UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

Analysis of Applying Dynamic Time Warping, Multi-Dimensional Scaling, and DBSCAN Methods to Cluster Scanpaths Data in Application of Air Traffic Management

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

In partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

JUNEHYUNG LEE
Norman, Oklahoma
2022

Analysis of Applying Dynamic Time Warping, Multi-Dimensional Scaling, and DBSCAN Methods to Cluster Scanpaths Data in Application of Air Traffic Management

A THESIS APPROVED FOR THE

SCHOOL OF INDUSTRIAL AND SYSTEME ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Ziho Kang, Chair

Dr. Talayeh Razzaghi

Dr. Rui Zhu

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Federal Aviation Administration (FAA) is interested in efficient ways to train air traffic controllers (ATCs) due to the high cost and long training time. It is known that exposing trainees to the visual scanpaths of experts in en route air traffic control environment can be an efficient way to improve trainees' performance in a relatively short amount of time (Kang and Landry, 2014). In detail, Kang and Landry (2014) investigated whether novices can learn from observing experts' scanpaths (i.e. time ordered sequence of eye fixations connected by fast eye movement, which is defined as saccade). The task was to identify aircraft pairs on a radar display that might have a conflict (i.e. loss of separation (LOS)) in near future. The participants in the treatment group who were exposed to the experts' visual scanpaths had a smaller number of false alarms and miss rates compared to those of the participants in the controlled group.

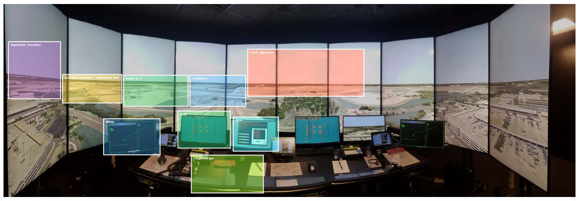


Figure 1. High-fidelity virtual environment with AOIs

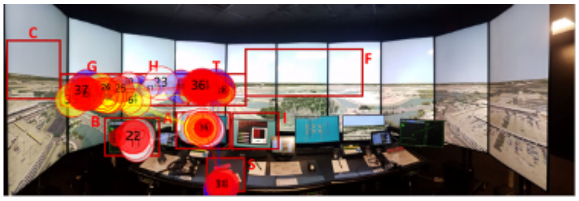


Figure 2. Example of experts' visual scanpath
(used from FAA project A18-0232)

It is unknown whether the experts' visual scanning behaviors can be leveraged to better train the novices in the tower environment. To do so, first there should be a way to effectively analyze experts' visual scanpaths. According to the research conducted by Kang and Landry (2015), scanpaths can be converted into two main formats for analysis: 1) visualizing the scanpaths on interrogated display with Areas of Interest (AOIs) (or eye fixations), or 2) enumerating the scanpaths into sequences of strings that represent AOIs or eye fixations. Here, Figure 1 has an example of the virtual reality tower environment in which ATCs practice. The colored boxes in Figure 1 indicate AOIs where the controllers are required to pay attention during the air traffic management task. To analyze two complex examples of experts' visual scanpaths, they can be drawn as they are on Figure 1 or enumerated into sequences of AOI letters.

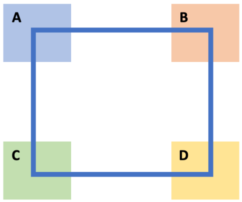


Figure 3. Simplest example 1

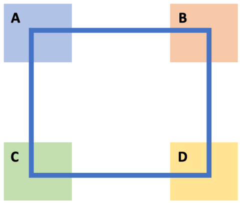


Figure 4. Simplest example 2

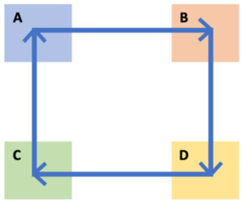


Figure 5. Simplest example 3

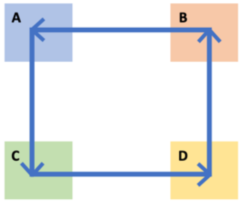


Figure 6. Simplest example 4

Another challenging issue of analyzing visual scanpaths is the time ordered aspects. Consider the simplest case with the two rectangles consisting of saccade, they might look as the same without the time ordered aspect (Figure 5, and 6). However, as shown in Figure 7, and 8, when the two simple scanpath examples 1 and 2 are characterized into scanpath sequence format, they are not identical anymore – scanpath sequence of Example 1: ABDCA, Example 2: ACDBA. Therefore, a dissimilarity measurement among scanpaths should be defined to compare and further cluster them. Kang et al. (2020) used Dynamic Time Warping (DTW) algorithm to calculate dissimilarities between visual scanpaths. Also, the researchers were able to cluster the visual scanpaths based on Density Based Spatial Cluster Application with Noise (DBSCAN) methods in offshore drilling field. Therefore, in this research, it was investigated whether the approach introduced by Kang et al. (2018) was applicable in application of air traffic management in tower environment.

1.1 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) algorithm was first introduced and developed in the 1960s and it started to heavily be used during the 1980s for applying the algorithm into speech recognition field (Myers, Rabiner, and Rosenberg, 1980; Sakeo, and Chiba, 1978). Not only that, DTW is still widely used in a variety of fields such as handwriting recognition (Efrat, Fan, and Venkatasubramanian, 2007; Tappert, Suen, and Wakahara, 1990), and sign language & gesture recognition (Corradini, 2001), and signal processing (Muller, M., Mattes, H., and Kurth, F., n.d.), and even for time series clustering (i.e. time series databases search) (Niennattrakul, and Ratanamahatana, 2007) etc.

The reason behind this algorithm being so versatile and easy to use in a variety of fields are mainly because of two aspects: First is that it can be easily applied if there are just two time series data. Nowadays a lot of data are associated with time ordered components – even if the data do not include the time ordered aspect, most of the data can be easily transformed into time series format by applying simple data processing and data cleansing phases.

Since the goal of DTW algorithm is to find the optimal warping path between two time series data which returns the shortest distance from a time series to another, DTW employs dynamic programming to attack the optimization problem. However, the distance (or dissimilarity) between two time series has a lot of flexibility in terms of its definition. For instance, Euclidean distance can be used as a distance measure between two time series. If the distance is defined appropriately, DTW can be applied to multi-variate time series as well. This aspect makes DTW algorithm more versatile and powerful.

To apply the DTW algorithm, two time series data are obtained – first data set are usually called as reference time series, while another one as test or query time series.

$X = \{x_1, x_2, \cdots, x_i, \cdots, x_m\}$: Reference Time Series

$Y = \{y_1, y_2, \cdots, y_j, \cdots, y_n\}$: Test or Query Time Series

The DTW algorithm utilizes dynamic programming method, which is searching for optimal values based on a cost function that keeps changing over the computation during the optimization process. Here, in DTW algorithm, the function is defined as the distance (or dissimilarity) between two time series data – it can be defined as simply as absolute value of difference between signal strength or in more complex ways as the researcher decides. The

$$d(i,j) = f(x_i, y_j) \geq 0$$

$$f(x_i, y_j) = |x_i - y_j|$$

or

$$= \sqrt{(x_{i1} - y_{j1})^2 + (x_{i2} - y_{j2})^2}$$ (based on researcher's decision)

DTW cost matrix can be defined as a cumulative distance matrix that is for calculating DTW warp and distance. Each element of DTW cost matrix is defined as the summation of the distance between the sequences and the minimum value among three possible pathways

$$\gamma(i,j) = d(i,j) + \min\{\gamma(i-1,j), \gamma(i,j-1), \gamma(i-1,j-1)\}$$

Once the distance or cost function has calculated, the warping path is built by DTW to evaluate the optimality during the optimization process. Warping path, or often called warping curve, can be defined as below. The goal of DTW algorithm is to minimize the distance of selected warping path.

$w_K = (i_K, j_K)$, where i_K: time index of series X selected at Kth warp

j_K: time index of series Y selected at Kth warp

$$DTW(X, Y) = \min \sum_{k=1}^{K} d(w_k)$$

For the last part of the dynamic programming, during every step of optimization process, the cost function or the distance at warping path is calculated and then guided to have the optimal value by constraints of the optimization problem.

1. Boundary constraint

   $w_1 = (1, 1)$, and $w_K = (m, n)$

2. Monotonicity constraint

   $i_{k-1} \leq i_k, j_{k-1} \leq j_k$

3. Step size constraint

   $i_k - i_{k-1} \leq 1, j_k - j_{k-1} \leq 1$

Once all the process of DTW algorithm is completed, the outputs of the algorithm would be DTW distance, the warping path between the sequences. If the algorithm was applied to n number of sequences, the algorithm would return ${}_nC_2$ DTW distances and the warping paths. In such case, the output of DTW algorithm can be formatted into a $n \times n$ symmetric matrix, which is called DTW distance matrix (which is different from DTW cost matrix). However, the distance matrix only provides a dissimilarity between sequences (i.e. visual scanpaths) while the sequences needs to be allocated on a certain dimensional space (e.g. 2-dimensional space with xy coordinates) to be visualized and clustered. Therefore, Multi-Dimensional Scaling method can be utilized to compute the projection of n sequence data points into lower dimensional space.

1.2 Multi-Dimensional Scaling (MDS)

Multi-Dimensional Scaling (MDS) method has been developed by the mathematical psychologists Carroll and Arabie (1983). Although its first application was in behavioral science, the method has become popular and widely used in a variety of areas in application. The basic understanding of MDS is that utilizing the proximity data of n objects to estimate a configuration of points in a certain space (e.g. 2-dimensional xy space). Here, each point indicates one of the n objects or individual from data set, and the distances between pairs of objects are matched as possible as with the proximity data between n objects during the computation.

In MDS method, proximity is defined as closeness between objects r and s in any space that is being discussed. Similarity (s_{rs}) and dissimilarity (δ_{rs}) are the two types of proximity, which imply how similar or dissimilar the objects are. Also, proximity in MDS has non-negativity feature. For instance, a similarity between a pair of two identical objects would be the maximum value as it could be, while the dissimilarity between the pair would be zero, which is the lowest value both similarity and dissimilarity could have. The most popular dissimilarity measures that are widely used in various fields are listed in Table 1 (Cox and Cox, 2008).

Table 1. Dissimilarity measures for quantitative data

Dissimilarity measure	Formula
Euclidean distance	$\delta_{rs} = \left\{\sum_i (x_{ri} - x_{si})^2\right\}^{\frac{1}{2}}$
Weighted Euclidean distance	$\delta_{rs} = \left\{\sum_i w_i \cdot (x_{ri} - x_{si})^2\right\}^{\frac{1}{2}}$
City block metric	$\delta_{rs} = \lvert x_{ri} - x_{si} \rvert$
Canberra metric	$\delta_{rs} = \frac{\lvert x_{ri} - x_{si} \rvert}{x_{ri} + x_{si}}$
DTW distance metric	$\min \sum_{k=1}^{K} d(w_k)$

With a set of dissimilarities between n objects, the overall steps of MDS algorithm will discover coordinates from the distances between pairs of points of objects as followings.

1. Form matrix A = $[-\frac{1}{2}\delta_{rs}^2]$
2. Form matrix $B = H \cdot A \cdot H$. Here, H indicates the centering matrix where $H = I - n^{-1} \cdot 1_n \cdot 1_n^T$ (1_n: a vector of ones).
3. Find V and Λ that meet the $B = V \cdot \Lambda \cdot V^T$ which are the spectral decomposition of B. Once the eigenvalues of B are identified, Λ is the diagonal matrix that has the eigenvalues of B, and V has the corresponding eigenvectors as a matrix form.
4. Consider whether the points of n obejcts were in p-dimensional space originally, the first p eigenvalues of B are going to be nonzero and the remaining $n - p$ eigenvalues will be zero. Afterwards, discard the zeros from Λ and rename it as Λ_1, and discard the eigenvalues that are comparable to first p eigenvalues from V then rename it as V_1.
5. The last step of the MDS algorithm would be computing the coordinates of the points which are going to be presented as the rows of χ, where $\chi = V_1 \cdot \Lambda_1^{\frac{1}{2}}$.

Note that the solution χ won't be unique due to the aspect that another solution could be generated if any translation, rotation, or reflection of the configuration of the points of n obejcts.

In this study, DTW distance between pairs of visual scanpaths was selected as proximity measure of dissimilarity. Once the DTW distance matrix is computed from n visual scanpaths, the MDS method can be applied to provide the configurations (i.e. coordinates on a MDS plane) of each visual scanpath based on the proximity data (i.e. DTW distance matrix). Subsequently, a clustering method can be applied on the points of visual scanpaths that are laid on a space to identify specific or non-predictable patterns.

1.3 Density-Based Spatial Clustering of Application with Noise (DBSCAN)

Density-Based Spatial Clustering of Application with Noise (DBSCAN) algorithm was first introduced in 1996 by Ester et al. (1996). The algorithm has inspired a lot of density-based clustering methods and is still one of the most famous density-based clustering algorithms. The widely known advantages of DBSCAN are 1) no need for predetermining the number of clusters before the analysis, and 2) highly efficient in finding clusters in unpredictable forms and spotting outliers (Ester et al., 1996; Ma and Angryk, 2017). It is also known to be practical in a lot of real-life applications (Bäcklund et al., 2011). For instance, DBSCAN is widely used in discovering anomaly patterns in specific data sets (e.g. credit fraud, health condition, etc.). The clustering method is also known for measuring inconsistency in temperature patterns due to the environmental changes (i.e. global warming) (Çelic et al., 2011).

The basic approach of most density-based clustering methods is first dividing the data into separate spaces based on how dense the data points are (i.e. high density area vs. low density area). Once the spaces are laid into multiple subspaces, the groups of data points in high density area are categorized as clusters, while the data points in low density area are marked as noise or outliers from the overall dataset. DBSCAN algorithm is also starting from defining the density of a data point in dataset first. The two parameters of DBSCAN algorithm – the radius, and the number of minimum points for a cluster—are devised for specifying a "density" for a data point p from another data point q in the dataset, D.

Eps (ϵ): The radius utilized for determining whether a data point in D is in the same neighborhood with another data point from D. $N_\epsilon(p)$ (i.e. ϵ-neighborhood of data point p) indicates all data points that are located within ϵ from the data point p. Therefore, it can be demonstrated as below for a data point q in the dataset D.

$$N_\epsilon(p) \coloneqq \{q \in D \mid d(p,q) \leq \epsilon\}$$

where $d(p,q)$ is defined as the distance between data points p and q. For this study, DTW distance is used as a distance between data points, since DTW is computed as a dissimilarity metric for visual scanpaths.

MinPts: The minimum number of neighborhood data points that are required to grouped into a cluster. For instance, if the $N_{\epsilon}(p)$ has the greater number of data points than MinPts, density of p is considered as high and dense enough to form a cluster. Otherwise, the density of p is evaluated as low. Based on the level of density, each point in D can be categorized into three types.

- Core point: a data point in D that has high density for its ϵ-neighborhood. It is located inside the ϵ-neighborhood.
- Border point: a data point in D that has low density but still located in the ϵ-neighborhood of a core point. The position of border points is on the edge of clusters.
- Noise point: any data points in D that are neither classified as core point or border point.

Once each data point is evaluated as one of the three-point types, the algorithm initiates the clustering by expanding from a core point to another—Density-reachable point—within the dataset.

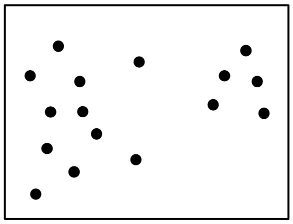


Figure 7. Data points in DBSCAN

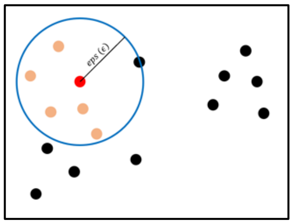


Figure 8. Core point in DBSCAN

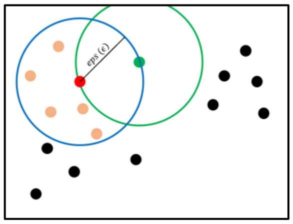


Figure 9. Border point in DBSCAN

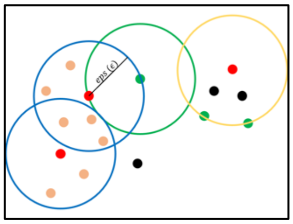


Figure 10. Density-reachable points in DBSCAN

Lastly, to evaluate the DBSCAN method, Silhouette score or coefficient is introduced, which is a metric that indicates how the clusters are separated from one to another. Silhouette score is ranging from -1 to 1, and it can be interpreted as clusters are not properly grouped if the value is -1, clusters are well separated and distinguishable if the value is 1. If the Silhouette score value is near 0, that means clusters are grouped but the distinction between the clusters is not significant. It can be calculated as $\frac{b-a}{\max(a,b)}$, where a indicates the average intra-cluster distance, and b indicates the average intra-cluster distance of the clusters.

Chapter 2. Case Study

2.1 Univariate DTW computed on x coordinates of Clockwise and Counterclockwise

Before actual ATCs' visual scanpath data were analyzed, DTW, MDS, and DBSCAN was applied to simple scanpath examples to see how the algorithms work. According to the research conducted by Kang, Bass, and Lee (2014), air traffic controllers use different scan patterns to detect possible conflict between aircraft. One of the most common scan patterns used by ATCs is circular scan pattern, which also can be further categorized based on the direction – clockwise and counterclockwise. Here, clockwise, and counterclockwise scan patterns are virtually created as most simple case for the demonstration of DTW algorithm. Both of the scanpaths are recorded on a square shaped display during 4-unit time periods using eye movement tracking device. Both scanpaths' first eye fixation has occurred at the bottom left corner of the display, and starting from there, clockwise, and counterclockwise scanpath has been recorded.

Step 1. Defining the sample scanpath

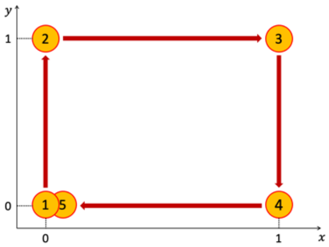


Figure 11. Clockwise scanpath example

Table 2. Raw data extracted for clockwise scanpath example

Time (seconds)	Fixation coordinates x	Fixation coordinates y	Fixation duration (second)
1	0	0	1
2	0	1	1
3	1	1	1
4	1	0	1
5	0	0	1

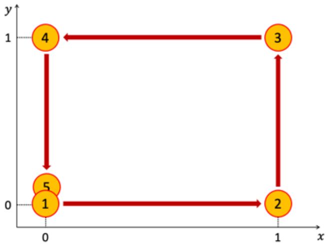


Figure 12. Counterclockwise scanpath example

Table 3. Raw data extracted for counterclockwise scanpath example

Time (seconds)	Fixation coordinates x	Fixation coordinates y	Fixation duration (second)
1	0	0	1
2	1	0	1
3	1	1	1
4	0	1	1

Step 2. Preprocessing the sequences before applying the algorithm

Before the DTW algorithm was applied to the two example scanpath sequences, the example scanpaths can be separated to x and y coordinates, since xy plane with an additional time axis might not be a best option in terms of visualization. Therefore, DTW algorithm can be applied first between x coordinates of clockwise and counterclockwise, then y coordinates of clockwise and counterclockwise. For last, actual DTW distance can be calculated by applying the multivariate DTW algorithm to both of xy coordinates of the sample sequences at the same time.

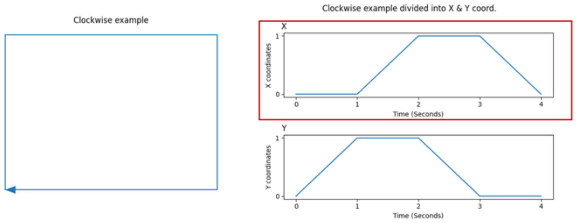


Figure 13. Clockwise example separated to x and y coordinates

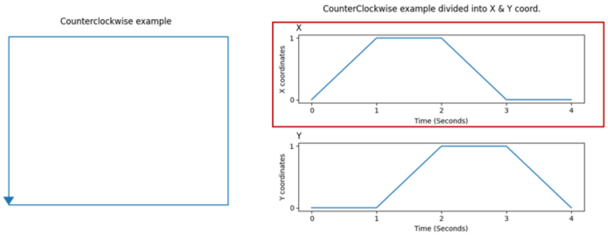


Figure 14. Counterclockwise example separated to x and y coordinates

Step 3. Calculating DTW Cost Matrix

DTW cost matrix can be calculated based on the following equation when we have a reference sequence, length of P, and a query sequence, length of Q. For univariate DTW algorithm, the distance between two sequences at time = (i, j) can be defined as an absolute value of the subtraction between the reference and query values $|A_i - B_j|$. The DTW cost matrix would look like Figure.

$$\gamma(i,j) = d(i,j) + \min\{\gamma(i-1,j), \gamma(i,j-1), \gamma(i-1,j-1)$$

Table 4. Query and reference sequence length of P and Q respectively

Index	Query value
1	A_1
2	A_2
⋮	⋮
i	A_i
⋮	⋮
P	A_P

Index	Reference value
1	B_1
2	B_2
⋮	⋮
j	B_j
⋮	⋮
Q	B_Q

Reference sequence	$j = Q$	$\gamma(1,Q)$	$\gamma(2,Q)$	$\gamma(3,Q)$	⋯	$\gamma(P,Q)$
	⋮	⋮	⋮	⋮	⋰	⋮
	$j = 3$	$\gamma(1,3)$	$\gamma(2,3)$	$\gamma(3,3)$	⋯	$\gamma(P,3)$
	$j = 2$	$\gamma(1,2)$	$\gamma(2,2)$	$\gamma(3,2)$	⋯	$\gamma(P,2)$
	$j = 1$	$\gamma(1,1)$	$\gamma(2,1)$	$\gamma(3,1)$	⋯	$\gamma(P,1)$
		$i = 1$	$i = 2$	$i = 3$	⋯	$i = P$

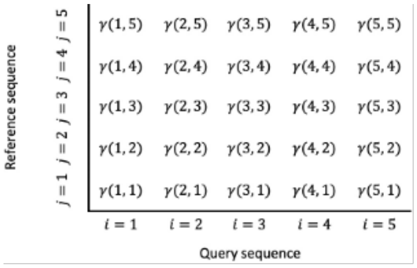
Query sequence

Figure 15. DTW cost matrix between the query and the reference sequence

Since the DTW distance between x coordinates of clockwise and counterclockwise are going to be calculated, the raw data of each simple visual scanpath can be formatted into Table 2.1-4. The DTW cost matrix between two circular scanning pattern scanpaths would be described as shown in Figure 2.1-6.

Table 5. Clockwise and counterclockwise scanpaths turned into the query and reference sequence

Index	Query value	Index	Reference value
1	0	1	0
2	1	2	0
3	1	3	1
4	0	4	1
5	0	5	0



$j = 5$	$\gamma(1,5)$	$\gamma(2,5)$	$\gamma(3,5)$	$\gamma(4,5)$	$\gamma(5,5)$
$j = 4$	$\gamma(1,4)$	$\gamma(2,4)$	$\gamma(3,4)$	$\gamma(4,4)$	$\gamma(5,4)$
$j = 3$	$\gamma(1,3)$	$\gamma(2,3)$	$\gamma(3,3)$	$\gamma(4,3)$	$\gamma(5,3)$
$j = 2$	$\gamma(1,2)$	$\gamma(2,2)$	$\gamma(3,2)$	$\gamma(4,2)$	$\gamma(5,2)$
$j = 1$	$\gamma(1,1)$	$\gamma(2,1)$	$\gamma(3,1)$	$\gamma(4,1)$	$\gamma(5,1)$
	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$

Figure 16. DTW cost matrix between the clockwise and counterclockwise scanpaths

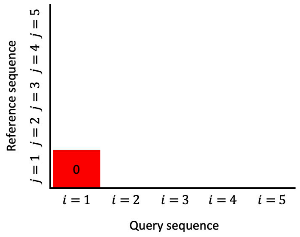
Step 3-1. $i = 1$ *(i.e. x coordinate of clockwise at time 1),* $j = 1$ *(i.e. x coordinate of counterclockwise at time 1)*

According to the equation provided in Step 3, first element of the DTW cost matrix between x coordinates of clockwise and counterclockwise can be determined as: $\gamma(1,1) = d(1,1) + \min\{\gamma(0,0), \gamma(1,0), \gamma(0,1)\}$, where $\gamma(0,0), \gamma(1,0)$, and $\gamma(0,1)$ are three possible path ways toward $\gamma(1,1)$. However, since the $\gamma(1,1)$ is the very first element of the cost matrix, $\gamma(0,0)$, $\gamma(1,0)$, and $\gamma(0,1)$ do not exist ($\because i \in [1, \cdots, 5], j \in [1, \cdots, 5]$). Therefore, the first element could be identified as: $\gamma(1,1) = d(1,1)$.

$$\therefore \gamma(1,1) = d(1,1) = |A_1 - B_1| = |0 - 0| = 0$$

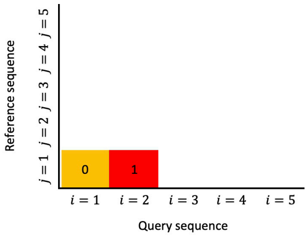
Step 3-2. $i = 2, j = 1$

$$\begin{aligned} \gamma(2,1) &= d(2,1) + min\{\gamma(1,0), \gamma(2,0), \gamma(1,1)\} \\ &= |1 - 0| + \gamma(1,1) \\ &= 1 + 0 \end{aligned}$$



Reference sequence
$j = 1$ $j = 2$ $j = 3$ $j = 4$ $j = 5$
0
$i = 1$ $i = 2$ $i = 3$ $i = 4$ $i = 5$
Query sequence

Figure 17. The element of DTW cost matrix at (i, j)=(1, 1)



Reference sequence
$j = 1$ $j = 2$ $j = 3$ $j = 4$ $j = 5$
0
1
$i = 1$ $i = 2$ $i = 3$ $i = 4$ $i = 5$
Query sequence

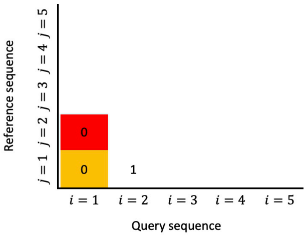
Figure 18. The element of DTW cost matrix at (i, j) = (2, 1)

Step 3-3. $i = 1, j = 2$

$$\begin{aligned} \gamma(1,2) &= d(1,2) + min\{\gamma(0,1), \gamma(0,2), \gamma(1,1)\} \\ &= |0 - 0| + \gamma(1,1) \\ &= 0 + 0 \end{aligned}$$

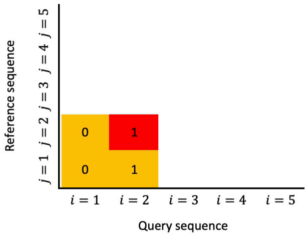
Step 3-4. $i = 2, j = 2$

$$\begin{aligned} \gamma(2,2) &= d(2,2) + min\{\gamma(1,1), \gamma(1,2), \gamma(2,1)\} \\ &= |1 - 0| + \gamma(1,1)\, or\ \gamma(2,1) \\ &= 1 + 0 \end{aligned}$$



Reference sequence
$j = 1$ $j = 2$ $j = 3$ $j = 4$ $j = 5$
0
0
1
$i = 1$ $i = 2$ $i = 3$ $i = 4$ $i = 5$
Query sequence

Figure 19. The element of DTW cost matrix at (i, j)=(1,2)



Reference sequence
$j = 1$ $j = 2$ $j = 3$ $j = 4$ $j = 5$
0
1
0
1
$i = 1$ $i = 2$ $i = 3$ $i = 4$ $i = 5$
Query sequence

Figure 20. The element of DTW cost matrix at (i, j)=(2,2)

By iterating over each element of the DTW cost matrix, the matrix can be calculated as shown in Figure below.

