Visualising Moving Clusters using Cluster Flow Diagrams

Jason Thompson[1] **Torab Torabi**[2]

Department of Computer Science and Computer Engineering
La Trobe University,
Melbourne, Victoria 3086
Email: j9thompson@students.latrobe.edu.au[1], t.torabi@latrobe.edu.au[2]

Abstract

Moving clusters represent groups of objects that move together, for instance, groups of people evacuating a building. However, because moving clusters are composed of lists of clusters, they are not directly interpretable by analysts in their raw form. Hence, this paper introduces 'cluster flow diagrams', a clear, concise and aggregated visualisation of a collection of these clusters. In particular, cluster flow diagrams give a snapshot of all of the formations of moving clusters, the changes within them and their interrelationships with one another. Additionally, the clusters are characterised by their member's spatial movements through two composite visualisations. The diagrams are generated using a four stage method which includes provisions to minimise noise in the data and artifacts from the process. Lastly, cluster flow diagrams are generated and evaluated using a synthetic evacuation scenario, a gaze tracking experiment, and a collection of storm tracks, truck trips and naval vessel trajectories.
Keywords: Moving clusters, spatio-temporal, visualisation, moving points, trajectories, groups, clusters

1 Introduction

A moving object could be a person moving about a building, a boat moving about an ocean or even a person's point of regard as they view an image. The movements of these objects can, in the aggregate, give insights into the underlying processes that generated them. One way to analyse moving objects in the aggregate is to group them and then to analyse the formation, dispersal and evolution of these groups. But, this is challenging because the underlying data is structured as lists which are not directly interpretable by analysts in their raw form. One way to address this problem is to visually represent the groups. A straightforward solution would be to apply existing literature on visualising time varying groups to spatial objects. A time varying group is a collection of objects, e.g. collections of documents, that change membership over time. Those members may transfer from one group to another, and the groups may in turn form and disperse at any point in time.

Whilst there has been some recent research into the problem of visualising collections of time varying groups, most researchers only propose specialised visualisations for different types of underlying data e.g. topics in a corpus of documents(Cui et al. 2011), groups of people (Reda et al. 2011) and/or tweets(Xu et al. 2013) within a social network, scenes of characters in narratives(Tanahashi & Ma 2012)(Liu et al. 2013) and groups of programmers in a software engineering project(Ogawa & Ma 2010). Conversely, there has been some general research into visualising groups of vertices in dynamic graphs(Sallaberry et al. 2013). This specialisation is necessary because it allows the groups to be characterised in terms of the underlying data. If textual data is used then keywords can be rendered over the groups(Xu et al. 2013), or if a narrative is used then the groups can be annotated with the names of the characters and their location(Liu et al. 2013). This additional information produces a richer visualisation. However, there has not been much work to address the problem of visualising time varying groups of moving objects which is the focus of this paper. Specifically, the contributions of this paper are:

- a novel visualisation of groups of moving objects, 'cluster flow diagram';
- a 'noise cluster' which describes how organised the dataset is over time;
- an approach to process moving clusters to make them suitable for visualisation; and
- two composite visualisations that characterise groups in terms of their member's movements

2 Related Work

The related work provides the frame of reference for the proposed cluster flow diagram. Section 2.1 and 2.2 review work that addresses a similar problem whilst section 2.3 and 2.4 review the foundational work that cluster flow diagrams build upon.

2.1 View of Individuals

Views that depict the individual objects preserve the outliers in the dataset and their unique behaviours. There are two common trade-offs with this approach: the visual clutter in a diagram increases with the number of objects and clusters must be manually identified and/or characterised by an analyst. One approach is to project objects from two dimensional space to one dimensional space at regular time intervals and to form trajectories from the new points (Crnovrsanin et al. 2009)(Shrestha et al. 2013). The resulting visualisation can, in theory, be used to identify some of the movement patterns explained in (Dodge et al. 2008). Another approach is to segment an environment into regions of interest and then plot each trajectory as a time series with the region ID on the y-axis and time on the x-axis (Sookhanaphibarn et al. 2011)(Raschke et al. 2014). The overlapping parts of the time series can be interpreted as groups of objects. In contrast, groups of objects can be visualised by colour coding their constituent trajectories (Andrienko & Andrienko 2013). Lastly, the time varying changes in the members of a cluster can be compared to one another by stacking each object's trajectory in a three dimensional visualisation (Tominski et al. 2012). Though this method shows how the state of a group changes with time, it does not show how different clusters relate to each other over time.

2.2 Aggregate View

An aggregated view of the objects eliminates outliers, noise, and high frequency variations to provide a summarised view of the underlying phenomena. One aggregate approach visualises time varying clusters using stacked bar graphs to represent the relative membership size of clusters over time (Bremm et al. 2011)(von Landesberger et al. 2012). However, this approach cannot scale well when the number of time samples is increased; it becomes cluttered. Alternatively, using a river-like design motif that is structurally similar to cluster flow diagrams, the movements of objects between areas of interest can be visualised(Burch et al. 2013). While the aforementioned research emphasizes the temporal characteristics of the clusters, the relationships of groups with respect to their spatial environment can be examined using flow maps (Andrienko & Andrienko 2011).

