**Affine transformation of virtual 3D object using 2D localization of fingertips**

Mohammad Mahmudul Alam, S. M. Mahbubur Rahman

1. Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000, Bangladesh

* Corresponding author, mahbubur@eee.buet.ac.bd

**Abstract** Background Interactions with virtual 3D objects in the virtual reality (VR) environment using the gesture of fingers captured in a wearable 2D camera have emerging applications in real-life. Method This paper presents an approach of a two-stage convolutional neural network, one for the detection of hand and another for the fingertips. One purpose of VR environments is to transform a virtual 3D object with affine parameters by using the gesture of thumb and index fingers. Results To evaluate the performance of the proposed system, one existing, and another developed egocentric fingertip databases are employed so that learning involves large variations that are common in real-life. Experimental results show that the proposed fingertip detection system outperforms the existing systems in terms of the precision of detection. Conclusion The interaction performance of the proposed system in the VR environment is higher than that of the existing systems in terms of estimation error and correlation between the ground truth and estimated affine parameters.

**Keywords** Affine transformation; Detection of fingertips; Detection of hand; Human-computer interaction; Virtual reality

**1 Introduction**

The recent trend in the research areas of augmented reality (AR), virtual reality (VR), and mixed reality (MR) have moved towards blending the real and virtual worlds to generate new environments and visualizations, where the physical objects interact with the virtual objects almost instantaneously\(^\text{[1-4]}\). As a result of the gradual improvement of the processing power of modern gadgets such as smartphones, virtual reality-based applications are becoming an essential part of these devices. It has shown that fingertip based interaction with a virtual object is easy to use and more satisfactory than that with a mouse even though the mouse requires comparatively less task completion time\(^\text{[5]}\). However, existing dedicated VR and MR devices including Oculus and Microsoft HoloLens are expensive for common use. Thus, an affordable and versatile interaction
system may be a good solution to reach the mass of the users. Such a system would use a freehand gesture with a wearable 2D color camera in the egocentric vision for establishing an interaction between the real and the virtual world. Since hand gesture plays a vital role in visual communication in many cases, an interaction system with the fingers of the hand using a wearable camera would be of great interest. Nevertheless, recognizing finger gesture and movement in a real-life environment using a wearable 2D color camera is a challenge. A Kinect sensor-based depth camera can perform well for detecting hand and finger landmarks\textsuperscript{[6–8]}. But the challenges increase further while detecting fingertips using a regular 2D color camera which is an inexpensive option. In convention, there are image processing- and machine learning-based approaches to address the problem of detection of fingertips using a 2D camera. Image processing approaches perform operations directly on images to extract information whereas machine learning approaches learn parameters iteratively from the data. The image processing-based techniques\textsuperscript{[6, 9–12]}, have the dependency of the background, and hand shape and color and tend to fail in the presence of a complex environment. On the other hand, machine learning-based approaches perform better than image processing-based techniques\textsuperscript{[13]}. However, the existing machine learning-based techniques are also prone to detection errors in the presence of real-life movements of fingers\textsuperscript{[14–18]}. For instance, if two fingers are in close proximity these methods often fail to detect their positions. Particularly, their relative distance carries significant gesture-based information and that needs to be preserved to realize an interactable VR environment.

This paper addresses the above issues by proposing a convolutional neural network (CNN)-based approach to detect fingertips without much dependency on the shape or color of the hand or the background of the environment. The proposed system is robust to localize the fingertips that can trace even the close proximity of the fingertips in natural movements of hand and fingers which is desirable for any VR applications. Specifically, a two-stage CNN is proposed for the detection of hand and fingertips. In the first stage, the network learns the detection of hand. The network localizes the hand and predicts the probability of having a hand in an image. If the detected hand probability exceeds a given threshold, the related portion of it is cropped from the image for further processing. Afterward, the second network learns to locate the thumb and index fingertips from this cropped part of the hand. Finally, the detected fingertips are used to control and interact with a virtual object by designing an interactive VR environment. With the gesture of the thumb and index fingertips and by tracking the hand, the variation of the scale, rotation, translation, and in general, the affine transformation of the virtual 3D object in the VR environment is demonstrated. Furthermore, different virtual environments are created to show the performance comparison of the proposed system with the existing systems with the aid of different participants. The experimental results reveal that the proposed system outperforms the existing systems in terms of the estimation error of fingertips coordinates and the correlation between ground truth and estimated affine parameters. Prior to detailing the methodology of the proposed fingertip based autonomous interactive VR system, in the following subsections, the historical positioning, the related works, the scope of analysis, and the specific contributions are outlined sequentially.

1.1 Historical positioning
One of the earliest works of virtual object manipulation and interaction using the hand in the VR environment was presented in [19, 20]. These methods were limited to grabbing and selecting a virtual object. In 2004, Tomozoe et al. presented the VR interaction method with movability property attached to every virtual object [21]. Similarly, Kiyokawa and Takemura presented virtual object positioning and holding [22]. In these initial methods a special sensor, marker, or glove was used to locate hand position. However, with the incremental development of the processing power of modern computers image processing-based hand and fingertip detection for virtual object interaction became popular [23–30]. In recent days, due to the rise of deep learning-based methods and low-cost GPU, it is expected that virtual object interaction with such learning-based hand and fingertip detection methods will be on the rise such as that presented by Alam and Rahman [31].

1.2 Related works

The general approach of a fingertip-based interactive VR system is to localize the hand and fingertips first and then establish interaction with a virtual object by using the detected fingertips. For example, Lee and Hollerer presented an interactive AR system in which the hand takes part in the role of a marker [23]. This method detects the fingertips by using a curvature-based algorithm to segment hand adaptively. Only the translational transformation of a virtual object placed on the palm of the hand is considered in this method. Rani et al. presented a positional control of a 2D virtual object where by using image processing techniques such as image thresholding, contour extraction, and gesture detection, the hand is detected [24]. Likewise, Bai et al. developed a technique of manipulating a virtual object in 3D space for scaling, rotating, and translating individually in AR devices by using finger gesture [25]. A fingertip-based MR interface was developed by Song et al. to play games where fingertip tracking is performed using shape detection of the fingers [26]. An augmented assembly system is presented by Ong and Wang performing rotation and translation transformation of a virtual object in the AR environment using a 3D natural bare-hand interaction system [27]. Le and Kim proposed a hand gesture detection based framework for learning the 3D geometry of an object in an augmented environment where they incorporated scale, rotation, and translation transformation for controlling the objects [28]. Weichel et al. introduced an MR-based system for self-designed fabrication using hand gestures where the user can shape the virtual objects by interacting with the objects [29]. Lee et al. adopted a robotic approach to select and grip a virtual object using multiple senses of hand that is obtained from pinch glove, hand gesture, and vibrotactile feedback to provide natural interaction [30].

1.3 Scope of analysis

In practice, the hand serves as the key component to interact with the virtual environment. Hence, the existing literature focuses on detecting hand and fingertips first and later use the result of detection to interact with the virtual objects [31, 32]. The study reveals that the detection method of hand and fingertip using image processing technique has a dependency on background, and hand shape and color. On the contrary, the detection of fingertips using a machine learning approach is robust in the different environments usually having a complex background with a rare dependency on illumination variation and hand shape and color. Consequently, there remains a scope to incorporate a machine learning technique such as CNN for detecting
fingertips with a view to the realization of a robust VR interaction system. There exists an approach of interaction in the virtual environment using complex hardware such as the use of the Leap Motion controller or Microsoft Kinect depth sensor. But these devices are expensive for most users and hard to integrate into a smartphone as a portable device. Hence, the development of a fingertip-based VR interaction system using a single 2D wearable camera is in demand. In terms of interaction with a virtual object, most of the existing 2D camera-based VR interaction systems consider selection, translation, or scaling of a virtual object. Therefore, it is worthwhile to carry out a generalized affine transformation of a virtual 3D object interacting with fingertips using a CNN-based machine learning algorithm with the aid of a CCD camera.

