IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Tracking of Flexible Brush Tip on Real Canvas: Silhouette-based and Deep Ensemble Network-Based Approaches

JOOLEKHA BIBI JOOLEE[1], AHSAN RAZA[1], MUHAMMAD ABDULLAH[1] AND SEOKHEE JEON[1].

[1]Department of Computer Science and Engineering, Kyung Hee University Global Campus, Yongin, Republic of Korea.(e-mail: julekhajulie@gmail.com)

Corresponding author: Seokhee Jeon (e-mail: jeon@khu.ac.kr).

This research is supported by Ministry of Culture, Sports and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research Development Program 2019.

ABSTRACT Digital painting is a process of creating a digital artwork using modern human-computer interaction technologies. One of the core enabling technologies is the real-time tracking of user's strokes, which is generally supplied by a digital tablet with a stylus. While the digital tablet technology provides highly accurate tracking, the drawing should be done with a rigid stylus on a plastic surface. This sometimes destroys the realism of drawing, such as interaction with the digital tablet cannot provide the feedback of subtle texture, friction of the paper/fabric canvas and tension of soft painting brush. This becomes particularly problematic for traditional painting artists who are trained with and prefer real painting brush and paper/fabric canvas. Thus, the aim of this work is to present an alternative solution where the user's strokes can be tracked even when the actual brush and canvas are used. To this end, we proposed two approaches for digitally tracking the tip of flexible bristles of a soft brush, so that the painting can be created digitally on a computer. The first approach captures the silhouette of deforming bristles using a pair of well-aligned infra-red (IR) cameras, which extracts the tip from the silhouette, and reconstructs the 2D position of the tip. The second approach predicts the brush tip position through a deep ensemble network-based approach where the relationship between the brush tip position and brush handle pose are trained with our novel model comprising of Long-Short Term Memory Autoencoder and 1-D Convolutional Neural Network. The trained model is used to predict the brush tip position in realtime. Both approaches extensively evaluated through multiple tests. Furthermore, our model outperforms the state-of-the-art models.

INDEX TERMS Silhouette based tracking, Deep ensemble network, Long-Short Term Memory Autoencoder, 1-D Convolutional Neural Network.

I. INTRODUCTION

In recent years, the digital painting market has grown a lot in order to meet the modern art society's demand. Digital painting is accomplished by producing a digitized painting artwork using modern human-computer interaction (HCI) techniques on a computer. The interaction is usually done by a stylus and a digital tablet [1]–[4]. The main job for this digital tablet is to track the artist's strokes, and the state-of-the-art tablet technology allows very accurate tracking and capturing of the strokes.

Digital painting is less common in traditional painting. It is partially due to that many traditional painting artists who are trained with and prefer direct handling of a brush on a real canvas, over rigid stylus pen on a slippery tablet, which often destroys the realism of drawing, e.g., the feedback of subtle texture of the canvas. There are many attempts to provide the feeling of an actual brush and a canvas, e.g., brush-type stylus and matt tablet surface, subtle tension and friction feedback of which the artists make use for their performance, is still different from their real counterpart [5]–[7].

In order to overcome the aforementioned issues, the focus of the paper is to provide an alternative solution for the traditional painting artists where we produce the artwork digitally using the real brush and real canvas, which reflects the augmented reality-based interactive drawing. First, our system tries to get rid of tablets and stylus from the digital

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3004173, IEEE Access

painting. Instead, our solution allows the artists to draw actual artwork using the real brush and real canvas while still digitally storing or recreating the artwork by estimating the artist's stroking accurately in real-time. This indicates the need of new means of tracking the tip of the brush in the canvas space, which is the main aim of the present paper.

To achieve this, we proposed two approaches. First approach directly captures the silhouette of deforming bristles of a brush through a pair of well-aligned infra-red (IR) cameras. Then, the system extracts the tip out of the silhouette and reconstructs the 2D position of it in the canvas space. This is simple but very effective approach and, to our knowledge, the first system that tracks the tip of the deforming brush bristles in real-time through silhouette.

Since the first approach still needs a specially aligned frame and cameras and has shortcoming in usability, i.e., user's hand may occlude the brush, we introduce our second approach. Our second approach estimates the brush tip position based on a novel deep learning network. The relationship between the pose of the brush handle and the brush tip position is trained with our specially designed deep ensemble network using true data, and later the brush tip position is predicted using only the pose of the brush handle. In other words, once the deep ensemble network is trained, the silhouette-based tracking is not needed at all to track the brush tip, which makes the system simple, first and easy to use. To the best of our knowledge, this is also the first attempt to employ the deep learning-based approach for modeling the brush tip position.

Our approach is also advantageous over just capturing and storing the final product of the drawing. Since the approach can not only store the final product, but also capture the sequence of the trajectories of the drawing in real time, it can be used for other scenarios, e.g., sensorimotor skill training where the stroking sequence and trajectories of a master can be stored and used later for the training of students, and augmented reality drawing where the artwork is digitally visualized on real canvas in realtime even there is no actual drawing is made.

The main contributions of this work is summarized as follows.

- In order to track the brush tip position, a silhouette-based brush tip tracking approach is proposed, which captures the silhouette of deforming bristles of a brush through a pair of well-aligned infra-red (IR) cameras.
- As another alternative, a deep ensemble network comprising of 1-D CNN and LSTM Autoencoder is designed for predicting the brush tip position, which takes minimum information and accurately predict the tip position. The proposed 1-D CNN captures the spatial information whereas the LSTM Autoencoder obtains temporal features. The ensemble network is employed to avoid overfitting and overconfidence.
- Extensive experimental analysis is conducted in order to demonstrate the superiority of the two proposed approaches over state-of-the-art models.

This paper is organized as follows. After reviewing previous work on digital painting with special attention on the tracking techniques in Section II, we introduce our two approaches with implementation details in Section III and IV, respectively. We also extensively evaluated the performance of the approaches in Section V and summarize our contribution as a conclusion in Section VI.

II. RELATED WORK

Numerous efforts have been made for computer-mediated interactive drawing. The core technical component for this, are the 2D, 2.5D, or even 3D pose tracking of a drawing tool, i.e., stylus or brush. In general, tracking for digital drawing can be categorized into two; external camera-based vision tracking and capacitive-based or induction-based surface tracking. This section reviews relevant research examples on each category as well as researches particularly concerned with the tracking of flexible bristles of a brush.