2.3 Sankey Diagrams

Cluster flow diagrams can be considered as a specialisation of *Sankey diagrams*, which are graphs. Characteristically, each edge is associated with a value, and its thickness is proportional to that value. These diagrams also show the splitting and merging of different quantities and have been used extensively to describe energy transformations in a system, for example, energy usage in the United States (`https://flowcharts.llnl.gov/energy.html`). These types of diagrams are typically constructed manually. A list of software to construct such diagrams is available at `http://www.sankey-diagrams.com/sankey-diagram-software/`. Cluster flow diagrams build upon Sankey diagrams by attaching additional meaning to edges (or links), incorporating time and conserving the number of objects with a special purpose noise cluster (described in section 3.1).

2.4 Moving Clusters

A number of algorithms have been developed to identify groups of objects that match a particular spatial-temporal pattern in a given dataset, like a flock pattern (Jeung et al. 2011). However, in the case of cluster flow diagrams, only groups of closely located objects are relevant, for example, moving clusters (Kalnis et al. 2005), convoys (Jeung, Shen & Zhou 2008)(Jeung, Yiu, Zhou, Jensen & Shen 2008), dynamic convoys (Aung & Tan 2010), and swarms (Li et al. 2010). Moving clusters have two properties that make them a satisfactory candidate to segment a dataset into groups: an object belongs to only one group at a time which ensures that the number of objects depicted in a diagram is the same as the number objects in the dataset; and the transfer of membership between groups is recoverable which is necessary to show the relationships between clusters. That being said, in this paper, undesirable clusters are eliminated using constraints on duration and member count that are similar to those used in convoys, dynamic convoys and swarms.

3 The Cluster Flow Diagram

In this section, we propose cluster flow diagrams as a visualisation of groups of moving objects. We describe the two visual elements that make up a diagram those being clusters (section 3.1) and links (section 3.2). In a cluster flow diagram, the x-axis represents time and runs from left to right. The y-axis has no meaning so that visual elements can be shifted up and down to produce more intelligible layouts. Cluster flow diagrams contain two types of visual elements, links and clusters, which represent aggregated transfers of membership and clusters respectfully. An example cluster flow diagram is shown in figure 1. It was generated from the VAST dataset described in section 6.2. For illustrative purposes, the clusters and some of the links are labeled c_x and l_x, respectively, where x is a numerical identifier for that cluster or link.

Over existing methods our visualisation offers some advantages. It scales in terms of the number of objects (it does not become more cluttered with additional objects but it will with additional groups) unlike other approaches that draw every object, and it scales in terms of the number of samples (it does not become more cluttered with additional time samples). It addresses the issue of characterising the groups in terms of their member's movements. Lastly, by drawing groups as well defined bars and links with additional geometry, it provides richer and more precise semantics than other approaches that utilise an organic or river-like design motif(Burch et al. 2013)(Cui et al. 2011)(Xu et al. 2013).

3.1 Clusters

Clusters are groups of objects that are located close to one another and are designated by colour coded horizontal bars. The horizontal axis represents time, so the left and right ends of a horizontal bar correspond to the time that the cluster formed and the time that its members dispersed respectively. Furthermore, if a cluster appears to the right of another, it means that the right cluster formed later than the left. In figure 1, c_0 formed before c_4.

To show the relative memberships of the clusters, the thickness of a cluster's bar varies in proportion to its membership. For example, in figure 1, when c_2 gained more members towards the end of its lifetime, its thickness increased. Also, the thicker bars in a diagram correspond to the clusters with the most members. Note that links introduce additional geometry to the clusters to which they attach, so a cluster's membership may be smaller or larger than what is depicted at the connection point.

If at any point in time an object is not assigned to a group, it is placed in the noise cluster. The noise cluster gives insight into how organised a system is. Changes in the thickness of the noise cluster can indicate if the system is getting more or less organised as time progresses. In figure 1, for example, the noise cluster noticeably decreases in thickness as time progresses, correlating to the objects assembling at the exit points. The noise cluster is given a dark green colour to differentiate it from the other clusters which have no assigned meaning to their colour. However, when the movements of objects are visualised, they can be colour coded by their cluster to create a semantic connection between the two visualisations.

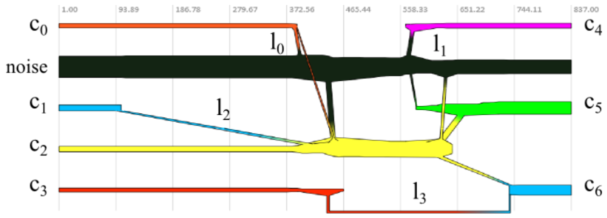

Figure 1: A sample cluster flow diagram generated from the VAST synthetic evacuation dataset.

3.2 Links

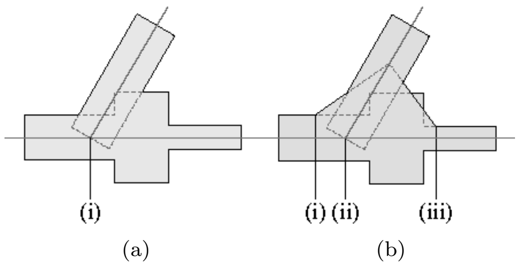

Figure 2: Two configurations that connect a link to a cluster. A link with no underlying variation (a) is positioned at the exit or entry time (i) of the objects. Conversely, a link with variation (b) depicts the minimum (i), average (ii) and maximum (iii) of the exit or entry times.