Reference sequence					
$j = 5$	2	1	1	0	0
$j = 4$	2	0	0	1	2
$j = 3$	1	0	0	1	2
$j = 2$	0	1	2	2	2
$j = 1$	0	1	2	2	2
Query sequence	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$

Figure 21. DTW cost matrix between x coordinates of clockwise and counterclockwise

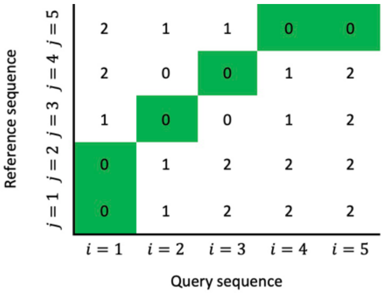
Step 4. Getting the optimal warping path and the DTW distance

Once the DTW cost matrix has been finalized, the warping path between the reference and the query sequence can be identified as the minimization of both length K and $d(w_K)$ until it reaches (P, Q) on the cost matrix, where K is the length of warping path itself. Here, the warping path (w_K, where $k = 1, \cdots, 5$) between the query and reference sequence is selected based on the cost matrix as below. Based on the DTW cost matrix and warping path, the DTW distance between x coordinates of two circular scanpaths is determined as 0. In other words, in terms of DTW there are no difference among two scanpaths' x coordinates.

$w_K = (i_K, j_K)$, where i_K: time index of reference sequence selected at Kth warp

j_K: time index of query sequence selected at Kth warp

$$w_1 = (1, 1)$$
$$w_2 = (1, 2)$$
$$w_3 = (2, 3)$$
$$w_4 = (3, 4)$$
$$w_5 = (4, 5)$$
$$w_6 = (5, 5)$$



Reference sequence					
$j = 5$	2	1	1	0	0
$j = 4$	2	0	0	1	2
$j = 3$	1	0	0	1	2
$j = 2$	0	1	2	2	2
$j = 1$	0	1	2	2	2
Query sequence	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$

Figure 22. Warping path selected from the DTW cost matrix

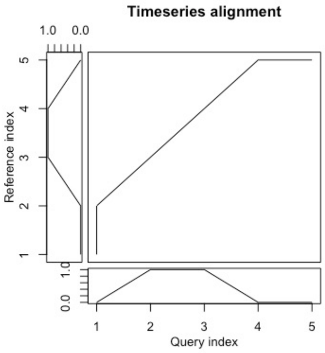


Figure 23. Warping path plot generated using R package

In Figure 24, the warping path alignment plot between the query and the sequence is shown. Here, it can be identified that the reference value at time index 1 is warped to the query value at time index 1 and 2. Also, another warp between two references can be observed at time index 5 of the reference value and index 4, 5 of query values. These behaviors of warping alignment can be also spotted on the warping curve on the DTW cost matrix.

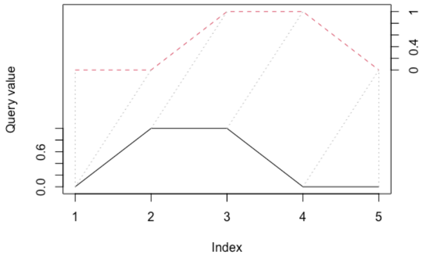


Figure 24. The alignment of the warping path between the query sequence (black) and the reference sequence (red dashed)

2.2 Univariate DTW computed on y coordinates of Clockwise and Counterclockwise

Afterwards, the DTW was applied to the y coordinates of two circular scanpaths since x coordinates do not address the entire characteristics or movements of the scanpaths. Therefore, the query and reference sequences are defined as below for the DTW algorithm (see Table).

Step 1. Defining the sample scanpath

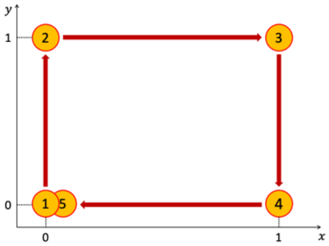


Figure 25. Clockwise scanpath example

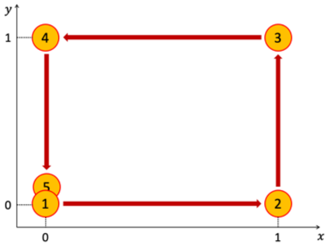


Figure 26. Counterclockwise scanpath example

Table 6. Raw data extracted for clockwise scanpath example

Time (seconds)	Fixation coordinates x	Fixation coordinates y	Fixation duration (second)
1	0	0	1
2	0	1	1
3	1	1	1
4	1	0	1
5	0	0	1

Table 7. Raw data extracted for counterclockwise scanpath example

Time (seconds)	Fixation coordinates x	Fixation coordinates y	Fixation duration (second)
1	0	0	1
2	1	0	1
3	1	1	1
4	0	1	1

Step 2. Preprocessing the sequences before applying the algorithm

Here, as shown in Figure 1 and 2, the y coordinates of two circular scanpaths can be defined. Interestingly, the y coordinates of clockwise scanpath are identical to those of counterclockwise scanpath. In details, the query sequence from Chapter 2.1 is going to be the reference sequence in this chapter, and vice versa. The DTW algorithm was still applied in this chapter to investigate whether there is a difference on the DTW cost matrix, warping path, and the DTW distance.

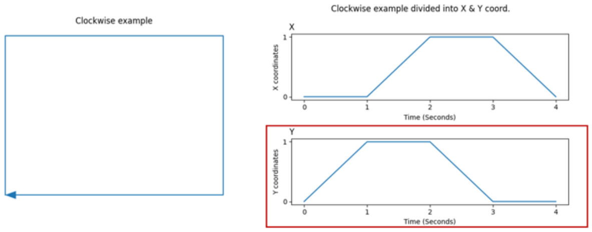


Figure 27. Clockwise example separated to x and y coordinates (y is selected)

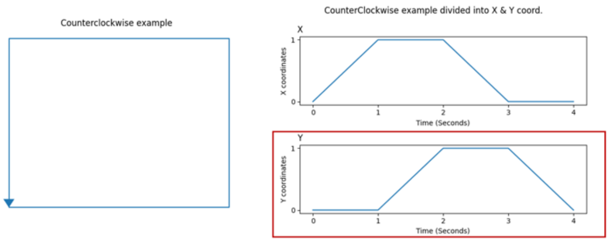


Figure 28. Counterclockwise example separated to x and y coordinates (y is selected)

Step 3. Calculating DTW Cost Matrix

Once the DTW algorithm was applied, the DTW cost matrix is computed as Figure, which is a transpose matrix of the DTW cost matrix in Chapter 2.1. Again, the warping path and the DTW distance can be computed as the cost matrix has been finalized.

Reference sequence					
$j = 5$	2	2	2	2	0
$j = 4$	2	2	1	1	0
$j = 3$	2	2	0	0	1
$j = 2$	1	1	0	0	1
$j = 1$	0	0	1	2	2
	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$

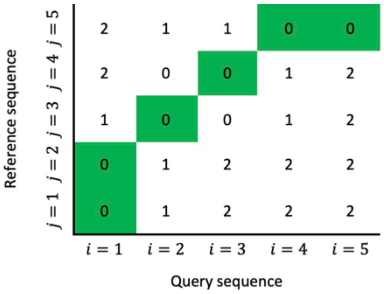
Query sequence

Figure 29. DTW cost matrix between y coordinates of clockwise and counterclockwise

Step 4. Getting the optimal warping path and the DTW distance

Since the DTW cost matrix between the y coordinates of clockwise and counterclockwise is a transpose matrix of the cost matrix between the x coordinates of two circular visual scanpaths, the warping path is also identified as flipped version of that of x coordinates. Again, the warping path can be determined as below. The warping path is also visualized on the cost matrix as in Figure.

$$\begin{aligned} w_1 &= (1,1) \\ w_2 &= (1,2) \\ w_3 &= (2,3) \\ w_4 &= (3,4) \\ w_5 &= (4,5) \\ w_6 &= (5,5) \end{aligned}$$



Reference sequence	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
$j=5$	2	1	1	0	0
$j=4$	2	0	0	1	2
$j=3$	1	0	0	1	2
$j=2$	0	1	2	2	2
$j=1$	0	1	2	2	2

Query sequence

Figure 30. Warping path selected from the DTW cost matrix

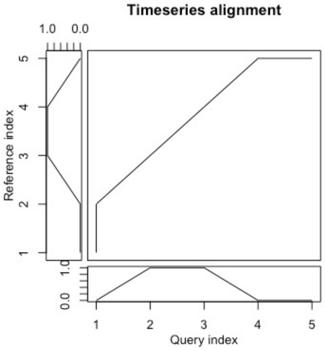


Figure 31. Warping path plot generated using R package

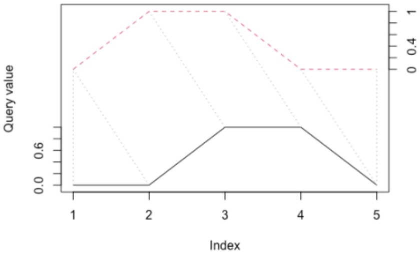


Figure 32. The alignment of the warping path between the query sequence (black) and the reference sequence (red dashed)

2.3 Multivariate DTW computed on Clockwise and Counterclockwise

According to the result of Chapter 2.1 and 2.3, the DTW algorithm was not able to detect any dissimilarities between x coordinates and y coordinates of clockwise and counterclockwise circular scanpath. Therefore, multivariate DTW was applied to consider the x and y coordinates at the same time. Here, the distance (i.e. cost function) $d(i,j)$ that has been utilized for DTW is defined as follwong

$$d(i,j) = \sqrt{\left(qx_i - rx_j\right)^2 + \left(qy_i - ry_j\right)^2}$$

Step 1. Defining the sample scanpath

The same scanpaths from Chapter 2.1 and 2.2 are investigated except that both x and y coordinates of scanpaths were considered as the query and reference sequence at the same time (see Table).

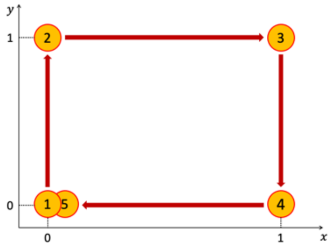


Figure 33. Clockwise scanpath example

Table 8. Raw data extracted for clockwise scanpath example

Time (seconds)	Fixation coordinates x	Fixation coordinates y	Fixation duration (second)
1	0	0	1
2	0	1	1
3	1	1	1
4	1	0	1
5	0	0	1

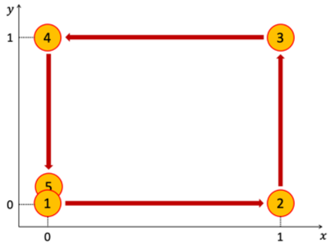


Figure 34. Counterclockwise scanpath example

Table 9. Raw data extracted for counterclockwise scanpath example

Time (seconds)	Fixation coordinates x	Fixation coordinates y	Fixation duration (second)
1	0	0	1
2	1	0	1
3	1	1	1
4	0	1	1

Step 2. Preprocessing the sequences before applying algorithm

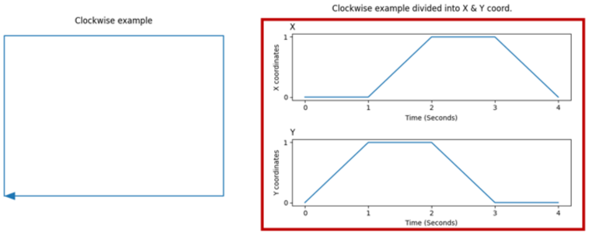


Figure 35. Clockwise example separated to x and y coordinates (both selected)

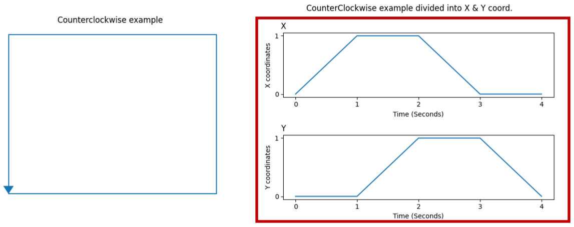
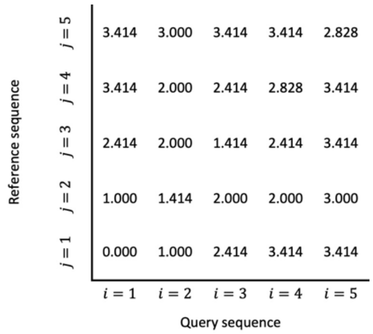


Figure 36. Counterclockwise example separated to x and y coordinates (both selected)

Step 3. Calculating DTW Cost Matrix

Here, the clockwise scanpath was used as the query sequence and the counterclockwise scanpath as the reference sequence. The DTW cost matrix is computed as shown in Figure 39 below.



Reference sequence					
$j = 5$	3.414	3.000	3.414	3.414	2.828
$j = 4$	3.414	2.000	2.414	2.828	3.414
$j = 3$	2.414	2.000	1.414	2.414	3.414
$j = 2$	1.000	1.414	2.000	2.000	3.000
$j = 1$	0.000	1.000	2.414	3.414	3.414
	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
	Query sequence				

Figure 37. DTW cost matrix between x and y coordinates of clockwise and counterclockwise

Step 4. Getting the Optimal Warping Path and the DTW Distance

The warping path and the multivariate DTW distance between the two circular scanpaths were determined. Interestingly, there was no warp occurred between the sequences to identify the shortest distance among them. However, multivariate DTW algorithm was able to spot the dissimilarity between the clockwise and counterclockwise scanpaths. Here, the DTW distance is computed as 2.828.

2.4 Multivariate DTW expanded to other simple cases with linear movements (Z and N shape)

To further study how DTW algorithm would work on simple visual scanpaths that have only 4 AOIs or eye fixations, multivariate DTW algorithm that was applied in Chapter 2.3 was expanded to other simple linear visual scanpaths – N shape and Z shape.

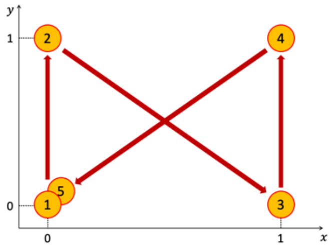


Figure 38. N shape scanpath example

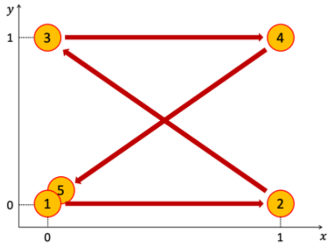


Figure 39. Z shape scanpath example

Table 10. Raw data extracted for clockwise scanpath example

Time (seconds)	Fixation coordinates x	Fixation coordinates y	Fixation duration (second)
1	0	0	1
2	0	1	1
3	1	0	1
4	1	1	1
5	0	0	1

Table 11. Raw data extracted for counterclockwise scanpath example

Time (seconds)	Fixation coordinates x	Fixation coordinates y	Fixation duration (second)
1	0	0	1
2	1	0	1
3	0	1	1
4	1	1	1

Now the dissimilarity (i.e. DTW distance) between each pair of four simple visual scanpaths are obtained. Subsequently, the MDS can be applied to the dissimilarity matrix to plot the visual scanpaths into 2-dimnesional subspace.

Table 12. The input of Multidimensional Scaling method – DTW distance matrix

Linear 2	2.000	2.000	2.000	0.000
Linear 1	2.000	2.000	0.000	2.000
Counterclockwise	2.828	0.000	2.000	2.000
Clockwise	0.000	2.828	2.000	2.000
	Clockwise	Counterclockwise	Linear 1	Linear 2

2.5 MDS Result based on DTW distances among all 4 simple cases

Table 13.The output of Multi-dimensional Scaling method

Scanpath example	MDS axis x	MDS axis y
Circular (Clockwise)	0.242005	0.295367
Circular (Counterclockwise)	-0.234874	0.295007
Linear (Z shape)	0.383859	-0.291347
Linear (N shape)	-0.334893	-0.323312

Since the dissimilarity (i.e. DTW distance) matrix between every pair among 4 simple visual scanpaths are converted to x and y coordinates, they can be plotted on 2-dimensional plane as the following figure. Once the visual scanpaths can be laid on a 2-dimensional space, any clustering method can be applied to further analyse and compare them. Here, DBSCAN algorithm, one of the mostly used density-based clustering method, was applied to see how it can cluster similar visual scanpaths respectively (i.e. whether they can differentiate between circular and linear scanpaths).

DBSCAN algorithm has a two parameters – *minPts* and *eps*(ϵ). The rules of thumb to determine the *minPts* for the algorithm are that *minPts* should be greater or equal to the dimensionality of the data and the value of *minPts* is usually calculated as multiplying 2 to the number of dimensions. For the case study here, the *minPts* can be computed as $2 \times 2 = 4$, following the rule of thumb, since we have laid the 4 simple visual scanpaths onto a 2-dimensional xy coordinate space. However, if the parameter – *minPts* of DTW, were set to 4, the algorithm is naturally going to cluster 4 visual scanpaths altogether into a same cluster. Therefore, the lower bound of *minPts* was used for this case study.

Once the *minPts* was determined, another parameter – eps(ϵ) can be identified subsequently. To get the optimal value of the distance which is used within DBSCAN algorithm to define whether a point is in high or low-density area, heuristic approach (i.e. elbow method) is applied. The kth nearest distance from each point in the dataset is calculated and sorted to spot the elbow point. The value of k is usually selected as the *minPts*-1 for DBSCAN method. In details, which value of eps can return best clustering result in terms of applying DBSCAN method with $minPTS$ by looking at the $(minPts - 1)$th nearest distance plot. In Figure, the kth nearest distance ($k = 1$ for this case since *minPts*=2) for each visual scanpath has sorted. Here, the elbow point (or knee point) to determine the *eps*(ϵ) can be characterized as sudden increase of kth distance on x axis $= 1$. After the elbow point has caputred, the kth distance of the elbow point can be assigned to an eps as a parameter of DBSCAN method.

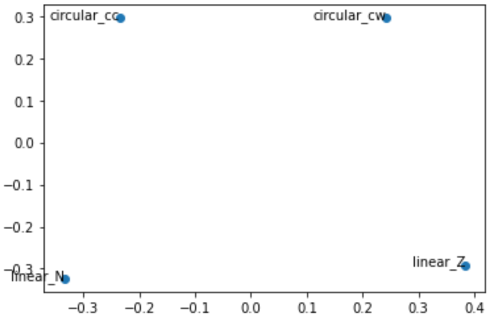


Figure 40. MDS result on 4 simple scanpaths

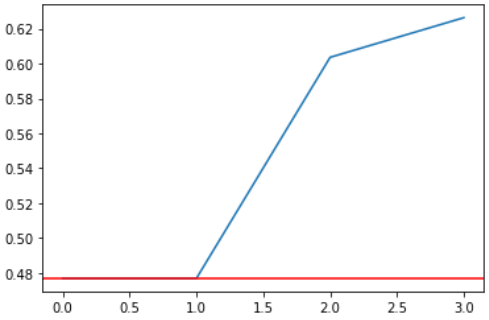


Figure 41. kth nearest distance plot based on MDS result

2.6 DBSCAN result among all 4 simple cases

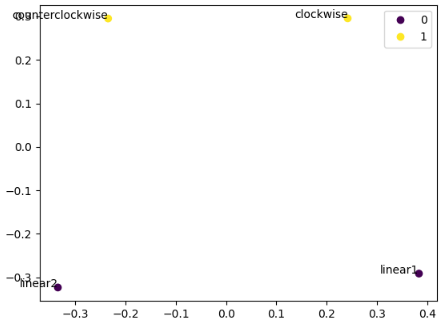


Figure 42. DBSCAN result on 4 simple scanpaths

DBSCAN was applied on the coordinates obtained from MDS to cluster the visual scanpaths based on their density feature. Here, the parameters for DBSCAN, the minimum distance that is used for determining the density of the cluster, was set to $eps = 0.48$, and the minimum number of points that are required to form a cluster is set to 2 – since a cluster with only one data point is not necessary. As a result, DBSCAN algorithm clustered the clockwise and the counterclockwise visual scanpaths as one group, while the rest of the visual scanpaths (Z and N shape) were categorized as outliers since they were classified as low density points (i.e. noise points) according to the algorithm.

Since the DBSCAN provided only a single cluster with two outliers as the result, Silhouette score which is used for evaluating the performance of DBSCAN (i.e. how the clusters are well separated from one to another) cannot be calculated here. However, if Silhouette score is computed while assuming the two outliers as another cluster, the score would be 0.18 – which is larger than 0, but it is still closer to 0 than 1. The reason behind that Silhouette score looks not so promising here would be the intra-cluster distance of linear scanning patterns calculated too high (since they are low-density points).

2.7 Conclusion of Case Study

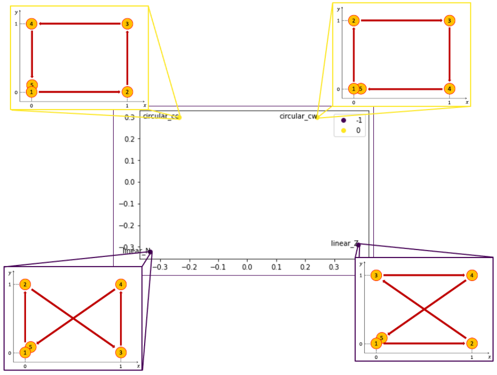


Figure 43. DBSCAN result shown with each scanpath visualization

According to the result shown in Chapter 2.6, the approach – defining DTW distance as a dissimilarity between scanpaths, then laying them on a 2-dimensional MDS space, and clustering them using DBSCAN algorithm, successfully completed the visual scanpaths clustering by the visual scanning strategies (i.e. circular scanning pattern vs. linear scanning pattern). Therefore, if the approach could be studied further so whether it can be replicated to the actual ATCs' visual scanpaths and cluster them by scanning strategies, it will help the researchers and FAA better understand and characterize the visual scanpaths. Having more clear insights on ATCs' visual scanpaths and mapping the scanning patterns with a variety type of tasks in air traffic control can lead to reducing the amount of time and cost required for training the controllers.

Chapter 3. Experiment: Air Traffic Management

In Chapter 3, it is explained how the visual scanpath data were obtained throughout the experiment which was conducted as a part of the FAA funded project A18-0232. All the scenarios and the experiment settings were designed by the team in FAA, and the visual scanpath data were collected by the team as well. The researcher of this study had participated on the project A18-0232 as one of the research assistants. The visual scanpath data acquired from expert ATCs during the experiment were approved to be used for this study.

The main goal of the FAA funded project A18-0232 was to investigate how those findings identified on the visual scanpaths of ATCs in experimental settings could be applied to the visual scanpaths of ATCs in field airports. Therefore, the experiments were conducted on active ATCs in field from the following airports: MCO (Orlando International Airport), MSP (Minneapolis-Saint Paul International Airport), DEN (Denver International Airport), and APA (Centennial Airport). For this study, the data collected from MCO airport have been selected to see whether the approach demonstrated by Kang et al. (2020) in offshore drilling area and also shown with simple visual scanpaths in Chapter 2 would be comparable to the application of air traffic management tasks in tower settings.

3.1 Participant

Four ATCs who are active in the field and currently working at the MCO airport at the point were selected to participate in the experiment. The controllers have multiple years of experiences in the field. More information related to the participants are not available at the moment, and the researcher had requested the information from FAA.

3.2 Apparatus

One of the goals of the experiment was to get a naturalistic response from ATCs as best as possible. Therefore, a high-fidelity virtual environment of air traffic control tower was needed for the experiment. To create such an environment, a variety of futuristic devices were introduced. The ten 60" displays were vertically attached to each other to simulate the views from inside of the control tower to the taxiway and the runway of the airport. Other monitor screens were displayed to simulate the radar screens in air traffic control tower (e.g. ground radar, weather radar, etc.).

To collect the eye movements of ATCs during the air traffic management tasks, the eye movement tracker (Tobii Pro Glasses 2) was used for recording eye movements and POV (point of view) of ATCs throughout the experiment. The software Tobii Pro Lab 3 provided by the manufacturer of the eye tracker is used for processing and extracting the data for further cleansing and analyzing phases.

Figure 44. Tobii Pro Glasses 2 (Tobii Pro, 2018)

3.3 Scenario

The scenarios were developed to collect ATCs' cognitive response during the tasks by capturing eye movements of the controllers. The movements of aircraft (i.e. taking off, lining up, landing, holding, and etcetera.) were predetermined to be presented to the participants. Aircraft movements that are included in the scenario were played on the simulating screen. Given the aircraft movement based on the scenario, ATCs were instructed to give clearances (i.e. Cleared to Land, Cleared to Take Off, Hold Short, and Line Up and Wait) to the aircraft.

Table 14. Clearances that were instructed to the participants to give during the tasks

Clearances	Abbreviation	Definition
Cleared to Land	CTL	A command to an aircraft approaching the airport that the runway is cleared and ready for the landing
Cleared to Take Off	C2TO	Similar to CTL, but aircraft is taking off. C2TO is a call sign for an aircraft to initiate the taking off.
Hold Short	HS	Waiting in line before the aircraft is entering the runway or taxiway
Line Up and Wait	LUAW	For the aircraft to departure, positioning the aircraft onto the runway—used to called "position and hold" in the past

The AOIs were defined within scenario for the experiment. The AOIs defined for the MCO airport are Departure Corridor (C), Arrival Crossing/Departure End (G), Hotel 3-4 (H), Touchdown (T), Final Approach (F), BRITE (B), ASDE (A), ATIS(S), Flightstrips (I). Theses AOIs are specified based on the inputs that were provided by veteran ATCs and Dr. Jerry Crutchfield from FAA for the project A18-0232. The same AOI configurations were used for this study with a permission. Even though the AOIs were determined within scenario, they were hidden during the experiment to the participants. The predetermined AOIs should be defined within Tobii Pro Lab as well for the following data processing.

