# Harmonic synchronisation of hand gestures with visual displays

Torem Ozturk

MSc Data Science
The University of Bath
May 2021

# Table of Contents

Chapter 1 ..... 3

Introduction ..... 3

Chapter 2 ..... 5

Literature Review ..... 5

2.1 Motion Correlation ..... 5

2.2 Motor Perception and Behaviour ..... 9

2.3 Object Detection ..... 10

Chapter 3 ..... 16

Methodology ..... 16

3.1 Study Design ..... 16

3.2 Timeline ..... 16

Chapter 4 ..... 16

Conclusion and Future Work ..... 16

Bibliography ..... 17

# Chapter 1

# Introduction

Most if not all modern-day technical systems are controlled by dedicated input devices that use tactile feedback. Many benefits have arisen from such devices especially from remote and wireless controllers which allow users to be mobile and exhibit control over technical systems. A multitude of papers discuss the use of external devices that map predefined functionality of physical objects to commands, (Greenberg and Boyle, 2002), (Avrahami, Wobbrock and Izadi, 2011) and (Held *et al.*, 2012).

There is however a myriad of issues that can arise from using such dedicated devices, the most notable being a lack of 'improvisation' (Corsten *et al.*, 2013). For example, take the common case of when an individual wishes to change the setting on a TV and cannot find the remote or it is simply out of reach, then the usability and convenience of such systems are drastically reduced. This can be a frustrating experience and by the time they have the remote for the task that it was needed for, e.g., increase the volume for a particular scene, may have passed. If the strict relationship between device and system could be loosened, then this could greatly benefit the user in a variety of situations.

To address these issues users can use objects that can be paired with a technical system, there are multiple triggers in which this can be done. (Corsten *et al.*, 2013).

1) User triggered – when a user picks up an object it is detected and spatially paired with the system.
2) Time triggered – after a certain duration of time of non-use the object is automatically paired with the system.
3) Proximity-triggered – the object pairs when in close proximity with another object or reference point.
4) Situation-triggered – when an application is running such as PowerPoint then an object automatically becomes an input device.

The drawback from pairing an object through time and proximity triggers is that they are prone to false positives as these methods will increase the cognitive load of the user. As both methods will require the user to remember specific temporal or spatial measurements which can be confusing and result in mistakes made by the user. This is likely to occur if there are no visual displays given, and if they are included, they may potentially clutter the screen.

Corsten uses user triggered instantiation when pairing objects to a technical system (Corsten *et al.*, 2013). Again, this form of pairing is prone to false positives. For example, if the user wanted to use an object such as their phone, they must make sure that system does not mistakenly detect the action as pairing when they pick up the device to take a phone call or scroll social media. This is likely to be a common issue given the fact that 88% of Americans use a second screen whilst watching TV (*BOND - Internet Trends 2019*, 2019).

One way this relationship could be loosened further is by using motion correlation detection, the synchronisation of an object's movement with an external stimulus. When using motion of the object to synchronise with the system, hands are of a particular interest. This is because how people arrange and orientate their hands and bodies relative to their surroundings provides important information about the nature of their intentions and attitudes towards others and the environment (Kendon, 2004).

There have been examples where motion correlation has been successfully implemented, most notably MatchPoint (Clarke and Gellersen, 2017), where users are able to spontaneously instantiate any object, including the user's hands by matching the motion displayed on the screen. This is done with a technical system using only a basic webcam. They address several shortcomings of other implementations such as 'Midas Touch', where users accidentally instantiate the system when trying to do another activity. Using non-trivial movements, the user is able to overcome this issue by only instantiating an object after having done a deliberate action.

This project intends to expand on MatchPoint's work by focusing on hand gestures and their available functionality. Contributing further to motion correlation literature by examining rotational and grasping movements, fundamental hand gestures. Hands are of particular interest as arguably they are what separates man from other animals, without them our ancestors would not have been able to make tools, build cities and I would not be able to type these words you are reading now. Even though users can use any object within MatchPoint, the movements available after instantiating the object are limited to translational movements.

The intended contributions made by this project are as follows:

1) Develop the ability to spontaneously instantiate the user's hands with a technical system through multiple techniques. This will allow for the system to detect whether the user wishes to move their hand translationally, rotate their hand or carry out a grasping motion.

2) Accurately measure the distance moved, rotation angle or amount of grasping done by the user's hands and thus their intended actions using a standard webcam. This will be done using user studies and requiring feedback on the sensitivity of the hand detection. This will allow the user to control the system with precision.

3) The creation of graphical designs that visually represent the motions of circular movement, rotation and grasping to the user. This is so users clearly understand the options available to them when controlling the system.

This all attends to be achieved whilst using a device that has an Intel Core i5-7200U CPU and a RAM of 8GB.

# Chapter 2

# Literature Review

A comprehensive literature review has been conducted in the proceeding areas concerning Motion Correlation, Motor Perception and Behaviour and Object Detection.

## 2.1 Motion Correlation

The method of motion correlation, the spatial coupling of an object and technical system through synchronous movements may help address the potential issues that can arise from other triggers discussed in the introduction. Previous work has already explored the use of gaze and or body part motion as spatial input (Clarke and Gellersen, 2017), (Vidal, Bulling and Gellersen, 2013) and (Esteves *et al.*, 2015). Motion correlation is defined by the three following properties (Velloso *et al.*, 2017):

1) *Objects available for interaction are represented by a motion displayed to the user.*
2) *Users express their intent to interact with an object through movement at the interface that corresponds with the object's motion.*
3) *Correlation of the system's output and the user's input determines the selection of objects for interaction.*

The underlying principle is that any selectable target on the interface represents a distinct movement that the user can select through synchronisation. A match between the object, the input, and the displayed target, output, will be detected based on the alignment of both their spatial and temporal properties according to some similarity measure. Spatial properties refer to the any property relating to space (e.g. position), and temporal properties any that relate to time (e.g. velocity).