1.4 Specific contributions

This paper proposes a new CNN-based approach of fingertip detection to develop a VR interaction system where the movement of the fingertips drives the affine transformation of virtual objects. The specific contributions of the paper are as follows:

- Development of CNN-based automatic detection of thumb and index fingertips in the sequence of 2D images for obtaining the affine transformation of virtual objects in a VR environment
- Evaluation of the performance of the proposed CNN-based hand and fingertip detection system using two databases, an existing and a newly developed egocentric fingertip databases, the latter of which is publicly released
- Evaluation of the performance of proposed fingertip-based VR interaction system for generalized affine transformation (e.g., scale, rotation, and translation) of the virtual 3D objects with the aid of 2D video clips labeled for fingertip positions

The paper is organized as follows. In Section 2, the detail of the proposed system is presented. Section 3 describes the experiments, comparisons, and results of the detection of the fingertips and interaction in the VR environment. Finally, the conclusion is provided in Section 4.

2 Proposed method

The fingertips are detected in the proposed system to establish an interaction between the fingertips and a virtual 3D object in the VR environment. A cascaded two-stage CNN architecture is proposed for the detection system of the fingertip. In this architecture, the hand is recognized in the first stage using an object detection algorithm. In the second-stage, fingertips are detected from the cropped portion of the hand. Finally, a virtual 3D object is controlled using the coordinate positions of the fingertips. In the following subsections, the hand detection system is delineated, then the fingertip detection system is described, and finally, the interaction in the virtual environment is presented.
2.1 Detection of hand

The hand detection system is learned using the popular real-time object detection and classification algorithm, named, You Only Look Once (YOLO)\(^{33}\). The algorithm divides the entire image into a partition of \(N \times N\) cells and predicts the class label for each of the cells. The partition cell that is on the center of an object is responsible for detecting the object. Each cell predicts for five parameters. These parameters in the consecutive order are the probability of having an object in that cell \(p_c\), top-left coordinate \(h_{x_1}\) and \(h_{y_1}\), and bottom-right coordinates \(h_{x_2}\) and \(h_{y_2}\). Therefore, each grid cell in the output tensor predicts a vector \(\mathbb{H}\) of five elements given by

\[
\mathbb{H} = [p_c \ h_{x_1} \ h_{y_1} \ h_{x_2} \ h_{y_2}]^T
\]  

(1)

The aim of the proposed system is to employ the algorithm for detecting a single object, i.e., hand. The algorithm is trained by employing Darknet-19 architecture\(^{33}\). To optimize the network, the proposed loss function is given by

\[
L_1(\mathbb{H}, \hat{\mathbb{H}}) = \frac{1}{N^2M} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{M} \mathbb{H}[0]_{(ijk)} \ln \hat{\mathbb{H}}[0]_{(iijk)} + (1 - \mathbb{H}[0]_{(iijk)}) \times \ln(1 - \hat{\mathbb{H}}[0]_{(iijk)}) \\
+ \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{M} \mathbb{H}[0]_{(iijk)} \times \{ \mathbb{H}[1:4]_{(iijk)} - \hat{\mathbb{H}}[1:4]_{(iijk)} \}^2
\]  

(2)

where \(\mathbb{H}\) and \(\hat{\mathbb{H}}\) are the ground truth and predicted vector of the hand, and \(N\) and \(M\) represent the grid size and batch size, respectively. Here, the sum over each grid is taken using \(i = 1, 2, \cdots, N\) and \(j = 1, 2, \cdots, N\), and the sum over the batch is taken using \(k = 1, 2, \cdots, M\).

2.2 Detection of fingertips

The detected hand in the first stage is cropped and normalized in size, and then fed to the second stage of the CNN. This stage estimates the coordinate positions of the fingertips in the cropped image. Among the fingertips of a hand, only thumb and index ones are detected. Let the fingertip detection model predicts a vector of four elements given by

\[
\mathbb{F} = [f_{x_t} \ f_{y_t} \ f_{x_i} \ f_{y_i}]^T
\]  

(3)

where \(f_{x_t}, f_{y_t}, f_{x_i},\) and \(f_{y_i}\) are the respective coordinate positions of thumb and index fingertips. For the feature learning, four different CNN architectures, viz., VGG-16\(^{34}\), InceptionV3\(^{35}\), Xception\(^{36}\), and MobileNetV2\(^{37}\) are employed. In each case, the output of the feature learning stage is flattened to a vector. Besides, two fully connected layers (FCs) are added back-to-back at the output stage for better detection of fingertips. Each of these FCs is followed by a rectified linear unit (ReLU) activation layer and a dropout layer. At the end of this stage, an FC is added so that the feature vector size is reduced to the same as that of \(\mathbb{F}\). Finally, a sigmoid activation function is used such that the coordinate points of the fingertips remain within
the cropped hand image. The fingertip detection model directly regresses the thumb and index fingertip coordinate positions from the input image. The mean squared error (MSE) loss function defined to optimize the fingertip detection model is given by

$$L_2(F, \hat{F}) = \frac{1}{PM} \sum_{i=1}^{P} \sum_{j=1}^{M} \left\{ F_{(ij)} - \hat{F}_{(ij)} \right\}^2$$  \hspace{1cm} (4)

where $F$ and $\hat{F}$ represent the ground truth and predicted coordinates of the fingertips, and $P$ and $M$ represent the length of vector $F$ and the batch size, respectively. First, the mean over the batch is taken using $j = 1, 2, \ldots, M$ and then the mean over each element is taken using $i = 1, 2, \ldots, P$. An overview of the proposed fingertip detection system is presented in Figure 1. The activation functions and dropout layers are not shown in this figure for the sake of brevity.

![Figure 1](image.png) A block diagram representation of the fingertip detection system. The detected hand is shown in a blue bounding box and the detected thumb and index fingertips positions are shown in using green and red filled circles, respectively, in the output image.

### 2.3 Training and optimization

Commonly-referred adaptive moment estimation (ADAM) optimizer is utilized to optimize the networks for detecting hand and fingertips. This optimizer uses the moving averages of both the first and second moment of the gradient of the loss function that is given by:\[38\]

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \left( \frac{dl_q}{dw_t} \right)$$ \hspace{1cm} (5)

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left( \frac{dl_q}{dw_t} \right)^2$$ \hspace{1cm} (6)

where $q (q \in 1, 2)$ and $\beta_1$ and $\beta_2$ ($0 < \beta_1, \beta_2 < 1$) are the two hyper-parameters that control the decay rate of the moving averages and $t$ stands for a particular iteration. Finally, the update of the weights of the model is given by

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{v_t} + \epsilon} m_t$$ \hspace{1cm} (7)
where \( \eta (\eta > 0) \) is the learning rate and \( \epsilon (\epsilon > 0) \) is an infinitesimal number used for avoiding zero division error.

2.4 VR environment and interaction

A Vuforia-based interactive system is designed first, and then the coordinate positions of thumb and index fingertips are communicated in this environment and incorporated it with the fingertip detection system. The proposed VR environment requires the assistance of a marker, i.e., an image target to interact with a virtual object. The image captured in the camera is compared with the given marker image using feature matching to locate and track the virtual 3D object. To establish an interaction with the virtual object in the VR environment, the real object, i.e., hand, needs to be placed in between the camera and the image target. Later, the coordinate positions of the hand fingertips are transferred into the virtual environment with a view to an affine transformation of the virtual object. This transformation can be achieved by performing the matrix operation between the affine parameters and object coordinates.