When objects leave one group and join another, a link forms between the two corresponding horizontal bars. For example, l_2 shows a number of objects left c_1 and joined c_2. Links aggregate individual transitions between clusters so that the relative number of transitions between clusters is clear which would not be the case if a large number of transitions were rendered individually on top of one another. Also, aggregating the links makes the diagrams less cluttered in appearance (see figure 7). The thickness of a link is proportional to the number of transitions it represents: thicker links correspond to larger numbers of transitions. As these links represent multiple similar, but not necessarily equal, transitions, the starting location of a link is represented by three values, the minimum, maximum and average starting times of the underlying transitions that make up the link. The ending location of a link is represented similarly. In place of the minimum and maximum values, the first and third quartiles could have been used to reduce the effects of outliers. Figure 2(b) shows how the three values are depicted. The purpose of the geometry is to provide an insight into the period of time during which the objects transitioned and the skew of the starting and ending times.

When a link connects two clusters with the same y-coordinates, it is drawn in a 'u' shape below the clusters, e.g., l_3 in figure 1. If these links were to be drawn as straight lines starting inside one cluster and ending inside of another, the min, average and max times could not be shown, and they would obscure their start and end clusters and potentially other horizontal links.

As links represent transitions from one cluster to another, they have a particular direction. This direction is, in part, represented by the link's colour gradient. The colour of the cluster from which the link originates fills the majority of the link while the colour of the cluster at which the link terminates has only a minor influence. The gradient is only needed to disambiguate situations where a link is nearly vertical; otherwise, the direction can be determined by the angle of the link. Because time runs along the x axis and objects cannot travel backwards through time, the objects transition from the left endpoint to the right endpoint, e.g., in figure 1 the members of l_1 can only transition from the noise cluster to c_4.

4 Characterising Groups

The groups as they are represented now are completely non-descript; they could be collections of tweets, actors, or anything. A richer representation of the groups and their membership is required. We propose two composite visualisations to characterise the groups in terms of their member's movements: a dynamic composition and a static composition.

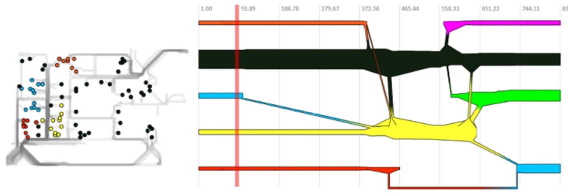

Figure 3: A dynamic composition of a visualisation of the spatial domain and the groups present in the dataset.

The dynamic composition consists of an animated visualisation of the spatial domain and the cluster flow diagram. Moving objects are displayed on top of their trajectories in the animated visualisation. The moving objects are shown as dots coloured according to their current group. When the animation is played, the dots trace out the path of their corresponding objects and change colour as they change groups. So that the animation and the cluster flow diagram are synchronised in the temporal domain, a vertical red bar is overlayed on top of the cluster flow diagram at the animations current time. In this configuration, an analyst can playback the dataset and jump to particular points in the dataset. The analyst is able to characterise the groups by viewing them in the spatial domain as well as on the cluster flow diagram. An example is shown in figure 3.

While an animated view may be useful for exploratory analysis of data, a static figure is still more efficient – all of the information can be seen at a glance. Thus, we propose a static composition. It again includes a visualisation of each object's trajectory and the moving objects colour coded by group; however, the elements are static in this configuration. This approach allows an analyst to annotate the cluster flow diagram with static views of the dataset. In this way, key events in the evolution of the groups can be marked and viewed in the spatial domain. An example is shown in figure 4; note that a red bar is used to indicate to which time that the static visualisation corresponds.

Figure 4: A static composition where the cluster flow diagram has been annotated with multiple visualisations of the spatial domain.

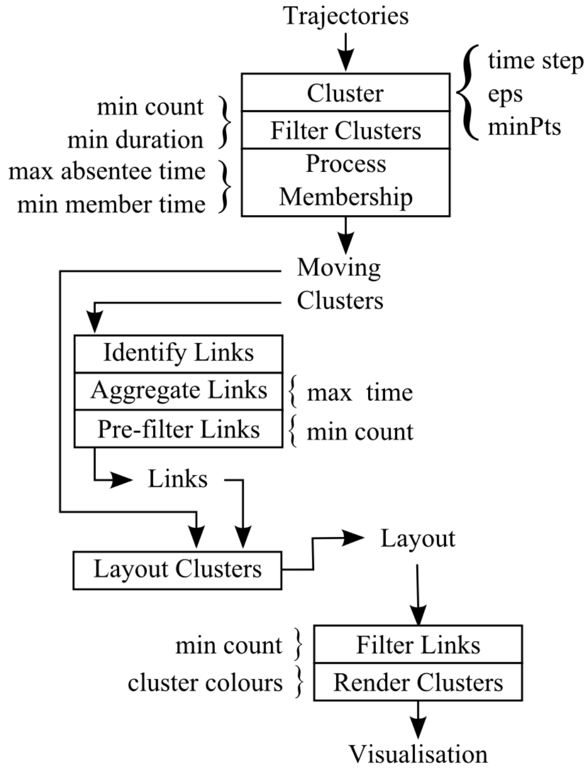

Figure 5: A flow chart showing how cluster flow diagrams are generated from trajectories. Arrows represent the flow of information, parameters are listed next to the curly brackets and the groups of boxes correspond to the four main stages: identify moving clusters (section 5.1), identify links between clusters (section 5.2), layout the diagram (section 5.3) and render the diagram (section 5.4).

