Search and Surveillance in emergency situations – A GIS-based approach to construct optimal visibility graphs

Michael Morin
Department of Computer Science and Software Engineering
Université Laval, Québec, Qc, Canada

Irène Abi-Zeid
Department of Operations and Decision Systems
Université Laval, Québec, Qc, Canada
Irene.Abi-Zeid@osd.ulaval.ca

Thanh Tung Nguyen
Department of Computer Science and Software Engineering
Université Laval, Québec, Qc, Canada

Luc Lamontagne
Department of Computer Science and Software Engineering
Université Laval, Québec, Qc, Canada

Patrick Maupin
Defence Research and Development Canada
Valcartier, Qc, Canada

ABSTRACT

We present a methodology to construct optimal visibility graphs from vector and raster terrain data based on the integration of Geographic Information Systems, computational geometry, and integer linear programming. In an emergency situation, the ability to observe an environment, completely or partially, is crucial when searching an area for survivors, missing persons, intruders or anomalies. We first analyze inter-visibility using computational geometry and GIS functions. Then, we optimize the visibility graphs by choosing vertices in a way to either maximize coverage with a given number of watchers or to minimize the number of watchers needed for full coverage.

Keywords

Visibility graphs, Geographic Information System, set cover, maximum coverage, surveillance, search and rescue.

INTRODUCTION

In an emergency situation, the ability to observe completely or partially an environment is crucial when searching an area for survivors, missing persons or intruders. In this paper we present a methodology to construct the smallest optimal *visibility graphs*[1] from vector and raster data based on the integration of Geographic Information System (GIS) tools, computational geometry, and integer linear programming. Geographic information systems have been widely used for emergency management (Emrich, Cutter and Weschler, 2011) and constitute a valuable tool for practitioners in the field especially with the advances in technology. Furthermore, many applications require the computation of visibility graphs (De Floriani and Magillo, 2003). For instance, in a Search and Rescue context where we wish to plan optimal search plans for searchers looking for a mobile search object with uncertain detection (Morin, Lamontagne, Abi-Zeid, Lang, and Maupin, 2010; Morin, Papillon, Laviolette, Abi-Zeid, and Quimper, 2012), an adequate representation of the terrain is important. This is also true in searches with certain (perfect) detection or visibility such as pursuit-evasion problems (e.g., Pellier and Fiorino, 2005), art gallery related problems (De Berg, 2000) or security monitoring (Murray, Kim, Davis, Machiraju and Parent, 2007).

[1] A visibility graph is made of two sets: a set of vertices and a set of edges. Each edge is a pair of vertices that represents inter-visibility between the vertices. When the vertices are geo-referenced, the visibility graph defines an inter-visibility relation between areas (points) of a map.

We apply our methodology to *structured* (i.e. built) environments, and *unstructured environments* such as outdoor forested areas. We formulate our problem as follows: A group of *watchers* is needed to cover an environment either to secure a perimeter, or to search for missing persons. In this context, the objectives we attain with our methodology are the following:

1. Analyze inter-visibility using vector or raster data and construct visibility graphs;
2. Find the smallest number of watchers necessary and their positions, whether they are human spotters, sensors or cameras to cover an area;
3. Given a fixed number of watchers, position the watchers to maximize the visibility coverage of an environment.

The solution to the optimization problem in step 2 and 3 is a visibility graph of the environment with the smallest number of nodes possible. In order to attain our first objective, we developed two approaches for analyzing inter-visibility and constructing visibility graphs: For vector data, we rely on an algorithm from computational geometry (Latombe, 1990). For raster data, we use functionalities from ESRI's ArcGIS 9.2. For our second and third objectives, we have formulated the optimization problems as integer linear programs (ILP), namely the *set cover* problem and the *maximum coverage* problem. We solve these problems using CPLEX 12.5 solver and the OPL modeling language[2].

