Spatial Visibility Clustering Analysis in Urban Environments Based on Pedestrians' Mobility Datasets

Oren Gal, Yerach Doytsher
Mapping and Geo-information Engineering
Technion - Israel Institute of Technology
Haifa, Israel
e-mail: {orengal,doytsher}@technion.ac.il

Abstract—**In this paper, we propose a Spatial Visibility Clustering (SVC) method estimating the number of clusters (groups) k, based on 3D visible volumes analysis in urban environments. We extend our previous work and propose fast and exact 3D visible volumes analysis in urban scenes based on an analytic solution. We test and analyze the SVC method by using real records of pedestrians' mobility datasets from the city of Melbourne and by setting control points for efficient monitoring and control using a K-means clustering algorithm. By testing large databases, we also propose time zones division for optimal control points.**

Keywords-Visibility; 3D; Urban environment; Spatial analysis.

I. Introduction and Related Work

'Clustering methods' refers to the division of data sets into groups, each containing similar objects. Data modeling is extensively studied in statistics, mathematics and machine learning [1]. Most of the common clustering methods can be divided into hierarchical and partitioning methods.

Partitioning algorithms determine the clusters directly, such as the well-known K-Means method, where by a hierarchical mechanism, builds the clusters gradually.

Clustering methods of 2D spatial data (such as GIS database) were also studied, defining data proximity by using, inter alia, a Delaunay diagram. These methods focused on performances and low complexity, by keeping K-nearest neighbors using a connectivity graph where clusters become connected components [6].

Our research contributes to the spatial data clustering field, where, as far as we know, visibility analysis has become a leading factor for the first time. The SVC method, while mining the real pedestrians' mobility datasets, enables by a visibility analysis to set the number of clusters.

The efficient computation of visible surfaces and volumes in 3D environments is not a trivial task. Accurate visibility computation in 3D environments is a very complicated task demanding a high computational effort, which could hardly have been done in a very short time using traditional well-known visibility methods [15].

Most of these works have focused on approximate visibility computation, enabling fast results using interpolations of visibility values between points, calculating point visibility with the Line of Sight (LOS) method [4,5]. Lately, fast and accurate visibility analysis computation in 3D environments has been presented [7,8,9,10].

In this paper, we introduce a unified method for estimating the number of clusters using 3D visible volumes analysis, SVC. Based on our previous work, we use a fast and efficient analytic solution, setting visibility boundaries of visible surfaces from the viewpoint. We extend our solution to 3D volumes, computing 3D visible volumes. By using *F*-criteria, we set the optimal number of clusters from the visibility aspect.

We demonstrate our method using real datasets from the city of Melbourne's 24-hours pedestrians monitoring system, localizing control points at each hour during the day, using a K-means algorithm with SVC output, i.e., number of clusters *k*. We analyze pedestrians' mobility behavior and suggest dividing the day into four time zones, based on our datasets and setting optimal control points during these time zones.

In Section II, we first introduce the SVC method, and the extended visible volumes analysis. In Section III, we present the SVC simulation using the city of Melbourne's datasets. Eventually, we present our approach by dividing a day's hours into four time zones and setting optimal control points.

II. Spatial Visibility Clustering (SVC) Method

We present, for the first time as far as we know, a unified spatial analysis defining the number of clusters in a data set based on analytic visibility analysis, SVC. The output of our method can be efficiently used by common clustering methods (e.g., K-means or hierarchical). The number of clusters in dense environments can be used for civil and security applications in urban environments, based on 3D visibility analysis from points of view.

For the last twenty years, many methods were proposed in order to estimate the number of clusters in data sets [2,3,11,18]. As previously mentioned [11], the approaches can be divided into global and local methods.

First, we introduce the main steps of our method and formulate the problem of estimating the number of clusters and the proposed volumes visibility analysis in 3D. Later, we present the analysis of the number of clusters using the SVC method, based on real pedestrians' mobility data sets. Finally,

ISBN: 978-1-61208-326-1

we examine a unique division of a twenty four-hour day into four different time zones in Melbourne [13], for control points based on pedestrians' mobility datasets in a number of points of interest, presented in Figure 1.

