applied sciences

MDPI

Article

Augmented Reality Assisted Assembly Training Oriented Dynamic Gesture Recognition and Prediction

Jiaqi Dong [1], Zeyang Xia [2,*] and Qunfei Zhao [1]

[1] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China; saberfate@sjtu.edu.cn (J.D.); zhaoqf@sjtu.edu.cn (Q.Z.)
[2] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
* Correspondence: zy.xia@siat.ac.cn

Abstract: Augmented reality assisted assembly training (ARAAT) is an effective and affordable technique for labor training in the automobile and electronic industry. In general, most tasks of ARAAT are conducted by real-time hand operations. In this paper, we propose an algorithm of dynamic gesture recognition and prediction that aims to evaluate the standard and achievement of the hand operations for a given task in ARAAT. We consider that the given task can be decomposed into a series of hand operations and furthermore each hand operation into several continuous actions. Then, each action is related with a standard gesture based on the practical assembly task such that the standard and achievement of the actions included in the operations can be identified and predicted by the sequences of gestures instead of the performance throughout the whole task. Based on the practical industrial assembly, we specified five typical tasks, three typical operations, and six standard actions. We used Zernike moments combined histogram of oriented gradient and linear interpolation motion trajectories to represent 2D static and 3D dynamic features of standard gestures, respectively, and chose the directional pulse-coupled neural network as the classifier to recognize the gestures. In addition, we defined an action unit to reduce the dimensions of features and computational cost. During gesture recognition, we optimized the gesture boundaries iteratively by calculating the score probability density distribution to reduce interferences of invalid gestures and improve precision. The proposed algorithm was evaluated on four datasets and proved to increase recognition accuracy and reduce the computational cost from the experimental results.

Keywords: augmented reality assisted assembly training; human-machine interaction; gesture recognition and prediction

Citation: Dong, J.; Xia, Z.; Zhao, Q. Augmented Reality Assisted Assembly Training Oriented Dynamic Gesture Recognition and Prediction. *Appl. Sci.* **2021**, *11*, 9789. https://doi.org/10.3390/app11219789

Academic Editor: Jiro Tanaka

Received: 21 September 2021
Accepted: 14 October 2021
Published: 20 October 2021

1. Introduction

Industrial assembly is performed by grouping individual parts and fitting them together to create the finished commodities with great additional value. Thus, assembly is an important step to connect the manufacturing processes and the business processes. In assembly, training is significant for technicians to improve the skills. Effective assembly training can increase the efficiency and quality of assembly tasks to achieve more value. Therefore, many businesses and researches have paid attention to assembly training [1]. In traditional assembly training, trainees need repeated practice to improve assembly skills, which leads to high resource consumption. Furthermore, it is not easy to evaluate the standard and achievement during the traditional assembly training operations. Nowadays, these problems can be addressed by augmented reality (AR) technology.

AR is a novel human computer interaction (HCI) technique. AR can enable users to experience the real world in which virtual objects and real objects coexist, and interact with them in the real time. In the past two decades, AR application has been a trending research topic in many areas, such as education, entertainment, medicine, and industry [2,3]. Volkswagen intended to use AR to compare the calculated crash test imagery with the

actual case [4]. Fuchs et al. [5] developed an optical see-through augmentation for laparoscopic surgery which could simulate the view of the laparoscopes from small incisions. Pokémon GO [6] enabled users to capture and battle with different virtual Pokémon in a real environment by mobile phone. Liarokapis et al. [7] employed a screen-based AR to assist engineering education based on the Construct3D tool. AR is capable of assisting assembly training because of its low time costs and high effectiveness, which allows trainees to conduct real-time assembly training tasks at any place or time with minimum cost [8–11]. Furthermore, in an augmented environment, trainees can analyze the behaviors and achievements according to virtual information and real feedback. From AR interaction, trainees can obtain intuitive data to standardize their training operations.

Currently there are no commercial products for AR assisted assembly training (ARAAT), thus many research works have focused on it. Early ARAAT studies mainly converged on marker-based tools or gloves. Wang et al. [12] established an AR assembly workspace which enabled trainees to assemble virtual objects by real marked tools. Valentini [13] developed a system which allowed trainees to assemble the virtual components using a glove with sensors. These methods can be accurate and intuitive, but come with the high cost of the devices. Recently, benefiting from the rapid development of computer vision, researchers are increasingly focused on the vision-based bare-hand ARAAT which is natural and intuitive with the low costs of the vision cameras [14–16]. Most ARAAT tasks are conducted by real-time hand operations. Trainees need to use their real hands to operate virtual workpieces. Thus, precise gesture recognition plays an important role in the bare-hand ARAAT, and also in evaluating the standard and achievement of training tasks. Lee et al. [17] applied hand orientation estimation and collision recognition from trainees' hands to virtual substances. They proposed a hand interaction technique that ensured a seamless experience for trainees in the AR environment. Nevertheless, the precision of Lee's research depended on the range between trainees' hands and stereo cameras. Thus, calculation errors existed when only one finger was used and its application was limited. Wang [18] proposed a Restricted Coulomb Energy network to segment hands for AR empty-hand assembly. Virtual objects were controlled by two fingertips in the experiment to simulate assembly tasks. Since algorithms of fingertip tracking were implemented in 2D space without depth information, the results had lower recognition accuracy. Most current studies on the bare-hand ARAAT have received a low recognition accuracy. Hence, more effort has been made to raise the recognition precision. Choi [19] developed a hand-based AR mobile phone interface by executing the "grasping" and "releasing" gestures with virtual substances. The interface provided a natural interaction benefit from the hand detection, palm pose estimation, and finger gesture recognition. Figueiredo et al. [20] evaluated interactions on tabletop applications with virtual objects by hand tracking and gesture recognition. During the interaction, they applied the "grasping" and "releasing" gestures and used the Kinect device for hand tracking. These studies have increased the recognition rate by various image processing methods, but the interaction gestures are confined to few types of interaction gestures. Even though for the up-to-date AR device HoloLens (1st Generation) [21] that is broadly used in numerous AR applications, the operation gestures are also limited to only two gestures: "pointing" and "blooming". Limited types of interaction gestures are not only inadequate for practical industrial assembly tasks, but also giving unnatural experiences to trainees. In ARAAT, giving a realistic and natural experience in performing assembly tasks is also a significant issue as well as precise gesture recognition [22–24]. Aside from inadequate gestures, a long response time will also bring an unnatural interaction experience in ARAAT. Thus, many researches have focused on early recognition by predicting or estimating gestures to reduce the response time and make the process of assembly operations appear natural. Zhu et al. [25] proposed a progressive filtering approach to predict ongoing human tasks to ensure a natural and friendly interaction. Du et al. [26] predicted gestures using improved particle filters to accomplish the tasks of welding, painting, and stamping. With the help of additional physical properties of 3D virtual objects, Imbert et al. [27] found a more natural approach to doing assembly

tasks and the results showed that trainees could perform assembly tasks easily. There is a problem that current studies for bare-hand ARAAT mostly focus on the single gesture recognition rather than the whole assembly task evaluation. Compared with single gesture recognition, the whole task evaluation can analyze the trainees' operations overall and is more helpful to trainees for improving their standard and achievement of operations. However, the whole task evaluation remains a challenge because the whole assembly task contains many different gestures together which are difficult to be distinguished. Directly evaluating the performance of a whole ARAAT task is a complicated process because only limited types of gestures for ARAAT can be recognized and recognition accuracy is not high.

Based on the related ARAAT studies mentioned thus far, ARAAT has the following areas of improvement: (1) lack of the whole complex assembly task evaluation, (2) limitation of interaction gestures, (3) low recognition accuracy, and (4) unnatural interaction experiences resulting from long response time. With the aim of resolving these problems, in this paper, we develop an ARAAT system. The flowchart of ARAAT is shown in Figure 1. Trainees choose the ARAAT tasks and conduct the corresponding operations. The AR device (HoloLens) records trainees' gesture videos during the tasks and the multimodal features are extracted from the videos. After classification, these multimodal features are used for gesture segmentation and optimization. After recognition with the optimal gesture boundaries, the gesture results will be used to evaluate the standard and achievement of hand operations in ARAAT tasks. In ARAAT, we have made the following contributions: (1) Building a model for the whole complex assembly task evaluation. We decompose an ARAAT task into a series of hand operations. Each hand operation is further decomposed into several continuous actions. Each action can be considered as an identifiable gesture. Using the classification and sequences of gestures, we can easily distinguish actions and predict operations to evaluate the performance of ARAAT tasks. (2) Increasing the types of interaction gestures. We generalize three typical operations and six standard actions based on practical industrial assembly tasks. (3) Improving the recognition accuracy. For evaluating the standard and achievement of hand operations in ARAAT tasks, an algorithm for gesture recognition is proposed in this paper to improve recognition accuracy and efficiency. The ARAAT task is recorded into an input video by an AR device. To ensure precise interactions for trainees to work with virtual workpieces by real hands (empty hands or using assembly kits), virtual workpieces must match correctly to hands or tools according to spatial-temporal consistency. Based on the spatial-temporal consistency, we use Zernike moments combined histogram of oriented gradient and linear interpolation motion trajectories to simultaneously represent 2D static and 3D dynamic features, respectively. The directional pulse-coupled neural network is chosen as the classifier to recognize gestures. To reduce the computational cost, we define an action unit to reduce the dimensions of features. The score probability density distribution is defined and applied to optimize gesture boundaries iteratively to decrease the interference of invalid gestures during gesture recognition. (4) Decreasing the response time. We proposed an action and operation prediction method based on the standard operation order. The prediction method can early recognize the action and operation to reduce the response time and ensure a natural experience in ARAAT.

The subsequent sections of this paper are divided as follows: Section 2 describes the modeling for ARAAT; Section 3 presents the action categories, action recognition, and operation prediction; Section 4 details the experimental results compared with other algorithms on a homemade dataset and the experimental analysis; finally, Section 5 provides a short conclusion and suggestions for future research.