A. VISION-BASED TRACKING

Vision-based drawing tool tracking is usually employed when an application requires 3D pose of the tool, e.g., 3D interaction with a tangible tool in virtual or augmented reality environments. For instance, ARPen is introduced for a 3D modeling task, where a 3D-printed pen combined with a smartphone is used [8]. The interaction was done in the mid-air, which enables drawing and interacting with virtual objects. Visual markers and ArUco, an OpenSource library utilized are utilized to track the position of the pen tip. Milosevic et al. proposed a SmartPen for the sketch-based surface modeling in [9], where a stereo webcam and Inertial Measurement Unit (IMU) are utilized. These sensors provide sequences of sorted 3D points, which are used to estimate the absolute position and orientation of the pen tip. Their work permits to obtain the style lines of actual objects, including concave parts and shapes. Moreover, they presented sketch-based modeling for automatically producing the 3D virtual model using an interactive surface sketching approach. Similarly, Wu et al. [10] introduced a system for tracking of a passive stylus for drawing in augmented reality and virtual reality environments. In their work, a square marker on the 3D printed pen was applied and inter-frame corner tracking is performed. In their work, they applied the pyramidal LK optical flow algorithm to track the marker corners on each frame. In [11], a 6DOF digital pen was designed for performing drawing in the tablet as well as mid-air, where a Vicon motion capturing camera is employed for tracking the pen position and orientation. Although tangible tools gave a solid feeling in interaction, these systems sometimes require visually distracting markers that may disturb artists' creativity and did not consider the tracking of a brush with actual deforming bristles.

B. SURFACE TRACKING

The most common way of digital drawing is to use a digital tablet where a contact point between a tool and the surface

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3004173, IEEE Access

of the tablet can be tracked. In general, digital tablets are provided with a stylus rigid tooltip or sometimes support the tracking of bendable or flexible bristles for better feedback.

1) Rigid Stylus based

From both the consumer and academic perspective, the rigid stylus type drawing pen has been explored the most to perform the interactive digital painting and has a variety of diverse implementations. The pressure-sensitive stylus is the most common form of a stylus, which often supports the tracking of different pressures. The pressure sensor is utilized in [1], [2], [12] for performing the interactive digital drawing. Han et al. [1] introduced a 6-DOF Pen, namely IrPen for drawing on tablets, which includes a pen tip, a pressure sensor, 12 LEDs and a button. In their work, if the pen strokes the tablet, then the pen tip pushes the pressure sensor, which transmits the pressure value along with the button's state. The IrPen sensor's components are a trans-impedance amplifier and a band-pass filter. Similarly, in [3] and [13], the capacitive magnetic field sensor is employed for tracking the stylus position. Liang et al. [3] used thin magnetic sensor grid which formed of Winson WSH138 Hall sensors in a grid fashion. In their work, the magnetic field image is captured in the frame by frame manner. Once a new image is formed, the centroid of the employed magnetic field and the magnitude of the field are captured as the position and the magnitude respectively. Recently, FlexStylus was introduced for painting on a tablet using an optical bend sensing approach in [4]. Their work utilizes embedded optical sensors to track deflection, rotational and position. The FlexStylus uses four directional fiber optic sensors, which allows detecting the directional bend. The FlexStylus is attached to a computer utilizing an Arduino Uno. In the software part, the input of each flex is mapped to a value within zero and one.

2) Brush with flexible bristles

In contrast, brush with flexible bristles is considered only in a few studies for interactive digital drawing. For instance, Vandoren et al. presented a digital painting interface, which employed a new brush with infrared light carrying bristle fibers for painting [5]. In their work, the paint table uses an optical diffuse film surface whereas the paintbrush is designed with an IR-led. Later on, Vandoren et al. proposed an interactive canvas for digital paint system, which finds the contact point of the brush with the painting canvas and the direction of the brush bristles [6]. Their canvas includes three layers: the transparent surface layer, the diffuser screen, and the transparent support layer. The transparent surface layer is the main drawing surface and it includes the contact sensor. The second layer uses back-projection to display the painted drawing. Finally, the third layer gives mechanical stability. Furthermore, IR light is utilized to perform the interactive drawing. Da Vinci VIRTO is a painting brush introduced for drawing in the tablet using its conductive and well-protected brush fibers [7]. In [14] a new brush model for digital Chinese calligraphy was introduced where a set of energy functions was considered to establish the brush dynamics. The above examples tried to mimic the softness of the brush, but they were not perfect due to non-paper or fabric canvas and the form factor of the brush that differs from the real one.

III. SILHOUETTE-BASED BRUSH TIP TRACKING

The focus of the paper is to develop a hardware and software framework that realizes the aforementioned scenario, i.e., digitally capturing the artist's stroking when drawing is performed using a real brush on a real canvas. This allows traditional painting artists to focus on their performance during digital capture with minimum sense of difference from that they are familiar with. The following requirements should be set. First, the haptic texture of the surface touched by the brush should be real, indicating that the surface tracking techniques in normal digital tablet is not feasible in our scenario. Second, the brush tool itself should remain as intact as possible. According to personal communication with two traditional Korean Buddhist artists, attaching even small additional artifacts to the brush changes the weight of the brush and significantly distracts the artists from concentration.

Our first approach for the requirements is based on image processing of infra-red silhouette. The main idea of the approach is to use two IR cameras [15] to acquire silhouettes of deformed bristles from multiple directions and to reconstruct the 2D coordinate of the tip from the silhouettes.

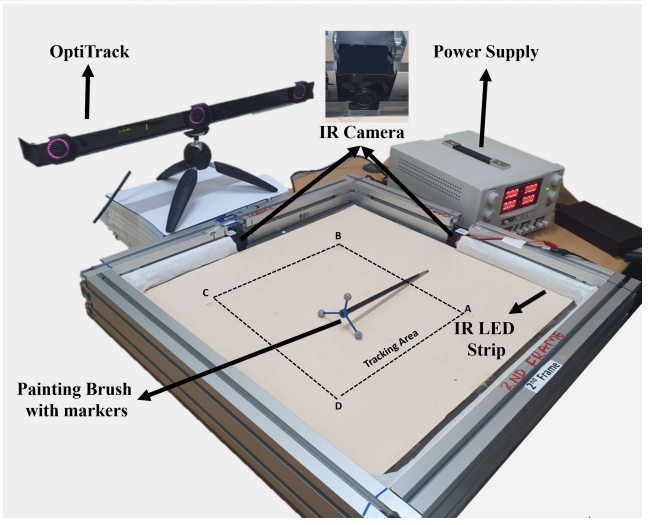


FIGURE 1. Real canvas and real brush with the drawing setup for tracking of flexible brush tip on real Canvas. A retro-reflective markers were attached to the brush handle which is used to track the brush position and orientation using an external tracker (V120 Trio; OptiTrack). Additionally, two IR (infrared) cameras (AR0330 CMOS) are attached to the rigid canvas plane, which are used to track the brush tip position. An array of IR LEDs on the opposite side of each camera wrapped with a semi-transparent paper that diffuses the IR light evenly.