5 Generating Cluster Flow Diagrams

An overview of the method used to generate cluster flow diagrams is provided in figure 5. The four groups of rectangles correspond to the four main stages: identify moving clusters (section 5.1), identify links between clusters (section 5.2), layout the diagram (section 5.3) and render the diagram (section 5.4).

The quality of the end diagram depends on the quality of the clustering procedure and also the randomness of the dataset. Hence, a number of strategies have been included to mitigate these two factors. First, all of the moving clusters that are identified are filtered if they are "short-lived" and/or do not contain enough members. Second, the transitions of objects between clusters are aggregated to form links and filtered if they do not contain enough members. Finally, to improve the aesthetics of a cluster flow diagram, the number of line intersections and non-horizontal lines are minimised by shifting clusters up and down the y-axis.

5.1 Identifying Moving Clusters

Moving clusters are identified in much the same way as described in (Kalnis et al. 2005): the lifetime of a dataset is divided into a number of equal time intervals; the objects are clustered during each time interval using the DBSCAN algorithm (Ester et al. 1996); and clusters that share common members are connected to form the moving clusters. The major departure from the above method is that the noise from each time interval is always placed into a noise cluster. The noise cluster is used to conserve the number of points represented on the diagram at any point in time. Also, the noise cluster cannot be filtered.

This procedure can produce peculiar results. An object's motion during a time interval is approximated by a point. If the time intervals are too long or an object moves too quickly, the sampled point will not be representative of the object's motion during that time interval. This reduces the quality of the resultant moving clusters as objects can appear to be absent from a cluster when, upon visual inspection, they should not be and vice versa. Conversely, the movements of an object during a time interval could be approximated by line segments. These could be clustered instead of points, as in TRACLUS (*Trajectory clustering: a partition-and-group framework* 2007). Note that clustering line segments introduces a number of parameters that do not have straightforward physical interpretations. Consequently, it was decided to approximate motion using points and to reduce the length of the time intervals to produce acceptable results.

In some cases, "short-lived" and/or small (containing few objects) moving clusters can be generated. These low quality clusters result from determining cluster membership at discrete points in time and irregularities in the object's movements. In any case, clusters that never contain more than *min count* objects at any point in time or that have a duration shorter than *min duration* are removed, and their members are reassigned to the noise cluster so that no objects are left unaccounted. This has the benefit of reducing clutter in the resultant diagram as well as reducing the number of clusters that need to be processed by the layout algorithm.

In some cases, a single object can momentarily join a moving cluster. This can happen if an object momentarily comes within range of a cluster or an object temporarily bridges two clusters. These short artificial periods of membership inflate the membership of clusters and introduce spurious transitions, so they are removed. An object must be a member of a cluster for a minimum amount of time, *max member time*; otherwise, its membership is ignored and it is placed in the noise cluster. Once these momentary memberships have been removed, the members of each cluster are recounted. If a cluster no longer has any members then it is removed.

Additionally, objects can appear to exit and rejoin a cluster, resulting in a lapse of membership. This happens for a number of reasons: an object on the edge of cluster temporarily moves out of range; a bridging object(s) temporarily moves out of range splitting a cluster; or an object genuinely leaves a cluster, for some reason, and re-joins it later. The first two cases are artifacts of the clustering process while the second is an interesting characteristic of the dataset. In this method, cases like the first two are identified and eliminated by way of a constraint on the duration of lapses in membership: if an object spends less than *max absentee time* outside of a cluster then that object is considered to have never have left the cluster.

Figure 6 shows the cluster membership of eight objects. Each row represents the lifetime of an object, and time runs from left to right. The colour of the row indicates which cluster that object belongs to at that point in time. Figure 6 shows the lifetimes of the objects before removing momentary memberships (a), after removing momentary membership (b) and after removing the artificial lapses in memberships (c). Although the intention is to remove artificial transitions, an inappropriately selected threshold can eliminate useful information. By using diagrams like the ones in figure 6, it should be possible to select an appropriate value.

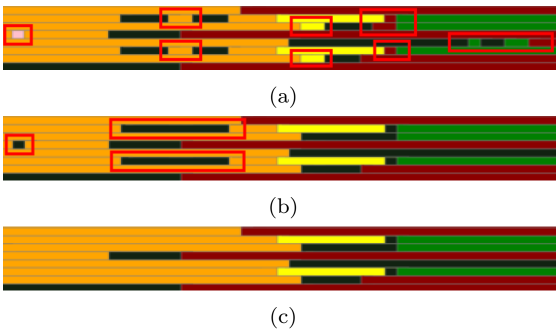

Figure 6: The trajectories before processing (a), momentary memberships are surrounded by red rectangles; after removing the momentary memberships (b), artificial lapses in memberships are surrounded by red rectangles; and after removing artificial lapses in memberships (c).

