An Unsupervised Machine Learning Approach To Assessing Designer Performance During Physical Prototyping

Matthew L. Dering
Computer Science and Engineering
Pennsylvania State University
University Park, Pennsylvania 16801
Email: dering@cse.psu.edu

Conrad S. Tucker
Industrial Engineering
Pennsylvania State University
University Park, Pennsylvania 16801
Email: ctucker4@psu.edu

Soundar Kumara
Industrial Engineering
Pennsylvania State University
University Park, Pennsylvania 16801
Email: u1o@engr.psu.edu

ABSTRACT

An important part of the Engineering Design process is prototyping, where designers build and test their designs. This process is typically iterative, time consuming, and manual in nature. For a given task, there are multiple objects that can be used, each with different time units associated with accomplishing the task. Current methods for reducing time spent during the prototyping process have focused primarily on optimizing designer to designer interactions, as opposed to designer to tool interactions. Advancements in commercially-available sensing systems (e.g., the Kinect) and machine learning algorithms have opened the pathway towards real-time observation of designer's behavior in engineering workspaces during prototype construction. Towards this end, this work hypothesizes that an object O being used for task i is distinguishable from object O being used for task j, where i is the correct task and j is the incorrect task. The contributions of this work are i) the ability to recognize these objects in a free roaming engineering workshop environment and ii) the ability to distinguish between the correct and incorrect use of objects used during a prototyping task. By distinguishing the difference between correct and incorrect uses, incorrect behavior (which often results in wasted time and materials) can be detected and quickly corrected. The method presented in this work learns as designers use objects, and infers the proper way to use them during prototyping. In order to demonstrate the effectiveness of the proposed method, a case study is presented in which participants in an engineering design workshop are asked to perform correct and incorrect tasks with a tool. The participants' movements are analyzed by an unsupervised clustering algorithm to determine if there is a statistical difference between tasks being performed correctly and incorrectly. Clusters which are a plurality incorrect are found to be significantly distinct for each node considered by the method, each with $p << 0.001$.

1 Introduction

The Engineering Design process is labor intensive and can involve many individuals working long hours across several disciplines. For example, during the early stages, market researchers may perform focus groups to assess consumer requirements, and those requirements are considered in terms of viability. Recent work allows designers to mine social network streams and enable designers to rapidly capture desired product features [1, 2]. Later stages require designers to select which requirements are suitable for ideation, a task which has had many proposed solutions [3–7]. The subsequent concept generation phase has been impacted by advancements in automation tools, with many methods available that can generate concepts [8–10]. Next, design validation and assessment can be accomplished using semi-automated simulation techniques, such as Computational Fluid Dynamics [11, 12]; however, this does not eliminate the need for physical prototyping. The physical prototyping phase of design, where a designer builds the proposed design, is integral to the overall concept generation and eventual product creation process [13–16]. This process can be time consuming, iterative, and highly manual, so reducing either time per iteration and number of iterations could reduce the time to market. Since physical prototyping is so labor intensive, these reductions will ease the cost of labor in settings that are under increasing budgetary pressure [17, 18]. This increased efficiency benefits designers as well [19]. Just as there exists methods that assess and validate the feasibility of design concepts (e.g., through CFD), methods are needed that assess and validate the efficiency of the physical prototyping process that realizes those concepts.

Given a common objective, there are many different tools and many different ways designers can use these tools to accomplish this objective, making this problem extremely complex. For example, designers working in teams to design a shopping cart could use a wide range of materials (e.g. aluminum, PVC, steel) and objects (e.g. drill, hammer, saw) to create a prototype. Furthermore, the variability in the expertise and design decisions by these designers could potentially influence the quality and efficiency of the prototyping process. The prototyping phase can be decomposed into a series of sub-tasks (for example, task 1 might be to cut a pipe, task 2 to hammer a section of the pipe, etc.), each with a correct and incorrect (less efficient) method. Since prototyping can be an iterative process and is difficult to automate, making efficiency decisions is key. In this work, efficiency is defined as choosing the suitable tool for a given task during the prototyping phase and utilizing this tool in a manner that is consistent with other designers that have performed similar tasks in the past. This work demonstrates how these decisions can be predicted based on designers' body movements.

This paper is organized as follows. This section provides an introduction and motivation for this research. Section 2 reviews related research. Section 3 introduces the method. Section 4 presents a case study involving participants in an engineering design workshop and demonstrates the feasibility of the methodology. Section 5 presents the results from the case study and discusses the potential of assessment systems integration during the prototyping phase of design. Finally, Sec. 6 concludes the paper.

2 Related Work

2.1 Automated Object Recognition

The prototyping phase of the engineering design process typically requires designers to interact with a set of design tools, towards the creation of a prototype/set of prototypes. A prototyping task that is being performed is highly dependent on the type of object being used and the designer(s) using that object, making an accurate identification of this object extremely important. Automated object recognition has been successfully employed in a variety of applications. Two-dimensional systems have employed Histograms of Oriented Gradients (HOGs [20]), or Oriented FAST and Rotated BRIEF (ORB [21]) for real-time reasoning on construction sites [22]. For Depth-enabled sensors, Bo et al. introduced a kernel descriptor based on RGB-D data [23] along with a data set to evaluate models. Building on this, Ren et al. used these sensors to achieve scene labeling [24].

These descriptors and methodologies take advantage of the ease with which 3D scenes are segmented, as well as their ability to estimate the pose of an object. However, given the limited scope of training data available for 3D based recognition, a 2D dimensional Convolutional Neural Network (CNN) is employed in this work. These were chosen because of their high accuracy over large numbers of classes [25] as well as the abundance of 2D image training data (e.g., video data). Such networks have been shown to perform well on large datasets in a competition setting [26, 27].

While these works consider the problem of object recognition in images, none consider objects in real-world contexts. Rather, they apply image recognition technologies to various datasets and demonstrate their abilities in controlled environments. This work makes advances by employing one of these technologies in conjunction with a depth enabled sensor, and uses that sensor to direct the attention of the image recognition technology, bridging the gap between static datasets and real-world applications.

2.2 Virtual/Augmented Reality For Task Learning

In the past, Virtual and Augmented Reality (VR/AR) have been employed for task learning. Dunleavy and Dede performed a literature review of instructional VR/AR systems across different fields [28]. Specifically in the engineering field, different approaches using this technology have been studied. Kosmadoudi et al. reviewed the potential use of game elements in Computer Aided Design settings, including both user experience measurements and user engagement [29]. Since many design tasks are 3-dimensional in nature, the utility of VR environments is clear. For example, Jezernick and Hren proposed a tool which can be used to create virtual worlds on web platforms [30], and Bourtdot et al. proposed a more general framework for working with CAD models in a virtual environment [31]. Verlinden and Horváth examined interactive prototyping through AR in the form of hints for specific applications [32], and Fiorentino et al. [33] proposed an augmented reality approach for manipulation of CAD designs. Notably, Vélaz et al. applied VR technology to the field of assembly learning, with participants learning how to assemble a product in several different modes [34]. While this attempted to solve a similar problem, it was primarily concerned with task completion, rather than assessment, and found that no improvement was offered by virtual environments.