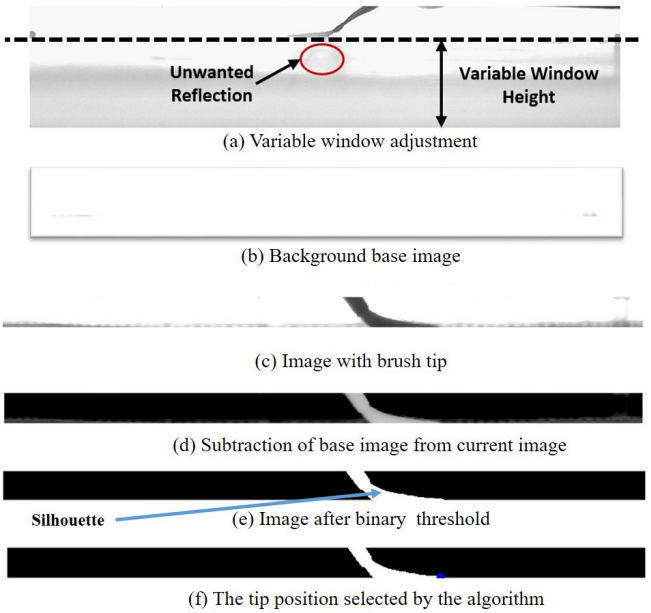
The hardware setup is shown in Figure 1. An external tracker (V120 Trio; OptiTrack) [16] is installed in the system and tracks the position and orientation (6DOF pose) of the rigid handle of the brush. We specifically chose the OptiTrack since it provides reasonably low position tracking latency

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3004173, IEEE Access

with higher accuracy compared to the IMU sensor-based tracking [9], which well fits to our purposes. Furthermore, the tracking system is very user friendly. This tracker has three purposes. First, the pose of the brush handle is used to detect the contact between the brush and the canvas. Second, the possible area containing the contact is estimated from the handle data and used to increase the accuracy and reliability of the silhouette tracking. Third, captured handle data is used for the evaluation of the system later.

For tracking the pose of the rigid handle, a retro-reflective markers were attached to the handle as shown in Figure 1. The weight of the markers is very small and perceptually negligible (0.95 g). The Motive software from the OptiTrack provides the 6DOF pose of the brush handle.

Additionally, two IR cameras (AR0330 CMOS) [15] are attached to the rigid canvas plane in such a way that the canvas plane is exactly at the center of and perpendicular to the image plane of the camera. Two IR cameras are placed at different sides of the rectangular canvas as shown in the Figure 1. The cameras are wide-angle cameras (field of view of 170 degrees) to increase the tracking space. Total tracking space is 300 mm $\times$ 300 mm. The distortion of each camera is estimated through an intrinsic parameter calibration process with a checkerboard pattern and compensated in the tracking procedure. The clarity of silhouette is enhanced by bright IR background using an array of IR LEDs on the opposite side of each camera wrapped with a semi-transparent paper that diffuses the IR light evenly. Whenever the brush comes between the camera and the corresponding LED, the body of the brush creates a sharp silhouette as shown in Figure 2(e).



(a) Variable window adjustment

(b) Background base image

(c) Image with brush tip

(d) Subtraction of base image from current image

(e) Image after binary threshold

(f) The tip position selected by the algorithm

FIGURE 2. Illustration of the procedures used in our approach. Detailed explanation of the figures are in the following algorithms.

The software part of the system is implemented by following a sequence of procedures. The procedure is explained using a step-by-step approach as follow:

- **Step 1: Brush Tip position T_{pos} Estimation**
  - Compute the current T_{pos} from the brush handle pose coming from OptiTrack.
- **Step 2: Intersection Point P_i Computation**
  - Calculate P_i from the intersection of T_{pos} and the tracking plane $ABCD$ defined on the surface of canvas.
- **Step 3: Camera Perspective Scaling factor S**
  - Calculate distance d_x and d_y from the camera reference line AB (x-axis) and BC (y-axis) respectively.
  - Find the first order linear relation between the horizontal plane and the camera perspective projection.
    $$(S_x\ ,\ S_y) = (a_x d_x + c_x,\ a_y d_x + c_y),$$
    where S and d represents the scaling factor and the distance from the camera line, respectively. a and c are the coefficients of the first order equation, and the subscripts x, y are the corresponding axes.
- **Step 4: Height of Cropping Window**
  - The camera reference line distance is used to calculate the height of the cropping window in order to remove possible brush reflections on the canvas surface as shown in Figure 2(a).
- **Step 5: Background Base Image I_b**
  - Capture background base image I_b without performing the drawing and remove the distortion using the camera distortion parameters as presented in Figure 2(b).
- **Step 6: Silhouette Extraction**
  - Capture the current Image I_c which includes the brush during drawing and remove the distortion, as presented in Figure 2(c).
  - Crop both images I_b and I_c based on threshold value and the window height calculated.
  - Convert the images to gray-scale and subtract I_c from I_b, as illustrated in Figure 2(d).
  - Convert the subtracted image to binary by applying a threshold value, which is shown in Figure 2(e).
- **Step 7: Brush Tip Position (x, y) in 2D-Coordinates**
  - In order to find the brush tip from the silhouette of the brush, each horizontal line of the binary image is summed starting from the lowest line.
  - If the summation of pixel values exceeds a certain threshold the line is considered to contain the tip.
  - Our approach finds the first pixels on either end of the silhouette. The column of each selected pixel is summed and the pixel with the minimum sum is considered as the position of brush tip in pixels P_x and P_y along the designated axis (P_x and P_y in case of I_c taken camera 1 and camera 2, respectively).
  - The horizontal coordinate of the brush tip is recorded in pixel number, as presented in Figure 2(f).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3004173, IEEE Access

– The position in pixels P_x, P_y, and S are further used to calculate the position of brush tip in real world 2D-coordinate (x, y) mm by using the following relation:

$$(x, y) = \left(\frac{P_x \times S_x}{\text{No. of Pixels in Horizontal line}}, \frac{P_y \times S_y}{\text{No. of Pixels in Horizontal line}}\right)$$

The proposed silhouette-based approach can effectively track the brush tip position. However, this approach needs a specially aligned frame and cameras, which may increase the complexity of the system. In addition, the user's hand may occlude the brush and occlusion problems may arise in the IR cameras during drawing. As an alternative to remedy these shortcomings, we introduce the second approach in the next section.

