Article:

Oyekan, J. (2019) Utilising low cost RGB-D cameras to track the real time progress of a manual assembly sequence. Assembly Automation. ISSN 0144-5154

https://doi.org/10.1108/AA-06-2018-078

Utilising low cost RGB-D cameras to track the real time progress of a manual assembly sequence

Abstract

Purpose
The purpose of this paper is to explore the role that computer vision can play within new industrial visions such as Industry 4.0 and in particular to support production line improvements to achieve flexible manufacturing. As Industry 4.0 requires 'big data' it is accepted that computer vision could be one of the tools for its capture and efficient analysis. RGB-D data gathered from real-time machine vision systems such as Kinect ® can be processed using computer vision techniques.

Design
This research exploits RGB-D cameras such as Kinect® to investigate the feasibility of using computer vision techniques to track the progress of a manual assembly task on a production line. Several techniques to track the progress of a manual assembly task are presented. The use of CAD model files to track the manufacturing tasks is also outlined.

Findings
This research has found that RGB-D cameras can be suitable for object recognition within an industrial environment if a number of constraints are considered or different devices/techniques combined. Furthermore, through the use of a HMM inspired state-based workflow, the algorithm presented in this paper is computationally tractable.

Originality
Processing of data from robust and cheap real-time machine vision systems could bring increased understanding of production line features. In addition new techniques that enable the progress tracking of manual assembly sequences may be defined through the further analysis of such visual data. The approaches explored within this paper make a contribution to the utilisation of visual information 'big data' sets for more efficient and automated production.

Index Terms—: Industry 4.0, RGB, depth, computer vision, manufacturing

1.0 **Introduction**

Recent visions for computer networked Manufacturing, such as Industry 4.0 and the Industrial Internet, promote the benefits of providing production line flexibility for the manufacture of customized and personalized products. In the process of realizing such manufacturing visions, large amounts of data will be obtained by sensors and then transmitted, analysed and stored. Through the use of computer vision techniques, data could be extracted from manufacturing lines in a non–intrusive way (Stork, 2015) through the utilisation of camera type sensors. This concept, as illustrated in Figure 1, is further supported by General Electric (Hryniewicz et al., 2015). In Figure 1, real time machine vision could gather data from workstations for entry into an industrial data system. The analysis of the data captured in the industrial data system could then be used to compose or augment the base “Big Data” set. Once processed, this data could be presented visually enabling the identification of potential changes to manufacturing processes. This cycle could enhance the continuous improvement process of the manufacturing effort at the workstation level.

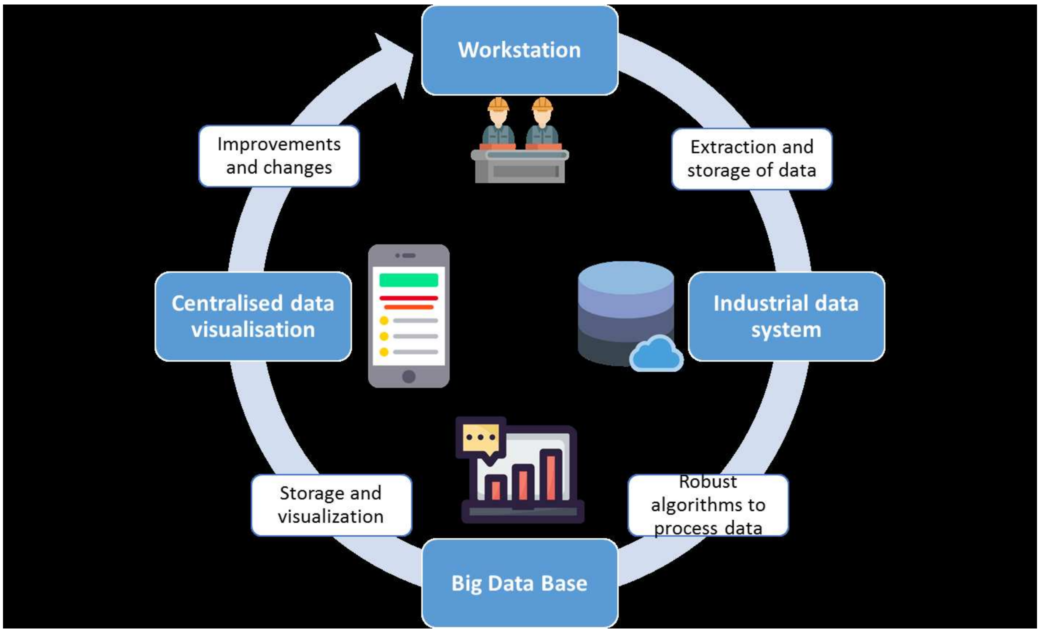

Workstation
Extraction and storage of data
Industrial data system
Robust algorithms to process data
Big Data Base
Storage and visualization
Centralised data visualisation
Improvements and changes

Figure 1: An approach for the extraction of value from data collected from manufacturing workstations

This paper focuses on the use of a computational agent to understand the tasks a human is performing on a workpiece. The novelty of this approach is that until recently there only been a relatively limited amount of literature that attempted research in this direction, especially in the context of industry 4.0. A HMM (Hidden Markov Model) inspired workflow has been used to encode a well-defined manual assembly task. The HMM was used because every product assembly could theoretically be reduced to a sequence of steps that could be encoded as a HMM. The novelty to this approach lies in developing a computationally tractable algorithm that combines HMM theory with computer vision to achieve tracking of a manual assembly in real time. This enables the use of machine learning techniques to gain insights into conditions that could boost the productivity of workers performing manual assembly tasks as well as offer feedback to them relating to their work. Furthermore, the research in this paper initiates a discussion on the non-intrusive collection of data on the human element of manufacturing which is as important as collecting data from machines on the shop floor for further processing.”

2.0 Relevant Literature

In order to be able to track the progress of an assembly task as well as assess quality, at least two high level computational processes are required: object recognition; assembly state recognition.

2.1 Object recognition:

Object recognition is very popular in manufacturing as it is advantageous for robots and automated machines to be able to recognize the parts or work pieces that they process. Most of the approaches of object recognition have focused on the detection of object features both in 2D and 3D, such as boundaries, contour, colour, as well as the development and combination of feature-based rules to estimate the objects being observed.

Using 20 significant RGB-D (RGB Depth) datasets Cai et al. (2016) showed how parameters such as object speed, object type and colour could be utilised in object recognition. In addition Fu et.al. (2017) investigated issues of depth perception in their approach involving video segmentation of RGB-D video. Barron and Malik (2016) demonstrate a method which enables the recovery of image features such as brightness, shape, reflectance and shading from a single image taken from an RGB-D sensor; a development of their previous method called SIRFS (“Shape, Illumination and Reflectance From Shading”) (Barron and Malik, 2015). It was realized by Barron and Malik (2016) that SIRFS did not perform well on images containing occlusions and variations in illumination; thus they created the scene-SIRFS method, a more robust and accurate method based on mixture of shapes and illuminations.

Another technique was developed by Lowe. (1999) called Scale Invariant Feature Transform (SIFT). This technique was invariant to object rotations, translations and changes in brightness level, while still producing accurate results. In a development of the method of Lowe (1999) Rothganger et al. (2006) proposed to model and recognise an object by redesigning the invariants. This approach uses the local image descriptors and the luminosity and colour of an image to identify the object. In research by Matas and Obdrzalek (2004) the 2D contour of the objects and their features such as lines and circles were used as well as a method to map an object in 3D from an image.

Wu and Bainbridge-Smith (2011) explained that a RGB-D camera such as Kinect® is a fast and accurate way to extract 3D information of an object compared to other optical devices. This results in a real time 3D point-cloud database which is then post-treated in order to identify a part. As in 2D object recognition, object features often provide a mechanism to understand and recognise an object. For example, Gupta et al. (2014) used depth features as well as used object contour detection, and segmentation in their work on RGB-

D cameras. Gupta et al. (2015) also followed a similar approach and presented algorithms for 3D contour detection and hierarchical segmentation. Prior obtained datasets enabled them to train a classifier to classify the objects detected.

Another assumption that can be used in 3D object detection and recognition is depth saliency. The central assumption of this approach is that the salient object tends to stand out from its surroundings (Ju et al., 2015). Ju et al. (2015) emphasised that this approach is suitable for fast and precise object recognition and can potentially be utilized for tracking the progress of an assembly task. Borji et al. (2015) also made the point that saliency detection is an important focus for future research in this area and put forward a benchmark for object detection formed from the qualitative and quantitative assessment of 41 models. Using the saliency assumption, spatiotemporal background priors were also proposed by Xi et al. (2016).

Wang and Posner (2015) propose a sliding window approach to the analysis of 3D point cloud data, with a search space composed of objects positioned at any orientation, utilising a voting system. An approach often neglected in 3D object detection research, due to inefficiency in operation, the sliding window when used with voting is claimed by Wang and Posner (2015) to be comparable in efficiency to sparse convolution. The approach proceeds by transforming the point cloud into a 3D grid then a fixed dimension feature vector is mapped for each occupied cell within the grid (empty cells are mapped to zero). The resulting entity is a feature grid through which the detection window transitions in all three dimensions for each angle of rotation. In essence detection windows passing from different orientations cast votes on occupied cells in the matrix.

Even though the above techniques have been used for decades, new approaches such as deep learning offer greater robustness in object detection and recognition tasks. For example, they are more resilient to illumination changes and object variability (Tang et al., 2017, Ballester and Araujo, 2016). This is often achieved via. The collection of a large data set from the problem domain on which a neural network is trained in order to derive mathematical models (as in the case of CNN) or “sequence rules” (as in the case of RNN (Sutskever, 2014)). The volume of the dataset ensures that all or most possible variations of the domain are taken into consideration.

Sedgahat et al. (2017) utilised deep learning in 3D object classification; this approach can also be used with object detection techniques such as those utilising a 3D sliding window method. A 3D voxel grid representation of an image is fed into the algorithm of Sedgahat et al. (2017) which is based on VoxNet (Maturana and Scherer, 2015). Within the training stage the object orientation is noted and made an explicit part of that phase. The 3D object is rotated multiple times and presented to the network, voting is then utilised to determine the object’s classification. It was found that by requiring the network to capture the orientation, an improved classification could be achieved. Qi et al. (2017) introduce a Neural Network approach to object classification named PointNet. This approach is able to process point cloud representations directly without the need for a voxel grid or image collection. Qi et al. (2017) claim that this approach addresses three tasks involved with 3D object recognition: object classification; part segmentation and semantic segmentation.

Even though the aforementioned machine learning concepts are often used in computer vision processing tasks such approaches are best suited to situations where the rules cannot be easily defined (Amazon, 2017). In the case of this paper, machine learning was not used. This is because most assembly tasks are well defined and follow a sequence of steps. Hence, it is possible for a human to write the sequence rules in algorithmic form. Embedding these rules in a computational agent equipped with an RGB-D camera enables the achievement of assembly state recognition.

2.2 Assembly State Recognition:

The unique capabilities of RGB-D cameras enable them to be used to map a room, a shop floor and a work bench as well as capture the gestures of workers efficiently in different environments (Bu et. al, 2016).