### 2.1.1 Motion Correlation with Everyday Objects

Clarke *et al.* explores using objects as a means of spontaneous coupling with the technical system, this is done through both TraceMatch (Clarke *et al.*, 2016) and MatchPoint (Clarke and Gellersen, 2017), which uses TraceMatch as its processing pipeline. The benefit of TraceMatch is that it allows for the instantiation of objects without any prerequisite training and can be done with a standard inexpensive webcam. This is done through two stages when analysing a scene from the camera feed, the first stage is 'generous' and will consider any motion within the frame to be a potential object for recognition which thus could be turned into an input device. The system will track the movement of any object feature. The second stage is the matching phase, where the motion of the detected object is matched against the movement of the visual display. This is done through both the use of path correlation and model-fitting. To avoid the emergence of 'Midas Touch' the visual displays used circular movement, as it was assumed to be unlikely that the user would accidentally match with the technical system when doing other activities in front of the webcam.

The object detection algorithm used in stage one is *Features from accelerated segment test (FAST)* (Rosten and Drummond, 2006) which extracts features of interest depending on neighbouring pixels. If the intensity of these neighbouring pixels is systemically higher or lower, then that feature is extracted. The benefit of using FAST is that it is computationally efficient, making it much faster compared to other existing feature detectors. This makes it a suitable algorithm for object detection within videos. Within the second stage the Pearson product-moment correlation coefficient was used to measure the similarity between the motion of the object and the display on the screen. The equation below represents the horizontal correlation of the Pearson's x product coefficient, $corr_x$. The inputs of this equation are the horizontal positions of both the input device, $Input_x$, and the object that is displayed on the screen, $Obj_x$. This will be calculated for the y axis as well. If a certain threshold is surpassed by both these correlations, then the object will sync with the system. $\overline{Input_x}$ and $\sigma_{Input_x}$ are the mean and standard deviation of the horizontal input's position and $\overline{Obj_x}$ and $\sigma_{Obj_x}$ are the same measurements but for the object's horizontal position. In TraceMatch if the correlation coefficient for both axes was above a certain threshold, then an instantiation would be registered.

$$corr_x = \frac{E\left[(Input_x - \overline{Input_x})(Obj_x - \overline{Obj_x})\right]}{\sigma_{Input_x}\sigma_{Obj_x}} \qquad (1)$$

Clarke *et al.* wanted to understand how often the system would pick up false positives, so they encouraged the users to carry out normal activities in front of the screen by allowing the users to watch TV or browse the internet followed by talking with a researcher. This part of the study will also be carried out in this proposed project, to understand the impact that day-to-day activities will have on producing false positives when synchronising with visual displays that are not only circular but also rotational and grasping.

When the targets were moving at a relatively slow rate - 0.25Hz, taking four seconds to complete a full cycle - the study found that accurate synchronisation occurred more frequently when the circular widgets increased in size. This effect however was not as apparent when the frequencies were higher and more accurate synchronisation occurred. Implying that faster targets allow for the more accurate synchronisation with the circular widget. This may be the case because for slower targets the eye may be unable to fixate on them and a saccade, a rapid movement of the eyes to a fixed point, will occur more frequently. A saccade is followed by a delayed movement of the hand that occurs approximately 100 milliseconds later when trying to move the hand to where the eye is fixated (Prablanc *et al.*, 1979). This may prevent the hand from effectively synchronising with the moving target because saccades occur continuously at slower target velocities.

The initial design within this proposal for detecting rotation will be a semicircle with a target moving from one side to another. Synchronisation will be achieved by rotating the hand in tandem with the moving target. It is assumed that the same results as Clarke *et al.* will be found, with the sensitivity of the system increasing with both the size and speed of the moving target. The same result for grasping is unlikely to be found, this will be done using a display of a shrinking and growing circle and therefore people's movement will be limited by both the size of their hand and palm.

Ways in which this motion correlation pipeline can be used in real-world settings is discussed within MatchPoint (Clarke and Gellersen, 2017). MatchPoint builds upon TraceMatch by allowing users to select a control on the system and pair it with an object as the input device. The other pipeline used within MatchPoint supports the tracking of the object after it has been spatially instantiated, this allows the user to manipulate the control by moving that object. The pipeline used for tracking was able to cope with unpredictable movements and changes in perspectives of the object or body part. This is important as the user may move the respective object or body part into a different location within the frame that is closer or further away. Tracking that accounts for this is vital, as it prevents the requirement to instantiate the object repeatedly.

Clarke *et al.* goes on to evaluate these two pipelines by discussing their performance on three different objects, the user's head, the user's hand, and a cup. Figure 1 shows their results from the study, a general trend can be observed from both results, as the target size increases, the movement time and the target re-entries decrease. Target re-entries are defined as when the pointer, in this case the respective object, enters the target region, leaves, and then re-enters the target region. If this occurs twice in ten trials then this will be recorded as a score of 0.2 (Mackenzie, Kauppinen and Silfverberg, 2001).

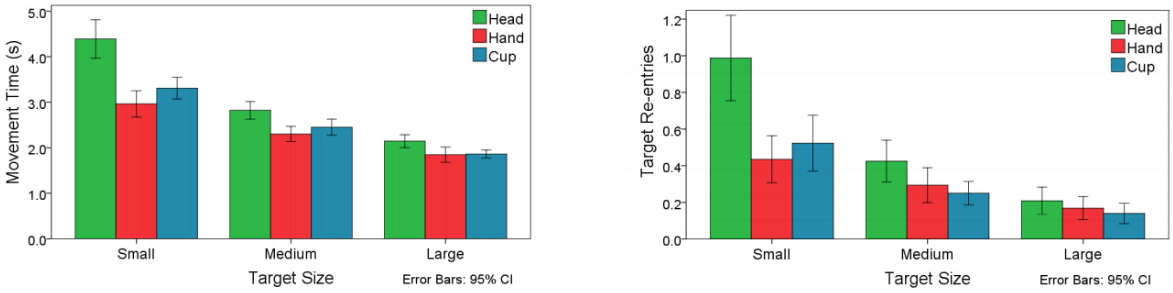


**Figure 11. Movement times for each target size and input modality.**

**Figure 12. Target re-entries for each target size and input modality.**

Figure 1. Left, movement times for each target size and input modality. Right, target re-entries for each target size and input modality (Clarke and Gellersen, 2017).