Our methodology is illustrated using two environments: the Université Laval campus map (a structured environment), and the Montmorency forest map (an unstructured environment with an approximate area of 66 km^2). The paper is structured as follows: The first section presents the data processing phase for both vector and raster data. The second section presents the set cover problem and the maximum coverage problem. The third section provides experimental results.

THE DATA PROCESSING PHASE: COMPUTING A FULL VISIBILITY GRAPH

In the case of vector data, the connected polygons from the triangular irregular network (TIN) representation are first grouped and simplified using ArcGIS 9.2 built-in functionalities to obtain a 2-D environment (Figure 1) whose vertices represent possible positions for watchers in a visibility graph. The polygons represent obstacles to visibility between vertices. Subsequently, a Visual Basic for Application (VBA) script that we developed converts the data to geometrical objects[3], namely polygons, to be used by our C++ visibility graph computation code[4] (Lamontagne L., Rouet F. H., Abi-Zeid I., 2008). Finally, we apply a VBA script that we developed to draw the resulting visibility graph in the GIS (Figure 2).

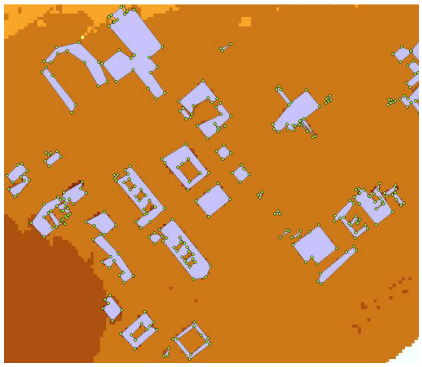

Figure 1. The 2-D polygons of the Université Laval campus

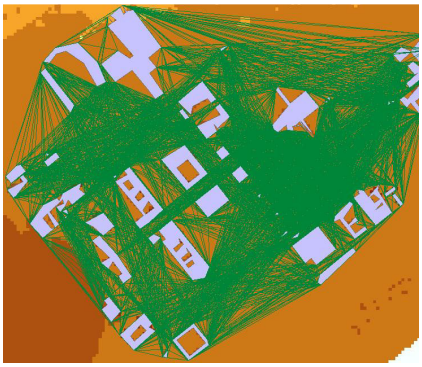

Figure 2. Visibility graph obtained for vector data – Université Laval campus

In the case of raster data, the approach we use to construct a visibility graph relies heavily on the *Viewshed Analyst* function in ArcGIS. We begin by superimposing over the digital terrain elevation model a uniform grid of square cells (Figure 3). Then, the point in the center of each cell is analyzed to determine its visible region based on line of sight visibility within a specified radius (Figure 4). This analysis results in a full visibility

[2] http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/

[3] The data structures we developed use the Boost library (Dawes, Abraham and Rivera, 2009) and the geometrical objects found in the CGAL library (Fabri and Pion, 2009.

[4] The implemented C++ code uses the software library VisiLibity (Obermeyer and collaborators, 2008). VisiLibity is capable of computing the visibility graph in $O(n^3)$ using an algorithm based on (Latombe, 1990).

graph of the area of interest (Figure 5). The visibility graphs obtained either from raster data or vector data are subsequently used in the optimization phase.

