CP 8215 Literature Review

KODY MANASTYRSKI

K-visibility is the field within computational geometry which focuses on whether a point is able to 'see' a region within a polygon. Unlike visibility as humans and animals experience it (though not excluding it) k-visibility allows for k obstacles to be between the boundary and the origin of transmission or the viewer. This study has applications in wireless communications primarily. Within this article a selection of 5 recent articles on the topic of k-visibility will be critiqued to give a snapshot of what the study of k-visibility focuses on, and its common methods.

ACM Reference Format:

Kody Manastyrski. 2021. CP 8215 Literature Review. 1, 1 (October 2021), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The study of computational geometry is a sub-discipline within computer science focusing on various geometric applications and problems. The field of k-visibility is the subset of computational geometry which is interested in 'illumination' problems. This class of problems asks about visibility regions, which is to say regions for which a line connecting two points is either completely uninterrupted or intersects at most k boundaries (where k is some real positive integer). With this definition, the visibility that humans typically experience is considered 0-visibility, or normal visibility. Extending beyond normal visibility has applications in wireless communications and graphics.

In this paper five papers on k-visibility will be summarized and critiqued, and extensions to the work will be proposed. In Section 2 summaries are provided per paper. Section 3 contains a discussion of potential further work for the results of each paper.

2 SUMMARIES

In this section we briefly outline the details of each paper. The papers will be ordered chronologically, starting from the oldest.

2.1 A Time Space Trade-off for Computing the k-visibility Region of a Point in a Polygon

In this work, Bahoo et al. ([2]) provided an algorithm for computing the k-visibility region of a simple polygon. A simple polygon is a polygon which is not self-intersecting. The work was done under the assumption of limited memory, which is specifically divided into a read/write input size for memory and a write only output stream. Under this model, called a limited workspace, a word is considered to be log(n) bits of the input size, n. This situation is common for wireless routers and distributed sensors, and, at the time of writing, mobile devices.

The main contribution of this work is a pair of algorithms which bring the trade off between time complexity and space complexity into the discussion. The first algorithm is primarily concerned with memory, while the second is

Author's address: Kody Manastyrski.

Manuscript submitted to ACM

interested in speed. The first algorithm works by first checking for a critical vertex (a vertex which has its incident edges on one side of a ray cast from a query point). If no critical vertex is found, then the k-visibility region is the polygon, otherwise the coordinate system is centered on the query point, q, and transformed in order to make the found critical vertex lay on the positive x-axis. A ray is then cast, and scanned in counterclockwise order, looking for the edge which has rank k+1. k+1 is chosen, because all edges of rank less than that are 0-visible from q. This is done for every critical vertex exhaustively. A special condition is created after the first critical vertex is processed. In this condition, a cone may be encountered while scanning the ray, which is the result of two ray regions intersecting. In this case, the intersection of this cone and the polygon's boundary is added to the reported results, since this represents a constrained region of visibility. In the event that an intersection is found at or before the k+1st ranked edge of the current or previous edge, report the chain of intersected edges until the cone is left. In this algorithm, a constant space is used, and the time complexity is O($kn + cn$), where n is the total count of vertices in the polygon, and c is the number of critical vertices in the polygon. The second algorithm uses the first as a springboard, but improves on speed. It begins the same, however it selects the first s vertices with the smallest positive angles measured from q. These angles are then processed as above, but the candidates for future windows (rank ≥ k+1) are placed into a balanced binary tree. This is done until all vertices are processed. The algorithm runs in O(s) space, and O($\frac{cn}{s} + c \log s + \min \left\{ \left\lceil \frac{k}{s} \right\rceil n, n \log \log_s n \right\}$) expected time.

The paper's methods build off of previously established methods. The paper starts with an O(nr) algorithm, which is not described, but rather cited. This algorithm is described to work in a similar manner to the constant space algorithm, but only computes a 0-visibility region. The work, then, is an extension of the aforementioned work to a general k-visibility model. As with many papers presenting algorithms, proofs are provided.

2.2 Combinatorics and Complexity of Guarding Polygons with Edge and Point 2-Transmitters

The k-visibility problem, as with any other problem, is often researched in a single case capacity. Cannon et al ([5]) consider the 2-visibility problem, and show NP-hardness in addition to showing a lower bound for the number of transmitters. In this article, attention was paid to the type of transmitter, where previously either a location was given or the type was implied to be vertex transmitters. The proofs for NP-hard membership of both cases follow Cooke's method of reduction to another problem. In the case of the point transmitter, the line coverage problem was used, and in the edge transmitter's case the 3-SAT problem. The proofs themselves range from straightforward to quite involved. Having proven the membership, the bounds for point and edge transmitters are then considered. For point transmitters, a bound of $\lceil n \rceil 6$ 2-transmitters was found to be sometimes necessary, while for edge transmitters bounds rely on specific polygon properties.