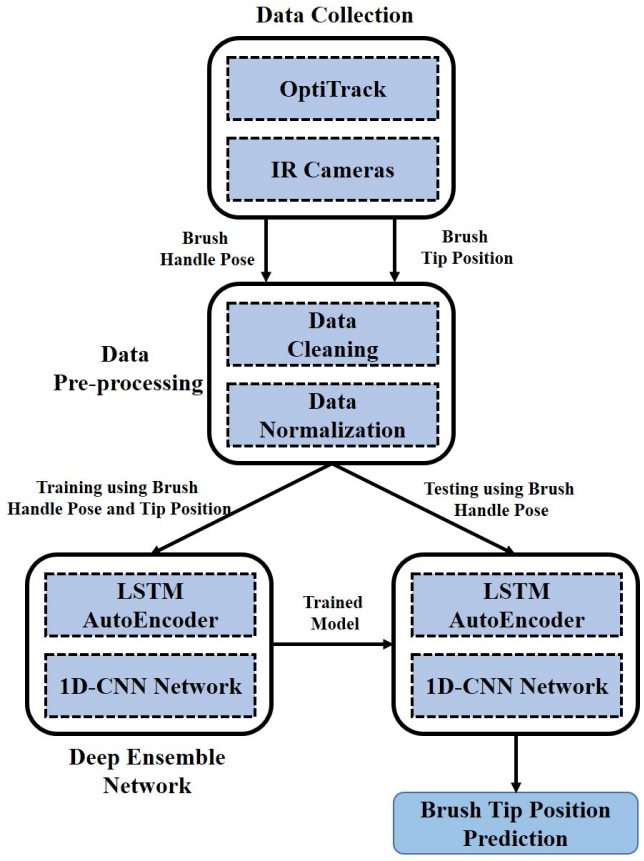


FIGURE 3. Proposed framework for deep ensemble network-based brush tip estimation. Optitrack and IR cameras are used for collecting the data for the deep network training. Once the network is trained, our model estimate the tip position during actual drawing, taking the brush handle pose from the Optitrak as an input.

IV. DEEP ENSEMBLE NETWORK-BASED BRUSH TIP ESTIMATION

This section presents our second approach based on a novel deep learning network, which overcome the issues of the silhouette-based approach. A newly designed deep ensemble network is trained in offline using data captured through an external tracker (Optitrack V120) and the silhouette-based approach. The network captures the relationship between the 3D pose of a brush handle (6DOF data) and the 2D brush tip position on the canvas. During actual drawing, the trained network estimates the brush tip position by taking the brush handle pose as an input, allowing us to use real canvas with a real brush.

Figure 3 illustrates the overall data flow of the approach. First step is the data collection where data for network training are captured using the external trackers. We then perform data pre-processing, which includes data cleaning and normalization. Data cleaning is done in order to remove the noise and outliers from the tracking data, while data normalization is employed so that all data is in the proper scale.

The characteristic of the data and relationship among them are as follows. For each 6DOF time-series data (handle pose), the network should produce corresponding 2DOF time-series data (tip position). The tip position depends not only on current input, but also on previous inputs and previous outputs. In order to cope with such characteristics, our design combines LSTM Autoencoder and 1D Convolutional Neural Network (CNN) as shown in Figure 4. This ensemble network is employed mainly to increase the representing power while minimizing overfitting. Both the proposed LSTM Autoencoder and 1-D CNN are responsible for the analysis of the time-series data. From our extensive test, we confirmed that the ensemble of the two networks significantly outperforms single network, as shown in Section V. The following sections discuss the details of the LSTM Autoencoder and 1-D CNN.

A. LSTM AUTOENCODER

The sequential information can be effectively mapped by the Recurrent Neural Network (RNN) through hidden states. However, if the inputs are long sequences, then simple RNN-based methods may experience gradient explosion and gradients vanishing problems. To train long sequences of time-series data, the Long-Short Term Memory (LSTM) was introduced, which includes input, forgetting, and output gates [17]. The equations of LSTM at time t can be represented as follows.

$$i_t = \sigma(W_i(x_t + h_{t-1}) + b_i) \quad (1)$$

$$f_t = \sigma(W_f(x_t + h_{t-1}) + b_f) \quad (2)$$

$$O_t = \sigma(W_O(x_t + h_{t-1}) + b_O) \quad (3)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot tanh(W_c(x_t + h_{t-1})) \quad (4)$$

$$h_t = O_t \odot tanh(C_t) \quad (5)$$

Where σ is the sigmoid function. i, f, c, o and h denote the input, forget, memory cell, output gates and hidden layer state, respectively. W_i, W_f, W_O and b_i, b_f, b_O represents the weight matrices and bias terms.

LSTM still suffers from prediction accuracy problem: its accuracy for time series data is not always optimal [18]. To improve the performance of LSTM, the LSTM Autoen-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3004173, IEEE Access

