KIT- AR

System Design Proposal for Assembly error detection

Abstract:

This system provides a computational model that focuses on the combination of Augmented Reality (AR) with machine learning for detecting errors during a manual assembly process. Our approach combines AR graphics with adaptive guidance to try and mitigate errors during the assembly process. The system will identify where the error has occurred and also specify the type of errors.

Assembly error detection:

The ability to combine abstract concepts and 3D spatial information in the context of real-world objects makes AR an ideal tool for training in situations which require manipulation of objects, such as manual assembly of custom PCs.

The drawings include provided a better understanding of the system

Fig 1: Conceptually illustrates the flowchart of an assembly system during deployment
Fig 2: Illustrates the process for performing error detection and correction

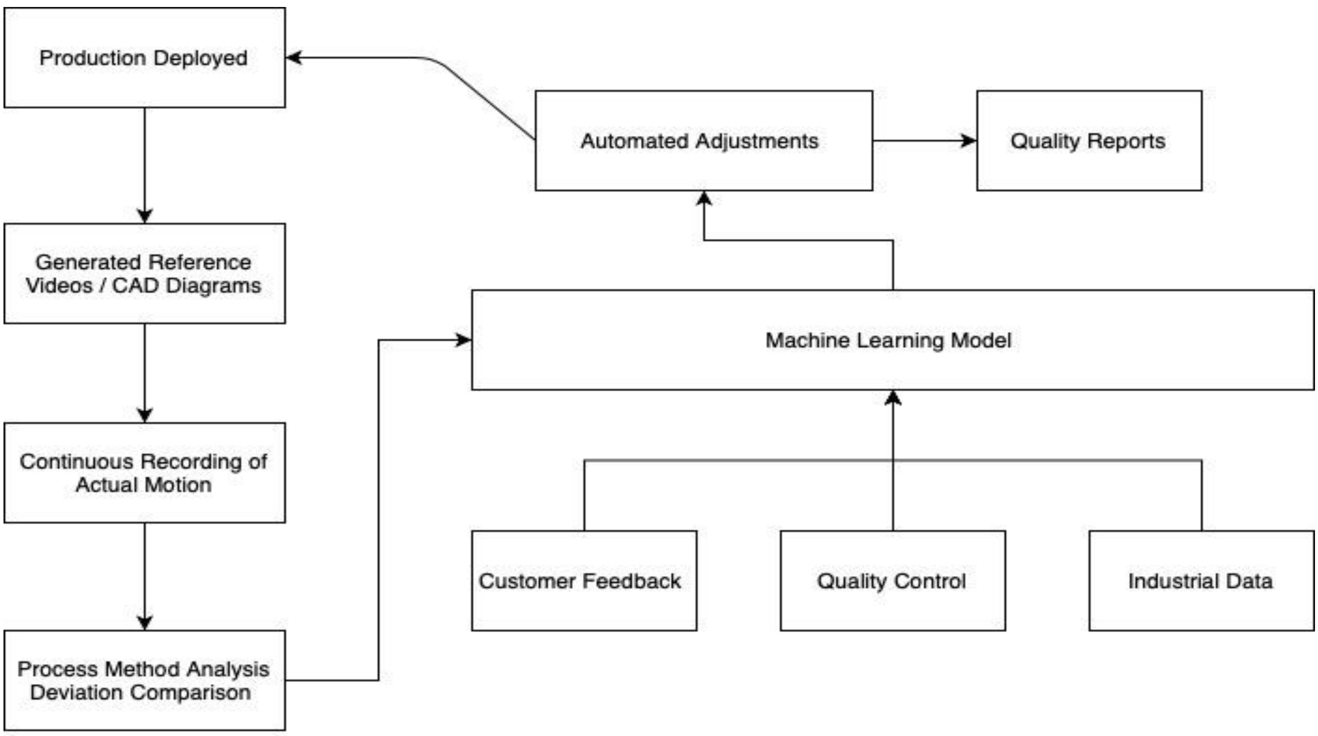

Fig -1

Description of Fig – 1

Figure 1 illustrates a flowchart for implementing a production line deployment.

1. **Production Deployment** :

In this step, the production deployment is commenced. The workflow of an example production lines typically includes various steps(nodes) in which the operator assembles a PC. The nodes are organised sequentially as the logical steps involved in the assembly of a PC.

