Dynamic phase and group detection in pedestrian crowd data using multiplex visibility graphs

Poole, N. , Mo, T. and Brusey, J.

Original citation & hyperlink:
Stephen, C. (2015) Dynamic phase and group detection in pedestrian crowd data using multiplex visibility graphs. Procedia Computer Science, volume 53 : 410–419.
http://dx.doi.org/10.1016/j.procs.2015.07.318

Note:
Paper presented at the INNS Conference on Big Data 2015 Program San Francisco, CA, USA 8-10 August 2015.

Procedia Computer Science

Volume 53, 2015, Pages 410–419

2015 INNS Conference on Big Data

Dynamic Phase and Group Detection in Pedestrian Crowd Data Using Multiplex Visibility Graphs

Colin Stephen[1]

Department of Computing
Faculty of Science and Engineering
Coventry University, UK
colin.stephen@coventry.ac.uk

Abstract

We study pedestrian crowd dynamics and the detection of groups in a scene. We propose a novel method to analyse pedestrian trajectories by translating them to multiplex networks, whose properties can be studied using the tools of graph theory. Our results show that simple measures on the resulting multiplex graphs accurately reflect both the global dynamics and local clustering within scenes.

Keywords: crowd data, dynamical systems, visibility graph, clustering

1 Introduction

Automated crowd data analysis is used to monitor environments where individuals and groups interact. Particular applications include maintaining crowd safety during large sporting events, and monitoring usage patterns of public spaces such as a car parks or urban intersections. Two key challenges faced by crowd monitoring systems are: (1) robust identification of groups within a scene, including detecting their formation and dispersal; (2) detection of unusual movement patterns, including fights and accidents, that may erupt.

Here we implement a new method for detecting groups and anomalous movement patterns from possibly noisy pedestrian trajectory data, which relies on just one core data structure: a directed graph. We assume that a reliable algorithm for detecting people exists, and whose time-varying positions are the input to our method.

1.1 From time series to graphs

Horizontal visiblity graphs (HVGs) have recently been applied to translate univariate time series data in to graphs [4, 11, 14]. In the field of dynamical systems, the resulting graphs have been shown to preserve many of the important structural features of the original time series, enabling standard graph measures such as degree distribution, clustering coefficient,

Selection and peer-review under responsibility of the Scientific Programme Committee of INNS-BigData2015

doi:10.1016/j.procs.2015.07.318

and mean path length, to be used to analyse the time series: [7, 8, 10, 12]. Generalising to multidimensional time series results in multiplex visibility graphs (MVGs) [6, 16]. These are vectors of HVGs, whose components correspond to the graphs in the component time series. Multiplex graphs have well-defined (multivariate) measures, such as graph overlap and interlayer mutual information, which can give insight in to the dynamics and correlations underlying the original data set [4, 16]. These measures are what we explore below.

1.2 Studying crowd motion using graph theory

Given a crowd of M interacting agents represented as $2M$ positional time series $\mathbf{x}_t^{[\alpha]} = (x_t^{[\alpha]}, y_t^{[\alpha]})$ where $\alpha = 1, 2, \ldots, M$, we apply the following sequence of steps: (1) translate trajectory time series data to directed multiplex visibility graphs; (2) calculate average edge overlap as a coarse measure of coherent motion in the scene, corresponding to the "dynamical phase" of the crowd; (3) extract pairwise and average inter-layer degree correlation information to measure global and local "group associations" within the scene.

We apply the multiplex measures to complete scenes and also to fixed-length moving windows, to monitor changes over time. The main finding in Section 3 is that changes to the average edge overlap within a scene's MVG signals a transition from coherent motion to turbulent motion, such as when a fight erupts. In Section 4 we show that interlayer information flow across a scene's MVG captures group associations, a means to track fusion and splitting.

2 Multiplex Graphs of Scene Data

Our test data consist of pedestrian trajectories for three scenes in the public BEHAVE behaviour classification dataset [5]. We define a scene to be a period of time during which the number of visible agents in the video frame is constant. All extracted scenes are long enough to contain a variety of diffent behaviours and are visualised in Figure 1 for reference.