2.3 Modem Illumination of Monotone Polygons

Aichholzer et al ([1]) provide a concrete upper bound on x-monotone polygons. The article is split into two categories, simple and orthogonal. The work follows a mathematical structure, providing lemmas and proofs for every step made. It is worth note that the convention in this work for vertex labelling is specifically stated to be ascending order by x coordinate. This is mentioned because all of the work relies on shorthand that relies on this convention. In addition, the work adopts the term k-modem to refer to the center of the k-visibility region. Because of this structuring, a summary would end up being a concise list of the lemmas and thus will not be included here. Instead, we will detail the major contributions from the two sections. The simple polygon section provides an upper bound of $\left\lceil \frac{n-2}{2k+3} \right\rceil$ k-modems for any polygon. In addition to this, a polygon of $n = 2k + 5$ may be fully illuminated by a single k-modem placed near the center on the left or right half. Finally, a polygon of $n = 2k + 3$ may be illuminated by a modem placed at the $k + 2nd$ vertex. The orthogonal section is much shorter, and thus has less to offer in terms of quantity. The most major contributions are for polygons of $n = k + 7, k \geq 3$ and polygons of $n \leq k + 5$. These contributions are similar in that they reference the extreme right or left edge of the polygon. In the former case, a stair polygon (either upper or lower, and either left or right) may be fully illuminated by a k-modem placed on a point on the extreme face opposite the stair. In the latter case, all polygons fitting the constraint may be fully illuminated by a point on their extreme left or right edges.

2.4 Watchtower Crossing for k-visibility

The watchtower problem is another classic problem in computational geometry. In this case, the boundary is not necessarily a closed polygon, and the goal is to place towers at such positions and heights in order to 'illuminate' the entire terrain. This can be likened to watchtowers along a mountain range. Within this work, Bahoo et al ([3]) make the assumption that the given boundary is a x monotone chain. This assumption provides for a terrain which does not overhang itself. There are two constraint definitions which are considered, discrete and continuous placement. In the discrete case, a tower may only be placed at vertices defining the terrain, whereas the continuous case allows for placement anywhere. An additional assumption of a 2D plane is provided. This assumption simplifies the problem, and means that the solution may not scale to higher dimensions.

The continuous algorithm works by bounding the plain at the bottom by the terrain, and at the top and sides with horizontal and vertical lines (respectively). A k-kernel, which is a set of all points for which all other possible positions are k-visible, is computed using an algorithm in the paper's background research. The algorithm is not described, but its time complexity is, and is O($n^2 logn + h$), where h is the complexity of the kernel in terms of boundary vertices. This k-kernel then defines a region of possible tower endpoints. Finding the minimal length line segment between the kernel and the terrain is then the optimal solution for 1 tower. This may be done in linear time by considering only the region of the kernel above the terrain, and moving along linearly checking the distance and maintaining a minimal distance record. If the distance is ever 0, then an early end may be had, and the optimal solution is found by definition. The algorithm runs in O($(n^2 + h)logn$) time.

In the discrete case, the algorithm works off of a similar strategy. Instead of a kernel, however, a k-visibility region is calculated for each vertex. The overlap of these visibility regions is then only considered, since this represents the sum total of the feasible locations for the top of a tower. The optimal location may be located, then, by threading a vertical line through this region at each of the vertices. Next, the endpoints of each visibility region are examined, and when a segment which contains at least one of each visibility region's endpoints is found, then a solution region is found, and the lowest point in all such possibilities is the optimal solution.

2.5 Computing the k-visibility Region of a Point of a Polygon

In our final article, Bahoo et al ([4]) propose an algorithm for computing the k-visibility region centered on a point in a simple polygon. This work was motivated by a want to bridge the gap between the previous algorithms for k-visibility of $\Theta(nlogn)$ (when k > 0) and the computation of a 0-visibility region algorithms which run in O(n) time. The result of this investigation yielded an O(nk) algorithm, which represents a marked improvement. The algorithm rakes in a polygon of n vertices, and works by first ensuring that no vertex is co linear with the query point q. If a vertex is, then the polygon is rotated until this is not the case. Next, a horizontal line passing through q is used to partition the polygon into two regions, which are then closed. This is done by finding the points where the line intersects with the boundary of the polygon, and creating a three edge sequence in order to form a complete boundary in such a way that the regions are fully closed and singular polygons. The next step is to obtain a radial decomposition by translating q to $\pm\infty$ (depending on whether the region is upper or lower), then converting the coordinate system to homogeneous coordinates and projectively transforming it centering on q. Lines are then tracked from each vertex towards and away from q until they intersect with a boundary, this results in what is known as a trapezoidal decomposition. Then the inverse transformations are applied to the trapezoidal decomposition to create a radial decomposition about q. This then defines the 0-visibility region of q, and it is simply a matter of extending the region to k further intersections to find the solution for k-visibility. To do this, the algorithm checks points between two points in the current visibility region, and adds new points to the visibility region (thereby constructing a m+1 region, where m is the current k value of the visibility region). Once the kth iteration is reached, the k-visibility region has been computed, and the region is returned.