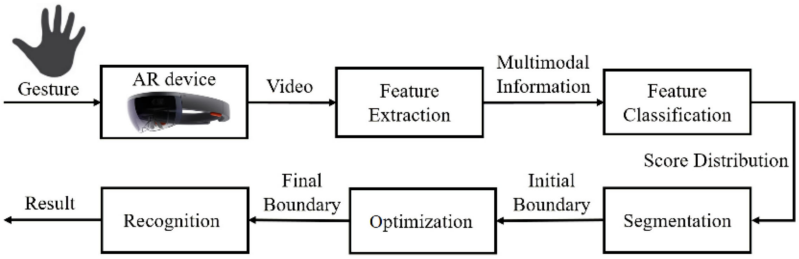

Figure 1. The framework of AR assisted assembly training (ARAAT) operation recognition. HoloLens takes trainee's gesture videos of corresponding ARAAT task operations and the multimodal features are extracted from the gesture videos. After classification, the features are used for gesture segmentation and optimization. By the recognition with the optimal gesture boundaries, the gesture results will be given to evaluate the standard and achievement of hand operations in ARAAT tasks.

2. Modeling for Augmented Reality Assisted Assembly Training

ARAAT tasks are mainly conducted by hand operations. Directly evaluating the performance of a whole ARAAT task is a complicated process, so we evaluate the performance of ARAAT tasks according to the standard and achievement of hand operations. For this purpose, we consider that a task can be decomposed into a series of hand operations, each of which can be decomposed into several continuous actions. Each action is related to a standard gesture based on the practical assembly task. The model of ARAAT conducted based on this decomposition is illustrated in Figure 2.

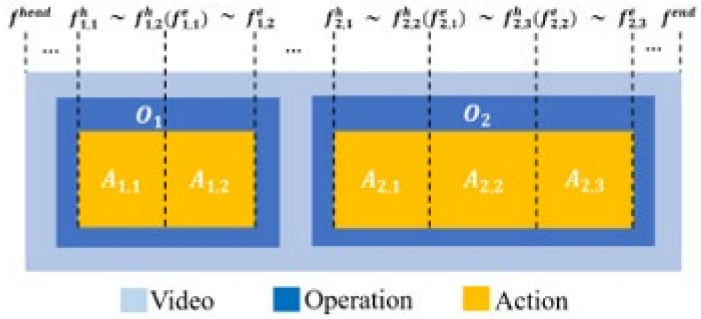

Figure 2. The model of Augmented Reality assisted assembly training.

Let $\boldsymbol{T}$ be a given assembly task and $\boldsymbol{V}$ be the recorded input video corresponding to $\boldsymbol{T}$. $\boldsymbol{V}$ can be expressed as a series of frames in digital images, that is,

$$V = \{f_t \,,\ t = 1, 2, \ldots\} \tag{1}$$

where f_t is the tth image frame.

For the convenience of formalizing $\boldsymbol{T}$ and the related definitions, we define a concatenation operator $\oplus$, where "$f \oplus g$" means that f occurs just after g. This gives the following definitions:

Definition 1. *Let O_i be the ith operation of task $\boldsymbol{T}$, and $\boldsymbol{N}$ be the number of operations in $\boldsymbol{T}$. Then,*

$$T = \left\{ O_i \middle| O_1 \oplus O_2 \oplus \ldots \oplus O_N \text{ and } \bigcap_{i=1}^{N} O_i = \varnothing \right\} \tag{2}$$

Definition 2. *Let $A_{i,j} \subset V$ be the jth action of ith operation O_i, M_i be the number of actions in O_i. Then,*

$$\begin{cases} O_i = \left\{ A_{i,j} \middle| A_{i,1} \oplus A_{i,2} \oplus \ldots \oplus A_{i,M_i} \right\} & i = 1, 2, \ldots, N \\ A_{i,j} = \left\{ f_{i,j}^{\mathrm{h}}, \ldots, f_{i,j}^{\mathrm{e}} \middle| f_{i,j}^{\mathrm{h}} \oplus \ldots \oplus f_{i,j}^{\mathrm{e}} \right\} & j = 1, 2, \ldots, M_i \end{cases} \tag{3}$$

where $f_{i,j}^{\mathrm{h}}$ and $f_{i,j}^{\mathrm{e}}$ are the head frame and the end frame of $\mathrm{A}_{i,j}$, respectively. We consider that the end frame of an action is the head frame of the next action, that is,

$$f_{i,j}^{\mathrm{e}} = f_{i,j+1}^{\mathrm{h}},\ i = 1, 2, \ldots, N, j = 1, 2, \ldots, M_i - 1 \tag{4}$$

To evaluate the performance of the task $\boldsymbol{T}$ in ARAAT, it is necessary to clarify which operations O_i are conducted for the task $\boldsymbol{T}$. Operation recognition is the identification of the corresponding sequences of actions $\{A_{i,j}\}$. To distinguish every action $A_{i,j}$ for each $O_i \in \boldsymbol{T}$, we need to label all the frame f_t of the input video $\boldsymbol{V}$ and recognize the gestures included in the action $A_{i,j}$. This dynamic gesture recognition allows for the performance of tasks in ARAAT to be evaluated.

3. Dynamic Gesture Recognition in Augmented Reality Assisted Assembly Training

The difficulty of dynamic gesture recognition in ARAAT lies with simultaneously segmenting and labeling gestures of actions in an operation. One exhaustive method of dynamic gesture recognition is to label all frames in the searching space, but this is time-consuming when dealing with long gestures. Therefore, a more efficient dynamic recognition algorithm is proposed in this section, consisting of three parts: action categories, action recognition, and operation prediction.

3.1. Action Categories

According to the American Society of Mechanical Engineers (ASME) standard operations [28], there are five typical types of assembly tasks in practical industrial assembly: "matching", "conjugating", "joining", "fastening", and "meshing", as shown in Figure 3. The essential operations for conducting these assembly tasks can be categorized into "inserting", "fastening", and "screwing", presented in Figure 4. Based on these typical tasks and operations, we conclude six basic actions [29]:

- "Rotating": trainees can change the orientation of objects;
- "Moving": the movement of an object or a tool;
- "Grasping": trainees can gain an object or a tool;
- "Releasing": trainees can put down an object or a tool;
- "Pointing": the selection action of an option or a virtual workpiece in AR environment;
- "Scaling": trainees can resize objects.

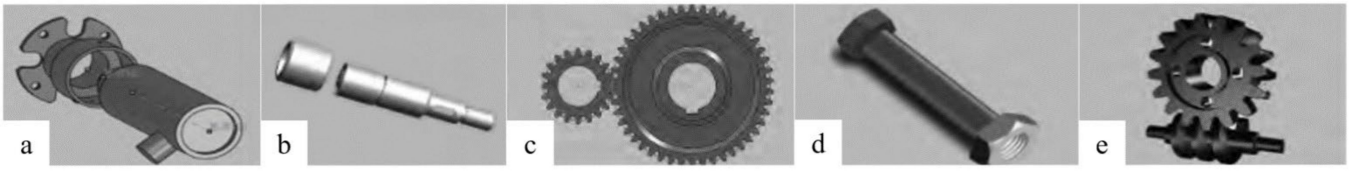

Figure 3. The types of typical assembly tasks: (**a**) matching, (**b**) conjugating, (**c**) joining, (**d**) fastening, and (**e**) meshing.

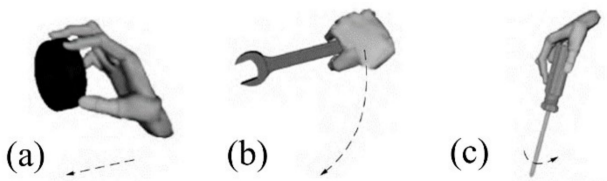

Figure 4. The types of typical assembly operations: (**a**) inserting, (**b**) fastening, and (**c**) screwing.

The actions are demonstrated in Figure 5:

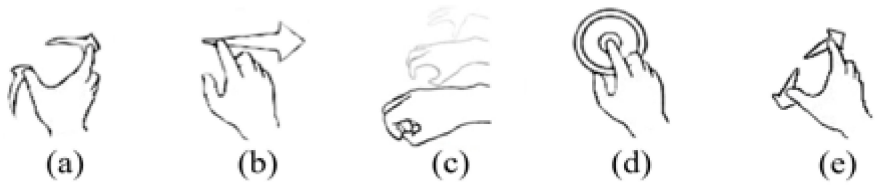

Figure 5. The types of standard actions for assembly: (**a**) rotating, (**b**) moving, (**c**) grasping and releasing, (**d**) pointing, and (**e**) scaling.

In ARAAT, trainees usually complete assembly tasks with their hands and several virtual tools, similar to practical industrial assembly. Thus, the above six standard actions are also the basis in ARAAT. To evaluate the standard and achievement of assembly tasks, we first need to recognize trainees' gestures of basic actions.

Different from single action recognition, meaningless transition actions between important actions in practical assembly operation are unavoidable. The transition actions make no sense and may impact negatively on the output in gesture recognition. To solve the disturbance of transition actions, a "Null" action is added in the standard basic action set, that is,

$$\boldsymbol{A} = \{\text{Pointing, Moving, Grasping, Releasing, Scaling, Rotating, Null}\} \tag{5}$$

3.2. Action Recognition

Action recognition mainly involves the following parts: feature extraction, gesture classification, and boundary segmentation. The specific processing steps are outlined below.

3.2.1. Feature Extraction

The actions in ARAAT are all dynamic gestures with movements. It is difficult to distinguish gestures with only 2D static features, so both static and dynamic features are extracted simultaneously to provide greater accuracy of action recognition. The static features are mainly the 2D representation characters of gestures and the dynamic features are the trajectories of hand motions.