1. Meeting. Two groups approach one another from different directions, merge and stand together, then separate in to two (different) groups which walk away together. **792 frames**.

2. Fight. An individual and couple form a small group, another couple approaches and various fights ensue between different individuals before they split apart. **591 frames**.

3. Private discussion. An individual approaches a group and stands with it, then two separate groups split apart for a short time. The large group then reforms before an individual exits the frame. **1822 frames**.

2.1 A directed MVG construction for trajectories

A time series visibility algorithm is a map from a time series $\{x_t\}_{t=1}^N$ to a graph containing N vertices, with an edge between vertices i and j if the pair (x_i, x_j) satisfies a given 'visibility condition'. In the horizontal visibility algorithm, the visibility condition is particularly simple: (i, j) is an edge exactly when $x_k < \inf\{x_i, x_j\}$ for all x_k with $i < k < j$. The resulting graph is called a horizontal visibility graph (HVG).

In the multivariate version an M-dimensional time series $\{\mathbf{x}_t\}_{t=1}^N$ is mapped to a M-layer 'multiplex network' in which each layer corresponds to one of the M HVGs of the

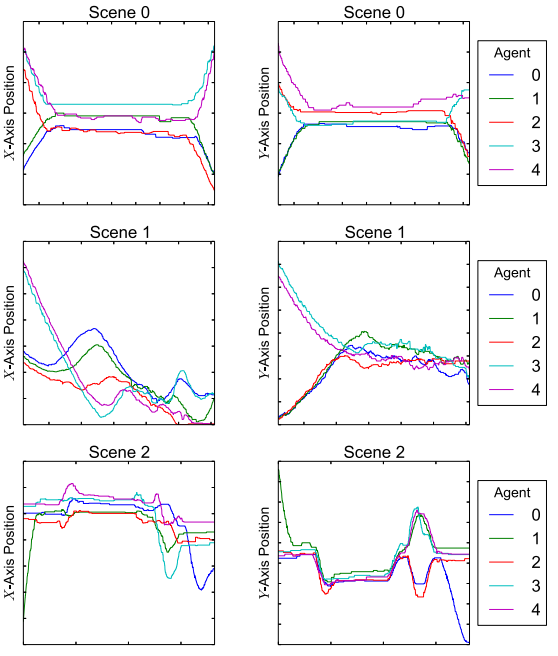


Figure 1: Agent trajectories in scenes, visualised as x (left) and y (right) time series.

components $\{x_t^{[\alpha]}\}_{t=1}^N$ where $1 \leq \alpha \leq M$. The vector of adjacency matrices of the layers, $\mathcal{A} = \{A^{[1]}, A^{[2]}, \ldots, A^{[M]}\}$, where $A^{[\alpha]} = \{a_{ij}^{[\alpha]}\}$ and $a_{ij}^{[\alpha]} = 1$ if and only if (i, j) is an edge in layer α, is known as the multiplex visibility graph (MVG) of the generating time series.

Additional information about a time series can be captured using a *directed* horizontal visibility graph (DHVG). Since our data represent a dynamical system, we will use directed edges to encode velocity information: source vertices will point to target vertices with lower values. In other words, we impose the additional condition that $x_i < x_j$ for (i, j) to be an edge in the graph.[1] This imposes an asymmetry in the adjacency matrix of the graph allowing us to distinguish upward from downward motion. While the magnitude of the velocity is lost, a useful signal, namely relative spatial direction between connected vertices, is retained.

2.2 DMVG construction and visualisation

The DMVG conditions in section 2.1 were applied to the x and y time series for agents visible in the chosen scenes. Each frame gave a vertex in the scene MVG, which was connected to others in the same layer by a directed edge when the HVG conditions hold. Figures 2a to 2c show the resulting adjacency matrices. Since the properties of the adjacency matrices of a graph provide insights in to its structure and dynamics [6], our method involves studying these properties.