2.4.1 Scale transformation

Let the scale transformation matrix \( \mathbf{S} \) be defined as\[^{39} \]

\[
\mathbf{S} = \begin{pmatrix}
    s_x & 0 & 0 \\
    0 & s_y & 0 \\
    0 & 0 & s_z
\end{pmatrix}
\]

where \( s_x, s_y, \) and \( s_z \) are the amount of scale transformation along the \( x, y, \) and \( z \) axis. For the scale transformation of the virtual object, the distance between the fingertips is mapped to the scale value along all three axes. At first, the distance between the thumb and index fingertips is calculated, and later, the distance maps the scale transformation using a piece-wise linear function. The Euclidean distance \( D \) between the two fingertips is estimated as

\[
D = \sqrt{(f_{x_t} - f_{x_i})^2 + (f_{y_t} - f_{y_i})^2}
\]

In order to have a better experience in the VR interaction, i.e., to eliminate the inter-personal variation, minimum and maximum thresholds of the distances of the fingertips are set. The relation between \( D \) and the amount of scale transformation \( s \ (s \in s_x, s_y, s_z) \) of the virtual 3D object is mapped using a piece-wise linear function. Let \( \tau_u \) and \( \tau_l \) be the upper and lower limits of the distance of the fingertips that correspond to limits of scale transformation \( \lambda_u \) and \( \lambda_l \), respectively. Then the piece-wise linear function can be expressed as

\[
s \ (s \in s_x, s_y, s_z) = \begin{cases}
    \lambda_u, & D > \tau_u \\
    b \times D + c, & \tau_l \leq D \leq \tau_u \\
    \lambda_l, & D < \tau_l
\end{cases}
\]

where \( b \) and \( c \) are constants.
where \( b \) and \( c \) represents the slope and vertical axis intercept of the medial part of the linear function given by

\[
\begin{align*}
    b &= \frac{\lambda_u - \lambda_l}{\tau_u - \tau_l} \\
    c &= \frac{\lambda_l \tau_u - \lambda_u \tau_l}{\tau_u - \tau_l}
\end{align*}
\] (11) (12)

The values of the \( \lambda_l \) and \( \lambda_u \), and \( \tau_l \) and \( \tau_u \) are user-dependent that can be set during the experimentation.

### 2.4.2 Rotation transformation

The rotation transformation matrix along the three axes \( \mathbf{R}_x \), \( \mathbf{R}_y \), and \( \mathbf{R}_z \) can be defined as

\[
\begin{align*}
    \mathbf{R}_x(\alpha_x) &= \begin{pmatrix} 1 & 0 & 0 \\
    0 & \cos \alpha_x & -\sin \alpha_x \\
    0 & \sin \alpha_x & \cos \alpha_x \end{pmatrix} \\
    \mathbf{R}_y(\beta_y) &= \begin{pmatrix} \cos \beta_y & 0 & \sin \beta_y \\
    0 & 1 & 0 \\
    -\sin \beta_y & 0 & \cos \beta_y \end{pmatrix} \\
    \mathbf{R}_z(\gamma_z) &= \begin{pmatrix} \cos \gamma_z & -\sin \gamma_z & 0 \\
    \sin \gamma_z & \cos \gamma_z & 0 \\
    0 & 0 & 1 \end{pmatrix}
\end{align*}
\] (13) (14) (15)

where \( \alpha_x \), \( \beta_y \), and \( \gamma_z \) are the amount of rotation about the \( x \), \( y \), and \( z \) axis. As a whole, the complete rotational transformation matrix \( \mathbf{R} \) can be expressed by

\[
\mathbf{R} = \mathbf{R}_x(\alpha_x) \mathbf{R}_y(\beta_y) \mathbf{R}_z(\gamma_z)
\]

\[
\begin{pmatrix} 
    \cos \beta_y \cos \gamma_z & -\cos \alpha_x \sin \gamma_z + \sin \alpha_x \sin \beta_y \cos \gamma_z & \sin \alpha_x \sin \gamma_z + \cos \alpha_x \sin \beta_y \cos \gamma_z \\
    \sin \alpha_x \sin \beta_y \cos \gamma_z & \cos \alpha_x \cos \gamma_z + \sin \alpha_x \sin \beta_y \sin \gamma_z & -\sin \alpha_x \cos \gamma_z + \cos \alpha_x \sin \beta_y \sin \gamma_z \\
    -\sin \beta_y & \sin \alpha_x \cos \beta_y & \cos \alpha_x \cos \beta_y \end{pmatrix}
\] (16)

To rotate the virtual object about any given axis, the angle \( \theta \) produced by the joining line of thumb and index fingertips with the axis of the 3D real world can be used. However, due to the use of a 2D camera for fingertip detection, the angle of rotation around the \( z \) axis can be calculated as

\[
\theta (\theta \in \gamma_z) = -\tan^{-1} \frac{f_x t - f_x i}{f_y t - f_y i}
\] (17)

In other words, the calculated angle is then directly mapped as \( \gamma_z \) to rotate the virtual object around the \( z \) axis.

### 2.4.3 Translation transformation
The translation transformation matrix \( T \) with the amount of translation \( t_x, t_y, \) and \( t_z \) along the three axes can be defined as\(^{[39]}\)

\[
T = \begin{pmatrix}
  t_x & t_x & t_x \\
  t_y & t_y & t_y \\
  t_z & t_z & t_z 
\end{pmatrix}
\]

(18)

It is for using the 2D camera, the amount the translation \( t_x \) and \( t_y \) along the \( x \) and \( y \) axis can be realized. In this case, the normalized center of the thumb and index fingertips is utilized for the translation of the virtual object. Similar to the scale transformation, the translational amount of each axis is mapped between the minimum and maximum units, \( t_{\text{min}} \) and \( t_{\text{max}} \), respectively, using a linear transformation. If the normalized centroid of the thumb and index fingertips is \( c_x \) and \( c_y \) then the amount of transformation \( t \) can be defined as

\[
t (t \in t_x, t_y) = (t_{\text{max}} - t_{\text{min}}) \times c (c \in c_x, c_y) + t_{\text{min}}
\]

(19)

Here, the values of the \( t_{\text{min}} \) and \( t_{\text{max}} \) are user-dependent that can be set during the experimentation.

### 2.4.4 Affine transformation

Finally, the affine transformation can be achieved by multiplying to the 3D positional matrix \( V \) of the virtual object with the scale transformation matrix \( S \) and rotation transformation matrix \( R \), and summing to the translation transformation matrix \( T \), which is given by\(^{[39]}\)

\[
V' = SRV + T
\]

(20)

where \( V' \) is the new 3D positional matrix of the virtual object after the affine transformation. The block diagram of the overall affine transformation of the virtual object using a 2D camera is shown in Figure 2. The step-by-step process of the fingertip detection system for the affine transformation of an object in the virtual environment is presented in Algorithm 1.
Figure 2 The block diagram of the overall system for the affine transformation of the virtual object using 2D localization of fingertips.

