has two DOF which are represented in two Euler angles. Based on these angles and the hand size of the subjects, a synthetic hand image close to the raw data can be generated as shown in the second image of Fig.3.

#### 3.2.3 Cycle-GAN (Baseline 4)

The GANerated Hands [17] used two-stream Cycle-GAN [34] to generate real hand images from simulated hand images, which still has to overcome a relatively huge domain gap between synthetic hands and real hands. Nevertheless, removing tiny markers on the hand is a much more straightforward task, so we first utilize a standard Cycle-GAN as an initial baseline for realistic marker-removal. A ResNet50 [6] is used as the backbone for the generator and the discriminator while the cycle-consistency loss uses L1 Loss, as shown in the middle of Fig.3. To train the network, bare-hand data also needs to be collected from the subjects, the details will be mentioned in Section 4.

#### 3.2.4 Marker-removal Network (Proposed Method)

The above GAN-based method does show effects in removing markers from the hands, however, since the generated bare hand image is not compared to a real image using a pixel-perfect L1 loss, it sometimes adds unnatural artifacts to the original data, as shown in Fig.4, the keyboards and the clothes of the subjects are distorted because they are not taken into account by the discriminator. It is ideal if a pixel-to-pixel translation can be performed, but a pair of hand im-

| Data Split | Task | Subjects | Keystroke | No. of Frame | | | |
|---|---|---|---|---|---|---|---|
| | | | | R | L | B | Sum |
| Train (EP) | R,L,B1-B6 | 6 | no | 150K | 110K | 400K | 660K |
| Train (GP) | R,L,B1-B11 | 10 | yes | 263K | 217K | 834K | 1314K |
| Val (EP+GP) | B1-B6(EP),B1-B11(GP) | 1+1 | no | 0 | 0 | 155K | 155K |
| Test (EP) | Free Play (R,L,B) | 1 | no | 23K | 17K | 65K | 105K |
| Test (GP) | Free Play (R,L,B) | 2 | yes | 50K | 36K | 166K | 252K |
| Total | / | 21 | / | 513K | 403K | 1696K | **2486K** |

Table 1. The details of the PianoHand2.5M Dataset. EP: data taken in the electric piano studio. GP: data taken in the grand piano studio.

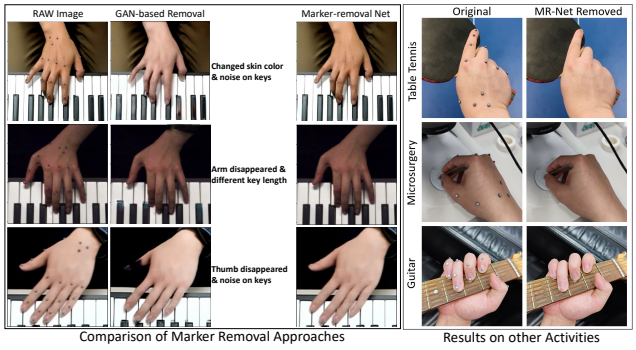


Figure 4. (Left) Some failures cases of GAN-based marker removal compared with the proposed marker-removal net. (Right) Examples of MR-Net on the other hand activities.

ages w/ and w/o markers is required to realize this, which is not practical. Therefore, we introduce the Marker-removal network (MR-Net) which generates synthetic markers to bare-hand images to be used as training data for the generator, where the output bare-hand image can be directly compared with the original input using L1 Loss.

The MR-Net consists of two stages: marker synthesization and marker-removal. The marker synthesization stage passes an input bare-hand image $\boldsymbol{B}$ through a CNN backbone (ResNet50 till level 4 in this paper) to extract visual features for marker estimation, which regresses a 2D markers position vector $\boldsymbol{p}$ (size: 2x21). The backbone here is pretrained using some good results by the CycleGAN-based method which are manually picked out. Simultaneously, 21 marker images $\boldsymbol{MI}$ are randomly chosen from the marker image set and alpha blended to the corresponding positions. For data augmentation, a maximum 5% noise is randomly added to the $\boldsymbol{p}$ and the size of the $\boldsymbol{M}$, meanwhile, Gaussian blur and brightness adjustment are performed to the marker image $\boldsymbol{M}$. Finally, a hand image with synthetic markers $\boldsymbol{S}$ is generated. For the marker removal, a U-Net [21] like encoder-decoder network using the same CNN backbone (ResNet50) is employed for image translation. The output $\boldsymbol{S}$ from stage I is fed to this network to extract these marker features where the convolutional layer shares the weights of the previous marker estimation. These markers are then removed in the deconvolutional phase. Skip connections are added after each convolutional block to better maintain the rest of the image. Finally, a marker-removed bare hand image $\boldsymbol{R}$ is output and we use an L1 loss function: $\mathcal{L}_{\mathbf{1}} = \sum_{i=1}^{n} |\boldsymbol{R}_i - \boldsymbol{B}_i|/n$ to realize a pixel-comparison.