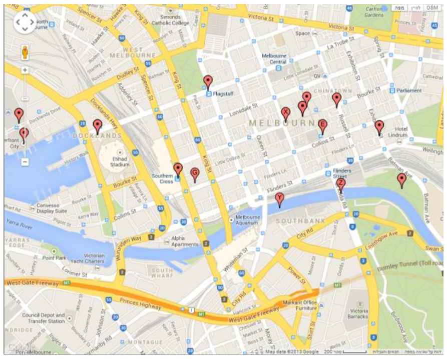


Figure 1. Melbourne Sensors Location for Monitoring Pedestrians' Mobility Data

A. *Spatial Visibility Clustering - Main Stages*

Our data set $\{X_{ij}\}$, *i=1,2,..,n, j=1,2,..,p,* consists of *p* features measured on *n* independent observation, where observation points marked with blue circles are illustrated in Figure 2. We clustered the data into *k* clusters, $C_1, C_2, .. C_k$. For cluster r, denote as C_r with n_r observations:

$$V_r = \sum_{i\in C_r}\sum_{j\in C_r}\left\|V(x_i)-V(x_j)\right\| \qquad (1)$$

$$V_r = \sum_{i\in C_r}\left\|V(x_i)-V(\bar{x})\right\| \qquad T_k = \sum_{r=1}^{k}\frac{1}{S}V_r$$

where $V(x)$ denotes the visible volumes from a viewpoint *x* bounded inside the total volume *S*, V_r is the sum of the absolute visibility differences of all viewpoints from their cluster visibility mean, and the normalized visible volumes T_k for all clusters *r=1..k*, called ***dispersion***.

Similarly to many other methods estimating the number of clusters [11,17], we define reference data sets distributed uniformly inside bounding volume *S*. We define our reference data sets with the same size of the original data set *X*, and calculate the dispersion of these datasets, T_k^*.

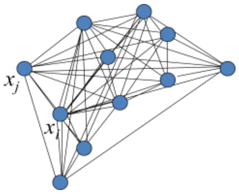


Figure 2. Pedestrians' location architecture based on monitoring datasets, observation points marked with blue circles

Based on *F* statistic, datasets are analyzed, where adding another cluster does not give a better modeling of the data, also known as *F-test* criteria. By setting a group's visibility variance, the number of clusters can be estimated efficiently:

$$SVC_n(k) = T_k^* - T_k \qquad (2)$$

Fast and efficient visibility volume computation from a specific viewpoint, bounded in volume *S*, is presented in the next subsection.

SVC steps can be summarized as follows:

1. Calculate the sum of absolute visibility differences of all points from their cluster visibility mean. Normalize this sum for all possible clusters T_k, also called dispersion.
2. Generate a set of reference datasets, simulated by a uniform distribution model inside bounding volume *S*.
3. Calculate the dispersion of each of these reference datasets, and calculate their mean visibility values.
4. Define SVC for each possible number of clusters as: Expected dispersion of reference datasets - Dispersion of original dataset.

Originally, *F* statistic was used to test the significance of the reduction in the sum of squares as we increase the number of clusters [11]. In general, when the number of clusters increases, the in-cluster decay first declines rapidly. From a certain *k*, dividing a dataset into *k+1* clusters decreases the value of *F-test* function which depends on *k*.

Approximated *F-test* function: Assuming that T_k is the partition of n instances into *k* clusters, and T_{k+1} is obtained from T_k splitting one of the clusters, then the overall mean ratio can be approximated as:

$$F_k = \frac{SVC_n}{SVC_{n+1}} \qquad (3)$$

We adapted aspects of previous F statistic theory for visibility analysis. More detailed F statistic analysis can be found in [11].

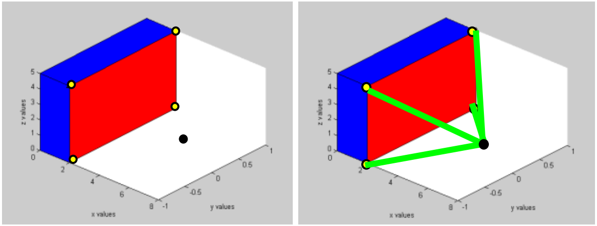
The spatial meaning of this mathematical clustering formulation can be simplified as a group of viewpoints with minimal difference to the average visible volume in the same bounding box.

B. *Analytic 3D Visible Volumes Analysis*

In this section, we present fast 3D visible volumes analysis in urban environments, based on an analytic solution which plays a major role in our proposed method of estimating the number of clusters. We extend our previous work [7] for surfaces visibility analysis, and present an efficient solution for visible volumes analysis in 3D.

We analyze each building, computing visible surfaces and defining visible pyramids using analytic computation for visibility boundaries [7]. For each object we define Visible Boundary Points (VBP) and Visible Pyramid (VP).

A simple case demonstrating analytic solution from a visibility point to a building can be seen in Figure 3(a). The visibility point is marked in black, the visible parts colored in red, and the invisible parts colored in blue where VBP marked with yellow circles.