Because of their RGB stream, RGB-D cameras can be used for typical real-time machine vision inspection tasks while their Depth stream enables the possibility to extend typical 2D inspection tasks to the third dimension by making use of 3D geometrical features.

Due to these capabilities, machine vision offers the potential to be used as a substitute in inspection tasks typically performed by humans. For example, Sture et al. (2016) showed that Salmon deformities and wounds can be identified using real-time machine vision achieving a detection rate of e 86% for deformities and 89% for wounds. Furthermore, Schmitt et al. (2015) designed a real-time machine vision system that can detect significant quality deficiencies in fibre-reinforced plastics under certain conditions. Li and Huang (2015) developed a method to inspect tyres, gathering geometrical data from images these authors then assessed the quality using tyre features identified from images.

Machine vision could also be extended to the real-time monitoring of tasks carried out by humans for inspection as well as digital assisted assembly. In literature, most of the research involves the use of Augmented Reality (AR) to provide instructions and aid to humans during manufacturing tasks. For example, Radkowski and Oliver (2013) used natural feature tracking in order to realise the tracking of rigid objects for an on-site assembly assistance system. The tracking system tracked multiple circuit boards without the need of markers. An AR system was used to provide feedback to the worker on what part to pick up next. A similar approach was followed in Radkowski (2015, 2016) where they went on to develop a 3D tracking method, also with AR, for use in a mechanical engineering assembly environment with different degrees of complexity.

The use of RGB-D cameras and 3D part recognition for assembly state estimation is the focus of Gu et al. (2018). The goal of the work is skill capture for replication of manufacturing assembly actions by robot. Gu et al. (2018) find that their Portable Assembly Demonstration (PAD) system is able to generate an assembly script suitable for implementation by a robot. Future work with this approach will involve development of the technique to address the improved detection of occluding and occluded objects.

Funk et al. (2015) introduced a combined projection and AR (Augmented Reality) method for the order picking process within a warehouse environment. The research, while utilising marker assisted methods, also envisages using an AR generated marker in future developments. Bi and Kang (2014) present a technique to reconstruct surfaces applied to flexible machining systems based on feedback from vision sensors. Aehnelt et al. (2014) examined the challenges related to activity detection in an industrial setting and concluded that the need to be able to detect discrete tasks within a workflow still exists as a required research target. Towards solving this challenge, Hartmann (2011) identified 3 methods for the recognition of discrete assembly tasks performed by a worker: statechart model, Hidden Markov Model (HMM) and Dynamic Bayesian Network (DBN). The statechart model worked for the recognition of tasks where uncertainty does not affect the data. HMM and DNB have delivered highly positive results in terms of task recognition at 95%. DBN achieved better results; however, due to its complexity, the time required to analyse data and convert it into an appropriate format was unacceptable in practice. The DBN technique also requires that context based knowledge of the task is acquired before the implementation of a solution.

In manufacturing lines that involve manual assembly context based knowledge of the tasks can be readily obtained from available manuals. The major challenge is to ensure that this knowledge can be converted easily into a computational format with little set up time when deployed across many domains. Also, in order to ensure widespread adoption in manufacturing industries, the developed approach must be affordable for Small and Medium sized Enterprises.

This paper aims to investigate if the aforementioned challenge can be partly solved through the use of HMM-inspired workflows and natural feature tracking through the use of object colour features and low cost RGB-D cameras.

Along their use in gaming low cost RGB-D cameras such as the Kinect are becoming popular for use in manufacturing applications. Such cameras have been specifically designed to track the skeleton of a human, research is still ongoing for their use in robust and accurate object detection and recognition in manufacturing.

As a result, despite the relevant work found in literature, object recognition in the form of tracking the progress of an assembly task still requires more research. Towards this goal, this paper presents an investigation of combined 2D and 3D techniques for assembly state tracking and recognition.

3.0 Methodology

In order to achieve widespread adoption in manufacturing enterprises, the Microsoft Kinect RGB-D camera was used in this work. An object's characteristics in terms of colour and geometry can be extracted from the data streams of the Kinect using both 2D and 3D techniques drawn from literature. In terms of 2D techniques the work of Matas and Obdrzalek (2004) appears to be robust as it is able to deal with occlusions, in addition the work of Gupta et al. (2014) is insightful as it provides approaches for object contour recognition and datasets creation. The work of Prabhu et al. (2015) outlines a method to enable the tracking of the progress of a manual wheel assembly process.

Building on the above techniques, this research aims to:

- Investigate techniques that enable the tracking of workpieces based on their features (2D and 3D).
- Investigate techniques that enable the extraction of manual assembly progress through the tracking of workpiece feature changes
- Use readily available context based knowledge in the creation of HMM-inspired workflows for tracking manual assembly tasks.
- Validate the above techniques with a number of use case scenarios

In order to ensure industrial relevance, a number of visits were made to an electronic circuit board manufacturer to gather data and understand the requirements of the shop floor managers. These visits involved the installation of the Kinect® on all workstations of a manual manufacturing line. Datastreams containing both RGB and depth data were obtained through the use of Kinect®. A CAD model of the facility was also produced as part of this research which was an important asset in the identification of the key features for use in the development of the computer vision techniques.

3.1 **Exploiting 2D Natural features on objects**

3.1.1 Applying Colour and distance thresholds

In order to ensure easy setup and possibility of deploying to various domains, the first 2D technique investigated was based on the colours of workpieces in the considered domain. The image retrieved from the Kinect® has a resolution of (640 * 480) where each pixel is defined by an RGB value. Through the use of pre-defined thresholds, it is possible to identify objects through their colours. By using a distance measure, a combination of colours in a radius could be used to identify an object as well as the progress in its assembly state.

In setting the parameters of colour thresholds and distance to an appropriate context informed value, the program was able to detect and track objects with a particular RGB as well as the progress in an assembly task. Figure 2 shows the recognition of parts within a scene. Initial tests utilized coloured blocks.

The drawback of this technique is that like all computer vision applications, it is easily affected by lighting conditions.

As a result, and in order to get consistent results, controlled lighting was used as well as a white background as shown in Figure 2.

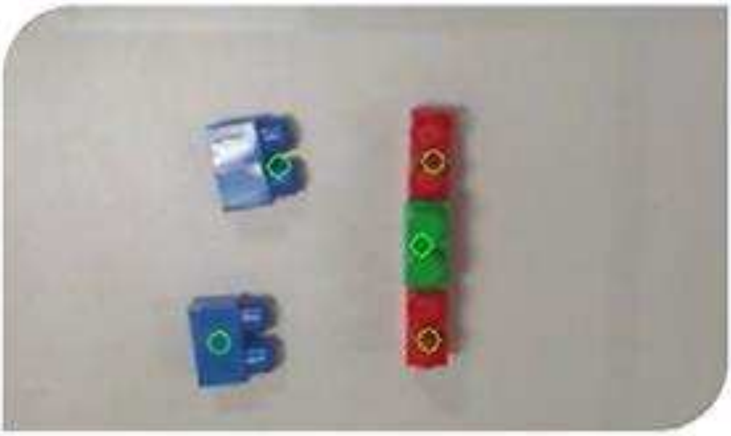


Figure 2: Parts recognition and tracking using colour feature

3.1.2 Applying shape recognition

In order to make the colour recognition and distance measure approach more robust, a second technique that relied on the 2D geometrical shape features of an object was investigated. This technique detected and used edges derived from the image to reconstruct the contours of objects in the scene. Using context informed minimum area and perimeter thresholds, it was possible to filter out unnecessary objects in the scene that do not correspond to the task being tracked. Figure 3 highlights the contours of three simple metallic parts within an industrial environment. In this way the shape recognition technique tracks parts and return the location of the centre of mass of the object in real time. The location of the centre of mass of detected objects is then used in the distance measure already described in 3.1.1 to track assembly state of workpieces.

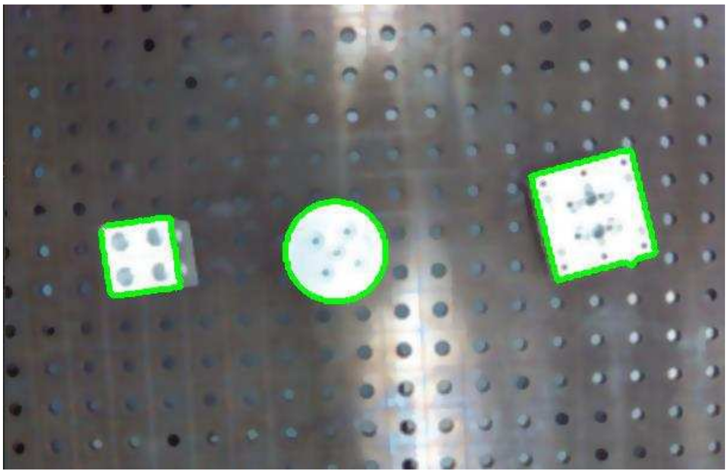


Figure 3**:** Contours of 3 metallic parts

3.2 **Investigation of 3D Solutions**

In the Kinect® two components, the infrared projector and the infrared (IR) CMOS (Complementary Metal-Oxide-Semiconductor) sensor, enable depth data to be obtained. The value of one pixel of the depth image corresponds to the distance between the sensor and the location where the IR ray is reflected. These values were normalized in order to get a range of data from 0 to 255, where 0 is black and 255 is white. As a result, it was possible to get a greyscale image of the observed scene.

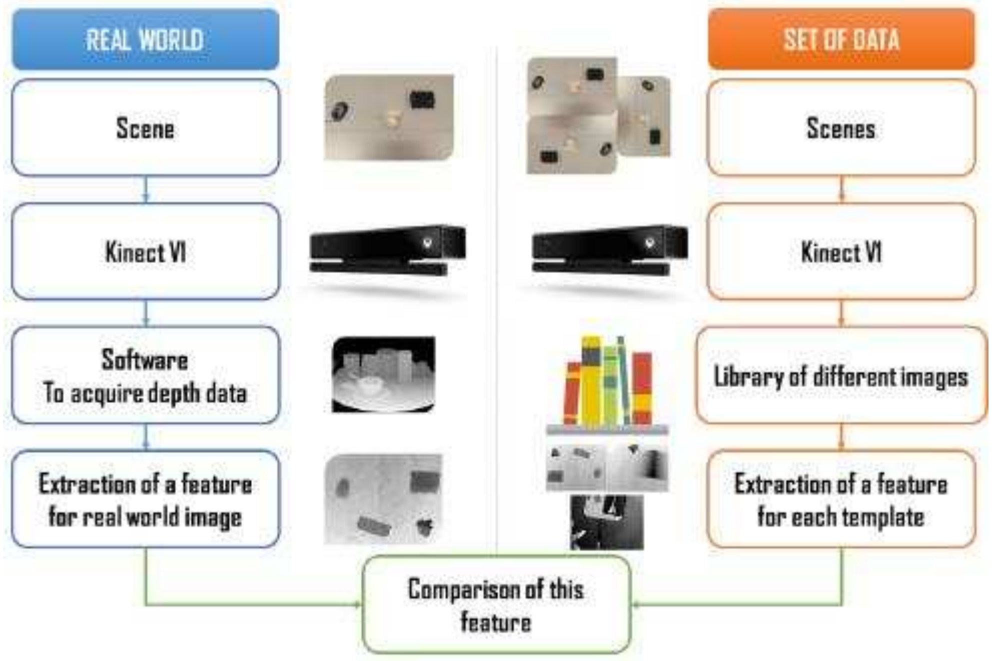


Figure 4: Methodology for the comparison of two scenes using a stored library set of object templates

Using the greyscale images of the observed scene, two techniques were investigated for recognizing 3D objects: Sum based algorithm and Full image template matching.