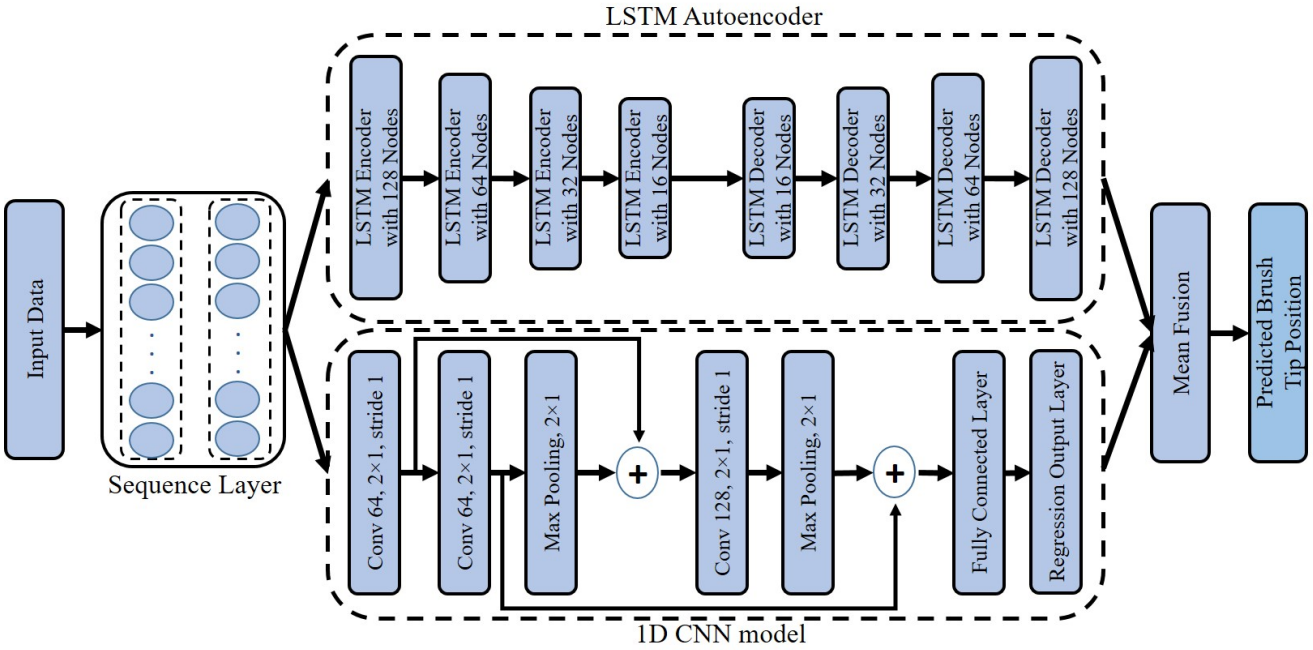


FIGURE 4. Proposed deep ensemble network for predicting brush tip position. The proposed deep ensemble network combines LSTM Autoencoder and 1D Convolutional Neural Network (CNN). LSTM Autoencoder includes four encoders and four decoders whereas 1D Convolutional Neural Network includes three convolution layers, two max-pooling layers, and a fully connected layer.

coder model was introduced particularly for video representation [19]. It includes one encoder LSTM and one decoder LSTM in its model. In their model, the LSTM encoder accepts a sequence of frames and encodes them to a fixed range feature vector, while the LSTM decoder takes the feature vector and decodes it to produce a target sequence. Later on, Sagheer et al. presented the LSTM-based stacked autoencoder for multivariate time series forecasting problems where three LSTM autoencoders are sequentially stacked [18].

In the present paper, we also use LSTM Autoencoders for the prediction. However, instead of sequentially stacking autoencoders (encoder-decoder pair) as in [18], we first put a list of LSTM encoders with gradually decreasing number of nodes, followed by LSTM decoders with increasing number of nodes. The overall design of our LSTM autoencoder network is shown in the upper part of Figure 4. The first LSTM encoder reads the input data and produces 128-feature outputs with 3 time steps. The second LSTM encoder takes the 3×128 input from the previous encoder layer and decreases the feature-length to 64, while the third and fourth LSTM encoder reduces the feature size to 32 and 16, respectively. Afterward, LSTM decoder modules decode the features. Additionally, a Repeat Vector layer is employed between the encoder and decoder which replicates the feature vector and operates as a bridge. At the end, a time distributed layer is utilized to obtain the output, which allows one-to-one relations between input and output data. We expect that the proposed LSTM autoencoder can learn more complex relationships among input layers and output layers for the given input (i.e. brush pose and brush tip position). Here, the high-level layers can learn features from lower layers and obtain higher-order and can have better summarizing power of inputs. Furthermore, it compresses the useful information layer by layer and brings performance improvement, compared with [18] where the output of one LSTM autoencoder is the input to the next LSTM autoencoder.

B. 1D CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNN) based prediction is the most commonly investigated deep learning approach in various fields of computer vision and image processing and has achieved outstanding performance [20] [21] [22]. While original CNN is usually studied to capture the features from images, we proposed a novel 1D CNN model with residual connection for the investigation of the time series data (brush tip position, brush handle pose over time). Previously, in [24], the authors introduced a novel 1D CNN network for time series data prediction, which is composed of two convolutional layers, two max pooling, followed by a fully connected layer.

Our proposed 1D CNN with residual connection is composed of three 1D convolutional layers and two 1D max-pooling layers followed by one fully connected layer as shown in the lower part of Figure 4. Compared to [24], the proposed deeper 1D CNN captures more discriminative features, which helps to improve the prediction accuracy. The convolutional layers extract the features, and the max-pooling layers reduce the dimensionality of the individual feature map. For the first two convolutional layers, 64 filters are employed with kernel size 2, and for third convolutional layers, 128 filters are applied with kernel size 2, while max-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3004173, IEEE Access

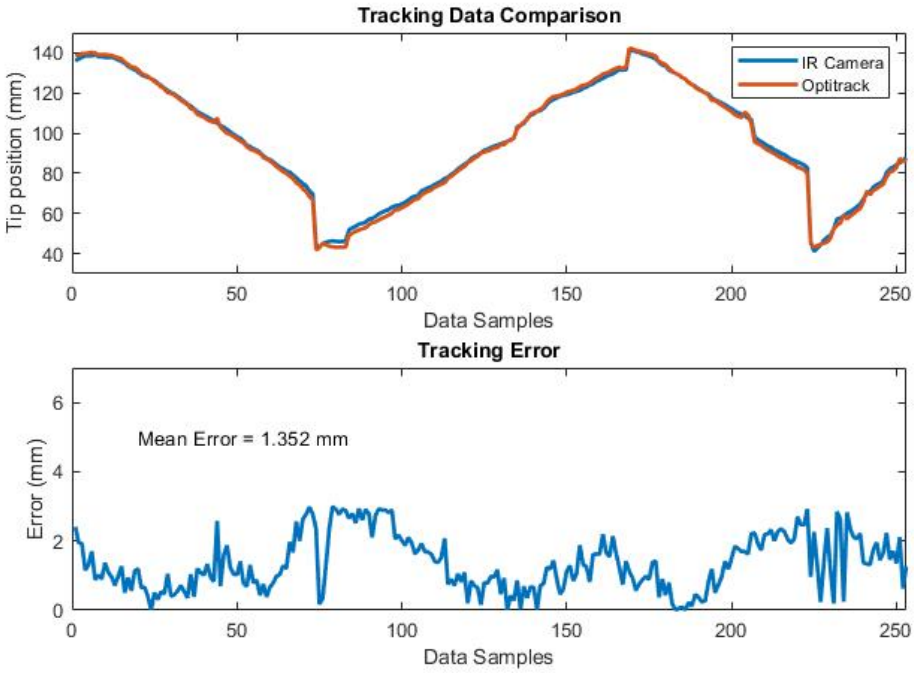


FIGURE 5. Examples of true and estimated data trajectories (upper) and the errors between two (lower) for our first approach (Silhouette-based).

pooling is performed over a window size 2. Furthermore, each convolution process is followed by Rectified Linear Unit (ReLU), which is a nonlinearity function. In the residual connection, the output of the first convolutional layer is concatenated with the output of the first max-pooling layer while the output of the second convolutional layer is concatenated with the output of the second max-pooling layer. Afterward, a flatten layer is applied before the fully connected layer to convert the feature size to a 1D vector. The network concludes with a fully connected layer and a regression output layer. In this model, the loss function used is Root Mean Squared Error (RMSE) and the optimizer employed is Adam.

Lastly, the outputs from the both networks are combined through a mean operation at the end of the model, as shown in the right-most part of Figure 4, generating the final predicted brush tip position.