(1) Static Features

In this paper, Zernike moments [30] and histogram of oriented gradient (HOG) are used to extract static features [31]. Zernike moments were first proposed by Frits Zernike in 1934 to uniquely describe functions on the unit disk and were then extended to describe images for feature extraction. Zernike moments have the properties of shift-invariant, scale-invariant, and rotation-invariant, and are always used as descriptors for gestures. Therefore, differently sized and shaped artifacts will not have a great influence on the gesture recognition results. However, Zernike moments cannot achieve good results in texture recognition. HOG is a feature descriptor for object detection in the static image that counts occurrences of gradient orientation in localized portions of an image. Thus, HOG is used to compensate for the recognition of local texture features in images.

For $\forall f_t \in A_{i,j} \subset V, i = 1,2,\ldots,N, j = 1,2,\ldots,M_i, t = 2,3,\ldots$, the static feature matrix from f_1 to f_t can be defined as

$$F_t^{\text{Static}} = \begin{bmatrix} f_1^{\text{Static}} \\ f_2^{\text{Static}} \\ \vdots \\ f_t^{\text{Static}} \end{bmatrix} \tag{6}$$

where $f_t^{\text{Static}} = (\boldsymbol{Z}_t^{nm}, V_t^{\text{HOG}}, f_t^{\text{Type}})$ is the static feature vector of frame f_t, $\boldsymbol{Z}_t^{nm}$, V_t^{HOG} and f_t^{Type} are Zernike moments, HOG feature vector and the gesture type, respectively.

Before calculating Zernike moments, each frame must first be normalized. Let $p_t(x_t, y_t)$ is a pixel of frame f_t. m_f and n_f are the length and width of f_t, and $0 \leq x \leq m_f$, $0 \leq y \leq n_f$. Then, we conduct a mapping transformation of f_t to a normalized polar image f'_t. We define $p'_t(x_t, y_t)$ as the pixel of f'_t, where $-n_f/2 \leq x, y \leq n_f/2$. The Zernike moments of order n with repetition m for gestures are calculated as follows,

$$Z_t^{nm} = \frac{\mathrm{n}+1}{\pi} \sum_{x_t} \sum_{y_t} p'_t(x_t, y_t) R_t^{nm*}(\rho_t) e^{jm\theta_t} \tag{7}$$

$$\theta_t = \arctan(y_t / x_t) \tag{8}$$

$$\rho_t = \sqrt{{x_t}^2 + {y_t}^2} \tag{9}$$

where n and m are nonnegative integers, $m \leq n$ and $n - m$ is even. R_t^{nm} is a radial polynomial and R_t^{nm*} is the complex conjugate. ρ_t is polar value and θ_t is polar angle.

We define the gesture type f_t^{Type} as the finger number used in each gesture, that is,

$$f_t^{\text{Type}} = \begin{cases} 1, \, if \text{ performing gestures only using the index finger} \\ 2, \, if \text{ performing gestures using both the thumb and index finger} \\ 5, \, if \text{ performing gestures using all fingers} \end{cases} \tag{10}$$

(2) Dynamic Features

In addition to the static representation of gestures, the dynamic motions of gestures also need to be acquired.

We define $p\left(x_t^C, y_t^C\right)$ as the centroid of hand in each frame $f_t \in A_{i,j} \subset V, i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M_i, t = 2, 3, \ldots$. We construct two vectors by $p\left(x_t^C, y_t^C\right)$, namely $X_t = \left(x_1^C, x_2^C, \ldots, x_t^C\right)$ and $Y_t = \left(y_1^C, y_2^C, \ldots, y_t^C\right)$. Considering that the range and speed of movements vary from person to person, a mean operation is conducted for the shift invariant and robustness of features. We obtain two new vectors X'_t and Y'_t,

$$\begin{aligned} X'_t &= \left(x_1^C - \overline{X}, x_2^C - \overline{X}, \ldots, x_t^C - \overline{X}\right) \\ Y'_t &= \left(y_1^C - \overline{Y}, y_2^C - \overline{Y}, \ldots, y_t^C - \overline{Y}\right) \\ \overline{X_t} &= \frac{1}{t} \sum_{i=1}^{t} x_i^C \\ \overline{Y_t} &= \frac{1}{t} \sum_{i=1}^{t} y_i^C \end{aligned} \tag{11}$$

where $\overline{X_t}$ and $\overline{Y_t}$ are the mean values of X_t and Y_t. The dynamic features then can be presented by

$$F_t^{\text{Dyn}} = \begin{bmatrix} X'_t \\ Y'_t \end{bmatrix} \tag{12}$$

(3) Feature Matrix

In summary, for $\forall f_t \in A_{i,j} \subset V, i = 1, 2, \ldots, N, j = 1, 2, \ldots, M_i, t = 2, 3, \ldots$, the feature matrix F_t representing gestures from f_1 to f_t is expressed as:

$$F_t = \begin{bmatrix} \left(F_t^{\text{Static}}\right)^T \\ F_t^{\text{Dyn}} \end{bmatrix} \tag{13}$$

3.2.2. Gesture Classification

We use the feature matrix F_t as the input of the classifier C and recognize the gestures. In other words, classifier C can be seen as a mapping for $\forall f_t \in A_{i,j} \subset V$, $i = 1, 2, \ldots, N$,

$j = 1, 2, \ldots, M_i, t = 2, 3, \ldots$. By calculating the feature matrix F_t, we can obtain the recognition result $C(f_t, F_t) \in \boldsymbol{A}$. The classifier C can be defined as:

$$C : f_t \times F_t \rightarrow \boldsymbol{A} \tag{14}$$

In this paper, the directional pulse-coupled neural network (DPCNN) is chosen as the classifier. The DPCNN can classify and recognize dynamic gestures by template matching and is often applied to real-time applications, which is verified in our previous work [32]. The DPCNN can select the neuron firing directions by different excitations to reduce computational complexity and time of traditional PCNN. The DPCNN can also improve recognition accuracy by the choice of reasonable firing directions. For each gesture in the standard basic action set $\boldsymbol{A}$, that is, {Pointing, Moving, Grasping, Releasing, Scaling, Rotating, Null}, we construct a single gesture video dataset that is used as a training template to train parameters of the DPCNN classifier. We input the feature matrix F_t of each single gesture into the DPCNN and then train the classifier.

To increase the efficiency of gesture recognition, an action unit is introduced in the process of feature classification. We define L as the length of the action unit. An L that is too small will lead to poor computational efficiency, while an L that is too large will increase computational complexity. To determine the value of L, we collected assembly gesture video data online and conducted an experimental statistical analysis. By trial and error, we determined that $L = 20$ is the best length for the action unit. Because of the use of the action unit, the dimension of features is reduced to 20 instead of all frames. For $\forall f_t \in A_{i,j} \subset V, i = 1, 2, \ldots, N, j = 1, 2, \ldots, M_i, t = 20, 21, \ldots$, f_t and 19 frames before f_t construct an action unit denoted by U_t, that is,

$$U_t = \{f_{t-19}, f_{t-18}, \ldots, f_t | f_{t-19} \oplus f_{t-18} \oplus \ldots \oplus f_{t-1} \oplus f_t\} \tag{15}$$

The multiple features of the action unit U_t are then gained by the feature extractor. Based on the features, the classifier assigns each frame in the action unit U_t to the appropriate action category and gives scores to each action category, which are shown in Figure 6a. This reduces the computational cost and time, as well as the impact of video length on complexity, allowing even long operation videos to be processed quickly and efficiently. The scores then produce a unit probability distribution $p_U(A)$ for the action class. The distribution suggests the probability of actions assigned to each class. As the action unit moves forward along frames, the probability distribution $p(A)$ on the whole video can be produced, as illustrated in Figure 6b.

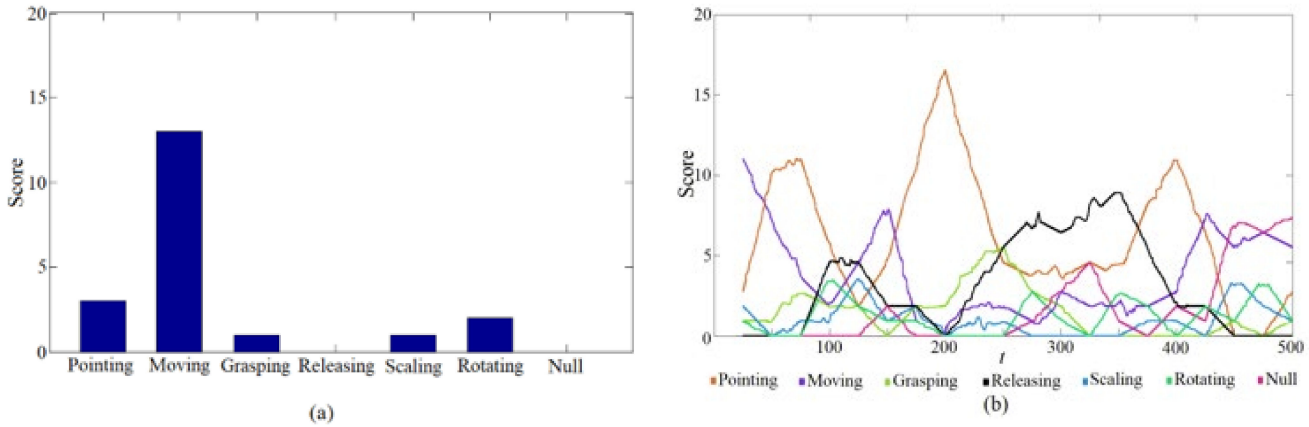

Figure 6. The output of the classifier. (**a**) Scores of each action in an action unit, and (**b**) the score distributions of each action on an input video.

3.2.3. Boundary Segmentation

As mentioned in Section 2, the operations of a task are constructed by actions, which are performed by gestures. For accurate recognition of gestures, it is necessary to first define the boundary of each action. In this section, we use density distribution optimization to segment the action boundary.

Let $p(A|f_t)$ be the distribution of action class A on frame $f_t \in V$. We define the candidate boundary frame $F^{\text{Candidate}}$ (as seen in Figure 7),

$$F^{\text{Candidate}} = \left\{ f_k^{\text{h}} \middle| k = 1, 2, \ldots \right\} \tag{16}$$

where f_k^{h} is kth candidate boundary frame and $f_k^{\text{h}} = f_i$ satisfies $s.t = argmax\ p(A|f_{i-1}) \neq argmax\ p(A|f_i)\,, i = 20, 21, \ldots, t$.