3.2.1 Sum based algorithm

In the first technique, a prior set of data representing the objects to be recognized was created. The data of each object was then reduced to a sum of all the depth elements and used as a sum template s. This resulted in a library set, S, of stored object sum values (Figure 4).

During real time operation, the real life images obtained from the scene were converted to a sum value i and compared to the stored sum templates. The difference between i and any of the stored sum templates s was used to determine which sum template best matches with the real world image. Hence, the smaller the difference, the better the match.

This implemented technique is robust against rotation or orientation of observed objects. However, the approach requires the Kinect® to be set up in the same position and height all the time. Indeed, if the Kinect® is set up at a different height between the real world capture and the template acquisition, the comparison would not produce any result as the objects in the scene would appear either smaller or bigger than the actual case.

3.2.2 Full image template matching

The second technique investigated is the full image template matching. This technique involves sliding a full image template that represents the object to be recognised across the actual image.

The templates used in this work came from CAD (Computer Aided Design) models that can be readily found on the open source 3D repository called Thingiverse. The methodology used for template matching using the CAD model is described in Figure 5. These models are normally in STL (Standard Triangle Language) format; STL represents an unstructured triangulated surface composed of faces and vertices.

In order to generate a grayscale image, it was necessary to keep only the top surface image of an object. An algorithm that stored the top values of an STL file in a matrix was created. This matrix represents the top surface of the CAD parts as ‘seen’ from the viewpoint of a Kinect® sensor. The algorithm then converted the matrix into a grayscale image for full image template matching.

The advantage of this approach is that downloaded CAD models can be automatically rotated as well as resized in order to obtain various orientations and distances from the Kinect®. This results in a template library of different grayscale images that correspond to different orientations and distances for a particular object. The algorithm uses prior generated and stored templates to find the best match in a real time image by sliding the stored template across the scene. The problem with this approach is that as the library of stored templates increases, it becomes less computationally tractable.

Nevertheless, using stored CAD models of a part has an industrial relevance because most manufactured parts would have a CAD model. This enables the novel use of stored domain knowledge.

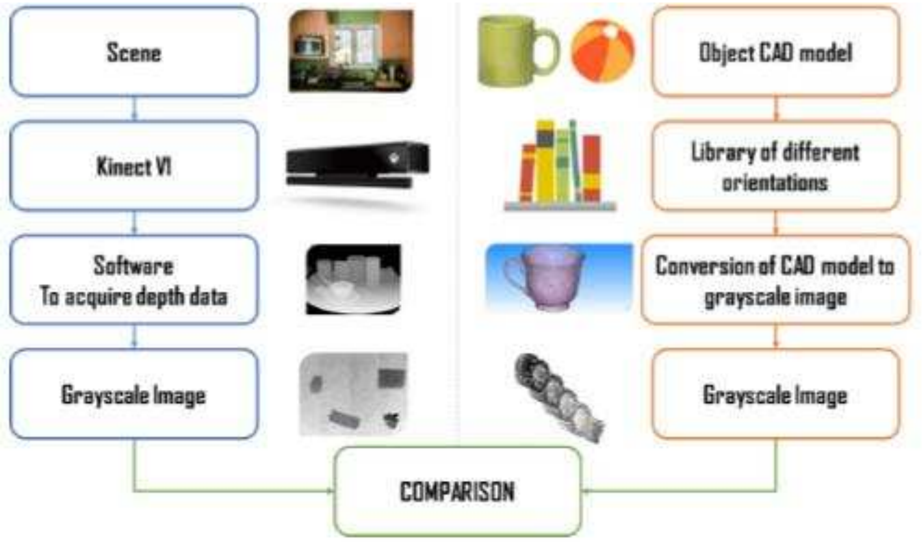


Figure 5: Methodology for template matching using CAD model

In order to test this technique, a V8 cylinder model as well as a Stanford bunny's CAD model (Figure 6) were downloaded and processed according to the right hand side of the methodology shown in Figure 5. Consequently, several templates of both models in different orientations were generated and stored.

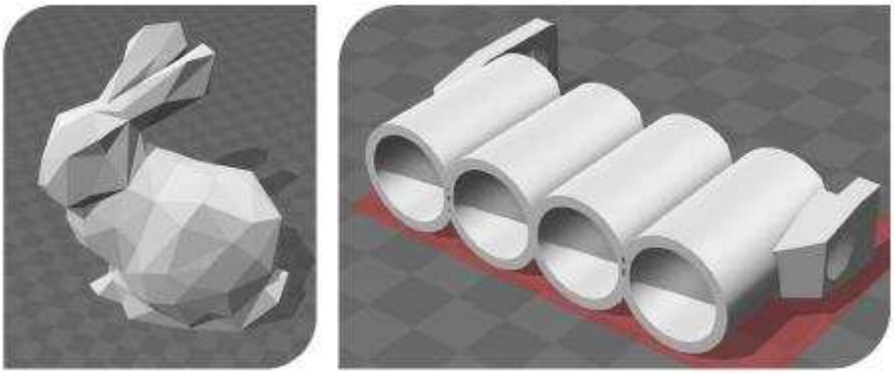


Figure 6: CAD model view in 3D Builder. Rabbit and V8 cylinder body

During the testing phase, the bunny was 3D printed and both the orientation and position of the bunny was moved in real time in order to test the approach. Figure 7 shows an example of the output of the implemented solution for various scenes.

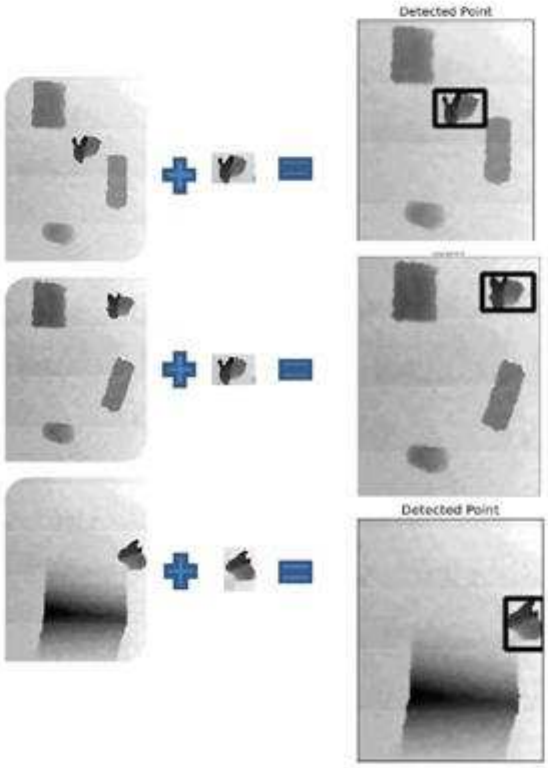


Figure 7: Results of full image template matching for Stanford bunny example

Due to the small size of the parts, the output from the Kinect® was not always reliable especially when the part presented sharp edges. The sharp edges of the V8 cylinder led to a loss of data due to the fact that the edges deflected the IR beams away from the receiver component of the Kinect. This meant that such regions showed up as black in the generated grayscale image.

3.3 Deriving HMM inspired state-based workflows using context based knowledge

The object affordance theory states that objects cannot move or transform themselves. Humans make use of an object's characteristics and features to transform or assemble them into finished products. As a result, it is possible to use an object's transformations to track the progress of a task and suggest what the human needs to do at the next step. Using a simple workflow in the form of a finite state machine, it is possible to keep track of the past, current and future steps in a manual assembly task. In order to track the manual assembly task, the use of 3D feature changes and 2D feature changes were investigated as well as prior knowledge of the manual assembly task.

This was converted into a workflow. The global work flow for a theoretical assembly task on a production line is given in Figure 8. As a first stage a set of key features linked to the workstation are identified (Figure 9). The workspace buffers in Figure 9 were used to give information about the current state of the workstation. For example, if two parts are available in buffer 1, and one part exits from buffer 1, it can then be assumed that this part will be processed in the assembly area.

In the same way if buffer 2 receives a part from the assembly area, the assembly process is considered as complete. With this simple reasoning, data is only extracted from the view of the physical area of the workstation. In Figure 10, the position of the operator's hands in the scene can also be extracted. Hence, this position can give information concerning the area where the hands are completing a task. Using the Kinect®, the height of the hands, can also provide information relevant to assembly progress.

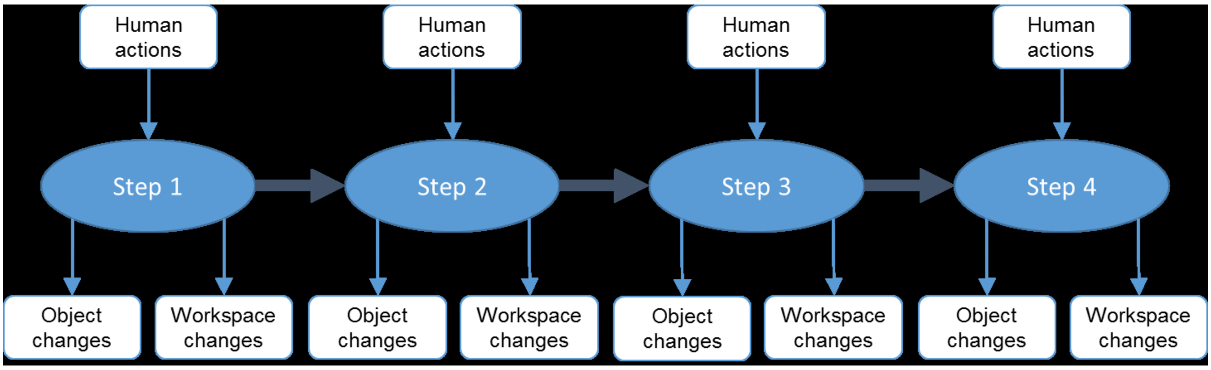


Figure 8: Workflow for a theoretical assembly process.