Of these works which focused on task learning, the application to an engineering design space were limited (with the exception of [34]). The method presented in this paper will aid in task learning by learning and inferring a user's intent. The works which were more germane to engineering design focused on automating the design iteration process, rather than the physical prototyping stage. Instead this work considers the prototyping work performed in the physical space. However, once the design had been finalized and needed to be built, these methods were unable to provide any guidance for how to properly approach the following stage. While VR and AR are helpful in the field of rapid prototyping, implementation of a proposed design is still a manual one, thus there exists a knowledge gap in proper hands-on instruction for the prototyping phase.

2.3 Human Motion Modeling And Mining

The prototype stage of engineering design often includes manually and physically-involved processes. Therefore, in addition to automatically recognizing objects in a design environment, it is important to automatically recognize humans and how they move in a design environment. Human motion modeling has become more practical with the advent of commercially available, 3D body tracking sensors. Bradski et al. was able to determine poses as well as conduct motion-based music based on 2D video [35]. Using 2D video, Ribeiro and Santos-Victor were able to identify whether a human was running, walking, fighting, active, or inactive [36].

The Depth capabilities of RGB-D sensors have been successfully used by many works as well. By using 2.5D video data (that is, visible portions of scenes in a 3D space), Li et al. created a bag of words descriptor based on human poses, towards recognizing a dictionary of activities [37]. Morato et al. employed several 2.5D sensors to accurately model human motion [38], allowing humans and robots to move safely together in the same space, fostering better collaboration. Behoora and Tucker use an RGB-D sensor to analyze the body language of design team members in order to quantify higher level emotions such as boredom and interest [39]. In the context of engineering design, Tucker and Kumara have laid the foundation towards a body language model that captures individuals' body language data to model and predict the object that they may be using [40]. However, two fundamental limitations of this approach are that i) the method relies on the availability of ground truth data in terms of what body patterns relate to the objects being used, which is typically unavailable due to the real-time nature of design and ii) the method does not take into account the recognition of the objects themselves. Therefore, without an understanding of the object being used or the context of the task being performed, human motion modeling techniques alone may result in high false positives or false negatives. The work presented in this paper addresses these limitations by using both supervised machine learning (to accurately identify objects in an engineering workshop) and unsupervised learning (to detect anomalous human body language patters pertaining to a specific task).

While there has been ample work in the automatic detection of human body movement, existing works focus on achieving a high-level, semantically meaningful activity label, rather than what the body movement implies in the context of a specific task. For example, in a given engineering design workshop, there may be several tasks that possess similar body movement patterns. Additionally, existing methods typically model human motion in a supervised learning capacity. Thus there exists a knowledge gap in this field, since each of these methods seeks to model the motion for recognition or prediction purposes. The method presented in this paper uses an unsupervised method, as it is difficult to learn what the correct/incorrect way of performing a task is given the numerous ways a task can be performed correctly/incorrectly.

2.4 Activity Recognition and Feedback Systems

Automated Feedback Systems have been proposed across a wide range of fields. The feedback provided by these systems come in many forms. However, the common thread among them is the ability to quickly provide clear feedback to users in order to encourage behavioral change. Automatically providing feedback pertaining to an object's use is typically performed in robotics and in the field of activity recognition. Lopes and Santos-Victor used machine learning with observation to determine different object uses, based on different types of grips, in order to provide context to an activity recognition

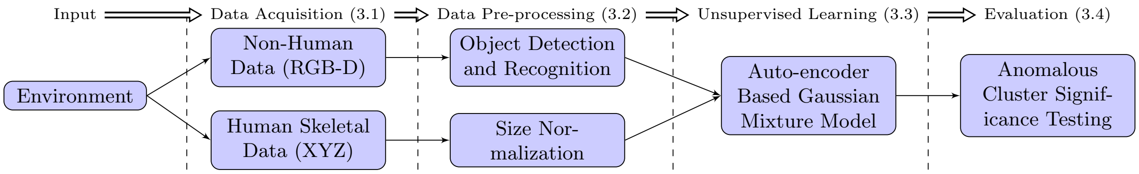

Input
Data Acquisition (3.1)
Data Pre-processing (3.2)
Unsupervised Learning (3.3)
Evaluation (3.4)
Environment
Non-Human Data (RGB-D)
Human Skeletal Data (XYZ)
Object Detection and Recognition
Size Normalization
Auto-encoder Based Gaussian Mixture Model
Anomalous Cluster Significance Testing

Fig. 1: An outline of the method presented in this work

system [41]. Jiang et al. used spatial data about scenes to determine which objects were likely to be used, based on ease of use and reachability measurements [42]. More recently, Koppula et al. used RGB-D to model activities as an ordered set of sub-activities, and were able to detect which activities were being performed under supervised learning conditions [43]. However, their method relied on training data for activity recognition. Similarly, Yu et al. proposed a method of activity recognition based on pose and temporal descriptors [44].

In the field of education, Weiling et al. found that students performed better in courses when given more evaluations with automated feedback [45]. Chen proposed a system that automatically provides feedback to students for programming assignments, encouraging them to fix mistakes, continue testing, or optimize their programs [46]. Calvo and Ellis proposed a system based on computational linguistic principles that provided comments about writing style and content for an assignment given to students [47].

While research in activity recognition and feedback is an active research area, the aforementioned methods rely on labeled training data of activities, towards the goal of a semantically meaningful and nuanced activity labeling. These methods assume that an interaction with an object is correct, and use this information towards predicting that label. Since these systems do not evaluate the object use for correctness, the feedback provided is a function of what is known *a priori* about correct or incorrect tasks. However, often the concept of correctness/incorrectness is not known *a priori*, since there are numerous ways to perform a task. Therefore, the method presented in this work limits the assumptions made about correct/incorrect task performance to that which is anomalous to the patterns observed by a wide range of designers that have performed a similar task in the past during the prototyping process. Table 1 summarizes which knowledge gaps the method will help to bridge. This work proposes an unsupervised model which can consider many modes of object use, and since it is unsupervised, requires little to no domain knowledge. This will also allow the model to change over time as methods evolve and new uses of objects present themselves. Additionally, training a model on every possible correct and incorrect use of an object is intractable, and would present many challenges in generalizing or scaling to similar problems. Instead the method in this work yields a model which can simply observe designer object use and learn to distinguish between correct and incorrect behavior, referred to as *Unsupervised Activity Analysis* in Table 1.

