EG UK Computer Graphics & Visual Computing (2024)
A. Slingsby and D. Hunter (Editors)

Comparing distance metrics in space-time clustering to provide visual summaries of traffic congestion

P. Baudains[1] and N. S. Holliman[1]

[1]CUSP London, Department of Informatics, King's College London, UK.

Abstract

Smart Cities are characterised by their ability to collect and process large volumes of sensor data. Visual analytics is then often required to make this data actionable and to allow decisions to be made in support of the well-being of inhabitants. In this study, using Bus Open Data, we consider how space-time clustering can be used to generate visual summaries of traffic congestion. Using a space-time extension of DBSCAN, our clustering procedure is evaluated with respect to both Euclidean distance and street network distance. Results show that network-based distance metrics improve the clustering procedure by generating clusters with less uncertainty. Moreover, congestion clusters derived from network-based distances are also more likely to last longer and to precede future congestion appearing nearby. We suggest that network-based distances might provide greater opportunity for more impactful traffic control room decision-making and we discuss steps towards a near real-time system design that can be used in support of operational decision-making.

CCS Concepts

• ***Human-centered computing*** → ***Visual analytics; Geographic visualization; Information visualization;*** • ***Information systems*** → ***Clustering;*** *Sensor networks; Data analytics;*

1. Introduction

A proliferation of sensors within our urban environment, which has led to concepts such as the Smart City, provides substantial challenges for visual analytics [ZWC*16]. For situations in which timely decisions may have a significant impact on human well-being, there is growing recognition that a human-in-the-loop can provide accountability and safeguards against algorithmic unfairness. However, this stresses the importance of designing human-in-the-loop systems that are reliable, trustworthy and manageable for the humans interacting with them. In high-volume data environments, visual analytics supports this interaction, and often performs the role of summarising the data for human decision-makers in a way that focuses their attention towards more salient features (see, for example, [PCRHS18]).

In this study, using data from the Bus Open Data Service in England [Dep], we design and evaluate visual summaries of traffic congestion to support human decision-makers in situations such as traffic control rooms. We define traffic congestion as space-time regions constrained by the street network with a high density of vehicles travelling at low speed. As such, we adopt a density clustering algorithm known as DBSCAN [EKSX96], with adjustments for the identification of clusters in space-time. Such algorithms are well-suited to the identification of traffic congestion due to their ability to identify observations in low density regions as noise, avoiding false positive identification of congestion events [AAH*11].

Our visualisation is designed to be used within a transport control room, supporting decision-makers who are required to identify appropriate responses to remediate the emergence of unexpected congestion for example due to an accident or protest. This adds a requirement for near real-time operation, so that timely decisions can be made in response to congestion. Our visualisation design uses bivariate *vizent* glyphs [HcF*24] plotted at centre points of identified congestion clusters over a geographic map of the study area. In our work to date, we have used animation to visualise time, overlaying clusters on moving bus trajectories to demonstrate that the clusters effectively capture congestion. An example of such a visualisation, which we have presented to transport stakeholders, is provided in supplementary materials, and a still is shown in Figure 1. However, for control room settings, in which data visualisations may be competing for attention with other priorities, plotting individual trajectories can increase visual clutter [AA11]. Furthermore, overlapping of points masks the true size of congestion events. There is a need, therefore, to aggregate trajectories so that only congestion events are displayed. For such cluster-only visualisations, assurance is required that clusters presented to the viewer faithfully reflect the congestion situation on the road network.

As a case study, we use a two-week period 1st-14th November 2023 within central London. This period contains two days during which there were known significant disruptions. On 8th November 2023, a protest by the group Just Stop Oil unexpectedly disrupted

DOI: 10.2312/cgvc.20241233

Figure 1: *A still from an animation demonstrating the use of cluster centre points to summarise traffic congestion, overlaid on individual bus trajectories (in orange). Each cluster is visualised using vizent glyphs introduced in [HcF*24], plotted at the geographic centre of each cluster. An animated version is provided in supplementary materials. Combining trajectories with cluster points is a useful validation exercise, but adds visual clutter that may be undesirable in control room settings.*

traffic on Waterloo bridge, preventing any traffic from passing for over an hour [Jus23]. The 11th November 2023 was also a day of significant disruption to ordinary road traffic within central London. Remembrance Day commemorations at London's Cenotaph took place, shutting roads that buses would ordinarily travel along. In addition, a large demonstration relating to the conflict in Palestine with a reported 300,000 participants took place in central London, closing roads and potentially inducing congestion on diversion routes. This protest also attracted a small counter-demonstration, which may have additionally caused traffic disruption [Al 23].

This paper makes two important contributions. First, we implement two alternative specifications for our clustering algorithm. Since the DBSCAN algorithm depends crucially on an appropriate method for identifying neighbouring observations, we consider whether street network distances provide 'better' clusters than Euclidean distances between observations. Second, we present a novel evaluation procedure for the identified clusters that enables us to determine whether one approach is 'better' than the other. Our evaluation procedure is derived by considering the end goal of the visualisation, which is to support decision-making in control room settings. In addition to presenting novel visualisations of traffic congestion that use bivariate glyphs to represent both the duration and extent of a traffic congestion cluster, our work also contributes to the visual analytics literature by exploring how the choice of data processing algorithm influences resulting visualisations.

This article proceeds as follows. In Section 2, we provide an overview of previous work to identify street network congestion, with a focus on studies that use visual analytics to support operational decision-making. In Section 3, we provide details of our analytical approach, including the clustering algorithm adopted, our visualisation design, and the metrics used within our evaluation procedure. In Section 4, we present the results of this procedure and compare the performance of clusters generated via Euclidean distance metrics with those generated using network distances. Finally, in Section 5, we conclude our study and discuss further opportunities for research.