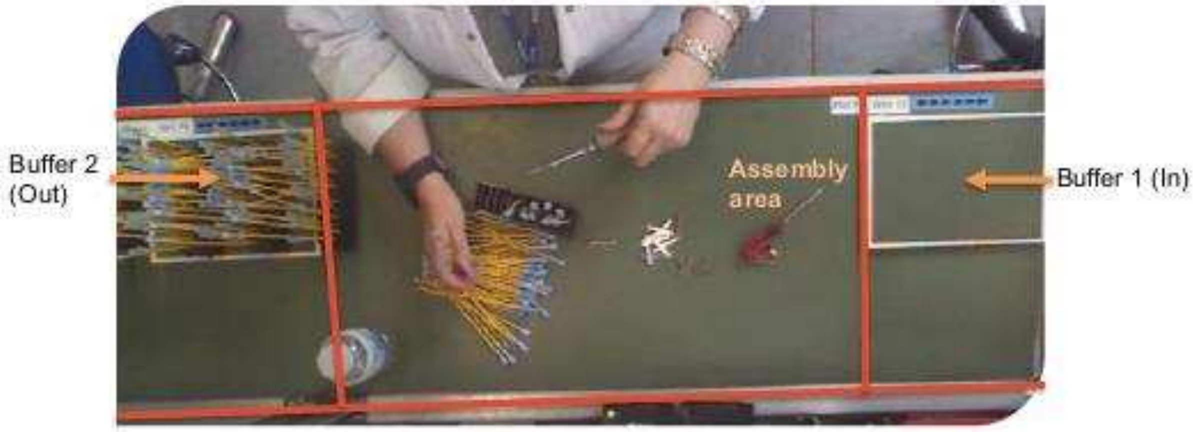


Figure 9: Buffers on a workstation in a manual production line

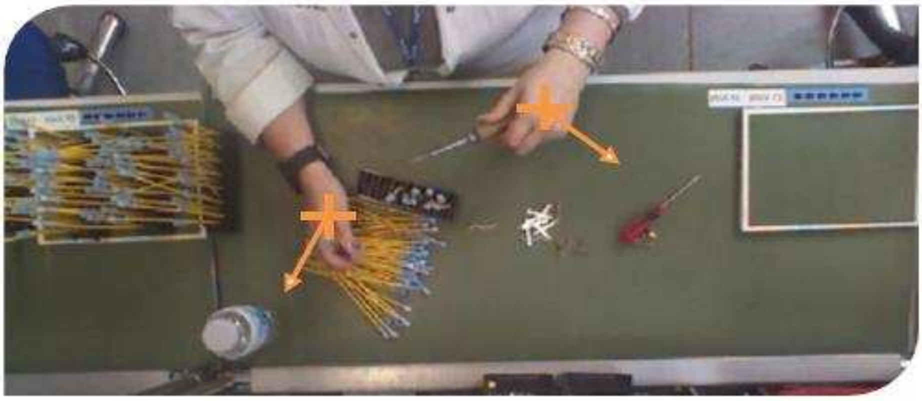
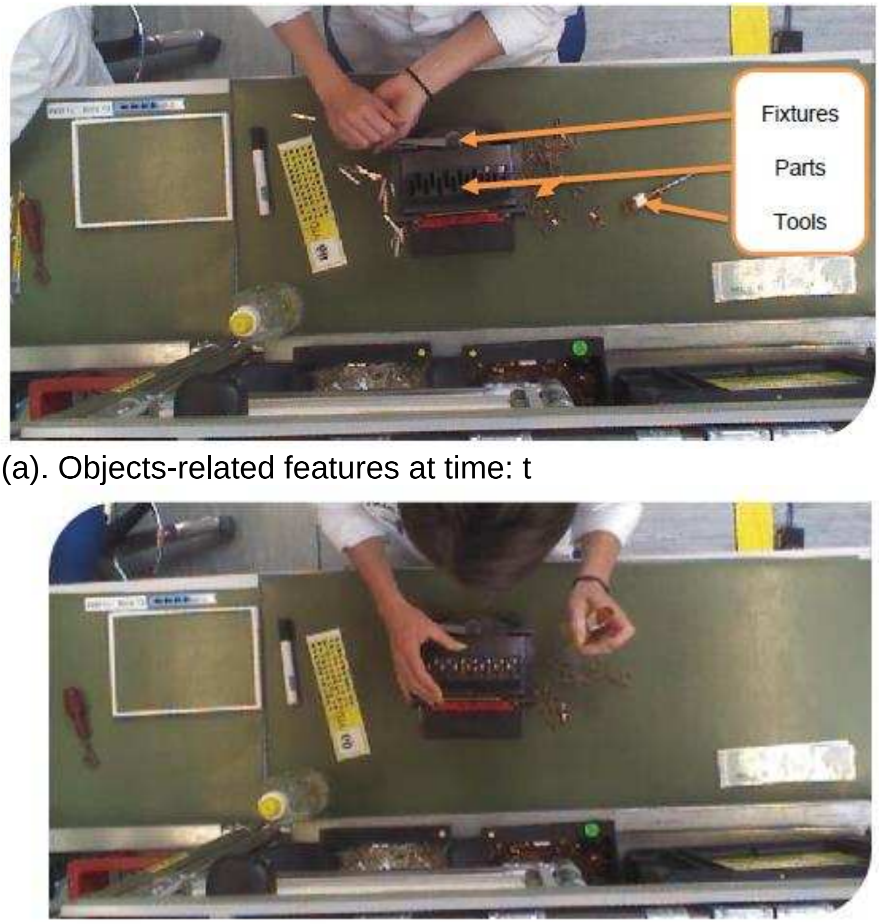


Figure 10: Hand Position and Assembly State for Feature Extraction in a manual production line.

As a result of the aforementioned process, information about the task being performed can be inferred and a descriptive statement established. In an Industry 4.0 context, the tracking of the hand could enable the potential improvement in the efficiency of the line as it could provide an insight into alternative ergonomics for the workstation. In addition body movements of the operator could be tracked in order to identify work arduousness or drudgery linked to the movements and effort made by the operator.



(a). Objects-related features at time: t

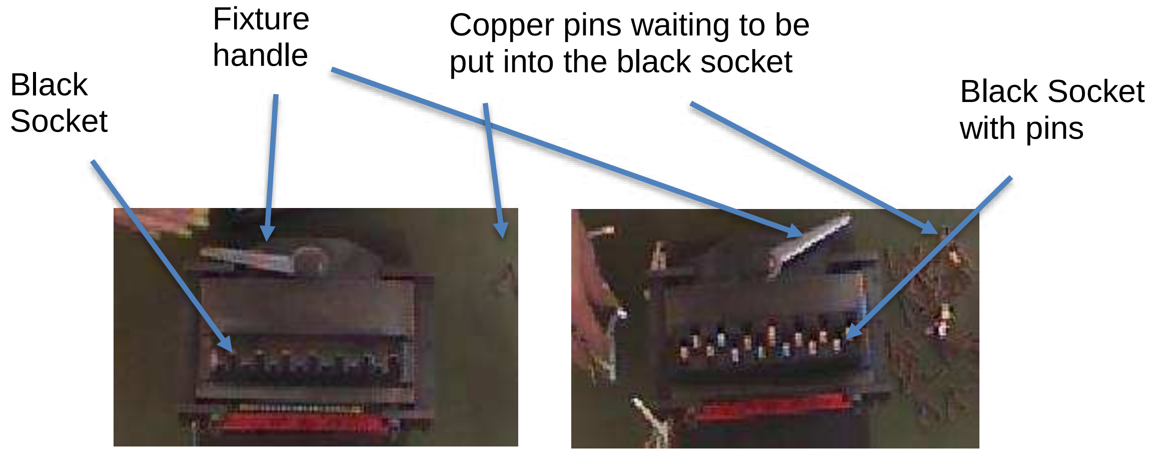
(b). Objects-related features at time t +

Figure 11: Trackable object related features on the production line

Feature changes on objects can also be extracted from the parts being assembled. Figure 11a and Figure 11b show changes in the black socket block. The changes indicate a modification of the assembly state. In Figure 11, the addition of copper-coloured pins into the black socket block show a change in the assembly state of the socket.

Tools are also key features in the assembly process; the displacement of one tool can imply the beginning of a task. The position of a tool can be compared to both the part and hand so that if the difference between two features is lower than a threshold, it can be assumed that the assembly task has started; for example, in Figure 11b, the hand has picked up the tool which induces the start of another state of the assembly task.

The state of a fixture on the workstation can also provide information. For example, Figure 12 shows two different assembly states. The modification of the handle position on the fixtures could indicate that the assembly has reached another state.



(a) (b)

Figure 12: Modification of a fixture shown between two different assembly states. In (a), a fixture handle is depressed to the left to clamp the black socket in place while in (b), the fixture handle is turned to the right in order to release the black socket.

3.4 Theoretical insights into the use of HMM inspired state-based workflows

In the theoretical assembly line in Figure 8, each of the steps could be seen as a state in a Hidden Markov Model. There are a finite number N, of such states q_i in the model λ that describes the sequence of states $Q = \{q_1, q_2, .., q_N\}$ required to build a final product P_i.

Each state q_i will emit a set of observations, $V_i = \{v_{i1}, v_{i2}, v_{i3}, .., v_{iM}\}$ where M is the number of observation symbols in state q_i. Using a known model λ of the assembly process (because most assembly lines will be properly documented and well-rehearsed to boost production), the state transition probability distribution A, will be determined by the number of actions in each state q_i required to transition the object to the next state q_j. If there are k number of actions in state q_i for example and l is the number of completed steps at time t, then the transition probability at time t is given by $\frac{l}{k}$. This means that there is a heavy reliance on a human to complete the required actions in the right sequence in order to transition to another state. As such, we rely on the following assumptions.

Assumptions:

1. Assuming the human operator is rational; obeys a set of instruction in a manual and operates according to the principle of obtaining maximum reward, the transition probabilities from one state to another is

dependent on time and the number of successful sequential actions, SA, performed by the human on the object. In such a scenario, it is beneficial for the human to operate on the principle of obtaining maximum reward according to Equations 1 and 2.

$$\Gamma(q) := argmax_a \sum_{q'} \{P_a(q,q')(R_a(q,q') + V(q'))\} \tag{1}$$

$$V(q) := \sum_{q'} \{P_{\Pi(q}(q,q')(R_{\Pi(q)}(q,q') + V(q'))\} \tag{2}$$

Where P is the state transition function, R is the reward function, Q are the states, Π is the policy which contains the actions a and V is the value of the reward. As mentioned, it is beneficial for the human to maximize the number of successful sequential actions, SA, because if she or he is deemed inefficient, he or she could be replaced.

As a result, the transition probabilities between states is mostly dependent on time or the number of successful actions completed. As a result, provided there are successful actions being completed, the longer the part stays in a particular state, the higher the probability it will change state to the next state in the sequence.

2. Each observation v_i is unique so that when in a known state q_i, it is possible to use a optimality criterion over the set of observations observable in just that state i.e. $V_i = \{v_{i1}, v_{i2}, v_{i3}, .., v_{iN}\}$ to detect what stage l in the state q_i the product is in.

3. That each product is assembled from the start every time and that an automated observer is able to initialise itself to start tracking from Q_o.

4. The changes in the object's state is caused by human actions SA as the object is inanimate. As a result, the object's change from q_i to q_j is caused by a human action. Consequently, $\Delta q \approx SA$ and as such, Δq is an observation v_i. By tracking $\{v_i, v_j, ., v_N\}$, it is possible to understand the object's progress on the assembly line. The more completed actions l are made towards a Δq, the greater the probability of emitting V_i as $t \longrightarrow \infty$

3.4.1 Implications for algorithm development

Using assumption 3, the automated observer starts tracking an object from the first time it enters into the assembly area. At this stage, it is at the initial state Q_o.