For a full description of the horizontal visibility translation, the reader is referred to [11], but for the present purposes we give some intuition in to the resulting matrices. A mark appears in position (r, c) in the adjacency matrix exactly when the nodes representing times r and c are connected by an edge in the HVG. Since every HVG is connected via its underlying linear order on timestamps, we remove these redundant off-diagonal $(r + 1, r)$ elements for clarity.

[1] By convention, when $x_i = x_j$ then (i, j) is an edge only if $i < j$.

Scene	Axis	Vertices	DHVG Edge Counts	Adjacency Matrix Non-Zero Ratio (%)
0	x	792	799, 795, 817, 858, 908	0.127, 0.127, 0.130, 0.137, 0.145
	y	792	799, 802, 801, 838, 815	0.127, 0.128, 0.128, 0.134, 0.130
1	x	591	695, 716, 653, 750, 672	0.199, 0.205, 0.187, 0.215, 0.192
	y	591	677, 650, 647, 660, 679	0.194, 0.186, 0.185, 0.189, 0.194
2	x	1822	1924, 1897, 1882, 1909, 1880	0.058, 0.057, 0.057, 0.058, 0.057
	y	1822	1919, 1941, 1956, 1872, 1885	0.058, 0.058, 0.059, 0.056, 0.057

Table 1: Sparsity of adjacency matrices of the scene MVG layers, calculated as a % ratio $|E|/|V|^2$. Dominance of zero entries ensures graph computations are time-efficient.

The remaining edges are a "signature" for the time series, where edges close to the diagonal indicate *local minima* for the given axis, and edges off-diagonal indicate wider-ranging *global minima* for that axis.

2.3 Complexity of DHVG translation

Table 1 illustrates the sparsity of the computed adjacency matrices. This compresses the original data and speeds up subsequent calculations as we will see below. Moreoever, performing the translation to DMVGs is itself efficient and was one of the motivations for defining HVGs and MVGs in the first place [16]. Our implementation[2] creates a multiplex graph in a single run through the source time series data, making it $O(n)$ for each time series, and therefore $O(n)$ for any DHVGs with a bounded number of agents.

3 DMVG Edge Overlap and Chaotic Motion

The first measure we apply to the scene data is the *average edge overlap* of the DMVG $\mathcal{A}$:

$$\langle o \rangle := \frac{1}{K} \sum_{i,j} o_{ij}, \qquad o_{ij} := \frac{1}{M} \sum_{\alpha} a_{ij}^{[\alpha]}, \tag{1}$$

where K is the total number of pairs of vertices connected on at least one layer of $\mathcal{A}$. Value $\langle o \rangle$ is a measure of how similar the edge connectivity patterns of the layers of $\mathcal{A}$ are, reaching value $\langle o \rangle = 1$ exactly when the layers are identical to one another.

When a scene contains mainly "coherent" motion so that its trajectories are relatively smooth, and when it also contains mainly individuals who are clustered in to groups, we can expect the average overlap measure to be relatively stable and also relatively close to its maximum value of 1. On the other hand, when a scene conatins largely "chaotic" motion with rapidly varying trajectories, or when it contains primarily individuals whose motions are not correlated by group memberships, we can expect the average overlap measure to be relatively unstable and also relatively far from its maximum value.

3.1 Complexity of edge overlap

Since an edge overlap calculation is effectively a matrix summation, the theoretical time complexity is $O(n^2)$, but in the context of DHVGs this is strongly mitigated by two factors: matrix

[2]The code used for the analysis in this paper is available from: https://gitlab.com/colinstephen/multiplex-visibility-graphs.

