Sensors Placement in Urban Environments Using Genetic Algorithms

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

***Abstract*—Optimized coverage using multi-sensors is a challenging task, which is becoming more and more complicated in dense and occluded environments such as the urban environments. In this paper, we propose a multi-sensors placement solution for optimized coverage in dense urban environments. Our main contribution is based on a unique concept when facing partially visible objects, such as trees, in an urban scene, extending our previous work and proposing fast and exact 3D visible volumes analysis in urban scenes based on an analytic solution. We consider several 3D models for 3D visibility analysis and present an optimized solution using genetic algorithm, suited to our problem's constraints. We demonstrate the results through simulations with a 3D neighborhood model, taking trees into account.**

Keywords- Visibility; 3D; Urban environment; Spatial analysis; Genetic algorithm; Sensor coverage.

I. INTRODUCTION AND RELATED WORK

Modern cities and urban environments are becoming denser and denser, more heavily populated and are still rapidly growing, including new infrastructures, markets, banks, transportation etc.

In the last two decades, more and more cities and mega-cities have started using multi-camera networks in order to face this challenge, mounting cameras for security monitoring needs; however, this is still not enough [25]. Due to the complexity of working with 3D and the dynamic constraints of the urban terrain, sensors were placed in busy and populated viewpoints, to observe the occurrences in these major points of interest.

These current multi-sensors placement solutions ignore some key factors, such as: visibility analysis in 3D models, which also consist of unique objects such as trees, changing the visibility analysis aspect from visible or invisible states to semi-visible cases, such as trees, and above all, optimization solutions which take these factors into account.

Multi-sensor placement in 3D urban environments is not a simple task. The optimization problem of the optimal configuration of multi sensors for maximal coverage is a well-known Non-deterministic Polynomial-time hard (NP-hard) one [4], even without considering the added complexity of urban environments.

An extensive theoretical research effort has been made over the last four decades, facing a much simpler problem in 2D known as the art gallery problem, with unrealistic assumptions such as unlimited visibility for each agent, while the 3D problem has not received special attention [6].

The coupling between sensors' performances and their environment's constraints is, in general, a complex optimization problem. In this paper, we study the multi-sensors placement optimization problem in 3D urban environments for optimized coverage based on genetic algorithms using novel visibility analysis.

Our optimization solution for this problem relates to maximal coverage from a number of viewpoints, where each 3D position (x, y, z coordinates) of the viewpoint is set as part of the optimized solution. The search space contains local minima and is highly non-linear. The Genetic Algorithms are global search methods, which are well-suited for such tasks. The optimization process is based on randomly generating an initial population of possible solutions (called chromosomes) and, by improving these solutions over a series of generations.

Multi-sensor placements are scene- and application-dependent, and for this reason generic rules are not very efficient at meeting these challenges. Our approach is based on a flexible and efficient analysis that can deal with this complexity.

The total number of sensors is a crucial parameter, due to the real-time outcome data that should be monitored and tracked, where too many sensors are not an efficient solution. We address the sensor numbers that should be set as the tradeoff of coverage area and logical data sources that can be monitored and tracked.

Online visibility analysis is a very complicated task. Recently, off-line visibility analysis, based on preprocessing, was introduced. Cohen-Or et al. [3] used a ray-shooting sample to identify occluded parts.

Since visibility analysis in 3D urban environments is a very complicated task, it is therefore our main optimization function, known as Fitness. We introduce an extended visibility aspect for the common method of Boolean visibility values, "1" for objects seen and "0" for objects unseen from a specific viewpoint, and treat trees as semi-visibility values (such as partially seen, "0.5" value), thereby including in our analysis the real environmental phenomena which are commonly omitted.

 ISBN: 978-1-61208-383-4

We extend our previous work and propose fast and exact 3D visible volumes analysis in urban scenes based on an analytic solution, integrating trees into our 3D model and it is demonstrated with real urban scene model from Neve-Sha'anan (within the city of Haifa) neighborhood.