3 Method

This work hypothesizes that an object O being used for task i is distinguishable from object O being used for task j, where i is the correct task and j is the incorrect (and therefore less efficient) task in a sequence of tasks needed to complete a prototype. To test this hypothesis, a depth enabled sensor is used (Sec. 3.1), whose data streams are processed (Sec. 3.2)

Table 1: The contributions of selected related works, and where this work contributes

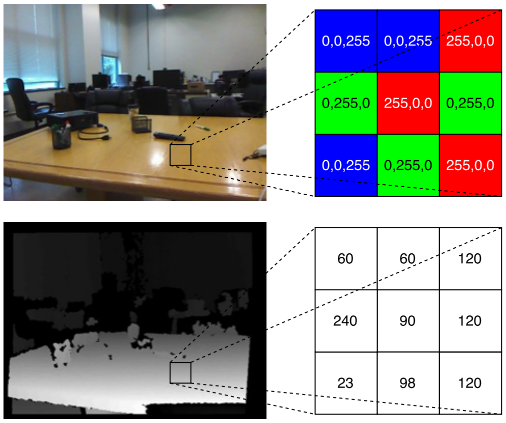
Work	Object Recognition	Activity Recognition	Object Use Detection	Sup'd Act Fdbk	Unsup'd Act Analysis
Krizhevsky et al. [26]	▨				
Ribeiro & Santos-Victor [36]		▨			
Patel et al. [48]				▨	
Tucker & Kumara [40]	▨	▨	▨		
This Work	▨	▨	▨		▨

so that objects can be recognized in the scene (Sec. 3.2.1). Section 3.3 describes the algorithm used to learn proper object use and how this information is used to determine if an anomalous design decision is being made while performing a given prototyping task. Finally, Sec. 3.4 discusses how this method will be evaluated.

3.1 Data Acquisition

3.1.1 Non-Human Data

An environment is considered to contain stationary and non-stationary objects. It is understood that objects at rest tend to stay at rest, unless acted upon by a non-stationary object. In the context of engineering design prototyping, these non-stationary objects are typically designers, or objects with which designers are interacting with. In this work, it is assumed that two synchronized data streams are acquired to sense an environment: i) data pertaining to color and ii) data pertaining to depth, as seen in Fig 2. Color data is represented by a matrix of pixels $h \times w$ in size, each pixel having a red, green, and blue value. By combining these three values, the color sensed per pixel can be determined. For example, a pixel with a color value of 255, 255, 255 represents the color *white*. This color data comprises one portion of the input to the object recognition system outlined in Sec. 3.2.1. This method also assumes a data stream of depth pixels of equal dimensions, $h \times w$. Each depth pixel contains one value, signifying the distance from the sensor to that pixel. This data provides a 2.5 dimensional representation of the surroundings, enabling fast and accurate object localization, and comprises part of the input to the object recognition system described in Sec. 3.2.1. This object recognition analyzes the content of bounding boxes to identify the object contained within them. To minimize false positives, it is important to ensure that these bounding boxes are likely to contain an object. Often, these boxes are generated based on the content of the image in question, however, this method proposes to use depth as the defining characteristic that generates these bounding boxes. This represents a large improvement over other approaches, such as naively searching an entire image for each box of a given size or sizes, as this would result in a large number of candidate boxes.



0,0,255	0,0,255	255,0,0
0,255,0	255,0,0	0,255,0
0,0,255	0,255,0	255,0,0

60	60	120
240	90	120
23	98	120

Fig. 2: Top: the RGB image data gathered from a scene (in RGB order); bottom: the depth information given by the sensor (in mm)

Depth data is also assumed to be a matrix of depth pixels. Each depth pixel contains just one value, corresponding to depth from the sensor to that pixel. Depth data provides a 2.5 dimensional representation of the surroundings, but cannot be relied upon alone for classification measures.

Finally, each x, y, z value is voxelized. This will arrange all points in a 3 dimensional grid according to their nearest coordinate. These coordinates are spaced out according to the margin of error of the sensor. This voxel distance d can be found in the documentation of any structured depth sensor, and serves as a sensitivity for this method. This eliminates sensor noise by changing the location of each point to the closest grid location, and reduces data volume by discarding points which are too close together without losing information. This data structure provides a 2.5D description of everything in view of the sensor, including color and depth, which will be used to analyze how a designer behaves in a given engineering design workspace. This work makes two assumptions about the body language exhibited by designers during a specific task. The first is that engineering tasks are performed on a workspace (e.g., table in Fig. 3) that is approximately half as tall as the designer. The second is that activities performed above the workspace (e.g., with the hands) are not necessarily influenced by nodes (e.g., feet) below the workspace.

3.1.2 Human Skeletal Data

Object use is characterized by human motion with respect to object position. In order to measure human motion, body node positions are acquired. Body node data is represented by a matrix of xyz values for each node in a human body, for each detected human, as shown in Fig. 3. For example in Fig 3, the rows in the tab represent the successive positions of the right hand node that are captured at each time interval. There are a wide range of sensing systems capable of capturing this data, such as OpenNI and Kinect. These values represent the location in a 2.5D space whose origin is the sensor, where x is the horizontal offset from the sensor, y is the vertical offset and z is the depth offset. For example, a position of the right hand node at the 20th instance (i.e. row 20 in Fig. 3) of (1,1,1) indicates that an individual's right hand is lifted above their head, where the values (1,1,1) represent 1 meter to the right (x), 1 meter above (y) and 1 meter back (z) from the sensing system. In subsequent sections, depending on an object that a person has in their hand, this may be indicative of the process of swinging an object. This data also contains node orientation data (i.e., *ow*, *ox*, *oy*, *oz* in Fig. 3). These four variables relating to node orientation are referred to as a quaternion and compactly represent an orientation of each node with respect to the parent node (for example, left wrist is the parent of left hand). These values represent both an angle of rotation and direction (such as rotated 90 degrees clockwise about the x-axis). These quaternions encode an orientation similar to Euler angles (that is, roll, pitch and yaw), without gimbal lock, a common problem of ambiguity when using Euler angles. Given that the sensing system is stationary and there exists variability in the size and relative position of individuals, a subsequent normalization step is needed in order to minimize bias in object-designer associations.

3.2 Data Pre-Processing

Given that the hypothesis of this work is that an object O being used for task i is distinguishable from object O being used for task j, where i is the correct task and j is the incorrect (and less efficient) task, the objective of the pre-processing step is to remove noise pertaining to the identification of objects and tasks. In order to detect anomalous task performance, it is necessary to determine which activity is being performed by the designer. An activity is defined by two attributes: the designer motion profile M and the identified object O, which is being used to perform an activity. These two attributes $\langle M, O \rangle$ define the activity being performed, with the first, corresponding to the feature space, defined by the second. In other words, each activity is defined by the combination of the object being used and the motion of the designer performing the action. Towards this end, the object detection and recognition section and size normalization will enable accurate detection and classification of tasks.

3.2.1 Object Detection and Recognition

Since this method is concerned with evaluating activities with respect to the objects being used, a highly accurate and reliable object recognition system must be employed. To accomplish this, a convolutional neural network is employed. This network requires two inputs: the first is the image of the scene, as described in Sec. 3.1.1. The other is a list of bounding boxes which may contain an object. In some settings, these boxes will be programmatically generated and proposed, which is a time consuming process. This method automatically segments these boxes based on the depth data provided by the sensor. To isolate the area around a skeletal node, Algorithm 1 is proposed. This fill-based algorithm includes a stopping criteria to prevent an entire body from being filled (as can occur, since the body is continuous). The rationale for this algorithm is that most tools are hand held [49], thus great effort should be taken to only identify regions closely associated with the hand. This ensures that an accurate classification can be made, while preserving spatial information, such as “the right hand holds a hammer”.