One might argue that these synthetic markers still create domain gaps which contradicts our major motivation, but a pilot test suggests that domain gaps of synthetic markers are tiny and have less impact on the marker-removal. This is also proved by the results of the later experiment.

## 4. PianoHand Dataset

The contents of the dataset are shown in Table 1.

**Subjects:** The whole data are captured from 21 unique experienced pianists (13 females, 9 males, Avg.age=27.75, SD=7.44). Most subjects start learning piano at the age of 4, and the average experience is 22.7 years. Among them, two subjects are professional pianists who played in international concerts, four work as piano instructors or related professions, and the remaining are all piano students from local art colleges/universities. The subjects are divided into two groups, 2 professionals and 6 students are asked to perform in the EP studio and the other 13 students are captured in the GP studio.

**Tasks:** The whole task is designed based on several considerations and suggestions from experienced pianists. Overall, there are two types of tasks to be performed by the subjects: unimanual (single hand) performance using either right (R) or left (L) hand, and bimanual (both hand) performance (B). The unimanual tasks are conducted in the same way for both groups of subjects, in which 10 phrases are played with the right hand and 8 phrases are played with the left hand. These phrases include fundamental patterns of hand movements in piano playing (e.g. scale, arpeggio), some of which involve complex changes in the hand posture with self-occlusions of the fingers. Each subject is asked to repeat each phrase 5 times, which results in 50 phrases for the right and 40 phrases for the left hand. For the bimanual tasks, particular excerpts of 11 pieces of music are chosen to be played. All the subjects are told to play each excerpt repeatedly for approx. one minute. Namely, about 11 minutes of data (about 80k frames under 120 fps) are recorded for each subject in the bimanual task. More details about the music, the notes, and the reason for the choice can be found in the supplementary document. Also, please note that two subjects (one from the EP and one from the GP) are used as

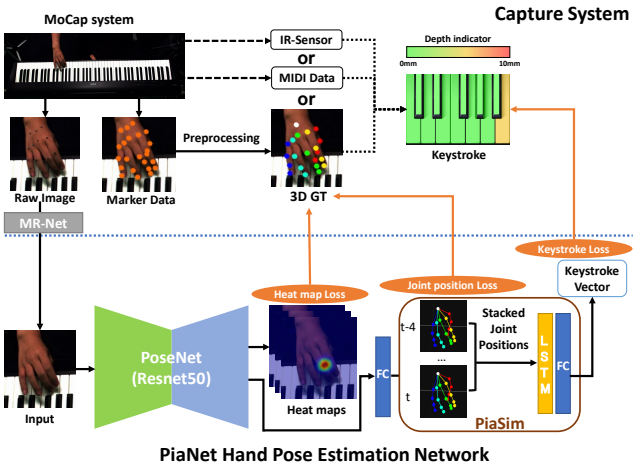


Figure 5. An overview of the data capture and the proposed PiaNet.

pilot testers, where the unimanual tasks are not conducted. Their data are therefore used for validation which is mentioned later. To evaluate the generality, three subjects (one from EP, two from GP) are asked to play random music freely for a total time of about 10 minutes (approx. 1.5, 2, and 6.5 min for left, right, and bimanual, respectively) for each of them, of which the data are used as test data.

**Data splits:** Finally, the collected data are downsampled to 120 fps to reduce redundant images and a total number of 2486K image data are collected. The whole dataset is divided into training, validation, and test sets for performing a between-subject evaluation. The training set consists of data from 16 subjects (6 EP, 10 GP), with a total number of 2.0M. The validation set, as mentioned above, is using the two pilot testers' data, thus no R and L data are included. The test data consists of the free play of three subjects, as mentioned above. The detailed contents of the dataset can be seen in Table.1. For all data, the ground truth bounding box information is used to crop the hands. In addition, the real-time keystroke data of the piano as well as the MIDI are also recorded in the GP scenario.