Scene	$\langle o \rangle_x$	$\langle o \rangle_y$	$\langle o \rangle_{xy}$
0	0.71585	0.81919	0.76752
1	0.50122	0.52838	0.51480
2	0.73982	0.75023	0.74503

Table 2: Full scene average edge overlap measures, given for individual axes and as an average. The fight Scene 1 has consistently lower edge overlap compared with the coherent Scenes 0 and 2, suggesting a heuristic threshhold of around $\langle o \rangle_{xy} \leq \langle o \rangle_{\mathbf{fight}} \approx 0.60$ to distinguish a chaotic fight scene.

sparsity (see Table 1), and the use of finite moving windows. In particular, matrix summation is bounded with respect to the number of non-zero entries in the matrix, which in our model cluster around the diagonal. Additionally, fixed moving window matrix addition has complexity $O(1)$ at each step, adding up to $O(n)$ for a full scene. The effective time complexity for a full scene analysis is therefore $O(\sqrt{2}n + n) = O(n)$ in the worst case.

3.2 Edge overlap scene analysis

Our calculations of average and moving-window edge overlaps on the scene trajectories, presented in Figure 3, establish a clear threshhold between the group interactions of Scenes 0 and 2, and the fight Scene 1.

Computing overlaps for a moving window of length 200 frames shows that setting an average overlap threshhold value of $\langle o \rangle_{\mathbf{fight}}^{[200]} \approx 0.60$ cleanly separates the dynamics of a chaotic fight scene from more regular and smooth pedestrian movements in other scenes. Despite the overlap measure being a relatively coarse indicator of trajectory correlations, it nevertheless allows us to propose a realistic heuristic for detecting chaotic or turbulent pedestrian motion in a scene: $\langle o \rangle_{xy}^{[200]} > 0.6$ when the scene is coherent, and $\langle o \rangle_{xy}^{[200]} \leq 0.6$ when the scene is chaotic.

Experimentation shows that the value of 200 used for the moving window length can be varied up and down considerably before the average overlap evolution diverges from the trends illustrated here so it is a robust measure. For example the full scene overlaps shown in Table 2 are well separated across the proposed bound of 0.6.

Figure 3 also shows that $\langle o \rangle_x$ and $\langle o \rangle_y$ vary widely from one another in Scene 1, but closely track one another in Scene 0 and Scene 2. This discrepancy highlights a reduction in correlations flowing between the x and the y trajectories defining the scene. It suggests that the HMM underlying the data has evolved away from one with a low covariance between the axes; in other words it has become more chaotic.

4 Mutual Information and Group Formation

We next use the scene DMVGs to cluster individuals in to groups. Our method does this efficiently by computing correlations between the sparse adjacency matrices describing the layers, and thereby avoids using computationally expensive approaches such as K-means clustering conditioned on density estimation [2].

The measure we apply is the *interlayer mutual information* of $\mathcal{A}$:

$$I_{\alpha,\beta} := \sum_{k^{[\alpha]}} \sum_{k^{[\beta]}} P(k^{[\alpha]}, k^{[\beta]}) \log \frac{P(k^{[\alpha]}, k^{[\beta]})}{P(k^{[\alpha]}) P(k^{[\beta]})}, \tag{2}$$

where $P(k^{[\alpha]})$ and $P(k^{[\alpha]})$ are the respective degree distributions of layers α and β, and $P(k^{[\alpha]}, k^{[\beta]})$ is the joint probability of finding a vertex with degree $k^{[\alpha]}$ on layer α *and* degree $k^{[\beta]}$ on layer β.[3] A useful representation of these pairwise measures is the *graph of layers* $\mathcal{G}$ of the MVG. In $\mathcal{G}$ the vertices correspond to layers and are connected by edges with weights $\{I_{\alpha,\beta}\}$.

Intuitively, the interlayer mutual information measures how far two layers' degree distributions are from being mutually independent; in other words, how far the joint degree distribution varies from being the product of its marginals. Higher values of $I_{\alpha,\beta}$ therefore indicate stronger correlations between layers α and β.

4.1 Complexity of information flow

Computing the interlayer information flow for a pair of layers primarily relies on computing the joint probability distribution of degrees in each layer over each vertex, along with the marginal probabilities of the degrees within a layer. All of these can be done in $O(n)$ time since the number of edges in our sparse graphs is $O(\sqrt{2}n) = O(n)$.