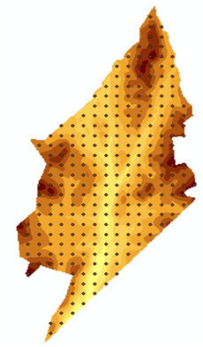

Figure 3. A grid of 50 by 50 m square cells over the Montmorency forest

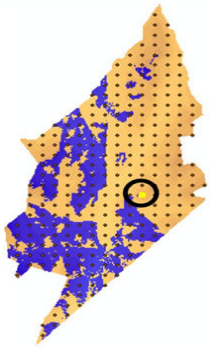

Figure 4. Visible regions in blue from the circled point

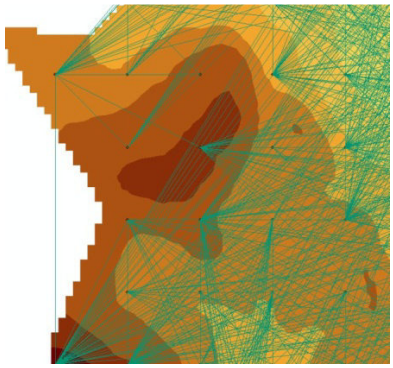

Figure 5. A zoom on part of a visibility graph over the Montmorency forest

THE OPTIMIZATION PHASE: OBTAINING OPTIMAL VISIBILITY GRAPHS

Given a full visibility graph, the question that arises is what is the smallest number of watchers necessary in order to ensure full coverage of the nodes. This can be formulated as a classical set cover problem (Goodchild and Lee, 1989) as shown in the ILP formulation on Figure 6. Let n be the number of vertices in the graphs. Let the presence of a watcher at vertex i be denoted by y_i, which is 1 if there is a watcher at position vertex i of the visibility graph, and 0 otherwise. We wish to minimize the number of watchers such that all the vertices are visible from at least one vertex. Let x_{ji} be 1 if vertex j is visible from vertex i and 0 otherwise.

Given a fixed number of watchers and a full visibility graph, another question that arises is what is the largest number of nodes that can be covered? And where should the available watchers be located in order to maximize the number of visible vertices? This can be formulated as a maximum coverage problem (Goodchild and Lee, 1989) as shown in the ILP formulation of Figure 7. Let n be the number of vertices in the graphs, and p the number of available watchers. Let the presence of a watcher at vertex i be denoted by y_i, which is 1 if there is a watcher at position vertex i of the visibility graph, and 0 otherwise. Let z_i be 1 if vertex i is not visible by any watcher (not covered). We wish to minimize the number of uncovered vertices given a number of watchers p. Let x_{ji} be 1 if vertex j is visible from vertex i and 0 otherwise.

$$\text{minimize} \sum_{i=1}^{n} y_i \qquad \text{subject to} \quad \sum_{i=1}^{n} x_{ji} y_i \geq 1$$
$$j = 1..n$$
$$y_i \in \{0,1\}$$

Figure 6. A set cover ILP formulation

$$\text{minimize} \sum_{i=1}^{n} z_i \qquad \text{subject to} \quad \sum_{i=1}^{n} y_i = p$$
$$\sum_{i=1}^{n} x_{ji} y_i = 1 - z_j$$
$$j = 1..n$$
$$y_i, z_i \in \{0,1\}$$

Figure 7. A maximum coverage ILP formulation

EXPERIMENTATION AND RESULTS

The goal of our experimentation is twofold: To illustrate the optimization phase of our methodology on structured and unstructured environments, and to evaluate the capacity of the presented ILP to produce the optimal visibility graphs within a limited solving time, and a limited maximal number of allowed watchers. All experiments were run on an Intel i7 Q740 processor with 8GB of RAM. The default CPLEX configuration is used for all experiments.

The Université Laval campus visibility graph contains a total of 255 vertices. We found, using the set cover OPL model, that the required number of watchers to cover the whole graph (smallest full coverage graph) is 16. CPLEX obtained the optimal solutions in less than 1 second. It is worth noting that this problem has no feasible solution if we require that each vertex be covered by exactly one other vertex which means that there is some multiple coverage. Multiple coverage implies redundancy which is a valuable property of solutions for

surveillance related problems such as in security monitoring (Murray, Kim, Davis, Machiraju and Parent, 2007), especially in emergency situations when new obstacles to visibility may emerge, such as following an earthquake for example. Figure 8 presents the achieved coverage on the Université Laval campus environment with a number of watchers varying from 1 to 16.