The relatively poor performance of the head may be due to two reasons that Clarke discusses, firstly the head is not usually used as a pointing device by the user compared to their hand or a cup, which would effectively be an extension of their arm. Secondly due to the smaller range of movement of the hand compared to the other two inputs, movement may result in fractional changes per pixels between frames. As anti-jitter is applied to the cursor point, discrepancy changes in location and a smoother movement for the cursor may introduce a time lag when using the head as the pointing device.

There are however several shortcomings with MatchPoint, firstly it does not account for hand occlusion, the proposed system in this project will not suffer from is issue as it will not detect objects besides hands. There is the possibility for one hand to occlude another however this is unlikely to happen as no displayed action will require the user to overlap hands. Secondly simultaneous motion could result in incorrect tracking, for example detecting the elbow instead of the hand as the entire arm moves. This would not happen with this system as the detection will only account for hands.

## 2.1.2 Motion Correlation with Hand Gestures

Touchless in-air gestures using hand movements have been investigated by Freeman *et al.* in 'Do That, There' (Freeman, Brewster and Lantz, 2016). In their paper they discuss the use of different feedback mechanisms that can be used in the absence of a screen, this includes audio, tactile and interactive light displays to indicate to the user where they needed to place their hand when synchronising with the technical system. This is their 'There' part of the paper, showing users where to gesture. When performing hand movements using a gesture system it is important for the user to be able to identify where to perform gestures, otherwise they may not be successfully detected.

In their 'Do That' section they demonstrate to the users how to direct input to the technical system and provide audio, tactile or light displays when the user successfully synchronised with the device. This is done through a new input technique that they introduce called *rhythmic gestures*, where gestures are repeated periodically in time with a rhythm. They use rhythmic light animations due to the reasons discussed in 2.1 Motor Perception and Behaviour. Figure 2 below shows the five different rhythmic gesture movements that users were asked to do in the study.

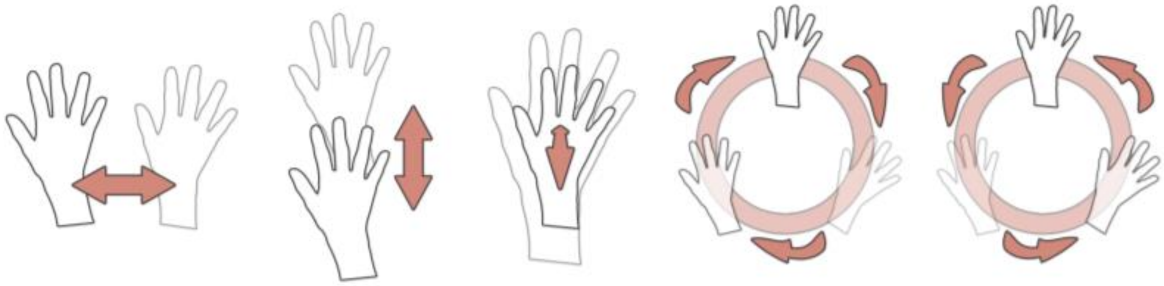
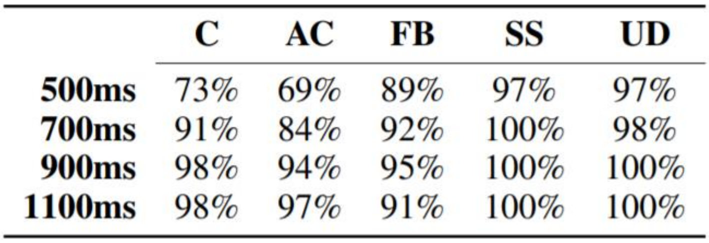


Figure 2. Five different rhythmic gesture movements from left to right: Side-to-Side, Up-and-Down, Forwards-and-Backwards, Clockwise and Anticlockwise circular movements. (Freeman, Brewster and Lantz, 2016)

The synchronisation of these five different gesture types were done at four different time intervals, the time in which one cycle was completed: 500, 700, 900 and 1100 milliseconds. The concluding remarks of this paper are relevant to this proposal, as they draw an insight to how a study should be conducted to obtain the desired result of effective synchronisation. Firstly, use interactive light to show users where to gesture, as users tended to spend less time trying to find where to gesture when provided with visual displays compared to other sensor feedback. The external stimulus provided to the user in this project will be visual displays. Secondly, feedback should be given to the user about rhythmic gestures from the start of the movement. This project could potentially display to the user the current location of their hand as a cursor on the screen. Thirdly, use side-to-side and up-and-down when possible as they were the easiest rhythmic movements to mimic, instead of circular movements. The issue with using these vertical and horizontal movements is that they are likely to suffer from the Midas touch problem. Vertical and horizontal movements may simply be too trivial, causing too many instances of false positives occurring. This project will intend to examine the likelihood of these issues when implementing rotational and grasping motions.

Finally, the intervals of the rhythmic gestures should last at least 700 milliseconds and for circular gestures at least 900 milliseconds. They define the time intervals as follows, for circular displays it was the time needed to complete one oscillation whilst for other movements it was the time taken until stopping the hand, double this amount of time would mean that the hand would have returned to its original position. Table 1 below shows the success rate of matching for each gesture at different speeds. These results suggest that higher time intervals, at least for those examined in this paper, are the most appropriate to use when implementing a touchless gesture system.



| | C | AC | FB | SS | UD |
|---|---|---|---|---|---|
| **500ms** | 73% | 69% | 89% | 97% | 97% |
| **700ms** | 91% | 84% | 92% | 100% | 98% |
| **900ms** | 98% | 94% | 95% | 100% | 100% |
| **1100ms** | 98% | 97% | 91% | 100% | 100% |

Table 1. Displays the success for each gesture at different time intervals. (Anti-)Clockwise (AC, C), Forwards-and-Backwards (FB), Side-to-Side (SS), and Up-and-Down (UD). (Freeman, Brewster and Lantz, 2016)

## 2.2 Motor Perception and Behaviour

To determine the most appropriate method for motion correlation to be achieved, it must be understood how humans synchronise with external stimuli and the most effective way in which they do so. The literature in this section below develops upon work discussed in Clarke's thesis (Clarke, 2020).