5.2 Identifying Links between Clusters

A transition denotes the change in cluster membership of an object. It consists of a source cluster, a destination cluster, a start time which is the last time that the object is a member of the source cluster, and an end time which is the first time that the object is a member of the destination cluster. There are two cases when a transition is outputted. The first case is when an object transitions from a moving cluster, to the noise cluster and then to another moving cluster. The resultant transition joins the two moving clusters ignoring the time spent in the noise cluster. The rationale for ignoring the transition to and from the noise cluster is that when an object transitions from one cluster to another there may be a period of time where it appears as noise, but it is actually purposefully moving between clusters. Ignoring the time that the object spent as noise produces a more meaningful transition. Of course there is the risk of ignoring time that an object genuinely spent as noise before joining another cluster. This case was not considered in the implementation, but it might be identified by looking at the amount of time that the object spent as noise or maybe the tortuosity of its path. Alternatively, a domain specific strategy may yield more meaningful results. The second case, the obvious one, produces a transition when an object transitions from any moving cluster to another. This case applies only when the first case does not.

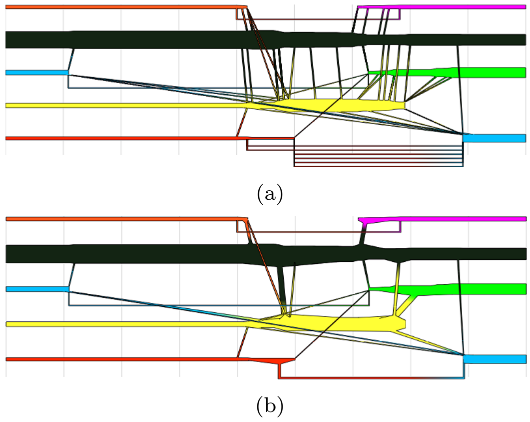

Figure 7: The VAST cluster flow diagram with none of the links aggregated (a) and the links aggregated with *max time* = 15 (b).

At this stage, a lot of transitions can be generated. Not all of the transitions are unique and some are very similar. Consequently, the transitions can be easily aggregated into groups called *links*. Notably, operating on links offers some benefits such as it speeds up the layout stage and reduces the appearance of clutter in the final visualisation in addition to the reasons stated in Section 3.2. Links are formed by grouping all of the transitions that are *similar* to one another where two transitions are *similar iff* the following holds true:

- the start times differ by no more than *max time*
- the end times differ by no more than *max time*
- they start at the same cluster
- they end at the same cluster

Note that *max time* is a user set parameter. Larger values of *max time* correspond to less discriminate clusters. A link is the largest collection of transitions that are all *similar* to one another either directly or indirectly through another transition. Additionally, none of the transitions inside of a link can be similar to the transitions inside of another link.

As an alternative definition, a graph can be formed where vertices correspond to transitions and two vertices are connected if their corresponding transitions are similar. Clusters are the connected components of this graph. The concept of connected components is described in (Skiena 2008). This type of clustering was selected because it has one straightforward parameter, *max time*, and does not require the number of clusters to be known beforehand. Moreover, it produces satisfactory results.

Lastly, links can be filtered if they do not represent many transitions. For example, compare the unfiltered diagram in figure 7 (b) with the filtered version in figure 1. The filtered version does not show links that represent two or fewer transitions.

5.3 Layout Problem

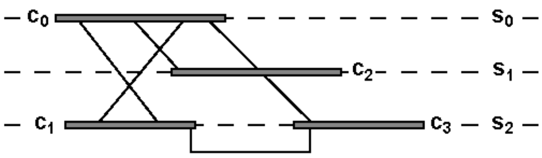

Figure 8: An example layout model of a cluster flow diagram with three slots, s_0, s_1, and s_2, and four clusters, c_0, c_1, c_2, and c_3.

This section will describe the layout problem. The layout algorithm is not the contribution of this paper, so it is not described in detail. In reality any layout algorithm would suffice, e.g., the evolutionary algorithm approach used to optimise storyline visualisations(Tanahashi & Ma 2012).

A simplified form of the cluster flow diagram is used in the layout stage to make the layout algorithm tractable. Time runs along the x axis, and the y axis has no meaning assigned to it. Clusters are represented by horizontal lines with no thickness. The line's leftmost coordinate corresponds to the time that the moving cluster formed and the rightmost coordinate corresponds to the time that the cluster dispersed. Each cluster is assigned a slot which defines its y-coordinate. Slots are spaced equally along the y axis. The minimum number of slots is the maximum number of concurrently defined moving clusters. As an example, the maximum number of concurrently defined moving clusters in figure 8 is three which is also the number of slots.

Links between clusters are represented as lines, again, without any thickness. The first x-coordinate of a line corresponds to the average starting time of the link, and the first y-coordinate corresponds to the slot of the starting cluster. The second x and y-coordinates are defined similarly. As an example, figure 8 has five links, one of which is horizontal. Horizontal links are always placed halfway towards the next slot and below the clusters they connect. While it is possible to test multiple locations for horizontal links, placing the links below the clusters produced reasonable results. A particular layout of some clusters is an ordered list of slots. The first element corresponds to the first cluster and so forth. For example, the layout of figure 8 is (s_0, s_2, s_1, s_2).

A function is defined to describe the quality of any given layout. The function used was the weighted sum of the number of overlapping lines and the number of non-horizontal lines. The layout algorithm finds the layout with smallest value, i.e., the highest quality layout. In the prototype, we used a directed search algorithm; however, any appropriate optimisation algorithm would suffice.