(a) (b)

Figure 3. (a) Visibility Volume Computed with the Analytic Solution. (b) Visible Pyramid from a Viewpoint (marked as a Black Dot) to VBP of a Specific Surface

In this section, we introduce our concept for visible volumes inside bounding volume by decreasing visible pyramids and projected pyramids to the bounding volume boundary. First, we define the relevant pyramids and volumes.

The Visible Pyramid (VP): we define $VP_i^{j=1..Nsurf}(x_0, y_0, z_0)$ of the object *i* as a 3D pyramid generated by connecting VBP of specific surface *j* to a viewpoint $V(x_0, y_0, z_0)$.

In the case of a box, the maximum number of N_{surf} for a single object is three. VP boundary, colored with green arrows, can be seen in Figure 3(b).

For each VP, we calculate Projected Visible Pyramid (PVP), projecting VBP to the boundaries of the bounding volume S.

Projected Visible Pyramid (PVP) - we define $\mathrm{PVP}_i^{j..N_{surf}}(x_0, y_0, z_0)$ of the object *i* as 3D projected points to the bounding volume *S*, VBP of specific surface *j* trough viewpoint $V(x_0, y_0, z_0)$. VVP boundary, colored with purple arrows, can be seen in Figure 4.

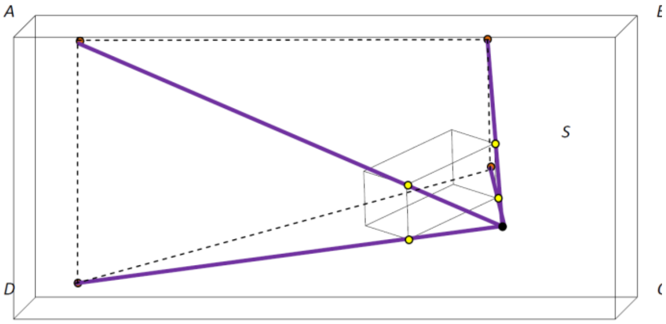


Figure 4. Invisible Projected Visible Pyramid Boundaries colored with purple arrows from a Viewpoint (marked as a Black Dot) to the boundary surface ABCD of Bounding Volume *S*

The 3D Visible Volumes inside bounding volume *S*, $\mathrm{VV_S}$, computed as the total bounding volume *S*, $\mathrm{V_S}$, minus the Invisible Volumes $\mathrm{IV_S}$. In a case of no overlap between buildings, $\mathrm{IV_S}$ is computed by decreasing the visible volume from the projected visible volume, $\sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (\mathrm{V}(\mathrm{PVP}_i^j) - \mathrm{V}(\mathrm{VP}_i^j))$.

$$\mathrm{VV_S} = \mathrm{V_S} - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} \mathrm{IV}_{S_i}^j \qquad (4)$$

$$\mathrm{VV_S} = \mathrm{V_S} - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (\mathrm{V}(\mathrm{PVP}_i^j) - \mathrm{V}(\mathrm{VP}_i^j))$$

By decreasing the invisible volumes from the total bounding volume, only the visible volumes are computed, as seen in Figure 5. Volumes of VPV and VP can be simply computed based on a simple pyramid volume geometric formula.

In a case of two buildings without overlapping, $\mathrm{IV_S}$ computed for each building, as presented above, as can be seen in Figure 6.

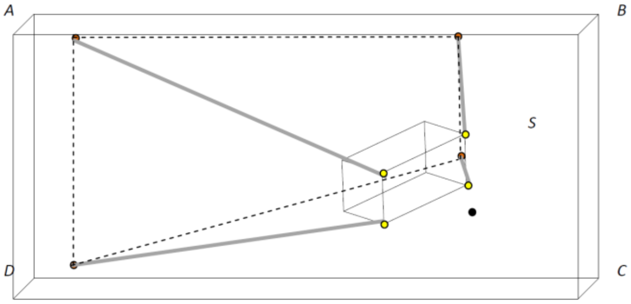


Figure 5. Invisible Volume $\mathrm{V}(\mathrm{PVP}_i^j) - \mathrm{V}(\mathrm{VP}_i^j)$ Colored in Gray Arrows. Decreasing Projected Visible Pyramid boundary surface ABCD of Bounding Volume S from Visible Pyramid