FIGURE 6. Examples of pictures drawn by a master of the Korean traditional Buddhist art.

V. ACCURACY EVALUATION

We perform a series of experiments to assess the accuracy of the tracking. All the experiments are conducted on Intel(R) Core(TM) i5-7600 CPU @3.50GHz with 16GB RAM running Windows 10. The proposed deep ensemble network is trained offline using data captured through an external tracker (Optitrack V120) and the silhouette-based approach. During actual drawing, the trained network estimates the brush tip position by taking the brush handle pose as an input, allowing us to use real canvas with a real brush. During the testing process, the system works in real-time, since at that time, it only tracks the brush handle pose (position and orientation) and the proposed deep ensemble network takes this brush handle pose as input and predicts the brush tip position in real-time.

A. SILHOUETTE-BASED TRACKING

To get ground-truth position data of the brush tip, a very small (diameter of 3 mm) spherical retro-reflective marker is glued at the tip of the painting brush. The position of the marker can be tracked through OptiTrack tracker. Note that this

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3004173, IEEE Access

TABLE 1. Number of instances used for training and testing the proposed deep ensemble model. Each instance consists of brush tip position, brush handle position and brush handle orientation.

Drawing no.	Number of instances for training the model	Number of instances for testing the model
Drawing 1	4325	6974
Drawing 2	5037	13869
Drawing 3	7573	12045
Drawing 4	11248	11171
Drawing 5	13244	N/A
Drawing 6	17597	N/A

setup cannot be used in our application scenario since real paint blocks the marker. For data collection, we performed multiple strokes for 60 seconds on the surface inside the tracking region. The position data was recorded with both systems, i.e., OptiTrack and our silhouette-based tracking system. Figure 5 shows the comparison of data recording with both systems.

The Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) recorded from the experiment for tracking brush tip position is 1.352 mm and 1.57 mm, respectively. Overall, in most cases, less than 1.5 mm error was observed.

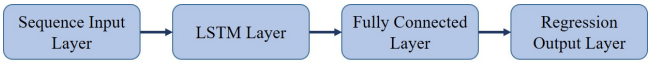


FIGURE 7. Architecture of single Layer LSTM, which includes a sequence input layer, an LSTM layer, and a fully connected layer.

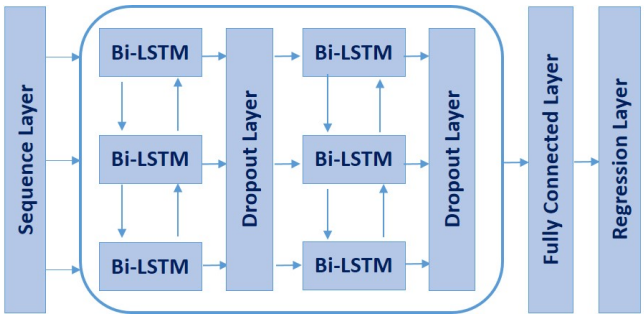


FIGURE 8. Architecture of multi-layered Bi-LSTM network, which comprises of a sequence input layer, two Bi-LSTM layers and followed by a fully connected layer.

B. ENSEMBLE NETWORK-BASED ESTIMATION

In order to obtain data for training the deep network in our second approach, a traditional Korean Buddist painting master was invited and asked to actually perform his painting. The master has drawn several drawings, which took almost 2 hours. Figure 6 shows some examples of the paintings that he has drawn. While he performs, we collected data of the brush tip position using our first approach. Note again that direct attachment of small marker at the tip was not feasible since the master was actually drawing the artwork with real paints. Instead, we decided to use the results of our silhouette-based system as a ground-truth data for model training. Although the first approach indeed has some tracking error, we still think that it can be used for the purpose of the evaluation of the second approach due to two reasons. First, the error is very small since it is direct vision-based measurement. Second, errors from the silhouette-based tracking do not significantly affect the results in this section since this section examines how well our second approach estimates the input, whatever the input is. In order to evaluate the performance of the proposed deep ensemble network, total 10 drawings are performed, of which 6 drawings are used for training the deep ensemble model and 4 drawings are performed for testing the proposed deep ensemble network. Table 1 illustrates the number of instances used for training and testing the proposed deep ensemble model. Each instance consists of brush tip position, brush handle position and brush handle orientation.

To demonstrate the superiority of the proposed deep network model over other state-of-the-art models, we additionally implemented 6 other networks and trained them with the same data. The six models are an ARIMA predictor [23], 1D CNN [24], single Layer LSTM, deep long-short term memory (DLSTM) [25], multi-layer Bi-LSTM and LSTM-based stacked autoencoder (LSTM-SAE) [18].

Figure 7 and 8 illustrates the architecture of the single-layer LSTM and multi-layered bi-LSTM model, respectively. In the single-layer LSTM, the network starts with a sequence input layer followed by an LSTM layer. The network ends with a fully connected layer and a regression output layer. The multi-layered bi-LSTM comprises of two bi-LSTM layers followed by the fully connected layer and a regression output layer.

The error metric was the Root Mean Square Error (RMSE), which can be expressed by

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(x_i - \overline{x}_i)^2}, \tag{6}$$

where M is the total sample, $\overline{x}_i$ represents the predicted value and x_i is the ground-truth of the $i-th$ sample.

Figure 9 shows the examples of measured and estimated trajectories for the drawings. The predicted trajectories coincide quite well with the measured ones. For better visualization, Figure 10 plots the correlation between the measured and estimated data. For all cases, the correlation coefficient reaches up to 0.91. Statistics on the estimated errors for all the models are summarized in Figure 11. As it is clearly shown, the proposed framework outperforms the other approaches by showing significant improvement. Table 2 presents the RMSE of the test models. For all the experiments, the proposed deep ensemble network shows the lowest RMSE. These experiments prove the superiority of the proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3004173, IEEE Access