2. Related work

Traffic congestion is a well-studied phenomena due to its importance in urban planning and transportation systems. Availability of large scale city-wide traffic data, via GPS sensors [LYW*17], automated traffic detectors (e.g. via inductive loops installed into the road surface, [LKJ*20]), and image processing (e.g. via automatic number plate recognition, [CTBH13]), have resulted in a number of data-driven studies of road congestion. In comparison to traffic de-

tectors that monitor traffic from a static fixed point, the use of GPS devices to monitor vehicle trajectories and to derive vehicle speeds can provide more comprehensive spatial coverage within a dense urban street network. Such approaches vary, but usually rely on a relatively small number of 'probe' vehicles—typically vehicles related to public transportation such as taxis or buses where GPS traces are made available as open data—to provide data that can serve as a proxy for all road users. The use of GPS data, however, is not without its problems such as GPS signals failing in heavily built-up areas or in tunnels. Nevertheless, GPS data can provide a cheap and effective approach that does not rely on installing expensive speed monitoring equipment [KCPL18].

From such data, estimates of road congestion are derived. One approach focuses on partitioning the study area into individual road segments across distinct time intervals. Within each time-interval and for each road segment, the average speed of vehicles along the segment is determined. When this average speed becomes low in comparison to historical distributions, then a congestion event has occurred. Such approaches, as used in [LYW*17, LKJ*20], are well-suited to situations where data is plentiful along the street segments of interest, for example from automated traffic detectors which measure vehicle speed from a static fixed point. However, as outlined in [YLCZ19] and [SWT*21], average speeds of probe vehicles may be reduced for reasons other than congestion, including via temporary traffic signals, bus stops or common taxi pickup areas, or via parking on the street in question.

Some recent approaches have instead adopted a density-based approach to defining and measuring traffic congestion, in which congestion is defined as areas with a high-density of slow-moving vehicles [AAH*11, SWT*21]. Relying on multiple vehicles being in close proximity provides some safeguards against spurious identification of congestion from a small number of probe vehicles. Density-based clustering algorithms can be adjusted to incorporate network-based distances [YM04] and several recent approaches place an emphasis on the structure of the street network. For example, in [ZXL*22], the authors use the structure of the street network to consider phenomenon such as bottlenecks. In their clustering procedure, [SWT*21] use network distance in addition to several other amendments including map-matching and moving object orientation alignment to ensure the identified clusters reflect the underlying dynamics as close as possible. Some specific network-based clustering procedures have also been introduced [ZHK18, WRLT19, NNB*23], although these are not presented within a dynamic cluster (i.e. temporally-varying) context.

The design space that has been explored to date for visualising traffic congestion is diverse [CGW15] but recent examples of visualising real-time traffic congestion favour network-based visualisation approaches (e.g. [KCPL18]), perhaps in part due to their common use by commercial online traffic services such as Google Maps (https://maps.google.com/) and Waze (https://www.waze.com/), although the latter of these uses icons to indicate road closures, possible hazards, and roadworks that might impact journey times.

Other approaches focus on providing congestion information in real-time [AAF*15, GSV*18, YYC*19], with the goal of informing operators about the current state of congestion on the roads. Considerations for real-time systems include the ability to perform data processing and aggregation within short time-scales as well as providing a visualisation system that can respond quickly to changes.

3. Data and methods

3.1. UK Bus Open Data

Data is provided by the UK Department for Transport Bus Open Data Service (BODS) [Dep], which provides vehicle location data for buses in England, updated every 10 seconds (although the GPS devices situated on each bus only update every 10 to 30 seconds [Dep20]). Our study area is in Central London. In Greater London as a whole, bus journeys are estimated to comprise over 20% of road user journeys [Tra23], indicating the prominence of buses on London's street network. Despite the use of dedicated bus lanes, and the requirement to frequently stop for short time periods, tracking bus speeds in near real-time can be indicative of broader congestion and disruptions on the road network. However, it is important to recognise that bus routes do not travel on all available roads; they are usually restricted to main arterial routes (i.e. A and B roads).

Data is collected from the BODS API every 60 seconds, providing a snapshot of the latest bus locations. The data is loaded into a Neo4j graph database (Neo4j Enterprise 5.15), using a schema that updates journeys if a new observation is found in each snapshot of data. For this purpose, a journey is defined as the same vehicle travelling on a single route on the same inbound or outbound direction. A graph database was chosen in order to implement efficient graph-style calculations over the data, including the calculation of network distances. A depiction of our data model in graph format is shown in Figure 2. Observation data comprises of geographic coordinates, a timestamp and a vehicle-journey identifier. The average speed value between observations is calculated. In total, there are 37.7 million observations, representing 1.4 million vehicle-journeys, used in our analysis. Our data model also includes a representation of the street network in Central London. Data from Open Street Map (OSM, https://www.openstreetmap.org/) is loaded into the Neo4j graph database via the Python package OSMnx [Boe24].

3.2. Clustering procedure

Our clustering procedure is based on the DBSCAN algorithm [EKSX96], which was chosen since it can identify clusters of arbitrary shape and can distinguish between samples that appear within clusters and samples that are treated as noise. The ability to distinguish between clusters and noise is critical for our application, ensuring that a slow moving or stationary vehicle is not necessarily identified as being part of a cluster. Indeed, since buses stop frequently at bus stops and intersections, sometimes for longer than 60 seconds, this is an important feature. Furthermore, our objective in producing clusters as visual summaries of congestion is not to highlight every slow-moving region of traffic flow but instead to identify regions with problematic high densities that risk persisting for some time or cascading into further clusters in other parts of the street network. We do not incorporate the orientations of vehicles within our study, as is done in [AAH*11, SWT*21]. This is due to