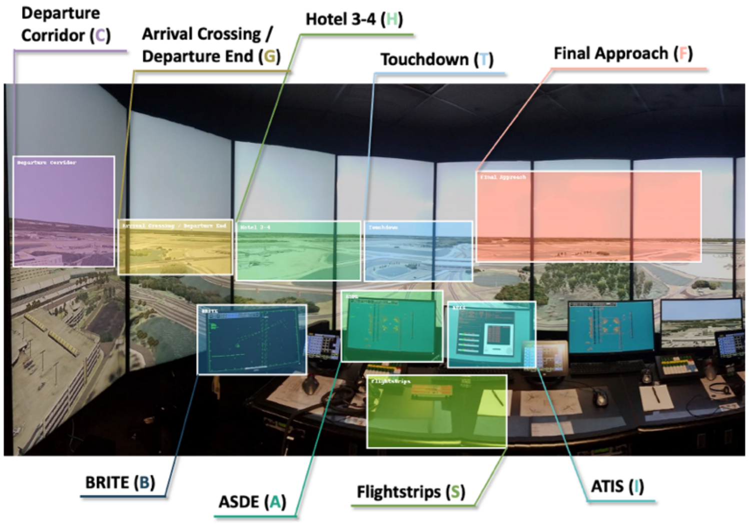


Figure 45. AOIs defined and overlaid on high-fidelity experiment

Table 15. AOIs with the abbreviation and description

AOI Name	AOI Letter	Description
Departure Corridor	C	Where aircraft can be last seen when the aircraft took off the airport and departuring
Arrival Crossing/Departure End	G	Ending point of the runway where aircraft takes off
Hotel 3-4	H	Part of taxiway in the middle of the runway where aircraft can intersect and move accordingly
Touchdown	T	Part of runway where aircraft touches upon during the landing
Final Approach	F	Where aircraft can be first seen when they are approaching to the airport for landing
BRITE	B	The radar display which has the information of aircraft in the sky around the airport
ASDE	A	Ground radar shows layout of aircraft on the surface of airport
ATIS	I	Weather radar for forecasting the climates near around the airport
Flightstrips	S	The paper tags that the controllers use to keep track of flight

3.4 Task and Procedure

The task for the experiment was to obtain a recording of eye movements of the controllers. To begin the experiment, first a participant wore the eye movement tracker. Then the calibrating process was done for each participant, so the quality of the recordings could be guaranteed. Once the calibrating phase was completed, the controllers were instructed to give appropriate clearances (e.g. CTL, C2TO, HS, and LUAW) to the aircraft, based on the situations that are simulated from the scenarios. During the entire task, the eye movements and POV of ATCs are recorded through the tracking device.

3.5 Data Extraction and Processing

The raw data pulled out from the software would look like Table. Each row shows the eye movement that has been collected every 10 ms. Only the eye movement types that match fixation are selected to extracted here (i.e. saccades and other types of eye movements are filtered). The first column has the timestamp information which increases 10 ms as it proceeds to a next row which is the capturing rate of the tracker device. The two following columns indicate x and y coordinates of eye movement that are converted to lie on the snapshot of the virtual environment. The following columns indicate on which AOI the eye movement has placed. The values for these columns – 0 meaning no fixation on the AOI, and 1 meaning that the eye movement has occurred on the certain AOI. For instance, here in Table, one eye fixation that has occurred at the AOI Departure Corridor for 580 ms is shown as an example.

Table 16. An Example of Raw Data extracted from Tobii Pro Lab software

Recording timestamp	Mapped fixation X	Mapped fixation Y	Gaze event duration	AOI hit [Butter Bars]	AOI hit [Departure Corridor]	AOI hit [Departure End/Intersection]	⋯
640	766	441	580	0	1	0	⋯
650	766	441	580	0	1	0	⋯
660	766	441	580	0	1	0	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
929	766	441	580	0	1	0	⋯
939	766	441	580	0	1	0	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

To obtain scanpaths of expert ATCs that have the duration of 30 seconds prior to the clearance, the above raw data have been chunked into 30 seconds length data with a different format which is shown in following Table. Here, the rows of data are still eye fixation that has occurred during the experiment, but it doesn’t mean the 10 ms rate that was presented in the raw data. The first two columns indicate the AOIs where the fixations have occurred. The column 3 and 4 are start and end time of that eye fixation, and the ‘Gaze_Time’ is the duration of each eye fixation, which can be computed as the subtraction between ‘End_Time’ and ‘Start_Time’.

Table 17. The data processed for Participant 2 Scenario 1 Cleared to Land Sample 6

AOI_Hit	AOI Letter	Start_Time	End_Time	Gaze_Time
Arrival Crossing / Departure End	G	1607876	1608496	620
Departure Corridor	C	1608556	1608756	200
Flightstrips	S	1609775	1610135	360
ASDE	A	1610695	1611075	380
BRITE	B	1611494	1612833	1339
Departure Corridor	C	1613253	1613773	520
Arrival Crossing / Departure End	G	1613833	1613973	480
ASDE	A	1614992	1615292	680
Flightstrips	S	1616371	1616871	500
ASDE	A	1617291	1617751	460
Touchdown	T	1618110	1618630	520
Flightstrips	S	1619349	1619609	260
ASDE	A	1620229	1620349	840
Touchdown	T	1621868	1622008	2379
Arrival Crossing / Departure End	G	1625166	1625946	2620
Touchdown	T	1630443	1630703	640
Hotel 3-4	H	1631182	1631242	460
ASDE	A	1632102	1632382	280
BRITE	B	1632981	1633121	1039
Touchdown	T	1634700	1635380	1760
BRITE	B	1637398	1638138	740

Once the data have been processed into a format of Table, they can be further processed into two types for better analysis: 1) visual scanpaths and 2) scanpath sequences. Here, in Figure, both of processing method, 1) and 2) are demonstrated. The scanpath examined in Table can be visualized as a visual scanpath as Figure, while it could be converted into scanpath sequences: **GCSABCGASATSATGTHABTB**.

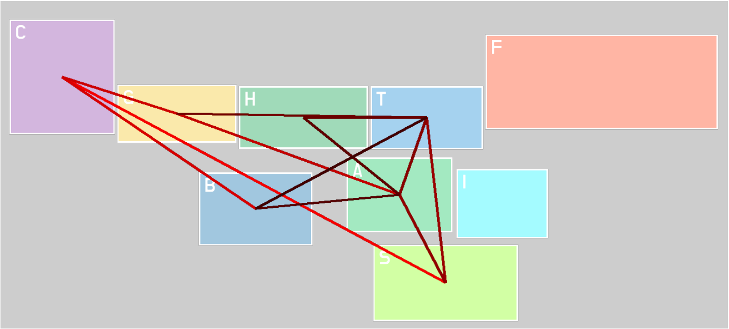


Figure 46. The visual scanpath (sample 6) of Participant 2, 30 seconds prior to Cleared to Land

3.6 Results

3.6.1 10 Samples Considering Single Participant, Single Clearance (Cleared to Land)

1) Participant 1 (P1) and Cleared to Land (CTL)

The approach – defining a dissimilarity between visual scanpaths with DTW, projecting the scanpaths on a 2-dimensional space, and then clustering them using DBSCAN algorithm performed well on the case study with 4-

simple visual scanpath sequences. It has successfully separated the visual scanpaths with circular movements and the others with linear movements. In detail, the DBSCAN algorithm clustered two visual scanpaths with circular movements (i.e. clockwise, and counterclockwise) within a same group while appointing other two visual scanpaths with linear movements (i.e. N shape and Z shape) as noises (i.e. outliers).

Subsequently, the approach was applied to the visual scanpath sequence data of ATCs active at MCO airport in a similar manner to the case study. There are 160 visual scanpath data in total – 10 sample data for all 4 participants and 4 clearances. Since the case study was conducted on 4 sequence data, it was applied to 10, and 40 numbers of samples in order, to see how the result might vary from that of Chapter 2. First, it was applied to the 10 selected samples considering only a single participant and a single clearance (the configuration of Participant 1 (P1) and Cleared to Land (CTL) was demonstrated in Chapter 3.6.1).

First, the dissimilarity between the pairs of visual scanpath sequences have been calculated and then formatted into a distance matrix format as shown in Table 18 below. (The detailed steps of calculating the dissimilarity between visual scanpath sequences are skipped in this Chapter). The distance matrix always has a $n \times n$ symmetric shape. In details, there are $2 \times {}_nC_2$ dissimilarity elements (only upper ${}_nC_2$ elements are shown in Table 18) and n diagonal elements which equal to 0 based on the definition of dissimilarity (i.e. $n^2 = 2 \times {}_nC_2 + n$). Afterwards, Multi-Dimensional Scaling method was applied to the distance matrix to project the 10-sample data on a 2-dimensional subspace.

Table 18. The DTW distance matrix of 10 samples considering Participant 1 and Cleared to Land

	sample9	sample3	sample6	sample4	sample1	sample2	sample7	sample5	sample10	sample8
sample9	0.00	2797.19	1638.04	2082.22	2028.06	2394.35	1556.80	2372.46	2013.60	2285.80
sample3		0.00	2404.03	2526.74	2505.58	2563.80	2618.00	2017.72	2244.97	2720.89
sample6			0.00	1349.60	1535.88	1906.97	890.85	2332.13	1335.43	1954.74
sample4				0.00	2132.63	2476.04	1772.70	2101.14	1721.74	1618.55
sample1					0.00	3090.97	1647.73	3136.24	1678.45	1685.33
sample2						0.00	1787.02	2052.71	2546.85	2529.58
sample7							0.00	2025.48	1269.71	1787.13
sample5								0.00	2103.29	1733.78
sample10									0.00	2102.67
sample8										0.00

Once the 10 visual scanpath sequence data considering a single participant and a single clearance has lied on a 2-dimensional space, they can be clustered for further analysis and comparison. For this study, DBSCAN was selected for the clustering method, therefore, the $minPts$ and eps (ϵ) – which are the parameters for the algorithm-- should be determined before applying the algorithm. The $minPts$ can be easily determined by following the rule of thumb – the lower bound as the dimensionality of the space, and $minPts = 2 \times (dimensionality\ of\ the\ dataset)$. On the other hand, to determine eps (ϵ), the other parameter of DBSCAN, the kth nearest distance plot is needed to observe the elbow point. Here, to identify the eps (ϵ), 3rd nearest point distance was computed and sorted as shown in Figure. There was a subtle bump at point index $= 2$ on x-axis, therefore, the value around 0.35 was selected for the parameter value.

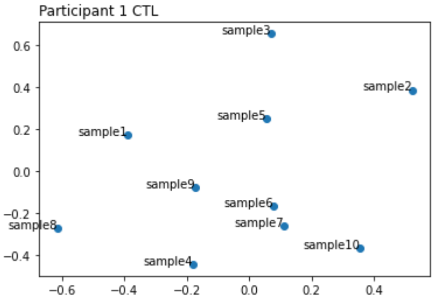


Figure 47. MDS result of 10 samples considering P1 and CTL

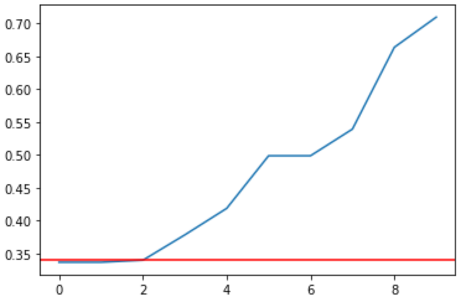


Figure 48. kth nearest plot computed from 10 samples considering P1 and CTL

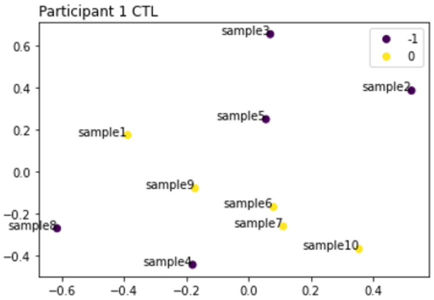


Figure 49. DBSCAN result on 10 samples considering P1 and CTL

Therefore, DBSCAN with parameters of $eps = 0.35$, $minPts = 4$ was applied to cluster 10 samples (considering Participant 1 and Cleared to Land). On the above plot, label '-1' indicates outlier and '0' indicates the cluster of the visual scanpath sequence points grouped together based on DBSCAN clustering method. Since the DBSCAN result provides a single cluster, Silhouette score was not able to be computed (the metric requires minimum of 2 clusters to be calculated). In Table 19, each sample was listed as visual sequences, and highlighted accordingly to show how each of the samples were clustered by DBSCAN algorithm.

Table 19. DBSCAN result of 10 samples considering P1 and CTL in scanpath sequences

Participant 1, Cleared to Land	
sample 1	**GATASGTBGBHG**
sample 2	**TGBATBTGAGTABTI**
sample 3	**GHAHGHBHSGASB**
sample 4	**THGATSTGTGAB**
sample 5	**BHGBTHGHTSTGABT**
sample 6	**TGAGTABH**
sample 7	**TATGTHTBT**
sample 8	**ASGHTSTGAGBGT**
sample 9	**AGTHTHTHTHTABTG**
sample 10	**THTHTHGATBGBH**

The scanpaths that were grouped into the same cluster are visualized as visual scanpaths in the following figures (Figure 52 – 56) to see whether we could visually identify a clear difference shown in Figure below. Here, it can be identified that most of the visual scanpath sequences are sharing similar geometrical characteristics. First, all 5 of sequences are sharing the eye movement between three AOIs – Arrival Crossing / Departure End (G), Hotel 3-4 (H), and Touchdown (T) that are placed in the middle of top row. The Touchdown (T) and BRITE (B) are the AOIs that the ATCs paid attention most frequently. It supports that the clearance used here for the analysis was Cleared to Land, so the ATCs had to keep check on whether the Touchdown (T) area was cleared for the landing. Also, the controllers had to make sure that if there are any other aircraft that can't be spotted by looking at the window of the control tower by paying attention to the ground radar (i.e. BRITE (B)).

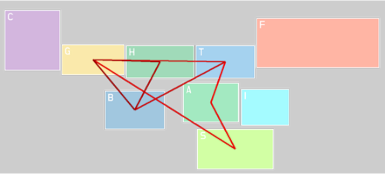


Figure 50. Sample 1 of P1 CTL

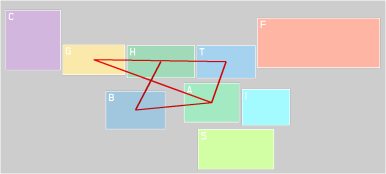


Figure 51. Sample 6 of P1 CTL

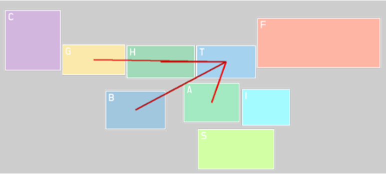


Figure 52. Sample 7 of P1 CTL

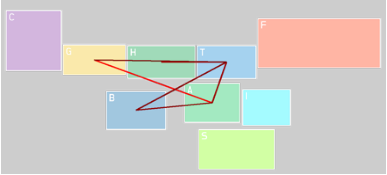


Figure 53. Sample 9 of P1 CTL

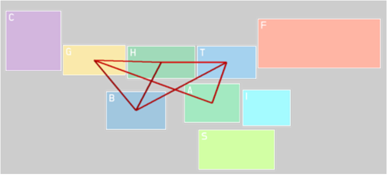


Figure 54. Sample 10 of P1 CTL

Another finding from a single participant and a single clearance clustering could be identifying a representative scanpath since there is only single cluster. A representative scanpath can be described as a golden standard scanpath that contains most of the features that can illustrate the configuration (a single participant and a single clearance here). If a center point of cluster can be determined, the center point is highly likely to be the representative scanpath of the cluster. However, DBSCAN algorithm doesn't provide a center point of each cluster – the algorithm only specifies whether each data point is a core point, border point, or noise point. Therefore, it was assumed that the core points which have the relatively small dissimilarity values to other data points can be the representative scanpath candidates.

In 10 samples considering P1 and CTL, the data points which are categorized as core points are sample 6, 7, and 9. Therefore, the DTW distance between ${}_3C_2$ pairs out of these scanpath sequences can be revisited to determine representative scanpath. In Table 20, DTW distance of three pairs are compared. The sample 6 and 7 have the shortest DTW distance which means that the least dissimilarity exists between two sequences among 10 samples considering P1 and CTL. The next pair of sequences that has second smallest value is the sample 6 and 9. Therefore, the sample 6 can be selected as a representative scanpath for P1 on CTL.

Table 20. DTW distance revisited for selecting the representative scanpath

index	query	DTW distance
sample 6	sample 10	1829.72
sample 6	sample 7	1841.69
sample 6	sample 4	1906.85
sample 6	sample 1	2133.42
⋮	⋮	⋮

2) Participant 2 (P2) and Cleared to Land (CTL)

Afterwards, the approach was applied to other CTL samples from the rest of the participants (P2 in this section). Again, 10 CTL samples are gathered for each participant since there are total of 40 samples for CTL clearance. As it is shown in Figure 57, DBSCAN method was able to detect 2 noise visual scanpaths (i.e. sample 1 and 6) from P2 on CTL. In addition, the clustering result was shown with scanpath sequences in Table 21. The first that was noted was that the two outliers – sample 1 and 6, are marginally longer than the rest of the sequences. This finding can be interpreted as the DTW distance is easily affected by the length of sequences.

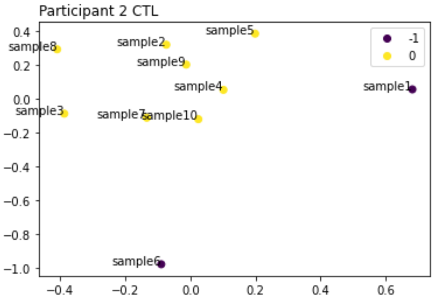


Figure 55. DBSCAN result on 10 samples considering P2 and CTL

Table 21. DBSCAN result of 10 samples considering P2 and CTL in scanpath sequences

Participant 2, Cleared to Land	
sample 1	**THBGASBTHTHATGBATBH**
sample 2	**THASBTGTBT**
sample 3	**ASATBTGABT**
sample 4	**ATATATGBSTAB**
sample 5	**THBGBSATBTBTI**
sample 6	**GCSABCGASATSATGTHABTB**
sample 7	**BTIGCGABT**
sample 8	**SABGHTHGTSBTG**
sample 9	**TGCTATATBTH**
sample 10	**ATATGCGTATHBTG**

In Table 21, it can be shown that sample 4 and 10 both start with back-and-forth eye movements between the AOIs – A (ASDE) and T (Touchdown). For further comparison of these results, 4 samples among 10 are selected to be visualized. From Figure 58 to 61, the eye movements from sample 2, 4, 9, and 10 are drawn.

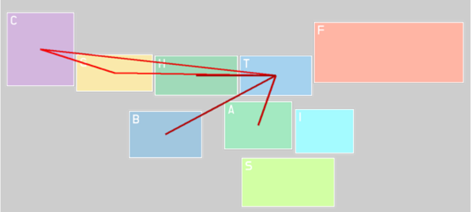


Figure 56. Sample 9 of P2 and CTL

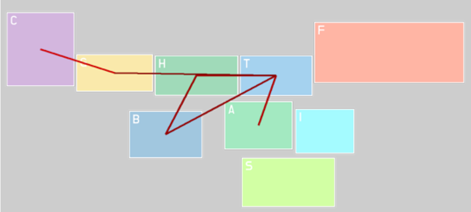


Figure 57. Sample 10 of P2 and CTL

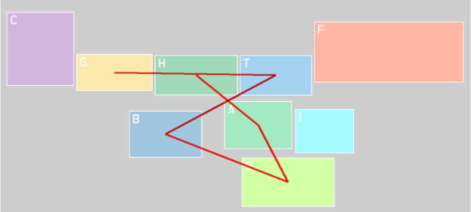


Figure 58. Sample 2 of P2 and CTL

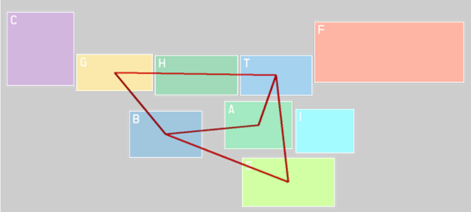


Figure 59. Sample 4 of P2 and CTL

Every point in the visual scanning pattern cluster '0' for 10 samples considering P2 and CTL was categorized as a core point during the DBSCAN algorithm. The sample 9 was identified as a representative scanpath for the configuration (P2 and CTL) since it has the shortest DTW distance (i.e. the least dissimilarity) to other sequences among 10 sequences. The visualization of eye movements from sample 9 can be found in Figure 58.

3) Participant 3 (P3) and Cleared to Land (CTL)

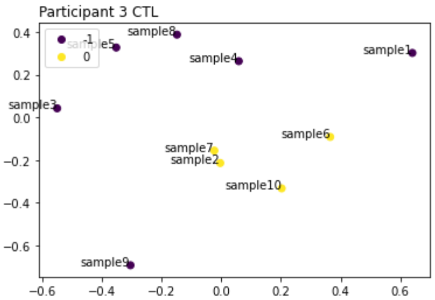


Figure 60. DBSCAN result on 10 samples considering P3 and CTL

For 10 samples considering P3 and CTL, the result can be illustrated as shown in Figure 62. The visual scanning pattern cluster '0' is consist of 4 visual scanpath sequence samples – sample 2, 6, 7, and 10. The layout of the data points in cluster '0' is indicating that sample 2 and 7 are going to be almost identical to each other while the sample 6 and 10 have certain distance from the sample 2 and 7, but still grouped into the same cluster. The scanpath sequences of sample 2 and 7 provided in Table 22 coincide with this finding, the first portion of the samples starts with back-and-forth eye movements between H (Hotel 3-4) and T (Touchdown), and they both end with same AOI sequence – S (Flightstrips), B (BRITE), and I (ATIS), at the last part. It can be interpreted as the participant was checking the runway first to verify whether an aircraft is still located then checking at the near sky to see if there are other aircraft approaching to the airport.

Table 22. DBSCAN result of 10 samples considering P3 and CTL in scanpath sequences

Participant 3, Cleared to Land	
sample 1	GSBGTBSTS
sample 2	**STHTHTSBI**
sample 3	ABSHGTSTIT
sample 4	ASASTHGSTSBSTG
sample 5	SGTABIHGHSB
sample 6	**HGSTBSB**
sample 7	**HTHGHGHTSBI**
sample 8	BAGCSTGSBT
sample 9	STSGSBTSBTG
sample 10	**SATHTABS**

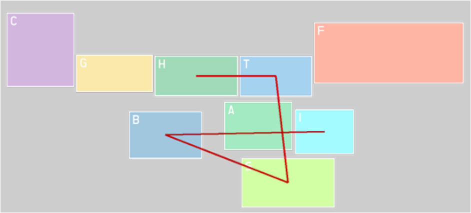


Figure 61. Sample 2 of P3 and CTL

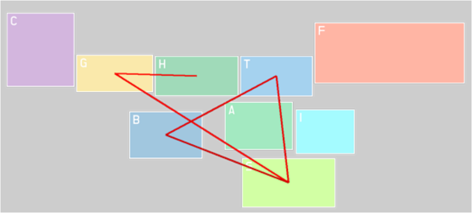


Figure 62. Sample 6 of P3 and CTL

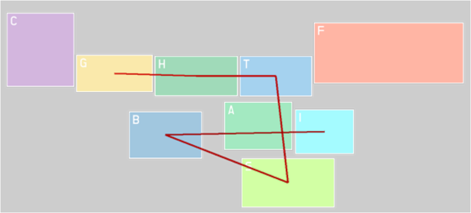


Figure 63. Sample 7 of P3 and CTL

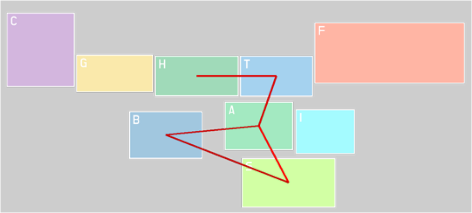


Figure 64. Sample 10 of P3 and CTL

The visualization of saccade movements in sample 2 and 7 supports the finding which was discussed above (see Figure 63 and 65). The sample 10 has the somewhat similar layout to that of the sample 2 and 7, while the sample 6 has totally different layout from them. Therefore, it might be interpreted as that the MDS x axis in Figure 62 indicates the layout (i.e. overall geometrical pattern) of the visual scanpath sequences

The core points of 10 samples considering P3 and CTL are the sample 2 and 10. The sample 2 has the shortest DTW distance to other data points in the visual scanning pattern cluster '0'. Therefore, the sample 2 can be determined as a representative scanpath for this configuration (P3 and CTL). The eye movements from the sample 2 sequence is already shown in Figure 63.

4) Participant 4 (P4) and Cleared to Land (CTL)

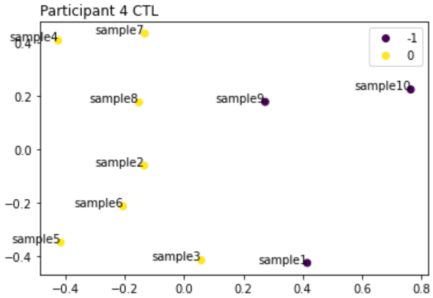


Figure 65. DBSCAN result on 10 samples considering P4 and CTL

The DBSCAN result for 10 samples considering P4 and CTL is shown in Figure 67 above. The Figure 67 illustrates that the sample 1, 9, and 10 are categorized as noise points and grouped into outliers, while the rest of the data points were clustered into the same visual scanning pattern cluster '0'. In detail, the cluster '0' shown in

Figure 67 has formulated a vertical shape where the elements grouped into the cluster are sharing similar MDS x values to each other.

However, when the sample 2 and 8 – which has the most identical x values to each other--are examined in a scanpath sequence format as provided in Table 23, there are no huge similarities or specific patterns that could be captured. The samples 2, 3, 6, and 8 were selected for the further comparison and the eye movements from these samples are visualized throughout Figure 68 to 71. But it was still not able to find any similarities or patterns between the samples 2, 6, and 8 which have similar MDS x values in Figure 67. The reasons behind why interpreting this configuration considering P4 and CTL are further discussed in Chapter 4.

Table 23. DBSCAN result of 10 samples considering P4 and CTL in scanpath sequences

Participant 4, Cleared to Land	
sample 1	BATFTHTHGBHTH
sample 2	TATHTHGCGB
sample 3	ATHGATABHG
sample 4	ATAHTBAGABGCGHTATB
sample 5	HGHTFTHTHTHGBABHT
sample 6	TFTAHGBT
sample 7	GTATATHGCGCGHTABT
sample 8	TSBTBGHTBA
sample 9	GHGCGHBATBG
sample 10	CGHTATHTATBHG

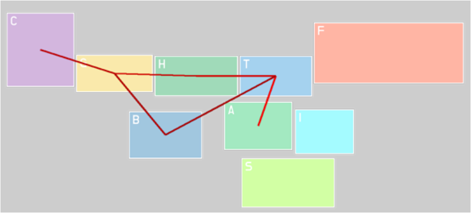


Figure 66. Sample 2 of P4 and CTL

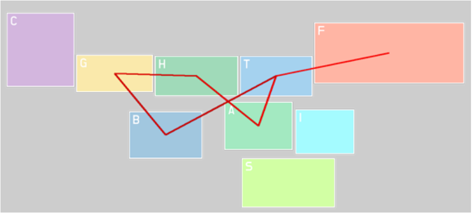


Figure 67. Sample 6 of P4 and CTL

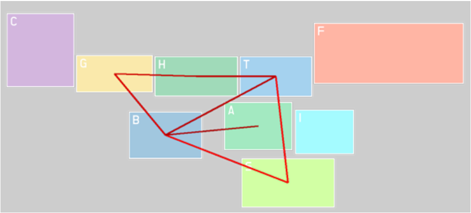


Figure 68. Sample 8 of P4 and CTL

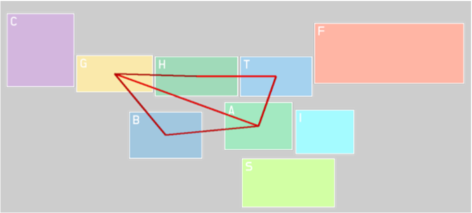


Figure 69. Sample 3 of P3 and CTL

The core points among the 10 samples considering P4 and CTL are the samples 2, 6, and 8. Since the DTW distance between the samples 2 and 6 have the shortest value among the pairs from 10 samples, both the sample 2 and 6 can be selected as the representative scanpaths. But, looking at the aspect that both sequences don't have features or characteristics which representing other scanpaths, it is difficult to determine them as the representative scanpath pair. Again, the reasons behind this interpretation are going to be further discussed in following Chapter.