Using assumption 4, Δq triggers a subroutine indicating that the human has just carried out an action. Since each observation $\{v_{1j}, v_{2j}, v_{3j}, .., v_{Nj}\}$ is unique (assumption 2), using an optimality criterion, the automated observer searches the model λ observation set V for which observation is the closest match to the current observation. However, since we know what state q_j we are in, we can reduce the search space to just V_j thereby making it more computationally tractable. Using assumption 1, that the human is rational and will try to maximize reward, the object's changes Δq will follow a logical sequence $\{q_1, q_2, q_3, ..q_N\}$ of states to completion. The transition from one state to another is also guaranteed.

4.0 Results

As discussed above, one of the priorities in this research was to ensure that the approach developed could be utilized across many domains. In this section, it is shown how the approach of this paper was tested, utilising simple lab based approaches in the first instance culminating in a number of increasingly difficult use cases.

4.1 Assembly of Coloured Blocks use case:

The first sequence to be considered was the assembly of some coloured blocks. These parts were chosen because of the different colours available and their simplicity of assembly. The Figure 13 shows the assembly sequence developed.

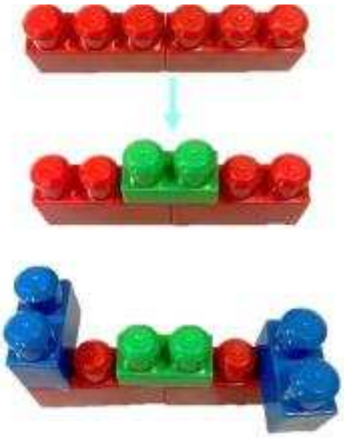


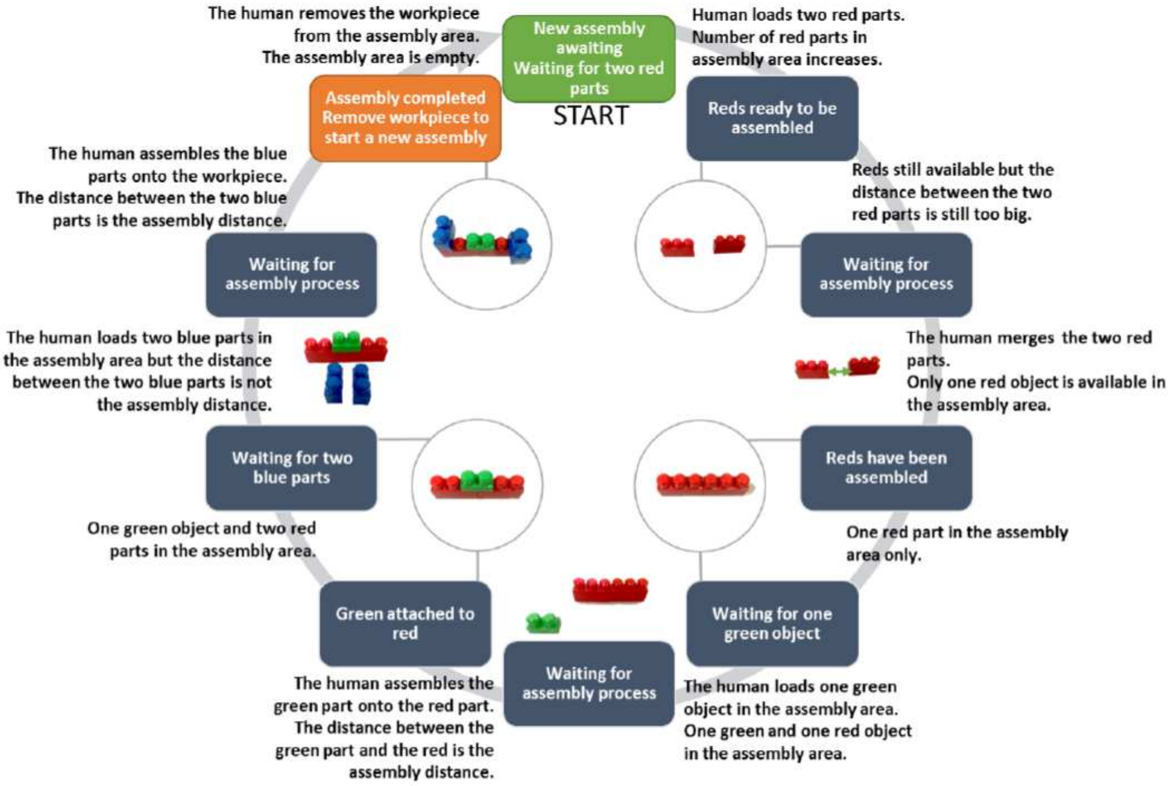
Figure 13: Assembly sequence of blocks

The workflow of the sequence was developed and converted into a state machine of the solution. In Figure 14, the assembly sequence and actions undertaken between each step are highlighted.

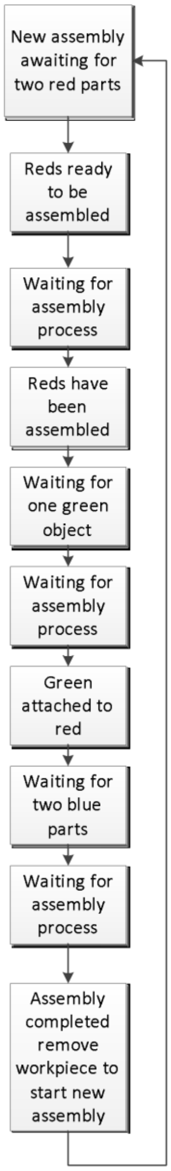
The program was tested under the following conditions: only the coloured parts are available in the scene, the operator then follows the instructions provided by the program.

The results are shown, in Figure 15, which presents the different states of the assembly process. The program is able to track the different stages of the assembly task and give instructions for future steps. This was based on the number of parts in the assembly area and the distance between the centre of mass to give a statement about the assembly state (see section 3.1.1 and 3.3).

Once a workpiece is fully assembled, the program proposes to remove the workpiece to start a new assembly process. The operator has to adhere to the instructions given by the program to follow the assembly process as it cannot skip steps. This ensures that: (1) the operator does not miss any steps; (2) the operator receives assistance from the system and (3) an inline inspection is carried out constantly thereby eliminating a need for another workstation dedicated to visual inspection tasks.

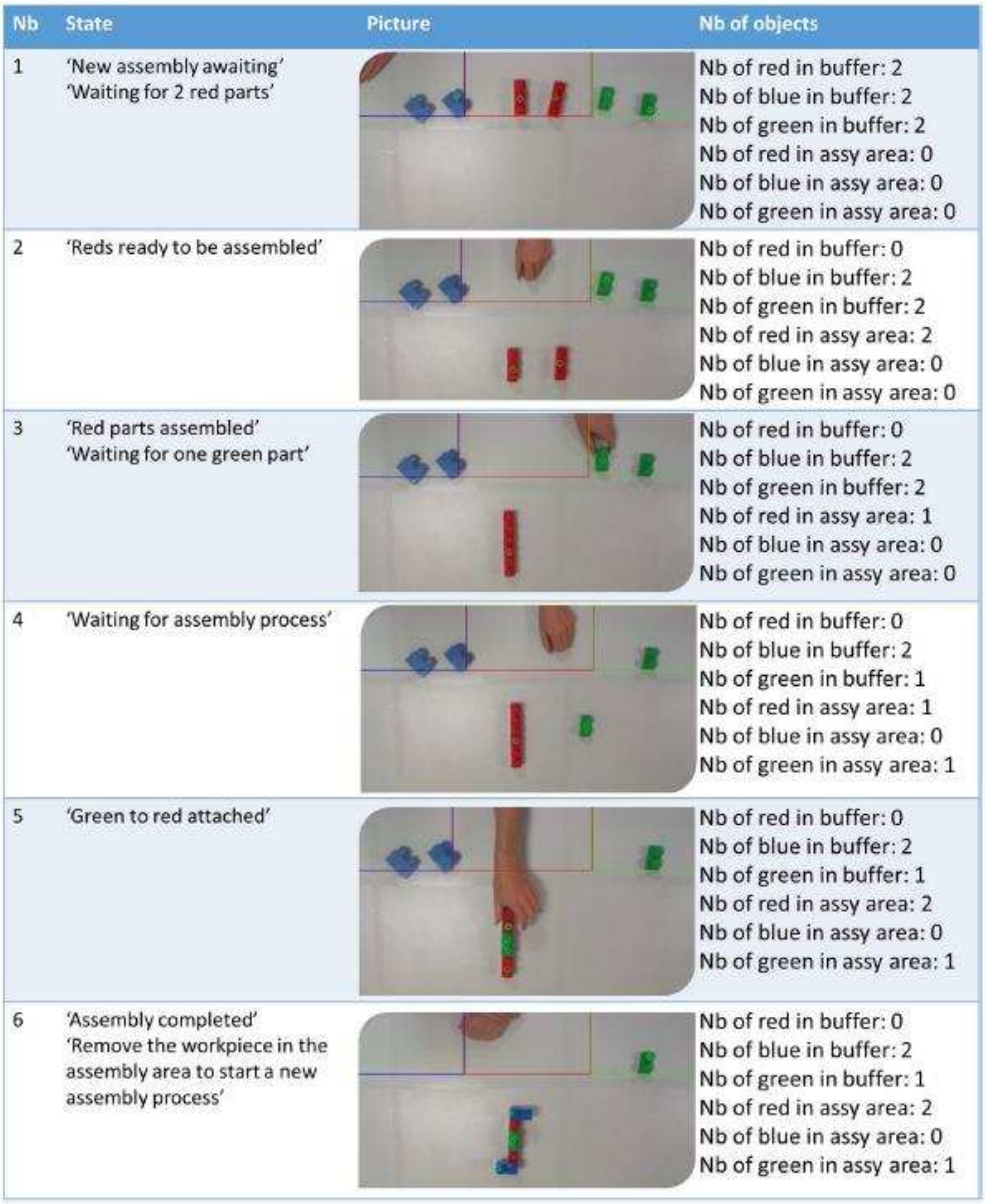

New assembly awaiting Waiting for two red parts
START
Human loads two red parts. Number of red parts in assembly area increases.
Reds ready to be assembled
Reds still available but the distance between the two red parts is still too big.
Waiting for assembly process
The human merges the two red parts. Only one red object is available in the assembly area.
Reds have been assembled
One red part in the assembly area only.
Waiting for one green object
The human loads one green object in the assembly area. One green and one red object in the assembly area.
Waiting for assembly process
The human assembles the green part onto the red part. The distance between the green part and the red is the assembly distance.
Green attached to red
One green object and two red parts in the assembly area.
Waiting for two blue parts
The human loads two blue parts in the assembly area but the distance between the two blue parts is not the assembly distance.
Waiting for assembly process
The human assembles the blue parts onto the workpiece. The distance between the two blue parts is the assembly distance.
Assembly completed Remove workpiece to start a new assembly
The human removes the workpiece from the assembly area. The assembly area is empty.