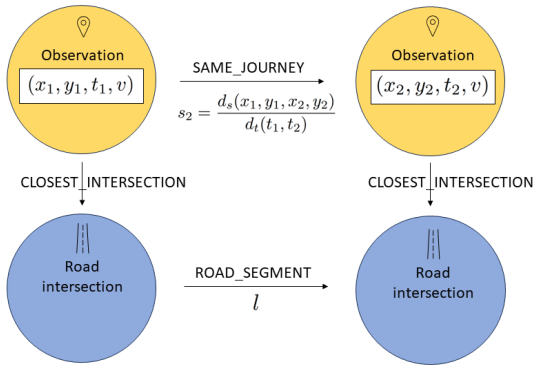

Figure 2: *A graph representation of the data model. Data comprises of geographic coordinates, represented by the points (x_i, y_i), a timestamp t_i, and a vehicle-journey identifier v. Ordered observations of the same vehicle-journey are linked within the database and statistics computed on the links between observations, such as speed s_2, as indicated. The road network is represented in the database as intersections linked by road segments, each with an associated length l. When calculating network distances, the observations are matched to their closest intersection.*

the orientation of vehicles potentially changing significantly within the 60 seconds between observations, meaning accurate estimates cannot be obtained. Central London is also a dense urban street network where disruptions can quickly propagate via intersections across different directions.

We define a congestion cluster as a group of slow-moving buses that are near in both space and time. We introduce a parameter μ_s, defined as the maximum speed for which a bus is considered to be slow-moving. The data is then filtered so that we only consider observations whose estimated speed (as derived from the previous observation from that vehicle-journey) is less than μ_s. The DBSCAN model identifies clusters as areas of high density, in which there are at least *minPts* within radius ϵ of at least one *core point*. Core points within *neighbourhoods* of other core points are allocated to the same cluster, and points within ϵ of all core points are called *border points* and also allocated to the cluster. DBSCAN can also be extended to handle clusters in both space and time (e.g. [BK07]), leading to the required identification of two epsilon parameters: ϵ_s in the spatial domain and ϵ_t in the temporal domain. Neighbourhoods are then defined using both spatial and temporal parameters as thresholds. That is, for an observation comprising of a two-dimensional spatial location and a timestamp, given by $\mathbf{u} = (u_x, u_y, u_t)$, the neighbourhood of $\mathbf{u}$ is then defined as points $\mathbf{v}$ such that

$$\mathcal{N}(\mathbf{u}) = \{\mathbf{v} | d_s(u_x, u_y, v_x, v_y) < \epsilon_s \wedge d_t(u_t, v_t) < \epsilon_t\}. \quad (1)$$

In total, there is the choice of two distance functions, d_s and d_t, and four parameters, μ_s, *minPts*, ϵ_s and ϵ_t, in the implementation of our clustering algorithm. For d_t, we use the absolute difference between the two timestamps, measured in seconds. For d_s, we consider two alternative specifications:

1. Euclidean distance, d_E, implemented by projecting GPS latitude and longitude coordinates to a projected coordinate system and then computing Euclidean distances between projected coordinates. Euclidean distance metrics are often used as default metrics in spatial clustering (for example, as in the DBSCAN implementation of the popular scikit-learn library) and we use this metric as our baseline. An alternative baseline is to use great circle distance, however the maximum possible difference between the two approaches within our study area within central London is less than 1.5m (while differences between actual distances computed between observations will be much less), so these approaches are considered equivalent for the present study.
2. Network distance, d_N, where for points $\mathbf{u}$ and $\mathbf{v}$,

$$d_N(\mathbf{u}, \mathbf{v}) = d_E(\mathbf{u}, \mathbf{n}_1) + \hat{d}_N(\mathbf{n}_1, \mathbf{n}_2) + d_E(\mathbf{n}_2, \mathbf{v}), \quad (2)$$

where $\mathbf{n}_1$ is the location of a node on the street network closest to point $\mathbf{u}$, $\mathbf{n}_2$ is the location of a node on the street network closest to point $\mathbf{v}$, and $\hat{d}_N$ is the network distance between two nodes as obtained from Dijkstra's algorithm, implemented in our study using Neo4j's Graph Data Science Library. Two separate versions of the street network are utilised: a simplified network and a full network. The simplified network reduces the number of nodes and edges by removing all nodes obtained from OSM that are neither an intersection nor a dead-end. The full network contains all available nodes and has the effect of potentially reducing the values of d_E within equation 2.

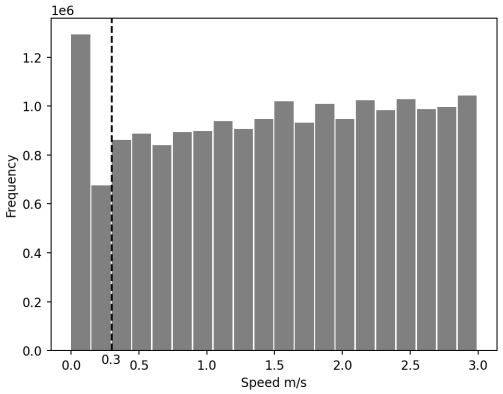

Figure 3: *Histogram of speed values, as calculated via linked space-time observations within vehicle-journeys. $\mu_s = 0.3ms^{-1}$ is selected as a point distinguishing two regimes of behaviour.*

We set $\mu_s = 0.3ms^{-1}$, which was derived by observing an minimum in a histogram of observations at this point (see Figure 3), suggesting two distinct regimes of behaviour. We then fix parameters ϵ_t and *minPts* using our knowledge on the constraints associated with the data generating process. Since we have observations of buses every minute, we wish to ensure that ϵ_t is large enough to detect multiple observations of the same bus. In the case of detecting congestion, repeated observations of a single vehicle can be used to our advantage since if it is in close spatial proximity to its previous observations, then there is an increased chance that a congestion event has occurred. On the other hand, we wish to ensure

ϵ_t is not so large that a single vehicle with multiple repeated observations could generate its own congestion cluster. To do this, we set $\epsilon_t = 300$ and $minPts = 10$. This choice ensures that at least one observation requires at least 9 samples within its neighbourhood before a cluster can be formed. With $\epsilon_t = 300$, only a maximum of 4 other observations from the same vehicle in the same neighbourhood are possible and we therefore require a further 5 samples in spatio-temporal proximity before a cluster can be formed, which must come from at least one other vehicle.