**Algorithm 1** Fingertip detection system for affine transformation of virtual objects

```plaintext
1 Import trained hand detection weights
2 Import trained fingertip detection weights
3 Initialize camera
4
5 while True do
6 capture image from the camera
7
8 if image is not captured then
9     break
10
11 bounding box = detect hand using hand detector
12
13 if hand is present == True then
14     \((h_{x1}, h_{y1}), (h_{x2}, h_{y2})\) = top-left and bottom-right coordinate of the bounding box
15     cropped image = image \([h_{y1} : h_{y2}, h_{x1} : h_{x2}]\)
16     \((\text{height, width, channel})\) = cropped image shape size cropped image by 128 x 128
17     \([f_{x1}, f_{y1}, f_{x2}, f_{y2}]\) = detect fingertip position
18
19 # transforming fingertips position to the real image
20     \(f_{xt} = f_{x1} \times \text{width} + h_{x1}\)
21     \(f_{yt} = f_{y1} \times \text{height} + h_{y1}\)
22     \(f_{xi} = f_{x1} \times \text{width} + h_{x1}\)
23     \(f_{yi} = f_{y1} \times \text{height} + h_{y1}\)
24
25 calculate distance, \(D\) according to Equation (9)
26 calculate \(s\) according to Equation (10)
27 calculate \(\theta\) according to Equation (17)
28 calculate \(t\) according to Equation (19)
29
30 apply the affine transformation using Equation (20)
```

3. Experiments and results

To substantiate the proposed fingertip-based VR interactive system, experiments are conducted. In this section, the characteristics of datasets are presented first. Then the data augmentation process of training is given. Next, the performance evaluation of the hand and fingertip detection model is presented. Finally, the
comparison of the proposed system with the existing systems and evaluation of the fingertip detection system in the case of scale, rotation, and translation transformation of virtual 3D objects are presented. The experiments are performed on a computer with Intel Core i5 4590 with 8 GB memory and Nvidia GTX1050 Ti GPU with 4 GB memory along with Logitech C270 720p HD Webcam. The codes of the hand and fingertip detection, virtual environment, and the pre-trained models are publicly released here:

https://github.com/MahmudulAlam/Fingertip-Mixed-Reality

3.1 Dataset

The South China University of Technology Egocentric Gesture (SCUT-Ego-Gesture) database\[^{16}\] is utilized first for the experimentation that has 16 different types of hand gestures captured in egocentric vision. Among them, the SingleEight gesture is considered since this gesture of hand images includes thumb and index fingers only. This SingleEight gesture dataset contains 3380 RGB hand images with only thumb and index fingers. The ground truth of a hand is a bounding box (defined by top-left and bottom-right coordinates) and that for a fingertip is the coordinate of the centroid of the fingertip. The ground truth data of the SingleEight dataset is provided by the developer of the database, i.e., the authors of the SCUT-Ego-Gesture database\[^{16}\].

Although the SingleEight gesture dataset has finger images with varying scale, background, color and size, the dataset lacks natural movements of fingers. Hence, a dataset is developed with 1000 hand images that include natural movement of thumb and index fingers. This dataset is referred to as Thumb Index 1000 (TI1K).

Overall, the TI1K dataset contains 1000 images of resolution $640 \times 480$ of both right and left hand but only one hand per image. The ground truth bounding box of the hand and the coordinates of the centroid of the thumb and index fingertips of the TI1K dataset are manually annotated and labeled by the authors of this paper. To prepare training, validation, and test data, each dataset is split into three parts and then combined respectively to create a generic dataset. Besides, all the validation and test images are horizontally flipped to synthetically generate images of the opposite hand. The total size of the dataset and the number of images used in the training, validation, and testing are provided in Table 1. A visual comparison between the images of two datasets is shown in Figure 3. In this figure the difference between TI1K and SingleEight datasets in terms of natural movements is evident. The Thumb Index 1000 (TI1K) Dataset is open-sourced and published along with the annotation which is available here:

https://github.com/MahmudulAlam/TI1K-Dataset

<table>
<thead>
<tr>
<th></th>
<th>SingleEight</th>
<th>TI1K</th>
<th>Generic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Set</td>
<td>2580</td>
<td>800</td>
<td>3380</td>
</tr>
<tr>
<td>Validation Set</td>
<td>400 + 400 (Flip)</td>
<td>100 + 100 (Flip)</td>
<td>1000</td>
</tr>
<tr>
<td>Test Set</td>
<td>400 + 400 (Flip)</td>
<td>100 + 100 (Flip)</td>
<td>1000</td>
</tr>
<tr>
<td>Total</td>
<td>4180</td>
<td>1200</td>
<td>5380</td>
</tr>
</tbody>
</table>

Table 1 Training, validation, and testing partition of the generic dataset.
Figure 3 A visual comparison between the finger images of two datasets, viz., SingleEight [16] and TIIK. (a) Straight fingers of SingleEight with no thumb and index finger movement. A varying distance of the thumb and index fingers with natural movement is seen in images of the TIIK dataset, examples of which are shown in (b), (c), and (d).

3.2 Data augmentation

To reduce the risk of overfitting of the training dataset, artificial data has been generated in the training session. In addition to the horizontal flip of the original image, we implement the on-the-fly data augmentation process. In this process, the new training dataset is generated by applying the random translation, rotation, scaling, shear, illumination variation, cropping, vertical flipping, additive Gaussian noise, and additive impulse noise. The augmented set of images has been generated randomly in each epoch of the batch processing. In this way, the trained model is learned from a gigantic dataset to ensure the generalization of each model.

3.3 Training of hand detection model

The YOLO algorithm is used to train the hand detection model using the dataset given in Table 1. First, the original input image is resized to \((224 \times 224)\) and partitioned into \((7 \times 7)\) number of grids, i.e., here \(N = 7\). The grid cell that is on the center of an object, i.e., hand is responsible for detecting the hand. Each cell predicts five parameters. Therefore, the final output tensor of the hand detection model will be of size \((7 \times 7 \times 5)\). The algorithm is trained by employing Darknet-19 architecture\(^{[33]}\) where a sigmoid activation function is employed in the final layer for normalized output. The model is trained for 200 epochs with a batch size of 32 and a learning rate of \(10^{-5}\) and later lower up to \(10^{-7}\) step by step for better convergence.

3.4 Training of fingertip detection model

The proposed CNN-based fingertip detection model is trained using the cropped and resized hand images of the size of \((128 \times 128)\) and the corresponding ground truth positions of the thumb and index fingertips. The output of this model is the vector \(F\) of length 4 that contains the \(x\) and \(y\) coordinates of the fingertips. For feature learning, four different CNN architectures: VGG-16\(^{[34]}\), InceptionV3\(^{[35]}\), Xception\(^{[36]}\), and MobileNetV2\(^{[37]}\) for experimentation. To realize the final output size of the proposed CNN model to be 4, the output of each of the feature learning stage is flattened to a vector. The output vector size of FC layers is chosen to be 1024, and the dropout rate to be 0.5. The fingertip detection model is trained for a total of 30 epochs, where the learning rate is initially chosen to be \(10^{-5}\) and later lower up to \(10^{-7}\) in a step-by-step process for better convergence. Figure 4 shows the convergence of the loss function for four feature learning
CNN architectures in the proposed fingertip detection model. In this figure, the learning curves are shown for both the training and validation stages, where the solid line indicates the training stage and the dashed line indicates the validation stage. It can be observed from the training and validation curves of this figure that the VGG-16 model is the best among the four types.