Numerous studies in the field of human physiology have shown that humans have the innate ability to register and synchronise with external stimuli. Consider visual cues presented in the form of moving targets that will allow for the hand to synchronise with a technical system as shown in TraceMatch, MatchPoint and Do That, There. All of which use moving targets to synchronise the user with a technical system. When presented with several targets, an individual's eyes and hands tend to move towards the same target (Gielen, van den Heuvel and van Gisbergen, 1984). This occurs regardless of whether the individual can view their limb or not (Pelisson *et al.*, 1986), since our central nervous system enforces a co-alignment of both our ocular and motor systems (Neggers and Bekkering, 2001). Therefore, the use of visual displays should not influence the individual's motor abilities when trying to focus on external stimuli and should not be of concern for this project.

The movement of the eye during this process exhibits two distinct stages, firstly there will be a saccade, a rapid movement of the eyes to a fixed point followed by a delayed hand movement roughly 100 milliseconds later (Prablanc *et al.*, 1979). Once an individual has fixated on a target and that target begins to move, then the eye movement that allows for primates to keep the object of interest in focus is pursuit. Pursuit is the continuous movement of the eye both slowly and smoothly to compensate for any motion of the visual target. This reduces any drift of the target's image across the retina that might otherwise blur the image (Krauzlis, 2005). Changes in smooth pursuits however are not velocity instantaneous and therefore cannot reach large velocities in short periods of time. It also cannot track extremely fast-moving objects. As a result of these limitations the ocular system will try and compensate by anticipating the future trajectory of the moving target. This compensation is again limited to when the future position of the target is predictable so that the participant can extrapolate from past movement, therefore anticipation fails in all scenarios in which there are unpredictable changes in the target's trajectory (Orban De Xivry and Lefèvre, 2007). This supports the notion that the trajectory and velocity characteristics of the displays used in this project should be predictable, to allow for the successful synchronisation of external stimulus and hand movement and thus motion correlation.

Sensorimotor synchronization (SMS), the desired outcome from the use of motion correlation, is defined to be the rhythmic coordination of perception and action (Repp, 2005). As this project will make the use of screens it makes sense to make use of visual stimuli to synchronise with. Findings from the literature reveal that performance of visuomotor synchronisation is improved when the visual stimulus and the user arm exhibit compatible spatial information, when moving in the same direction (Hove, Spivey and Krumhansl, 2010). If visual stimulus is moving in the opposite direction, the strength of the perception-action couple severely impedes the quality of the arm movement with that external event (Buekers *et al.*, 2000). Therefore, it makes intuitive sense to design displays that require the user to move in the same direction in this project. Even though this is the case, auditory SMS seems to remain superior to visual SMS (Hove, Spivey and Krumhansl, 2010). This form of sensory input however will not be considered in this project, as persistent auditory stimulus may become a nuisance to the user and the rest of the audience when the technical system is implemented in a real-world setting, such as when watching a film in the living room. The use of an auditory stimulus may be more accurate however it would stop the activity of watching a film entirely, preventing the generalisability of such a system to other activities.

It is also interesting to note that mechanical SMS is heavily impacted by the axis of rotation, for example Carson *et al* investigated the impact of synchronisation of hand rotation with an auditory metronome that increased in frequency over time. In their trials they investigated when the axis of rotation was either with the long axis of the forearm, above this axis, or below. They found that the stability of pronation of the wrist in time with the rhythm was most stable when the axis of rotation of the movement was below the forearm's long axis, this means when the rotation of the wrist was below the elbow inflexion point. On the other hand, the stability of supination was highest when the axis of rotation of the movement was above the forearm's long axis (Carson *et al.*, 2000). This finding demonstrates that an individual's ability to synchronise with an external stimulus is affected by the user's physical positioning. An individual's proprioception however may reduce this impact. In our investigation the user's wrist rotation is not intended to be restricted and the user will be allowed to synchronise regardless of the forearm's position in relation to their body. Users most likely will synchronise with the visual display in a way in which they are most comfortable and efficient to do so. This is because people's central nervous systems have evolved to generate motor patterns that conserve as much energy as possible, when intending to achieve their desired action (Sousa, Silva and Tavares, 2012).

Humans' abilities to synchronise with rhythmic external motion is founded in our proprioception capabilities. As like the literature before, TraceMatch, MatchPoint and 'Do, That There' this project will therefore make use of these natural abilities when requiring the user to synchronise with the system through rhythmic mimicry.

## 2.3 Object Detection

This project intends to build upon MatchPoint, and its limitation associated with the object detection used within it. It meets its required functionality; however, it is generic as it does not make use of specific functionality that comes from hand gestures. A different pipeline is required to detect more advanced movements made by an individual's hands, both for rotation and grasping motions.

Advancements in modern computer power and speed has led to an explosion in object detection algorithm sophistication in recent decades. This includes in real time detection, as seen in Figure 3 (Schroder and Ritter, 2017), with a fully implemented convolutional neural network algorithm being able to accurately distinguish between unknown objects and the user's hands regardless of shape, size, and colour, even when the object was not included in the training dataset.