3.2.2 Size Normalization

Because designers may be at different locations relative to the sensor, and because of varying heights, an additional normalization step is needed. This will allow for better learning, and will reduce misclassification due to acceptable anomalies such as individuals who are too tall or too short, or individuals who decide to use an object at a different location relative to

Algorithm 1: Bounding Box Algorithm

```
Data: A list of Node locations $J$, depth matrix $D$
Result: A list of Bounding Boxes, Boxes
begin
    Boxes $\longleftarrow$ []
    for $j \in j$ do
        $Q \longleftarrow$ []
        let sensitivity = sensitivity of the sensor
        push $j$ onto $Q$
        // Calculate bounding line
        $r$ = Parent node of J
        // for example, the parent of hand is wrist
        $s = -\frac{r_x - j_x}{r_y - j_y}$
        line = line defined by slope $s$ intersecting $r$
        // slope of line perpendicular to line connecting $j$ and $r$
        let distance($p$,line) be distance between point $p$ and line
        while $Q$ is not empty do
            $p$ = pop($Q$)
            Push $p$ onto P
            for $n \in$ neighbors($p$) do
                if $|D[n] - D[p]| <$ sensitivity&&distance($p$,line) $\geq 1$ then
                    push $n$ onto $Q$
                end
            end
        end
        Box $= [\max_x(\mathsf{P}), \max_y(\mathsf{P}), \min_x(\mathsf{P}), \min_y(\mathsf{P})]$
        push Box onto Boxes
    end
end
```

the sensor. To accomplish this normalization, each node location is subtracted from the location of the designer's head node (see Fig 3). This serves to re-center the space, so that anomalies due to unusual placement in the workspace can be ignored, as these do not represent an incorrect task. Next, these new node distances from the head are divided by an empirically derived factor representing designer height. This factor is defined by the distance between two nodes which best correlates with designer height. This allows the method to successfully avoid anomalies caused by designers being of unusual height, either short or tall.

3.3 Unsupervised Learning

3.3.1 Auto-encoder

Given a feature vector $M = \langle m_1, \cdots m_p \rangle$ where m_i represents a node dimension (x, y, z, ow, ox, oy, oz), that is captured for a time interval $t_1, \ldots, t_m$, the goal of an auto-encoder is to transform this data from a p dimensional space to a q dimensional space and, in the process, discover inherent relationships between the dimensions. This is notably different from other unsupervised techniques, which can only discover relationships across samples, so the use of an auto-encoder is key to this method. While the node data exhibits temporal characteristics (for example, the position of hand node x at time t_3 is related to the position of hand node x at time t_2), the unsupervised learning algorithm takes as input the normalized data without a temporal component and assumes temporal independence of samples. In order to avoid arbitrary specification of temporal window size, independence is assumed, with the inherent patterns existing within the data set and discovered by the auto-encoder itself.

To analyze node location of skeletal data, first an auto-encoder is used, a common first step in unsupervised learning tasks [50]. An auto-encoder has two major components (each a neural network), the encoder and the decoder. The encoder works by changing the dimension of the data through a series of layers, which transforms the input into a new representation as output. The decoder then accepts this same output as input, and processes the output in to a faithful representation of the original input. These models attempt to minimize difference between the input and the output. In other words, an auto-encoder minimizes the loss L according to Eq. (1).

$$L = M - D(E(M)) \tag{1}$$

where M is the motion profile adjusted for height and head position from Sec. 3.2.2, E is the encoding network, whose input is the data M, and D is the decoding network, whose input is the encoded data $E(M)$, and whose output should be close to M after training. In this method, 80% of the motion data is provided as training input to the auto-encoder (one instance of a 5 fold cross validation). Since no label is provided along with this data, this is not a supervised learning task. After an auto-encoder has been trained, this innermost representation (that is, the output of the encoder), $M' = \langle m'_1, \cdots m'_q \rangle$, where m'_i represents a transformation of m_i, can be used for clustering.

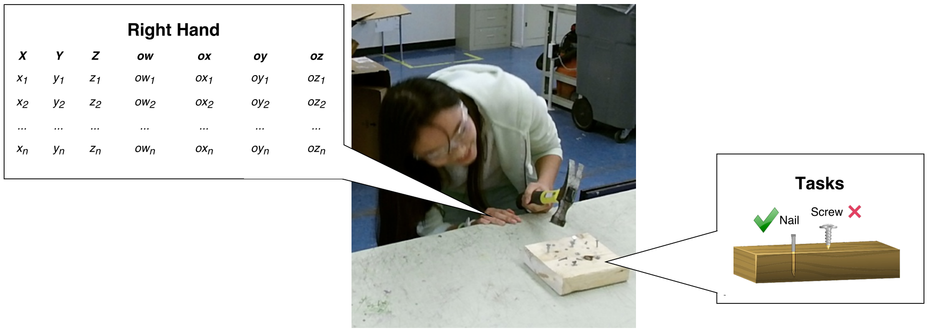
3.3.2 Gaussian Mixture Model Clustering

Next, a Gaussian Mixture Model (GMM) is used to cluster the transformed node vector M'. A GMM will model a set of data as a combination of Gaussian distributions, whose parameters are learned. This is the most common method of clustering when the number of clusters is unknown. For example, n number of designers using the same object for s tasks where some designers are performing the task correctly and some are not, would result in more than one cluster. For each vector M in the testing set of the size-normalized information from Sec. 3.2.2, the data cluster is given by Eq. (2).

$$C = GMM(E(M)) \tag{2}$$

where M is the motion profile adjusted for height and head position from Sec. 3.2.2, E is the encoding network, whose input is the data M, GMM is the trained Gaussian Mixture Model, and C is the cluster to which the vector M belongs.

The GMM output provides several different clusters that describe groups of designers that exhibit similar motion characteristics while performing certain tasks during the engineering prototyping process. Given these clusters, it would then be evident what clusters of motions by designers exhibit anomalous patterns, and what others do not. Since the tasks that can be



Right Hand

X	Y	Z	ow	ox	oy	oz
x_1	y_1	z_1	ow_1	ox_1	oy_1	oz_1
x_2	y_2	z_2	ow_2	ox_2	oy_2	oz_2
...	...	...	...	...	...	...
x_n	y_n	z_n	ow_n	ox_n	oy_n	oz_n

Fig. 3: A student beginning the "hammer screw" task in the engineering lab. For the two types of tasks (hammering a screw and a nail), this method determines if the tool they are using is efficient.

performed in an engineering workshop are known to designers, some clusters may contain more than one activity. Similarly, activities may span many clusters, therefore these clusters, whose residents were found to have similar movements, must be evaluated against the ground truth.