2. **Reference Video** and **CAD Diagrams:**

In this step, one / more reference videos or CAD diagrams are used to provide manufacturing/assembly instructions through the AR interface to a particular node. That is, each node in the workflow can receive instructions in the assembly process which also helps with error checking. These videos can also be model specific.

3. **Continuous Recording of Actual Motion:**

In this step, motion data is recorded from the operator and stored. This data is later used in the Assembly error detection process. This data can be recorded with an MYO device. This data is captured for each node in the assembly process.

4. **Process Method Analysis Deviation:**

In this step, we try to identify if there are any deviations from our reference video and MYO data from our reference video. We can assess any deviations using AI/Machine learning models that can be used to classify deviations from our comparison model and make inferences on how the assembly may be affected. The comparison can be done per node or as a whole.

5. **Machine Learning Model:**

A machine learning model is called on this step based on a collection of various input data. They may include examples of ideal assembly processes, CAD diagrams and MYO data. The model can be augmented from customer feedback, industrial or domain-specific data and quality control data.
* more information about the model is described below.

6. **Automated Adjustments:**

Once deviations from the comparison model have been detected from the steps above, automated adjustments can be made to improve the assembly process at a per-node level. For example,
Video instructions or animate graphics can be displayed on the AR headset to help the operator in that particular assembly process. These instructions can be done in real-time or during fixed intervals.
These adjustments can also be used to provide quality reports that describe the quality of various parts based on how they might have been assembled.