**Data Augmentation:** Common augmentations are performed to the annotated data, such as resizing, rotation, color, and brightness adjustments, and adding motion blur by Gaussian filter. Unlike most data augmentation, flipping the hand is not considered because we assume that right-hand (high part) and left-hand (bass part) motions are different in piano playing.

**Other data:** For comparative purposes, the same number of data is also generated using the synthetic method, the simple-removal method, and the GAN-based method. Furthermore, the GAN-based method requires similar barehand images to train the discriminator. Therefore, we ask the subjects to play all the tasks again barehanded for only once, and a total number of 530K barehand images are collected (which is not counted for the 2.5M data).

# 5. PiaNet Baseline

To show the practical effect and to serve as a baseline for future works, we propose the PiaNet, which takes a single RGB image $\mathbf{I}$ as an input and outputs a real-time 3D hand joint position $\mathbf{P}$, Fig.5 shows an overall architecture of PiaNet. The input $\mathbf{I}$ is the cropped hand image obtained by the ground truth bounding boxes which are resized to $256 \times 256$. The PiaNet itself does not provide a hand detection and bounding box because the background is mostly fixed with only hands moving in the piano scenario, simple background subtraction are sufficient for detection.

## 5.1. 3D Hand Key Point estimation

For 3D hand key point estimation, the PiaNet uses an encoder-decoder PoseNet architecture which is similar to the networks by Zimmerman et al. [35] and Ge et al. [5] with a ResNet50 [6] backbone. Given the features $\mathbf{F}$ which is extracted from $\mathbf{I}$ by the encoder, 3D heat maps $\mathbf{H}_k$ are predicted containing the probability information of the $k$-th joint position in 3D. The final joint position coordinate $\mathbf{P}_k$ is obtained as the integration of all locations $\mathbf{p}$ in the domain $\Omega$, weighted by their probabilities:

$$\mathbf{P}_k = \sum_{p_z=1}^{D} \sum_{p_y=1}^{H} \sum_{p_x=1}^{W} \mathbf{p} \cdot \tilde{\mathbf{H}}_k(\mathbf{p})$$

Here, the $\tilde{\mathbf{H}}_k$ is the normalized heat map $\mathbf{H}_k$. The $D, H, W$ indicates the resolution on depth, height, and width of the heat maps, respectively. To train this PoseNet module, L2 loss is used for the heat maps.

## 5.2. PiaSim Module

In piano performance, the timing of sound (when a key is pressed) is considered to be the most essential factor. Accordingly, the fingertip position (also the PIP rotation) needs to be accurate. To enhance the training to be more specific towards piano hand motions, a PiaSim network is developed to output keystrokes based on an input hand sequence. The network consists of a long-short term memory (LSTM) [7] layer to extract time series motions and a fully-connected layer to reproduce the keystroke. The input stacks the last 5 frames of hand postures, of which the 3D positions are relative to the piano (same as the dataset), resulting in an input size of $5 \times 21 \times 3$. The output is a 1D array with a size $n$ = 12 (for the 12 keys) showing the key-depth $\mathbf{k}_i$ of the $i$-th key (ranged from 0-10 mm, normalized to 0-1). Since not all the keys are always pressed, using an L2 loss is not suitable for such a sparse vector. We developed a keystroke loss function with the ground truth is $\mathbf{k}^*$, and the number of values greater than zero in $\mathbf{k}^*$ is $\mathbf{N}_{(k^*>0)}$:

$$\mathcal{L}_{\mathbf{key}} = \sum_{i=1}^{n} \|\mathbf{k}_i - \mathbf{k}_i^*\|_2 / \mathbf{N}_{(k^*>0)} \quad (2)$$

The output size is set to 12, so an octave (a note and the same higher note is played, for example, C4 and C5) is

| Training Set | Result on Validation Set (Same task) | | |
|---|---|---|---|
| | MPJPE(mm) | MPJAE(°) | PCK(%) |
| Raw | 14.44 | 9.9 | 66.1 |
| Simple R. | 12.04 | 8.9 | 74.1 |
| Synthetic | 11.12 | 7.9 | 75.2 |
| GAN-based | 9.98 | 7.8 | 78.8 |
| **MR-Net** | **9.22** | **7.3** | **81.6** |
| | Result on Test Set (Different task) | | |
| | MPJPE(mm) | MPJAE(°) | PCK(%) |
| Raw | 22.40 | 13.5 | 54.3 |
| Simple R. | 21.38 | 13.0 | 57.2 |
| Synthetic | 19.11 | 12.1 | 60.2 |
| GAN-based | 10.97 | 8.5 | 76.7 |
| **MR-Net** | **9.95** | **7.9** | **80.3** |