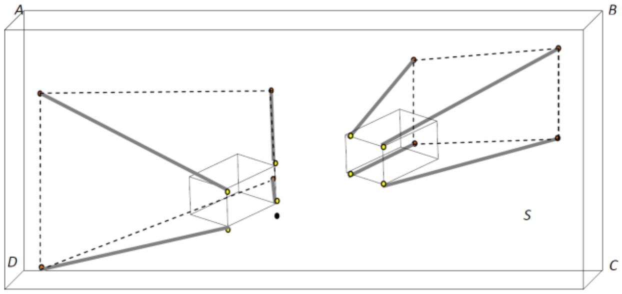


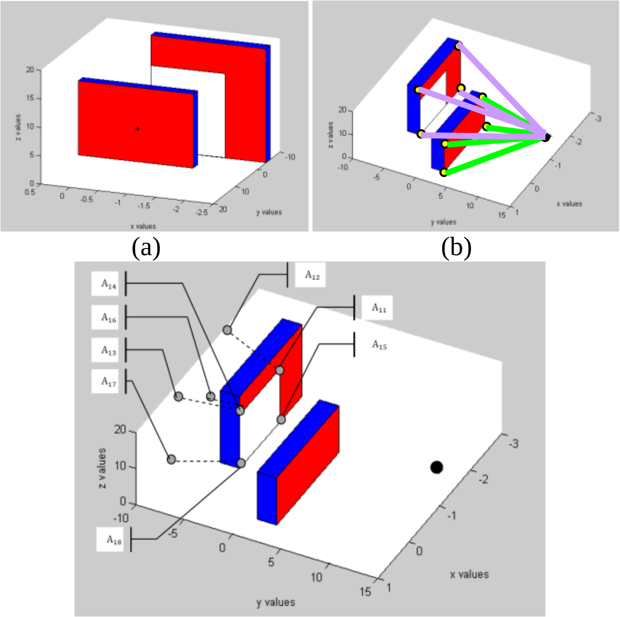
Figure 6. Invisible Volume $\mathrm{V}(\mathrm{PVP}_i^j) - \mathrm{V}(\mathrm{VP}_i^j)$ Colored in Gray Arrows. Decreasing Projected Visible Pyramid boundary surface ABCD of Bounding Volume S from Visible Pyramid

Considering two buildings with overlap between object's Visible Pyramids, as seen in Figure 7(a). In Figure 7(b), VP_1^1 boundary is colored by green lines, VP_2^1 boundary is colored by purple lines and the hidden and Invisible Surface between visible pyramids $\mathrm{IS}_{\mathrm{VP}_1^i}^{\mathrm{VP}_2^i}$ is colored in white.

Invisible Hidden Volume (IHV) - We define Invisible Hidden Volume (*IHV*), as the *Invisible Surface (IS)* between visible pyramids projected to bounding box *S*.

For example, IHV in Figure 7(c) is the projection of the invisible surface between visible pyramids colored in white, projected to the boundary plane of bounding box *S*.

In the case of overlapping buildings, by computing invisible volumes IV_S, we decrease IHV twice between the overlapped objects, as can be seen in Figure 7(c), *IHV* boundary points denoted as $\{\mathrm{A}_{11}, .., \mathrm{A}_{18}\}$. The same scene is presented in Figure 8, where Invisible Volume $V(PVP_i^j) - V(VP_i^j)$ is colored in purple and green arrows for each building.



(a) (b)

(c)

Figure 7. (a) Computing Hidden Surfaces between Buildings , VP_2^1 Base Plane, $IS_{VP_1^i}^{VP_2^i}$ (b) The Two Buildings - VP_1^1 in green and VP_2^1 in Purple (from the Viewpoint) and $IS_{VP_1^i}^{VP_2^i}$ in White (c) IHV boundary points colored with gray circles denoted

The *PVP* of the object close to the viewpoint is marked in black, colored with pink circles denoted as boundary set points $\{B_{11}, .., B_{18}\}$ and the far object's *PVP* is colored with orange circles, denoted as boundary set points $\{C_{11}, .., C_{18}\}$. It can be seen that *IHV* is included in each of these invisible volumes, where $\{A_{11}, .., A_{18}\} \in \{B_{11}, .., B_{18}\}$ and $\{A_{11}, .., A_{18}\} \in \{C_{11}, .., C_{18}\}$.

Therefore, we add *IHV* between each overlapping pair of objects to the total visible volume. In the case of overlapping between objects' visible pyramids, 3D visible volume is formulated as:

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (V(PVP_i^j) - V(VP_i^j) + IHV_i^j) \qquad (5)$$

The same analysis holds true for multiple overlapping objects, adding the IHV between each two consecutive objects.

In Figure 9, we demonstrate the case of three buildings with overlapping. The invisible surfaces are bounded with dotted lines, while the projected visible surfaces to the overlapped building are colored in gray. In order to calculate the visible volumes from a viewpoint, *IHV* between each two buildings must be added as a visible volume, since it is already omitted at the previous step as an invisible volume.