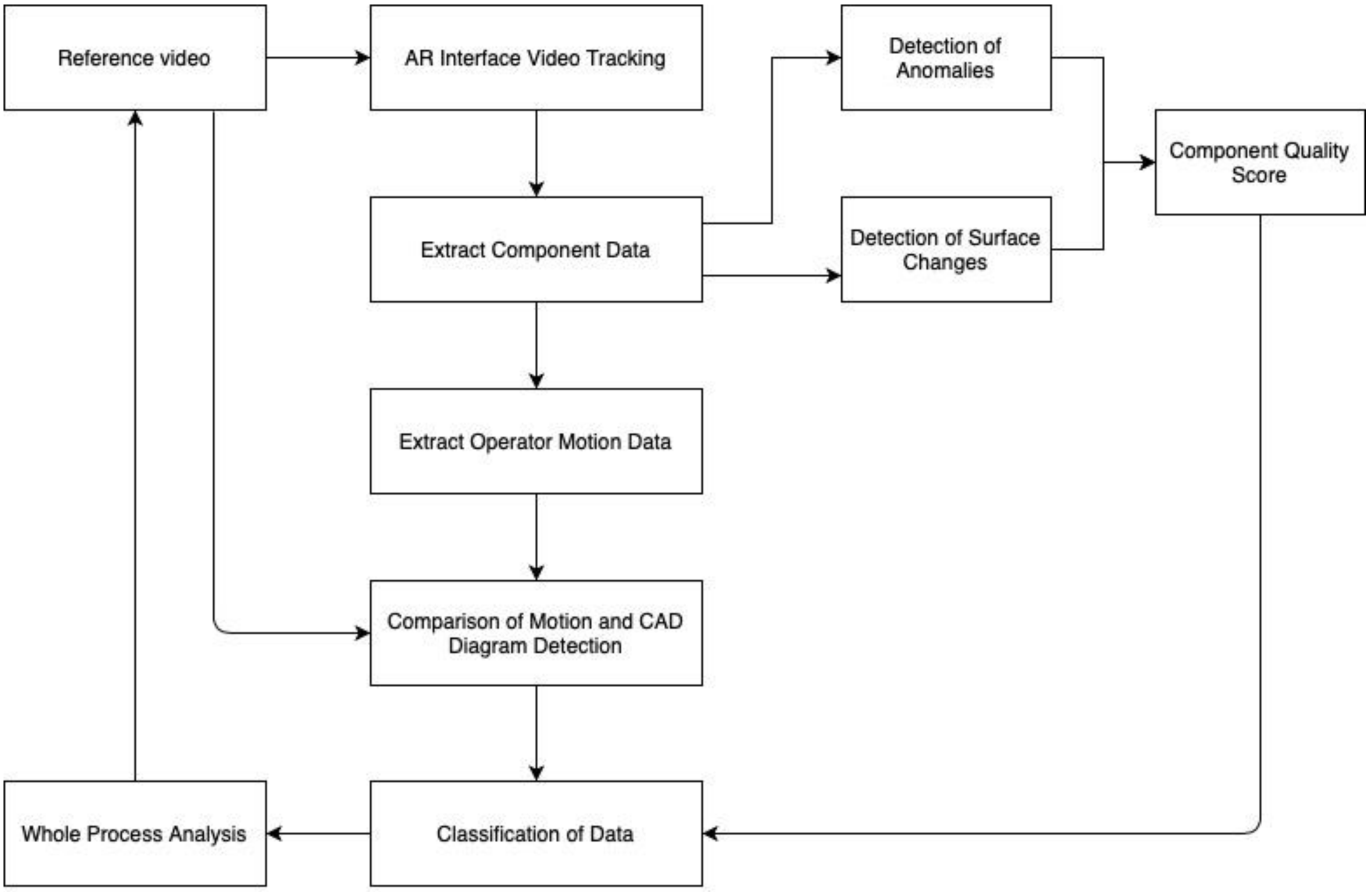

Fig - 2

Description of Fig - 2

Fig 2, illustrates the error detection process for performing error detection analysis.

1. Beginning at the first step (**Reference Video**) :

A process of using a reference video is implemented. This video would be the correct assembly process of an operator assembling the pc.

2. In the next step (**AR Interface Video Tracking)**:

Video tracking of the operator is performed using an AR Interface. This video is then analysed and data is extracted from this video

3. In the next two steps (**Extract Component Data** and **Extract Operator Motion Data**):

Processing is performed to analyse the video from the AR headset
E.g., background extraction can be performed to isolate the movements/ components in the video. Once background extraction is completed, the processed video contains only motion or video data relating to the assembly operator and the components/parts involved in the assembly of the PC.

As illustrated by the diagram additional processing operations such as anomaly detection, detection of surface changes, part classification and quality scoring are implemented.

The video processing steps are performed using various image processing/signal processing techniques. Along with AI/Machine learning algorithms and/or classifiers to perform the above operations(more information referenced later)

4. In the next step (**Comparison of Motion and CAD diagram Detection)**:

We compare the motion data from the AR device with the original reference video and the original CAD diagrams. This can involve motion capture data from one or more operators and can also include MYO device data.
This motion comparison can further predict if there are any more errors in their assembly.

5. In the next step (**Classification of Data)**:

Variance/Quality classifications for various parts can be performed. It is in this step that we classify the components into different tiers such as

- Missing components - screws and connectors.
- Incorrectly installed components – screws and connectors which are lost.
- Incorrect assembly – connectors installed in the wrong place.

After classifications and differences have been made and labelled, we can proceed with the next step

6. Final Step (**Whole Process Analysis**):

This step is an analysis of the entire workflow.
Based on the classifications/differences determined in the previous steps, we can display correction/error graphics on the AR display to guide the operator on how to fix these errors.

Machine Learning Model

Machine learning model such as Convolutional Neural Networks(CNN's) and support vector machines can be used.
We can also use classification models such as K-means clustering algorithm and DBSCAN for outlier detection.

First, CNN's can be used to classify features of an operators hands and articles of manufacture in different configurations

Second, reinforced learning and RL agents can be used and rewarded for achieving desired outcomes both from CNN classifications and predefine desirable outcomes

Third, Generative Adversarial Networks can be used to choose between confliction RL agents. This would require minimal human intervention.

Finally, RNN's can take the winning RL agents as input nodes o create a feedback system to continue to improve our machine learning model.

Inputs for our system:

The inputs we would require would be reference videos of the assembly process by an ideal operator and CAD designs for the PC being build. These CAD designs can be used to identify the system model being build and also for error detection and classification

We would also be storing large amounts of data which would be streamed from the sensors and AR headsets. Therefore, we would require the integration Apache Kafka, Apache Storm, and MongoDB in big data processing systems so that the data can be promptly processed, stored, and presented in real-time.