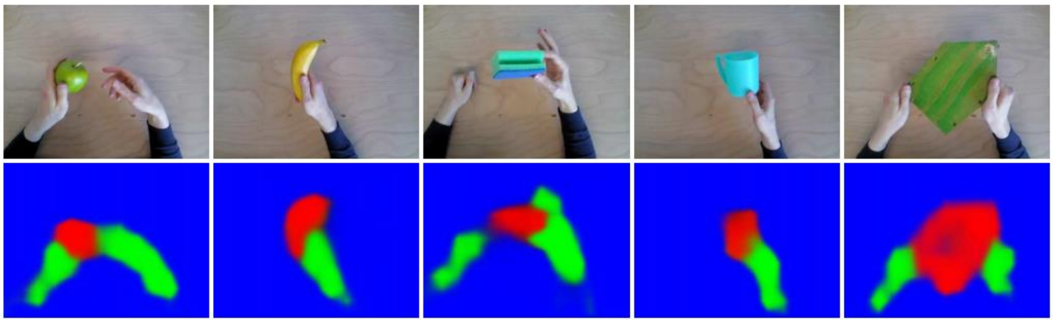


Figure 3. Examples of hand-object discrimination with unknown objects. First row: input images. Second row: Fully convolution network class probability outputs. (Schroder and Ritter, 2017)

These advancements however are not universally accessible as the financial costs associated with the hardware can be significant, Schroder and Ritter use an RGB-D sensor for detecting objects in real time and a GTX 1070 GPU to train the neural network. This reduces the transferability of such an application. This project attempts to address this issue by only using a webcam, making it deployable in many application domains, including televisions, smartphones, laptops, and tablets. The aim of hand gesture detection to a sufficient standard for this dissertation all attempts to be achieved on a device that has an Intel Core i5-7200U CPU and a RAM of 8GB.

## 2.3.1 Rotational Detection

This paper intends to evaluate the detection of hand rotation in real time, so it is logical to discuss the current literature on rotation detection. There are two common deep neural network methods that can be used to detect objects that have been rotated:

- Using segmentation masks that calculate rotated bounding boxes around objects
- Inferring the rotated bounding box directly

Segmentation masks are calculated using Mask Region Based Convolutional Neural Networks, Mask-RCNN. This is an extension of Faster R-CNN through the addition of a branch that is used for predicting an object mask in tandem with the existing branch for bounding box recognition. The bounding box being the coordinates of a rectangular border that fully contains the object of interest (He *et al.*, 2020). A mask is a binary image that consists of zero and non-zero values. When a mask is applied to another image all pixels which are zero in the mask are set to zero in the output image whilst the remaining pixels in the output image remain the same (*HIPR*, 2004). Both these deep neural network methods are two-stage detectors that reduces regions into more granular segment detection. These methods can lead to accurate results for objects that are axis-orientated, this is because the assumption is that minimum bounding boxes tend to be axis-aligned. However, since these deep neural networks are two-stage procedures, the speed at which images are processed tend to be low. Furthermore, producing a rotated bounding box using object masking with post processing can lead to error prone results (Howe and Skinner, 2020).

Inferring the rotated bounding box directly does not suffer from these issues as no post processing is required and because it is a single-stage detector. Single-stage detectors tend to rely on comparing ground truth bounding boxes, hand labelled bounding boxes that specify where the object is in the image, to anchor boxes, which represents the best location, size, and shape of the object it is best tailored to predict.

If the value of the intersection over union (IoU), calculated by dividing the area of overlap over the area of union between the anchor box and the ground truth, is over 0.5 during training then the parameters - the minimum x and y coordinates, the width and height - that define the anchor box are regressed so that the difference between both the anchor box and the ground truth is minimised.

When dealing with rotated boxes an additional parameter must be defined, the angle theta, this increases the number of anchor boxes by a multiple that is equivalent to the number of angles specified. (Howe and Skinner, 2020) graphically demonstrate the impact of the angle being included as an additional parameter. Axis-aligned bounding anchor boxes are represented in blue in Figure 4, with three scales and three aspect ratios for a single location in an image feature space. In red are the rotated anchor boxes for three rotation angles, -π/6, 0 and π/6 radians, using the same scales and aspect ratios. This added layer of complexity prevents the IoU from being calculated in the same manner as when the object is axis-aligned.

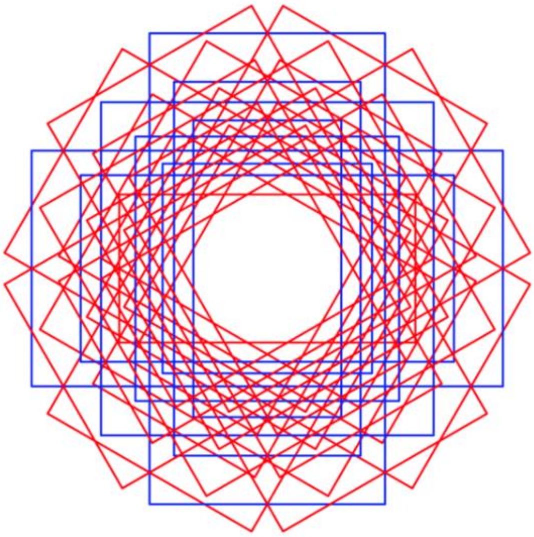


Figure 4. Axis-aligned anchor boxes (blue); rotated anchor boxes (red and blue) for rotation angles -π/6, 0 and π/6 radians. (Howe and Skinner, 2020)

To calculate the IoU for a rotated object a new concave polygon is constructed from the overlapping anchor and ground truth. In Figure 5 the new polygon is denoted using both red vertices, the edges of where the two boxes overlap and green vertices which are within the boxes that are being compared. These points must be calculated for all anchor and ground truth comparisons.

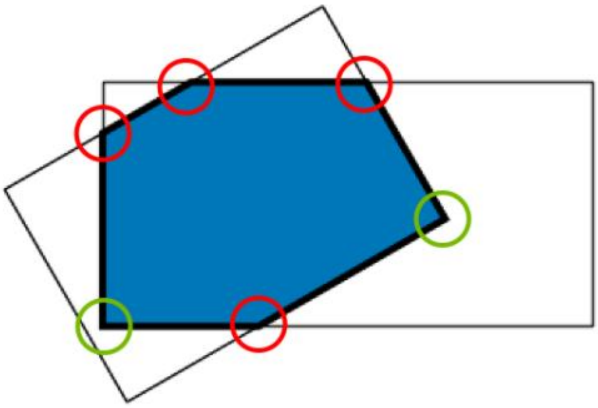


Figure 5. An example of a polygon constructed (solid blue) when overlapping two rotated rectangles. (Howe and Skinner, 2020)

This problem can be solved using (de Berg *et al.*, 2008) ‘Mapoverlay’ algorithm set out in their book Computational Geometry: Algorithms and Applications. This algorithm works by sequentially and recursively reducing the size of an initially defined polygon by using one of the boxes that is being used in the comparison. This algorithm works by determining whether there are any intersections with the edges of the other box. If this is the case, then these intersections are defined to be the vertices and new edges are declared. This result is then again compared to the comparison box until there are no remaining edges. Only if a polygon with more than two edges remains then the IoU can be calculated, otherwise there is no intersection and the IoU is zero.