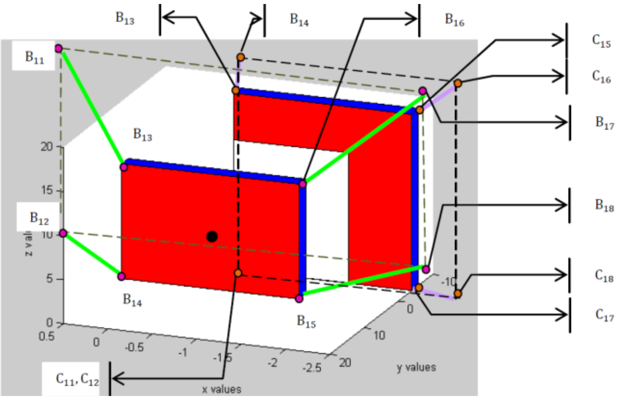


Figure 8. Invisible Volume $V(PVP_i^j) - V(VP_i^j)$ colored in purple and green arrows for each building. PVP of the object close to viewpoint colored in black, colored with pink circles and the far object PVP colored with orange circle

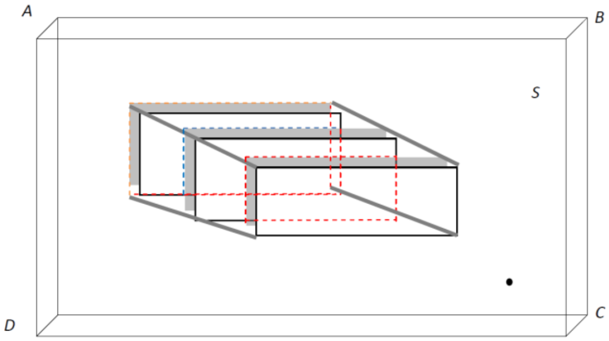


Figure 9. Three overlapping buildings. Invisible surfaces bounded with dotted lines, projected visible surfaces of the overlap building colored in gray

C. Simulations

In this section, we demonstrate the SVC method of estimating the number of clusters based on pedestrians' mobility datasets. For each pedestrian's location datasets, we analyze the 3D visible volumes inside bounding volume *S*, defined as a 3D box.

Our datasets are based on the city of Melbourne's 24-hour pedestrian monitoring system (24PM). This system measures pedestrian activity at several Points of Interests (POI) with counting sensors. Pedestrian mobility datasets are available online with interactive maps, as seen in Figure 10(a), and can be downloaded for a specific date.

Our datasets include the number of pedestrians in each hour during the 2nd of July 2013, at different seventeen points of interest in Melbourne where counting sensors are located and defined as observation points.

Based on these datasets, we approximated the pedestrians' location using the well-known and common kinematic model for pedestrians presented by Hoogendoorn [12] etc. Based on this model, pedestrian 2D location can be estimated as:

$$x(t + \Delta t) = x(t) + V(t)\Delta t + w \qquad (6)$$

where *w* is a white noise of a standard Wiener Process which reflects the uncertainty in the expected traffic condition, described as Gaussian distribution.

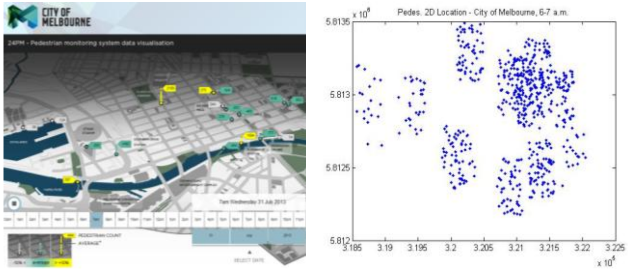
Pedestrian speed *V* can be divided into three major groups:

1. Fast: 1.8 meters per second
2. Standard: 1.3 meters per second
3. Slow: 0.8 meters per second

$$\begin{gathered} \mathrm{V(t)} \sim \mathrm{N}(\mu = 1.3, \sigma^2 = 0.5) \\ \mathrm{w} \sim \sqrt{\Delta \mathrm{t}}\,\mathrm{N}(0,1) \end{gathered} \tag{7}$$

The kinematic model of a pedestrian is only a part of the estimation and prediction of his movement in an urban environment. For simplicity, we use only a kinematic model for a pedestrian's future location, since decision-making in this field is very complicated.