3.4 Evaluation

These clusters are evaluated for their cluster accuracy, and significance. To test the validity of the hypothesis, clusters that exhibit (i.e. argmax) anomalous activity patterns, are evaluated for significant statistical independence from other clusters. To do this, a two-sample Kolmogorov-Smirnov test is performed, which is a test of whether two distributions are significantly different, and yields a significance value indicating the likelihood that these are statistically significantly different [51]. This ensures that these outlying transformed skeletal nodes (M') are statistically significant after being detected by the GMM model. The null hypothesis H_0 being tested states that cluster i containing correct task performance and cluster j containing anomalous task performance are not statistically distinguishable. Thus, the alternate hypothesis, H_1 is that cluster i is distinguishable from cluster j. Knowledge gained from testing this hypothesis will inform designers of the efficient and inefficient approaches to performing a specific task during the physical prototyping process, as well as the best object for that task.

4 Case Study

A case study is presented in this section which evaluates the feasibility of the method presented in this paper. This study was conducted in accordance with the Pennsylvania State University IRB guidelines. 36 participants from an introductory Engineering Design class were provided with class credit and entered into a raffle.

4.1 Data Acquisition

4.1.1 Non-Human Data

This work utilized a Kinect 2 sensor as the primary sensor for the environmental data acquisition step. The Kinect 2 color images are 1920 × 1080 resolution, while the depth images are 512 × 424. The sensor was placed in an engineering design workshop that is typically used to design and create physical prototypes, as seen in Fig. 3. The sensor was placed 1 meter from the edge of the table on which participants would be working. The sensitivity of the sensor is 1mm.

4.1.2 Human Skeletal Data

The Kinect 2 is able to track 26 nodes, which were used for the unsupervised learning step (please see Sec. 3.3 of the method). Using a hammer, participants were asked to hammer three objects, a small nail, a larger nail, and a screw, into a piece of wood. They were instructed to hammer each of these objects into the wood until they were completely hammered or they felt they could not continue. The purpose of using a screw was to ensure that participants would need to hammer harder and in an unusual way, so that this model can be validated. Participants were not instructed to use the hammer in an unusual way so as to minimize bias that would be created by informing them of the task differences (e.g., hammering a screw with a hammer). As discussed previously, this method will distinguish between data for the correct task (hammering a nail), and data for the incorrect task (hammering a screw). The student in Fig. 3 is performing the hammer screw activity. This image shows the student attempting to line up the screw with the wood, in preparation to hammer it in to the wood (an incorrect and inefficient method of putting a screw in wood). After the sensor data acquisition was completed, participants were asked to take a short survey to ascertain their experience level with engineering, robotics, AI systems, as well as personal details such as height and area of study. Three of these participants' data was discarded due to collection errors.

4.2 Data Pre-processing

4.2.1 Object Detection and Recognition

Since this method requires a definitive identification of any object being used, the ability of the system to identify objects needs to be verified. The CNN was trained on 21 objects which fit three criteria. First, these objects must be available and annotated with bounding boxes from Imagenet. Second, these objects must have a sufficient number of samples available. The smallest class in the PASCAL object challenge contained 200 instances, so this served as the minimum. Finally, the object must be found in an engineering design environment. After these removals, 21 classes remained. Of these classes, Hammer, Scissors, Screwdriver and Goggles were available for testing the detection and recognition system. A model based on the design proposed by Krizhevsky et al. is trained for this task [26]. If objects are to be used in activity detection, they must be identified in a participant's hand. Of the 21 objects that were available, 4 objects were tested in a participant's hand. The precision, recall and F_1 of these objects can be seen in Table 2.

While these results are not accurate enough to claim that this system is reliable under all circumstances, the in-hand object system does function very accurately under controlled conditions. When a participant is asked to hold an object out away from their body against an unobtrusive background, the object is recognized nearly every time. Since the processing time of this system is approximately two seconds, this involves a participant standing still with an object for just two seconds prior to beginning use. So, while this system does not yet function reliably enough under real-world conditions, by employing this pre-condition before use, an accurate identification can be made, and the system can be confident in proceeding with this measurement. While this may slow down the process, it needs only happen once per user, and the alternative may be lack of access to the space, or using an improper tool for the task at hand. The results under these conditions can be seen in Table 3.

Table 2: Top-3 Precision and Recall of objects in hand.

Class	Precision	Recall	F_1
Screwdriver	0.463	0.886	0.608
Goggles	1.0	0.470	0.639
Scissors	0.6	0.125	0.207
Hammer	0.662	0.681	0.672

Table 3: Top-3 Precision and Recall of hand held object classes held out to the side away from the body

Class	Precision	Recall	F_1
Screwdriver	0.619	0.929	0.743
Scissors	1.000	0.111	0.200
Hammer	0.702	0.961	0.811

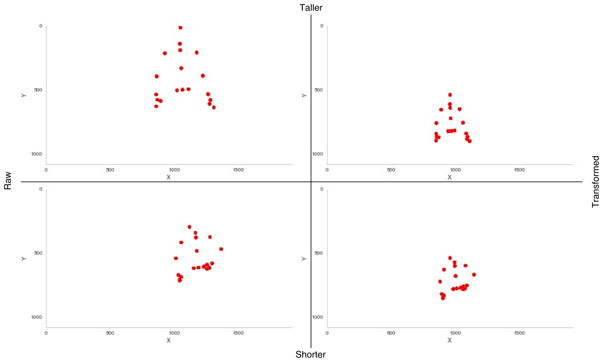


Fig. 4: The transformed skeletons of the shortest and tallest participants in the study

Table 4: Correlation across surveyed participants between node distances and height

Node Pair	Correlation
Right Hip to Spine Middle	0.913
Right Hip to Head	0.906
Right Hip to Spine Shoulder	0.905

4.2.2 Size Normalization

At each frame, the x, y and z values of a participant's head node is subtracted from each x y and z value. Additionally, each distance is divided by the distance between two nodes, derived empirically from this case study. Finally, the lower nodes are discarded since they are occluded by the desk. For each of these nodes, the following information was recorded: x, y, z positions in space, relative to the sensor (*3 elements*) and an orientation quaternion for the node, relative to its parent node (*4 elements*). These 7 attributes combine to form a vector of 84 (7 attributes for each of the 12 upper body nodes tracked) elements in length. The raw positions of two skeletons can be seen in Fig. 4. The large variation in size and position