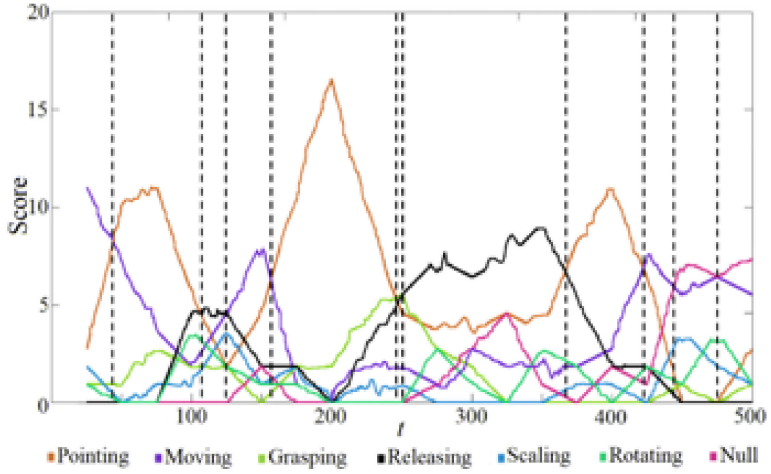

Figure 7. The initial boundary points in the distribution of actions.

Then, we optimize the segmentation on the log density distribution $logp(A_{i,j})$ by tuning $(f_{i,j}^{\text{h}}, f_{i,j}^{\text{e}})$ with the candidate boundary frame $F^{\text{Candidate}}$, that is,

$$\begin{aligned} &argmax \sum_j logp(A_{i,j}) \\ &s.t = A_{i,j} = f_{i,j}^{\text{h}} \oplus \ldots \oplus f_{i,j}^{\text{e}} \end{aligned} \tag{17}$$

Equation (17) is solved by dynamic programming and the optimal result of boundaries $f_{i,j}^{\text{h}} \oplus \ldots \oplus f_{i,j}^{\text{e}}$ is obtained. We propose a simplified algorithm which can search for the optimal segment boundary in real time. The procedure of dynamic programming is shown in Algorithm 1.

In Algorithm 1, we need to calculate maximal $C_{f_{i,j}^{\text{h}} \cdots f_W}$ which is the optimal segmentation cost. The cost $C_{f_{i,j}^{\text{h}} \cdots f_W}$ is calculated from the last action to the first one. In the search range of each action, the cost $C_{f_{i,j}^{\text{h}} \cdots f_W}$ is the sum of maximal log density distribution from $f_{i,j}^{\text{h}}$ to $f_{i,j}^{\text{e}}$ and the cost from $f_{i,j}^{\text{e}}$ to the last frame f_W. B_j records the segment boundary $f_{i,j}^{\text{e}}$ in the optimal segmentation.

After each boundary optimization, we need to reduce the influence of invalid actions. In a real assembly operation, each action usually lasts more than 60 frames. Thus, any pair of $f_{i,j}^{\text{h}}$ and $f_{i,j}^{\text{e}}$ shorter than 10 frames in an operation is more likely to be an invalid action and is removed. When the pair of $f_{i,j}^{\text{h}}$ and $f_{i,j}^{\text{e}}$ is more than 10 frames but shorter than 60 frames and is recognized to not be a transition action "Null", the pair is merged to the former segment. Then, we solve (17) again. The procedure is repeated until neighboring $f_{i,j}^{\text{h}}$ and $f_{i,j}^{\text{e}}$ are appropriate and stable.

Algorithm 1: Dynamic Programming for Optimal Boundary Segmentation

Input: boundary frames, $(f^{\mathbf{h}}_{i,j}, f^{\mathbf{e}}_{i,j})$; the number of actions, M; the search range of $f^{\mathbf{h}}_{i,j}$, α_j; the number of frames W

for $j = M, M-1, \ldots, 1$

$C_j = -\infty, B_j = -\infty$;

  for $f^{\mathbf{h}}_{i,j}$ **in** α_j

for $f^{\mathbf{e}}_{i,j}$ **in** α_{j+1}

$C_{f^{\mathbf{h}}_{i,j} \oplus \ldots \oplus f_W} = C_{f^{\mathbf{e}}_{i,j} \oplus \ldots \oplus f_W} + \mathbf{max}logp\left(f^{\mathbf{h}}_{i,j} \oplus \ldots \oplus f^{\mathbf{e}}_{i,j}\right)$;

if $C_j < C_{f^{\mathbf{h}}_{i,j} \oplus \ldots \oplus f_W}$

$C_j = C_{f^{\mathbf{h}}_{i,j} \oplus \ldots \oplus f_W}, B_j = f^{\mathbf{e}}_{i,j}$;

end if

end for

end for

end for

Find $(f^{\mathbf{h}}_{i,j}, f^{\mathbf{e}}_{i,j})$ **from** B_j;

Output: The optimal boundary segmentation frames $(f^{\mathbf{h}}_{i,j}, f^{\mathbf{e}}_{i,j})$;

The flowchart of boundary segmentation is shown in Figure 8 and the algorithm is given below:

(1) Optimize the boundary by solving (17), go to (2).

(2) If the pair of $f^{\mathrm{h}}_{i,j}$ and $f^{\mathrm{e}}_{i,j}$ is shorter than 10 frames, remove the pair and go to (3). Else, go to (3).

(3) If the pair of $f^{\mathrm{h}}_{i,j}$ and $f^{\mathrm{e}}_{i,j}$ is shorter than 60 frames, go to (4). Else, the optimized boundaries are obtained.

(4) If the pair of $f^{\mathrm{h}}_{i,j}$ and $f^{\mathrm{e}}_{i,j}$ is a transition action, remove the pair and go to (1). Else, merge the pair to the former segment and go to (1).

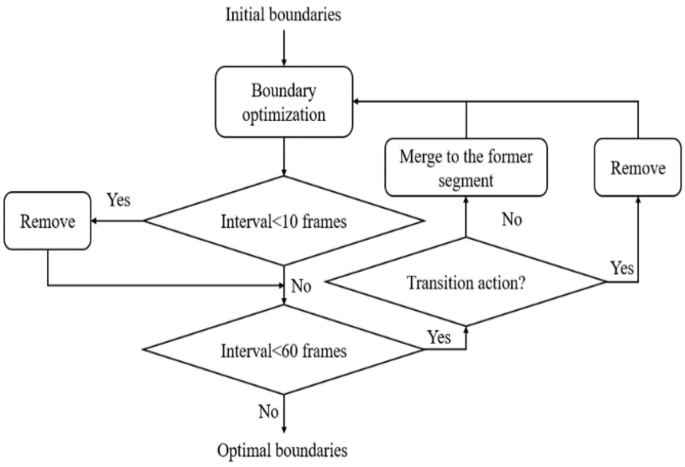

Figure 8. The algorithm of boundary segmentation.

3.3. Action and Operation Prediction

The purpose of ARAAT is to improve the standard for trainees by evaluating performance of hand operations. Thus, the prediction of actions and operations is as necessary as action recognition. The prediction algorithm can identify the assembly operations of trainees early by recognizing actions and the judgement of current orders, allowing for the evaluation of the standard order and achievement of the following actions.

In ARAAT, trainees are required to use their hands or various virtual equipment to complete different assembly training tasks. As mentioned in Section 3.1, "inserting", "fastening", and "screwing" are typical practical assembly operations. To conduct these training operations, trainees need to hold some equipment as assistance, such as a wrench or screwdriver. By moving

the workpieces with their hands, trainees can carry out the "inserting" operation; by rotating the wrench, trainees can perform the "fastening" operation; by rotating the screwdriver, trainees can perform the "screwing" operation. In ARAAT, trainees also need to complete the assembly training through controlling the virtual workpiece by their hands or equipment.

Each assembly operation in ARAAT contains several actions in a standard order, concluded as follows (refer to Figure 9):

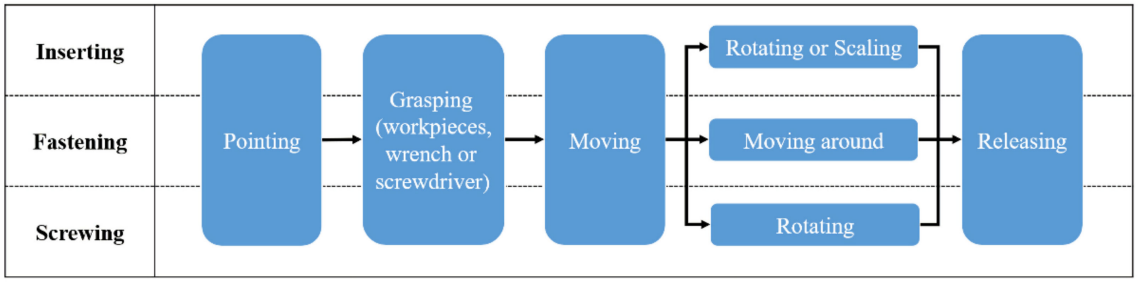

Figure 9. The action orders of three typical operation "inserting", "fastening", and "screwing" in ARAAT.

Inserting: grasping (the workpieces) → moving → rotating/scaling →releasing (the workpieces), that is,

$$\text{Inserting} = \text{grasping} \oplus \text{moving} \oplus \text{rotating/scaling} \oplus \text{releasing} \quad (18)$$

Fastening: grasping (the wrench) → moving → moving around →releasing (the wrench), that is,

$$\text{Fastening} = \text{grasping} \oplus \text{moving} \oplus \text{moving around} \oplus \text{releasing} \quad (19)$$

Screwing: grasping (the screwdriver) → moving → rotating → releasing (the screwdriver), that is

$$\text{Screwing} = \text{grasping} \oplus \text{moving} \oplus \text{rotating} \oplus \text{releasing} \quad (20)$$

We can evaluate the performance of actions and operations by recognizing the gestures and predicting the action order which the trainee is performing. Furthermore, when the trainee carries out an operation in ARAAT, the virtual object needs to give the corresponding reaction. If there is a delay in the reaction, the operation will not be smooth and the interaction will be unnatural. In order to avoid inconsistencies with the operation, it is necessary to predict the current and next actions and give the computer enough time to make real-time feedback reactions. By the recognition of frames in the action unit, we can predict the uncompleted actions and reduce the response time, providing trainees with a smooth and natural human-machine interaction experience in ARAAT.