![Figure 4 Curves of loss function for four feature learning architectures of the fingertip detection model. The solid line indicates the training loss and the dashed line indicates the validation loss.](image)

### 3.5 Performance analysis for hand and fingertips detection

The performance of the proposed fingertip detection system is evaluated on the test dataset of 1000 images. At first, the proposed hand detection system that uses the YOLO algorithm is tested. In this algorithm, each image is divided into a grid of cells, and then the confidence level $p_c$ for each cell is predicted. Therefore, a confidence threshold is required. Because of the binary representation of the output, the confidence threshold is chosen to be 50%. To determine the closeness of the hand of the predicted bounding box with that of the ground truth, the intersection over union (IOU) of the bounding boxes is calculated as

$$ IOU = \frac{A_{gt} \cap A_{pr}}{A_{gt} \cup A_{pr}} $$

(21)

where $A_{gt}$ and $A_{pr}$ are the area of the ground truth and the predicted bounding boxes, respectively. If the IOU score exceeds 50%, it is considered a correct prediction. Therefore, the accuracy $A$ of the hand detection system is determined by

$$ A = \frac{\# \text{ images where } (p_c > 50\% \& IOU > 50\%)}{\# \text{ images in test set}} $$

(22)
Table 2 shows the accuracy and mean execution time of the proposed hand detection system for both the individual and generic datasets. It is seen from this table that the accuracy of the proposed hand detection system is above 92%.

Table 2 Performance metrics of the proposed hand detection system in terms of accuracy and mean execution time

<table>
<thead>
<tr>
<th>Ground Truths</th>
<th>SingleEight</th>
<th>TI1K</th>
<th>Generic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated</td>
<td>800</td>
<td>200</td>
<td>1000</td>
</tr>
<tr>
<td>Accuracy</td>
<td>749</td>
<td>172</td>
<td>921</td>
</tr>
<tr>
<td>Execution Time</td>
<td>20.22 ms</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The coordinates of thumb and index fingertips are predicted using the proposed CNN model by using the detected hands. The mean absolute error (MAE) of coordinates between the ground truth and the predicted positions of the fingertips in pixels (px) units is estimated as

\[ \mathcal{E} = \frac{1}{N} \sum_{j=1}^{N} | \mathcal{F}_j - \tilde{\mathcal{F}}_j | \]  

where \( N \) represents the total number of detected hands, \( \mathcal{F}_j \) and \( \tilde{\mathcal{F}}_j \) represent the ground truth and the predicted coordinates of the fingertips in the \( j \)th hand, respectively. Here, the mean over the detected hands is taken using \( j = 1, 2, \ldots, N \).

Table 3 Performance metrics of prediction of coordinate positions of fingertips in terms of MAE for different feature learning models

<table>
<thead>
<tr>
<th>CNN Architecture for Feature Learning</th>
<th>Resolution Pixels (px)</th>
<th>MAE</th>
<th>Execution Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(</td>
<td>f_x - \tilde{f}_x</td>
</tr>
<tr>
<td>VGG-16</td>
<td>640 × 480</td>
<td>4.5462</td>
<td>4.4525</td>
</tr>
<tr>
<td></td>
<td>320 × 240</td>
<td>2.2646</td>
<td>2.3841</td>
</tr>
<tr>
<td></td>
<td>320 × 240</td>
<td>3.4781</td>
<td>3.8757</td>
</tr>
<tr>
<td></td>
<td>320 × 240</td>
<td>6.3769</td>
<td>7.6442</td>
</tr>
<tr>
<td>MobileNetV2</td>
<td>640 × 480</td>
<td>7.0558</td>
<td>7.4055</td>
</tr>
<tr>
<td></td>
<td>320 × 240</td>
<td>3.3906</td>
<td>3.7996</td>
</tr>
</tbody>
</table>

Table 3 shows the MAE of the estimated coordinates of the fingertips in two different resolutions, (640 × 480) and (320 × 240), for four feature learning networks employed in the proposed fingertip detection model. Besides, the mean execution time of the fingertip detection for each feature learning network are also reported. As the images are always converted to a size of (128 × 128) for fingertip detection, the
execution time of each network is independent of the resolution of the input image. It is seen from Table 3 that the MAE of the distance between thumb and index fingertips is minimum for VGG-16 architecture. It is also inferred from the results presented in the table that the MAE decreases with lowering the resolution of the input image. Since VGG-16 network architecture performs the best, in the rest of the paper, the results of the fingertip detection will be reported only for the VGG-16 architecture. In other words, the CNN-based fingertip detection system with the VGG-16 network in the feature learning stage will be referred to as the Proposed Method.

### 3.6 Comparison with existing systems

The proposed fingertip detection system is compared with two existing CNN-based fingertip detection systems, namely, DeepFinger\[14\] and ‘Dual Target Fingertip Detection’ (DTFD)\[18\]. In the following subsections, the performance comparison of fingertip detection is carried out on the test set of 2D images at first, and then the performance of the interaction of the detected fingertips with the object in the VR environment is evaluated.

#### 3.6.1 Performance of predicting coordinate positions

To have a fair judgment comparing the fingertip detection systems, namely, the DeepFinger\[14\], DTFD\[18\], and the proposed method are implemented on the same training dataset. While comparing these algorithms, YOLO is used in the first stage for the detection of hand for all three systems. The results on the test dataset in terms of the MAE and execution time per frame to predict the coordinate positions of the three methods are reported in Table 4. From this table, it is seen that the proposed fingertip detection method provides 20 – 25% less MAE error as compared to the other methods. However, it requires approximately 10 ms more time than that of the other methods because the proposed method uses a much higher number of parameters in its model to reduce the MAE error. Even though the proposed method requires higher execution time than others, it is still much less than the minimum real-time requirement which is 33.33 ms (30 FPS video). Therefore, the proposed method outperforms the two existing methods by a large margin of MAE at the expense of a comparable amount of computational time.

**Table 4 Performance metrics of prediction of coordinate positions of fingertips to compare the proposed and existing methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Trainable Parameters</th>
<th>MAE</th>
<th>Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed</td>
<td>24,158,020</td>
<td>4.5462, 4.4525, 4.2661, 5.4328, 6.4022</td>
<td>12.60 ms</td>
</tr>
<tr>
<td>DeepFinger</td>
<td>1,519,908</td>
<td>18.0483, 17.8458, 14.1705, 14.7952, 26.1425</td>
<td>2.84 ms</td>
</tr>
<tr>
<td>DTFD</td>
<td>568,132</td>
<td>21.5664, 20.8403, 16.1194, 14.8466, 30.35</td>
<td>2.22 ms</td>
</tr>
</tbody>
</table>

#### 3.6.2. Performance of interaction in VR environment

To evaluate the performance of interaction using the detected fingertips, the affine transformation of a number of virtual 3D objects is realized using the hands of different participants in the Unity platform. In the
experiments, it is ensured that the hands have varying views, size, shape, and skin color of fingers. For instance, the performance of the interaction of both the left and right hands of a participant is evaluated. In the experiments, representative results of 12 participants are shown. Out of them, 4 subjects evaluated the performance of scale transformation, 4 evaluated the rotation transformation, and 4 evaluated the translation transformation. In other words, the subjects are mutually exclusive in terms of their participation. Based on the preference of the subjects for scale, rotation, or translation transformation, they were instructed to do specific finger gestures in front of a camera and were told to keep their hand within the frame which is recorded for objective evaluations. For each type of transformation, each participant used one of the four virtual objects, namely, Helicopter, Earth, Ship, and Tree. A view of these virtual 3D objects is shown in Figure 5. It is noted here that the wings of the Helicopter have rotational motion and the rest of them are static.

![Figure 5](image)

**Figure 5** A view of the experimental 3D objects in the virtual environment. The objects are (a) Helicopter, (b) Earth, (c) Ship, and (d) Tree.