Our visual analytics system then interacts with the clustering procedure via the following steps:

1. For the study area at time T, for the period $[T - \delta T, T)$, identify all observations within the database that travelled at a speed less than μ_s and record their positions, times, and vehicle identifiers.
2. Implement DBSCAN on the resulting data, with ϵ_s and ϵ_t as neighbourhood thresholds in the spatial and temporal dimensions, and $minPts$ as the minimum number of observations within a neighbourhood for a new cluster to be formed.
3. For each cluster identified, calculate its geographic centre point based on the spatial positions of all observations included in the cluster. Identify the start and end times of each cluster by the recorded position times of the first and last observations included in the cluster and count the number of distinct vehicles represented in the cluster. Return the cluster duration and the number of distinct vehicles to be plotted using the bivariate vizent glyphs.

Our implementation makes use of scalable cluster analysis techniques [PFV*12]. This method discretises the temporal dimension of the study area into a series of *frames* and performs clustering within each frame. The construction of the frames includes sufficient overlaps such that clusters appearing in consecutive frames can be matched on the basis that they would have been in the same cluster had the clustering procedure been run on the full data. This implementation has two consequences. First, it means our algorithm is capable of running over long time periods, with relatively small compute requirements. Second, it makes the implementation on historical data parallelisable. A similar approach to splitting up the study area (although this time in the spatial dimension) is adopted in [YYC*19] to provide real-time clustering. From an operational perspective, this approach means that small batches can be incorporated into the clustering procedure as soon as the data becomes available for the next frame, making the clustering possible in near real-time. We discuss this as a possible extension in Section 5.

3.3. Visualisation

Due to extensive overlapping and intersections, visual summaries of high volume trajectory data are necessary to understand movement patterns [AA11]. Moreover, reducing visual clutter can also reduce the cognitive load on the viewer. Our visualisation design seeks to achieve this by highlighting geographic areas of congestion using the geographic centre points of identified clusters. An example is shown in Figure 4, where the temporal evolution of the congestion is depicted using small multiples of a geographic area of interest. In particular, our visualisation design plots cluster centres (taken as the mean location of points making up the cluster) in geographic space as bivariate visual entropy glyphs [HcF*24]. The use of these glyphs allows us to incorporate two channels of information for each cluster, without needing to increase the glyph size, which would lead to excess occlusion. Experimental work described in [HcF*24] suggests that these glyphs can be used for fast ordering of the secondary shape channel. Within the central disc of each glyph, we encode with colour the duration of the cluster, which indicates to a viewer how long the cluster has persisted. Within the outer part of glyph, we encode via the shape frequency an indication of the number of unique vehicles forming each cluster. The associated vizent Python library [MB23] is used to generate these glyphs overlaid on map tiles provided by Mapbox (`https://www.mapbox.com/about/maps/`) using data from OpenStreetMap (`http://www.openstreetmap.org/copyright`).

3.4. Performance Metrics

Measuring the performance of clustering procedures in the context of traffic congestion is not trivial since there is no ground truth: the spatial and temporal extents of congestion do not admit precise definitions. This difficulty is reflected in the different approaches taken in previous work to evaluate clustering procedures. One approach is to measure cluster coherency, using statistics such as the silhouette score [WRLT19] or the Davies-Bouldin index [NNB*23]. Such measures require computing a single value of distance between every pair of observations. However, such a pairwise distance calculation was not required in the implementation of our DBSCAN clustering algorithm. This is because spatial and temporal distances were computed separately and used in a sequential manner, comparing the resulting distances to ϵ_s and ϵ_t respectively to define neighbourhoods. For statistics that require a single distance metric, a weighting between the spatial (measured in metres) and temporal (measured in seconds) distances used in our clustering procedure would be necessary, which is difficult to determine. Other approaches to cluster validation include comparison with aggregate data on traffic speeds [CLLK17], alternative algorithm specifications [YLCZ19], and through the use of simulated data [SWT*21]. To achieve our goal of evaluating clustering procedures for their use in control room settings, and their ability to faithfully reflect the real congestion situations on the road network, we opt against using simulated and/or aggregated data. Similarly, comparing our proposed procedure with the performance of an alternative algorithm gives a biased view in the context of the chosen algorithm with which to compare. For this reason, we propose a novel evaluation procedure, which we explain in what follows.

To ensure our performance metrics relate to our operational challenge of providing visual summaries of traffic congestion in a control room setting, we develop three performance measures for our clustering procedure. Our first measure, which we term *cluster purity*, is derived as the proportion of all observations in proximity to a cluster centre that are also slow-moving during the cluster lifespan. That is, for a cluster with centre point $\mathbf{c} = (c_x, c_y)$ which is active from t_{min} to t_{max}, we first obtain the set $\mathcal{D}$ defined as observations $\mathbf{o} = (o_x, o_y, o_t, o_s)$ such that:

$$\mathcal{D} = \{x | d_s(c_x, c_y, o_x, o_y) < \nu \wedge t_{min} \leq o_t \leq t_{max}\}, \quad (3)$$