At time step t, pedestrian location $x(t)$, is taken from a specific POI from our dataset, and the estimated pedestrian location $x(t + \Delta t)$ can be computed. In our simulations we set Δt for five minutes. For example, pedestrians' 2D location in UTM coordination, using the Hoogendoorn model [12], etc., between 6-7 a.m., can be seen in Figure 10(b).



(a) (b)

Figure 10. (a) City of Melbourne's 24-hour pedestrian monitoring system (24PM) – Online Visualization Map. (b) Pedestrians' 2D estimated location using the Hoogendoorn model [12] etc. between 6-7 a.m.

Each of pedestrian locations is processed as a viewpoint for estimating the number of clusters from spatial visibility aspects. The 3D visible volumes computation presented in the previous section are applied for computing $\mathrm{T_k}$, as described in Section II-A.

At each POI, we set the reference dataset of the pedestrian location distributed uniformly around the POI location, where the reference dataset size is the same one as the original dataset for the same POI, computing T_k^* .

We set the possible number of clusters from one to ten, demonstrating the SVC method. The number of clusters based on visible volumes analysis per day hour is presented in Figure 11.

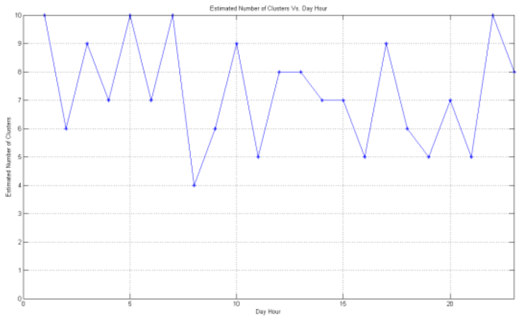


Figure 11. Number of Clusters for each Hour of 2/7/2013 Using SVC

As we can see in Figure 11, there is a correlation between the number of clusters and the pedestrians' mobility behavior. The number of clusters is close to the maximum (ten clusters in our case) during 6-9 AM, as can be predicted due to pedestrians' mobility while going to work. The number of clusters drops to a figure between eight to four clusters during the midday hours, and climbs again during nigh hours. More incentives analyzing pedestrians' mobility patters are presented in the next section.

III. Analyzing Pedestrians' Mobility Datasets

A. *Control Points*

In this section, we analyze pedestrians' mobility datasets during one day, estimating the number of clusters by using the SVC outcome, which is based on visibility analysis. Upon that, we use the K-means clustering method.

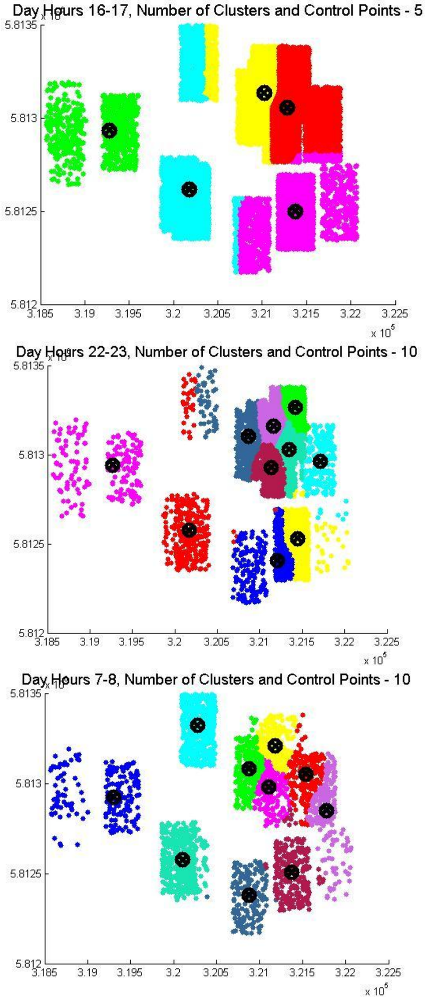
K-means clustering intends to partition n objects into k clusters, where each object belongs to the cluster with the nearest mean. The centroid of all objects in each cluster is set as control point. This method produces exactly k different clusters, where k is predefined from the SVC method. The objective of K-means clustering is to minimize total intra-cluster variance, or the squared error function.

K-means algorithm is a heuristic algorithm which depends on initial cluster. K-means can be very slow to converge, but in practice can be handling as polynomial convergence case which is the same case for our data sets [19].

By using K-means and SVC method, control points location can be seen in Figure 12.

It can be noticed in Figure 12 that, in some cases, the geometric location of the sensor location is separated into two different clusters. Our maximal number of clusters is set to ten, whereas there are seventeen sensors. We set the maximal number of clusters to be smaller than the number of sensors on the scene. One of the major contributions of our work, related to the adaptive clustering capability, is separating datasets into a different clustering and setting the control points from a visibility aspect. Moreover, control point location should cover more than one area, as can be seen in Figure 12, and also depends on pedestrians' mobility during this hour, as can be seen in the next sub-section.