The movements of the hands of participants are captured using a CCD video camera with a framerate of 10 fps and a resolution of 640×480. The ground truth coordinates of the bounding box of the hand and the coordinates of the centroid of the thumb and index fingertips in each frame of a video of a participant are manually annotated and labeled by the authors of this paper. Using the ground truth coordinates of the fingertips, the ground truth values of affine parameters are calculated using (9), (10), (17), and (19). The proposed system predicts the coordinates of the fingertips of thumb and index fingers of each frame of a video clip for 5s. From the predicted coordinates, the Euclidean distance $D$ between the fingertips, the angle $\theta$ created between the vertical axis of the frame and the joining line of the fingertips, and the center coordinate of the joining line $(c_x, c_y)$ are calculated. Using the value of $D$, the amount of scale $s$ along $x$, $y$, and $z$ axis of the virtual 3D object is calculated according to the piece-wise linear function given in (10). The angle $\theta$ is directly utilized as $\gamma_z$ for rotation about the $z$ axis, and the center coordinates $(c_x, c_y)$ is used to translate the virtual object along $x$ and $y$ axis.

The thresholds $\tau_l$ and $\tau_u$ of the linear function given in (10) are chosen to be 100 pixels and 180 pixels, respectively, by considering that in the experiments the average of the distance between the fingertips is 140 pixels. In the experiments, the threshold values $\lambda_l$ and $\lambda_u$ are chosen as 0.05 and 0.20 to calculate scale transformation. In the case of rotational transform, the virtual object is rotated about the $z$ axis according to $\theta$ in the range of $-180^\circ$ to $180^\circ$. For the translational transform, $c_x$ is used to translate the object along $x$ axis and $t_{\text{max}}$ and $t_{\text{min}}$ are chosen to be 1 and $-1$ unit, respectively. Similarly, $c_y$ is used to translate the object along $y$ axis and $t_{\text{max}}$ and $t_{\text{min}}$ are chosen to be 0.5 and $-0.5$ unit, respectively. Tables 5, 6, and 7 show the interaction performance in terms of MAE and Pearson’s correlation coefficient between the
predicted and ground truth values of the variation of affine parameters, namely, scaling, rotational, and translational parameters of the virtual 3D objects, respectively. The MAEs and correlation coefficients are estimated from the coordinates of fingertips that are predicted for the video frames using the Proposed Method, DeepFinger \cite{14}, and DTFD \cite{18}. In these tables, the variations of affine parameters of the ground truth values of each participant are reported in terms of mean \( \mu \) (\( \mu \in \mu_s, \mu_\gamma, \mu_t \)) and standard deviation \( \sigma \) (\( \sigma \in \sigma_s, \sigma_\gamma, \sigma_t \)).

**Table 5 Performance metrics of interaction in terms of MAE and correlation coefficient due to scale variation of 3D virtual objects using fingertips**

<table>
<thead>
<tr>
<th>Participant</th>
<th>Object</th>
<th>Hand</th>
<th>System</th>
<th>MAE</th>
<th>( D - \hat{D} ) (px)</th>
<th>( s - \hat{s} ) (unit)</th>
<th>( P_D )</th>
<th>( P_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID: 01 ( \mu_s: 0.0719 ) ( \sigma_s: 0.0470 )</td>
<td>Helicopter</td>
<td>Left</td>
<td>Proposed Method</td>
<td>4.2670</td>
<td>1.22e-03</td>
<td>0.9985</td>
<td>0.9985</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeepFinger</td>
<td>14.0256</td>
<td>2.09e-03</td>
<td>0.9908</td>
<td>0.9955</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DTFD</td>
<td>11.7237</td>
<td>5.69e-03</td>
<td>0.9806</td>
<td>0.9909</td>
<td></td>
</tr>
<tr>
<td>ID: 02 ( \mu_s: 0.1369 ) ( \sigma_s: 0.0722 )</td>
<td>Earth</td>
<td>Left</td>
<td>Proposed Method</td>
<td>8.7988</td>
<td>1.16e-03</td>
<td>0.9967</td>
<td>0.9976</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeepFinger</td>
<td>62.8313</td>
<td>2.42e-02</td>
<td>0.915</td>
<td>0.9208</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DTFD</td>
<td>29.7889</td>
<td>8.91e-03</td>
<td>0.9716</td>
<td>0.9432</td>
<td></td>
</tr>
<tr>
<td>ID: 03 ( \mu_s: 0.0989 ) ( \sigma_s: 0.0570 )</td>
<td>Ship</td>
<td>Right</td>
<td>Proposed Method</td>
<td>20.1374</td>
<td>1.20e-02</td>
<td>0.9822</td>
<td>0.9648</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeepFinger</td>
<td>26.576</td>
<td>3.59e-02</td>
<td>0.9139</td>
<td>0.7604</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DTFD</td>
<td>24.7794</td>
<td>3.08e-02</td>
<td>0.9218</td>
<td>0.8609</td>
<td></td>
</tr>
<tr>
<td>ID: 04 ( \mu_s: 0.1439 ) ( \sigma_s: 0.0682 )</td>
<td>Tree</td>
<td>Right</td>
<td>Proposed Method</td>
<td>8.5641</td>
<td>4.97e-04</td>
<td>0.9958</td>
<td>0.9998</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeepFinger</td>
<td>24.4022</td>
<td>2.74e-03</td>
<td>0.9742</td>
<td>0.9928</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DTFD</td>
<td>18.1808</td>
<td>4.08e-03</td>
<td>0.9727</td>
<td>0.9893</td>
<td></td>
</tr>
</tbody>
</table>

**Table 6 Performance metrics of interaction in terms of MAE and correlation coefficient due to rotational variation of 3D virtual objects using fingertips**

<table>
<thead>
<tr>
<th>Participant</th>
<th>Object</th>
<th>Hand</th>
<th>System</th>
<th>MAE</th>
<th>( \gamma_z - \hat{\gamma}_z ) (degree)</th>
<th>( P_{\gamma z} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID: 05 ( \mu_\gamma: -13.7225^\circ ) ( \sigma_\gamma: 14.5027^\circ )</td>
<td>Helicopter</td>
<td>Left</td>
<td>Proposed Method</td>
<td>4.8870</td>
<td>0.9907</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeepFinger</td>
<td>5.2581</td>
<td>0.9323</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DTFD</td>
<td>8.9171</td>
<td>0.8538</td>
<td></td>
</tr>
<tr>
<td>ID: 06 ( \mu_\gamma: -18.7532^\circ ) ( \sigma_\gamma: 22.4175^\circ )</td>
<td>Earth</td>
<td>Left</td>
<td>Proposed Method</td>
<td>3.4782</td>
<td>0.9909</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeepFinger</td>
<td>9.3307</td>
<td>0.9221</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DTFD</td>
<td>8.4961</td>
<td>0.8778</td>
<td></td>
</tr>
<tr>
<td>ID: 07 ( \mu_\gamma: 17.6527^\circ ) ( \sigma_\gamma: 7.4468^\circ )</td>
<td>Ship</td>
<td>Right</td>
<td>Proposed Method</td>
<td>7.5618</td>
<td>0.7187</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeepFinger</td>
<td>17.2435</td>
<td>-0.0900</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DTFD</td>
<td>15.0343</td>
<td>-0.5669</td>
<td></td>
</tr>
<tr>
<td>ID: 08 ( \mu_\gamma: 27.5098^\circ ) ( \sigma_\gamma: 12.0430^\circ )</td>
<td>Tree</td>
<td>Right</td>
<td>Proposed Method</td>
<td>5.5763</td>
<td>0.8552</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DeepFinger</td>
<td>23.9389</td>
<td>0.1159</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DTFD</td>
<td>16.5958</td>
<td>0.1222</td>
<td></td>
</tr>
</tbody>
</table>
Table 7 Performance metrics of interaction in terms of MAE and correlation coefficient due to translational variation of 3D virtual objects using fingertips