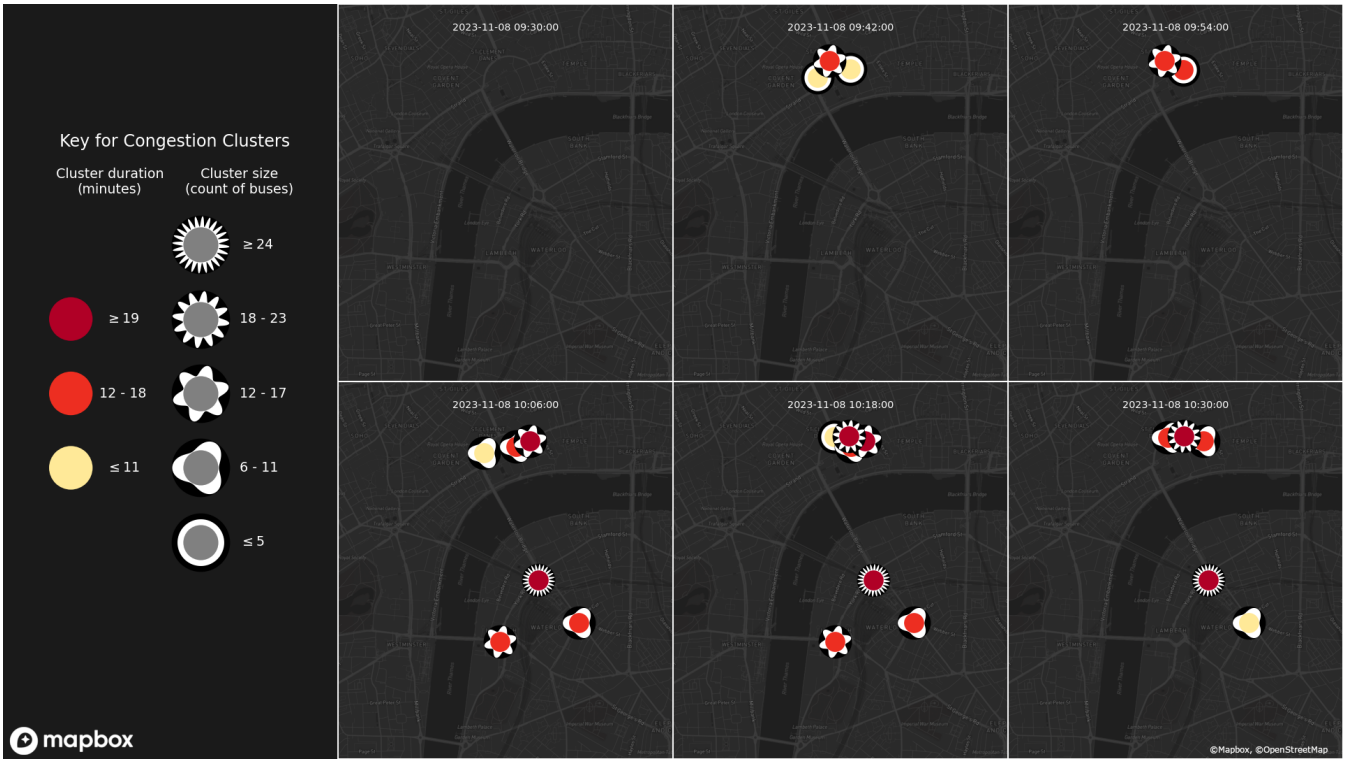

Figure 4: *A series of small multiple images, showing the evolution of a set of congestion clusters over time. The cluster centre points are plotted using vizent glyphs described in [HcF* 24]. To mitigate against occlusion, the clusters with the largest number of unique vehicles are plotted on top of small clusters.*

where $\mathbf{o} = (o_x, o_y, o_t, o_s)$ represents an observation from the BODS database at spatial position o_x, o_y, at time o_t, travelling at speed o_s. The set $\mathcal{D}$ comprises all observations in proximity to the cluster centre prior to filtering on speed with μ_s, which was the first step of the clustering algorithm described in Section 3.2. Then, cluster purity is defined as:

$$CP = \frac{||\mathbf{o} \in \mathcal{D} | o_s < \mu_s||}{||D||}. \tag{4}$$

The value in equation 4 gives the proportion of all observations in close proximity to the cluster (defined by the set $\mathcal{D}$) that are slow-moving. Cluster purity operates as an effective evaluation metric because it incorporates the data that was filtered and therefore excluded from the first step of the clustering procedure. It can be interpreted as a measure of epistemic uncertainty since it captures the extent to which clusters are genuinely measuring congestion. That is, values of CP close to one indicate that the majority of observations in proximity to the cluster were also slow-moving, and were therefore likely affected by the congestion. On the other hand, values of cluster purity close to zero indicate that a majority of traffic was still moving freely, despite a congestion cluster having been identified. Higher values of cluster purity indicate a true blockage or congestion event where limited traffic can travel freely. For the purposes of the present study, we select a value of ν as $200m$. While smaller values of ν will likely lead to high values of cluster purity, this value was chosen to ensure a sufficient number of observations were included in each cluster purity calculation.

Our second and third evaluation measures are designed to compare what would be observed in a control room setting (where future information on the evolution of the clusters is unknown) with a ground truth version that contains all relevant future data. To do this, we run a clustering procedure for every 15-minute interval in the study period. That is, at time T, we use BODS data for the period $[T - \delta T, T)$, where δT is chosen to be 2 hours, a value selected since it is larger than the maximum typical life-span of any clusters identified. We call these *viewing time procedures* since they emulate what would have been viewed from within a control room. We use as a *ground truth* the clustering applied over the full study period. For each cluster identified within the *viewing time procedures*, we perform a matching procedure, matching to ground truth clusters that were present at the viewing time and whose centers were within 50m of the viewing time procedure cluster center. Then, for each matched cluster i, which is present in the ground truth for the interval $[t^i_{min}, t^i_{max}]$, and for viewing time T such that $t^i_{min} \leq T \leq t^i_{max}$ we define

$$\tau_i = t^i_{max} - T, \tag{5}$$

which is the duration in seconds between the viewing time T and the maximum time for which the cluster persists into the future. We

then take

$$\tau(T) = \sum_i \tau_i, \tag{6}$$

for all clusters i that are matched at time T. $\tau(T)$ represents the overall lifespan of clusters following the identification of a cluster in a control room. We call this value the *post-view cluster duration*, which is our second evaluation measure. Higher values indicate that once clusters have been identified by the clustering procedure, they are likely to persist for longer. Clustering procedures with higher post-view cluster duration values are more desirable in a decision-making context because it means that decisions taken are more likely to impact an ongoing congestion incident on the road. If clusters do not persist for very long (corresponding to low values of τ), then decisions might be taken when the congestion would have soon cleared without any intervention, thereby potentially wasting valuable resources.