3.6.2 10 Samples Considering Single Participant, Single Clearance (Cleared to Take Off)

1) Participant 1 (P1) and Cleared to Take Off (C2TO)

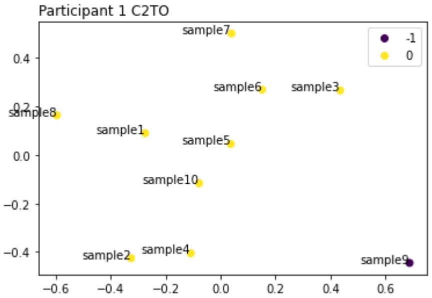


Figure 70. DBSCAN result on 10 samples considering P1 and C2TO

In Figure 70, the DBSCAN result for 10 samples considering P1 and C2TO is provided. The interesting part of the Figure is that there was only a single noise point (i.e. outlier) during the clustering phase. The rest of the data points, indicating the visual scanpaths of P1 giving the C2TO clearance, were grouped into the same visual scanning pattern cluster. Also, it seems that there is a somewhat quadratic relationship existing among the samples 4, 10, 5, 6, and 3. In Table 24, the scanpaths are shown in sequence format for the initial comparison of them. The sample 5 and 10 in Table 24, do not really show a similarity in scanpath sequence format. There are still some back-and-forth eye movements between A (ASDE) and B (BRITE), and eye fixations can be spotted on S (Flightstrips), G (Arrival Crossing / Departure End), H (Hotel 3-4), and T (Touchdown).

Table 24. DBSCAN result of 10 samples considering P1 and C2TO in scanpath sequences

Participant 1, Cleared to Take Off	
sample 1	**GAHAGAGAGTHGA**
sample 2	**HTABHBHGAHBSBATGTG**
sample 3	**GCBTGCBTGA**
sample 4	**GABAGABHGTBHSHTHG**
sample 5	**ABCGATGABAGASGHT**
sample 6	**TCATASTHGTATHGT**
sample 7	**GCGAGTHTHGHAGASATGHT**
sample 8	**AGBGSGASAGTGHGAGTGS**
sample 9	**HBTHGCBHTHTGAHTHGHTHATHGHT**
sample 10	**STGABAGABGTASTHGT**

The 3 samples – the sample 5, 6, and 10, were selected among a group of samples that seems to have a quadratic relationship and lastly the sample 1, which is one of the core points in this configuration considering P1 and C2TO. Here, the eye movements between A and B on the sample 5 and 10 can be captured also in Figure 71 and 72. However, the quadratic pattern found between the samples in Figure 70 was not easy to be interpreted since the scanpaths are too complex. In detail, in Chapter 2.6 it was less complicated to interpret the linear relationship or having a near MDS y axis value since the scanpaths that were used for the case study was simple and straightforward. This part is going to be further discussed in the next chapter.

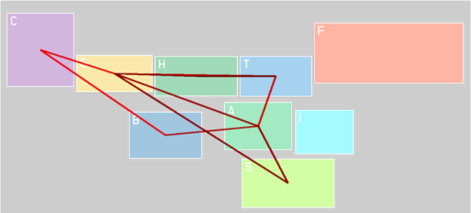


Figure 71. Sample 5 of P1 and C2TO

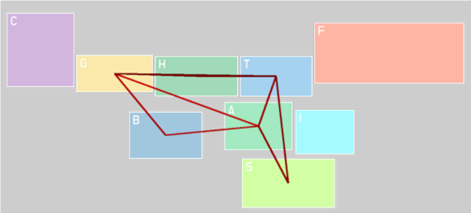


Figure 72. Sample 10 of P1 and C2TO

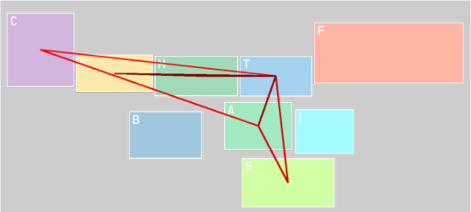


Figure 73. Sample 6 of P1 and C2TO

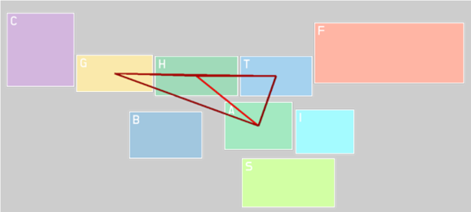


Figure 74. Sample 1 of P1 and C2TO

The other core points that are determined during the clustering phase are the sample 5, 6, and 10. The sample 6 has the shortest DTW distance to other scanpaths in the configuration. Therefore, the sample 6 can be identified as the representative scanpath for 10 samples considering P1 and C2TO.

2) Participant 2 (P2) and Cleared to Take Off (C2TO)

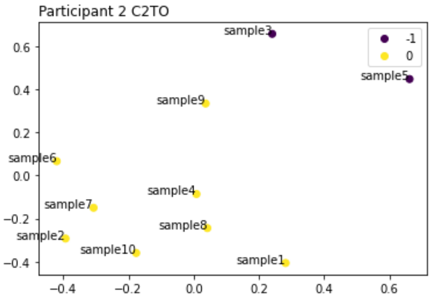


Figure 75. DBSCAN result on 10 samples considering P2 and C2TO

Here, according to the Figure 75, the sample 3 and 5 were classified as noise points (i.e. outliers), and the rest of the samples were clustered to the visual scanning pattern cluster '0' altogether. Another possible quadratic pattern can be found in Figure 75, which was observed in Figure 70 as well. The 10 scanpath sequence samples considering P2 and C2TO were provided as sequence format in Table 25. If the sample sequences of samples 4 and 5 are compared, it can be captured that both sequences share eye fixations on same AOIs, but not the exact saccade movements. In detail, both the samples 4 and 8 have saccade movements consist of eye fixations on G (Arrival Crossing / Departure End)-A (ASDE)-B (BRITE) and H (Hotel 3-4)-T (Touchdown)-A (ASDE), but in a different order—the sample 4 has BGA movement while the sample 8 has GAB movement, and the sample 4 has HTA movement while the sample 8 has THA movement. One way to describe this scanning strategy is applying circular scanning pattern locally in multiple sub AOIs (e.g. G-A-B, and H-T-A).

Table 25. DBSCAN result of 10 samples considering P2 and C2TO in scanpath sequences

Participant 2, Cleared to Take Off	
sample 1	**TGTASATBTHGTSTG**
sample 2	**AGBGABABASAT**
sample 3	**BCGTAHCTACAB**
sample 4	**STABGAHTAHTAHTAG**
sample 5	**BGTHTABASATGBTABCGTHGC**
sample 6	**BASAHGHABTABGA**
sample 7	**AGTASASABGHTHG**
sample 8	**ABTHGABAGATHASTG**
sample 9	**BTGAGAHTABAGABCG**
sample 10	**ATBGSABATGB**

The saccade eye movements selected samples (including the samples 4 and 8) were visualized as below in the Figure 78 to the Figure81 for the further comparison. The participant 2 applying the circular scanning pattern locally among sub-AOIs could be also captured in Figure 76 and 77. A triangular—or an uncompleted triangular without a side—feature can be found in both Figure 76 and 77. The overall shape or pattern of the samples 4 and 8 doesn't coincide with each other though.

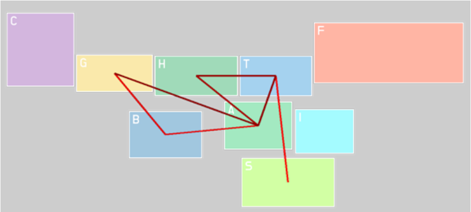


Figure 76. Sample 4 of P2 and C2TO

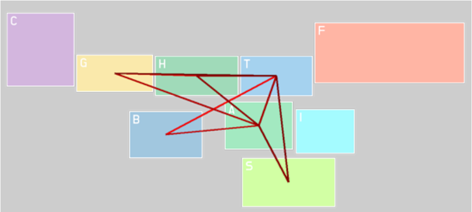


Figure 77. Sample 8 of P2 and C2TO

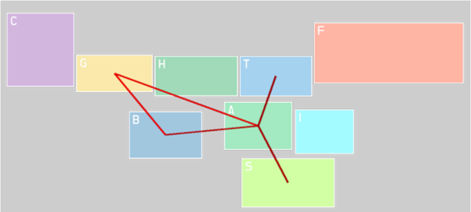


Figure 78. Sample 2 of P2 and C2TO

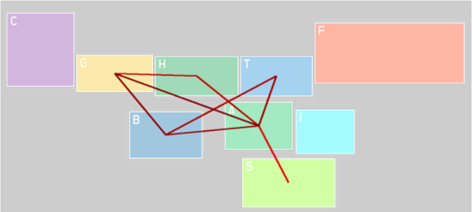


Figure 79. Sample 6 of P2 and C2TO

The core points of this configuration—considering P2 and C2TO—are the samples 2, 4, 6, 7, 8, and 10. The DTW distance between the pairs could be compared to identify the representative scanpath. Here, the samples 2 and 6 had the shortest DTW distance to each other. In Figures 78 and 79, it can be determined that the samples 2 and 6 are almost identical except that the sample 6 has eye fixation on H, and eye movement between B and T while the sample 2 do not.

3) Participant 3 (P3) in Cleared to Take Off (C2TO)

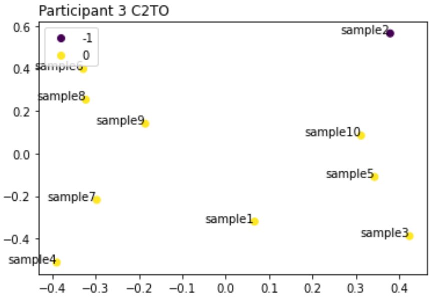


Figure 80. DBSCAN result on 10 samples considering P3 and C2TO

The 10 samples considering P3 and C2TO were categorized into a single scanning strategy cluster '0' with 9 samples and a single outlier (i.e. noise point) in Figure 80. The 10 samples were demonstrated as scanpath sequence and highlighted accordingly to their clustering result in Table 26. One interesting finding is that the sequences can be divided into two clusters based on the layout of the data points, if there was no sample 1 data point. But there are no specific patterns or similarities which can be found in scanpath sequences format among the samples 3, 5, and 10, or the samples 4, 7, and 9.

Table 26. DBSCAN result of 10 samples considering P3 and C2TO in scanpath sequences

Participant 3, Cleared to Take Off	
sample 1	TSTHGASTHT
sample 2	GAGSGTHTHGSASGTG
sample 3	BASABAGASA
sample 4	GASASTH
sample 5	SAGSBSHTG
sample 6	HGTSTITGSGTHGTA
sample 7	TFASAHAGTHT
sample 8	THTGHGCGHTATHT
sample 9	GFTAGSGTGA
sample 10	TGAGSTSTG

To further compare them, the eye movements behavior from the samples 7, 9 and 5, 10 were visualized in the following figures (the Figures 81-84). In detail, it is complicated to capture a similarity or specific patterns on the overall layout of saccade movements of the samples 7, 9 and 5, 10. But it was able to capture that the participant was using circular scanning pattern at the beginning of the sample 9. Here, it was found that the controller started their eye fixation on G then moving to F → T → A then followed by back-and-forth eye movements between G and S. On the sample 7, P3 started with somewhat clockwise scanning pattern on T → F → A, but the rest of the scanpaths were consist of linear eye movements. One of the interesting parts from comparing the samples 7 and 9 is that the participant was looking at the AOI F (Final Approach), which is still the part of sky field near the airport where the aircraft approaches. This might be due to there were another clearance was given to the aircraft during the participant's response was being recorded.

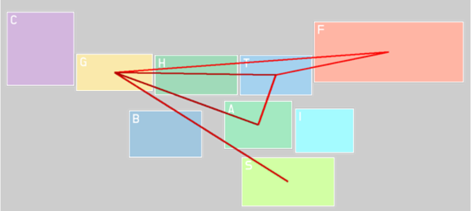


Figure 81. Sample 9 of P3 and C2TO

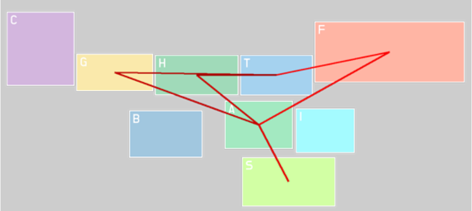


Figure 82. Sample 7 of P3 and C2TO

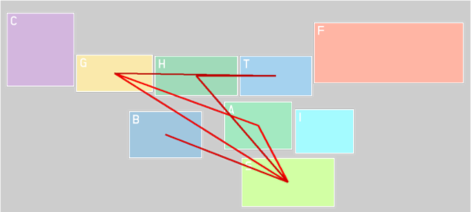


Figure 83. Sample 5 of P3 and C2TO

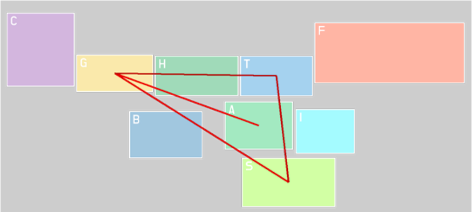


Figure 84. Sample 10 of P3 and C2TO

4) Participant 4 (P4) and Cleared to Take Off (C2TO)

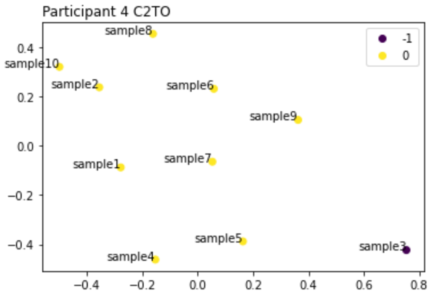


Figure 85. DBSCAN result on 10 samples considering P4 and C2TO

Again, in Figure 85, clustering result of the 10 samples considering P4 and C2TO was shown. Here, only the sample 3 has been categorized as an outlier (i.e. noise point according to DBSCAN algorithm). The rest of the samples were clustered into the same visual scanning pattern group '0' as shown in Figure 85. The 10 samples were described in scanpath sequences, and the clustering result was also highlighted in Table 27.

Table 27. DBSCAN result of 10 samples considering P4 and C2TO in scanpath sequences

Participant 4, Cleared to Take Off	
sample 1	**ABAHTAHA**
sample 2	**BASABGHAT**
sample 3	**HAGCGHTAGSTSTG**
sample 4	**AGATAHTBAGABC**
sample 5	**GAGASABAHATCG**
sample 6	**BTHTFTAHGHGBATHG**
sample 7	**BHBTGATATCG**
sample 8	**HTAHGBGTITHGT**
sample 9	**FAHTHGAGBABATG**
sample 10	**AHTBGATHTHGBSA**

Here, the eye movements of the samples 5, 7 and 6, 9 were visualized to further investigate and capture the visual scanning pattern utilized in the cluster. Overall layout of the samples 7 and 5 were compared in Figure 86 and 87. Both the samples 5 and 7 had the eye fixations and movements on the AOIs A (ASDE), T (Touchdown), and G (Arrival Crossing / Departure End). But the controller examined the AOIs using circular (i.e. counter clockwise) scanning pattern on the sample 7, while the participant was more using back-and-forth eye movement on the sample 5. This finding indicates that the controllers not only using the mixture of the visual scanning strategies in a single scanning but might using different types of scanning strategy within the same clearance (i.e. task) over different scanpaths. But it seems that the DBSCAN clustering wasn't sensitive enough to capture these changes during scanning tasks under the circumstances described in this Chapter.

The eye fixations on F (Final Approach) were also found on the samples 6 and 9 (see the Figures 88 and 89) as it was in the Figures 81 and 82 in the previous section. This also could be that the other clearance commands were involved during the 30-seconds prior to the C2TO commands, or other factor defined in the scenario could affect the participant to fixate their eyes on F (Final Approach).

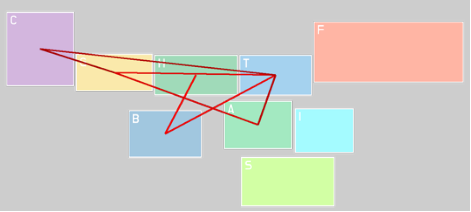


Figure 86. Sample 7 of P4 and C2TO

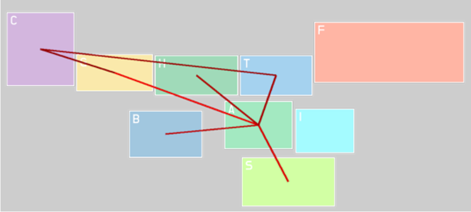


Figure 87. Sample 5 of P4 and C2TO

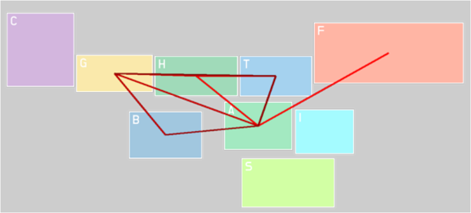


Figure 88. Sample 9 of P4 and C2TO

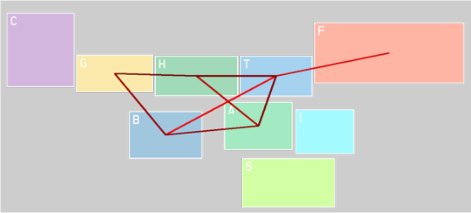


Figure 89. Sample 6 of P4 and C2TO

The core points of the 10 samples considering P4 and C2TO were the samples 1, 2, 4, 6, 7, 8, and 10. As the DTW distances between the core points and the rest of points in the cluster was compared, the representative scanpath for the 10 samples considering P4 and C2TO was identified as the sample 1, which is shown below in Figure 90. But it only has eye fixations on 4 AOIs (A, B, H, and T), and fewer eye movements between the AOIs compared to other scanpath sequences. This can be further discussed in the next chapter.

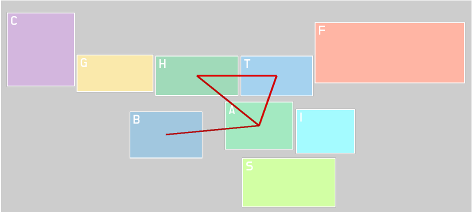


Figure 90. Sample 1 of P4 and C2TO

3.6.3 10 Samples Considering Single Participant, Single Clearance (Hold Short)

1) Participant 1 (P1) and Hold Short (HS)

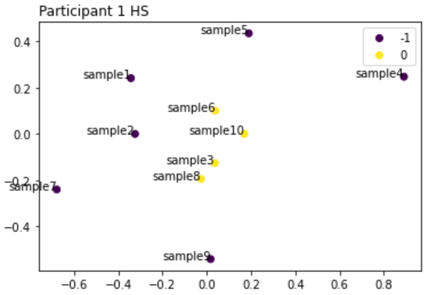


Figure 91. DBSCAN result on 10 samples considering P1 and HS

Now, the 10 samples considering HS clearance are analyzed. Here, P1 and HS were considered for the clustering. As a result, the four samples—sample 3, 6, 8, and 10—were clustered into the same scanning strategy cluster '0' as it is shown in the Figure 91. Each data point of the scanpaths were converted into scanpath sequences and colored according to their clustering result in Table 28. The four scanpath samples were visualized in the following figures for further analysis.

Table 28. DBSCAN result of 10 samples considering P1 and HS in scanpath sequences

Participant 1, Hold Short	
sample 1	BGBATHABGAHAGAG
sample 2	BHBGHGAHGBAG
sample 3	**GTGTGSGAG**
sample 4	GACABAGTSTGAGBGH
sample 5	GTGAGHGAGAHBAGAG
sample 6	**AHGHAHGAGBTAGAGH**
sample 7	HGAHBTHGHGHABGAHG
sample 8	**HGTGABAG**
sample 9	TGTHTHBHBAGBGH
sample 10	**TATHAHGAGTGAGAG**

In the Figures 92, and 93, the controller used linear scanning strategy as their main scanning scheme to complete the task (i.e. giving HS clearance). Especially in the Figure 93, the layout of scanpath resembles the Z shape which was introduced during the simple case study in Chapter 2. In Figure 92, it can be found that the controller didn't scan the AOIs using one of the simple scanning scheme (i.e. Z shape) but still a lot of linear (or back-and-forth) eye movements were made based on the AOI G (Arrival Crossing / Departure End). One of the possible reasons behind that the certain eye movements were made around the AOI G could be that an aircraft was waiting to be positioned on the AOI. In the Figures 94 and 95, the controller used the back-and-forth eye movements for the most of the time during the 30 seconds period.

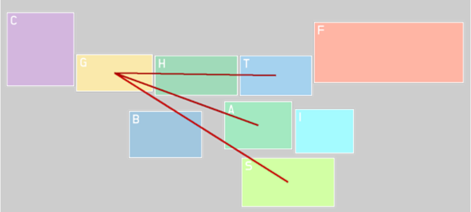


Figure 92. Sample 3 of P1 and HS

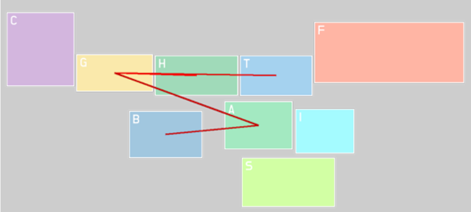


Figure 93. Sample 8 of P1 and HS

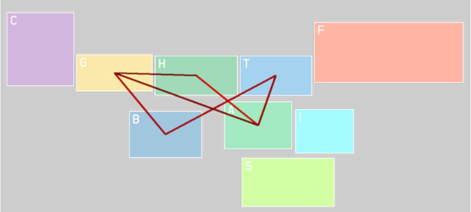


Figure 94. Sample 6 of P1 and HS

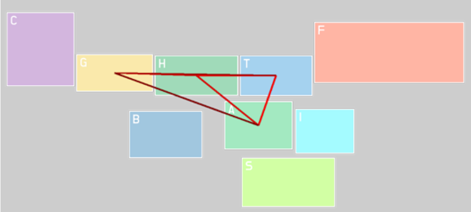


Figure 95. Sample 10 of P1 and HS

2) Participant 2 (P2) and Hold Short (HS)

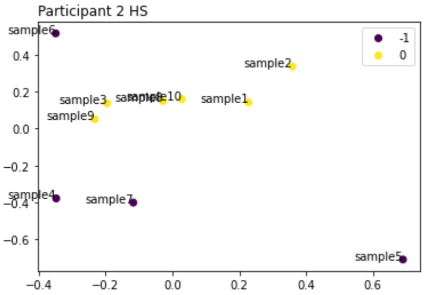


Figure 96. DBSCAN result on 10 samples considering P2 and HS

The 10 samples considering P2 and HS were clustered as shown in Figure 96. The first noticeable finding from the Figure 96 is that all the data points grouped into the same cluster have similar MDS y axis values to each other, while the outliers—noise points—are scattered all over the 2-dimensional MDS space. The second finding is that the sample 8 and 10 were lied close to each other

Table 29. DBSCAN result of 10 samples considering P2 and HS in scanpath sequences

Participant 2, Hold Short	
sample 1	ATAGBAB
sample 2	BGAGABABGAG
sample 3	AHGBTBGAGTG
sample 4	ABGATHSBTAT
sample 5	STSABSTCGBASABGTHT
sample 6	SAHGHGATABATGAGH
sample 7	BHTHIBATAGAT
sample 8	ATATBAIGAG
sample 9	ABTABGASAGTA
sample 10	ABASATGBGAGB

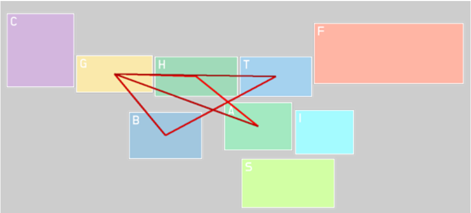


Figure 97. Sample 3 of P2 and HS

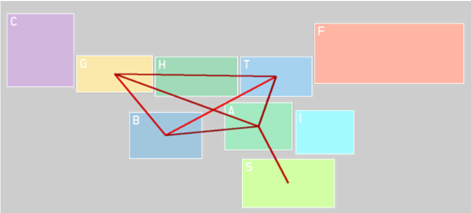


Figure 98. Sample 9 of P2 and HS

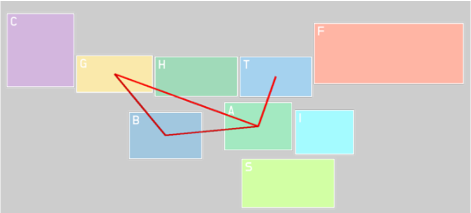


Figure 99. Sample 1 of P2 and HS

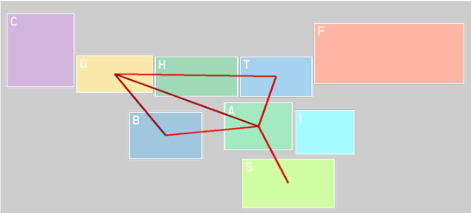


Figure 100. Sample 10 of P2 and HS

3) Participant 3 (P3) and Hold Short (HS)

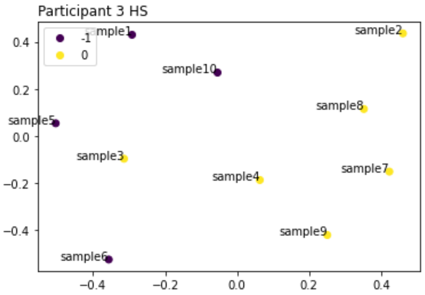


Figure 101. DBSCAN result on 10 samples considering P3 and HS

The 10 samples considering P3 and HS were clustered into 4 noise points and a single cluster consists of 6 visual scanpath sequence data points. In Figure 102, the overall allocation of data points is that each data point is scattered somewhat equally compared to other DBSCAN results even the samples 2, 3, 4, 7, 8, and 9 were clustered into the same group. The samples are demonstrated as scanpath sequences in Table 30.

Table 30. DBSCAN result of 10 samples considering P3 and HS in scanpath sequences

	Participant 3, Hold Short
sample 1	TSTGAG
sample 2	GATSTASA
sample 3	SGSGHTSBGHAG
sample 4	AGTHTSTGAG
sample 5	GSTGCSCBGSTSGH
sample 6	STHTFTGAS
sample 7	TSTBHGA
sample 8	GTASCBSAG
sample 9	ABGCSTSTGAGA
sample 10	GAGCGATGFTAGS

The samples 8, 9 and 3, 4 were selected for the further visualization and shown in the following Figures. Here it can be noted that the Participant 3 was using circular-like scanning pattern on the sample 8 (starting from G then T → A → S → C (Clockwise movement among the AOIs). The counterclockwise movement (A → B → G → C) was found in the sample 9 followed by a couple of linear back-and-forth saccade movements—between S, T, and A, G—on the sample 9.