| Participant | Object  | Hand | System    | MAE $\frac{|t_y - \hat{t}_y|}{(\text{unit})}$ | $\frac{|t_x - \hat{t}_x|}{(\text{unit})}$ | $\mathcal{P}_{ty}$ | $\mathcal{P}_{tx}$ |
|-------------|---------|------|-----------|---------------------------------------------|---------------------------------------------|--------------------|--------------------|
| ID:09       | Helicopter | Left | Proposed Method DeepFinger DTFD | 0.0086 | 0.1477 | 0.9993 | 0.9906 |
| 𝜇: -0.1286 & -0.0158 unit 𝜎: 0.0817 & 0.2943 unit | | | | | | | |
| ID:10       | Earth    | Left | Proposed Method DeepFinger DTFD | 0.0121 | 0.1103 | 0.9993 | 0.9776 |
| 𝜇: -0.0641 & 0.1821 unit 𝜎: 0.0574 & 0.3324 unit | | | | | | | |
| ID:11       | Ship     | Right| Proposed Method DeepFinger DTFD | 0.0234 | 0.0411 | 0.9964 | 0.9733 |
| 𝜇: -0.0990 & -0.1563 unit 𝜎: 0.0614 & 0.2591 unit | | | | | | | |
| ID:12       | Tree     | Right| Proposed Method DeepFinger DTFD | 0.0176 | 0.0662 | 0.9980 | 0.9976 |
| 𝜇: -0.0428 & 0.0759 unit 𝜎: 0.2036 & 0.2882 unit | | | | | | | |

Table 5 shows the performance of scale variation of the fingertip based interactive system where the MAE and correlation coefficient of scale parameters and the Euclidian distance between the ground truth and predicted distance of the fingertips are reported. The correlation coefficient of the distances and the scale parameters are denoted as $\mathcal{P}_D$ and $\mathcal{P}_S$, respectively. For all four participants, the proposed method provides the least MAE error both in distance between the fingertips and the scale parameters. Moreover, the proposed method achieves the highest correlation values as compared to the other methods. Table 6 shows the performance of rotation transformation in terms of MAE and the correlation coefficient $\mathcal{P}_{\gamma_z}$ of the rotation parameter around $z$-axis. From this table, it can be observed that the proposed method achieves the least MAE of angle about $z$-axis in degrees, and the minimum value is found to be $3.4782^\circ$. Likewise, the correlation coefficient value achieved by the proposed method is higher than that of the other two methods, and the highest value is found to be $\mathcal{P}_{\gamma_z} = 0.9909$. Similarly, Table 7 shows the MAE and correlation coefficients $\mathcal{P}_{tx}$ and $\mathcal{P}_{ty}$ of the translation parameters in $x$ and $y$ axis, respectively. Among the comparing methods, the proposed method provides much less translation error as compared to the others and the minimum MAEs for translation along the $x$ and $y$ axis are found to be 0.0086 and 0.0081, respectively. Besides, the highest correlation coefficients values attained by the proposed method are found to be $\mathcal{P}_{tx} = 0.9976$ and $\mathcal{P}_{ty} = 0.9993$ along the $x$ and $y$ axis, respectively.

The values $\mu$ and $\sigma$ in Tables 5, 6, and 7 reveal that a diverse set of affine transformations such as scale up or down, clockwise or anti-clockwise rotation, and left or right side translations of the 3D virtual objects are considered in the experiments. It is a common observation from all these tables that the proposed system provides the least MAE and the highest correlation coefficient for variations of each of the affine parameters and for all virtual objects when compared with that of the existing systems, viz., DeepFinger$^{14}$, DTFD$^{18}$. 
For the scale and translation transformations, the proposed system provides a correlation coefficient of at least 0.96 and 0.97, respectively. The challenging scenario is observed for the rotational transformation in the objects *Ship* and *Tree*. But in this case, also the proposed system is ensured to provide a positive and highest correlation coefficient among the comparing systems.

Figures 6, 7, and 8 show the frame-wise variations of the ground truths and estimated values those associated with three kinds of affine transformation, namely, scaling, rotation around z axis, and translation along x and y axis, respectively. These interactions with the virtual 3D objects are evaluated for three experimental systems, namely, DeepFinger[14], DTFD[18], and the proposed method. It is seen from these figures that the ground truths and the values estimated by the proposed system are almost overlapping each other, whereas the estimated values of the existing systems deviate significantly from the ground truths. Thus, the performance of the proposed method is robust in terms of virtual interaction.

**Figure 6** Frame-wise variations of scale parameter to compare the ground truth with the estimation of the experimental systems. The parameters are: $\tau^s_l = 100$ pixels, $\tau^s_u = 180$ pixels, $\lambda^s_l = 0.05$, and $\lambda^s_u = 0.20$.

**Figure 7** Frame-wise variations of rotational parameter around z-axis to compare the ground truth with the estimation of the experimental systems. The range of rotation is between $-180^\circ$ to $180^\circ$. 
Figure 8 Frame-wise variation of translational parameters along $x$ and $y$ axis to compare the ground truth with the estimation of the experimental systems. Here, $t_{\text{min}}$ and $t_{\text{max}}$ are set as $-1.0$ and $1.0$ (unit) along $x$ axis and $-0.5$ and $0.5$ (unit) along $y$ axis, respectively.

To evaluate the performance of real-life difficult scenarios such as scene clutter, occlusion, and illumination effects are artificially included in the interaction. For example, to represent scene clutter, salt noise with a probability of 0.10 is randomly added. Similarly, for simulating occlusion, a coarse dropout of a block of pixels, each having a size 2% of the image size is randomly applied. For the illumination effect, the brightness of the image is randomly changed between 50 – 100%. Typical images of these effects are shown in Figure 9. As the performance of interaction depends on the accuracy of detecting the coordinates of fingertips by the comparing methods, the performance of the methods in these difficult scenarios is objectively evaluated by using the ground-truth coordinates of the 12 users. Table 8 shows the performance of fingertips coordinate prediction by the methods in the presence of scene clutter, occlusion, and illumination variation in terms of MAE and correlation coefficient of the coordinates of the fingertips.
Figure 9 Typical images of real-life difficult scenarios during the interaction with a virtual object. (a) Original image. (b) Scene clutter represented by salt noise with a probability of 0.10. (c) Occlusion represented by random coarse dropout with block of pixels each having a size 2% of the total image. (d) Illumination effects represented by a random variation of brightness between 50 - 100%.