Our third measure uses the same quasi-experimental setup by matching clusters from *viewing time procedures* with a derived *ground truth*. In this case, for each matched cluster, we identify the number of distinct vehicles that join each cluster after it has been identified. We add to this the number of vehicles that contribute to congestion clusters in close spatial proximity (within 500m) within the next 60 minutes to obtain a measure of the potential for wider impact of an identified cluster. We call this measure the *post-view cluster vehicle count*. This final measure is necessary because identified congestion clusters have the potential for cascading across the network, producing additional congestion points at nearby locations in the near future. Higher values of the *post-view cluster vehicle count* measure indicate that more vehicles are likely to be impacted by each congestion cluster. From a decision-making perspective, decisions might be prioritised according to the expected number of vehicles likely to be impacted.

4. Results

To select an appropriate value of ϵ_s for each distance metric, we rerun the clustering procedure for different values of ϵ_s for the period on 8th November 2023, between 06:30 and 13:00, during which congestion occurred due to the Just Stop Oil protest on Waterloo bridge. The average cluster purity was measured together with the total number of clusters (Figure 5). Based on the largest increases in cluster purity, while still retaining a significant number of clusters, we selected $\epsilon_s = 25$ for the Euclidean distance metric and $\epsilon_s = 50$ for the network distance metric. These values are used in the results that follow. In this parameter selection task, we only included simplified street networks, as described in Section 3.2.

Average cluster purity values for the different distance metrics across the entire study period from 1st-14th November 2023 are presented in Table 6, while a rolling average of cluster purity over time is shown in Figure 7. An improvement in cluster purity when using network distance metrics in comparison to Euclidean distances can be seen, equating to around a 6 to 9 percentage point increase. Based on our definition of cluster purity, clusters identified by the network distance metric are more likely to reflect *genuine congestion*, in which a greater proportion of vehicles in proximity to the cluster center are also impacted by the congestion. In other words, the network distance metric produces clusters with less epistemic uncertainty. We also note that cluster purity increases when using the full street network over the simplified version. In our experiments, clustering over the two-week study period for the full network took 52 minutes versus 21 minutes for the simplified network using an Intel i7-12700H, with 64GB RAM. Clustering with Euclidean distances took 32 minutes. It is interesting to note comparable computation speeds for network distances versus Euclidean distances, although optimisations might be possible in both cases (see Section 5). However, due to increased computation time, it may not be feasible to use the full network in operational settings. Nevertheless, the cluster purity results taken in isolation confirm our hypothesis that a better clustering procedure can be obtained with a more accurately defined distance metric.

Distance metric	ϵ_s	Cluster purity	Cluster count
Euclidean	100	0.21	85
Euclidean	75	0.23	77
Euclidean	50	0.27	67
Euclidean	25	0.44	33
Euclidean	15	0.47	12
Network (simplified)	100	0.29	62
Network (simplified)	75	0.36	41
Network (simplified)	50	0.46	38
Network (simplified)	25	0.57	5
Network (simplified)	15	NA	0

Figure 5: *Average cluster purity and cluster counts for 06:30 - 13:00 on 8th November 2023 for different values of ϵ_s.*

Results for mean post-view cluster duration and mean post-view cluster vehicle count are also presented in Table 6. For the simplified network, we find that clusters derived using network distance are likely to persist for a further 69.9 seconds on average, and lead to 2.22 additional vehicles being identified in a cluster for the next 1-hour period. This can be compared to 57.8 seconds duration and 1.68 vehicles for the Euclidean metric. For the full network, these values increase 102.8 seconds for duration and to an additional 3.05 vehicles. This suggests that clusters identified with network distance metrics are more likely to involve a greater number of vehicles in the short term, meaning that any actions taken as a result of identifying these clusters from within a control room are likely to have a larger impact. A time series version of these metrics is also presented in Figure 8. A similar pattern is observed for both Euclidean and Network distances, with few clusters identified overnight during the study period. However, with a small number of possible exceptions, we see larger values on average in the network metric than the Euclidean metric (as evidenced by the values in Table 6), emphasising a consistent improvement in the use of network metrics over this time period.

5. Conclusion and discussion

We have presented a visual analytics approach to summarising city-wide road traffic congestion via space-time clusters, with the goal of providing a near real-time understanding of traffic conditions for situations such as a control room. We have compared Euclidean dis-

Distance metric	Cluster count	Cluster duration (s)	Vehicles per cluster	Cluster purity (%)	Post-view duration	Post-view 1-hour nearby vehicle congestion count
Euclidean 25*m*	503	672	8.43	33.0	57.8	1.68
Network 50*m*, simplified	536	686	8.04	39.1	69.9	2.22
Network 50*m*, full network	846	668	7.86	42.5	102.8	3.05

Figure 6: *Table of summary statistics and average evaluation metrics, taken every 15 minutes for the period 1-14 November 2023.*

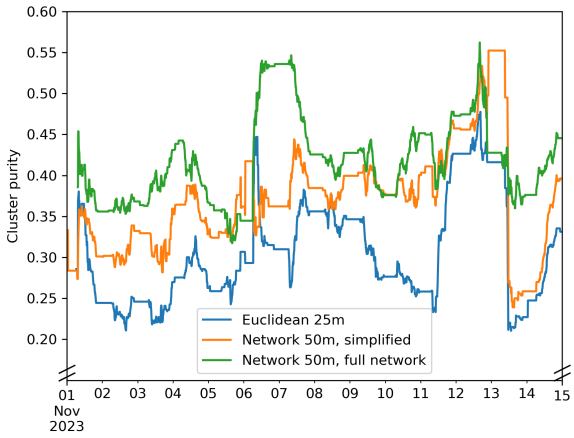

Figure 7: *A 24-hour rolling average of cluster purity over the duration of the study period (1-14 November 2023) for the three distance metrics in Table 6.*