5.4 Rendering the Diagram

The moving clusters are straightforward to render. Their colour is selected from a predefined list. Special care is given to ensure that no link completely covers another link which is tantamount to deleting a link. This is accomplished by putting thicker links below thinner links. In addition, horizontal links are arranged in the y direction to keep multiple horizontal links from overlapping one another, e.g., see the horizontal links in figure 10.

Parameter	VAST	Truck	Eye	Imis	Storm
time slices	100	500	200	200	100
eps	5	1000	60	45	320
min points	5	5	10	15	10
min count (a)	10	50	10	10	15
min duration	7	25	15	20	20
min member time	10	10	0	10	10
max absentee time	10	20	10	10	10
max time (b)	15	15	15	15	15
min count (c)	12	5	2	2	2
min count (d)	0	0	3	2	0

Table 1: (a) the minimum number of members in a cluster, (b) the max time between transitions for them to grouped in a link, (c) the minimum number of transitions in a link for it to be used in the layout stage, (d) the minimum number of transitions in a link for it to be drawn

6 Results

The prototype was developed as a web application using KineticJS and KnockoutJS. No processing times are given because they varied significantly between executions most likely due to the browser environment. This section presents representative cluster flow diagrams generated from four real world datasets and one synthetic dataset. The parameters for each dataset are given in table 1.

6.1 Truck Dataset

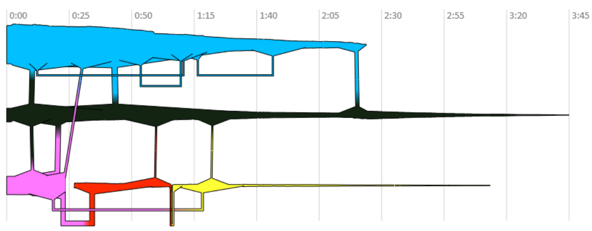

Figure 9: No. Traj. = 966, No. Samples = 42882 and Avg. Duration of Traj. (hrs) = 1.41

The revised trucks dataset (retrieved from `http://www.chorochronos.org/?q=node/10`) contains the trajectories of 50 trucks as they deliver concrete around the Athens metropolitan area in Greece. Initially there was no apparent grouping in the trucks; However, when the trajectories were shifted to start at the same time, the trucks that took similar routes appeared to move in groups. The truck dataset cluster flow diagram (figure 9) shows that initially the dataset is split up into three clusters. It is interesting to note that the noise cluster is the smallest of the three which indicates that the system is initially well organised. All of the clusters decrease in volume over time because the trucks have unequal lifetimes. The diagram shows that the three main clusters are defined throughout the lifetime of the dataset. The three bottom clusters can be seen to form one cluster that breaks up and reforms over time.

6.2 Evacuation Dataset

The VAST dataset (*IEEE VAST 2008 Challenge* 2008) (figure 1) is a synthetic dataset depicting office workers evacuating a building in response to a bomb that has exploded. Besides changing the file format, this dataset was not modified. The leftmost part of the diagram shows four clusters and a large noise cluster. This indicates that initially a noteworthy number of workers are scattered throughout the building. In the middle of the diagram, the clusters converge on the yellow cluster and the noise cluster which is the response to the bomb. Next, the yellow cluster breaks up into smaller clusters which correspond to the evacuation zones. Hence, the workers have reorganised themselves in response to the explosion. The decrease in thickness of the noise cluster indicates that the system is more organised at the end of the dataset than at the beginning.

6.3 Gaze Tracking Dataset

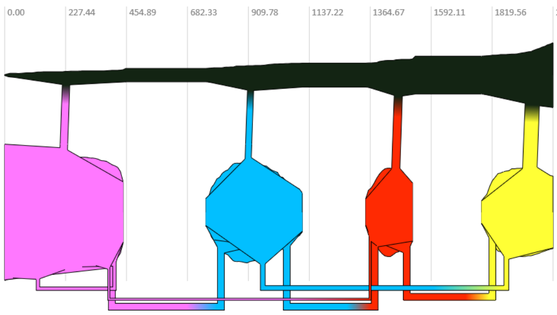

Figure 10: No. Traj. = 326, No. Samples = 667648 and Avg. Duration of Traj. (ms) = 8192

Besides changing the file format, this dataset was not modified. The gaze tracking dataset (KASPROWSKI 2004) (retrieved from `http://www.kaggle.com/c/emvic/data`) shows the point of regard of a group of human test subjects in response to an animated visual stimulus. Note that only the right eye's point of regard from the test dataset was used. The resulting diagram (figure 10) has some distinctive features. The noise cluster consistently increases in volume indicating that the behaviour of the test subjects is becoming more random as the experiment progresses. There are no links from the noise cluster to any other cluster; this is another indication that the system is tending to disorder. Lastly, the system alternates between well organised and completely random as evidenced by the sequence of clusters. The decreasing width of the clusters indicates that the subjects are becoming desensitised to the organising stimulus as the experiment progresses.