Table 2. Result of the between-datasets comparison.

considered to be pressing the same key in the prediction. For training, given that the keystroke information is not obtained for the EG group, we developed a keystroke simulator to simulate keystroke information from either MIDI or ground truth hand poses. An ablation study comparing the networks using different keystroke ground truth is conducted in section 6.2. Finally, the overall loss function for the training procedure is as follows, where $\lambda_1$, $\lambda_2$and $\lambda_3$ are the weights for the joint position loss, heat map loss, and keystroke loss, respectively:

$$\mathcal{L} = \lambda_1 \|\mathbf{P} - \mathbf{P}^*\|_2 + \lambda_2 \|\mathbf{H} - \mathbf{H}^*\|_2 + \lambda_3 \mathcal{L}_{\mathbf{key}} \quad (3)$$

# 6. Quantitative Experiment

In this experiment, we aim to evaluate our data generation method and compare our results with other state-of-the-art methods on a variety of publicly available datasets. To perform a fair comparison, we show the results of the three most common metrics: the mean per joint position error (MPJPE), the mean per joint angle error (MPJAE), and the Percentage of Correct Keypoints (PCK).

## 6.1. Between-dataset Comparison

Firstly, to show the effect of our marker-removal network over synthetic data and GAN-based methods, a comparative study between the three methods is conducted. A model trained with the raw data (marker unremoved) and a simple-remover (Simple R.) that replaces markers with skin-colored circles is also included to serve as a baseline. All datasets are trained using PiaNet with the PiaSim module of which the keystroke ground truth is generated from the hand pose ground truth.

Table 2 shows the results under three metrics. For the validation set (same task as the training set), besides the raw-data trained model which falls behind, tiny differences in performance are shown between the other four generation methods. The situation changes for the test set evaluation (the task is different), models trained with simple-removal and synthetic data show an obvious drop back in all three metrics while the GAN-based and MR-Net-based methods still remain a relatively high precision. Overall, the proposed MR-Net performs the best in both evaluations.

| Ablation | MPJPE | MPJAE | PCK |
|---|---|---|---|
| w/o PiaSim (Direct R.) | 10.72 | 8.6 | 76.1 |
| PiaSim + pose | **9.95** | 7.6 | 80.2 |
| PiaSim + MIDI | 10.09 | 7.9 | 79.7 |
| PiaSim + Keystroke | **9.95** | **7.5** | **80.3** |

Table 3. Result of the ablation study.

## 6.2. Ablation Study

We carry out an ablation study to find out whether the PiaSim module improves the performance of PiaNet. Four ablations of not using PiaSim (direct regression), a PiaSim trained with a pose-simulated keystroke (the one used in previous experiments), a PiaSim trained with MIDI-simulated keystroke, and a PiaSim trained with raw keystroke data obtained from the IR-sensors are evaluated. All conditions are trained and tested only on the GP data, which has aligned keystroke data. The result is shown in Table 3, which suggests that all three methods using PiaSim outperform w/o PiaSim condition while using directly obtained keystroke data shows the best accuracy. To our surprise, the MIDI-based solution falls behind the pose-based method. Since raw keystroke data cannot be easily captured, using a pose-based keystroke with close accuracy becomes optimal.

## 6.3. Between-method Comparison

Finally, to show the robustness of our dataset and network, we compared with the FreiHand [36], and the InterHand [15]. and Zhou et al.'s method [33], which are SOTA hand pose estimation methods with pre-trained weight released. For a fair comparison, we evaluate the different models on the piano sequences of the CMU Panoptic Hand dataset [22], only top-view images where the hands can be clearly seen are chosen, so approx. 10k images from *161029_piano1, 161029_piano2, 161029_piano3* are used for testing.