(a) Diagrammatic assembly sequence of the coloured blocks in Figure 14.


New assembly awaiting for two red parts
Reds ready to be assembled
Waiting for assembly process
Reds have been assembled
Waiting for one green object
Waiting for assembly process
Green attached to red
Waiting for two blue parts
Waiting for assembly process
Assembly completed remove workpiece to start new assembly

(b) Flowchart of the assembly sequence

Figure 14: Workflow for the assembly of coloured blocks



Nb	State	Picture	Nb of objects
1	'New assembly awaiting' 'Waiting for 2 red parts'		Nb of red in buffer: 2 Nb of blue in buffer: 2 Nb of green in buffer: 2 Nb of red in assy area: 0 Nb of blue in assy area: 0 Nb of green in assy area: 0
2	'Reds ready to be assembled'		Nb of red in buffer: 0 Nb of blue in buffer: 2 Nb of green in buffer: 2 Nb of red in assy area: 2 Nb of blue in assy area: 0 Nb of green in assy area: 0
3	'Red parts assembled' 'Waiting for one green part'		Nb of red in buffer: 0 Nb of blue in buffer: 2 Nb of green in buffer: 2 Nb of red in assy area: 1 Nb of blue in assy area: 0 Nb of green in assy area: 0
4	'Waiting for assembly process'		Nb of red in buffer: 0 Nb of blue in buffer: 2 Nb of green in buffer: 1 Nb of red in assy area: 1 Nb of blue in assy area: 0 Nb of green in assy area: 1
5	'Green to red attached'		Nb of red in buffer: 0 Nb of blue in buffer: 2 Nb of green in buffer: 1 Nb of red in assy area: 2 Nb of blue in assy area: 0 Nb of green in assy area: 1
6	'Assembly completed' 'Remove the workpiece in the assembly area to start a new assembly process'		Nb of red in buffer: 0 Nb of blue in buffer: 2 Nb of green in buffer: 1 Nb of red in assy area: 2 Nb of blue in assy area: 0 Nb of green in assy area: 1

Figure 15: Tracking object assembly using coloured features

4.2 Assembly of metrology parts use case:

The second assembly process considered involved the use of metallic metrology parts. These parts were chosen as they were representative of real industrial parts. The purpose of this use case was to ascertain if the program can run within an industrial environment. The Figure 16 shows the metallic workpieces and their assembly sequence. The parts are assembled in a tower formation.

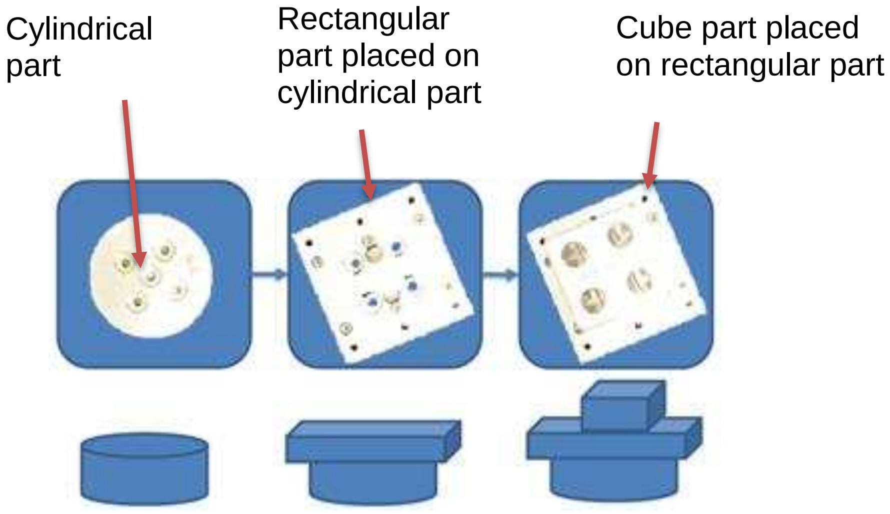

Cylindrical part
Rectangular part placed on cylindrical part
Cube part placed on rectangular part

Figure 16: Assembly sequence of metrology parts. A pictorial side view is also shown due to the lack of contrast between the metrology parts when viewed from above.

The metrology parts were chosen because of their simple shapes and metallic aspect. Therefore, the method involved here must recognize the parts using just shape features.

Once the assembly sequence was identified, the workflow was created. Two different tests were carried out, the first one involving buffers, and the second without buffers. The difference between the two tests is relevant because the workflow used to track the assembly progress will be different.

If buffers are used, the allocation of parts in the scene can provide a lot of information about the assembly sequence. Otherwise, if no buffer is available, the allocation of parts in the scene is not sufficient to identify the assembly state. In this case the recognition methods have to be very robust as reliance must be placed on the tracking of shape or colour features anywhere in the assembly area instead of just the parts in the buffers.

Although the solution worked, it was identified that the tracking of the last stage (assembly of small cube on top of the big square) was problematic. The colours of the two parts were very similar. As a result, another feature was chosen to identify the cube. This relied on the volume data from the depth information. As a result, this strategy involved the use of both 3D and 2D techniques.

4.3 A bottle packing use case

In this use case, the packing of bottles into a box was investigated. The centre of mass of bottles and a box as well as the number of the objects along with the buffer states on the workbench were all used. The distance measure between objects and the buffer as discussed in 3.3.1 was also used.

The workflow sequence in this use case involved an off screen worker loading the left buffer with the right number of bottles (two in this case). A seated worker loads the working area in the centre with a box. Then the seated worker takes the two bottles from the left buffer and loads them into the box. After loading the bottles, the worker moves it to the right buffer to be picked by another worker off screen (Figure 17).

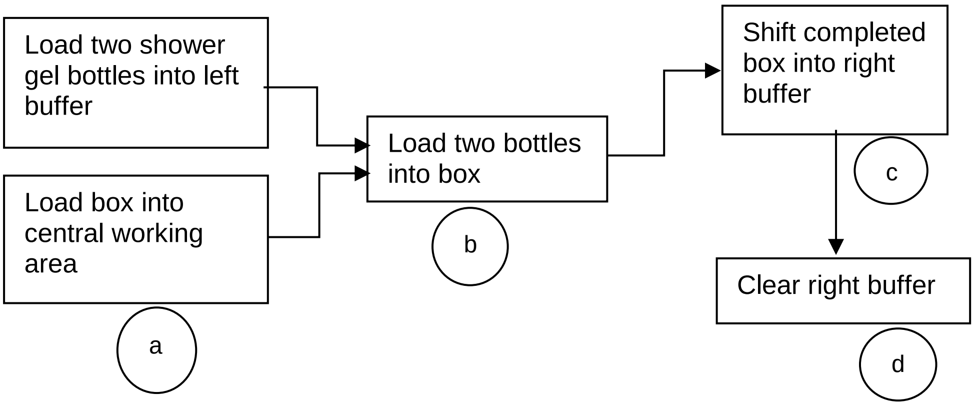
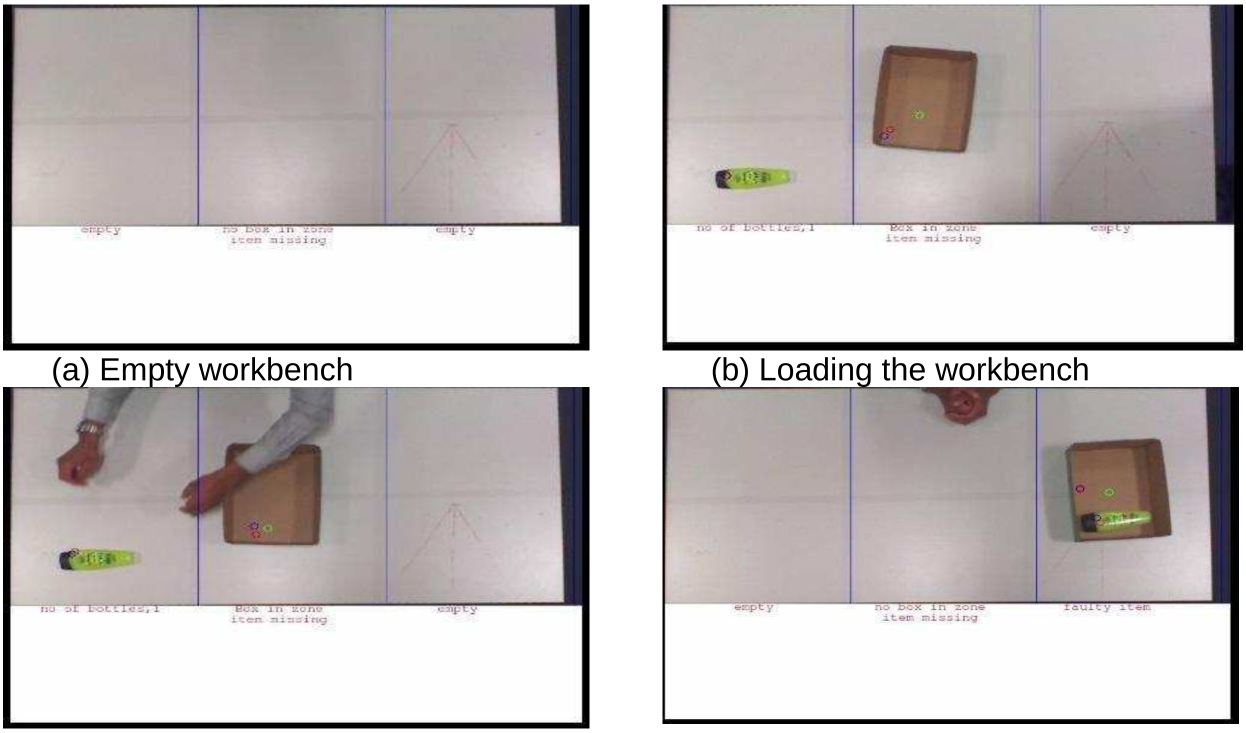

Load two shower gel bottles into left buffer
Load box into central working area
a
Load two bottles into box
b
Shift completed box into right buffer
c
Clear right buffer
d

Figure 17: Workflow sequence of loading shower gel bottles into a packing box

Note that the operations of loading the box and bottles can be completed in parallel. Using this workflow, any discrepancies were flagged up as a potential assembly error or problem. For example in Figure 18 (b-d), when the observed actions was different from the workflow state in Figure 17 (b), our system was able to display either: “item missing” or “faulty item”.