The IoU in this case is the defined to be the area of the polygon found through the sequential cutting method and dividing it by the area of the unions between the anchor and the ground truth. Minimising the difference of the x coordinate, y coordinate, width, height, sine, and cosine of theta calculates the absolute angle of the object orientation. This projects theta onto a unit circle (Howe and Skinner, 2020).

A more recent and novel way of estimating the rotation of an image without the implementation of bounding boxes was proposed by (Zhou *et al.*, 2019). This method makes use of spatial transformations of convolutional kernels, kernel-mappings, within convolutional neural networks. The convolution kernel operation can transfer spatial characteristics of a specific position, in this case rotation, of a feature to the next layer in the network.

They propose that the spatial transformation, in this case the angular rotation, can be considered a transformation function to the original image. This function is expressed as M, where $(x_r, y_r)$ and $(x, y)$ represent the coordinates before and after the transformation respectively:

$$[x_r, y_r]^T = M[x, y]^T \tag{2}$$

The transformed input after rotation can be expressed as having transformed functions applied to both its x and y coordinate values, $x_r = g(x, y)$ and $y_r = h(x, y)$, as the application of the kernel remains constant, going from left to right, before and after the transformation as shown in Figure 6. In this figure they show the dot product feature output mapping before and after a 90-degree counter-clockwise rotation. They then go on to discuss the mathematical formulation of a 45-degree rotation.

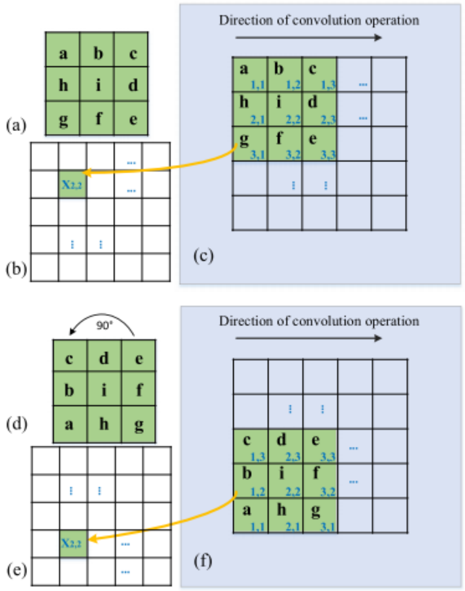

(a)
(b)
(c)
(d)
(e)
(f)
Direction of convolution operation
Direction of convolution operation
90°
X2,2
X2,2

Figure 6. An example of showing the matching property for the convolution operation at the 90◦ rotation. (a) is a 3 × 3 square convolutional kernel and the values at each grid point (a, b, . . . ,i) are the parameters for the convolutional kernels. (c) is the input image, ‘‘1, 1”, ‘‘1, 2”, . . ., ‘‘3, 3” are the values at each pixel. The convolutional kernel slides from the left to the right of the image during the convolution operation and outputs the feature map as shown in (b), in which X2,2 is a value of the outputs. (d), (f) and (e) are the corresponding pictures with a counter clockwise 90◦ rotation. (Zhou *et al.*, 2019)

The drawback of such rotation detection is that the technical system would only be able to register rotations in multiples of 45°, this would limit the user 8 options for each coupled device. This is assuming that their range of motion is not limited. This is unlikely to be the case, as due to skeletal factors the human hand cannot rotate 360-degrees.

To capitalise on the simple and intuitive advantages of single-shot algorithms for object detection Google’s MediaPipe will be used in this project, which can accurately detect hand features in both static images and in videos.

### 2.3.3 MediaPipe

MediaPipe is a cross-platform Google library that allows for ‘customizable ML solutions for live and streaming media’ (*MediaPipe*, no date). One of their solutions, MediaPipe Hands can detect and track hands and fingers in real time.

MediaPipe Hands works through multiple models operating together in tandem, firstly a palm detection model is applied to the entire image and returns a hand bounding box that is orientated in respects to the position of the palm. By only detecting the palm this drastically reduces the need for data augmentation in the second model in this pipeline of hand detection. This is because it allows the network to devote more of its capacity to predicting more accurately the coordinates of the key features on the hand. Detecting palms is easier for two reasons, firstly palms are smaller than hands and secondly require square bounding boxes. Smaller objects are less computationally demanding to detect as they tend to have fewer key features and by using square bounding boxes other aspect ratios can be ignored, in other terms rectangular bounding boxes, this reduces the number of anchors by a factor of 3 to 5. (*MediaPipe*, no date) MediaPipe Hands capitalises the use of a single-shot detector as discussed in 2.3.1, so allows for the fast processing of frames whilst accurately detecting the location of palms.

The second model detects hand landmarks, the key features of the hands, and operates on the cropped image region produced by the first palm model detector. It finds the coordinates of these landmarks by running regressions to give precise localizations of 21 different points. The model is trained on thirty thousand real-world images that were manually labelled. For additional supervision, to make the model robust, it was trained with various backgrounds to account for the noise different backgrounds can generate.

Furthermore, their MediaPipe solution allows for the detection of multiple hands at the same time which could allow for further functionality or for multiple users to duplicate the same function. Nicholas Renotte (Renotte, 2021) gives an example solution of how the angle can be calculated for each finger that is being detected using their respective joints. This solution can be adapted so that when multiple angles of fingers change in unison then this can be referred to be a grasping motion. This can also be expanded further to measure rotation and circular motion.