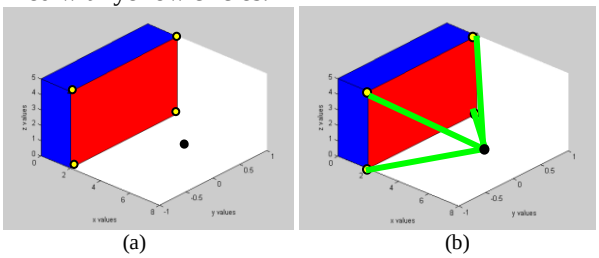
In the following sections, we extended the 3D visible volumes analysis which, for the first time, takes trees into account. Later on, we present the simulation using the Neve-Sha'anan (within the city of Haifa) neighborhood 3D model. Eventually, we present our genetic algorithm optimization stages and simulation based on our 3D visible volumes analysis, taking trees into account.

II. 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 shortly present our analysis presented in [22], extending our previous work [20] 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 [20]. For each object we define Visible Boundary Points and Visible Pyramid. We analyze each building, computing visible surfaces and defining visible pyramids using analytic computation for visibility boundaries [20]. 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 1(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 1. (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 (source: [22]).

In this section, we shortly 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 1(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 $PVP_i^{j..N_{surf}}(x_0, y_0, z_0)$ 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 2.

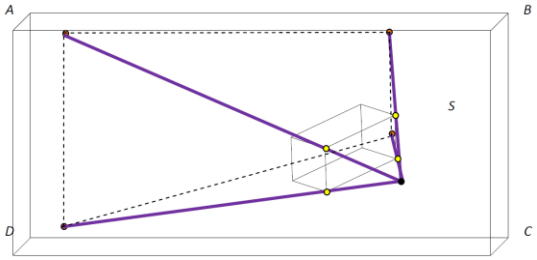


Figure 2. Invisible Projected Visible Pyramid Boundaries colored with purple arrows from a Viewpoint (marked as a Black Dot) to the boundary surface ABCD of Bounding Volume *S* (source: [22]).

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

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} IV_{S_i}^j \quad (1)$$

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

By decreasing the invisible volumes from the total bounding volume, only the visible volumes are computed, as seen in Figure 3. Volumes of VPV and VP can be simply computed based on a simple pyramid volume geometric formula.

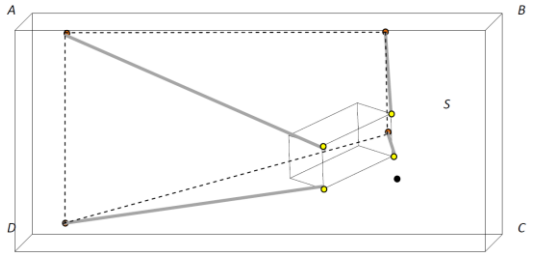


Figure 3. Invisible Volume $V(PVP_i^j) - V(VP_i^j)$ Colored in Gray Arrows. Decreasing Projected Visible Pyramid boundary surface ABCD of Bounding Volume S from Visible Pyramid (source: [22]).

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

 ISBN: 978-1-61208-383-4

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:

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (V(PVP_i^j) - V(VP_i^j) + IHV_i^j) \qquad (2)$$

The same analysis holds true for multiple overlapping objects, adding the IHV between each two consecutive objects, as can be seen in Figure 4.

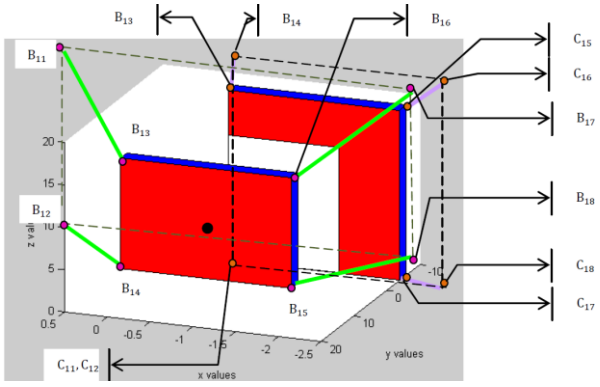


Figure 4. Invisible Volume $V(PVP_i^j) - V(VP_i^j)$ colored in purple and green arrows for each building. PVP of the object close to viewpoint colored in black, colored with pink circles and the far object PVP colored with orange circle (source: [22]).

Extended formulation for two buildings with or without overlap can be seen in [22].

A. Partial Visibility Concept - Trees

In this research, we analyze trees as constant objects in the scene, and formulate partial visibility concept. In our previous work, we tested trees as dynamic objects and their effect on visibility analysis [21]. Still, the analysis focused on trees' branches over time, setting visible and invisible values for each state, taking into account probabilistic modeling in time.