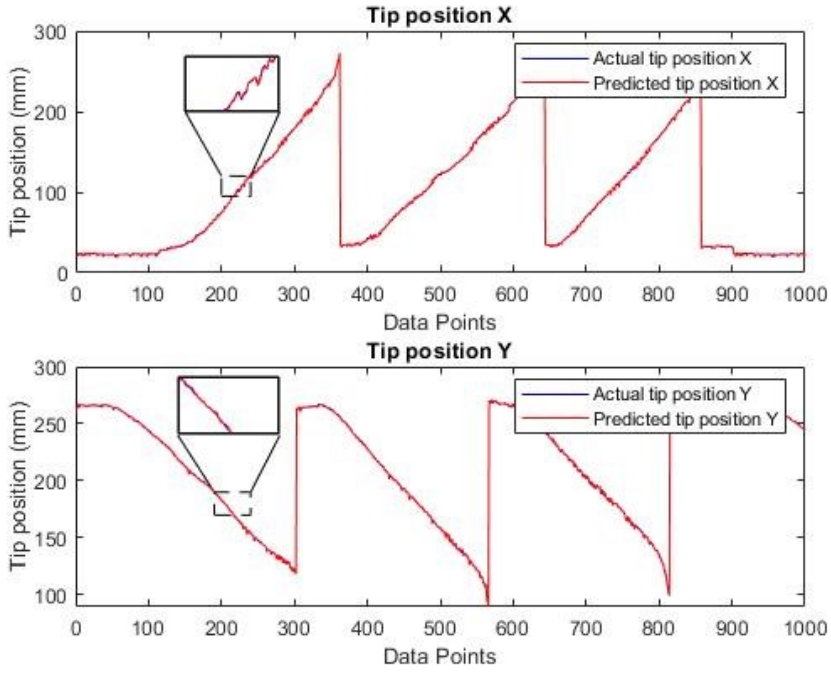


FIGURE 9. Data trajectories of true brush tip and predicted brush tip using our second approach (deep ensemble network) for both x and y axis. Blue line represents the actual tip position whereas red line represents the predicted tip position.

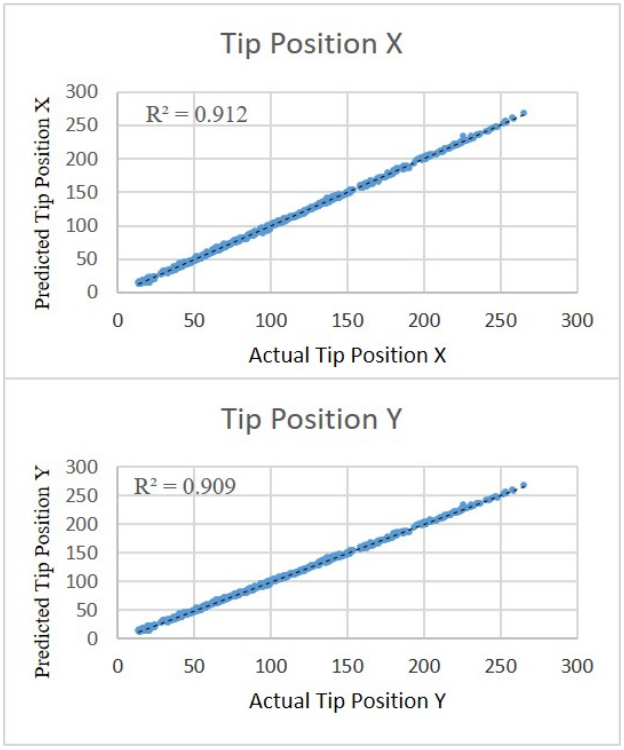


FIGURE 10. Correlation between predicted and actual brush tip position for all drawings using proposed deep ensemble network.

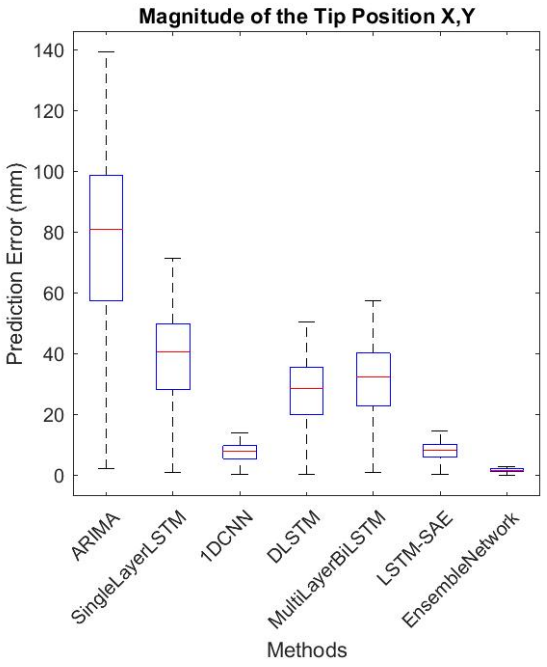


FIGURE 11. Statistics on the prediction error of our EnsembleNetwork method compared with other state-of-the-art methods. Tip position x and y are combined by taking the magnitude.

approach over state-of-the-art models.

The results demonstrate that the proposed deep ensemble network is capable of estimating the brush tip position with an average error of ± 1 mm. These results are satisfactory considering the size of the drawing canvas area (300×300

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3004173, IEEE Access

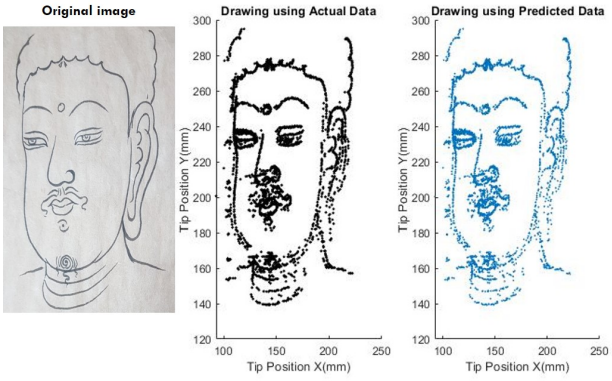


FIGURE 12. Original drawing (left) compared with digitally drawn images using true date (center) and predicted data (right).

TABLE 2. The results of comparison between proposed deep ensemble network and the state-of-the-art methods. The numbers represent the RMSE (root mean square error) in millimeter for both x and y tip position.

Methods	RMSE of brush tip position X (mm)	RMSE of brush tip position Y (mm)
ARIMA [23]	48.22	51.93
Single Layer LSTM	25.59	25.94
1-D CNN [24]	4.78	4.98
DLSTM [25]	19.82	15.83
Multi-Layer Bi-LSTM	21.71	19.49
LSTM-SAE [18]	5.25	5.15
Proposed 1-D CNN	2.24	2.23
Proposed LSTM Autoencoder	1.818	2.25
Proposed Ensemble Network	**0.972**	**1.06**

mm) and the size of the painting drawn. Figure 12 presents the qualitative result for visualizing the produced drawing by the silhouette-based approach utilizing actual brush tip position and deep ensemble network-based approach using the predicted brush tip position respectively. From this result, we can observe that the predicted one is almost equivalent to the ground truth.

VI. CONCLUSION

In this work, we introduced silhouette-based and deep ensemble network-based approaches to track the brush tip position for interactive drawing. The silhouette-based approach captures the silhouette of deforming bristles using a pair of well-aligned infra-red (IR) cameras, extracts the tip using our proposed tracking procedure and then the 2D position of the tip is reconstructed. However, this approach still needs a specially aligned frame and cameras and has shortcoming in usability. So, in order to overcome this limitation, we proposed a deep ensemble network that predicts the brush tip position by taking the brush handle position and brush orientation as input. Using this predicted brush tip position, we can achieve an interactive drawing. Lastly, experiments are conducted to demonstrate the superiority of the proposed deep ensemble network over state-of-the-art models.