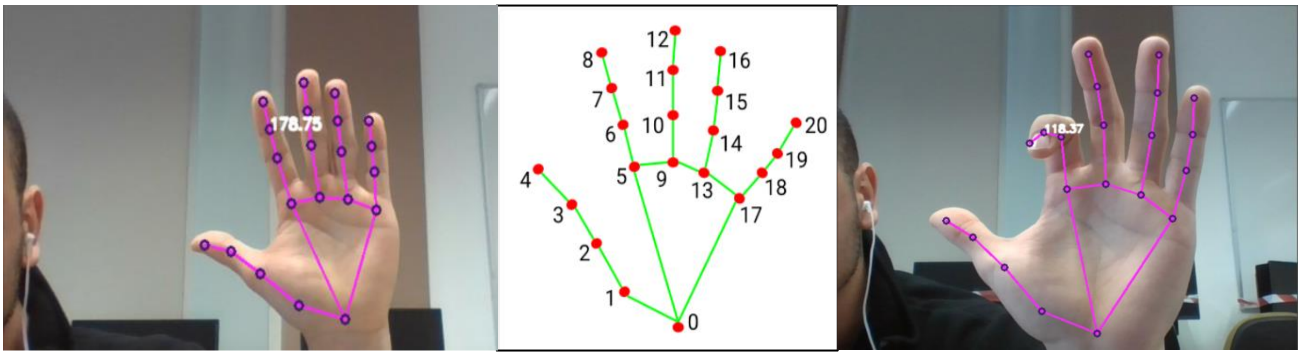


Figure 7. Implementation of renotte’s angle calculation code within python, it works by measuring the angle between two joint points using a third point as the midpoint. take for example my index finger, the joint features are 6, 7 and 8 in the figure on in the middle, the angle been 6 and 8 is then calculated using 7 as the midpoint.

# Chapter 3

# Methodology

## 3.1 Study Design

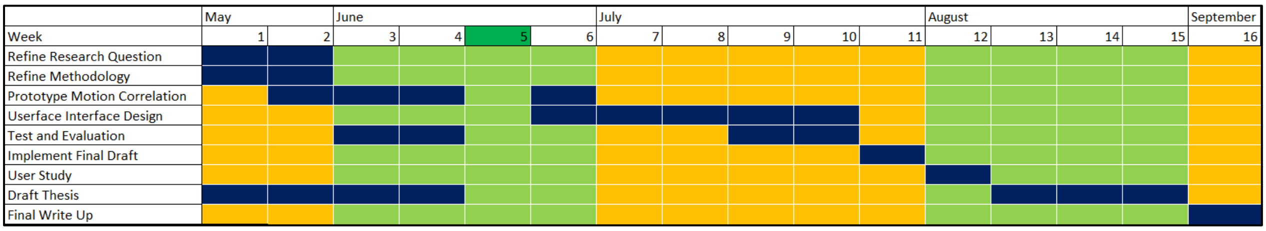
The study will take place on a one-to-one basis, most likely over a video call due to the current ongoing pandemic. The participant will be in front of a laptop and will converse with me, the researcher, for 10 minutes whilst the system is running. This is to encourage false positives to occur, the system will not record any video footage and will only detect motion.

After this, the participant will be asked to mimic visual displays with their hands, this will include moving their arm in a circular motion, rotating their hand, and doing a grasping motion. This will be done at multiple different speeds and sizes of motion. If the participant needs to take a break or feel any form of discomfort, then they can inform the researcher at any point.

The participants will then give feedback on the system and their ability to synchronise with these visual displays will be assessed.

## 3.2 Timeline

The timeline for this project has overlapping and different time intervals to maximise efficiency as some tasks can be done in tandem.



| | May | | June | | | | July | | | | | August | | | | September |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Week | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Refine Research Question | ■ | ■ | | | | | | | | | | | | | | |
| Refine Methodology | ■ | ■ | | | | | | | | | | | | | | |
| Prototype Motion Correlation | | ■ | ■ | ■ | | ■ | | | | | | | | | | |
| Userface Interface Design | | | | | | ■ | ■ | ■ | ■ | ■ | | | | | | |
| Test and Evaluation | | | ■ | ■ | | | | | ■ | ■ | | | | | | |
| Implement Final Draft | | | | | | | | | | | ■ | | | | | |
| User Study | | | | | | | | | | | | ■ | | | | |
| Draft Thesis | ■ | ■ | ■ | ■ | | | | | | | | | ■ | ■ | ■ | |
| Final Write Up | | | | | | | | | | | | | | | | ■ |

# Chapter 4

# Conclusion and Future Work

This research project proposal has demonstrated the motivation of this area, the literature surrounding it and the user study that will take place. It clearly demonstrates that using all the functionality of users' hands will be of great benefit to the field. The rest of the project will include the user study and its implications for future research and its practical use.

# Bibliography

Avrahami, D., Wobbrock, J. O. and Izadi, S. (2011) “Portico: Tangible interaction on and around a tablet,” in *UIST’11 - Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. doi: 10.1145/2047196.2047241.

de Berg, M. *et al.* (2008) *Computational Geometry Third Edition*. doi: 10.1007/978-3-540-77974-2.
*BOND - Internet Trends 2019* (2019). Available at: https://www.bondcap.com/report/itr19/ (Accessed: April 25, 2021).

Buekers, M. J. *et al.* (2000) “The synchronization of human arm movements to external events,” *Neuroscience Letters,* 290(3), pp. 181–184. doi: 10.1016/S0304-3940(00)01350-1.

Carson, R. G. *et al.* (2000) “Neuromuscular-skeletal constraints upon the dynamics of unimanual and bimanual coordination,” *Experimental Brain Research*, 131(2), pp. 196–214. doi: 10.1007/s002219900272.

Clarke, C. *et al.* (2016) “TraceMatch: a Computer Vision Technique for User Input by Tracing of Animated Controls.” doi: 10.1145/2971648.2971714.