**Results:** Table 4 shows the results of 2D and 3D positions, and 3D angle error, since the finger angles are considered to be more important in piano, we also show the detailed error of each specific joint. For the 2D and 3D positions and MPJAE metrics, the proposed PiaNets both outperform the other three baselines. It is interesting to notice that for the MCP error, the PiaNet (MR-Net) (MCPE = 7.24) falls behind the PiaNet (Synthetic) and Zhou et al.'s method, but for the PIP error, the proposed method greatly overgoes the others which might indicate that the network is trying to focus on the more important DIP angle.

| Method | 2D Position | | 3D Position | | 3D Angle Error | | | |
|---|---|---|---|---|---|---|---|---|
| | MAE | PCK | MPJPE | PCK | MCP | PIP | DIP | MPJAE |
| FreiHand [36] | 8.4 | 89.1 | 24.47 | 61.6 | 8.09 | 12.34 | 13.33 | 11.58 |
| InterHand [15] | 7.3 | 94.1 | 21.32 | 64.4 | 7.41 | 11.34 | 12.00 | 10.25 |
| Zhou et al. [33] | 6.8 | 95.0 | 16.11 | 67.3 | 7.02 | 8.12 | 12.56 | 9.22 |
| PiaNet (Synthetic) | 7.2 | 94.2 | 14.72 | 70.5 | **6.95** | 8.39 | 11.82 | 9.05 |
| **PiaNet (MR-Net)** | **5.6** | **98.1** | **12.49** | **73.7** | 7.04 | **7.12** | **7.66** | **7.30** |

Table 4. Quantitative results of the compared models tested on the piano sequences from the CMU Panoptic Hands dataset.

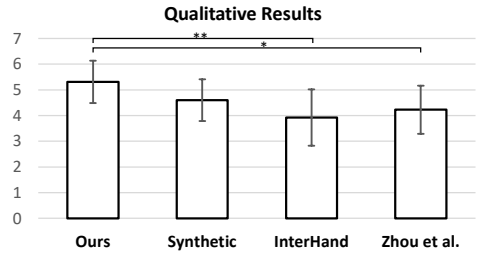


Figure 6. Results of the qualitative experiment. Brackets indicate significant differences. ($*p < 0.05, **p < 0.01$)

## 7. Qualitative Experiment

In the case of the piano, it is important whether the estimated hand poses naturally play the corresponding notes, which is difficult to tell from the previous quantitative experiments. Therefore, to show the practical effect, we collect 8 piano instruction videos to perform a qualitative blind study on the estimated results of the four best models in the between-method comparison (Due to the limited number of subjects, we abandoned the FreiHand [36] model to reduce condition). Eight experienced pianists are invited to take the ranking test, where the shuffled estimated results and the videos with the correct notes are given to them. The subjects can freely rotate the screen to view the 3D hand from different angles (details are explained in the supplementary document). A 7-point-Likert-scale questionnaire is given to subjects to choose how natural each hand pose is.

**Result:** The results are shown in the chart in Fig.6. A one-way ANOVA test [24] is performed to analyze the significant difference between each condition. The result ($F_{3,36} = 4.3597, p = 0.0102$) suggests significant differences within the condition, thus a posthoc Tukey HSD test [28] is applied to pinpoint which condition exhibits significant difference. As a result, there is a significant difference ($p < 0.01$) between Ours (Avg=5.31, SD= 0.82) and the InterHand (Avg=3.93, SD=1.01), and a significant difference ($p < 0.05$) between Ours and Zhou et al.'s method (Avg=4.23, SD=0.94). Even though no analytical significance was found, the PiaNet trained with synthetic data shows a higher average score (Avg=4.60, SD=0.81) than Zhou et al., which is opposite to the quantitative results.

## 8. Discussions & Limitations

Three quantitative experiments and a qualitative study are conducted to exemplify the effectiveness of the proposed dataset and network. The marker-removal network outperforms other data generation methods such as synthetic or GAN-based data which can be explained by a smaller domain gap. When compared with other SOTA estimations on piano-playing hands in a public dataset, our PiaNet trained with the PianoHand dataset shows the highest PCK with the smallest angle error, especially for the DIP angle of each finger, which is one of the major factors in piano performance. Lastly, the qualitative study of 10 experienced pianists observing the estimated hand pose results shows the effect of our dataset on practical purposes. Despite the results being achieved, several limitations of this work are also concluded:

- One difficult scenario for our method is when we need to collect data from a person with many features on the hand (such as heavy hairs or moles), the marker-removal might mistakenly remove these features as well.
- Although data from both hands are measured, the proposed PiaNet only focuses on single-hand estimation. Since hands interact in piano playing, including both hands as a bootstrapping might improve the performance, which needs to be studied in the future.
- Even though the proposed method outperforms other baselines in estimating hand poses during piano playing, the quantitative accuracy is still far from perfect, which might be the bottleneck of single RGB image-based estimation. Integration of multiple-camera-view or keystroke data as input can be considered in the future.