Table 8 Performance evaluation of the interaction in difficult real-life scenarios in terms of MAE and correlation coefficient of fingertip coordinates

| Scenario       | Method     | $|f_{x_t} - \hat{f}_{x_t}|$ (px) | $|f_{y_t} - \hat{f}_{y_t}|$ (px) | $|f_{x_i} - \hat{f}_{x_i}|$ (px) | $|f_{y_i} - \hat{f}_{y_i}|$ (px) | $\mathcal{P}_{x_t}$ | $\mathcal{P}_{y_t}$ | $\mathcal{P}_{x_i}$ | $\mathcal{P}_{y_i}$ |
|----------------|------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|-------------------|-------------------|-------------------|-------------------|
| Scene Clutter  | Proposed Method | 18.78 ± 20.40 | 17.76 ± 15.09 | 24.56 ± 22.36 | 25.30 ± 18.43 | 0.88 ± 0.18 | 0.78 ± 0.22 | 0.77 ± 0.21 | 0.72 ± 0.25 |
|                | DeepFinger | 31.24 ± 16.88 | 30.16 ± 14.83 | 33.68 ± 13.30 | 33.89 ± 19.18 | 0.66 ± 0.32 | 0.70 ± 0.15 | 0.62 ± 0.15 | 0.71 ± 0.22 |
|                | DTFD      | 35.52 ± 21.78 | 31.12 ± 21.34 | 37.46 ± 18.87 | 35.42 ± 21.28 | 0.71 ± 0.33 | 0.67 ± 0.24 | 0.62 ± 0.24 | 0.72 ± 0.26 |
| Occlusion      | Proposed Method | 10.00 ± 4.51 | 10.09 ± 7.20 | 11.95 ± 9.18 | 11.65 ± 6.37 | 0.92 ± 0.11 | 0.91 ± 0.09 | 0.87 ± 0.09 | 0.87 ± 0.16 |
|                | DeepFinger | 25.42 ± 11.82 | 28.53 ± 14.37 | 25.98 ± 15.47 | 24.00 ± 9.98 | 0.72 ± 0.34 | 0.79 ± 0.13 | 0.71 ± 0.13 | 0.84 ± 0.11 |
|                | DTFD      | 30.72 ± 13.24 | 29.12 ± 18.01 | 29.69 ± 10.82 | 23.02 ± 10.70 | 0.80 ± 0.28 | 0.74 ± 0.21 | 0.68 ± 0.21 | 0.85 ± 0.15 |
| Illumination   | Proposed Method | 9.50 ± 5.22 | 9.18 ± 7.54 | 12.47 ± 9.72 | 10.50 ± 6.56 | 0.96 ± 0.03 | 0.94 ± 0.07 | 0.90 ± 0.07 | 0.88 ± 0.15 |
|                | DeepFinger | 23.87 ± 11.84 | 24.25 ± 12.48 | 24.86 ± 15.34 | 20.44 ± 9.38 | 0.80 ± 0.29 | 0.85 ± 0.13 | 0.75 ± 0.13 | 0.88 ± 0.13 |
|                | DTFD      | 28.89 ± 12.83 | 27.15 ± 16.28 | 27.85 ± 11.36 | 22.12 ± 12.16 | 0.83 ± 0.34 | 0.80 ± 0.20 | 0.69 ± 0.20 | 0.85 ± 0.18 |
Here, $f_{x_t}, f_{y_t}, f_{x_i},$ and $f_{y_i}$ represent the ground truth x- and y-coordinate position of the thumb and index fingertips and $\mathcal{P}_{x_t}, \mathcal{P}_{y_t}, \mathcal{P}_{x_i},$ and $\mathcal{P}_{y_i}$ are the correlation coefficient between the ground truth and predicted values. The results for each method in each scenario shown in Table 8 represent the mean and standard deviation of the MAE and correlation coefficient of the 12 users. It is seen from the table that the proposed method performed the best among the comparing methods both in terms of mean values of MAE and correlation coefficient of coordinates. It is also seen from this table that in the case of occlusion and illumination, the proposed method provides the minimum standard deviation as compared to the other methods. But in the case of scene clutter, the proposed method and DeepFinger show a comparative performance in terms of the standard deviation. Thus, as per the results of the average value and standard deviation of the MAE and correlation coefficient, the overall performance of the proposed method is the best in terms of accuracy and robustness.

Figure 10 shows a visual output of the proposed thumb and index fingertip-based interaction system using the virtual object *Helicopter*. This figure shows the output of the virtual object when it is scaled up or down, rotated clockwise or counter-clockwise, and translated in the left or right directions. Finally, the output is shown when all kinds of affine transformations are applied to the virtual object at the same instance. Similar results are also obtained for other virtual objects but are not shown for the sake of brevity. From Figure 10 it is evident that the proposed fingertip-based interaction system is capable of transforming a virtual object using the affine parameters. A real-time demo of the overall affine transformation of a virtual object using the proposed system is available here:

https://github.com/MahmudulAlam/Fingertip-Mixed-Reality/blob/master/README.md
(a) Initial State: 0.10 scaling, 0° rotation, and 0.0-unit translation along x, y, and z axis

(b) Scale Up: 0.20 scaling along x, y, and z axis

(c) Scale Down: 0.05 scaling along x, y, and z axis

(d) Clockwise Rotation: 52.31° rotation about the z-axis

(e) Counter clockwise Rotation: -41.32° rotation about the z axis

(f) Translation: -0.75- and 0.22-unit translation along x and y axis, respectively

(g) Translation: 0.71- and 0.11-unit translation along x and y axis, respectively

(h) Scale, Rotation, and Translation: 0.20 scaling, 29.80° clockwise rotation, and 0.64- and 0.18-unit translation along x and y axis, respectively

Figure 10 Visual output of the proposed fingertip-based interaction system using the virtual object Helicopter. The initial state is shown in (a). Scale-up and down along all three axes are shown in (b) and (c). Clockwise and counter-clockwise rotations are shown in (d) and (e). Translation in left and right directions are shown in (f) and (g). Finally, the combination of all types of affine transformation is shown in (h).
4. Conclusion

In this paper, a fingertip-based interaction system for affine transformation of a virtual 3D object has been presented. In this system, a thumb and index fingers have acted as the medium of interaction in the VR environment. The well-known YOLO algorithm for object detection has been used first to detect the hand. The coordinate positions of thumb and index fingertips have been estimated from the detected hand by using the proposed CNN-based algorithm. From the coordinate positions, the distance between the fingertips has been calculated and mapped through a piecewise linear function to scale the virtual object. The angle created by the joining line of fingertips with the vertical axis of the 2D image has been calculated and applied to rotate the virtual object. To translate the object, the shift of the center point of the joining line of fingertips from that of the 2D image has been used. To evaluate the performance of predicting coordinates of the fingertips, a new database referred to as TI1K has been developed and publicly released. This database along with the commonly referred SCUT-Ego-Gesture database has been used to carry out the experiments. In comparison, the proposed system has outperformed the existing systems in terms of the estimation error of the position of fingertips. In particular, the accuracy of the detection of hand has been found to be at least 92.1%, and that of the prediction of coordinates of fingertips to be within 3 pixels at a low resolution and 6 pixels at a higher resolution. In the experiments of evaluating the performance of interaction with the VR environment, the proposed system has achieved the lowest MAE and the highest correlation between the ground truth and estimated values of the affine parameters as compared to the existing systems. The real-life difficult scenarios such as scene clutter, occlusion, and illumination variation are also included in the experiments. The results of these experiments reveal that the proposed method not only provides a high level of accuracy during an interaction but also it shows a higher level of robustness in these difficult scenarios. In conclusion, the proposed autonomous fingertip-based VR interaction system can play a significant role in the fourth industrial revolution.

5. CRediT authorship contribution statement

Mohammad Mahmudul Alam: Conceptualization, Software, Formal analysis, Data curation, Writing - Original Draft, Writing - Review & Editing. S. M. Mahbubur Rahman: Conceptualization, Formal analysis, Writing - Original Draft, Writing - Review & Editing, Supervision.

6. Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments, which have been useful in improving the quality of the paper.
References


7 Han J, Shao L, Xu D, Shotton J. Enhanced computer vision with Microsoft Kinect sensor: A review. IEEE transactions on cybernetics, 2013, 43 (5): 1318–1334