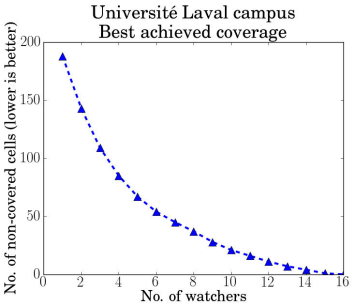

Figure 8. The best achieved maximum coverage for the Université Laval graph *versus* the number of watchers

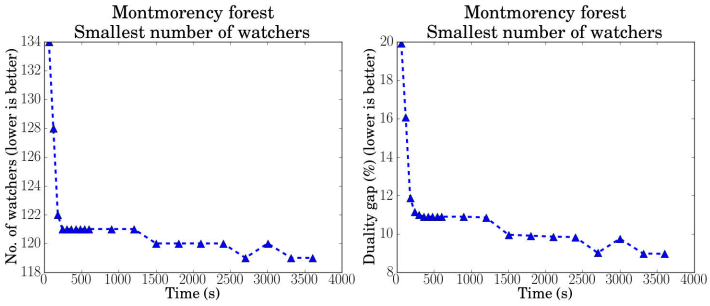

Figure 9. The smallest number of watchers for a set cover on the Montmorency forest (left) and its duality gap (right) versus the solving time in seconds

The visibility graph of our unstructured environment involves a total of 6025 vertices. The resulting visibility graph, the maximum coverage graph, is dense even though we allowed a maximal inter-visibility distance of 1 km in ArcGIS *Viewshed Analysis* tool. For this reason, in all evaluated cases, CPLEX was not able to prove the optimality of its solution within the allowed time.[5] Figure 9 shows the results for the set cover problem. On the left hand side of Figure 9, we present, for different allowed solving times, the minimal total number of watchers obtained by CPLEX. While the best solution, found after 45 minutes of solving time involved a total of 119 watchers, only 4 minutes were required to obtain a solution with only 2 more watchers (121 instead of 119). It is worth noting that after having let the program run for 12 hours, we were not able to improve our solution beyond 118 watchers. On the right hand side of Figure 9, we present the duality gap we obtained for the solutions of the left plot. In optimization, the duality gap may be interpreted as *how-far* the solver is from proving the optimality of its current best solution. In integer linear programming, a duality gap of 0 percent is achieved when the solution is optimal.

Figure 10 shows the results obtained on the unstructured environment of the Montmorency forest for the maximum coverage problem. Table 1 presents the detailed results for all problem instances. We allowed CPLEX a maximum of 10 minutes. As the total number of allowed watchers increases, the number of uncovered cells diminishes reaching a minimal value of 37. So even though the solver was unable to prove the optimality of its solution when we used 120 watchers, less than 1% of the vertices are left unobserved. We observe the same tendency in the case of 100 watchers where 1% of the vertices only are left unobserved.

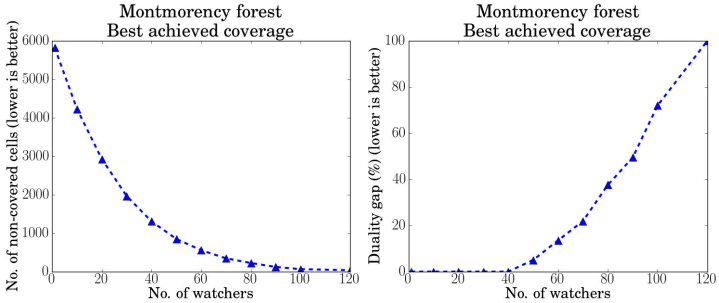

Figure 10. The best achieved maximum coverage on the Montmorency forest (left) and its duality gap (right) *versus* the number of watchers after 10 minutes of solving time

[5] The set cover problem and the maximum coverage problem are NP-hard problems (Garey and Johnson, 1979).

No. of watchers	Time (s)	Duality gap (%)	No. of non-covered cells
1	2.4	0	5820
10	3.7	0	4225
20	7.5	0	2923
30	65.7	0	1962
40	420.4	0	1304
50	600	5	848

No. of watchers	Time (s)	Duality gap (%)	No. of non-covered cells
60	600	13	555
70	600	22	345
80	600	38	223
90	600	49	121
100	600	72	63
120	600	100	37

Table 1. The best achieved maximum coverage on the Montmorency forest after 10 minutes of solving time

As we know from our set cover problem solutions, 120 watchers are enough to cover the whole visibility graph whereas no optimal solution under 118 watchers has been found. Given our software and hardware configurations, using the set cover problem formulation for larger number of watchers is more efficient than using the maximum coverage formulation.