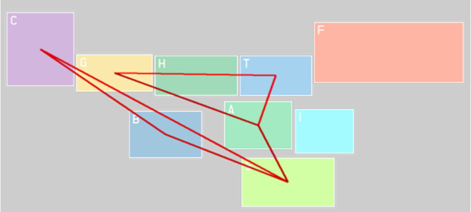


Figure 102. Sample 8 of P3 and HS

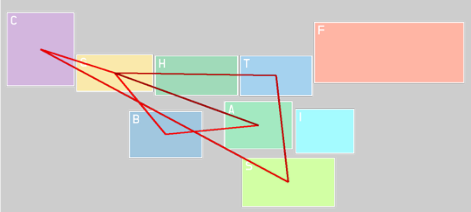


Figure 103. Sample 9 of P3 and HS

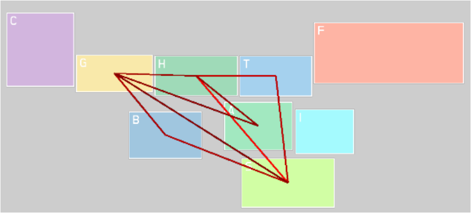


Figure 104. Sample 3 of P3 and HS

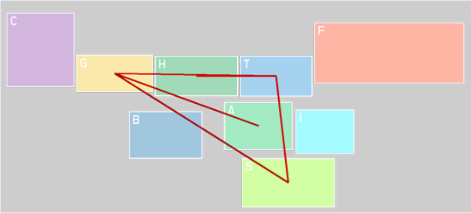


Figure 105. Sample 4 of P3 and HS

The data points that are assigned as core points during DBSCAN are the sample 4, 7, and 8. The sample 4 can be defined as the representative scanpath for the 10 samples considering P3 and HS by comparing the dissimilarity (i.e. DTW distance) between the core points and the other points in the same scanning pattern cluster '0'.

4) Participant 4 (P4) and Hold Short (HS)

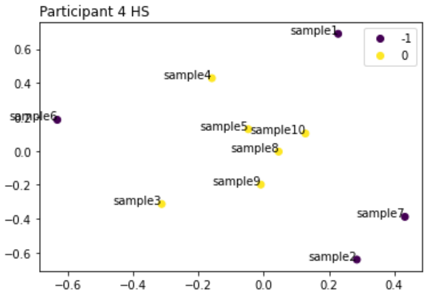


Figure 106. DBSCAN result on 10 samples considering P4 and HS

The 10 samples of Participant 4 giving HS clearance were clustered and shown in Figure 107. Here, the samples that are grouped into the same clusters are densely allocated in the middle of 2-dimensional MDS space. Scanpath sequences of 10 samples were shown in Table 31 below.

Table 31. DBSCAN result of 10 samples considering P4 and HS in scanpath sequences

Participant 4, Hold Short	
sample 1	BFTGCGBAG
sample 2	HBGHGHTAFAHTHGAG
sample 3	AHTHGAHAGAT
sample 4	GBAGCGABAG
sample 5	HGBGCBGATAGAGA
sample 6	GATHGAGBGBGAGAT
sample 7	HBHGHGAGTHTATHGAGH
sample 8	AHGBHTGAG
sample 9	ATHGAHTGAGB
sample 10	BGATBAGAGH

The four scanpath sequences in the center of the space—the samples 5, 8, 9, and 10—were further visualized to have a better analysis on the result. In Figure 110, it was captured that the Participant 4 using counterclockwise scanning pattern at the beginning of the sequence, starting from A → H →G →B → H, then the linear scanning movement was followed on the AOIs T, G, and A. The mixture of these circular—clockwise, and counterclockwise—and linear scanning patterns was dominant on other samples as well.

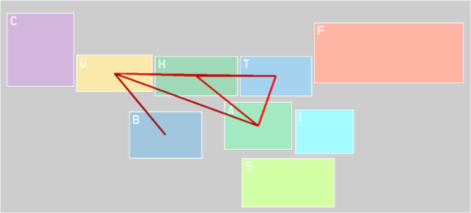


Figure 107. Sample 9 of P4 and HS

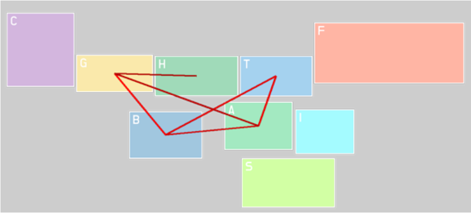


Figure 108. Sample 10 of P4 and HS

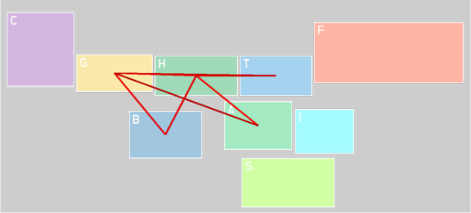


Figure 109. Sample 8 of P4 and HS

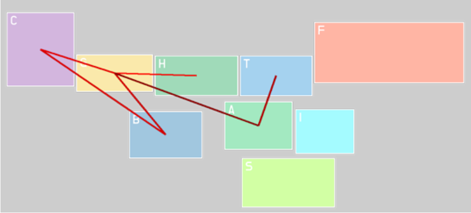


Figure 110. Sample 5 of P4 and HS

The core points that were determined by DBSCAN were the samples 5, 8, and 9. The representative scanpath among the 10 samples considering P4 and HS could be identified as the sample 8 by comparing the DTW distance between the core point and the other points in the cluster.

3.6.4 10 Samples Considering Single Participant, Single Clearance (Hold Short)

1) Participant 1 (P1) and Line Up and Wait (LUAW)

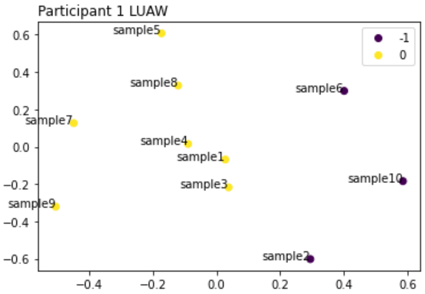


Figure 111. DBSCAN result on 10 samples considering P1 and LUAW

Now the 10 samples considering P1 and LUAW were lied to the 2-dimensional MDS plane and clustered based on the scanning patterns used, and the result was shown in Figure 111. The samples 2, 6, and 10 were defined as noise points and did not appointed into a cluster, while the rest of the data points in this configuration (i.e. 10 samples considering P1 and LUAW) were assigned to the single cluster. The scanpaths were described as scanpath sequences and colored based on their cluster results in Table 32.

Table 32. DBSCAN result on 10 samples considering P4 and HS in scanpath sequences

Participant 1, Line Up and Wait	
sample 1	GAGTHGATSTG
sample 2	AHBSBATGTGTG
sample 3	BTGCBFGAT
sample 4	BTAGAGHTGSTG
sample 5	GHAGASATGHTBTHASTG
sample 6	HASTGTHABGHAGAG
sample 7	GABGBTGBTHGTSH
sample 8	GATGABAGASGHTSTG
sample 9	BTGTBHGBTHGHTST
sample 10	ASAGTGHGAGTGSTSG

The eye movements that were captured for the samples 4 and 8 were visualized in Figure 112 and 113. The overall layout of saccade movements in the samples and 4 and 8 was almost identical to each other, except that the sample 4 has the eye movement between B and T, while the sample 8 does not and have the eye movement between B and A instead, which is not included in the sample 4. Both samples demonstrated here show that the controller was scanning the airport surface based on the AOI A (ASDE)—which is the radar display that visualize the ground surface area of the airport—and the AOI S (Flightstrips), the paper tags that are used for keeping track of the flight.

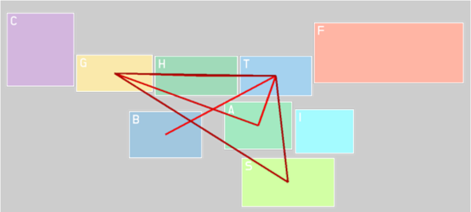


Figure 112. Sample 4 of P1 and LUAW

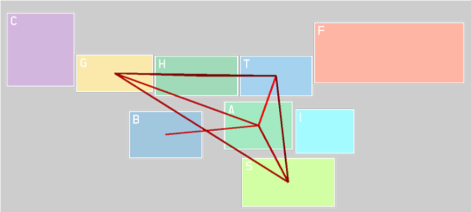


Figure 113. Sample 8 of P1 and LUAW

The core points defined during the DBSCAN are the samples 1, 4, 7, and 8. The sample 1 could be selected as the representative scanpath for this configuration (10 samples considering P1 and LUAW) by comparing the DTW distance between the core points and other data points within the cluster. The sample 1's eye movements are further visualized below in Figure 114. The sample 1 lacks a few eye fixations and movements from the two examples demonstrated above in the Figures 112 and 113.

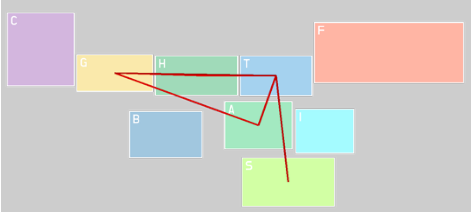


Figure 114. Sample 1 of P1 and LUAW

2) Participant 2 (P2) and Line Up and Wait (LUAW)

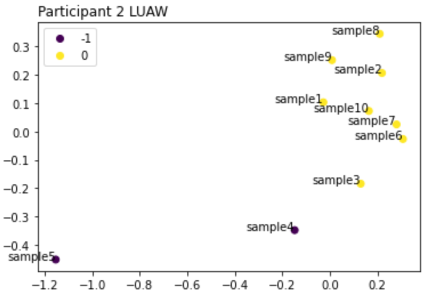


Figure 115. DBSCAN result on 10 samples considering P2 and LUAW

The 10 samples considering P2 and LUAW were clustered as shown in Figure 114. The samples 4 and 5 were categorized as outliers during the cluster phase. One noticeable finding from the Figure 115 is that the sample 5 has a quire far distance or position from rest of the data points in the same configuration (i.e. 10 samples considering P2 and LUAW) which might affected the determining process of eps (ϵ)—one of the parameters for the DBSCAN algorithm. The scanpath sequences were highlighted to match their clustering result and formatted into scanpath sequences in Table 33 below.

Table 33. DBSCAN result on 10 samples considering P2 and LUAW

Participant 2, Line Up and Wait	
sample 1	**ASABGTATATAT**
sample 2	**TGBGAGBTATAT**
sample 3	**THATHABGA**
sample 4	**GHBSASTSABSTC**
sample 5	**ATGBTABCGTHGCSABCGASATSATG**
sample 6	**FTGTATBIBATG**
sample 7	**ASAGTABTGSTHTHB**
sample 8	**SABTABATGBHBTSATH**
sample 9	**AHTAGTABGBGATAT**
sample 10	**ABATGSASABTABATG**

In Table 33, it is noticeable that the sample 5 has significantly longer length (i.e. the number of AOI letters in the sample) than other scanpath demonstrated in the table, which backs the finding from other configuration that DTW distance might be highly affected by the length of sequence. This finding could lead to a follow up research topic applying different types of DTW variant algorithm to have the dissimilarity metric between scanpaths less affected by the length of the scanpaths (i.e. number of AOI letters).

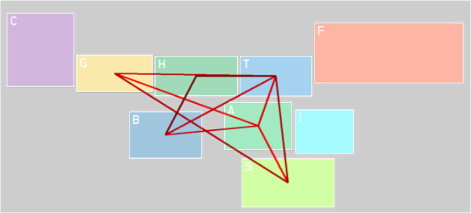


Figure 116. Sample 7 of P2 and LUAW

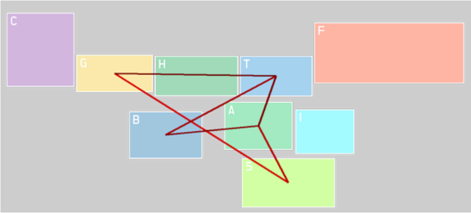


Figure 117. Sample 10 of P2 and LUAW

In Figures 116 and 117, the eye movements that are captured within samples 7 and 10 were visualized to be compared further. Both scanpaths share most of the AOIs where the controller had the eye fixations, and the overall layout of the eye movements resembles each other, but the sample 10 lacks a few eye movements from the sample 7. Every data point in the cluster '0' is defined as a core point by the DBSCAN method. The sample 3 could be identified as the representative scanpath once the DTW distance was compared between the core point and the other points in the same cluster. The representative scanpath is shown in Figure 118.

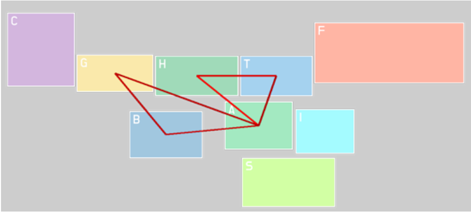


Figure 118. Sample 3 of P2 and LUAW

3) Participant 3 (P3) and Line Up and Wait (LUAW)

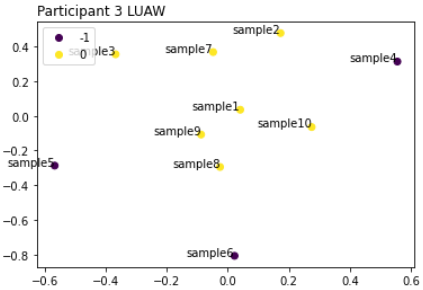


Figure 119. DBSCAN result on 10 samples considering P3 and LUAW

The 10 samples considering P3 and LUAW were cluster as shown in the Figure 119. The cluster was grouped in the top middle of the 2-dimensional MDS plane, and the noise points were located around the cluster toward each direction. The scanpaths were demonstrated as scanpath sequences in Table 34.

Table 34. DBSCAN result on 10 samples considering P3 and LUAW

Participant 3, Line Up and Wait	
sample 1	THGASTHTST
sample 2	BGTHSG
sample 3	SAHAGTHTST
sample 4	HSGTSABGCSTSTG
sample 5	TITGSGTHGTABSTHT
sample 6	HGCGHGSGSGASGATS
sample 7	SGSASCBSTGS
sample 8	GSBSHTGSTS
sample 9	SGTGATH
sample 10	GASATGATASTH

Here, in the Figures 120 and 121, the scanpath samples that have similar overall layout are compared side by side. Even though the sample 1 in Figure 120 lacks the eye movement between H and A, they are sharing the overall geometrical shape of the eye movements. But it can be found that the controller was using circular scanning pattern on the sample 1 (i.e. T → H →G → A → S → T), while the sample 3 is consist of linear or back-and-forth eye movements between T and S; T and H; A and H for the most of the time. In other words, the samples 1 and 3 might be the totally different scanpath or scanpaths that different scanning patterns were applied, even though their overall layout seems identical during the visual examining process.

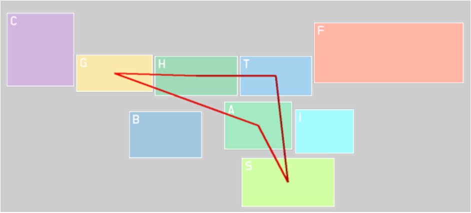


Figure 120. Sample 1 of P3 and LUAW

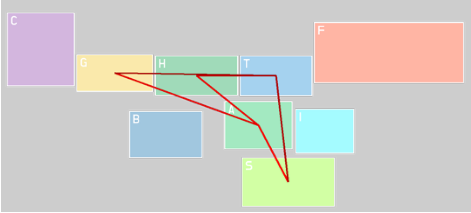


Figure 121. Sample 3 of P3 and LUAW

The core points selected by DBSCAN among the 10 samples are the sample 1, 7, 8, 9, and 10. The sample 9 has the least amount of dissimilarity among other data points grouped into the same cluster. Therefore, the representative scanpath for 10 samples considering P3 and LUAW could be determined as the sample 9 as shown in Figure 122.

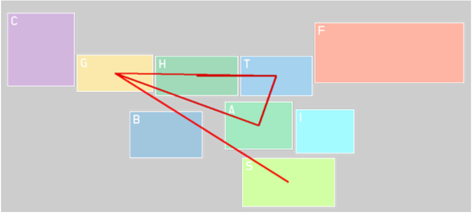


Figure 122. Sample 9 of P3 and LUAW

4) Participant 4 (P4) and Line Up and Wait (LUAW)

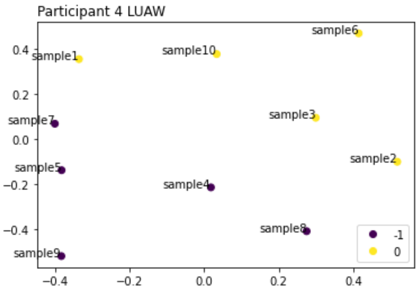


Figure 123. DBSCAN result on 10 samples considering P4 and LUAW

The 10 samples considering P4 and LUAW were clustered and the result was shown in the plot above (see Figure 123). Interestingly, the scanpath samples were divided into two— 5 data points grouped into a cluster vs. 5 data points labeled as outliers. The distance between each data point is not varying as much as it was on other configurations provided in previous sections. The scanpath samples were demonstrated as scanpath sequences in Table 35.

Table 35. DBSCAN result on 10 samples considering P4 and LUAW in scanpath sequences

Participant 4, Line Up and Wait	
sample 1	**AHTAHABA**
sample 2	**GHATSATA**
sample 3	**ATHGACGHTAT**
sample 4	SABAHATCGATG
sample 5	TITHGTBTGAT
sample 6	**ATHTHGBSATHGATA**
sample 7	AGSTSTGCATATH
sample 8	BATHGBGTHTHGATA
sample 9	GBTATHGAH
sample 10	**AFHGBHGA**

The core points determined during the clustering phase were the samples 3 and 10, which are visualized in Figure 124 and 125. By comparing the DTW distance between the core points and other scanpath sequences in the cluster, the sample 10 could be determined as the representative scanpath for this configuration.

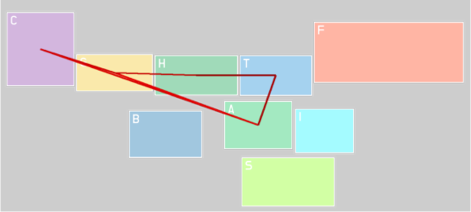


Figure 124. Sample 3 of P4 and LUAW

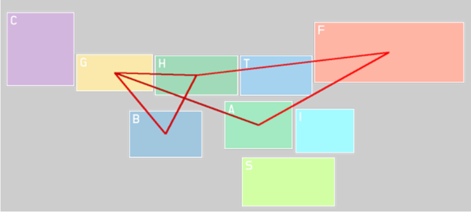


Figure 125. Sample 10 of P4 and LUAW

3.6.5 40 Samples Considering All Participants and Single Clearance (CTL)

Throughout Chapter 3.6.5 to 3.6.8, the dissimilarity between scanpath sequences—40 samples considering all participants and a single clearance—were calculated to lie them on a 2-dimensional space, and then clustered by scanning pattern using DBSCAN. In Chapter 3.6.5 40 samples considering CTL over all participants were interpreted.

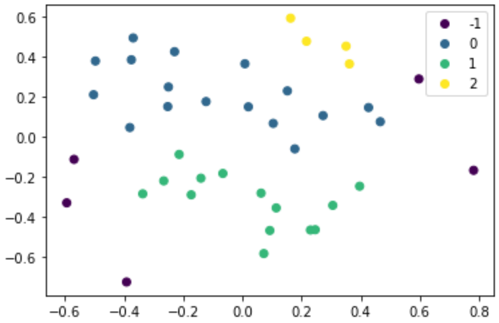


Figure 126. DBSCAN result of 40 samples considering all participants and CTL

In Figure 132, it is shown that the 40 samples were clustered into 3 scanning pattern clusters, ‘0’, ‘1’, and ‘2’, with 5 low-density points (i.e. noise points). The clear separation of clusters can be found among the three clusters. The Silhouette score of the clustering result was computed as 0.21, which supports the result shown in Figure 132. The 40 samples were formatted in scanpath sequences and highlighted as the clustering result in Table 36.

Table 36. DBSCAN result of 40 samples considering all Participants and CTL in scanpath sequences

Sample	Participant 1	Participant 2	Participant 3	Participant 4
1	GATASGTBGBHG	THBGASBTHTHATGBATBH	GSBGTBSTS	BATFTHTHGBHTH
2	TGBATBTGAGTABTI	THASBTGTBT	STHTHTSBI	TATHTHGCGB
3	GHAHGHBHSGASB	ASATBTGABT	ABSHGTSTIT	ATHGATABHG
4	THGATSTGTGAB	ATATATGBSTAB	ASASTHGSTSBSTG	ATAHTBAGABGCGHTATB
5	BHGBTHGHTSTGABT	THBGBSATBTBTI	SGTABIHGHSB	HGHTFTHTHTHGBABHT
6	TGAGTABH	GCSABCGASATSATGTHABTB	HGSTBSB	TFTAHGBT
7	TATGTHTBT	BTIGCGABT	HTHGHGHTSBI	GTATATHGCGCGHTABT
8	ASGHTSTGAGBGT	SABGHTHGTSBTG	BAGCSTGSBT	TSBTBGHTBA
9	GATASGTBGBHG	TGCTATATBTH	STSGSBTSBTG	GHGCGHBATBG
10	TGBATBTGAGTABTI	ATATGCGTATHBTG	SATHTABS	CGHTATHTATBHG

One noticeable finding is that the scanning pattern clustering results don’t coincide with the participants label here. In detail, all the participants—P1, P2, P3, and P4—are found to using the scanning pattern ‘0’ and ‘1’ throughout the 10 samples collected, except for the scanning pattern ‘2’, which is only used by the Participants 2 and 3.

3.6.6 40 Samples Considering All Participants and Single Clearance (C2TO)

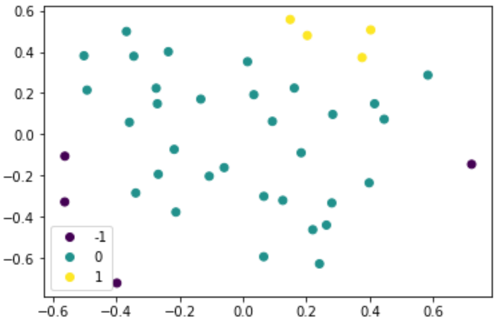


Figure 127. DBSCAN result of 40 samples considering all participants and C2TO

In Chapter 3.6.6, the 40 samples considering C2TO over all participants were lied to a 2-dimensional MDS plane and clustered using DBSCAN algorithm. Interestingly, the overall layout of the scanpath sequence data points (i.e. MDS results) in Figures 132, and 133 are alike to each other. But here in the Figure 133, the DBSCAN method categorized the chunks of the data points in the middle as an entire cluster while it was separated into two groups in the Figure 132. It led to the lower Silhouette score of 40 samples considering C2TO when it's compared to that of CTL—Silhouette score for C2TO: 0.21, CTL: 0.24. The scanpath sequences were shown in Table 37, with being highlighted accordingly to the clustering result.

Table 37. DBSCAN result of 40 samples considering all Participants and C2TO in scanpath sequences

Sample	Participant 1	Participant 2	Participant 3	Participant 4
1	GAHAGAGAGTHGA	TGTASATBTHGTSTG	ABAHTAHA	ABAHTAHA
2	HTABHBHGAHBSBATGTG	AGBGABABASAT	BASABGHAT	BASABGHAT
3	GCBTGCBTGA	BCGTAHCTACAB	HAGCGHTAGSTSTG	HAGCGHTAGSTSTG
4	GABAGABHGTBHSHTHG	TABGAHTAHTAHTAG	AGATAHTBAGABC	AGATAHTBAGABC
5	ABCGATGABAGASGHT	BGTHTABASATGBTABCGTHGC	GAGASABAHATCG	GAGASABAHATCG
6	TCATASTHGTATHGT	BASAHGHABTABGA	BTHTFTAHGHGBATHG	BTHTFTAHGHGBATHG
7	GCGAGTHTHGHAGASATGHT	AGTASASABGHTHG	BHBTGATATCG	BHBTGATATCG
8	AGBGSGASAGTGHGAGTGS	ABTHGABAGATHASTG	HTAHGBGTITHGT	HTAHGBGTITHGT
9	HBTHGCBHTHTGAHTHGHTHATHGHT	BTGAGAHTABAGABCG	FAHTHGAGBABATG	FAHTHGAGBABATG
10	STGABAGABGTASTHGT	ATBGSABATGB	AHTBGATHTHGBSA	AHTBGATHTHGBSA

Along with the finding from the previous section that each scanning strategy (or pattern) was shared by every participant when the participants were giving CTL clearance, here, a single scanning strategy—cluster '0'—was shared across the participants during they were giving C2T0 clearances to aircraft. Except that the scanning pattern indicated as '1' was not used by the Participant 1, while the rest of the participants had at least one scanpath sample that is clustered as '1' (see Table 37).

3.6.7 40 Samples Considering All Participants and Single Clearance (HS)

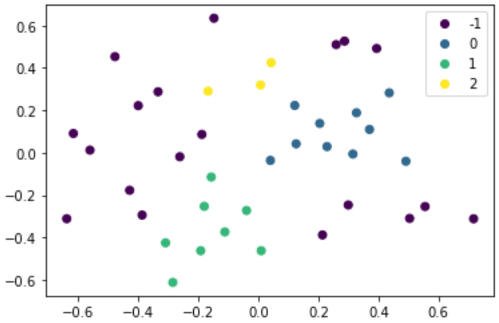


Figure 128. DBSCAN result of 40 samples considering all participants and HS

The 10 samples considering all participants and HS can be lied to a 2-dimensional MDS plane like above in the Figure 130. The MDS result (i.e. how the data points are lied to the 2-dimensional space) of this configuration is unique among other 40-sample configurations considering CTL, C2TO, and LUAW (see Figure 128, 129, 130, and 131). The Silhouette score computed based on the DBSCAN result was the highest among other configurations as 0.50, which had a significant difference compared to others' Silhouette coefficient.

Table 38. DBSCAN result of 40 samples considering all Participants and HS in scanpath sequences

Sample	Participant 1	Participant 2	Participant 3	Participant 4
1	BGBATHABGAHAGAG	ATAGBAB	TSTGAG	BFTGCGBAG
2	BHBGHGAHGBAG	BGAGABABGAG	GATSTASA	HBGHGHTAFAHTHGAG
3	GTGTGSGAG	AHGBTBGAGTG	SGSGHTSBGHAG	AHTHGAHAGAT
4	GACABAGTSTGAGBGH	ABGATHSBTAT	AGTHTSTGAG	GBAGCGABAG
5	GTGAGHGAGAHBAGAG	STSABSTCGBASABGTHT	GSTGCSCBGSTSGH	HGBGCBGATAGAGA
6	AHGHAHGAGBTAGAGH	SAHGHGATABATGAGH	STHTFTGAS	GATHGAGBGBGAGAT
7	HGAHBTHGHGHABGAHG	BHTHIBATAGAT	TSTBHGA	HBHGHGAGTHTATHGAGH
8	HGTGABAG	ATATBAIGAG	GTASCBSAG	AHGBHTGAG
9	TGTHTHBHBAGBGH	ABTABGASAGTA	ABGCSTSTGAGA	ATHGAHTGAGB
10	TATHAHGAGTGAGAG	ABASATGBGAGB	GAGCGATGFTAGS	BGATBAGAGH

Another noticeable finding in this section is that the 7 out of Participant 4's scanpath were identified as outliers (i.e. noise points) as shown in Figure 130 and Table 38. It can be interpreted as in perspective of comparing scanning strategies among four participants, Participant 4's scanning strategy wasn't able to cluster them. But DBSCAN algorithm was able to cluster the samples 3, 4, 5, 8, 9, and 10 of Participant 4 into the same cluster in Chapter 3.6.4. In addition, The scanning pattern labeled as '2' is only used by Participant 3 and 4, while the scanning patterns '0' and '1' were used by every participant while giving Hold Short clearance.