Clarke, C. (2020) *Dynamic motion coupling of body movement for input control - Research Portal | Lancaster University*. Available at: http://www.research.lancs.ac.uk/portal/en/publications/dynamic-motion-coupling-of-body-movement-for-input-control(ece40c35-cb20-4dfc-9385-343f208cae56).html (Accessed: May 16, 2021).

Clarke, C. and Gellersen, H. (2017) “MatchPoint: Spontaneous spatial coupling of body movement for touchless pointing,” in *UIST 2017 - Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. doi: 10.1145/3126594.3126626.

Corsten, C. *et al.* (2013) “Instant user interfaces: Repurposing everyday objects as input devices,” in *ITS 2013 - Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*. doi: 10.1145/2512349.2512799.

Esteves, A. *et al.* (2015) “Orbits: Gaze interaction for smart watches using smooth pursuit eye movements,” in *UIST 2015 - Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology*. doi: 10.1145/2807442.2807499.

Freeman, E., Brewster, S. and Lantz, V. (2016) “Do that, there: An interaction technique for addressing in-air gesture systems,” in *Conference on Human Factors in Computing Systems - Proceedings*. New York, NY, USA: Association for Computing Machinery, pp. 2319–2331. doi: 10.1145/2858036.2858308.

Gielen, C. C. A. M., van den Heuvel, P. J. M. and van Gisbergen, J. A. M. (1984) “Coordination of fast eye and arm movements in a tracking task,” *Experimental Brain Research,* 56(1), pp. 154–161. doi: 10.1007/BF00237452.

Greenberg, S. and Boyle, M. (2002) *Customizable Physical Interfaces for Interacting with Conventional Applications.*

He, K. *et al.* (2020) “Mask R-CNN,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2). doi: 10.1109/TPAMI.2018.2844175.

Held, R. *et al.* (2012) “3D puppetry,” in. doi: 10.1145/2380116.2380170.

*HIPR* (2004). Available at: https://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr_top.htm (Accessed: April 25, 2021).

Hove, M. J., Spivey, M. J. and Krumhansl, C. L. (2010) “Compatibility of Motion Facilitates Visuomotor Synchronization,” *Journal of Experimental Psychology: Human Perception and Performance*, 36(6), pp. 1525–1534. doi: 10.1037/a0019059.

Howe, J. and Skinner, J. (2020) *Detecting Rotated Objects Using the NVIDIA Object Detection Toolkit | NVIDIA Developer Blog*. Available at: https://developer.nvidia.com/blog/detecting-rotated-objects-using-the-odtk/ (Accessed: April 25, 2021).

Kendon, A. (2004) *Gesture: Visible Action as Utterance*. Available at: https://books.google.co.uk/books?hl=en&lr=&id=hDXnnzmDkOkC&oi=fnd&pg=PR6&dq=adam+kendon+gesture&ots=RLZWyeZViF&sig=8o9HUP6JFtJedi6t7F25lgpaM9A#v=onepage&q=adam%20kendon%20gesture&f=false (Accessed: May 1, 2021).

Krauzlis, R. J. (2005) “The control of voluntary eye movements: New perspectives,” *The Neuroscientist*, 11(2), pp. 124–137. doi: 10.1177/1073858404271196.

Mackenzie, I. S., Kauppinen, T. and Silfverberg, M. (2001) “Accuracy Measures for Evaluating Computer Pointing Devices,” in *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI ’01*. New York, New York, USA: ACM Press.

*MediaPipe* (no date). Available at: https://mediapipe.dev/index.html (Accessed: April 28, 2021).

Neggers, S. F. W. and Bekkering, H. (2001) *Gaze Anchoring to a Pointing Target Is Present During the Entire Pointing Movement and Is Driven by a Non-Visual Signal*. Available at: www.jn.org (Accessed: April 28, 2021).

Orban De Xivry, J. J. and Lefèvre, P. (2007) “Saccades and pursuit: Two outcomes of a single sensorimotor process,” *Journal of Physiology*, 584(1), pp. 11–23. doi: 10.1113/jphysiol.2007.139881.

Pelisson, D. *et al.* (1986) *Visual control of reaching movements without vision of the limb II. Evidence of fast unconscious processes correcting the trajectory of the hand to the final position of a double-step stimulus*, *Exp Brain Res*.

Prablanc, C. *et al.* (1979) “Optimal response of eye and hand motor systems in pointing at a visual target - I. Spatio-temporal characteristics of eye and hand movements and their relationships when varying the amount of visual information,” *Biological Cybernetics*, 35(2), pp. 113–124. doi: 10.1007/BF00337436.

Renotte, N. (2021) *nicknochnack/AdvancedHandPoseWithMediaPipe*. Available at: https://github.com/nicknochnack/AdvancedHandPoseWithMediaPipe (Accessed: April 28, 2021).

Repp, B. H. (2005) “Sensorimotor synchronization: A review of the tapping literature,” *Psychonomic Bulletin and Review*. Springer New York LLC, pp. 969–992. doi: 10.3758/BF03206433.

Rosten, E. and Drummond, T. (2006) “Machine learning for high-speed corner detection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, pp. 430–443. doi: 10.1007/11744023_34.

Schroder, M. and Ritter, H. (2017) “Hand-Object Interaction Detection with Fully Convolutional Networks,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*

*Workshops*. doi: 10.1109/CVPRW.2017.163.

Sousa, A. S. P., Silva, A. and Tavares, J. M. R. S. (2012) "Biomechanical and neurophysiological mechanisms related to postural control and efficiency of movement: A review," *Somatosensory & Motor Research*, 29(4), pp. 131–143. doi: 10.3109/08990220.2012.725680.

Velloso, E. *et al.* (2017) "Motion Correlation: Selecting Objects by Matching their Movement," *ACM Trans. Comput.-Hum. Interact*, 24(3). doi: 10.1145/3064937.

Vidal, M., Bulling, A. and Gellersen, H. (2013) "Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets," in *UbiComp 2013 - Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. doi: 10.1145/2493432.2493477.

Zhou, Y. *et al.* (2019) "Rotational objects recognition and angle estimation via kernel-mapping cnn," *IEEE Access*, 7. doi: 10.1109/ACCESS.2019.2933673.