Based on the above discussion, the framework of action and operation prediction is given in Figure 10.

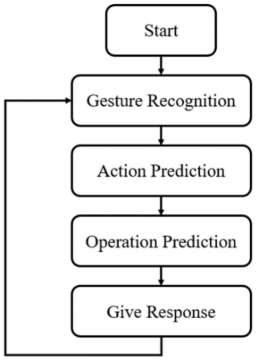

Figure 10. Action and operation prediction in ARAAT.

4. Experiments and Discussion

4.1. Experimental Design and Datasets

The experiments are divided into two sections to evaluate the proposed algorithm in this paper. The first section is to validate the recognition accuracy and efficiency of the proposed dynamic gesture recognition algorithm. We conduct the experiments on 4 datasets compared with other algorithms which contain 4 parts: frame recognition, action recognition, action boundary segmentation, and the effect of image resolution on recognition. The second section is to validate the effectiveness of proposed algorithm for operation recognition and prediction and the reliability of proposed ARAAT system. We invite participants to take real-time ARAAT tasks on HoloLens device to evaluate the naturalness of interactions in ARAAT.

In the first section, the proposed algorithm for dynamic gesture recognition is evaluated on two public datasets (Sheffield Kinect Gesture (SKIG) dataset [33] and Sebastien Marcel Dynamic Hand Posture Dataset [34]) and two homemade datasets (Assembly Gesture Video Dataset and HoloLens ARAAT Dataset).

The SKIG Dataset was captured by a Kinect device that included a RGB camera and a depth camera. It contains 10 gestures for Circle (clockwise), Triangle (anti-clockwise), Up-Down, Right-Left, Wave, "Z", Cross, Comehere, Turn-Around, and Pat. Six subjects performed each gesture 18 times and there are 1080 RGB image sequences in total.

The Sebastien Marcel Dynamic Hand Posture Dataset contains 4 hand gestures: Clic, Rotate, Stop-Grasp-Ok, and No. Each gesture in the dataset was performed 13–15 times and recorded by image sequences.

The sample image frames of the SKIG Dataset and Sebastien Marcel Dynamic Hand Posture Dataset are presented in Figures 11 and 12. These two datasets both comprise common gestures which are not very relevant to assembly. Thus, we introduced two new homemade datasets, called the Assembly Gesture Video Dataset and HoloLens ARAAT Dataset, to validate our approach.

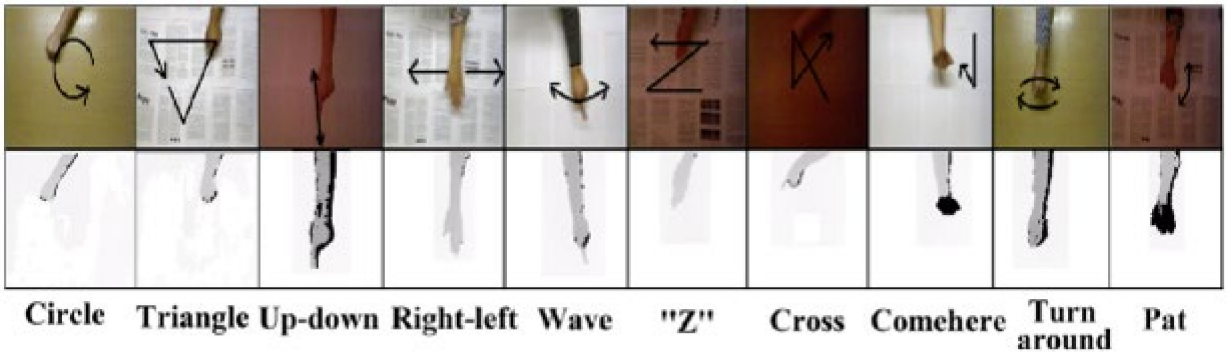

Figure 11. Image frames from SKIG Dataset.

Figure 12. Image frames from Sebastien Marcel Dynamic Hand Posture Dataset.

The Assembly Gesture Video Dataset contains 437 assembly operation gesture video sequences collected online from different uploaders on YouTube demonstrated in Figure 13. The dataset has 6 standard assembly actions: "Pointing", "Moving", "Grasping", "Releasing", "Scaling", "Rotating", and the transition action "Null". These 6 standard actions were generalized according to the ASME standard operations [28]. The resolution for each video is 640 × 480.

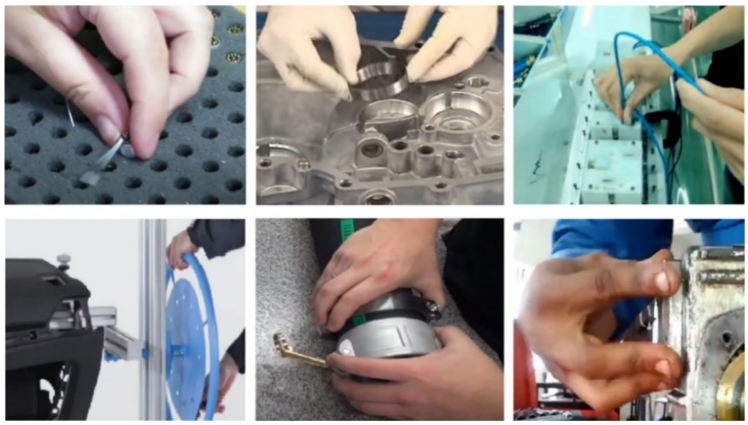

Figure 13. Image frames from Assembly Gesture Video Dataset.

The HoloLens ARAAT Dataset is constructed from various action videos from HoloLens RGB cameras. The videos of ARAAT tasks used in the experiments were captured by HoloLens and the resolution of the image is 1028 × 720 with 30 frames per second (fps). The dataset is divided into two parts: the single action dataset and the ARAAT task dataset. The single action dataset contains 6 types of actions: "Pointing", "Moving", "Grasping", "Releasing", "Scaling", and "Rotating". Videos were recorded of 30 participants, who conducted the actions of real assembly operations based on the ASME standard operations. Each participant performed 6 actions 10 times each, which created 60 videos for each participant and 1800 videos in total. Each video has approximately 100–150 frames. The single action dataset is split into the training and testing sets. The ARAAT task dataset was recorded by 30 participants and is used for testing sets in the boundary segmentation experiments. Each trainee performed the ARAAT tasks, containing 6 types of actions: "Pointing", "Moving", "Grasping", "Releasing", "Scaling", and "Rotating", in any order at will for 10 times, producing 300 testing task videos.

4.2. Result of Action Recognition

4.2.1. Public Datasets

The results on the SKIG Dataset and Sebastien Marcel Dynamic Hand Posture Dataset are presented in Tables 1 and 2, respectively.

Table 1. The recognition accuracy on SKIG Dataset.

Method	Accuracy
RGGP + RGB-D	88.7 ± 1.3
4DCOV	93.8 ± 0.6
Depth Context	95.4 ± 2.1
HOG + LBP	97.3 ± 1.7
DLEH2	98.4 ± 1.0
Proposed	99.2 ± 0.7

Table 2. The recognition accuracy on Sebastien Marcel Dynamic Hand Posture Dataset.

Method	Accuracy
DCCA	65.5 ± 5.5
TCCA	82.1 ± 2.7
Product Manifolds	88.4 ± 3.4
Genetic Programming	85.0 ± 1.1
Tangent Bundles	91.7 ± 0.9
Cov3D	93.3 ± 1.2
Proposed	96.8 ± 0.4

On the SKIG Dataset, the proposed algorithm was compared with RGGP + RGB-D [33], 4DCOV [35], Depth Context [36], HOG + LBP [37], and DLEH2 (DLE + HOG2) [38] and achieved state-of-the-art accuracy. The proposed algorithm was more attentive to the spatial-temporal consistency of gestures during the process of feature learning, which was not reflected in HOG + LBP and DLEH2. The comparison results confirm that spatial-temporal consistency plays an important role in gesture recognition. On the Sebastien Marcel Dynamic Hand Posture Dataset, the proposed algorithm was compared with Discriminative Canonical Correlation Analysis (DCCA) [39], tensor canonical correlation analysis (TCCA) [40], Product Manifolds (PM) [41], Genetic Programming (GP) [42], Tangent Bundles (TB) [43], and 3D Covariance spatio-temporal descriptor (Cov3D) [44]. As shown in Table 2, the proposed algorithm outperforms the others by at least 3% in recognition accuracy. Among the algorithms in Table 2, Cov3D produced a result second only to the proposed algorithm because it learned the spatial-temporal features of the gestures during the process. The result shown in Tables 1 and 2 is not as good as the proposed algorithm in this paper because only dynamic spatial features cannot fully represent the gestures. This further proves the importance of spatial-temporal consistency information in gesture recognition.

4.2.2. Homemade Datasets

We also evaluated the proposed algorithm with the others on the homemade datasets. Five-fold cross-validation was used in the experiments, which divided 80% of the data for training and 20% for testing each time. Tables 3 and 4 show the accuracy of the proposed algorithm on the Assembly Gesture Video Dataset and HoloLens ARAAT Dataset, respectively. Tables 5 and 6 present the recognition results and processing time on the homemade datasets compared with SSBoW [45], DSBoW [46], DTBoW [47,48], and DFW [49].

Table 3. The recognition accuracy on Assembly Gesture Video Dataset.