Video simulations showing control points locations using K-means clustering and SVC methods are available in [16].


Day Hours 16-17, Number of Clusters and Control Points - 5

Day Hours 22-23, Number of Clusters and Control Points - 10

Day Hours 7-8, Number of Clusters and Control Points - 10

Figure 12. Control Points Location and Clusters Presentation during Each Hour in a Day. Control points are marked with black circles. Pedestrians' mobility Clustered in different colors

B. Time Zones

In this section, we concentrate on learning pedestrians' patterns for setting optimal control points, i.e., control points for each time zone.

We divide the day into four time zones for efficient pedestrian monitoring: Morning hours (movement to work) – 6 - 9 AM; Mid-Day Hours (between morning and afternoon) – 10 AM to 16 PM; Afternoon hours (back from work and activity hours) – 17- 20 PM.; Night hours 20 - 23 PM.

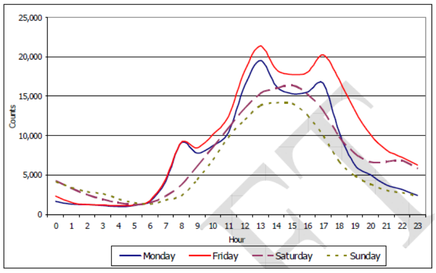


Figure 13. Pedestrian Activity Analysis [14]

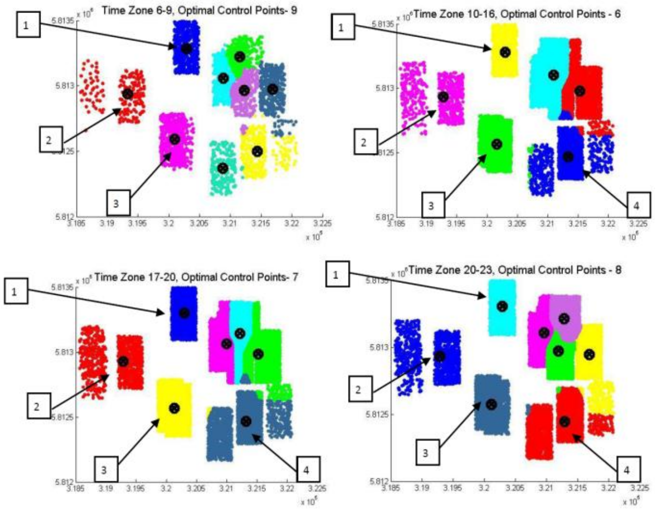
The suggested division of time zones partition can also be seen clearly in an official pedestrian monitoring report of the city of Melbourne [14], in Figure 13. The number of pedestrians counted by the monitoring system rises at the suggested time zones. In order to get reliable and comprehensive results regarding pedestrian mobility patterns, we tested a full month's (July 2013) dataset, analyzing each day for twenty-four hours.

Based on the average estimated number of clusters using SVC on these datasets, we found out that the number of optimal control points during these time zones is: Morning hours - Nine control points; Mid-Day Hours - Six control points; Afternoon hours - Seven control points; Night hours - Eight control points.

The localization of the optimal control points and the number of clusters for each time zone can be seen in Figure 14.

It can be seen that in the different time zones, three optimal control points and their cluster division are almost identically marked with arrows and numbers in Figure 14.

Four optimal control points with similar clustering can be seen in three time zones in Figure 14. These results can be applicable for personal-security and homeland security application in urban environments, localizing forces and sensors for optimal monitoring during a day's hours.


Time Zone 6-9, Optimal Control Points- 9

Time Zone 10-16, Optimal Control Points - 6

Time Zone 17-20, Optimal Control Points- 7

Time Zone 20-23, Optimal Control Points - 8

Figure 14. Optimal Control Points Location in Four Time Zones. Optimal Control points marked with black circles. Pedestrians' mobility Clustered in different colors

IV. Conclusions

In this paper, we presented a unified spatial analysis defining the number of clusters in a dataset based on an analytic visibility analysis, SVC.

The SVC method is based on fast and efficient 3D visible volumes computation. Estimating the number of clusters is based on minimum normalized visible volumes to reference datasets distributed uniformly inside bounding volume S. We demonstrated the SVC by using datasets from the city of Melbourne's 24-hour pedestrian monitoring system (24PM).