In the current work, we only consider a standard size brush. To increase the applicability of the system as a future work, we will consider identifying the different traits with different kinds of brushes as well as their calibration process.

REFERENCES

[1] Jaehyun Han, Seongkook Heo, Hyong-Euk Lee and Geehyuk Lee, "The IrPen: A 6-DOF Pen for Interaction with Tablet Computers", IEEE Computer Graphics and Applications, vol. 34(3), pp. 22-29, 2014.

[2] Neo Smartpen N2. [Online] Available: https://www.neosmartpen.com/ (accessed on 19 th January 2020).

[3] Rong-Hao Liang, Chao-Huai Su, Chien-Ting Weng, Kai-Yin Cheng, Bing-Yu Chen and De-Nian Yang, "GaussBrush: Drawing with Magnetic Stylus", In Proceedings SIGGRAPH Asia 2012 Emerging Technologies, 2012.

[4] Nicholas Fellion, Alexander Keith Eady and Audrey Girouard, "FlexStylus: A Deformable Stylus for Digital Art", In Proceedings of the CHI Conference Extended Abstracts on Human Factors in Computing Systems, pp. 2482-2489, 2016.

[5] Peter Vandoren, Tom Van Laerhoven, Luc Claesen, Johannes Taelman, Chris Raymaekers and Frank Van Reeth, "IntuPaint: Bridging the Gap between Physical and Digital Painting", In Proceedings of the 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems, 2008.

[6] P. Vandoren, L. Claesen, T. Van Laerhoven, J. Taelman, C. Raymaekers, E. Flerackers, and F. Van Reeth, "FluidPaint: An interactive digital painting system using real wet brushes," In Proceedings of the Interactive Tabletops Surfaces, pp. 53–56, 2009.

[7] Tablet brush da Vinci VIRTO. [Online] Available: https://www.davinci-defet.com/englisch/artist-brushes/special-markets/tablet-brush-da-vinci-virto.html/ (accessed on 16 January 2020).

[8] Philipp Wacker, Oliver Nowak, Simon Voelker, and Jan Borchers, "ARPen: Mid-Air Object Manipulation Techniques for a Bimanual AR System with Pen & Smartphone", In Proceedings of the CHI Conference on Human Factors in Computing Systems, 2019.

[9] Bojan Milosevic, Flavio Bertini, Elisabetta Farella, and Serena Morigi, "A SmartPen for 3D interaction and sketch-based surface modeling", The International Journal of Advanced Manufacturing Technology, vol. 84, pp.1625–1645, 2016.

[10] Po-Chen Wu, Robert Wang, Kenrick Kin, Christopher Twigg, Shangchen Han, Ming-Hsuan Yang, and Shao-Yi Chien, "DodecaPen: Accurate 6DoF Tracking of a

Passive Stylus", In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, pp. 365-374, 2017.

[11] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice and Karan Singh, "SymbiosisSketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ", In Proceedings of the CHI Conference on Human Factors in Computing Systems, pp. 1-15m 2018.

[12] Intuos Creative Stylus. [Online] Available: https://www.wacom.com/en-us/getting-started/intuos-creative-stylus-welcome (accessed on 19 th January 2020).

[13] AmazonBasics Stylus. [Online] Available: https://www.secretasianman.com/best-digital-pen-for-artists/ (accessed on 19 th January 2020).

[14] Bendu Bai Kam-Wah Wong and Yanning Zhang, "An Efficient Physically-Based Model for Chinese Brush", In Proceedings of the International Workshop on Frontiers in Algorithmics, pp 261-270, 2007.

[15] IR camera. [Online] Available: http://www.ircameras.com (accessed on 5 September 2018.

[16] OptiTrack V120. [Online] Available: https://optitrack.com/products/v120-trio/ (accessed on 5 September 2018).

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory", Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] Alaa Sagheer and Mostafa Kotb, "Unsupervised Pre-training of a Deep LSTM-based Stacked Autoencoder for Multivariate Time Series Forecasting Problems", Scientific Reports, Vol. 9, 19038, 2019.

[19] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using lstms", In Proceeding of the 32nd International Conference on International Conference on Machine Learning, pp. 843-852, 2015.

[20] A. Krizhevsky, I. Sutskever, G.E. Hinton, "ImageNet classification with deep convolutional neural networks", In Proceeding of the Advances in Neural Information Processing Systems, pp. 1097–1105, 2012.

[21] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition", In Proceeding of the International Conference on Learning Representations, 2015.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, "Deep Residual Learning for Image Recognition", In Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[23] J. D. Hamilton, "Time Series Analysis", vol. 2, Princeton, NJ, USA: Princeton Univ. Press, 1994.

[24] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel and Sheraz Ahmed, "DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series", IEEE Access, Vol. 7, pp. 1991-2005, 2018.

[25] Alaa Sagheer and Mostafa Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks", Neurocomputing, Vol. 323, pp. 203-213, 2019.

JOOLEKHA BIBI JOOLEE is currently a Ph.D. student of the Dept. of Computer Science and Engineering in Kyung Hee University, South Korea. She received her Masters degree in Computer Science and Engineering from kyung Hee University, South Korea and received her B.S degree in Computer Science and Engineering from International Islamic University Chittagong, Bangladesh in 2015. Her research interests include Human-computer Interaction, Augmented Reality, Haptics, and Deep Learning.

AHSAN RAZA received a B.S. degree in Computer Engineering from University of Engineering and technology (UET) Taxila, Pakistan in 2015. Currently, he is persuing his PhD. degree in the department of Computer Science and Engineering at Kyung Hee University, South Korea. His research interests include mid-air haptic feedback, haptic guidance and perception, and psychophysics.

MUHAMMAD ABDULLAH received his Masters degree in Computer Science and Engineering from Kyung Hee University, South Korea and his BS degree in electrical engineering from the National University of Science and Technology (NUST), Pakistan. His research interests include encountered-type haptics devices and virtual reality.

SEOKHEE JEON received the B.S. and PhD. degrees in computer science and engineering from the Pohang University of Science and Technology (POSTECH) in 2003 and 2010, respectively. He was a postdoctoral research associate in the Computer Vision Laboratory at ETH Zurich. In 2012, he joined as an assistant professor the Department of Computer Engineering at Kyung Hee University. His research focuses on haptic rendering in an augmented reality environment, applications of haptics technology to medical training, and usability of augmented reality applications.