The double sum over vertex degrees in Equation 2 suggests that the total time complexity may be as much as $O(m^2 n)$ where $m = \max_{\alpha,\beta}(k^{[\alpha]}, k^{[\beta]})$ is the maximum number of vertex degrees present in one of the layers, assuming every degree below this maximum value is possible. However, once again the actual situation is mitigated from this theoretical extreme. In particular, Luque et al. [14] show that for HVGs generated by time series generated from uniform, Gaussian, or power law distributions, the probability of a degree k at a node in the HVG decreases exponentially as k increases. It is therefore safe to assume that the HMM underlying the trajectory data of a group of pedestrians generates an HVG whose number of node degrees increases as $O(\log n)$ at worst. Thus the overall complexity of the information flow is bounded by $O(n \log n)$, which is comparable with the best clustering algorithms [1].

4.2 Moving window diameter

As in the case of edge overlap for scene dynamics, a pragmatic choice of window diameter is required for a robust analysis. We use a window size of 75 frames here, equating to 3 seconds of motion. This provides a good balance between resilience to noise and over-smoothing the correlation, as the numerical results show. Full optimisation of window length is likely to depend on the context of the analysis.

4.3 Mutual information scene analysis

Applying the mutual information measure to DMVGs corresponding to a scene does indeed extract the correct clusters, as can be seen in Table 3 and Figure 4, where we present the inter-layer correlations as weighted edges in the graph of layers for Scene 0.

The graphs of layers in Figure 4 are calculated with a window length of 75 frames at the beginning, middle, and end of the scene. They identifies two groups, containing agents (0,1) and agents (2,3,4) respectively merging together, then splitting to groups with agents (0,1,2) and agents (3,4). Cross-checking these groups with the trajectories shown in Figure 1 confirms that the correct clusters have been identified, indicating that agent 2 has moved across from one group to another during Scene 0. Similar clustering can be inferred for Scenes 1 and 2,

[3] In a directed graph, it is feasible to consider the ingoing, outgoing, or total degree of any vertex: k_{in}, k_{out}, or k_{tot}. In the results presented here, for simplicity we work with k_{in} only.

α	β	$I_{\alpha,\beta}^{[5135,5234]}$	$I_{\alpha,\beta}^{[5455,5604]}$	$I_{\alpha,\beta}^{[5877,5926]}$
0	1	0.123453	0.079655	0.195105
0	2	0.022849	0.061293	0.193017
0	3	0.024399	0.079655	0.092407
0	4	0.028575	0.070474	0.099228
1	2	0.023102	0.061293	0.193017
1	3	0.016591	0.079655	0.030227
1	4	0.019014	0.070474	0.083777
2	3	0.056464	0.061293	0.015544
2	4	0.045338	0.052564	0.077374
3	4	0.045338	0.070474	0.204106

Table 3: Scene 0 inter-layer mutual information at various points in the scene. These data are represented visually as graphs of layers in Figure 4

with the caveat that more chaotic behaviour, such as that in the fight scene, loses any strong mutual correlations between the layers of the DMVG.

5 Related Work

[13, 2, 3] investigate anomalous movements in crowd scenes using topology [13] and hidden Markov models [3, 2]. On the other hand [9, 15, 17] model pedestrian group formation using techniques such as social force models and optical flow.

None of the existing models use a single data structure for *both* of these key tasks. The distinctive feature of our method is that very simple measures defined on the same graph structure capture both dynamics and clustering. This enables faster data processing and storage in the context of real time high volume data, for example across nodes of a distributed surveillance sensor network used for automatic monitoring.