We model trees as two boxes [24], as seen in Figure 5. The lower box, bounded between $[0, h_1]$ models the tree's breed, leads to invisible volume and is analyzed as presented previously for a box modeling building's structures. On the other hand, the upper box bounded between $[h_1, h_2]$ is defined as partially visible, since a tree's leaves and the wind's effect are hard to predict and continuously change over time. Due to these inaccuracies, we set the projected surfaces and the Projected Visible Pyramid of this box as half visible volume.

Figure 5. Modeling a Tree Using Two Bounding Boxes.

According to that, a tree's effect on our visibility analysis is divided into regular boxes included in the total number of objects, N_{obj} (identical to the building case), and to the upper boxes modeling tree's leaves, denoted as N_{trees}. The total 3D visible volumes can be formulated as:

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (V(PVP_i^j) - V(VP_i^j) + IHV_i^j) - \sum_{i=1}^{N_{trees}} \sum_{j=1}^{N_{surf}} \frac{1}{2} (V(PVP_i^j) - V(VP_i^j) + IHV_i^j) \qquad (3)$$

B. Simulations

In this section, we demonstrate our 3D visible volumes analysis in urban scenes integrated with trees, presented in the previous section. We have implemented the presented algorithm and tested some urban environments on a 1.8GHz Intel Core CPU with Matlab. The Neve-Sha'anan Street in the city of Haifa was chosen as a case study, presented in Figure 6.

Figure 6. Views of Neve-Sha'anan Street, Haifa, Israel from Google Maps source: [15]

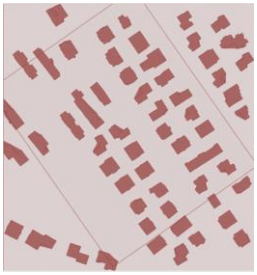


Figure 7. AutoCAD model of Neve-Sha'anan Street, Haifa, Israel.

We modeled the urban environment into structures using AutoCAD model, as seen in Figure 7, with bounding box S.

 ISBN: 978-1-61208-383-4

By using the Matlab©MathWorks software we automated the transformation of data from AutoCAD structure to our model's internal data structure.

Our simulations focused on two cases: (1) small-scale housing in dense environments; (2) Multi-story buildings in an open area. These two different cases are not taking into account the same objects. The first viewpoint is marked with black dot and the second one marked in purple, as seen in Figure 8. Since trees are not a part of our urban scene model, trees are simulated based on similar urban terrain in Neve-Sha'anan. We simulated fifty tree's locations using standard Gauss normal distribution, where trees' parameters h_1, h_2 are defined randomly $h_1 \in (0.3,0.9), h_2 \in (1.5,3)$, as seen in Figure 8.

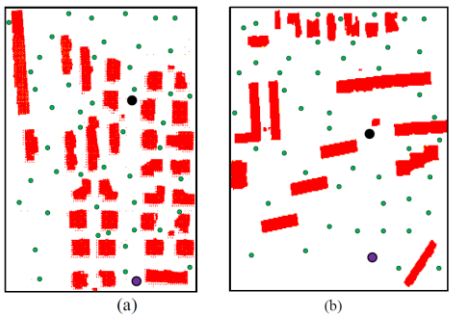


Figure 8. Tested Scenes with Trees marked with green points, Viewpoint 1 Colored in Black, Viewpoint 2 Colored in Purple : (a) Small-scale housing in dense environments; (b) Multi-story buildings in an open area.

We set two different viewpoints, and calculated the visible volumes based on our analysis presented in the previous sub-section. Visible volumes with time computation for different cases of bounding boxes' test scenes are presented in Table II and Table III.

One can notice that the visible volumes become smaller in the dense environments described in Table II, as we enlarge the bounding box. Since we take into account more buildings and trees, less volumes are visible and the total visible volumes from the same viewpoint are smaller.

III. Optimized Coverage Using Genetic Algorithms

The Genetic Algorithm (GA) presented by Holland [23] is one of the most common algorithms from the evolutionary algorithms class used for complex optimization problems in different fields, such as: pharmaceutical design, financial forecasting, tracking and coverage and bridge design. These kinds of algorithms, inspired by natural selection and genetics, are sometimes criticized for their lack of theoretical background due to the fact that in some cases the outcome is unpredictable or difficult to verify.