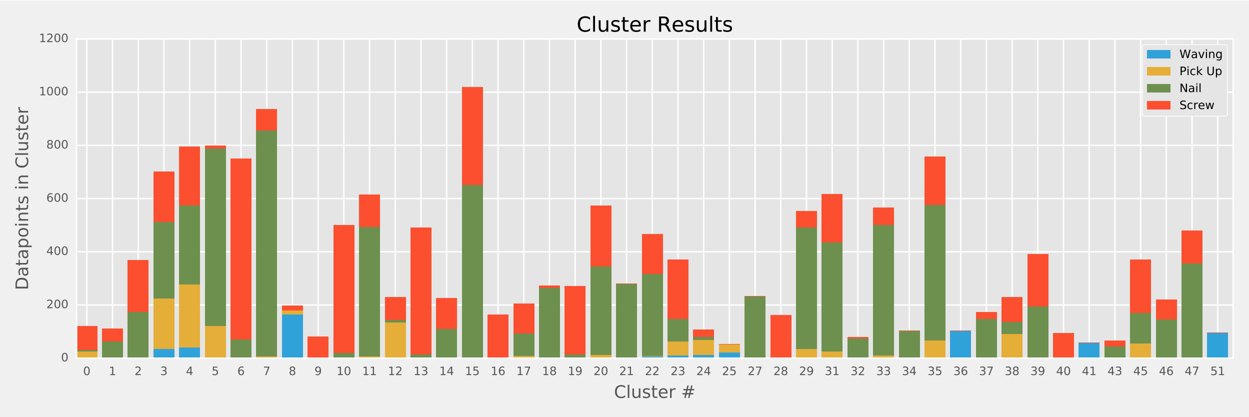


Fig. 5: Cluster Residency by activity label

motivates the normalization proposed in Sec. 3.3. To accomplish this, each location is first expressed with respect to the position of a participant's head. Next, because each participant may be a different size, the average distance between every pair of nodes in the participant is correlated to participant height, as shown in Table 4.

Since the distance between the right hip and mid spine is shown to correlate best with participant height, each relative position was divided by this distance, to ensure that all raw position values were adjusted for both difference in participant position and participant size. The entire transformed skeleton in 2D, shown in Fig. 4 on the right. This has greatly reduced the variation in both size and position of two participants.

4.3 Unsupervised Learning

4.3.1 Auto-encoder

The encoder and decoder of the auto-encoder used in this method consisted of four dense layers, of size 100, 200, 100, and 84 (the input size). While this did not change the dimensionality of the data, the data was transformed to become more context aware. That is, the data is now transformed by a machine which has been trained on all of the other data, so that the output of this auto-encoder is an encoded version of the input. This assists, in an unsupervised way, with discovering and encoding relationships between two features (for example, hand and wrist height are highly dependent), as well as learning patterns across samples (for example, heads are almost always oriented upwards towards the sensor, or downwards towards the table).

4.3.2 Gaussian Mixture Model

A DPGMM was used for this data with an upper bound of 200 clusters was set, but due to the nature of DPGMMs, this number is not necessarily reached, rather this serves as a stopping condition, which was not met, meaning that the number of clusters was sufficient to describe the data. The results of this clustering can be seen in Fig. 5. Along the x-axis each cluster is shown, while the y-axis shows the number of points belonging to each cluster. Each color represents a different activity, as labeled by the ground truth. The homogeneity measure of this clustering is 0.432. However, this method is primarily concerned with the clusters which are predominantly red (the *Screw* activity), which are referred to as *outlying* clusters.

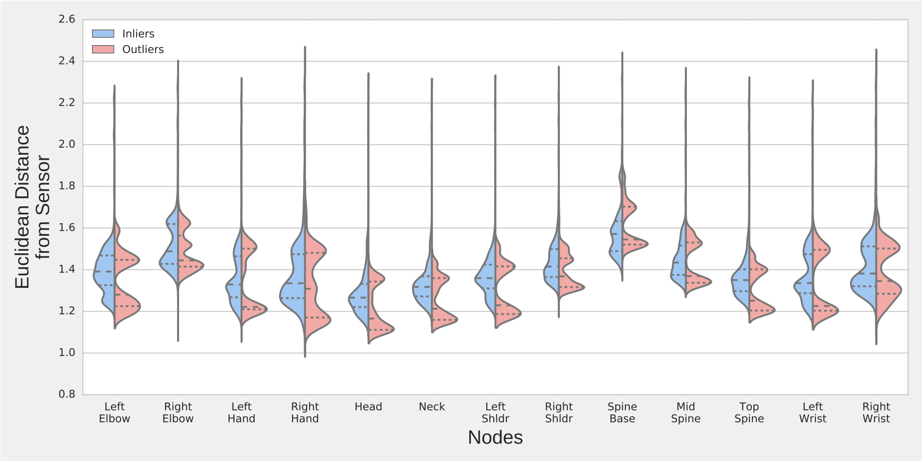


Fig. 6: Distribution plots for inlying and outlying distances

These clusters have a high probability of being an improper use of the tool, as defined by the ground truth. Since even improper use of an object may appear to be normal at a given moment, it is important to focus on when an identification can be made with high probability that a task is being done improperly. For example, a feedback system might provide feedback if too many irregular instances are detected in a short period. In other words, this figure shows for each detected cluster how many of each activity was assigned to this cluster by the model.

4.4 Evaluation

Figure 6 shows the distributions of the distance from the sensor for each node, across the testing set. The Y axis is the distance of the joint from the sensor, while the X axis gives the name of the joint. The left hand side of each plot represents the distribution of non-outlying clusters, while the right hand side represents the distribution of outlying clusters, for each node type, listed along the X axis. These show that the distributions are significantly different, with each distribution having $p << 0.05$ according to the two sample Kolmogorov Smirnov tests. These p-values can be seen in Table 5. This table shows each of the 12 nodes whose raw data is tested across cluster types (the head is omitted as it is always the origin). This test measures the probability (p-value) that the two one dimensional (that is, euclidean distance from sensor) distributions are the same. Note that not all nodes are included in this result, as lower body nodes were removed as noise, since the participant's lower body was occluded by the workspace table during the task in question. This p-value will enable a rejection of the null hypothesis H_0 if the value is low.

5 Results and Discussion

5.1 Object Detection and Recognition

These results indicate that the segmentation portion of this method is the current weak point, although an accurate identification of an object with 6 degrees of freedom may never be possible. For example, a system which identifies soda brand of a can of soda will never be able to identify the brand based on the bottom of the can. Similarly, many tools look similar from certain perspectives, given occlusion conditions and inaccuracies of the sensor which may interfere with either

Table 5: A two sample Kolmogorov Smirnov test of distance distributions

Node	p-value	Node	p-value
Left Elbow	2.62e-238	Right Elbow	2.41e-55
Left Hand	1.98e-323	Right Hand	2.70e-171
Top Spine	0.00e+00	Neck	0.00e+00
Left Shldr	0.00e+00	Right Shldr	1.36e-195
Right Wrist	2.97e-52	Left Wrist	2.85e-317
Spine Base	1.86e-168	Mid Spine	2.59e-162

the segmentation or identification portion of the sensor. One common problem encountered is that participants seem to face their tool towards the sensor, making identification of the tool unlikely since a hammer is very hard to identify looking from the top down. By adding in the condition that a participant must hold out the tool for identification beforehand, an accurate identification can be had nearly every time, without interfering with normal use of the object.

While this does not perfectly identify objects during all types of use, this is an important first step towards an accurate identification of objects as participants interact with them. Notable in these results is the high F_1 score for the screwdriver, despite the small size of the visible portion protruding from the hand. This may be in part due to the preponderance of screwdrivers in hands present in the training set. Future work might consider amassing a large enough training set of objects in hands to help improve the recognition for many types of objects. It is also worth noting that the goggles experiment was not a successful one, due to goggles' failure to show up on an IR based 2.5D sensor. Since goggles are transparent, they failed to reliably alter the IR beams that provide the depth measurements, thus the only detections of goggles came when the goggles were on the head of the participant. Still, the proposed segmentation algorithm performed well, given the sensor irregularities. In particular, longer thinner objects often had issues where the object would be mapped to the hand node rather than the the hand holding the object. However, holding the body in a more structured pose was shown by this work to counteract these problems.