## 9. Conclusion

We propose a novel marker-removal approach for collecting bare-hand data including a precise ground truth together with the first large-scale pianist 3D hand dataset, PianoHand2.5M. This method opens the possibility of creating precise and realistic hand pose datasets without a heavy workload for annotation. Also, a PiaNet for piano hand pose estimation is introduced for successors to use as a baseline, which also enables pianists to easily set up a motion capture for home use (e.g. practicing and instruction). We hope the proposed procedure of making a dataset for a specific application inspires future works to create vast datasets in various scenarios. We believe this can contribute to the vision community and also benefit motion capture manufacturers to embed our system to realize direct "marker-less" output.

## References


[1] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.

[2] Shinichi Furuya, Martha Flanders, and John F Soechting. Hand kinematics of piano playing. *Journal of neurophysiology*, 106(6):2849–2864, 2011.

[3] S Furuya, A Nakamura, and N Nagata. Acquisition of individuated finger movements through musical practice. *Neuroscience*, 275:444–454, 2014.

[4] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[5] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10833–10842, 2019.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] David Johnson, Daniela Damian, and George Tzanetakis. Detecting hand posture in piano playing using depth data. *Computer Music Journal*, 43(1):59–78, 2020.

[9] André Jonas, Erwin Wu, Shio Miyafuji, and Hideki Koike. Precise hand pose data collection for piano players. In *Proceedings of the Workshop on Human Augmentation for Skill Acquisition and Skill Transfer*, page No.11, 2021.

[10] Nozomi Kugimoto, Rui Miyazono, Kosuke Omori, Takeshi Fujimura, Shinichi Furuya, Haruhiro Katayose, Hiroyoshi Miwa, and Noriko Nagata. Cg animation for piano performance. In *SIGGRAPH '09: Posters*, SIGGRAPH '09, New York, NY, USA, 2009. Association for Computing Machinery.

[11] Samuli Laine, Jaakko Lehtinen, and Timo Aila. Self-supervised deep image denoising. *CoRR*, abs/1901.10277, 2019.

[12] Hui Liang, Jin Wang, Qian Sun, Yong-Jin Liu, Junsong Yuan, Jun Luo, and Ying He. Barehanded music: Real-time hand interaction for virtual piano. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '16, page 87–94, New York, NY, USA, 2016. Association for Computing Machinery.

[13] Yang Liu, Jie Jiang, Jiahao Sun, and Xianghan Wang. Internet+: A light network for hand pose estimation. *Sensors*, 21(20), 2021.

[14] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines, 2019.

[15] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision (ECCV)*, 2020.

[16] Amit Moryossef, Yanai Elazar, and Yoav Goldberg. At your fingertips: Automatic piano fingering detection. 2019.

[17] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1154–1163, 2017.

[19] Eita Nakamura, Yasuyuki Saito, and Kazuyoshi Yoshii. Statistical learning and estimation of piano fingering. *Information Sciences*, 517:68–85, 2020.

[20] Jakobine Paulig, Hans-Christian Jabusch, Michael Großbach, Laurent Boullet, and Eckart Altenmüller. Sensory trick phenomenon improves motor control in pianists with dystonia: prognostic value of glove-effect. *Frontiers in Psychology*, 5:1012, 2014.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[22] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[23] Srinath Sridhar, Franziska Mueller, Michael Zollhoefer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[24] Lars Sthle and Svante Wold. Analysis of variance (anova). *Chemometrics and Intelligent Laboratory Systems*, 6(4):259–272, 1989.

[25] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 824–832, 2015.

[26] Nobuhiro Takahashi, Shinichi Furuya, and Hideki Koike. Soft exoskeleton glove with human anatomical architecture: Production of dexterous finger movements and skillful piano performance. *IEEE Transactions on Haptics*, 13(4):679–690, 2020.

[27] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):1–10, 2014.

[28] John W Tukey. Comparing individual means in the analysis of variance. *Biometrics*, pages 99–114, 1949.

[29] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 118(2):172–193, 2016.

[30] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4866–4874, 2017.

[31] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking, 2020.

[32] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016.

[33] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[35] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[36] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russel, Argus Max, and Brox Thomas. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.