(a) Empty workbench

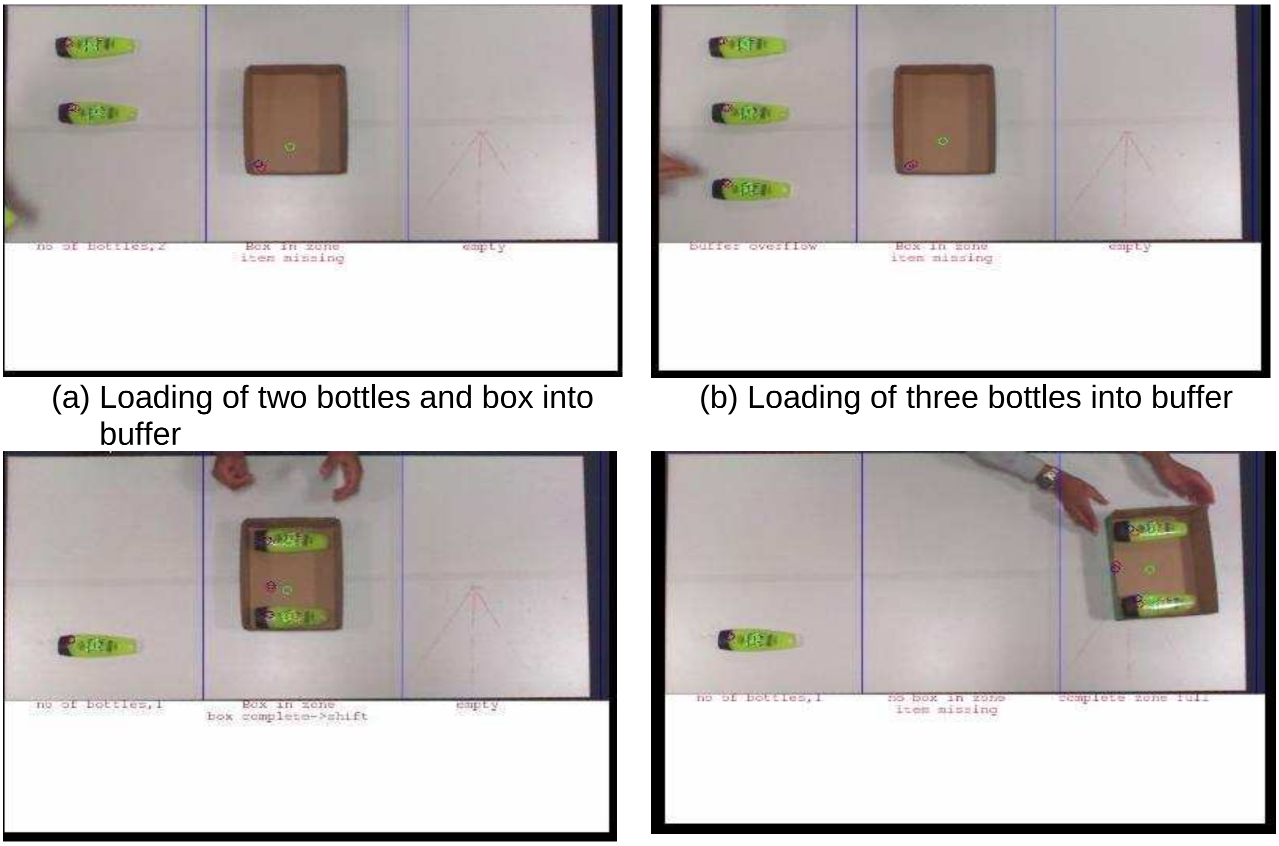
(b) Loading the workbench

(c) Worker filling the box

(d) Loading operation finished

Figure 18: Incomplete loading of shower gel bottles into a box.

Our system was able to show the number of bottles currently on the workbench. In Figure 19 (b), when the number of bottles exceeded the expected number according to the workflow sequence diagram in Figure 17, a “buffer overflow” message was displayed. Also, Figure 19 (c) shows that when the right number of bottles is loaded into the box, the system is able to display a “box complete->shift” message. Since the computer vision processing is completed at the workbench, the delivery of images in real time to a central computer is not required. Notice also in Figure 19 (d) that system is able to identify that the box was packed correctly for the next stage of the processing which could be wrapping.



(a) Loading of two bottles and box into buffer

(b) Loading of three bottles into buffer

(c) Filling of the box with two bottles

(d) Loading operation finished

Figure 19: Complete loading of shower gel bottles into a box.

4.4 Production line assembly of electronic components use case

Figure 20 shows an industrial workstation that was chosen because of its similarity to a real industrial environment. In this scenario, all the main features of the workstation are visible. The challenge here was in defining the workflow sequence for the assembly of the electronic component and if the sequence could be tracked.

For this use case the colour features of objects were considered. Nevertheless, due to the high number of parts, their small dimensions and the multiple occlusions that occur as a result of the human hand, the tracking of workpiece changes is difficult to achieve.

Use of the depth data stream from the Kinect was made within this case. Figure 18 shows the type of depth data from the production line. The black colour area on the depth image is considered as an area of errors, returning a 0 value. The error occurs because of the limitations of the Kinect. This could be due to the infra-red receiver not receiving a return ray due to specular reflection caused by the objects.

Nevertheless, the recognition of the parts within the workstation is possible but due to the many variations in this assembly use case, the implementation of a solution that tracks the workpiece changes in this type of environment remains a future research target.

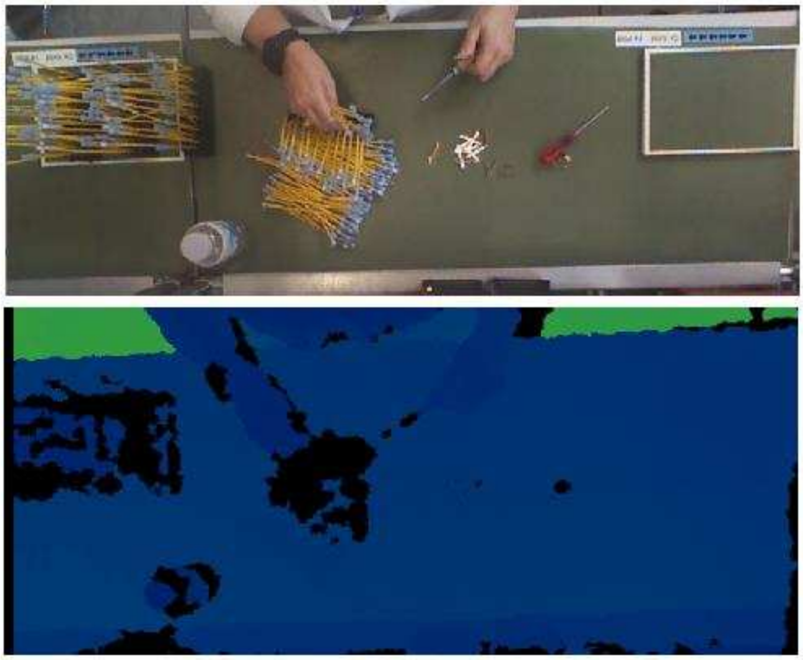


Figure 20: A RGB-D capture of a production line workstation (Colour based top, depth based bottom).

5.0 Discussion

The main objectives of this research were to investigate techniques that enable the detection of a workpiece based on features and use of sequence workflows to extract manual assembly progress through the tracking of workpiece feature changes. This involved the testing and validation of the developed techniques using various use case scenarios.

In order to implement a robust and reliable approach, (i) it is advisable to understand and analyse the assembly sequence as a first step; (ii) this can then be converted into a computational workflow to enable the detection of the progress of an assembly task.

This work relied on the use of 2D and 3D techniques for object recognition while the use of a workflow sequence enabled the encoding of the steps required to assemble a workpiece.

5.1 Using 2D techniques

The 2D related work, while utilising existing research, attempted to link these findings to an industrial environment. Although the recognition of parts can be achieved easily in a constrained environment, the 2D techniques appeared to be difficult to implement in a common industrial environment.

For example, colour-based techniques presented a drawback due to the propensity of colours to change as a result of rising and falling light levels. One core challenge is to constrain the scene to maintain the same colour appearance throughout the day. Lighting the scene with LED in a closed workstation could enhance the colour recognition. In the case of this solution the colour features would always be the same. Moreover, when considering the shape of the parts, the main challenge is that the gradient of colour between the parts and the workstation surface has to be significant to enable the recognition of the shapes. This requires a constrained environment, where the colour of both parts and the workstation surface are significantly different.

Deep learning techniques could have been applied in this work to solve some of the problems with illumination, but this requires the collection of a large dataset for training. Furthermore, as mentioned previously, if a domain is well understood, the use of machine learning could be bypassed and rules written to effectively capture the domain (Amazon, 2017).

5.2 3D techniques

3D related work has brought new perspectives and ideas for future development. The generation of grayscale images from a CAD model to track both objects and progress of an assembly task shows some promise despite the aforementioned limitations of the Kinect®.

Multiple strategies including sum and template matching were investigated for 3D object and task recognition. This presents an opportunity to develop more robust strategies involving mathematical functions that are capable of describing the image itself. Such future developments could enhance the efficiency and reliability of the recognition.

As the detection of manual assembly tasks has to be performed in real-time, another challenge is to implement the 3D techniques in real-time. Different 3D techniques currently receive an image as the input, not a real time video sequence. Therefore, although the template matching is a robust object recognition method, it appears that the time required to detect a single object is significant. As a result, it will be necessary to adapt this solution to cope with real-time requirements. One solution could be to select a single zone of interest from the scene in order to speed up the recognition method.

5.3 Combination of 2D and 3D techniques

Though 2D object recognition can be completed in real time, a combination of 2D and 3D methods could be interesting to implement in order to make a more robust and reliable technique. This combination could indeed improve the recognition and tracking of the parts as several methods based on different object features would be used at the same time. For example, combining both depth and colour of the scene can avoid the problems of misrecognition of objects using colour. However, one main challenge is the real-time perspective. Combining two recognition techniques would indeed increase the time for recognition. Therefore, a compromise between the techniques used and the time required for recognition has to be made.

5.4 Overall value of this work

The generation of a grayscale image from a CAD model appears to be very valuable. By using the generation of grayscale images from a CAD model, it is only necessary to add the CAD model into a library of workpieces so that the recognition system can then identify the workpiece on the workstation. Using the CAD model, the system can generate multiple object orientations to create different grayscale images owing to given orientations. Therefore, it would save time and increase the flexibility of the proposed method as a new workpiece can be inserted very quickly into the library and subsequently detected on the workstation.

Moreover, the use of low cost equipment is very attractive for industry. The setup of RGB-D cameras on the production line implies a cost effective solution that is highly portable with low set up times. Furthermore, most industrial applications of computer vision or machine vision utilize such technology for inspecting parts. This study pushes this barrier by presenting a way to track an actual manual assembly sequence on a production line. The data obtained from this approach could then be mined for crucial insights such as ergonomic improvement metrics, defects rates and their causes. This research also presents a feasible entry point for SMEs interested in utilizing industry 4.0 concepts in their production lines.

The use of an RGB-D camera for the recognition of objects has been demonstrated in a number of previous research works. However, according to present knowledge, no clear links between such research and industrial applications that track a manual assembly sequence have been identified. The findings within this paper demonstrate that RGB-D cameras can be suitable for object recognition and manual assembly tracking within an industrial environment if a number of constraints are applied. The approaches investigated in this paper could offer new perspectives to industry and increase the uptake of tracking of manual assembly tasks in an industrial environment. Furthermore, through the use of a HMM inspired state-based workflow as discussed in section 3.3 and 3.4, the developed algorithm is computationally tractable.