5.2 Activity Recognition

The design of the tasks was to provoke inefficient behavior, without instructing the participants to behave inefficiently. Indeed it is possible to hammer a screw, though not effectively. Additionally, this can introduce many other issues as well, such as splitting a board, or being unable to complete the task (indeed, many participants were unable to complete the task and were allowed to give up). At the same time, these actions needed to be natural, while still being incorrect. Overall the clustering was a success since the homogeneity of these clusters is quite high. Even more encouraging is that the screw activities have very high residencies in several clusters. Since it is possible to behave correctly at a given point in time even while doing an incorrect task, having several clusters that are entirely incorrect provides a good starting point for developing an automated method of differentiating correct from incorrect activities. In other words, cluster 6 is approximately 90% incorrect, meaning that if an activity is in cluster 6, there is a 90% probability that it is being done improperly. Moving forward, the judgment could be made to provide feedback to participants if they are currently falling into one of these highly incorrect clusters. It is also notable that the waving task also exhibited these same characteristics, although the sample size was much smaller for this class. In particular, clusters 8, 36, and 51 are almost entirely waving instances, suggesting that this method may have broader applications in activity recognition. In contrast, the picking up a hammer activity seems to be much more spread out. This is probably due to the fact that each task required an individual picking up task within it (each nail and screw needed to be picked up, and this was not distinguished since these actions are part of the larger task). In contrast, the largest cluster, 15, has many incorrect datapoints in it, as well as correct ones. This demonstrates that it is possible to appear to be correct at certain moments during an incorrect task, further emphasizing the need to rely on clusters which are mostly incorrect.

In inspecting the distribution of distances from the sensor for each node (see Fig. 6) for these anomalous clusters, it is clear that these anomalous clusters are indeed statistically distinct from the non-anomalous clusters. In fact, the highest p-value according to the two sample KS tests is $2.97 * 10^{-52}$ (see Table 5). All of these are substantially lower than even 0.001, the lowest threshold in common use. Furthermore, the head, neck, and spine distributions each had a p value of 0. Likewise, the hands, which were used in the hammering task given to the participants, also boasted extremely low numbers. The combination of these two aspects suggest a possible use in ensuring safety as well, since distinguishing a different head and hand posture would be requirements for determining if an activity is being performed safely (for example, assisting participants in keeping their tools far from their heads).

6 Conclusions and Future Work

This work proposed a novel method of recognizing when an inefficient activity is being performed. By leveraging the human sensing capabilities of a 2.5D sensor, it analyzed body position data over the course of time. The 2.5D sensor also enabled a highly accurate classification of objects to be made by solving a long standing issue of object localization in a scene. This enabled an over 90% accurate detection of several classes of objects, while minimizing the risks of false positives. Given these recognitions, an unsupervised model was proposed that was able to differentiate between correct use of tools for a task and incorrect use of tools for a task. These differences were found to be statistically significant with $p < 0.001$.

Future work might consider how to use this as an automated process scheduling method, so that each appropriate tool and activity can be matched and laid out automatically, prior to beginning the prototyping phase. The evaluation methodology proposed here may also be used for other types of activity evaluation, including ensuring designer safety, as well as determining if an activity is likely to result in success. This may also lead to an automated teaching tool which could instruct designers in the best way to complete a task. Integrating this with downstream CAD software might allow determinations for a given design such as equipment or facilities required, number of workers, as well as parallelizable tasks. Future work should also consider how best to provide feedback when an anomaly is identified in an automatic and meaningful way. This problem is challenging as it requires deriving not only a classification of inefficiency, but insight into how the participant could behave more efficiently. Another approach might consider altering this system to analyze how new products are used. This would have applications in the field of ergonomics, as well as product design. And by monitoring the use of objects by expert craftsmen, an ideal use model could be generated and used for instruction.

Acknowledgements

The authors would like to acknowledge the NSF DUE/IUSE #1449650: Investigating the Impact of Co-Learning Systems in Providing Customized, Real-time Student Feedback. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Tuarob, S., and Tucker, C. S., 2015. "Automated discovery of lead users and latent product features by mining large scale social media networks". *Journal of Mechanical Design,* **137**(7), p. 071402.

[2] Tuarob, S., and Tucker, C. S., 2015. "Quantifying product favorability and extracting notable product features using large scale social media data". *Journal of Computing and Information Science in Engineering,* **15**(3), p. 031003.

[3] Wassenaar, H. J., Chen, W., Cheng, J., and Sudjianto, A., 2005. "Enhancing discrete choice demand modeling for decision-based design". *Journal of Mechanical Design,* **127**(4), pp. 514–523.

[4] Hoyle, C., Chen, W., Ankenman, B., and Wang, N., 2009. "Optimal experimental design of human appraisals for modeling consumer preferences in engineering design". *Journal of mechanical design,* **131**(7), p. 071008.

[5] Agard, B., and Kusiak*, A., 2004. "Data-mining-based methodology for the design of product families". *International Journal of Production Research,* **42**(15), pp. 2955–2969.

[6] Kusiak, A., and Smith, M., 2007. "Data mining in design of products and production systems". *Annual Reviews in Control,* **31**(1), pp. 147–156.

[7] Tucker, C. S., and Kim, H. M., 2011. "Trend mining for predictive product design". *Journal of Mechanical Design,* **133**(11), p. 111008.

[8] Gershenson, J. K., Prasad, G. J., and Zhang, Y., 2004. "Product modularity: measures and design methods". *Journal of engineering Design,* **15**(1), pp. 33–51.

[9] Kurtoglu, T., Campbell, M. I., Arnold, C. B., Stone, R. B., and Mcadams, D. A., 2009. "A component taxonomy as a framework for computational design synthesis". *Journal of Computing and Information Science in Engineering,* **9**(1), p. 011007.

[10] Bonjour, E., Deniaud, S., Dulmet, M., and Harmel, G., 2009. "A fuzzy method for propagating functional architecture constraints to physical architecture". *Journal of Mechanical Design,* **131**(6), p. 061002.

[11] Wang, G. G., and Shan, S., 2007. "Review of metamodeling techniques in support of engineering design optimization". *Journal of Mechanical design,* **129**(4), pp. 370–380.

[12] Apley, D. W., Liu, J., and Chen, W., 2006. "Understanding the effects of model uncertainty in robust design with computer experiments". *Journal of Mechanical Design,* **128**(4), pp. 945–958.

[13] Hannah, R., Joshi, S., and Summers, J. D., 2012. "A user study of interpretability of engineering design representations". *Journal of Engineering Design,* **23**(6), pp. 443–468.

[14] Gerber, E., and Carroll, M., 2012. "The psychological experience of prototyping". *Design studies,* **33**(1), pp. 64–84.