CONCLUSION

We have presented an integrated GIS-based approach to obtain optimal visibility graphs. We illustrated the use of the set cover problem and the maximum coverage problem formulations to discretize structured and unstructured real practical environments. The detailed method integrates GIS tools, computational geometry, and optimization techniques. We have shown that near-optimal results can be obtained in very short times. We believe that in critical situations with short response times, an optimal visibility graph map, computed in a reasonable time, provides an efficient basis for real-time planning of complex emergency operations. Our main contribution resides in the integration of the GIS and optimization in order to solve real size problems in a reasonably short time. The method is compatible with other optimization techniques. Further research includes the development of dynamic visibility constraints such as smoke and fog.

ACKNOWLEDGMENTS

This project was partly funded by MITACS, National Science and Engineering Research Council of Canada and Defence R&D Canada-Valcartier. We thank F.-H. Rouet for having programmed part of the code in C++, and the anonymous reviewers for their insightful comments and suggestions.

REFERENCES

1. Dawes, B., Abrahams, D., and Rivera, R. (2009) Boost C++ libraries, *http://www.boost.org*.
2. De Berg, M. (2000) Computational geometry: algorithms and applications. Springer Science & Business.
3. De Floriani, L., and Magillo, P. (2003) Algorithms for visibility computation on terrains: a survey. *Environment and Planning B*, *30*(5), 709-728.
4. Emrich, C. T., Cutter, S. L., and Weschler, P. J. (2011) GIS and emergency management. *The SAGE Handbook of GIS and Society,* London, Sage, 321-43.
5. Fabri, A., and Pion, S. (2009) CGAL: The computational geometry algorithms library, Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 538-539.
6. Garey, M. R., and Johnson, D. S. (1979) Computers and intractability*: A guide to the theory of NP-Completeness*. San Francisco, CA. Freeman.
7. Goodchild, M. F. and Lee, J. (1989) Coverage problems and visibility regions on topographic surfaces. *Annals of Operations Research*, 18, 175–186.
8. Lamontagne L., Rouet F. H., Abi-Zeid I., in collaboration with J.-F. Potvin (2008), Étude d'algorithmes de poursuite-évasion sur graphes de visibilité, Research Report II, 40 pages.
9. Latombe, J.C. (1990) Robot Motion Planning, Springer International Series in Engineering and Computer Science.
10. Morin, M., Lamontagne, L., Abi-Zeid, I., Lang, P., and Maupin, P. (2010) The Optimal Searcher Path Problem with a Visibility Criterion in Discrete Time and Space, *Proceedings of the 12th International*

Conference on Information Fusion, Seattle, WA, 2217-2224.

11. Morin, M., Papillon, A.P., Laviolette, F., Abi-Zeid, I., and Quimper, C.G. (2012) Constraint Programming for Path Planning with Uncertainty: Solving the Optimal Search Path problem, *Proceedings of the 18th Conference on Principles and Practice of Constraint Programming*, Québec, Canada, 2012, 988-1003.
12. Murray, A. T., Kim, K., Davis, J. W., Machiraju, R. and Parent, R. (2007) Coverage Optimization to Support Security Monitoring, *Computers, Environment and Urban Systems*, 31, 133-147.
13. Obermeyer, K. J. and Contributors (2008), The VisiLibity library, A C++ library for floating-point visibilitycomputations, *http://www.VisiLibity.org*.
14. Pellier, D., and Fiorino, H. (2005) Coordinated exploration of unknown labyrinthine environments applied to the pursuit evasion problem, *Proceedings of the fourth international joint conference on Autonomous Agents and Multiagent Systems* (pp. 895-902).