6.4 IMIS Dataset

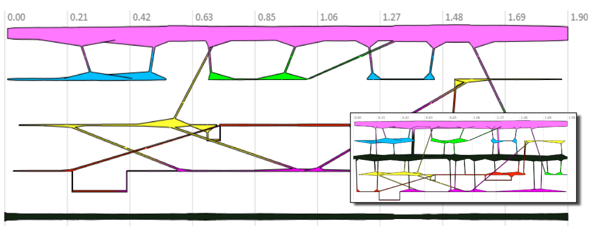

Figure 11: No. Traj. = 930, No. Samples = 442320 and Avg. Duration of Traj. (hrs) = 23.97

The Imis3days dataset (retrieved from `http://www.chorochronos.org/?q=node/8`) was collected by IMIS Hellas S.A. It contains the movements of naval vessels. In order to make parameter tuning and loading the file faster, the sampling frequency of this dataset was reduced from once every ~26 seconds (on average) to once every minute at most. This reduced the data to ~14% of the original number of samples and the file size from ~200Mb to ~20Mb. The average distance between samples beforehand was ~64m, and afterwards ~425m. If desired, once the parameters have been determined, a cluster flow diagram could be computed from the full dataset. Lastly, the latitude/longitude coordinates were converted to Cartesian coordinates using an equirectangular projection. The resultant cluster flow diagram (figure 11) is perhaps the most stable of all of the datasets. All of the clusters have a reasonably constant width and there are three clusters that are defined for the whole lifespan of the dataset. Lastly, this diagram is shown with (smaller diagram) and without (larger diagram) the links to the noise cluster. Note that the diagram with links to the noise cluster appears more cluttered and less intelligible.

6.5 Storm Dataset

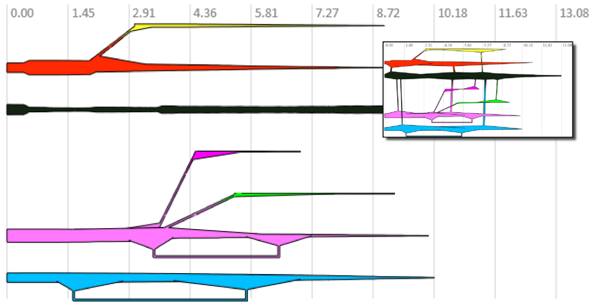

Figure 12: No. Traj. = 6823, No. Samples = 175743 and Avg. Duration of Traj. (days) = 6.6

IBTrACS-WMO v03r04 (The International Best Track Archive for Climate Stewardship-World Meteorological Organisation) dataset (retrieved from `http://www.ncdc.noaa.gov/ibtracs/index.php?name=wmo-data`) contains the movements of hurricanes from the mid 1800's to modern times. The storms were shifted to start at the same time for the same reason as the trucks dataset. Also, the latitude/longitude coordinates were converted to Cartesian coordinates using an equirectangular projection. The dataset was truncated at ~13 days because only a small number of storms had a lifespan that long, ~7% of the hurricanes; otherwise, it would produce a very wide diagram. It resulted in a loss of only ~3% of the samples. The cluster flow diagram shown in figure 12 does not offer much insight into this dataset because it is well organised and remains well organised. The storms appear to be naturally grouped. It may be more insightful to examine each cluster in isolation. Again the diagram is shown with (smaller diagram) and without (larger diagram) links to the noise cluster.

7 Summary, Discussion and Future Work

This paper introduces cluster flow diagrams as a snapshot of all of transfers of membership between groups and the formations and dispersals of groups themselves within a given a dataset. It also presented a four stage methodology to generate these diagrams. Additionally, the methodology has considerations to remove noise and unwanted artifacts of the clustering process. Additionally, we proposed two composite visualisations to further characterise the groups in terms of their members' movements and environment. This approach allows domain experts and analysts to identify at a glance trends in the membership of groups and systematic transfers of membership between groups. Additionaly, the noise cluster allows an analyst to see trends in how organised a dataset is over time.

The result section showed a number of representative cluster flow diagrams. Parameters must be set according to the research questions of the analyst; they cannot be set universally for a particular dataset. Some research questions may be more tolerant to filtering out data, whilst others may require all of the data to be represented. In the future, we plan on developing case studies to show the insight generating capabilities of the visualisation. In the process of doing this, we will develop visualisations to aid the setting of the parameters.

Some of the possible areas for future work include: encoding spatial attributes of the groups on the cluster flow diagram; introducing interactivity to allow users to isolate subsets of the data in which they are interested; and allowing users to interactively construct and modify parameters of cluster flow diagrams.

References

Andrienko, N. & Andrienko, G. (2011), 'Spatial generalization and aggregation of massive movement data.', *IEEE Transactions on Visualization and Computer Graphics* **17**, 205–219.

Andrienko, N. & Andrienko, G. (2013), 'Visual analytics of movement: An overview of methods, tools and procedures', *Information Visualization* **12**(1), 3–24.

Aung, H. H. & Tan, K.-L. (2010), Discovery of evolving convoys, *in* 'Proceedings of the 22nd international conference on Scientific and statistical database management', SSDBM'10, Springer-Verlag, Berlin, Heidelberg, pp. 196–213.

Bremm, S., von Landesberger, T., Andrienko, G., Andrienko, N. & Schreck, T. (2011), Interactive Analysis of Object Group Changes over Time, *in* 'EuroVA International Workshop on Visual Analytics'.

Burch, M., Kull, A. & Weiskopf, D. (2013), 'AOI Rivers for Visualizing Dynamic Eye Gaze Frequencies', *Computer Graphics Forum* **32**(3pt3), 281–290.

Crnovrsanin, T., Muelder, C., Correa, C. & Ma, K.-L. M. K.-L. (2009), 'Proximity-based visualization of movement trace data', *2009 IEEE Symposium on Visual Analytics Science and Technology* pp. 11–18.