6 Conclusions and Further Work

We have shown that multiplex visibility graphs associated with pedestrian trajectory data extracted from real-world CCTV are a suitable data representation for the purposes of: (1) inferring scene dynamics, in particular signalling chaotic movements associated with events such as fights; and (2) clustering groups of people who are interacting together within a scene. We examined two particularly simple multiplex measures, the average edge overlap and the average inter-layer mutual information, which both capture correlations inherent in the source data. These are efficient to compute using sparse matrix operations. Our numerical results show that edge overlaps are suitable discriminators of pedestrian scene dynamics, while mutual information is an alternative to standard trajectory clustering techniques for finding groups of people.

Moving-window average multiplex edge overlap measures are relatively stable with respect to the window lengths, but mutual inter-layer information measures are not so robust. A natural next step is to examine alternatives to the pairwise mutual information measure, $I_{\alpha,\beta}$, including conditional mutual information, or normalised variants such as the inter-layer redundancy $R_{\alpha,\beta} = \frac{I_{\alpha,\beta}}{H_\alpha + H_\beta}$ where H gives the marginal entropy of the (degree distribution of the) layer

in question. Many similar techniques developed for the study of deterministic and stochastic dynamical systems using MVGs are also candidates for analysing crowd movement data, given our results.

References

[1] Charu C Aggarwal and Chandan K Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition, 2013.

[2] Maria Andersson, Fredrik Gustafsson, Louis St-laurent, and Donald Prevost. Recognition of Anomalous Motion Patterns in Urban Surveillance. *IEEE Journal on Selected Topics in Signal Processing*, (7):102–110, 2013.

[3] E.L. Andrade, S. Blunsden, and R.B. Fisher. Modelling crowd scenes for event detection. *Proceedings of the 18th International Conference on Pattern Recognition*, 2006.

[4] Federico Battiston, Vincenzo Nicosia, and Vito Latora. Structural measures for multiplex networks. *Physical Review E*, 89:032804, 2014.

[5] S Blunsden and R. B. Fisher. The BEHAVE video dataset: ground truthed video for multi-person behavior classification. *Annals of the BMVA*, 2010:1–11, 2010.

[6] S Boccaletti, G Bianconi, R Criado, C I del Genio, J Gómez-Gardeñes, M Romance, I Sendiña Nadal, Z Wang, and M Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[7] Andriana S L O Campanharo, M. Irmak Sirer, R. Dean De Malmgren, Fernando M. Ramos, and Luís a Nunes Amaral. Duality between time series and networks. *PLoS ONE*, 6(8):1–12, 2011.

[8] Vincenzo Fioriti, Alberto Tofani, and Antonio Di Pietro. Discriminating chaotic time series with visibility graph eigenvalues. *Complex Systems*, 21:193–200, 2012.

[9] Sultan Daud Khan, Giuseppe Vizzari, Stefania Bandini, and Saleh Basalamah. Detecting Dominant Motion Flows and People Counting in High Density Crowds. *The Journal of Selected Topics in Signal Processing (J-STSP)*, 2008.

[10] Lucas Lacasa. On the degree distribution of horizontal visibility graphs associated to Markov processes and dynamical systems: diagrammatic and variational approaches. *arXiv preprint arXiv:1402.5368*, pages 1–45, February 2014.

[11] Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nuño. From time series to complex networks: the visibility graph. *Proceedings of the National Academy of Sciences of the United States of America*, 105(13):4972–4975, 2008.

[12] Lucas Lacasa and Raul Toral. Description of stochastic and chaotic series using visibility graphs. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 82:1–26, 2010.