-

3 DISCUSSION

We now move on to the discussion of each paper with a critical eye. As with the previous section, this will be divided per paper. Since there are multiple papers, the discussion will be kept brief as we have little in the way of criticism to add to the discussion.

3.1 A Time Space Trade-off for Computing the k-visibility Region of a Point in a Plane

Given that computational geometry directly lends itself to applications in wireless networks, the central focus of this work is without a doubt significant. As to the validity of the claim, it seems tentatively valid with a caveat. It seems that the reporting step, in the worst case, would use more than constant memory space. Since the O(s) memory algorithm builds off of the constant space algorithm, the criticism also extends.

3.2 Combinatorics and Complexity of Guarding Polygons with Edge and Point 2-Transmitters

The categorization of problems is significant, and a task which does not start from an obvious conclusion. In the case of this work, the conclusion of NP-hard categorization is valid, and the proof is clear to follow and well laid out. The claims for the bounds also follow sound logic, but fall short in significance. The bounds in the case of the edge guards are for very specific polygons (those with 6 and 12 vertices). The most significant claim from the second half of the paper is for point polygons, which improves the lower bound for transmitter count without limitation.

3.3 Modem Illumination of Monotone Polygons

As with any bound improvement, the contributions in this paper are significant. The logic is also beyond a reasonable doubt. We found it quite hard to come up with any criticism for the work that was not related to the style, which is outside the scope of our work here. The work does leave unanswered the question of what dimensions are being worked within. The only mention of dimensions is for a related work in the introduction. It seems that 2 dimensions is the assumed context, but without it being explicit the claims have some small doubt.

3.4 Watchtower Crossing for k-visibility

The watchtower problem is a classic in computational geometry. Adding k-visibility to the watchtower problem provides a significant interest. This work was well presented, and we can find no logical reason to dispute the either the claims or conclusions.

3.5 Computing the k-visibility Region of a Point of a Polygon

This work provides a significant contribution for computing k-visibility regions. The contribution of an O(kn) algorithm serves to display the relationship between the base 0-visibility situation, and the generalized k-visibility situation.

4 SYNTHESIS

In this section we discuss the possible directions research could continue in. Since we find a lot of common potential in the articles, we will not be discussing each article individually. Brief mention for specific directions of specific works will be provided, but as part of the general discussion rather than on a case by case basis.

For all of the works, we find that the most significant direction for further research would be to generalize to higher dimensions. Since the main application for k-visibility is in wireless networks, the questions of how the placement of a transmitter or modem in a 3D environment is quite a natural fit. Beyond this, none of the works mention so called holes, which are spaces in a polygon where the boundary defines an additional closed polygon which is excluded from the original. In this situation the bounds would no doubt change for modem requirement as well as algorithmic complexity. This situation does arise in real world scenarios where cavities exist within interiors of buildings specifically for support purposes. additionally the works could consider situations where the faces of a polygon have 'dampening' properties which require a k value other than 1 to penetrate.

5 CONCLUSION

We have seen a selection of five papers from the field of k-visibility within computational geometry. We have discussed what they did, what their significance is, and potential future endeavours for these works. The works themselves pose as an excellent way for those who are unfamiliar with the topic to be introduced, and become familiar with the various ways in which previous work has manifested. Despite what we have seen, k-visibility remains an immature field of study, with many venues beyond those proposed earlier.

REFERENCES

[1] Oswin Aichholzer, Ruy Fabila-Monroy, David Flores-Peñaloza, Thomas Hackl, Jorge Urrutia, and Birgit Vogtenhuber. 2018. Modem illumination of monotone polygons. *Computational Geometry: Theory and Applications* 68 (3 2018), 101–118. https://doi.org/10.1016/j.comgeo.2017.05.010

[2] Yeganeh Bahoo, Bahareh Banyassady, Prosenjit Bose, Stephane Durocher, and Wolfgang Mulzer. 2016. A Time-Space Trade-off for Computing the k-Visibility Region of a Point in a Polygon. (3 2016). https://doi.org/10.1016/j.tcs.2018.06.017

[3] Yeganeh Bahoo, Prosenjit Bose, and Stephane Durocher. 2019. Watchtower for k-crossing Visibility. http://www.cs.umanitoba.ca/~durocher/research/pubs/durocherCCCG2010.pdf

[4] Yeganeh Bahoo, Prosenjit Bose, Stephane Durocher, and Thomas C. Shermer. 2020. Computing the k-Visibility Region of a Point in a Polygon. *Theory of Computing Systems* 64 (10 2020), 1292–1306. Issue 7. https://doi.org/10.1007/s00224-020-09999-0

[5] Sarah Cannon, Thomas G. Fai, Justin Iwerks, Undine Leopold, and Christiane Schmidt. 2018. Combinatorics and complexity of guarding polygons with edge and point 2-transmitters. *Computational Geometry: Theory and Applications* 68 (3 2018), 89–100. https://doi.org/10.1016/j.comgeo.2017.06.004