There are four main limitations that were discovered in the course of this work. These limitations were related to the Kinect sensor used as well as the techniques utilized.

Firstly, it was discovered that the Kinect® gave poor depth quality when considering parts with shiny surfaces, sharp edges or objects whose sizes are very small.

Secondly, during the conversion of the CAD model to grayscale images, an increase in the number of vertices led to better resolution and accuracy. However, this leads to a heavy computational load during the conversion process. Nevertheless, this could be completed offline and then used in real time to detect objects.

Thirdly, the testing of the algorithm using various use cases has demonstrated that the use of a single technique to recognize the parts can result in a low level of reliability. This was especially true when the components to be manually assembled were stacked on top of each other. A further investigation that combines 2D and 3D feature detection could make object recognition more robust.

Fourthly, the approach relies heavily on the contextual knowledge of the manual assembly in order to generate a computational workflow. The authors believe that this is a trade-off for not using a deep learning approach such as Long-Short Term Memory network. The approach ensures that training data is keep to a very minimum or non-existent and can be deployed to many new use cases easily and rapidly. Nevertheless, the manual setting of the thresholds for the object recognition component is still required.

6.0 Conclusions

This project has exploited RGB-D cameras to investigate the feasibility of using computer vision techniques and a HMM inspired state-based algorithm to track the progress of a manual assembly task on a production line in real time.

Using a set of increasing complex of use cases, it has been shown that it is possible to track a manual assembly process being completed by a human. This was achieved by using: (i) contextual based object recognition based on known object features and (ii) a knowledge of the sequence of the manual tasks to be carried out on a workpiece.

For contextual based object recognition, 2D and 3D image processing techniques were investigated in order to recognise objects. 2D techniques highlighted in literature used several features of an object for object recognition. Features that were colour, shape or contour based were revealed to be acceptable features for industrial object recognition according to literature.

The techniques investigated during this project confirmed the expectations from literature. The selected 2D techniques achieved good results and the recognition of colours and shape of objects using 2D features was found to be robust. In contrast the identified 3D techniques used in isolation were unsuitable due to the resolution limits of the Kinect®. As Kinect® was designed for skeleton recognition, the tracking of small parts in real-time still needs to be investigated.

As a result, 2D techniques, were used to achieve the tracking of a manual assembly sequence in 4 different use cases. In the first use case, coloured blocks were as part of an experiment to track a manual assembly task. Even though encouraging results were achieved, the parts used had simple colours and shapes, and the surrounding environment was constrained. The second use case was similar to the first except that a 3D feature based on volume geometric data was used to mitigate the limitations of using a pure 2D based colour detection strategy.

In the third use case, the complexity of the manual assembly was increased. This involved packing a box of shower gels. The results of this use case showed that if: (i) the constraints on the surrounding environment were maintained as in the first use case; (ii) the workbench was clearly laid out and (iii) the human's actions in each unique area (i.e. buffer and central area) were predictable, then it is possible to track an assembly sequence in real time.

This means that if the human actions are less predictable, then they could pose a challenge to our approach due to the over reliance on a complete knowledge of the manual assembly process.

In the fourth and final use case, a manual assembly process on an actual production line was tracked. The results of this use case revealed two problems: (i) the arrangement of items in the central buffer used by the human was less predictable than the second use case. Furthermore, some of the parts in the central buffer were reflective and too small to be detected by the Kinect ®. As a result, it was more challenging to track objects and as such the manual assembly as a whole.

Overall, the tests and validation in different scenarios revealed that although the Kinect® resolution was too low, the recognition of objects as well as tracking a manual assembly progress in real time could be achieved if certain constraints were applied.

Acknowledgements
This work has been funded by EPSRC High Value Manufacturing Catapult Fellowship (EP/L017121) on 'Human Interventions in Autonomous Manufacturing: Digitization and Learning'. The Industrial Sponsor is the Manufacturing Technology Centre (MTC), Coventry, UK.

References

Aehnelt, M., Gutzeit, E. and Urban, B., (2014). Using activity recognition for the tracking of assembly processes: Challenges and requirements. WOAR, 2014, pp.12-21.

Amazon, (2017) Amazon machine learning, Developer's Guide.

Ballester, P. and de Araújo, R.M., (2016), February. On the Performance of GoogLeNet and AlexNet Applied to Sketches. In AAAI (pp. 1124-1128).

Barron, J. T. and Malik, J. (2016) “Intrinsic scene properties from a single RGB-D image”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.38, No.4, 2016, pp. 690–703. doi: 10.1109/TPAMI.2015.2439286.

Barron, J. T. and Malik, J. (2015) “Shape, illumination, and reflectance from shading”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.37, No.8, pp. 1670–1687. doi: 10.1109/TPAMI.2014.2377712.

Bi, Z.M. and Kang, B., (2014). Sensing and responding to the changes of geometric surfaces in flexible manufacturing and assembly. Enterprise Information Systems, 8(2), pp.225-245.

Borji, A., Cheng, M. M., Jiang, H. and Li, J. (2015) “Salient object detection: A benchmark”. IEEE Transactions on Image Processing, Vol. 24, No. 12, pp.5706-5722.

Bu, S., Zhao, Y., Wan, G., Li, K., Cheng, G. and Liu, Z. (2016) “Semi-direct tracking and mapping with RGB-D camera for MAV”, Multimedia Tools and Applications, doi: 10.1007/s11042-016-3524-x.

Cai, Z., Han, J., Liu, L., and Shao, L. (2016) “RGB-D datasets using microsoft kinect or similar sensors: a survey”, Multimedia Tools and Applications, Vol. 76, No. 3, pp. 4313-4355. doi: 10.1007/s11042-016-3374-6.

Fu, H., Xu, D. and Lin, S., (2017) “Object-based multiple foreground segmentation in RGBD video”. IEEE Transactions on Image Processing, 26(3), pp.1418-1427.

Funk, M., Shirazi, A.S., Mayer, S., Lischke, L. and Schmidt, A., (2015), September. Pick from here!: an interactive mobile cart using in-situ projection for order picking. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 601-609). ACM.

Gupta, S., Girshick, R., Arbelaez, P. and Malik, J. (2014) “Learning rich features from RGB-D images for object detection and segmentation”, Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, 8695 LNCS, pp. 345–360. doi: 10.1007/978-3-319-10584-0_23.

Gupta, S., Arbelaez, P., Girshick, R., and Malik, J. (2015) “Indoor Scene Understanding with RGB-D Images”, International Journal of Computer Vision, Vol. 112, No.2, 133-149

Hartmann, B. (2011) “Human Worker Activity Recognition in Industrial Environments”, 2011, KIT Scientific Publishing, Karlsruhe.

Hryniewicz, P., Banas, W., Sekala, A., Gwiazda, A., Foit, K., and Kost, G. (2015) “Object positioning in storages of robotized workcells using LabVIEW Vision”, In IOP Conference Series: Materials Science and Engineering, Vol. 95, pp. 012098. doi: 10.1088/1757-899X/95/1/012098.

Ju, R., Liu, Y., Ren, T., Ge, L. and Wu, G. (2015) “Depth-aware salient object detection using anisotropic centre-surround difference”, Signal Processing: Image Communication, Vol. 38, pp. 115–126. doi: 10.1016/j.image.2015.07.002.

Li, J. and Huang, Y. (2015) “Automatic Inspection of Tire Geometry with Machine Vision”, in Mechatronics and Automation (ICMA), 2015 IEEE International Conference on, pp. 1950–1954.

Lowe, D. G. (1999) “Object recognition from local scale-invariant features”, Proceedings of the Seventh IEEE International Conference on Computer Vision, Vol.2, No.8, pp. 1150–1157. doi: 10.1109/ICCV.1999.790410.

Matas, J. and Obdržálek, Š. (2004) “Object recognition methods based on transformation covariant features”. In Signal Processing Conference, 2004 12th European, pp. 1721-1728. IEEE.

Maturana, D. and Scherer. S., 2015, VoxNet: A 3D Convolutional Neural Networkfor Real-Time Object Recognition. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pages 922–928. IEEE.

Prabhu, V. A., Song, B., Thrower, J., Tiwari, A. and Webb, P. (2015) “Digitisation of a moving assembly operation using multiple depth imaging sensors”, International Journal of Advanced Manufacturing Technology. Vol. 85, No.1-4, pp.163-184. doi: 10.1007/s00170-015-7883-7.

Radkowski, R., (2016). Object tracking with a range camera for augmented reality assembly assistance. Journal of Computing and Information Science in Engineering, 16(1), p.011004.

Radkowski, R., Herrema, J. and Oliver, J., (2015). Augmented reality-based manual assembly support with visual features for different degrees of difficulty. International Journal of Human-Computer Interaction, 31(5), pp.337-349.

Radkowski, R. and Oliver, J., (2013), July. Natural feature tracking augmented reality for on-site assembly assistance systems. In International Conference on Virtual, Augmented and Mixed Reality (pp. 281-290). Springer, Berlin, Heidelberg.

Rothganger, F., Lazebnik, S., Schmid, C. and Ponce, J. (2006) “3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints”. International Journal of Computer Vision, Vol. 66, No.3, pp.231-259.

Schmitt, R., Fartjes, T., Abbas, B., Abel, P., Kimmelmann, W., Kosse, P. and Buratti, A. (2015) “Real-time machine vision system for an automated quality monitoring in mass production of multiaxial non-crimp fabrics”, in IFAC Proceedings Volumes (IFAC-PapersOnline), pp. 2393–2398. doi: 10.1016/j.ifacol.2015.06.446.

Sedaghat, N., Zolfaghari, M., Amiri, E. and Brox, T., (2017). Orientation-boosted voxel nets for 3D object recognition. In British Machine Vision Conference (BMVC), Imperial College London 4th-7th September 2017

Stork, A. (2015) “Visual Computing Challenges of Advanced Manufacturing and Industrie 4.0”, IEEE Computer Graphics and Applications, Vol. 35, No. 2, pp. 21–25. doi: 10.1109/MCG.2015.46.

Sture, Ø., Øye, E. R., Skavhaug, A., and Mathiassen, J. R. (2016) “A 3D machine vision system for quality grading of Atlantic salmon”, Computers and Electronics in Agriculture, Vol. 123, pp. 142–148. doi: 10.1016/j.compag.2016.02.020.

Sutskever, I., Vinyals, O. and Le, Q.V., (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

Tang, P., Wang, H. and Kwong, S., (2017). G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition. Neurocomputing, 225, pp.188-197.

Wang, D.Z. and Posner, I., (2015), July. Voting for Voting in Online Point Cloud Object Detection. In Robotics: Science and Systems (Vol. 1, p. 5).

Wu, H. H. and Bainbridge-Smith, A. (2011) “Advantages of using a Kinect Camera in various applications”, available at: http://www.academia.edu/2070005/Advantages_of_using_a_Kinect_Camera_in_various_applications (accessed 21 August 2017)

Xi, T., Zhao, W., Wang, H. and Lin, W. (2017) “Salient object detection with spatiotemporal background priors for video”. IEEE Transactions on Image Processing, Vol. 26, no. 7, pp. 3425-3436.