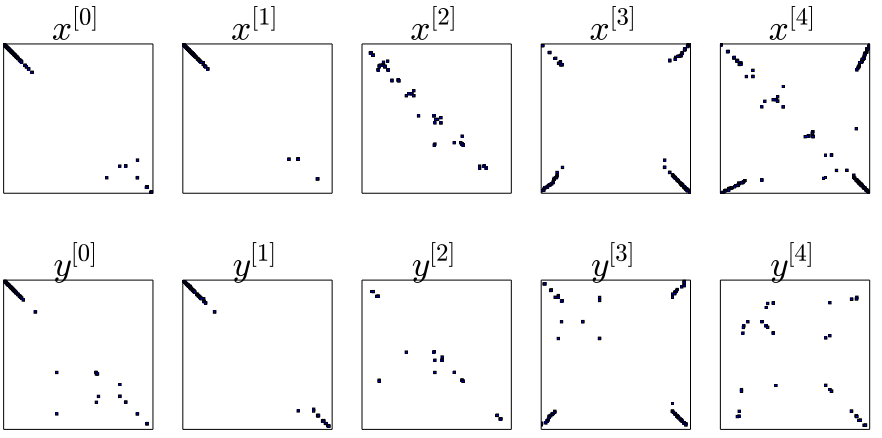
[13] Nan Li and Zhimin Zhang. Abnormal Crowd Behavior Detection Using Topological Methods. *2011 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 13–18, July 2011.

[14] B Luque, Lucas Lacasa, Fernando Ballesteros, and Jordi Luque. Horizontal visibility graphs: Exact results for random time series. *Physical Review E*, pages 1–17, February 2009.

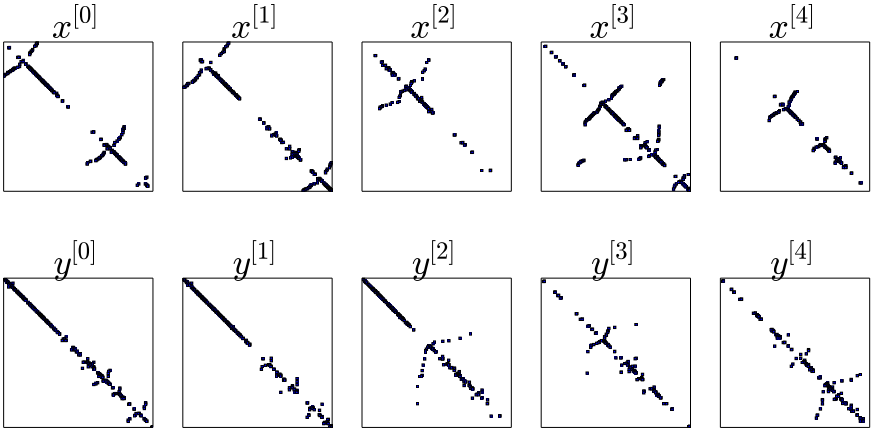
[15] Riccardo Mazzon, Fabio Poiesi, and Andrea Cavallaro. Detection and tracking of groups in crowd. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2013*, pages 202–207, 2013.

[16] Vincenzo Nicosia, Lucas Lacasa, and Vito Latora. From multivariate time series to multiplex visibility graphs. *arXiv preprint arXiv:1408.0925*, page 5, August 2014.

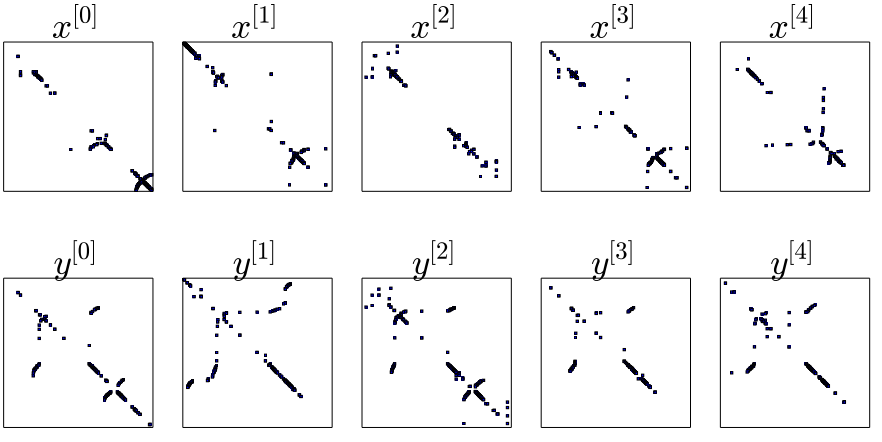
[17] Francesco Santoro, Sergio Pedro, Zheng-hua Tan, and Thomas B Moeslund. Crowd Analysis by Using Optical Flow and Density Based Clustering. In *18th European Signal Processing Conference (EUSIPCO-2010)*, pages 269–273, 2010.