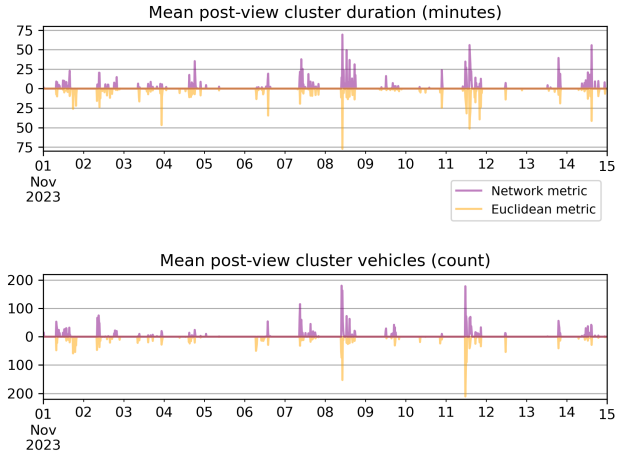

Figure 8: *A time series of post-view metrics used to assess network and Euclidean distances. Top: Lifespan of clusters following control room identification. Bottom: Count of vehicles in clusters within 500m of the identified cluster within the next hour.*

tance with network distance and shown that network distances provide advantages by generating clusters with higher values of *cluster purity*, a performance metric designed to capture the severity of an identified cluster and its associated epistemic uncertainty. Our results also show that clusters generated using network distance metrics are more likely to generate clusters that last longer and which impact more vehicles, which, we argue, is beneficial for control room operators who will not wish to invest time and resources being alerted to incidents that are more likely to be resolved sooner. Indeed, a course of action available to a decision-maker in a control room should be executed during the lifespan of an incident, which, as our results show, is more likely to occur with network distances.

Network distance metrics are more likely to accurately reflect the underlying dynamics on the street network within which vehicles are constrained. This, we argue, explains our findings. It is important to recognise that in some cases congestion events can propagate through the network. We believe that network distance metrics can more accurately capture such propagation, and hence are better for identifying traffic congestion. Our findings have important implications for traffic planners and developers of intelligent transportation and traffic control room systems.

Extensions to this work can be explored in multiple directions. First, the Bus Open Data Service is not only available in Central London but also across England. More cities and governments are also increasing the amount of open transport data published. It would be of interest to compare distance metrics across different geographic settings with qualitatively different street networks and bus service schedules to validate the universality of our findings. Second, our evaluation procedure would be well-suited to comparing a broader range of clustering algorithms and we have highlighted several alternative clustering procedures, some of which could be extended to either a network distance setting, or to a space-time clustering setting. Third, additional steps can be taken to analyse and optimise the performance of our methods with a view to minimising the delay in providing congestion clusters for decision-making. One approach might be via real-time monitoring procedures (e.g. [AAF*15]). In our implementation, the use of Dijkstra's algorithm had a similar computation time to a Euclidean distance metric, but it may be possible to optimise further, for example via the approach of [ZHK18]. Finally, future work might address the visualisation challenges in control room settings via assessment of our visual design and encoding. Consideration of alternative variables for congestion clusters that are less likely to be as correlated as the variables we have selected might further validate our visual design.

References

[AA11] ANDRIENKO N., ANDRIENKO G.: Spatial Generalization and Aggregation of Massive Movement Data. *IEEE Transactions on Visualization and Computer Graphics 17*, 2 (2011), 205–219. doi:10.1109/TVCG.2010.44. 1, 5

[AAF*15] ANDRIENKO N., ANDRIENKO G., FUCHS G., RINZIVILLO S., BETZ H.-D.: Detection, tracking, and visualization of spatial event clusters for real time monitoring. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (Paris, France, 2015), IEEE, pp. 1–10. doi:10.1109/DSAA.2015.7344880. 3, 8

[AAH*11] ANDRIENKO G., ANDRIENKO N., HURTER C., RINZIVILLO S., WROBEL S.: From movement tracks through events to places: Extracting and characterizing significant places from mobility data. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)* (Providence, RI, USA, 2011), IEEE, pp. 161–170. doi:10.1109/VAST.2011.6102454. 1, 3

[Al 23] AL JAZEERA: Hundreds of thousands join largest march in london so far against gaza war, 2023. [Accessed 12-06-2024]. URL: www.aljazeera.com/news/2023/11/11/thousands-join-pro-palestine-march-in-london. 2

[BK07] BIRANT D., KUT A.: ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering 60*, 1 (2007), 208–221. doi:10.1016/j.datak.2006.01.013. 4

[Boe24] BOEING G.: Modeling and Analyzing Urban Networks and Amenities with OSMnx. Working paper. URL: https://geoffboeing.com/publications/osmnx-paper, 2024. 3

[CGW15] CHEN W., GUO F., WANG F.-Y.: A Survey of Traffic Data Visualization. *IEEE Transactions on Intelligent Transportation Systems 16*, 6 (2015), 2970–2984. doi:10.1109/TITS.2015.2436897. 3

[CLLK17] CHIANG M.-F., LIM E.-P., LEE W.-C., KWEE A. T.: BTCI: A new framework for identifying congestion cascades using bus trajectory data. In *2017 IEEE International Conference on Big Data* (Boston, MA, 2017), IEEE, pp. 1133–1142. doi:10.1109/BigData.2017.8258039. 5

[CTBH13] CHENG T., TANAKSARANOND G., BRUNSDON C., HAWORTH J.: Exploratory visualisation of congestion evolutions on urban transport networks. *Transportation Research Part C 36* (2013), 296–306. doi:10.1016/j.trc.2013.09.001. 2

[Dep] DEPARTMENT FOR TRANSPORT: Bus Open Data Service. www.bus-data.dft.gov.uk/. [Accessed 22-05-2024]. 1, 3