3.6.8 40 Samples Considering All Participants and Single Clearance (LUAW)

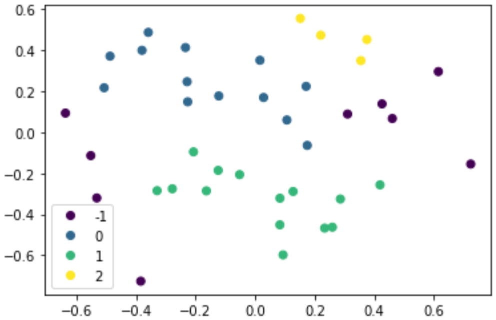


Figure 129. DBSCAN result of 40 samples considering all participants and LUAW

Lastly, the result of 40 samples considering all participants and LUAW is shown in Figure 131. As it was pointed in previous sections, the MDS and DBSCAN results resemble that of 40-sample configuration on CTL, C2TO, where three blocks of visual scanning pattern clusters—clusters ‘0’, ‘1’, and ‘2’—are stacked as the MDS y axis increases, also having the outliers distributed to the left and right side of the cluster blocks.

Table 39. DBSCAN result of 40 samples considering all Participants and LUAW in scanpath sequences

Sample	Participant 1	Participant 2	Participant 3	Participant 4
1	GAGTHGATSTG	ASABGTATATAT	THGASTHTST	AHTAHABA
2	AHBSBATGTGTG	TGBGAGBTATAT	BGTHSG	GHATSATA
3	BTGCBFGAT	THATHABGA	SAHAGTHTST	ATHGACGHTAT
4	BTAGAGHTGSTG	GHBSASTSABSTC	HSGTSABGCSTSTG	SABAHATCGATG
5	GHAGASATGHTBTHASTG	ATGBTABCGTHGCSABCGASATSATG	TITGSGTHGTABSTHT	TITHGTBTGAT
6	HASTGTHABGHAGAG	FTGTATBIBATG	HGCGHGSGSGASGATS	ATHTHGBSATHGATA
7	GABGBTGBTHGTSH	ASAGTABTGSTHTHB	SGSASCBSTGS	AGSTSTGCATATH
8	GATGABAGASGHTSTG	SABTABATGBHBTSATH	GSBSHTGSTS	BATHGBGTHTHGATA
9	BTGTBHGBTHGHTST	AHTAGTABGBGATAT	SGTGATH	GBTATHGAH
10	ASAGTGHGAGTGSTSG	ABATGSASABTABATG	GASATGATASTH	AFHGBHGA

Here, it is shown that the Participant 2 has only one noise point scanpath while Participant 1 has the greatest number of visual scanpaths that have been labeled as noise points in Figure 131 (the number of the outliers is not related to the participants’ performance in tasks). It could be concluded that Participant 2 used the scanning pattern most consistently than other controllers who participated in the experiment. Most of the participants used two scanning pattern strategies ‘0’ and ‘1’, while the participant 3 and 4 used scanning pattern ‘2’ additionally. The scanning pattern ‘2’ was not detected in the sample scanpaths of Participant 1 and 2.

Chapter 4. Discussion

In this study, it was investigated whether it was possible to cluster the visual scanpaths of ATCs active in the field based on their scanning patterns or strategies using DBSCAN algorithm, and the findings of this study was interpreted and discussed throughout the Chapter 3. This result could lead to a systematic approach to help the FAA or the researchers to evaluate the controller's performances on the task by having a better understanding on which patterns or strategies are used on which tasks. For instance, the number of scanning patterns used under certain tasks can be helpful to ATC trainees by introducing them to trainees. One of the advantages in applying this approach–using DBSCAN to cluster the visual scanpath sequences—was that there is no need of predetermining the number of clusters.

Identifying the scanning strategies and the representative scanpaths based on comparing DTW distance between the core points defined during DBSCAN could be suggested as a more objective way to analyze the scanning patterns and get a representative scanpath when it is compared to visual examining with bare eyes. For instance, when 10 sample scanpaths considering P2 and C2TO were visually examined, it was difficult to find the similarity between the samples 4 and 8 since the overall layout of the scanpath sequences doesn't resemble each other. But DTW distance and MDS method were able to lie the samples close to each other on the 2-dimensional MDS plane using the approach that was studied in this research.

Another interesting finding that could be further discussed in this section is that the value of Silhouette score (or coefficient) is varying over the 40 samples considering each clearance (i.e. Silhouette score was computed as 0.50 in Chapter 3.6.7—40 samples on HS—which is significantly higher than other 40 samples configuration considering CTL, C2TO, and LUAW). Also, considering that 3 clearances beside the HS—CTL, C2TO, and LUAW—had similar overall layout of the data points lied on the 2-dimensional MDS data points, one possible way to think of interpreting this result could be that the controllers in MCO airport share a couple of scanning patterns for CTL, C2TO, and LUAW clearances, while they have own distinct scanning strategy for HS. But further studies like interviewing the controllers regarding the HS scanning patterns or qualitative analysis on the scenarios of the experiment should be followed to investigate whether this interpretation is going to be valid or not. If it is true that the controllers are actually sharing standardized visual scanning pattern per airport, then comparing the standardized scanning patterns could be the future research topic of this study.

Lastly, the need of different (or additional) approaches for determining the eps (ϵ) parameter for DBSCAN algorithm (i.e. parameter tuning for DBSCAN) was appealed in this study especially when the small amount of data points were clustered. Throughout this study the eps (ϵ) value was derived from using kth nearest distance plot and the elbow method, which is a heuristic method getting an approximate value for the eps (ϵ) just by looking at the figure and finding the value where the kth distance between data points surges. However, in some cases it was not clear to determine the elbow point. Since the Silhouette score can be defined to evaluate the separation between clusters, using this value as an objective function, the eps (ϵ) could be found (i.e. tuned) at where the Silhouette score is optimized (i.e. maximized).

Chapter 5. Limitations and Future Research

In this chapter, the limitations and following research topics of this study are discussed. One limitation of the study could be that the experiment conducted to collect the visual scanpaths was focused on more naturalistic response of ATCs in field airports. In other words, the eye movements prior to the controller giving clearance which are the response to the clearances that were given at the end of the 30 seconds time frame were not able to be isolated from other factors (i.e. other clearances).

Another limitation is that not the actual visual scanpaths are compared to get a dissimilarity (i.e. DTW distance) between them in this study. The sequences that were used for applying DTW algorithm was generated based on the scanpath sequences, using the center point of the AOIs as the x and y coordinates of scanpaths. This approach enabled reducing the data processing time and enhanced ease of visualizing the scanpath sequences into visual scanpath sequences but applying DTW algorithm on the raw visual scanpath could return more precise and sensitive comparison between the scanpaths.

One of the following research topics from this study is applying other variants of DTW algorithm. Here, the simplest and most straight forward symmetric step pattern was investigated for determining a warping path between two sequences. However, there are multiple other step patterns out there, which has different heuristics to determine the local cost between the sequences and the warping path. Investigating how the outcomes might change as different step patterns were used and even developing a step pattern for specific application of DTW in a certain application (i.e. per task or airport in air traffic control) could be followed.

Another following research topic of this study could be applying other clustering methods different from DBSCAN method on the MDS results (e.g. k means clustering or other machine learning algorithms). In addition, other configuration (i.e. projecting the sequences on a different plane or 3-dimensional) of MDS method could be applied as well since the participants and clearances on the MDS plane do not show any specific patterns.

Lastly, a preliminary study can be suggested to better characterize the visual scanpaths based on the number of AOIs defined, the length of scanpaths, and certain eye movements (i.e. saccades) between the AOIs. In details, in Chapter 2, 4 simple visual scanpaths were identified with 4 AOIs, sequence length of 5, and circular and linear scanning patterns. On account of these simplest and characteristic scanpaths it was able to easily interpret the MDS x and y axis. However, in Chapter 3, the scanpaths had total of 9 AOIs defined, and 30 seconds length (but each sequence had different length according to the number eye fixations on AOIs). Therefore, studying the relationship between the number of AOIs and the length of scanpaths could be helpful on characterizing the scanpaths and give better understanding on visual scanpath clustering and interpretation.

References

Carroll, J. D., & Arabie, P. (1998). Multidimensional scaling. *Measurement, judgment and decision making*, 179-250.

Çelik, M., Dadaşer-Çelik, F., & Dokuz, A. Ş. (2011, June). Anomaly detection in temperature data using DBSCAN algorithm. In *2011 international symposium on innovations in intelligent systems and applications* (pp. 91-95). IEEE.

Corradini, A. (2001). Dynamic time warping for off-line recognition of a small gesture vocabulary. *in RATFG-RTS '01: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01).* Washington, DC, USA: IEEE Computer Society.

Cox, M. A., & Cox, T. F. (2008). Multidimensional scaling. In *Handbook of data visualization* (pp. 315-347). Springer, Berlin, Heidelberg.

Efrat, A., Fan, Q., and Venkatasubramanian, S. (2007). Curve matching, time warping, and light fields: New algorithms for computing similarity between curves. *J. Math. Imaging Vis.* vol. 27, no. 3, pp. 203–216.

Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *kdd* (Vol. 96, No. 34, pp. 226-231).

J. Stat. Soft., doi:10.18637/jss.v031.i07.

Jeon, J. (2018). Analysis of Eye Tracking Data to Measure Situational Awareness in Offshore Drilling Operations.

Kang, Z., and Landry, S.J. (2014). Using scanpaths as a learning method for a conflict detection task of multiple target tracking. *Human Factors: The Journal of Human Factors and Ergonomics Society*. 56 (1), 1150-1162.

Kang, Z., Bass, E.J., and Lee, D.W. (2014). Air traffic controllers' visual scanning, aircraft selection, and comparison strategies in support of conflict detection. *In Proceedings of the 58th Annual Meeting of the Human Factors and Ergonomics Society*. Chicago, IL, 68 (1), pp. 77- 81.

Kang, Z., Jeon, J., & Salehi, S. (2020). Eye tracking data analytics in virtual reality training: Application in Deepwater Horizon oil drilling operation. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 64, No. 1, pp. 821-825). Sage CA: Los Angeles, CA: SAGE Publications.

Ma, R., & Angryk, R. (2017). Distance and density clustering for time series data. In *2017 IEEE international conference on data mining workshops (ICDMW)* (pp. 25-32). IEEE.

Muller, M., Mattes, H., and Kurth, F. (n.d.). An efficient multiscale approach to audio synchronization.

Myers, C., Rabiner, L., and Rosenberg, A. (1980) Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing]*. vol. 28, no. 6, pp. 623–635.

Niennattrakul, V., and Ratanamahatana, C. A. (2007). On clustering multimedia time series data using k-means and dynamic time warping. *in Multimedia and Ubiquitous Engineering.* International Conference, pp. 733–738.

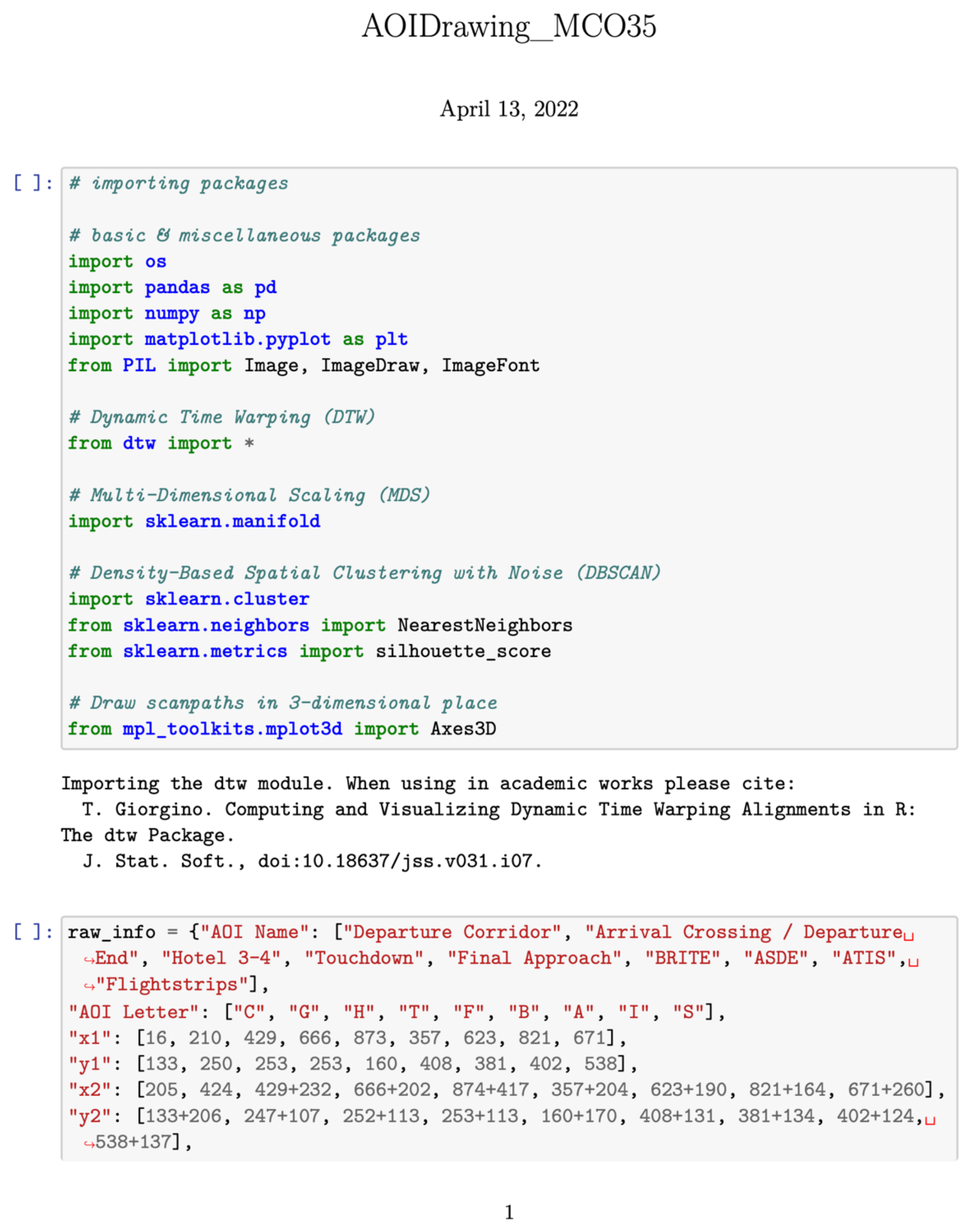
Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech, and Signal Processing.* 19 IEEE Transactions on, vol. 26, no. 1, pp. 43–49.

Salehi, S., Kiran, R., Jeon, J., and Kang, Z. (2018). Developing a cross-disciplinary, scenario-based training approach integrated with eye tracking data collection to enhance situational awareness in offshore oil and gas operations. *Journal of Loss Prevention in the Process Industries*, 56, 78-94.

T. Giorgino. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package.

Tappert, C. C., Suen, C. Y., and Wakahra, T. (1990). The state of the art in online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 12(8), 787-808.

Appendix A. python code for drawing AOIs and eye movements



AOIDrawing_MCO35

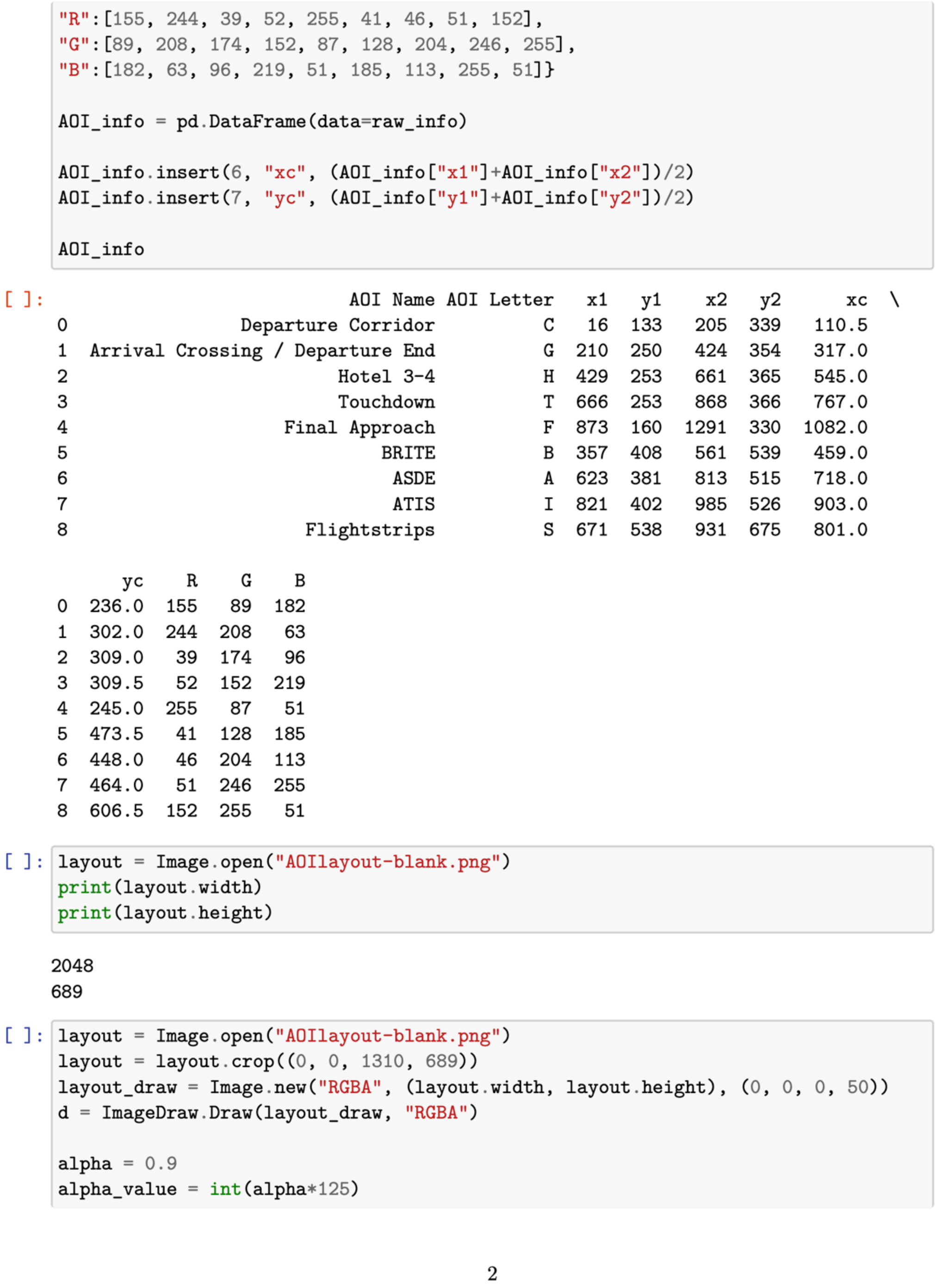
April 13, 2022

```
[ ]: # importing packages

     # basic & miscellaneous packages
     import os
     import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     from PIL import Image, ImageDraw, ImageFont

     # Dynamic Time Warping (DTW)
     from dtw import *

     # Multi-Dimensional Scaling (MDS)
     import sklearn.manifold

     # Density-Based Spatial Clustering with Noise (DBSCAN)
     import sklearn.cluster
     from sklearn.neighbors import NearestNeighbors
     from sklearn.metrics import silhouette_score

     # Draw scanpaths in 3-dimensional place
     from mpl_toolkits.mplot3d import Axes3D
```

```
Importing the dtw module. When using in academic works please cite:
  T. Giorgino. Computing and Visualizing Dynamic Time Warping Alignments in R:
The dtw Package.
  J. Stat. Soft., doi:10.18637/jss.v031.i07.
```

```
[ ]: raw_info = {"AOI Name": ["Departure Corridor", "Arrival Crossing / Departure␣
      ↪End", "Hotel 3-4", "Touchdown", "Final Approach", "BRITE", "ASDE", "ATIS",␣
      ↪"Flightstrips"],
     "AOI Letter": ["C", "G", "H", "T", "F", "B", "A", "I", "S"],
     "x1": [16, 210, 429, 666, 873, 357, 623, 821, 671],
     "y1": [133, 250, 253, 253, 160, 408, 381, 402, 538],
     "x2": [205, 424, 429+232, 666+202, 874+417, 357+204, 623+190, 821+164, 671+260],
     "y2": [133+206, 247+107, 252+113, 253+113, 160+170, 408+131, 381+134, 402+124,␣
      ↪538+137],
```

1



```
"R":[155, 244, 39, 52, 255, 41, 46, 51, 152],
"G":[89, 208, 174, 152, 87, 128, 204, 246, 255],
"B":[182, 63, 96, 219, 51, 185, 113, 255, 51]}

AOI_info = pd.DataFrame(data=raw_info)

AOI_info.insert(6, "xc", (AOI_info["x1"]+AOI_info["x2"])/2)
AOI_info.insert(7, "yc", (AOI_info["y1"]+AOI_info["y2"])/2)

AOI_info
```

[ ]:

```
                           AOI Name AOI Letter   x1   y1    x2   y2      xc  \
0                Departure Corridor          C   16  133   205  339   110.5
1  Arrival Crossing / Departure End          G  210  250   424  354   317.0
2                         Hotel 3-4          H  429  253   661  365   545.0
3                         Touchdown          T  666  253   868  366   767.0
4                    Final Approach          F  873  160  1291  330  1082.0
5                             BRITE          B  357  408   561  539   459.0
6                              ASDE          A  623  381   813  515   718.0
7                              ATIS          I  821  402   985  526   903.0
8                      Flightstrips          S  671  538   931  675   801.0

      yc    R    G    B
0  236.0  155   89  182
1  302.0  244  208   63
2  309.0   39  174   96
3  309.5   52  152  219
4  245.0  255   87   51
5  473.5   41  128  185
6  448.0   46  204  113
7  464.0   51  246  255
8  606.5  152  255   51
```

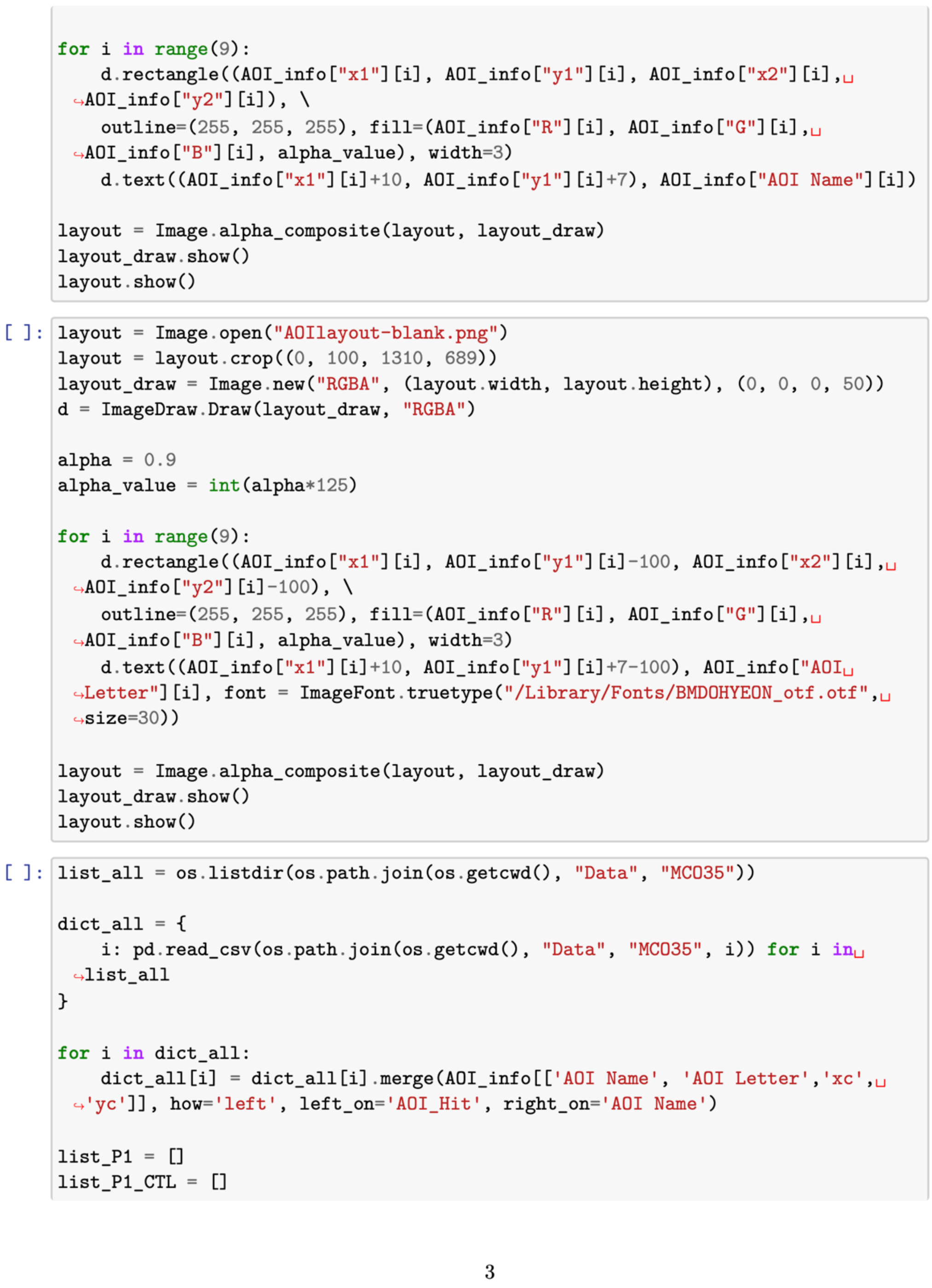
[ ]:

```
layout = Image.open("AOIlayout-blank.png")
print(layout.width)
print(layout.height)
```

```
2048
689
```

[ ]:

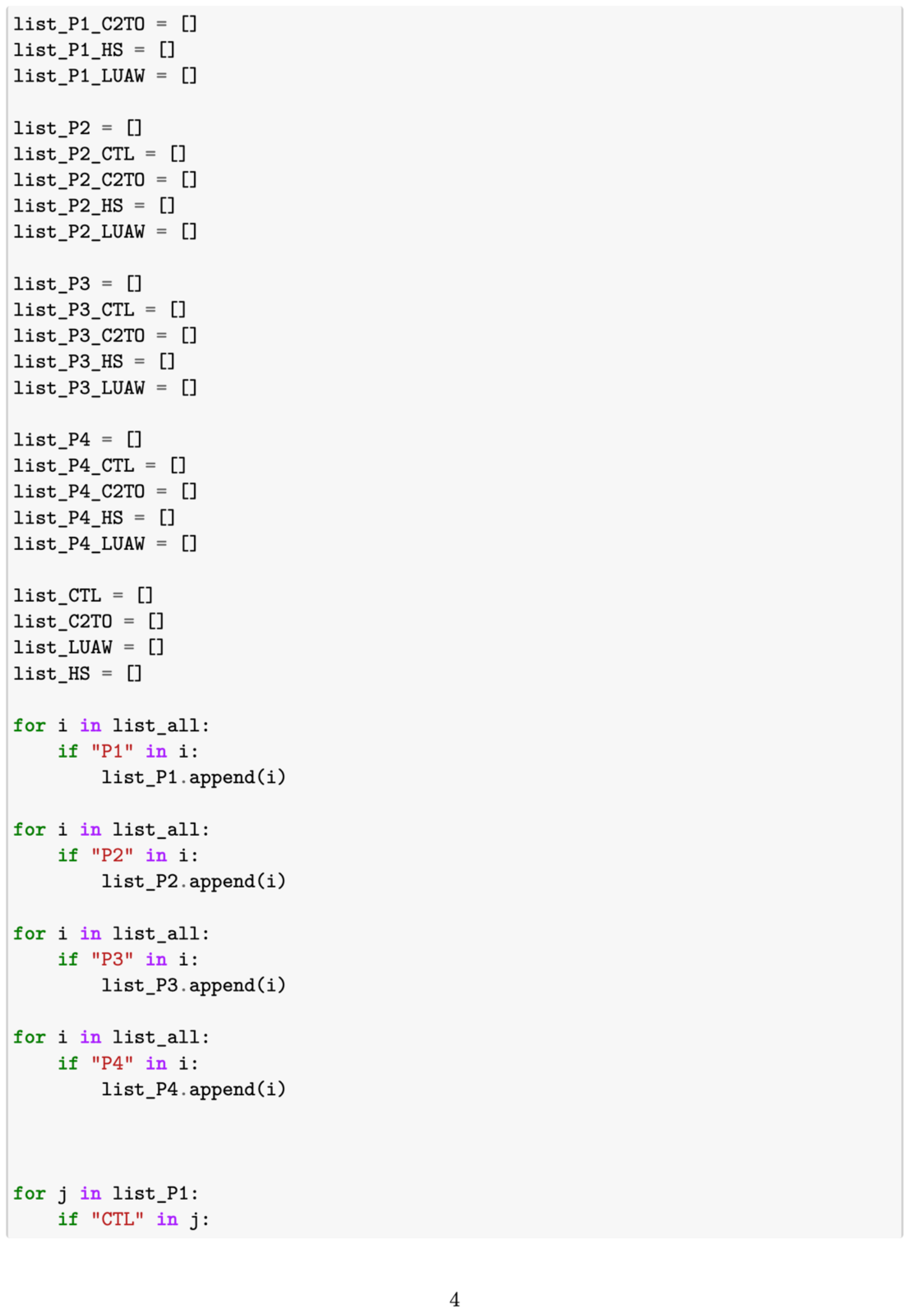
```
layout = Image.open("AOIlayout-blank.png")
layout = layout.crop((0, 0, 1310, 689))
layout_draw = Image.new("RGBA", (layout.width, layout.height), (0, 0, 0, 50))
d = ImageDraw.Draw(layout_draw, "RGBA")

alpha = 0.9
alpha_value = int(alpha*125)
```



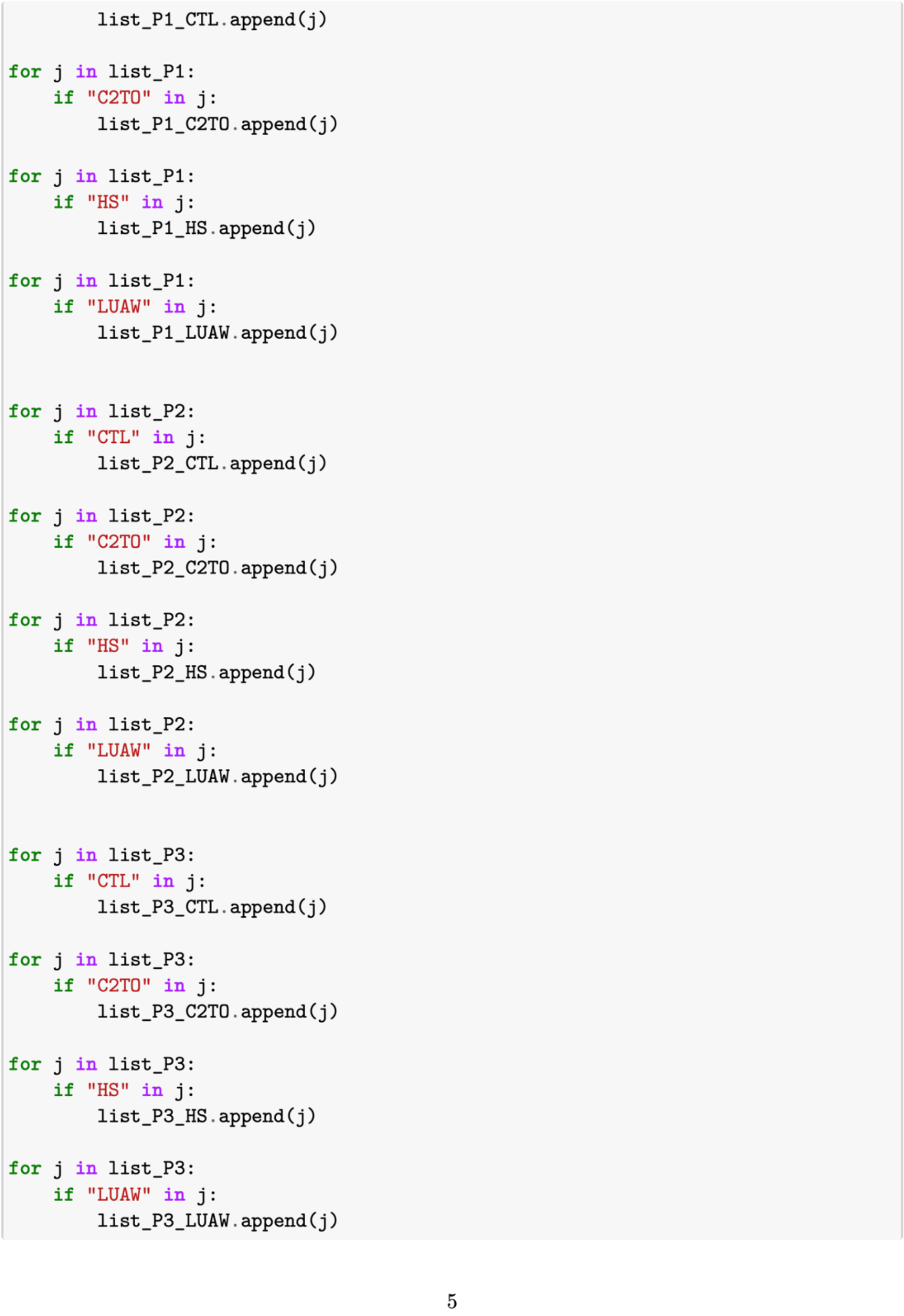
```
for i in range(9):
    d.rectangle((AOI_info["x1"][i], AOI_info["y1"][i], AOI_info["x2"][i],␣
 ↪AOI_info["y2"][i]), \
    outline=(255, 255, 255), fill=(AOI_info["R"][i], AOI_info["G"][i],␣
 ↪AOI_info["B"][i], alpha_value), width=3)
    d.text((AOI_info["x1"][i]+10, AOI_info["y1"][i]+7), AOI_info["AOI Name"][i])

layout = Image.alpha_composite(layout, layout_draw)
layout_draw.show()
layout.show()
```

```
[ ]: layout = Image.open("AOIlayout-blank.png")
     layout = layout.crop((0, 100, 1310, 689))
     layout_draw = Image.new("RGBA", (layout.width, layout.height), (0, 0, 0, 50))
     d = ImageDraw.Draw(layout_draw, "RGBA")

     alpha = 0.9
     alpha_value = int(alpha*125)

     for i in range(9):
         d.rectangle((AOI_info["x1"][i], AOI_info["y1"][i]-100, AOI_info["x2"][i],␣
      ↪AOI_info["y2"][i]-100), \
         outline=(255, 255, 255), fill=(AOI_info["R"][i], AOI_info["G"][i],␣
      ↪AOI_info["B"][i], alpha_value), width=3)
         d.text((AOI_info["x1"][i]+10, AOI_info["y1"][i]+7-100), AOI_info["AOI␣
      ↪Letter"][i], font = ImageFont.truetype("/Library/Fonts/BMDOHYEON_otf.otf",␣
      ↪size=30))

     layout = Image.alpha_composite(layout, layout_draw)
     layout_draw.show()
     layout.show()
```

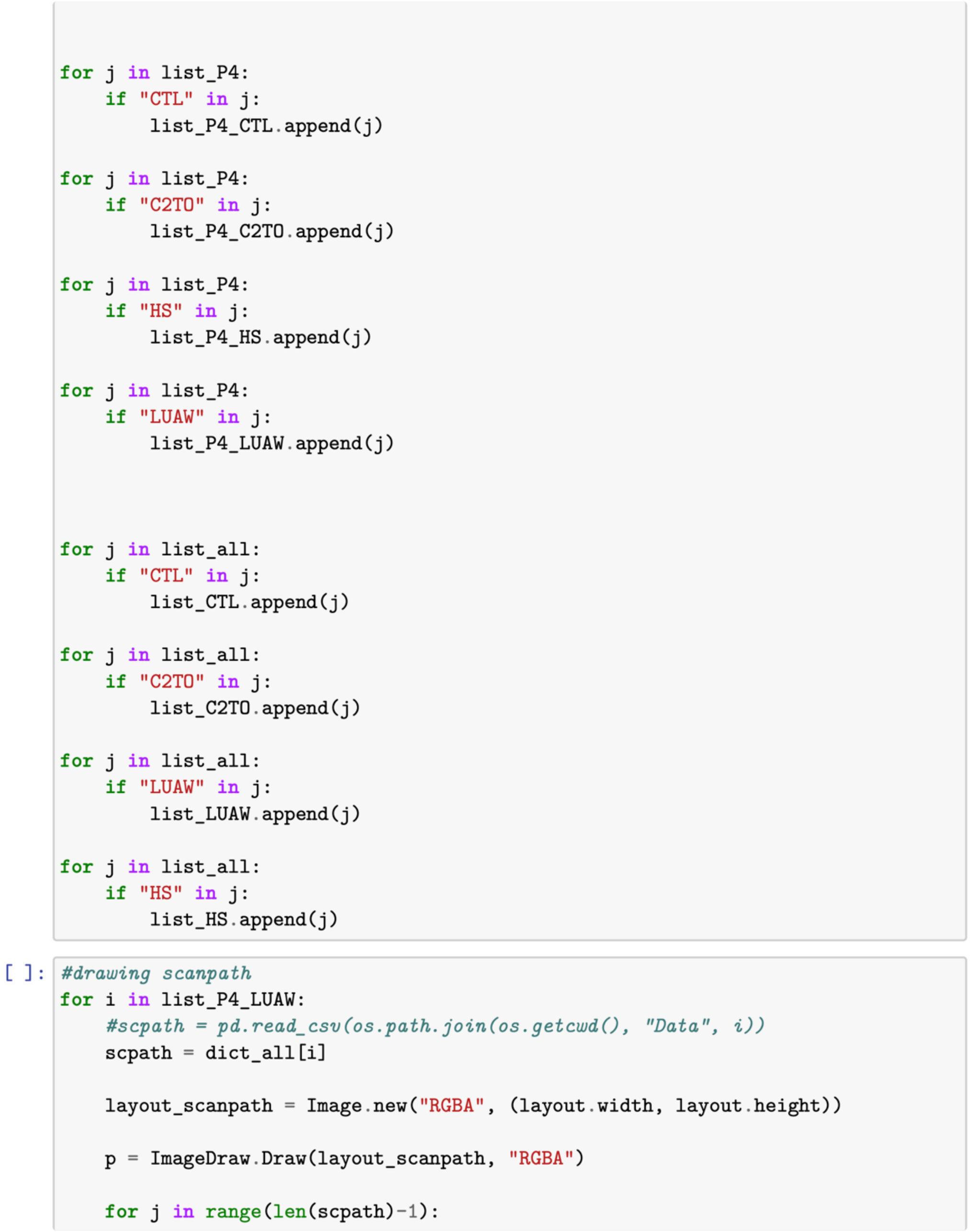
```
[ ]: list_all = os.listdir(os.path.join(os.getcwd(), "Data", "MC035"))

     dict_all = {
         i: pd.read_csv(os.path.join(os.getcwd(), "Data", "MC035", i)) for i in␣
      ↪list_all
     }

     for i in dict_all:
         dict_all[i] = dict_all[i].merge(AOI_info[['AOI Name', 'AOI Letter','xc',␣
      ↪'yc']], how='left', left_on='AOI_Hit', right_on='AOI Name')

     list_P1 = []
     list_P1_CTL = []
```



```
list_P1_C2TO = []
list_P1_HS = []
list_P1_LUAW = []

list_P2 = []
list_P2_CTL = []
list_P2_C2TO = []
list_P2_HS = []
list_P2_LUAW = []

list_P3 = []
list_P3_CTL = []
list_P3_C2TO = []
list_P3_HS = []
list_P3_LUAW = []

list_P4 = []
list_P4_CTL = []
list_P4_C2TO = []
list_P4_HS = []
list_P4_LUAW = []

list_CTL = []
list_C2TO = []
list_LUAW = []
list_HS = []

for i in list_all:
    if "P1" in i:
        list_P1.append(i)

for i in list_all:
    if "P2" in i:
        list_P2.append(i)

for i in list_all:
    if "P3" in i:
        list_P3.append(i)

for i in list_all:
    if "P4" in i:
        list_P4.append(i)



for j in list_P1:
    if "CTL" in j:
```



```
        list_P1_CTL.append(j)

for j in list_P1:
    if "C2TO" in j:
        list_P1_C2TO.append(j)

for j in list_P1:
    if "HS" in j:
        list_P1_HS.append(j)

for j in list_P1:
    if "LUAW" in j:
        list_P1_LUAW.append(j)


for j in list_P2:
    if "CTL" in j:
        list_P2_CTL.append(j)

for j in list_P2:
    if "C2TO" in j:
        list_P2_C2TO.append(j)

for j in list_P2:
    if "HS" in j:
        list_P2_HS.append(j)

for j in list_P2:
    if "LUAW" in j:
        list_P2_LUAW.append(j)


for j in list_P3:
    if "CTL" in j:
        list_P3_CTL.append(j)

for j in list_P3:
    if "C2TO" in j:
        list_P3_C2TO.append(j)

for j in list_P3:
    if "HS" in j:
        list_P3_HS.append(j)

for j in list_P3:
    if "LUAW" in j:
        list_P3_LUAW.append(j)
```



```
for j in list_P4:
    if "CTL" in j:
        list_P4_CTL.append(j)

for j in list_P4:
    if "C2TO" in j:
        list_P4_C2TO.append(j)

for j in list_P4:
    if "HS" in j:
        list_P4_HS.append(j)

for j in list_P4:
    if "LUAW" in j:
        list_P4_LUAW.append(j)



for j in list_all:
    if "CTL" in j:
        list_CTL.append(j)

for j in list_all:
    if "C2TO" in j:
        list_C2TO.append(j)

for j in list_all:
    if "LUAW" in j:
        list_LUAW.append(j)

for j in list_all:
    if "HS" in j:
        list_HS.append(j)
```

[ ]:

```
#drawing scanpath
for i in list_P4_LUAW:
    #scpath = pd.read_csv(os.path.join(os.getcwd(), "Data", i))
    scpath = dict_all[i]

    layout_scanpath = Image.new("RGBA", (layout.width, layout.height))

    p = ImageDraw.Draw(layout_scanpath, "RGBA")

    for j in range(len(scpath)-1):
        p.line([(scpath["xc"][j], scpath["yc"][j]-100), (scpath["xc"][j+1],␣
↪scpath["yc"][j+1]-100)], width=5, fill=(255-10*j,0,0), joint="curve")

    Image.alpha_composite(layout_draw, layout_scanpath).save(os.path.join("AOI␣
↪Drawing",i[:-4]+".png"), format="png")
```

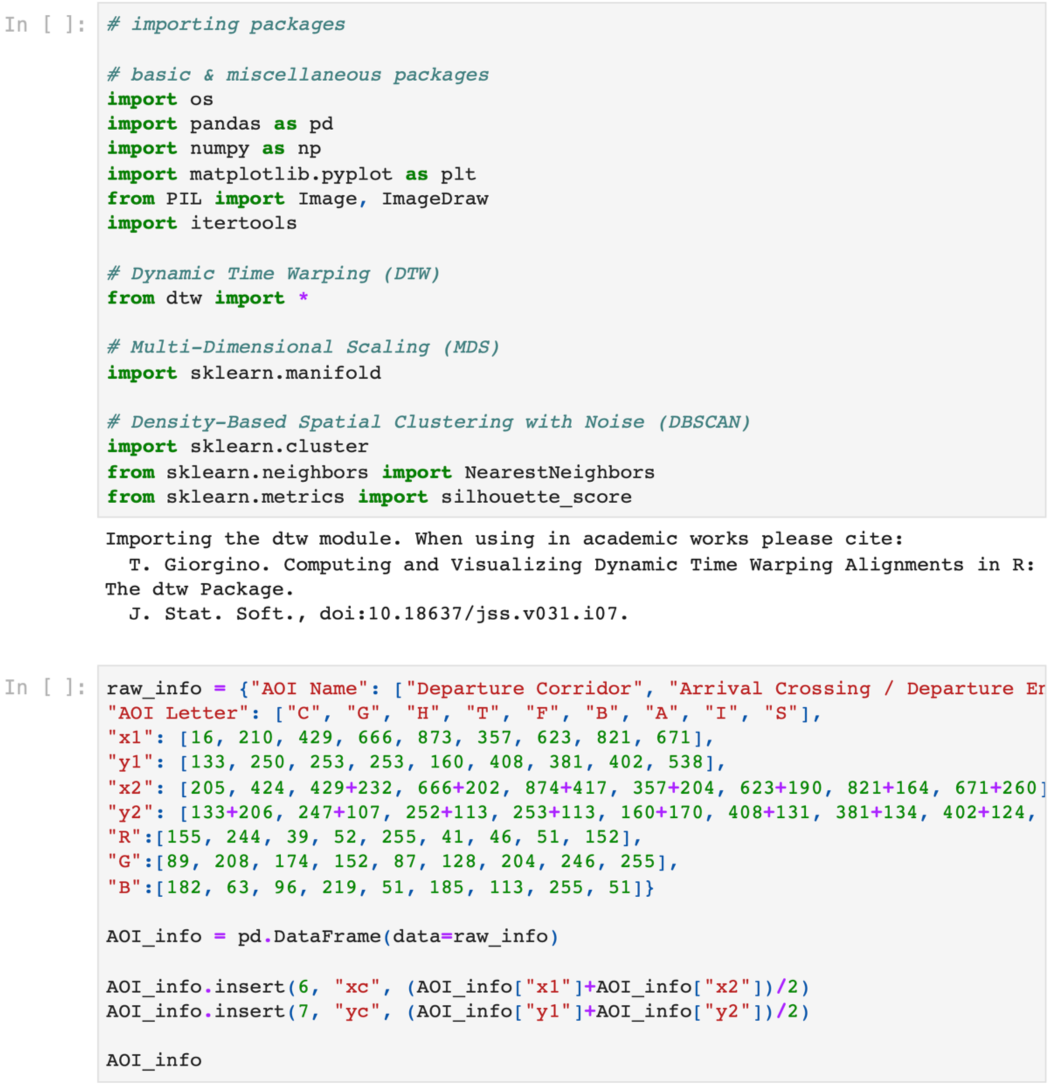
```
[ ]: AOI_info
```

```
[ ]:                         AOI Name AOI Letter   x1   y1    x2   y2      xc  \
     0               Departure Corridor          C   16  133   205  339   110.5
     1  Arrival Crossing / Departure End          G  210  250   424  354   317.0
     2                         Hotel 3-4          H  429  253   661  365   545.0
     3                         Touchdown          T  666  253   868  366   767.0
     4                    Final Approach          F  873  160  1291  330  1082.0
     5                             BRITE          B  357  408   561  539   459.0
     6                              ASDE          A  623  381   813  515   718.0
     7                              ATIS          I  821  402   985  526   903.0
     8                      Flightstrips          S  671  538   931  675   801.0

           yc    R    G    B
     0  236.0  155   89  182
     1  302.0  244  208   63
     2  309.0   39  174   96
     3  309.5   52  152  219
     4  245.0  255   87   51
     5  473.5   41  128  185
     6  448.0   46  204  113
     7  464.0   51  246  255
     8  606.5  152  255   51
```

```
[ ]: list(range(len(dict_all["P1_S0_CTL_30s_before_sample6_before.csv"])))

     dict_all["P1_S0_CTL_30s_before_sample6_before.csv"]
```

```
[ ]:                          AOI_Hit    X    Y  Start_Time  End_Time  \
     0                         Touchdown  757  332      598505    598865
     1  Arrival Crossing / Departure End  297  311      607480    608759
     2                              ASDE  708  451      609219    609439
     3  Arrival Crossing / Departure End  326  318      611058    611218
     4                         Touchdown  818  312      615115    615455
     5                              ASDE  703  454      616814    616954
     6                             BRITE  461  493      619733    620393
     7                         Hotel 3-4  519  299      630986    631106

        Gaze_Time                          AOI Name AOI Letter     xc     yc
     0       6678                         Touchdown          T  767.0  309.5
     1       1279  Arrival Crossing / Departure End          G  317.0  302.0
```

Appendix B. python code for applying DTW, MDS, and DBSCAN



```
In [ ]: # importing packages

        # basic & miscellaneous packages
        import os
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        from PIL import Image, ImageDraw
        import itertools

        # Dynamic Time Warping (DTW)
        from dtw import *

        # Multi-Dimensional Scaling (MDS)
        import sklearn.manifold

        # Density-Based Spatial Clustering with Noise (DBSCAN)
        import sklearn.cluster
        from sklearn.neighbors import NearestNeighbors
        from sklearn.metrics import silhouette_score
```

```
Importing the dtw module. When using in academic works please cite:
  T. Giorgino. Computing and Visualizing Dynamic Time Warping Alignments in R:
The dtw Package.
  J. Stat. Soft., doi:10.18637/jss.v031.i07.
```

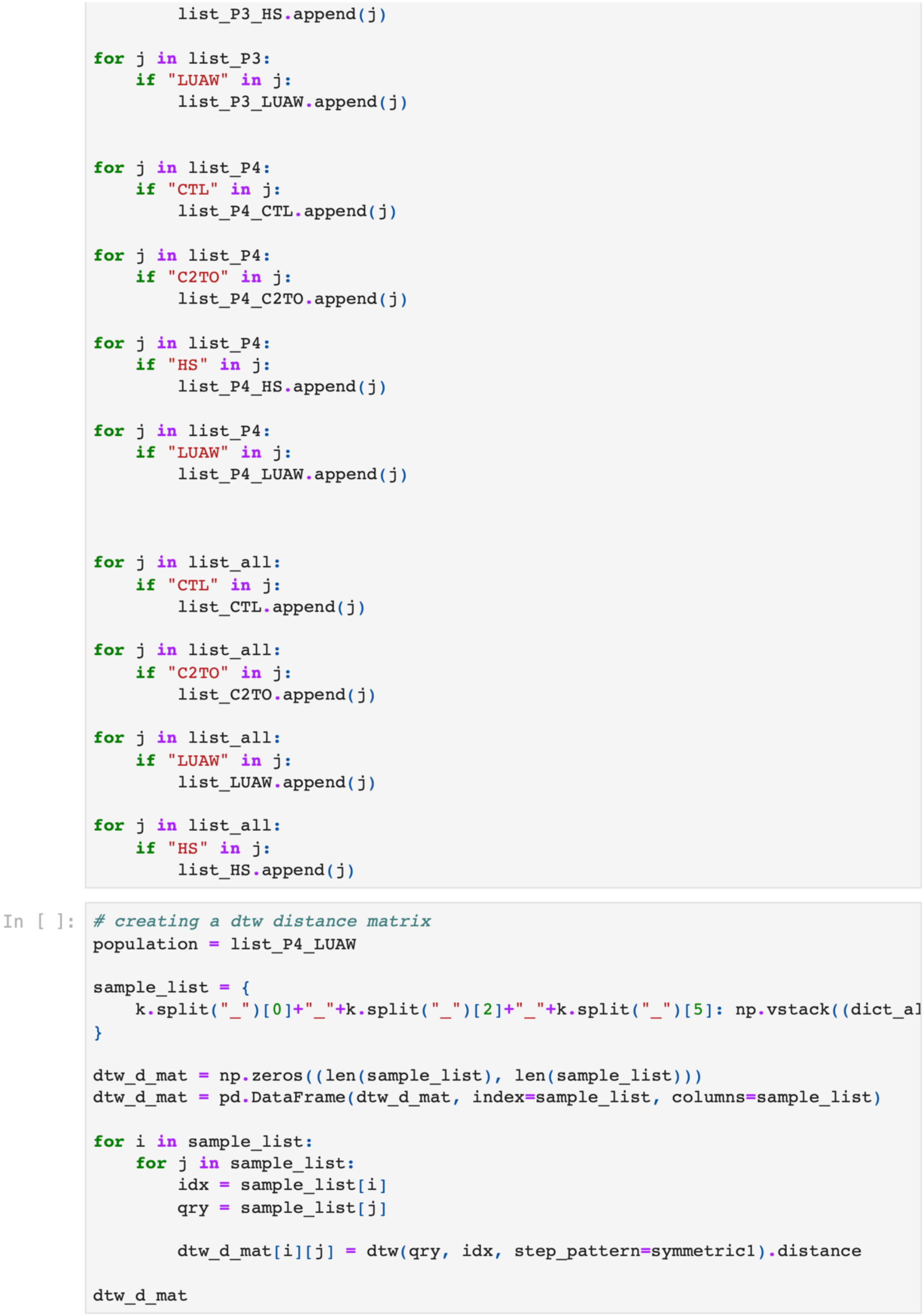
```
In [ ]: raw_info = {"AOI Name": ["Departure Corridor", "Arrival Crossing / Departure Er
        "AOI Letter": ["C", "G", "H", "T", "F", "B", "A", "I", "S"],
        "x1": [16, 210, 429, 666, 873, 357, 623, 821, 671],
        "y1": [133, 250, 253, 253, 160, 408, 381, 402, 538],
        "x2": [205, 424, 429+232, 666+202, 874+417, 357+204, 623+190, 821+164, 671+260]
        "y2": [133+206, 247+107, 252+113, 253+113, 160+170, 408+131, 381+134, 402+124,
        "R":[155, 244, 39, 52, 255, 41, 46, 51, 152],
        "G":[89, 208, 174, 152, 87, 128, 204, 246, 255],
        "B":[182, 63, 96, 219, 51, 185, 113, 255, 51]}

        AOI_info = pd.DataFrame(data=raw_info)

        AOI_info.insert(6, "xc", (AOI_info["x1"]+AOI_info["x2"])/2)
        AOI_info.insert(7, "yc", (AOI_info["y1"]+AOI_info["y2"])/2)

        AOI_info
```