Action	Pointing (%)	Moving (%)	Grasping (%)	Releasing (%)	Scaling (%)	Rotating (%)	Null (%)
Result	94.3	96.7	93.6	90.1	92.1	90.2	98.0
	92.5	96.1	93.3	90.9	90.4	90.8	97.1
	91.6	95.2	92.9	91.2	91.3	91.1	95.8
	93.1	97.4	91.7	90.6	91.9	90.9	97.6
	92.8	95.9	92.4	91.3	91.0	90.4	96.3
Average	92.9 ± 1.0	96.2 ± 0.7	92.8 ± 0.6	90.8 ± 0.2	91.3 ± 0.5	90.7 ± 0.1	97.0 ± 0.8

Table 4. The recognition accuracy on HoloLens ARAAT Dataset.

Action	Pointing (%)	Moving (%)	Grasping (%)	Releasing (%)	Scaling (%)	Rotating (%)	Null (%)
Result	93.4	96.2	91.6	91.2	91.7	90.6	95.2
	94.2	95.1	92.3	91.5	90.9	91.7	96.8
	93.1	97.2	91.7	92.1	92.5	90.4	95.9
	93.8	95.4	91.5	91.6	93.0	91.7	94.3
	94.0	96.5	93.1	92.9	92.2	92.1	97.6
Average	93.7 ± 0.4	96.1 ± 0.7	92.0 ± 0.6	91.9 ± 0.6	92.1 ± 0.7	91.3 ± 0.7	96.0 ± 1.1

Table 5. The recognition accuracy and processing time on Assembly Gesture Video Dataset with other algorithms.

Action	Pointing (%)	Moving (%)	Grasping (%)	Releasing (%)	Scaling (%)	Rotating (%)	Null (%)	Time (ms)
SSBoW	65.2 ± 13.5	81.7 ± 11.4	83.8 ± 12.1	88.2 ± 12.5	59.8 ± 18.8	76.5 ± 15.0	74.2 ± 20.7	1725.9
DSBoW	78.8 ± 10.3	83.8 ± 10.4	87.5 ± 7.2	86.1 ± 4.3	72.4 ± 10.9	81.7 ± 4.0	84.8 ± 19.1	1419.7
DTBoW	71.9 ± 11.5	91.4 ± 8.3	90.5 ± 3.6	86.7 ± 6.1	90.2 ± 4.7	87.6 ± 3.5	91.1 ± 9.2	1067.4
DFW	85.8 ± 6.2	94.0 ± 2.3	90.9 ± 7.8	88.3 ± 2.6	88.9 ± 3.4	85.1 ± 2.3	93.2 ± 5.8	893.1
Proposed	92.9 ± 1.0	96.2 ± 0.7	92.8 ± 0.6	90.8 ± 0.2	91.3 ± 0.5	90.7 ± 0.1	97.0 ± 0.8	524.3

Table 6. The recognition accuracy and processing time on HoloLens ARAAT Dataset with other algorithms.

Action	Pointing (%)	Moving (%)	Grasping (%)	Releasing (%)	Scaling (%)	Rotating (%)	Null (%)	Time (ms)
SSBoW	66.8 ± 10.1	83.3 ± 9.7	87.3 ± 10.9	89.6 ± 12.1	64.0 ± 20.2	80.1 ± 15.7	78.0 ± 22.6	2170.1
DSBoW	72.0 ± 12.1	89.7 ± 8.3	90.1 ± 4.1	90.3 ± 3.9	70.0 ± 12.7	82.4 ± 3.9	83.3 ± 15.6	1963.0
DTBoW	74.0 ± 10.4	94.2 ± 2.1	91.7 ± 2.0	90.9 ± 5.7	91.1 ± 2.6	88.6 ± 5.6	91.9 ± 8.8	1542.7
DFW	89.6 ± 3.2	93.7 ± 2.7	88.4 ± 5.1	86.1 ± 1.9	85.3 ± 2.8	87.7 ± 1.4	90.5 ± 6.0	1205.5
Proposed	93.7 ± 0.4	96.1 ± 0.7	92.0 ± 0.6	91.9 ± 0.6	92.1 ± 0.7	91.3 ± 0.7	96.0 ± 1.1	778.9

The recognition rates of the proposed algorithm for each action were all over 90%. Outstanding performances of over 96% were achieved for several actions, such as "Moving" and "Null". The average accuracy of all actions was 93.1% and 93.3% for the Assembly Gesture Video Dataset and HoloLens ARAAT Dataset, respectively. The results show that the proposed algorithm can obtain the highest recognition accuracy across all algorithms from the table. In particular, for "pointing" and "scaling", the proposed algorithm outperforms all the others by a margin that showcases its advantage in dynamic gesture recognition. Among all the algorithms, SSBoW performed the worst in all action recognitions; this is because each frame from the input video could not be clearly classified to the correct type by SSBoW, which led to an overall low recognition rate. Frame-to-frame recognition results are presented in Figures 14 and 15. In an action unit, the proposed algorithm considerably outperformed other approaches with the highest accuracy, while SSBoW produced the lowest results. Figures 14 and 15 explain why SSBoW could not produce good accuracy in each action.

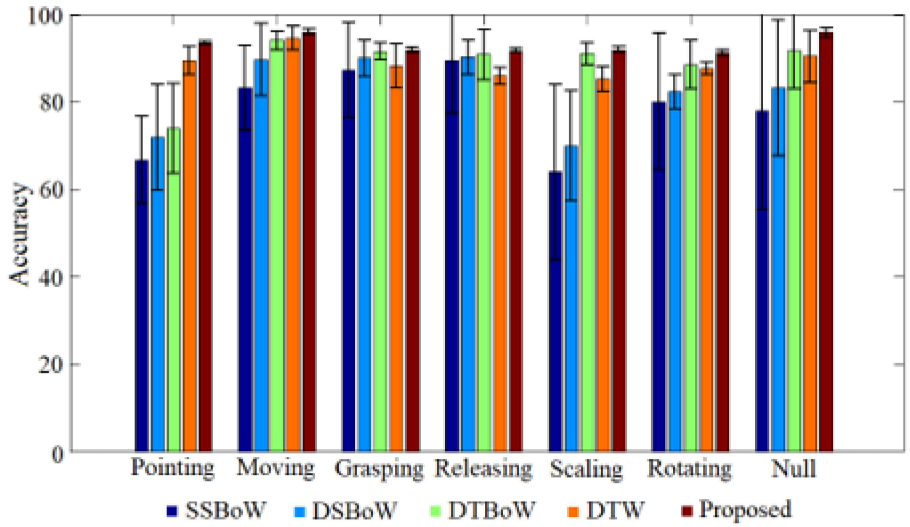

Figure 14. Accuracy of frames recognition in an action unit on Assembly Gesture Video Dataset.

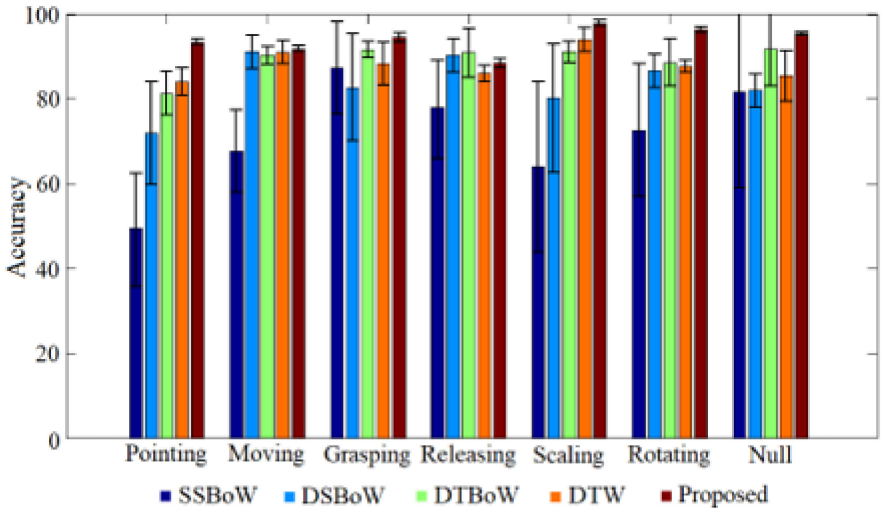

Figure 15. Accuracy of frames recognition in an action unit on HoloLens ARAAT dataset.

To evaluate the performance of the proposed algorithm, we also conducted boundary segmentation experiments on the HoloLens ARAAT dataset. We labeled the ground truth boundary segmentation of the videos in advance. Because the action "Null" is the meaningless transition action, we excluded it from the ground truth. We developed a measure to evaluate the segmentation accuracy (SA), expressed as:

$$SA = \frac{\left| F_{result} \cap F_{groundtruth} \right|}{F_{groundtruth}} \tag{21}$$

where F_{result} denotes the frames of the segmentation result and $F_{groundtruth}$ denotes the segmentation frames of the ground truth. Figure 16 demonstrates the accuracy of each action's segmentation on the HoloLens ARAAT dataset. DFW provided the best recognition results compared to SSBoW, DSBoW, and DTBoW. However, it is outperformed by the proposed algorithm. This is because DFW is incapable of segmenting each action explicitly, so incorrectly segmented frames will interfere with its recognition accuracy. Through the optimal boundary search, the proposed algorithm can precisely divide each action from a long input video. The segmentation results also prove why the proposed algorithm outperforms the other approaches in action recognition accuracy.