In the second part of this research, based on the SVC-estimated number of clusters, we analyzed pedestrian's mobility behavior, setting control points during a day's hours and dividing a day's hours into four time zones. We found a correlation of several optimal control points in different time zones.

Based on similar spatial analysis in other urban scenes, one can set optimal control points for various applications, such as entertainment events that can be efficiently visible at such points, or monitoring crowds' pedestrians from these control points in emergencies, equipped with medical assistance. Such results are also applicable for personal security and homeland security applications in urban environments, localizing police forces and sensors for optimal monitoring at different hours in a day.

V. References

[1] P. Arabie and L. J. Hubert, "An Overview of Combinatorial Data Analysis", in Arabie, P., Hubert, L.J., and Soete, G.D. (Eds.) Clustering and Classification, 1996, pp. 5-63, World Scientific Publishing Co., NJ.

[2] A. Borgers and H. Timmermans, "A model of pedestrian route choice and demand for retail facilities within inner-city shopping areas", Geographical Analysis, Vol. 18, No. 2, 1996, pp. 115-128.

[3] R. B. Calinski and J. Harabasz, “A Dendrite Method for Cluster Analysis”, Communications in Statistics ,vol. 3, 1974, pp. 1–27.

[4] Y. Doytsher and B. Shmutter, "Digital Elevation Model of Dead Ground", Symposium on Mapping and Geographic Information Systems (Commission IV of the International Society for Photogrammetry and Remote Sensing), Athens, Georgia, USA, 1994.

[5] F. Durand, "3D Visibility: Analytical Study and Applications", PhD thesis, Universite Joseph Fourier, Grenoble, France, 1999.

[6] V. Estivill-Castro and I. Lee, "AMOEBA: Hierarchical Clustering Based on Spatial Proximity Using Delaunay Diagram." In Proceedings of the 9th International Symposium on Spatial Data Handling. Beijing, China, 2000.

[7] O. Gal and Y. Doytsher, "Fast and Accurate Visibility Computation in a 3D Urban Environment", in Proc. of the Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services, Valencia, Spain, 2012, pp. 105-110, [accessed February 2014].

[8] O. Gal and Y. Doytsher, "Fast Visibility Analysis in 3D Procedural Modeling Environments", in Proc. of the, 3rd International Conference on Computing for Geospatial Research and Applications, Washington DC, USA, 2012.

[9] O. Gal and Y. Doytsher, "Fast Visibility in 3D Mass Modeling Environments and Approximated Visibility Analysis Concept Using Point Clouds Data", Int. Journal of Advanced Computer Science, IJASci, Vol 3, No 4, April 2013, ISSN 2251-6379, [accessed February 2014].

[10] O. Gal and Y. Doytsher, "Fast and Efficient Visible Trajectories Planning for Dubins UAV model in 3D Built-up Environments", Robotica, FirstView, Article pp 1-21 Cambridge University Press 2013 DOI: http://dx.doi.org/10.1017/S0263574713000787, [accessed February 2014].

[11] A. Gordon, Classification (2nd ed.), London: Chapman and Hall/CRC Press, 1999.

[12] S. P. Hoogendoorn and P. H. L. Bovy, "Microscopic pedestrian way finding and dynamics modelling," In Schreckenberg, M., Sharma, S.D. (eds.) Pedestrian and Evacuation Dynamics. Springer Verlag: Berlin, 2001, pp. 123-154.

[13] Melbourne City: http://www.pedestrian.melbourne.vic.gov.au/#date=31-08-2013&time=9 [accessed February 2014].

[14] Melbourne Report: https://docs.google.com/file/d/0B380gpj_lbU-dnRaRTFXdlh5Znc/edit [accessed February 2014].

[15] H. Plantinga and R. Dyer, "Visibility, Occlusion, and Aspect Graph," The International Journal of Computer Vision, vol. 5, 1990, pp. 137-160.

[16] https://sites.google.com/site/orenusv/home/svc [accessed February 2014].

[17] T. Schelhorn, D. Sullivan, M. Haklay, "STREETS: An agent-based pedestrian model," http://www.casa.ucl.ac.uk/streets.pdf. , 1999, [accessed February 2014].

[18] R. Tibshirani, G. Walther, T. Hastie, “Estimating the Number of Clusters in a Dataset via the Gap Statistic,” Journal of the Royal Statistical Society, Ser. B, vol. 32, 2001, pp. 411–423.

[19] A. Vattani, "K-means Requires Exponentially Many Iterations Even in the Plane", Discrete and Computational Geometry, vol. 45 (4), 2011, pp. 596–616. DOI:10.1007/s00454-011-9340-1, [accessed March 2014].