The main idea behind GA is based on repeated evaluation of individuals (which are part of a candidate solution) using an objective function over a series of generations. These series are improved over generations in order to achieve an optimal solution. In the next paragraphs, we present the genetic algorithms' main stages, adapted to our specific problem.

The major stages in the GA process (evaluation, selection, and reproduction) are repeated either for a fixed number of generations, or until no further improvement is noted. The common range is about 50-200 generations, where fitness function values improve monotonically [23].

TABLE I. Visible Volumes and Computation Time for Small-Scale Housing Case

Bounding Box	**Viewpoint**	**Visible Volumes** $[10^5 \cdot m^3]$	**Computation Time** [sec]
[100 m *100 m * 100 m]	Viewpoint 1	321.7	19.6
	Viewpoint 2	486.8	
[200 m * 200 m * 200 m]	Viewpoint 1	547.4	20.8
	Viewpoint 2	584.2	

TABLE II. Visible Volumes for Small Multi-story Buildings Case

Bounding Box [100 m *100 m * 100 m]	**Visible Volumes** $[10^5 \cdot m^3]$	**Computation Time** [sec]
Viewpoint 1	3453	22.9
Viewpoint 2	3528	

Population Initialization: The initialization stage creates the first generation of candidate solutions, also called chromosomes. A population of candidate solutions is generated by a random possible solution from the solution space. The number of individuals in the population is dependent on the size of the problem and also on computational capabilities and limitations. In our case it is defined as 500 chromosomes, due to the fact that 3D visible volumes must be computed for each candidate.

For our case, the initialized population of viewpoints configuration is set randomly, and would probably be a poor solution due to its random nature, as can be estimated. The chromosome is a 3xN-dimensional vector for N sensor's locations, i.e. viewpoints, where position and translation is a 3-dimensional (x,y,z) vector for each viewpoint location.

Evaluation: The key factor of genetic algorithm relates to individual evaluation which is based on a score for each chromosome, known as Fitness function. This stage is the most time-consuming in our optimization, since we evaluate all individuals in each generation. It should be noticed that each chromosome score leads to 3D visible volume computation N times. As a tradeoff between the covered area and computational effort, we set N to eight. In the worst case, one generation evaluation demands visibility analysis for four thousand different viewpoints. In such a case, one can easily understand the major drawback of the GA method in relation to computational effort. Nevertheless, parallel

 ISBN: 978-1-61208-383-4

computation has made a significant breakthrough over the last two decades; GA and other optimization methods based on independent evaluation of each chromosome can nearly be computed in linear time.

Fitness Function The fitness function evaluates each chromosome using optimization function, finding a global minimum value which allows us to compare chromosomes in relation to each other.

In our case, we evaluate each chromosome's quality using 3D visible volumes normalized to the bounding box S around a viewpoint:

$$f(i) = \frac{1}{S}\sum_{j=1}^{N} VV_S(x_j, y_j, z_j) \quad (4)$$

Selection: Once the population is sorted by fitness, chromosomes' population with greater values will have a better chance of being selected for the next reproduction stage. Over the last years, many selection operators have been proposed, such as the Stochastic Universal Sampling and Tournament selection. We used the most common Tournament, where k individuals are chosen randomly, and the best performance from this group is selected. The selection operator is repeated until a sufficient number of parents are chosen to form a child generation.

Reproduction: In this stage, the parent individuals chosen in the previous step are combined to create the next generation. Many types of reproduction have been presented over the years, such as crossover, mutation and elitism.

Crossover takes parts from two parents and splices them to form two offspring. Mutation modifies the parameters of a randomly selected chromosome from within a single parent. Elitism takes the fittest parents from the previous generation and replicates them into the new generation. Finally, individuals not selected as parents are replaced with new, random offspring.

A. Simulations

In this section, we report on simulation runs with our 3D visible volumes analysis in urban scenes integrated with trees, using genetic algorithms. The genetic algorithms were tested on a 1.8GHz Intel Core CPU with Matlab. We used Fallvile Island Sketchup Google Model [14] for simulating a dense urban scene with trees, as seen in Figure 9.

The stages of Crossover and Elitism operators are described as follows, with a probability of $p_c = 0.9$ (otherwise parents are copied without change):

1. Choose a random point on the two parents.
2. Split parents at this crossover point.
3. Create next generation chromosomes by exchanging tails.