[15] Yang, M. C., 2005. "A study of prototypes, design activity, and design outcome". *Design Studies,* **26**(6), pp. 649–669.

[16] Houde, S., and Hill, C., 1997. "What do prototypes prototype". *Handbook of human-computer interaction,* **2**, pp. 367–381.

[17] Eppinger, S., and Whitney, D., 1995. "Accelerating product development by the exchange of preliminary product design information". *Journal of Mechanical Design,* **117**, p. 491.

[18] Teizer, J., Venugopal, M., and Walia, A., 2008. "Ultrawideband for automated real-time three-dimensional location sensing for workforce, equipment, and material positioning and tracking". *Transportation Research Record: Journal of the Transportation Research Board*(2081), pp. 56–64.

[19] Lim, Y.-K., Stolterman, E., and Tenenberg, J., 2008. "The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas". *ACM Transactions on Computer-Human Interaction (TOCHI),* **15**(2), p. 7.

[20] Dalal, N., and Triggs, B., 2005. "Histograms of oriented gradients for human detection". In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, pp. 886–893.

[21] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., 2011. "Orb: An efficient alternative to sift or surf". In Computer Vision (ICCV), 2011 IEEE international conference on, IEEE, pp. 2564–2571.

[22] Chi, S., and Caldas, C. H., 2011. "Automated object identification using optical video cameras on construction sites". *Computer-Aided Civil and Infrastructure Engineering,* **26**(5), pp. 368–380.

[23] Bo, L., Ren, X., and Fox, D., 2011. "Depth kernel descriptors for object recognition". In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE, pp. 821–826.

[24] Ren, X., Bo, L., and Fox, D., 2012. "Rgb-(d) scene labeling: Features and algorithms". In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, pp. 2759–2766.

[25] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. "Imagenet large scale visual recognition challenge". *International Journal of Computer Vision,* **115**(3), pp. 211–252.

[26] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012. "Imagenet classification with deep convolutional neural networks". In Advances in neural information processing systems, pp. 1097–1105.

[27] Simonyan, K., and Zisserman, A., 2014. "Very deep convolutional networks for large-scale image recognition". *CoRR,* **abs/1409.1556**.

[28] Dunleavy, M., and Dede, C., 2014. "Augmented reality teaching and learning". In *Handbook of research on educational communications and technology*. Springer, pp. 735–745.

[29] Kosmadoudi, Z., Lim, T., Ritchie, J., Louchart, S., Liu, Y., and Sung, R., 2013. "Engineering design using game-enhanced cad: The potential to augment the user experience with game elements". *Computer-Aided Design,* **45**(3), pp. 777–795.

[30] Jezernik, A., and Hren, G., 2003. "A solution to integrate computer-aided design (cad) and virtual reality (vr) databases in design and manufacturing processes". *The International Journal of Advanced Manufacturing Technology,* **22**(11-12), pp. 768–774.

[31] Bourdot, P., Convard, T., Picon, F., Ammi, M., Touraine, D., and Vézien, J.-M., 2010. "Vr–cad integration: Multimodal immersive interaction and advanced haptic paradigms for implicit edition of cad models". *Computer-Aided Design,* **42**(5), pp. 445–461.

[32] Verlinden, J., and Horváth, I., 2009. "Analyzing opportunities for using interactive augmented prototyping in design practice". *AI EDAM,* **23**(3), pp. 289–303.

[33] Fiorentino, M., Uva, A. E., Monno, G., and Radkowski, R., 2012. "Augmented technical drawings: a novel technique for natural interactive visualization of computer-aided design models". *Journal of Computing and Information Science in Engineering,* **12**(2), p. 024503.

[34] Vélaz, Y., Arce, J. R., Gutiérrez, T., Lozano-Rodero, A., and Suescun, A., 2014. "The influence of interaction technology on the learning of assembly tasks using virtual reality". *Journal of Computing and Information Science in Engineering,* **14**(4), p. 041007.

[35] Bradski, G. R., and Davis, J. W., 2002. "Motion segmentation and pose recognition with motion history gradients". *Machine Vision and Applications,* **13**(3), pp. 174–184.

[36] Ribeiro, P. C., Santos-Victor, J., and Lisboa, P., 2005. "Human activity recognition from video: modeling, feature selection and classification architecture". In Proceedings of International Workshop on Human Activity Recognition and Modelling, pp. 61–78.

[37] Li, W., Zhang, Z., and Liu, Z., 2010. "Action recognition based on a bag of 3d points". In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, IEEE, pp. 9–14.

[38] Morato, C., Kaipa, K. N., Zhao, B., and Gupta, S. K., 2014. "Toward safe human robot collaboration by using multiple kinects based real-time human tracking". *Journal of Computing and Information Science in Engineering,* **14**(1), p. 011006.

[39] Behoora, I., and Tucker, C. S., 2014. "Quantifying emotional states based on body language data using non invasive sensors". In ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V01AT02A079–V01AT02A079.

[40] Tucker, C., and Kumara, S., 2015. "An Automated Object-Task Mining Model for Providing Students with Real Time Performance Feedback". In 2015 ASEE Annual Conference and Exposition.

[41] Lopes, M., and Santos-Victor, J., 2005. "Visual learning by imitation with motor representations". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* **35**(3), pp. 438–449.

[42] Jiang, Y., Lim, M., and Saxena, A., 2012. "Learning object arrangements in 3d scenes using human context". *arXiv preprint arXiv:1206.6462*.

[43] Koppula, H. S., Gupta, R., and Saxena, A., 2013. "Learning human activities and object affordances from rgb-d videos". *The International Journal of Robotics Research,* **32**(8), pp. 951–970.

[44] Yu, G., Liu, Z., and Yuan, J., 2014. "Discriminative orderlet mining for real-time recognition of human-object interaction". In Asian Conference on Computer Vision, Springer, pp. 50–65.

[45] Wieling, M., and Hofman, W., 2010. "The impact of online video lecture recordings and automated feedback on student performance". *Computers & Education,* **54**(4), pp. 992–998.

[46] Chen, P. M., 2004. "An automated feedback system for computer organization projects". *IEEE Transactions on Education,* **47**(2), pp. 232–240.

[47] Calvo, R. A., and Ellis, R. A., 2010. "Students' conceptions of tutor and automated feedback in professional writing".

Journal of Engineering Education, **99**(4), pp. 427–438.

[48] Patel, R. A., Hartzler, A., Pratt, W., Back, A., Czerwinski, M., and Roseway, A., 2013. "Visual feedback on nonverbal communication: a design exploration with healthcare professionals". In Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013 7th International Conference on, IEEE, pp. 105–112.

[49] Lamancusa, J. S., 2006. "The reincarnation of the engineering shop". In ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 849–857.

[50] Le, Q. V., 2013. "Building high-level features using large scale unsupervised learning". In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, pp. 8595–8598.

[51] Massey Jr, F. J., 1951. "The kolmogorov-smirnov test for goodness of fit". *Journal of the American statistical Association,* **46**(253), pp. 68–78.

List of Tables

List of Figures