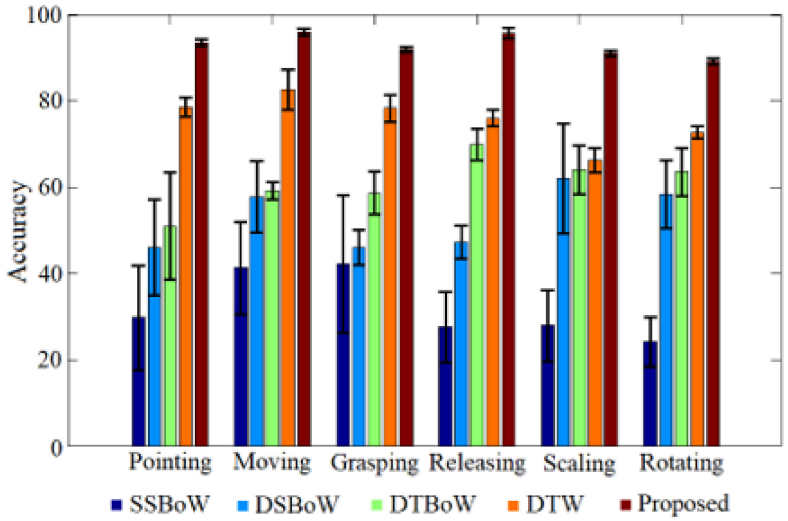

Figure 16. Accuracy of action boundary segmentation on HoloLens ARAAT dataset.

Figure 17 presents the accuracy of different resolutions to further support the effectiveness of the proposed algorithm. When the resolutions were over 160 × 120, the accuracies were over 90%. However, the accuracy was under 90% for 80 × 60 and only 64% for 40 × 30. Although the accuracy was low in some low resolutions, the proposed algorithm is still reliable in most situations because these low-resolution inputs are rarely used in current applications.

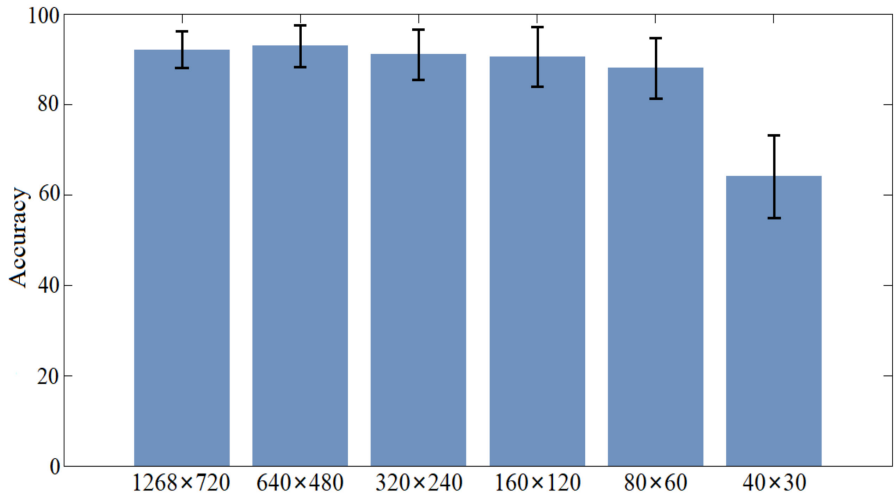

Figure 17. Effect of image resolution on accuracy.

4.3. Result of Operation Recognition and Prediction

In the ARAAT system, the action sequence is important for the evaluation of the standard and achievement of trainees' operations. Thus, recognition and prediction of the operation is as necessary as action recognition.

To validate the reliability of the proposed algorithm, 30 participants took part in performing real-time ARAAT tasks. Half of the participants were beginners in assembly operations and the other half had basic assembly operation knowledge but little practical experience. Participants completed assembly training tasks from an application on HoloLens written in C#. The demonstration scenario is shown in Figure 18. Participants could assemble workpieces in the tasks in an arbitrary order depending on their own needs. During assembly, participants chose whether to use their bare hands or various virtual tools, such as a wrench or screwdriver, to complete the tasks, as illustrated in Figure 18. The tasks were mainly completed by 6 actions: "Pointing", "Moving", "Grasping", "Releasing", "Scaling", and "Rotating", and 3 operations: "Inserting", "Fastening", and "Equipping".

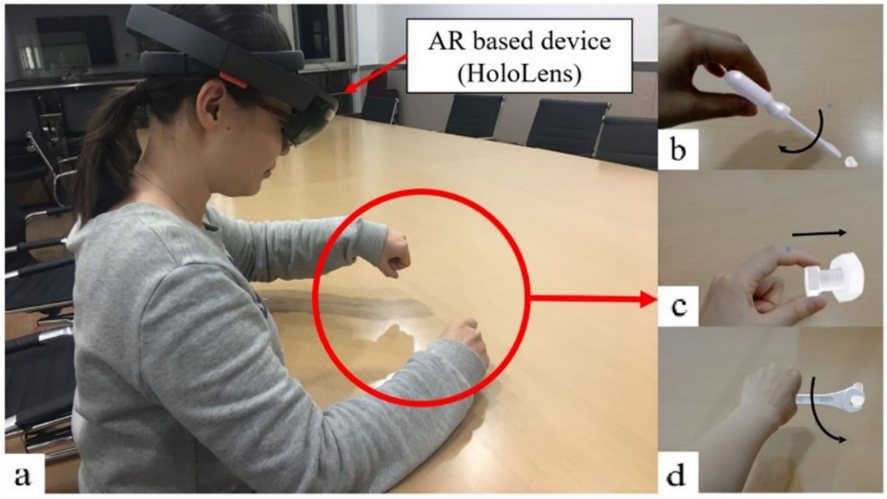

Figure 18. The experimental scenario of augmented reality assisted assembly training. (**a**) Experimental scenario, (**b**) screwing operation, (**c**) inserting operation, (**d**) fastening operation.

To evaluate the sequence recognition accuracy and prediction efficiency, we developed two measures: the sequence accuracy ($A_{sequence}$) and the degree of early recognition (DER), defined respectively as the following:

$$A_{sequence} = \frac{\sum_{i=1}^{N}\left(\sum_{j=1}^{M} E\left(\widetilde{A_{i,j}} = A_{i,j}\right)\right)}{N \times M} \tag{22}$$

$$DER = \frac{N_{re}}{N_{total}} \tag{23}$$

where $\widetilde{A_{i,j}}$ is the ground truth of $A_{i,j}$ and $E\left(\widetilde{A_{i,j}} = A_{i,j}\right)$ means if $\widetilde{A_{i,j}} = A_{i,j}$ is true, the output is 1 while 0 otherwise. N_{re} is the number of frames of the recognized operation and N_{total} is the total number of frames of the whole operation. A lower DER means that the operations are recognized early and a higher DER means that the operations are recognized late. The experiment results shown in Table 7 indicate that the proposed algorithm can predict the operations up to 40% of the time with a high recognition rate of 93.5%. These results mean that the prediction can give the machine enough time to make a reaction and provide the trainees with a smooth and friendly human-machine interaction in ARAAT.

Table 7. The recognition accuracy of Augmented Reality assisted assembly training operations.

Operation	Inserting (%)	Fastening (%)	Screwing (%)	Average (%)
$A_{sequence}$	94.1 ± 1.1	93.0 ± 1.3	93.1 ± 1.0	93.5 ± 2.4
DER	40.3 ± 2.1	23.4 ± 2.5	33.7 ± 2.8	32.5 ± 4.2

5. Conclusions

ARAAT is an effective and affordable technique for labor training in the automobile and electronic industry. In this paper, we developed an ARAAT system to transform the complicated ARAAT task evaluation into a problem of gesture recognition and proposed a gesture recognition and prediction algorithm. We built a complicated ARAAT task model where a task is decomposed into a series of hand operations and each hand operation is further decomposed into several continuous actions corresponding to gestures. We defined five typical tasks, three typical operations, and six standard actions based on the practical assembly works, defined an action unit to reduce the dimensions of features during the recognition, and defined a score probability density distribution iteratively to optimize gesture boundaries to reduce interference from invalid gestures. Furthermore, we simultaneously extracted 2D static and 3D dynamic features of standard gestures to improve the gesture recognition precision and proposed an action and operation prediction method for a short response delay time and a natural interaction. The proposed algorithm was evaluated on two public datasets and two homemade assembly datasets, and achieved a high recognition rate of 93.5% up to 40% of the time. The experimental results showed that the proposed algorithm can increase recognition accuracy and reduce the computational cost, which help to ensure reliability in the ARAAT task evaluation and improve the experience of human-machine interaction. Although the procedures of ARAAT are relatively static and predictable, it remains a challenge to handle the different assembly difficulties, various products, and rapid updating of assembly skills. Therefore, in future, we will pay more attention to the research of assembly operations which is adapted to different assembly difficulties and various products, and try to build a new ARAAT system which can provide guidelines to trainees on updated assembly skills.

Author Contributions: Conceptualization, Q.Z.; methodology, J.D.; software, J.D.; validation, J.D.; formal analysis, J.D.; investigation, J.D.; resources, Z.X.; data curation, J.D.; writing—original draft preparation, J.D.; writing—review and editing, Q.Z. and Z.X.; visualization, J.D.; supervision, Q.Z. and Z.X.; project administration, Q.Z.; funding acquisition, Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science Foundation of China (61773365, 61811540033), National Natural Science Foundation of China (U2013205), and Chinese Academy of Sciences Youth Innovation Promotion Association Excellent Member Program (Y201968).