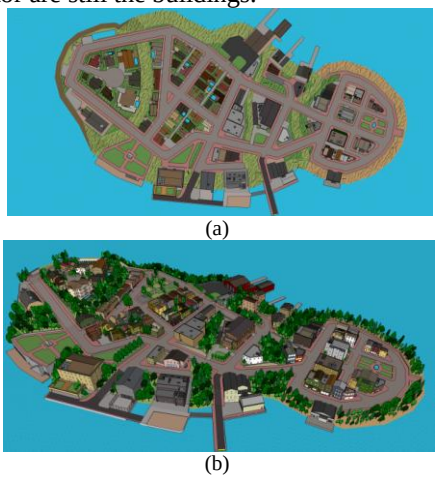
Where the Mutation operator modifies each gene independently with a probability of $p_m = 0.1$.

In order to process the huge amount of data, we bounded a specific region which includes trees and buildings, as seen in Figure 10. We imported the chosen region to Matlab and modeled the objects by boxes, neglecting roofs' profiles. Time computation for one generation was one hour long on average. As we could expect, the evaluation stage took up 94% of the total simulation time. We set the bounding box S as [500 m* 200 m* 50 m]. Population initialization included 500 chromosomes, each of which is a 24-dimensional vector consisting of position and translation, where all of them were generated randomly.

Based on the Fitness function described previously and the different GA stages and 3D visible volumes analysis, the location of eight viewpoints for sensor placement was optimized. Viewpoints must be bounded in S and should not penetrate buildings and trees. Stop criteria was set to 50 generations and Fitness function gradient.

Optimal coverage of viewpoints and visible volumes during ten running's simulations is seen in Figure 11, bounded in polygons marked with arrows. During these ten running simulations, we initialized the population randomly at different areas inside bounding box S.

These interesting results show that trees' effect inside a dense urban environment was minor, and trees around the buildings in open spaces set the viewpoint's location. As seen in Figure 11, polygon A and polygon B are both outside the areas blocked by buildings. But they are still located near trees, which affect the visible volumes, and we can predict that the same affect will occur in our real world. On the other hand, polygon C, which is closer to the area blocked by buildings, takes into account the trees in this region, but the major factor are still the buildings.



(a) (b)

Figure 9. Fallvile Island Sketchup Google Model Simulating Dense Urban Scene with Trees, [14]: (a) Topview; (b) Isometric view.

Figure 10. Bounded Area inside Bounding Box S marked in Black, inside Fallvile Island Sketchup Google Model.

Figure 11. Bounded Polygons of Optimized Cover Viewpoints Using GA marked with Arrows.

IV. CONCLUSIONS

In this paper, we presented an optimized solution for the problem of computing a maximal coverage from a number of viewpoints, using genetic algorithms method. As far as we know, for the first time we integrated trees as partially visible objects participating in a 3D visible volumes analytic analysis. As part of our research we tested several 3D models of 3D urban environments from the visibility viewpoint, choosing the best model from the computational effort and the analytic formulation aspects.

We tested our 3D visible volumes method on real a 3D model from an urban street in the city of Haifa, with time computation and visible volumes parameters.

In the second part of the paper, we introduced a genetic algorithm formulation to calculate an optimized solution for the visibility problem. We used several reproduction operators, which made our optimization robust. We tested our algorithm on the Fallvile Island Sketchup Google Model combined with trees, and analyzed the viewpoint's polygons results.

Our future work is related to validation between our simulated solution and projected volumes from sensors mounted in these viewpoints for optimal coverage.

V. REFERENCES

[1] N. Abu-Akel, "Automatic Building Extraction Using LiDAR Data," PhD Dissertation, Technion, Israel, 2010.

[2] D. Cohen-Or, G. Fibich, D. Halperin and E. Zadicario, "Conservative Visibility and Strong Occlusion for Viewspace Partitioning of Densely Occluded Scenes," In EUROGRAPHICS'98.

[3] D. Cohen-Or and A. Shaked, "Visibility and Dead- Zones in Digital Terrain Maps," Eurographics, vol. 14(3), pp.171-180, 1995.

[4] R. Cole and M. Sharir, "Visibility Problems for Polyhedral Terrain," Journal of Symbolic Computation, vol. 7, pp.11-30, 1989.

[5] Y. Chrysanthou, "Shadow Computation for 3D Interactive and Animation," Ph.D. Dissertation, Department of Computer Science, College University of London, UK, 1996.