Cui, W., Liu, S., Tan, L., Shi, C., Song, Y., Gao, Z., Qu, H. & Tong, X. (2011), 'TextFlow: Towards Better Understanding of Evolving Topics in Text', *Visualization and Computer Graphics, IEEE Transactions on* **17**(12), 2412–2421.

Dodge, S., Weibel, R. & Lautenschütz, A.-K. (2008), 'Towards a taxonomy of movement patterns', *Information Visualization* **7**, 240–252.

Ester, M., Kriegel, H.-p., Sander, J. & Xu, X. (1996), 'A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise', *Computer* **1996**, 226–231.

IEEE VAST 2008 Challenge (2008).
URL: *www.cs.umd.edu/hcil/VASTchallenge08*

Jeung, H., Shen, H. T. & Zhou, X. (2008), Convoy Queries in Spatio-Temporal Databases, *in* 'Proceedings of the 2008 IEEE 24th International Conference on Data Engineering', ICDE '08, IEEE Computer Society, Washington, DC, USA, pp. 1457–1459.

Jeung, H., Yiu, M. L. & Jensen, C. S. (2011), Trajectory Pattern Mining, *in* Y. Zheng & X. Zhou, eds, 'Computing with Spatial Trajectories', pp. 143–177.

Jeung, H., Yiu, M. L., Zhou, X., Jensen, C. S. & Shen, H. T. (2008), 'Discovery of convoys in trajectory databases', *Proc. VLDB Endow.* **1**(1), 1068–1080.

Kalnis, P., Mamoulis, N. & Bakiras, S. (2005), On Discovering Moving Clusters in Spatio-temporal Data, *in* 'Advances in Spatial and Temporal Databases', Vol. 3633, pp. 364–381.

KASPROWSKI, P., O. J. (2004), Eye movement in biometrics, *in* 'In Proceedings of Biometric Authentication Workshop, European Conference on Computer Vision', The IEEE/IARP International Conference on Biometrics (ICB), Springer-Verlag, Prague 2004, LNCS 3087, pp. 1–8.

Li, Z., Ding, B., Han, J. & Kays, R. (2010), 'Swarm: mining relaxed temporal moving object clusters', *Proc. VLDB Endow.* **3**(1-2), 723–734.

Liu, S., Wu, Y., Wei, E., Liu, M. & Liu, Y. (2013), 'Storyflow: Tracking the evolution of stories', *IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE InfoVis 2013* **19**(12).

Ogawa, M. & Ma, K.-L. (2010), Software evolution storylines, *in* 'Proceedings of the 5th International Symposium on Software Visualization', SOFTVIS '10, ACM, New York, NY, USA, pp. 35–42.

Raschke, M., Herr, D., Blascheck, T., Ertl, T., Burch, M., Willmann, S. & Schrauf, M. (2014), A visual approach for scan path comparison, *in* 'Proceedings of the Symposium on Eye Tracking Research and Applications', ETRA '14, ACM, New York, NY, USA, pp. 135–142.

Reda, K., Tantipathananandh, C., Johnson, A., Leigh, J. & Berger-Wolf, T. (2011), Visualizing the evolution of community structures in dynamic social networks, *in* 'Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization', EuroVis'11, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 1061–1070.

Sallaberry, A., Muelder, C. & Ma, K.-L. (2013), Clustering, visualizing, and navigating for large dynamic graphs, *in* W. Didimo & M. Patrignani, eds, 'Graph Drawing', Vol. 7704 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 487–498.

Shrestha, A., Miller, B., Zhu, Y. & Zhao, Y. (2013), Storygraph: Extracting patterns from spatio-temporal data, *in* 'Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics', IDEA '13, ACM, New York, NY, USA, pp. 95–103.

Skiena, S. S. (2008), *The algorithm design manual*, 2nd edn, Springer-Verlag London.

Sookhanaphibarn, K., Thawonmas, R., Rinaldo, F. & Chen, K.-T. (2011), Spatiotemporal Analysis of Circulation Behaviors Using Path and Residing Time displaY (PARTY), *in* 'Digital Media and Digital Content Management (DMDCM), 2011 Workshop on', pp. 284–291.

Tanahashi, Y. & Ma, K.-L. (2012), 'Design Considerations for Optimizing Storyline Visualizations', *Visualization and Computer Graphics, IEEE Transactions on* **18**(12), 2679–2688.

Tominski, C., Schumann, H., Andrienko, G. & Andrienko, N. (2012), 'Stacking-Based Visualization of Trajectory Attribute Data', *Visualization and Computer Graphics, IEEE Transactions on* **18**(12), 2565–2574.

Trajectory clustering: a partition-and-group framework (2007), *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* pp. 593–604.

von Landesberger, T., Andrienko, G., Andrienko, N., Tekusova, M. & Bremm, S. (2012), Visual analytics methods for categoric spatio-temporal data, *in* 'Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST)', VAST '12, IEEE Computer Society, Washington, DC, USA, pp. 183–192.

Xu, P., Wu, Y., Wei, E., Peng, T.-Q., Liu, S., Zhu, J. J. & Qu, H. (2013), 'Visual analysis of topic competition on social media', *IEEE Transactions on Visualization and Computer Graphics (Proceedings of IEEE VAST 2013* **19**(12).