Out[ ]:

	AOI Name	AOI Letter	x1	y1	x2	y2	xc	yc	R	G	B
0	Departure Corridor	C	16	133	205	339	110.5	236.0	155	89	182
1	Arrival Crossing / Departure End	G	210	250	424	354	317.0	302.0	244	208	63
2	Hotel 3-4	H	429	253	661	365	545.0	309.0	39	174	96
3	Touchdown	T	666	253	868	366	767.0	309.5	52	152	219
4	Final Approach	F	873	160	1291	330	1082.0	245.0	255	87	51
5	BRITE	B	357	408	561	539	459.0	473.5	41	128	185
6	ASDE	A	623	381	813	515	718.0	448.0	46	204	113
7	ATIS	I	821	402	985	526	903.0	464.0	51	246	255
8	Flightstrips	S	671	538	931	675	801.0	606.5	152	255	51

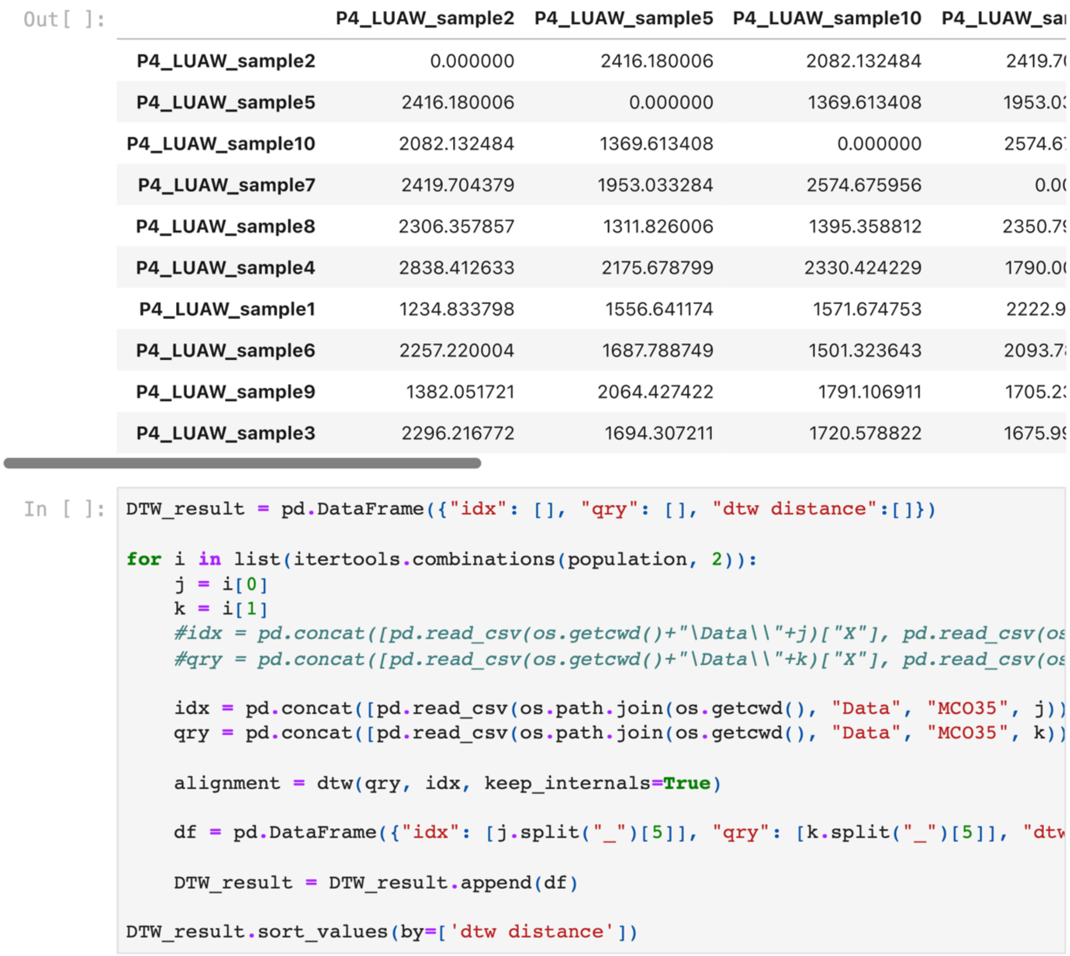
In [ ]:

```
list_all = os.listdir(os.path.join(os.getcwd(), "Data", "MCO35"))

dict_all = {
    i: pd.read_csv(os.path.join(os.getcwd(), "Data", "MCO35", i)) for i in list
}

for i in dict_all:
    dict_all[i] = dict_all[i].merge(AOI_info[['AOI Name', 'AOI Letter','xc', 'y

list_P1 = []
list_P1_CTL = []
list_P1_C2TO = []
list_P1_HS = []
list_P1_LUAW = []

list_P2 = []
list_P2_CTL = []
list_P2_C2TO = []
list_P2_HS = []
list_P2_LUAW = []

list_P3 = []
list_P3_CTL = []
list_P3_C2TO = []
list_P3_HS = []
list_P3_LUAW = []

list_P4 = []
list_P4_CTL = []
list_P4_C2TO = []
list_P4_HS = []
list_P4_LUAW = []

list_CTL = []
list_C2TO = []
list_LUAW = []
list_HS = []

for i in list_all:
    if "P1" in i:
```



```
        list_P3_HS.append(j)

for j in list_P3:
    if "LUAW" in j:
        list_P3_LUAW.append(j)


for j in list_P4:
    if "CTL" in j:
        list_P4_CTL.append(j)

for j in list_P4:
    if "C2TO" in j:
        list_P4_C2TO.append(j)

for j in list_P4:
    if "HS" in j:
        list_P4_HS.append(j)

for j in list_P4:
    if "LUAW" in j:
        list_P4_LUAW.append(j)



for j in list_all:
    if "CTL" in j:
        list_CTL.append(j)

for j in list_all:
    if "C2TO" in j:
        list_C2TO.append(j)

for j in list_all:
    if "LUAW" in j:
        list_LUAW.append(j)

for j in list_all:
    if "HS" in j:
        list_HS.append(j)
```

```
In [ ]: # creating a dtw distance matrix
population = list_P4_LUAW

sample_list = {
    k.split("_")[0]+"_"+k.split("_")[2]+"_"+k.split("_")[5]: np.vstack((dict_al
}

dtw_d_mat = np.zeros((len(sample_list), len(sample_list)))
dtw_d_mat = pd.DataFrame(dtw_d_mat, index=sample_list, columns=sample_list)

for i in sample_list:
    for j in sample_list:
        idx = sample_list[i]
        qry = sample_list[j]

        dtw_d_mat[i][j] = dtw(qry, idx, step_pattern=symmetric1).distance

dtw_d_mat
```

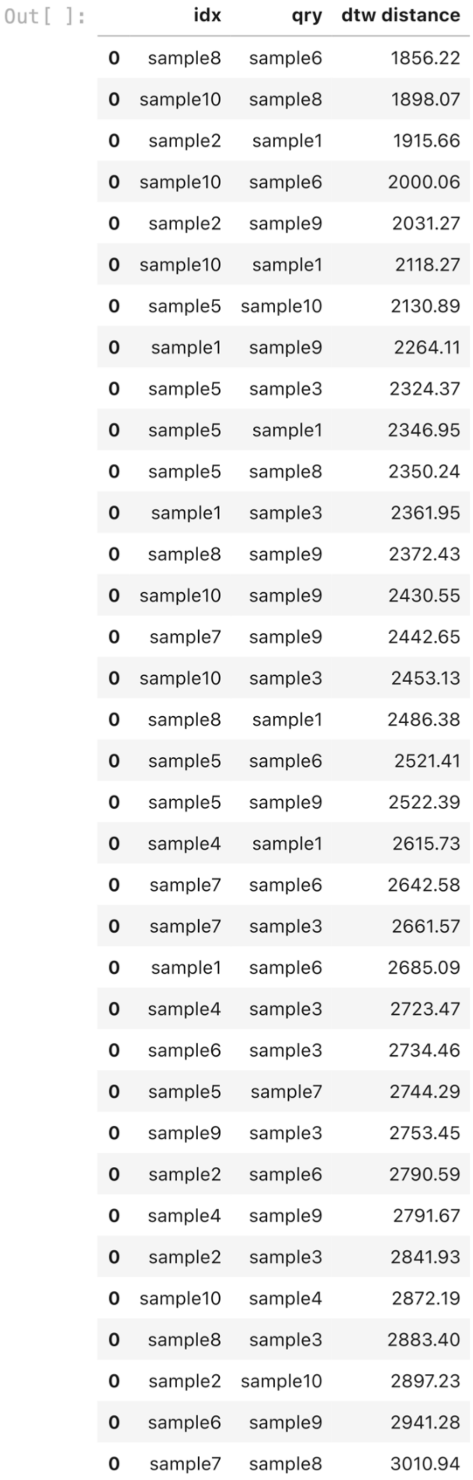


Out[ ]:

	P4_LUAW_sample2	P4_LUAW_sample5	P4_LUAW_sample10	P4_LUAW_sa
P4_LUAW_sample2	0.000000	2416.180006	2082.132484	2419.7
P4_LUAW_sample5	2416.180006	0.000000	1369.613408	1953.0
P4_LUAW_sample10	2082.132484	1369.613408	0.000000	2574.6
P4_LUAW_sample7	2419.704379	1953.033284	2574.675956	0.0
P4_LUAW_sample8	2306.357857	1311.826006	1395.358812	2350.7
P4_LUAW_sample4	2838.412633	2175.678799	2330.424229	1790.0
P4_LUAW_sample1	1234.833798	1556.641174	1571.674753	2222.9
P4_LUAW_sample6	2257.220004	1687.788749	1501.323643	2093.7
P4_LUAW_sample9	1382.051721	2064.427422	1791.106911	1705.2
P4_LUAW_sample3	2296.216772	1694.307211	1720.578822	1675.9

In [ ]:

```
DTW_result = pd.DataFrame({"idx": [], "qry": [], "dtw distance":[]})

for i in list(itertools.combinations(population, 2)):
    j = i[0]
    k = i[1]
    #idx = pd.concat([pd.read_csv(os.getcwd()+"\Data\\"+j)["X"], pd.read_csv(os
    #qry = pd.concat([pd.read_csv(os.getcwd()+"\Data\\"+k)["X"], pd.read_csv(os

    idx = pd.concat([pd.read_csv(os.path.join(os.getcwd(), "Data", "MCO35", j))
    qry = pd.concat([pd.read_csv(os.path.join(os.getcwd(), "Data", "MCO35", k))

    alignment = dtw(qry, idx, keep_internals=True)

    df = pd.DataFrame({"idx": [j.split("_")[5]], "qry": [k.split("_")[5]], "dtw

    DTW_result = DTW_result.append(df)

DTW_result.sort_values(by=['dtw distance'])
```



Out[ ]:

	idx	qry	dtw distance
0	sample8	sample6	1856.22
0	sample10	sample8	1898.07
0	sample2	sample1	1915.66
0	sample10	sample6	2000.06
0	sample2	sample9	2031.27
0	sample10	sample1	2118.27
0	sample5	sample10	2130.89
0	sample1	sample9	2264.11
0	sample5	sample3	2324.37
0	sample5	sample1	2346.95
0	sample5	sample8	2350.24
0	sample1	sample3	2361.95
0	sample8	sample9	2372.43
0	sample10	sample9	2430.55
0	sample7	sample9	2442.65
0	sample10	sample3	2453.13
0	sample8	sample1	2486.38
0	sample5	sample6	2521.41
0	sample5	sample9	2522.39
0	sample4	sample1	2615.73
0	sample7	sample6	2642.58
0	sample7	sample3	2661.57
0	sample1	sample6	2685.09
0	sample4	sample3	2723.47
0	sample6	sample3	2734.46
0	sample5	sample7	2744.29
0	sample9	sample3	2753.45
0	sample2	sample6	2790.59
0	sample4	sample9	2791.67
0	sample2	sample3	2841.93
0	sample10	sample4	2872.19
0	sample8	sample3	2883.40
0	sample2	sample10	2897.23
0	sample6	sample9	2941.28
0	sample7	sample8	3010.94

	idx	qry	dtw distance
0	sample10	sample7	3015.84
0	sample2	sample8	3060.76
0	sample7	sample4	3062.24
0	sample2	sample7	3134.69
0	sample7	sample1	3175.04
0	sample2	sample5	3364.54
0	sample8	sample4	3389.66
0	sample5	sample4	3422.53
0	sample4	sample6	3429.73
0	sample2	sample4	3728.15

In [ ]:

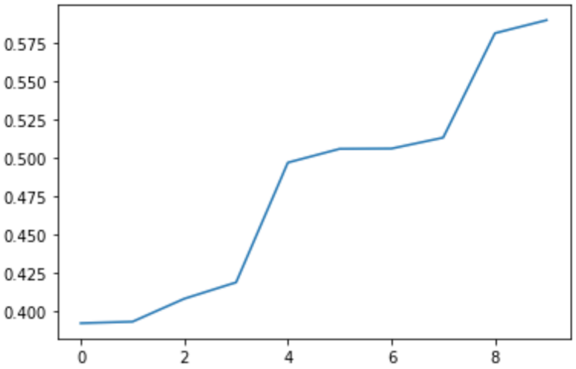
```
# Applying MDS based on DTW distance matrix
embedding = sklearn.manifold.MDS(n_components=2, max_iter=50, random_state=42,
mds = embedding.fit_transform(dtw_d_mat)

pd.DataFrame(mds, columns=["x", "y"])
```

Out[ ]:

	x	y
0	0.518220	-0.100670
1	-0.382670	-0.138227
2	0.033647	0.378762
3	-0.400700	0.068691
4	0.274912	-0.409872
5	0.018092	-0.214657
6	-0.336090	0.356272
7	0.414666	0.471093
8	-0.383356	-0.520615
9	0.299413	0.095357

In [ ]:

```
# DBSCAN parameters
# minPts = no._of_dimensions *2 =4
# eps
neighbors = NearestNeighbors(n_neighbors=4).fit(mds)
distances, idx = neighbors.kneighbors(mds)
distances = np.sort(distances, axis=0)
distances = distances[:, 3]

plt.plot(distances)
plt.show()
```



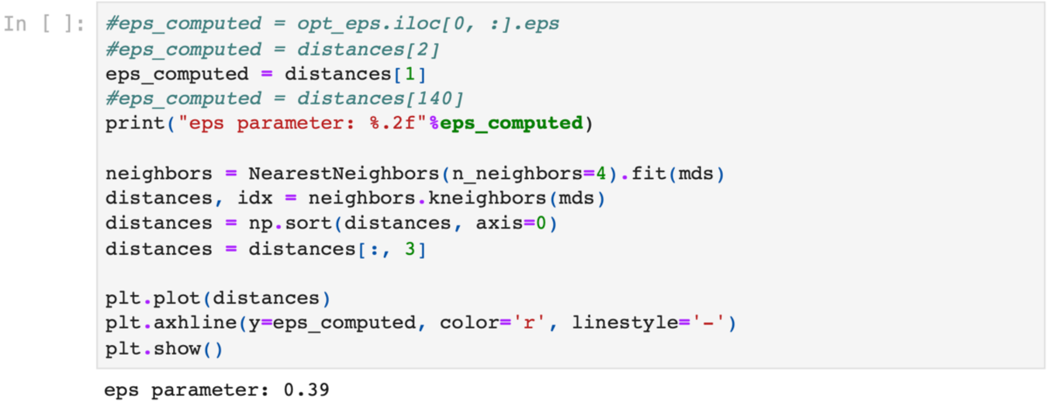
eps_sil_table = pd.DataFrame(columns=["eps", "silhouette"])

for i in distances: clustering = sklearn.cluster.DBSCAN(eps=i, min_samples=4).fit(mds)

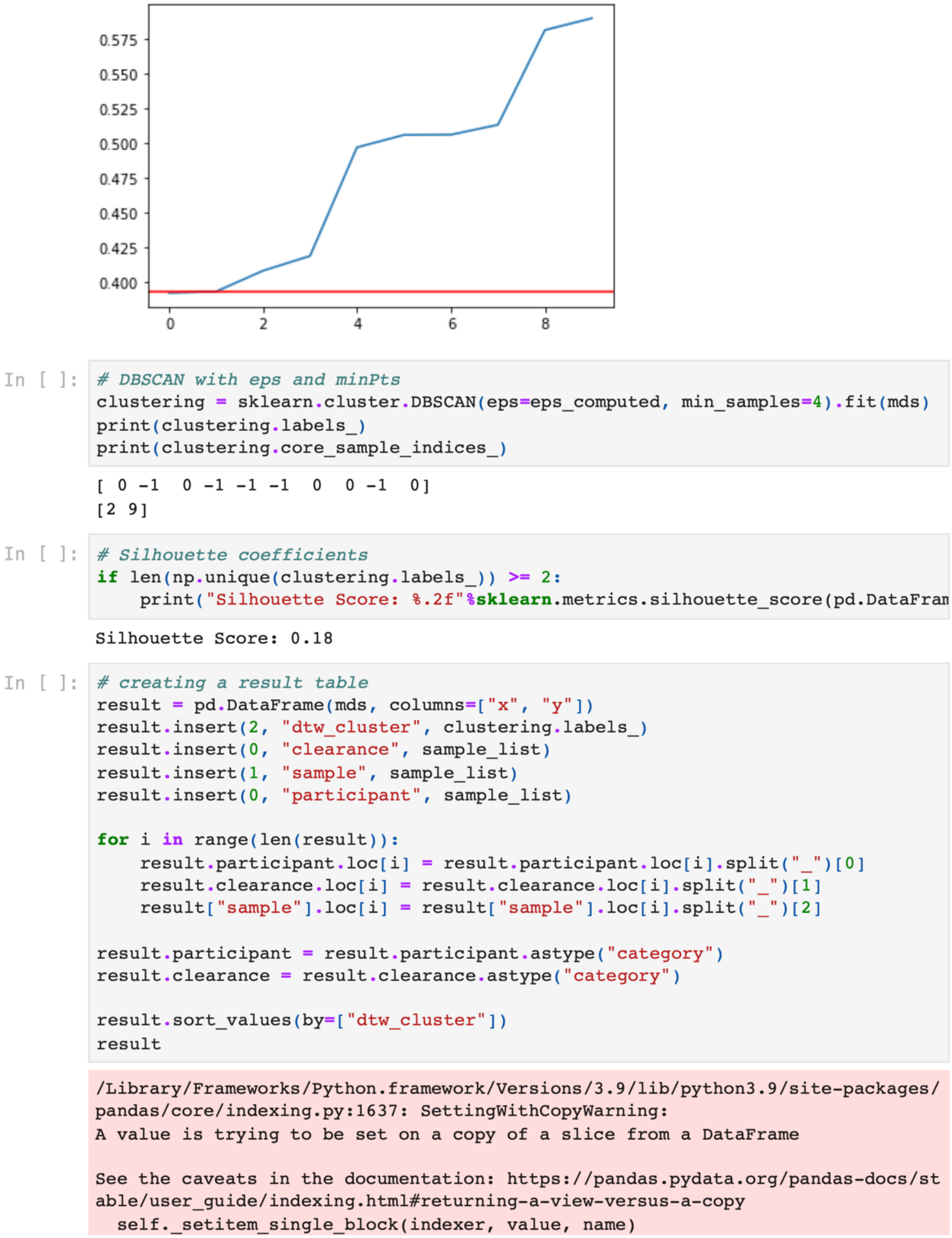
```
if len(np.unique(clustering.labels_)) >= 2:
    silhouette = sklearn.metrics.silhouette_score(mds,
clustering.labels_)
else:
    silhouette = 0

eps_sil_table = eps_sil_table.append({"eps": i, "silhouette":
silhouette}, ignore_index=True)
```

opt_eps = eps_sil_table[eps_sil_table.silhouette == max(eps_sil_table.silhouette)]

print(opt_eps.iloc[0, :]) print(eps_sil_table)



```
In [ ]: #eps_computed = opt_eps.iloc[0, :].eps
        #eps_computed = distances[2]
        eps_computed = distances[1]
        #eps_computed = distances[140]
        print("eps parameter: %.2f"%eps_computed)

        neighbors = NearestNeighbors(n_neighbors=4).fit(mds)
        distances, idx = neighbors.kneighbors(mds)
        distances = np.sort(distances, axis=0)
        distances = distances[:, 3]

        plt.plot(distances)
        plt.axhline(y=eps_computed, color='r', linestyle='-')
        plt.show()

        eps parameter: 0.39
```



```
In [ ]: # DBSCAN with eps and minPts
        clustering = sklearn.cluster.DBSCAN(eps=eps_computed, min_samples=4).fit(mds)
        print(clustering.labels_)
        print(clustering.core_sample_indices_)

        [ 0 -1  0 -1 -1 -1  0  0 -1  0]
        [2 9]

In [ ]: # Silhouette coefficients
        if len(np.unique(clustering.labels_)) >= 2:
            print("Silhouette Score: %.2f"%sklearn.metrics.silhouette_score(pd.DataFran

        Silhouette Score: 0.18

In [ ]: # creating a result table
        result = pd.DataFrame(mds, columns=["x", "y"])
        result.insert(2, "dtw_cluster", clustering.labels_)
        result.insert(0, "clearance", sample_list)
        result.insert(1, "sample", sample_list)
        result.insert(0, "participant", sample_list)

        for i in range(len(result)):
            result.participant.loc[i] = result.participant.loc[i].split("_")[0]
            result.clearance.loc[i] = result.clearance.loc[i].split("_")[1]
            result["sample"].loc[i] = result["sample"].loc[i].split("_")[2]

        result.participant = result.participant.astype("category")
        result.clearance = result.clearance.astype("category")

        result.sort_values(by=["dtw_cluster"])
        result

        /Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/
        pandas/core/indexing.py:1637: SettingWithCopyWarning:
        A value is trying to be set on a copy of a slice from a DataFrame

        See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/st
        able/user_guide/indexing.html#returning-a-view-versus-a-copy
          self._setitem_single_block(indexer, value, name)
```

Out[ ]:

	participant	clearance	sample	x	y	dtw_cluster
0	P4	LUAW	sample2	0.518220	-0.100670	0
1	P4	LUAW	sample5	-0.382670	-0.138227	-1
2	P4	LUAW	sample10	0.033647	0.378762	0
3	P4	LUAW	sample7	-0.400700	0.068691	-1
4	P4	LUAW	sample8	0.274912	-0.409872	-1
5	P4	LUAW	sample4	0.018092	-0.214657	-1
6	P4	LUAW	sample1	-0.336090	0.356272	0
7	P4	LUAW	sample6	0.414666	0.471093	0
8	P4	LUAW	sample9	-0.383356	-0.520615	-1
9	P4	LUAW	sample3	0.299413	0.095357	0

In [ ]:

```
if len(np.unique(clustering.labels_)) >= 2:
    print("Silhouette Score: %.2f"%sklearn.metrics.silhouette_score(result[["x"
```

```
Silhouette Score: 0.41
```

P1_CTL: 1150, P1_C2TO: 1650, P1_HS: 950, P1_LUAW: 1150,

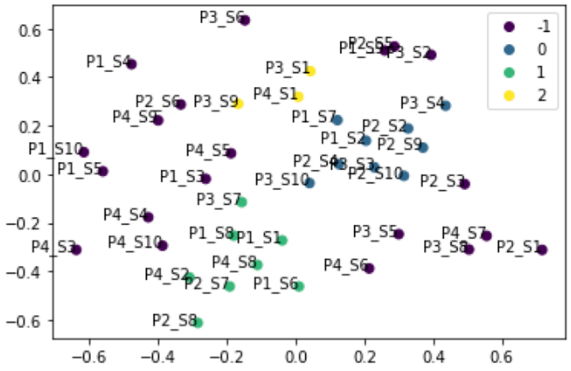
P2_CTL: P2_C2TO: P2_HS: P2_LUAW:

P3_CTL: P3_C2TO: P3_HS: P3_LUAW:

P4_CTL: P4_C2TO: P4_HS: P4_LUAW:

In [ ]:

```
sc = plt.scatter(result["x"], result["y"], c=result["dtw_cluster"])
for i in range(len(result)):
    plt.text(result["x"][i], result["y"][i], result["participant"][i]+"_S"+resu
plt.legend(handles=sc.legend_elements()[0], labels=list(range(-1, max(result["c
#plt.title("Participant " + result["participant"][0][-1] + " " + result["cleara
plt.show()
```

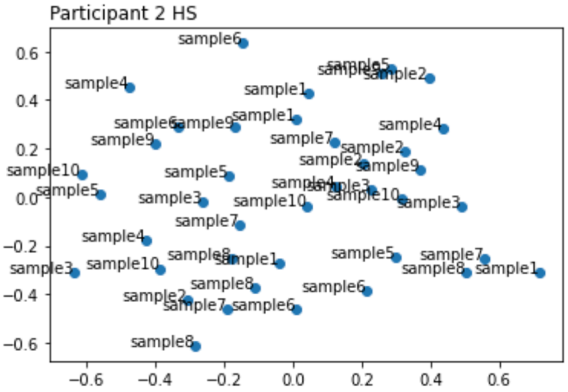


In [ ]:

```
sc = plt.scatter(result["x"], result["y"], c=result["dtw_cluster"])
plt.legend(handles=sc.legend_elements()[0], labels=list(range(-1, max(result["c
plt.show()
```

In [ ]:

```
sc = plt.scatter(result["x"], result["y"])
for i in range(len(result)):
    plt.text(result["x"][i], result["y"][i], result["sample"][i], horizontalali
plt.title("Participant " + result["participant"][0][-1] + " " + result["clearar
plt.show()
```



In [ ]:

```
plt.subplot(1, 2, 1)
sc1 = plt.scatter(result["x"], result["y"], c=result["dtw_cluster"])
plt.legend(handles=sc1.legend_elements()[0], labels=list(range(-1, max(result['
plt.title("DBSCAN Result based on DTW", loc="left")
plt.subplot(1, 2, 2)
sc2 = plt.scatter(result["x"], result["y"], c=result["participant"].cat.codes)
plt.legend(handles=sc2.legend_elements()[0], labels=["P1", "P2", "P3", "P4"])
plt.title(result["clearance"][0]+" colored by participants", loc="left")
plt.savefig("Pall_"+result["clearance"][0]+"_comparison.png")
plt.show()
```

In [ ]:

```
# generating the k-distance plot again and saving the figure

plt.plot(distances)
plt.axhline(y=eps_computed, color='r', linestyle='-')
plt.savefig(result["participant"][0]+"_"+result["clearance"][0]+"_kdist.png")
plt.show()
```