[6] R. Church and C. ReVelle, "The Maximal Covering Location Problem," Papers of the Regional Science Association, vol. 32, pp.101-118, 1974.

[7] L. De Floriani and P. Magillo, "Visibility Algorithms on Triangulated Terrain Models," International Journal of Geographic Information Systems, vol. 8, no. 1, pp.13-41, 1994.

[8] L. De Floriani and P. Magillo, "Intervisibility on Terrains," In P.A. Longley, M.F. Goodchild, D.J. Maguire & D.W. Rhind (Eds.), Geographic Information Systems: Principles, Techniques, Management and Applications, pp. 543-556. John Wiley & Sons, 1999.

[9] Y. Doytsher and B. Shmutter, "Digital Elevation Model of Dead Ground," Symposium on Mapping and Geographic Information Systems (ISPRS Commission IV), Athens, Georgia, USA, 1994.

[10] G. Drettakis and E. Fiume, "A Fast Shadow Algorithm for Area Light Sources Using Backprojection," In Computer Graphics (Proceedings of SIGGRAPH '94), pp. 223–230, 1994.

[11] F. Durand, "3D Visibility: Analytical Study and Applications," PhD thesis, Universite Joseph Fourier, Grenoble, France, 1994.

[12] A.E. Eiben and J.E. Smith, "Introduction to Evolutionary Computing Genetic Algorithms," Lecture Notes, 1999.

[13] U. M. Erdem and S. Sclaroff, "Automated camera layout to satisfy task- specific and floor plan-specific coverage requirements," Computer Vision and Image Understanding, vol. 103, no. 3, pp. 156–169, 2006.

[14] Fallvile, (2010) http://sketchup.google.com/3dwarehouse/details?mid=2265cc05839f0e5925ddf6e8265c857c&prevstart=0

[15] D. Fisher-Gewirtzman, A. Shashkov and Y. Doytsher, "Voxel Based Volumetric Visibility Analysis of Urban Environments," Survey Review, DOI: 10.1179/1752270613Y.0000000059, 2013.

[16] W.R. Franklin, "Siting Observers on Terrain," in D. Richardson and P. van Oosterom, eds, Advances in Spatial Data Handling: 10th International Symposium on Spatial Data Handling. Springer-Verlag, pp. 109-120, 2002.

[17] W.R. Franklin and C. Ray, "Higher isn't Necessarily Better: Visibility Algorithms and Experiments," In T. C. Waugh & R. G. Healey (Eds.), Advances in GIS Research: Sixth International Symposium on Spatial Data Handling, pp. 751-770. Taylor & Francis, Edinburgh, 1994.

[18] W.R. Franklin and C. Vogt, "Multiple Observer Siting on Terrain with Intervisibility or Lores Data," in XXth Congress, International Society for Photogrammetry and Remote Sensing. Istanbul, 2004.

[19] H. Furuta, K. Maeda and E. Watanabe, "Application of Genetic Algorithm to Aesthetic Design of Bridge Structures," Computer-Aided Civil and Infrastructure Engineering, vol. 10, no. 6, pp.415–421, 1995.

[20] O. Gal and Y. Doytsher, "Analyzing 3D Complex Urban Environments Using a Unified Visibility Algorithm," International Journal On Advances in Software, ISSN 1942-2628, vol. 5 no.3&4, pp:401-413, 2012.

[21] O. Gal and Y. Doytsher, "Dynamic Objects Effect on Visibility Analysis in 3D Urban Environments," Lecture Notes in Computer Science (LNCS), vol. 7820, pp.147-163, DOI 10.1007/978-3-642-37087-8_11, Springer, 2013.

[22] O. Gal and Y. Doytsher, "Spatial Visibility Clustering Analysis In Urban Environments Based on Pedestrians' Mobility Datasets," The Sixth International Conference on Advanced Geographic Information Systems, Applications, and Services, pp. 38-44, 2014.

[23] J. Holland, "Adaptation in Natural and Artificial Systems," University of Michigan Press, 1975.

[24] K. Omasa, F. Hosoi, T.M. Uenishi, Y. Shimizu and Y. Akiyama, "Three-Dimensional Modeling of an Urban Park and Trees by Combined Airborne and Portable On-Ground Scanning LIDAR Remote Sensing," Environ Model Assess vol. 13, pp.473–481, DOI 10.1007/s10666-007-9115-5, 2008.

 ISBN: 978-1-61208-383-4