(a) Scene 0



(b) Scene 1



(c) Scene 2

Figure 2: DHVG adjacency matrices for $x^{[\alpha]}$ and $y^{[\alpha]}$ for each agent α. See Section 2.2 for details.

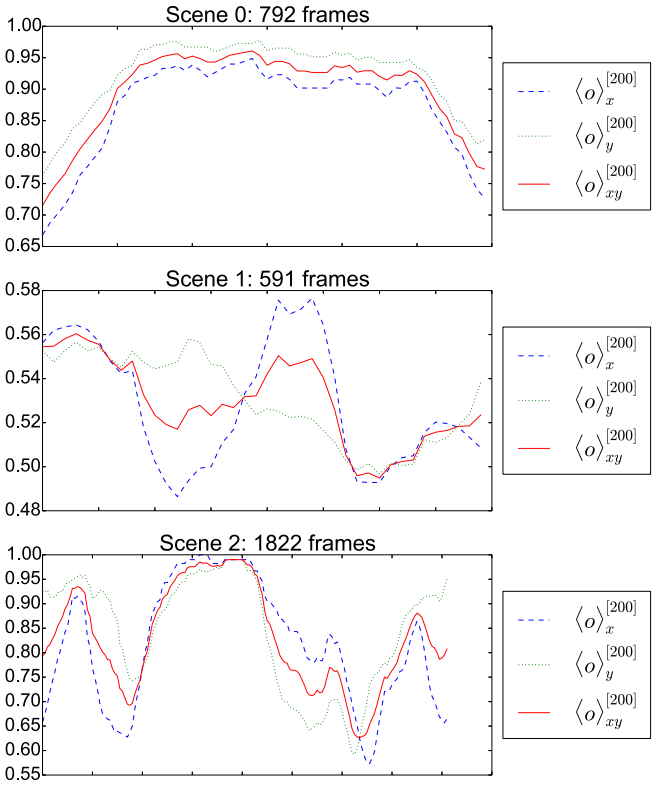


Figure 3: Evolution of average edge overlap $\langle o \rangle$ across scenes.

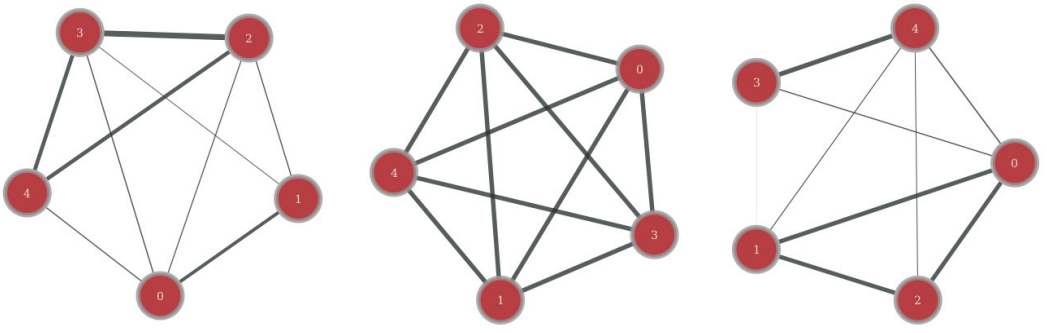


Figure 4: Inter-layer mutual information over 75 franes at beginning, middle end end of Scene 0, represented as *graphs of layers*. Thicker edges represent stronger inter-layer correlations. Clusters here do correspond to the groups active in the scene at the time: $(\{0,1\},\{2,3,4\}),(\{0,1,2,3,4\}),(\{0,1,2\},\{3,4\})$ respectively.