[Dep20] DEPARTMENT FOR TRANSPORT: Bus open data implementation guide. www.gov.uk/government/publications/bus-open-data-implementation-guide, 2020. [Accessed 22-05-2024]. 3

[EKSX96] ESTER M., KRIEGEL H.-P., SANDER J., XU X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (1996), AAAI, pp. 226–231. 1, 3

[GSV*18] GOMES G. A., SANTOS E., VIDAL C. A., COELHO DA SILVA T. L., MACEDO J. A. F.: Real-time discovery of hot routes on trajectory data streams using interactive visualization based on GPU. *Computers & Graphics 76* (2018), 129–141. doi:10.1016/j.cag.2018.09.008. 3

[HcF*24] HOLLIMAN N. S., ÇÖLTEKIN A., FERNSTAD S. J., MCLAUGHLIN L., SIMPSON M. D., WOODS A. J.: Entropy Ordered Shapes as Bivariate Glyphs. *Electronic Imaging 36*, 11 (2024), 206–1–206–10. doi:10.2352/EI.2024.36.11.HVEI-206. 1, 2, 5, 6

[Jus23] JUST STOP OIL: 8 November, 2023. www.juststopoil.org/2023/11/08/, 2023. [Accessed 22-05-2024]. 2

[KCPL18] KWEE A. T., CHIANG M.-F., PRASETYO P. K., LIM E.-P.: Traffic-Cascade: Mining and Visualizing Lifecycles of Traffic Congestion Events Using Public Bus Trajectories. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino Italy, 2018), ACM, pp. 1955–1958. doi:10.1145/3269206.3269216. 3

[LKJ*20] LEE C., KIM Y., JIN S., KIM D., MACIEJEWSKI R., EBERT D., KO S.: A Visual Analytics System for Exploring, Monitoring, and Forecasting Road Traffic Congestion. *IEEE Transactions on Visualization and Computer Graphics 26*, 11 (2020), 3133–3146. doi:10.1109/TVCG.2019.2922597. 2, 3

[LYW*17] LIU Y., YAN X., WANG Y., YANG Z., WU J.: Grid Mapping for Spatial Pattern Analyses of Recurrent Urban Traffic Congestion Based on Taxi GPS Sensing Data. *Sustainability 9*, 4 (2017), 533. doi:10.3390/su9040533. 2, 3

[MB23] MCLAUGHLIN L., BAUDAINS P.: vizent. https://pypi.org/project/vizent/, 2023. Version 1.1.2, [Accessed 20-12-2023]. 5

[NNB*23] NGUYEN T. T., NGUYEN L. T., BUI Q.-T., YUN U., VO B.: An efficient topological-based clustering method on spatial data in network space. *Expert Systems with Applications 215* (2023), 119395. doi:10.1016/j.eswa.2022.119395. 3, 5

[PCRHS18] PADILLA L. M., CREEM-REGEHR S. H., HEGARTY M., STEFANUCCI J. K.: Decision making with visualizations: a cognitive framework across disciplines. *Cogn. Research 3*, 1 (2018), 29. doi:10.1186/s41235-018-0120-9. 1

[PFV*12] PECA I., FUCHS G., VROTSOU K., ANDRIENKO N., ANDRIENKO G.: Scalable Cluster Analysis of Spatial Events. *EuroVA 2012: International Workshop on Visual Analytics* (2012). doi:10.2312/PE/EUROVAST/EUROVA12/019-023. 5

[SWT*21] SHI Y., WANG D., TANG J., DENG M., LIU H., LIU B.: Detecting spatiotemporal extents of traffic congestion: a density-based moving object clustering approach. *International Journal of Geographical Information Science 35*, 7 (2021), 1449–1473. doi:10.1080/13658816.2021.1905820. 3, 5

[Tra23] TRANSPORT FOR LONDON: Consolidated estimates of total travel and mode shares, 2023 update. www.tfl.gov.uk/corporate/publications-and-reports/travel-in-london-reports, 2023. [Accessed 06-06-2024]. 3

[WRLT19] WANG T., REN C., LUO Y., TIAN J.: NS-DBSCAN: A Density-Based Clustering Algorithm in Network Space. *ISPRS International Journal of Geo-Information 8*, 5 (2019), 218. doi:10.3390/ijgi8050218. 3, 5

[YLCZ19] YU Q., LUO Y., CHEN C., ZHENG X.: Road Congestion Detection Based on Trajectory Stay-Place Clustering. *ISPRS International Journal of Geo-Information 8*, 6 (2019), 264. doi:10.3390/ijgi8060264. 3, 5

[YM04] YIU M. L., MAMOULIS N.: Clustering objects on a spatial network. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data* (Paris France, 2004), ACM, pp. 443–454. doi:10.1145/1007568.1007619. 3

[YYC*19] YANG Q., YUE Z., CHEN R., ZHANG J., HU X., ZHOU Y.: Real-time detection of traffic congestion based on trajectory data. *The Journal of Engineering 2019*, 11 (2019), 8251–8256. doi:https://doi.org/10.1049/joe.2019.0872. 3, 5

[ZHK18] ZHANG Y., HAN L. D., KIM H.: Dijkstra's-DBSCAN: Fast, Accurate, and Routable Density Based Clustering of Traffic Incidents on Large Road Network. *Transportation Research Record: Journal of the Transportation Research Board 2672*, 45 (2018), 265–273. doi:10.1177/0361198118796071. 3, 8

[ZWC*16] ZHENG Y., WU W., CHEN Y., QU H., NI L. M.: Visual Analytics in Urban Computing: An Overview. *IEEE Transactions on Big Data 2*, 3 (2016), 276–296. doi:10.1109/TBDATA.2016.2586447. 1

[ZXL*22] ZENG J., XIONG Y., LIU F., YE J., TANG J.: Uncovering the spatiotemporal patterns of traffic congestion from large-scale trajectory data: A complex network approach. *Physica A: Statistical Mechanics and its Applications 604* (2022), 127871. doi:10.1016/j.physa.2022.127871. 3