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the nature of data collected, which does not involve any personal information that could lead to the later identification of the individual participants. The participant in Figure 18 is the first author.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The public datasets presented in this study can be found at: (1) http://lshao.staff.shef.ac.uk/data/SheffieldKinectGesture.htm and (2) https://www.idiap.ch/webarchives/sites/www.idiap.ch/resource/gestures/.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tsai, C.Y.; Liu, T.Y.; Lu, Y.H. A novel interactive assembly teaching aid using multi-template augmented reality. *Multimed. Tools Appl.* **2020**, *79*, 31981–32009. [CrossRef]
2. Mahmood, B.; Han, S.U.; Lee D, E. BIM-based registration and localization of 3D point clouds of indoor scenes using geometric features for augmented reality. *Remote Sens.* **2020**, *12*, 2302. [CrossRef]
3. Liu, W.; Wang, C.; Bian, X.; Chen, S.; Li, W.; Lin, X.; Li, Y.; Weng, D.; Lai, S.-H.; Li, J. AE-GAN-Net: Learning invariant feature descriptor to match ground camera images and a large-scale 3D image-based point cloud for outdoor augmented reality. *Remote Sens.* **2019**, *11*, 2243. [CrossRef]
4. Friedrich, W.; Jahn, D.; Schmidt, L. ARVIKA-augmented reality for development, production and service. *ISMAR* **2002**, *2*, 3–4.
5. Fuchs, H.; Livingston, M.A.; Raskar, R.; Keller, K.; Crawford, J.R.; Rademacher, P.; Drake, S.H.; Meyer, A.A. Augmented reality visualization for laparoscopic surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 934–943.
6. Pokémon, G.O. 2016. Available online: http://www.pokemongo.com/en-us (accessed on 2 September 2016).
7. Liarokapis, F.; Mourkoussis, N.; White, M.; Darcy, J.; Sifniotis, M.; Petridis, P.; Basu, A.; Lister, P.F. Web3D and augmented reality to support engineering education. *World Trans. Eng. Technol. Educ.* **2004**, *3*, 11–14.
8. Pathomaree, N.; Charoenseang, S. Augmented reality for skill transfer in assembly task. In *ROMAN 2005, Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication, Nashville, TN, USA, 13–15 August 2005*; IEEE: Piscataway, NJ, USA, 2005; pp. 500–504.
9. Arbeláez, J.C.; Roberto, V.; Gilberto, O.G. Haptic augmented reality (HapticAR) for assembly guidance. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2019**, *13*, 673–687. [CrossRef]
10. Tang, A.; Owen, C.; Biocca, F.; Mou, W. Experimental evaluation of augmented reality in object assembly task. In Proceedings of the International Symposium on Mixed and Augmented Reality, ISMAR, Darmstadt, Germany, 1 October 2002; pp. 265–266.
11. Woll, R.; Damerau, T.; Wrasse, K.; Stark, R. Augmented reality in a serious game for manual assembly processes. In Proceedings of the 2011 IEEE International Symposium on Mixed and Augmented Reality-Arts, Media, and Humanities (ISMAR-AMH), Basel, Switzerland, 26–29 October 2011; pp. 37–39.
12. Wang, X.; Kotranza, A.; Quarles, J.; Lok, B.; Allen, B.D. A pipeline for rapidly incorporating real objects into a mixed environment. In Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05), Vienna, Austria, 5–8 October 2005; pp. 170–173.
13. Valentini, P.P. Interactive virtual assembling in augmented reality. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2009**, *3*, 109–119. [CrossRef]
14. Jiang, S.; Li, L.; Xu, H. Stretchable E-skin patch for gesture recognition on the back of the hand. *IEEE Trans. Ind. Electron.* **2019**, *67*, 647–657. [CrossRef]
15. Liu, L.; Liu, Y.; Zhang, J. Learning-based hand motion capture and understanding in assembly process. *IEEE Trans. Ind. Electron.* **2018**, *66*, 9703–9712. [CrossRef]
16. Li, Y.; Shen, X. A real-time collision detection between virtual and real objects based on three-dimensional tracking of hand. In Proceedings of the 2010 International Conference on Audio Language and Image Processing, ICALIP, Shanghai, China, 23–25 November 2010; pp. 1346–1351.
17. Lee, M.; Green, R.; Billinghurst, M. 3D natural hand interaction for AR applications. In Proceedings of the 23rd International Conference on Image and Vision Computing New Zealand (IVCNZ 2008), Christchurch, New Zealand, 26–28 November 2008; Volume 1, pp. 26–28.
18. Wang, Z.; Shen, Y.; Ong, S.K.; Nee, A.Y.-C. Assembly design and evaluation based on bare-hand interaction in an augmented reality environment. In Proceedings of the International Conference on CyberWorlds, Bradford, UK, 7–11 September 2009; Volume 21, pp. 7–11.
19. Choi, J.; Park, H.; Park, J.; Park, J.-I. Bare-hand-based augmented reality interface on mobile phone. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality 2011, Science and Technology Proceedings, Basel, Switzerland, 26–29 October 2011; pp. 275–276.

20. Figueiredo, L.; Dos Anjos, R.; Lindoso, J. Bare hand natural interaction with augmented objects. In Proceedings of the 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Adelaide, SA, Australia, 1–4 October 2013; IEEE: Piscataway, NJ, USA; pp. 1–6.
21. Hilliges, O.; Kim, D.; Izadi, S. Grasping Virtual Objects in Augmented Reality. U.S. Patent 9,552,673, 24 January 2017.
22. Buchmann, V.; Violich, S.; Billinghurst, M.; Cockburn, A. FingARtips: Gesture based direct manipulation in augmented reality. In Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia (GRAPHITE '04), Singapore, 15–18 June 2004; pp. 212–221.
23. Reifinger, S.; Wallhoff, F.; Ablassmeier, M.; Poitschke, T.; Rigoll, G. Static and dynamic hand-gesture recognition for augmented reality applications. In *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 728–737.
24. Lee, T.; Höllerer, T. Hybrid feature tracking and user interaction for markerless augmented reality. In Proceedings of the IEEE Virtual Reality, Reno, NV, USA, 8–12 March 2008; pp. 145–152.
25. Zhu, T.; Zhou, Y.; Xia, Z. Progressive filtering approach for early human action recognition. *Int. J. Control. Autom. Syst.* **2018**, *16*, 2393–2404. [CrossRef]
26. Du, G.; Chen, M.; Liu, C. Online robot teaching with natural human–robot interaction. *IEEE Trans. Ind. Electron.* **2018**, *65*, 9571–9581. [CrossRef]
27. Imbert, N.; Vignat, F.; Kaewrat, C.; Boonbrahm, P. Adding physical properties to 3D models in augmented reality for realistic interactions experiments. *VARE* **2013**, *25*, 364–369. [CrossRef]
28. American Society of Mechanical Engineers. Special committee on standardization of therbligs, process charts, their symbols. In *ASME Standard Operation and Flow Process Charts*; American Society of Mechanical Engineers: New York, NY, USA, 1947.
29. Dong, J.; Tang, Z.; Zhao, Q. Gesture recognition in augmented reality assisted assembly training. *J. Phys. Conf. Series. IOP Publ.* **2019**, *1176*, 032030. [CrossRef]
30. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
31. Khotanzad, A.; Hong, Y.H. Invariant image recognition by Zernike moments. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 489–497. [CrossRef]
32. Dong, J.; Xia, Z.; Yan, W.; Zhao, Q. Dynamic gesture recognition by directional pulse coupled neural networks for human-robot interaction in real time. *J. Vis. Commun. Image Represent.* **2019**, *63*, 102583. [CrossRef]
33. Liu, L.; Shao, L. Learning discriminative representations from RGB-D Video Data. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Beijing, China, 3–9 August 2013.
34. Marcel, S.; Bernier, O.; Viallet, J.-E.; Collobert, D. Hand gesture recognition using input/ouput hidden-Markov models. In Proceedings of the 4th International Conference on Automatic Face and Gesture Recognition (AFGR), Grenoble, France, 28–30 March 2000; pp. 456–461.
35. Cirujeda, P.; Binefa, X. 4DCov: A nested covariance descriptor of spatio-temporal features for gesture recognition in depth sequences. In Proceedings of the 2014 2nd International Conference on 3D Vision, Tokyo, Japan, 8–11 December 2014; Volume 1, pp. 657–664.
36. Liu, M.; Liu, H. Depth context: A new descriptor for human activity recognition by using sole depth sequences. *Neurocomputing* **2016**, *175*, 747–758. [CrossRef]
37. Azad, R.; Asadi-Aghbolaghi, M.; Kasaei, S. Dynamic 3D hand gesture recognition by learning weighted depth motion maps. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 1729–1740. [CrossRef]
38. Zheng, J.; Feng, Z.; Xu, C.; Hu, J.; Ge, W. Fusing shape and spatio-temporal features for depth-based dynamic hand gesture recognition. *Multimed. Tools Appl.* **2016**, *76*, 20525–20544. [CrossRef]
39. Kim, T.-K.; Kittler, J.; Cipolla, R. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Trans. Pattern Anal. Mach. Intell* **2007**, *6*, 1–14. [CrossRef] [PubMed]
40. Kim, T.; Wong, S.; Cipolla, R. Tensor canonical correlation analysis for action classification. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
41. Lui, Y.M.; Beveridge, J.; Kirby, M. Action classification on product manifolds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 833–839.
42. Liu, L.; Shao, L. Synthesis of spatio-temporal descriptors for dynamic hand gesture recognition using genetic programming. In Proceedings of the 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Shanghai, China, 22–26 April 2013; pp. 1–7.
43. Lui, Y. Tangent bundles on special manifolds for action recognition. *IEEE Trans. Circ. Syst.Video Technol* **2012**, *22*, 930–942. [CrossRef]
44. Sanin, A.; Sanderson, C.; Harandi, M.T.; Lovell, B.C. Spatio-temporal covariance descriptors for action and gesture recognition. In Proceedings of the 2013 IEEE Workshop on Applications of Computer Vision (WACV), Clearwater Beach, FL, USA, 15 January 2013; pp. 103–110.
45. Laptev, I. On space-time interest points. *Int. J. Comput. Vis.* **2005**, *64*, 107–123. [CrossRef]
46. Laptev, I.; Marszalek, M.; Schmid, C.; Rozenfeld, B. Learning realistic human actions from movies. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 24–26.

47. Hao, Z.; Zhang, Q.; Ezquierdo, E.; Sang, N. Human action recognition by fast dense trajectories. In Proceedings of the 21st ACM international conference on Multimedia, 21 October 2013; pp. 377–380.
48. Wang, H.; Kläser, A.; Schmid, C.; Liu, C.L. Dense trajectories and motion boundary descriptors for action recognition. *Int. J. Comput. Vis.* **2013**, *103*, 60–79. [CrossRef]
49. Kulkarni, K.; Evangelidis, G.; Cech, J.; Horaud, R. Continuous action recognition based on sequence alignment. *Int. J. Comput. Vis.* **2015**, *112*, 90–114. [CrossRef]