CP 8215 Literature Review

KODY MANASTYRSKI

K-visibility is the field within computational geometry which focuses on whether a point is visible to a region within a polygon. Unlike visibility as humans and animals experience it (though not excluding it) k-visibility allows for a number of obstacles, k, to intersect lines between the point and the region. This study has applications in wireless communications primarily along with minor applications in graphics and robotics. Within this article a selection of 5 recent articles on the topic of k-visibility are 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 (November 2021), 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The study of computational geometry is a sub-discipline within computer science focusing on geometric problems and applications. The field of k-visibility is the subset of computational geometry which is interested in so called illumination problems. This class of problems deals with regions which have the property that 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, and is called normal visibility. Extending beyond normal visibility can be applied to wireless communications, as well as graphics and robotics.

In this paper five papers on k-visibility will be summarized and critiqued, and extensions to the work will be proposed. Section 2 provides summaries of each of the 5 papers of interest. Section 3 is about criticisms for each paper. Section 4 is about potential future work that is unmentioned in the papers.

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 non self-intersecting polygon without regions that are enclosed within an additional polygon (called a simple polygon). 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 of memory 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.

Author's address: Kody Manastyrski.

Manuscript submitted to ACM

The main contribution of this work is a pair of algorithms which show a trade-off between time and space complexities. The first algorithm is primarily concerned with memory, while the second is 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 line from a query point to the vertex). If no critical vertex is found, then the k-visibility region is the polygon, otherwise a coordinate system is defined with the query point, q, at its center and the found critical vertex somewhere on the positive x-axis. A line is scanned in counterclockwise order and each edge encountered during the scan is added to a list. In this list, each edge has a rank which indicates when it was encountered by the line. Scanning continues until the edge with rank k+1 is encountered. k+1 is chosen, because all edges of rank less than that are 0-visible from q. This is repeated exhaustively for all critical vertices. After the first critical vertex is processed, intersections between the current scan region and previous scan regions is possible. When this condition occurs, the intersection of the current cone and the polygon's boundary is added to the reported results, since this represents a constrained region of visibility, and continuesuntil the scan line is no longer is within the cone. In this algorithm, a constant amount of memory 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, where s is a positive integer less than or equal to the count of critical vertices. These vertices are then processed as above, but the candidates for future windows (rank $\geq$ 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 are based on a previous O(nr) algorithm, which is not described, but rather cited. From context it is clear that the cited algorithm computes a 0-visibility region in a similar manner to the constant memory algorithm above. As with many papers describing algorithms, some mathematical proofs are provided or referenced in place of empirical evidence.

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 that the problem belongs to the class of NP-hardness in addition to showing a lower bound for the number of transmitters. NP problems are able to be solved in polynomial time as long as the machine solving them is non-deterministic, hence the name non-Deterministic Polynomial (NP). A problem is NP-hard if there is at least one NP-complete problem that reduces to it, though NP-hard does not necessarily belong to NP. On the other hand a problem is NP complete if it belongs to the NP class and the NP-hard class. In this article, attention was paid to the type of transmitter, as opposed to a location being given explicitly or a vertex transmitter being assumed or implied. The first proof for NP membership was done by transforming the problem of finding a minimum point 2-transmitter to finding a minimum line covering. The second proof transformed minimum edge 2-transmitter coverage into a problem posed by Lee and Lin. This transformation and subsequent proof is unfortunately quite complicated to describe, and instead a reference to it is provided ([6]). The second half of the paper deals with upper bounds on the number of transmitters required to illuminate a polygon. For point transmitters, a value of $\lceil \frac{n}{6} \rceil$ 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. For context, something is orthogonal if it is either parallel or perpendicular to an axis. 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 comes from 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 provided. Instead, we will detail the major contributions from the two sections. The simple polygon section provides an upper bound of $\lceil \frac{n-2}{2k+3} \rceil$ 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. The most major contributions are for polygons of $n = k + 7, k \geq 3$. These contributions are similar in that they reference the extreme right or left edge of the polygon. This case deals with stair polygons. A stair polygon is a polygon which is bounded by a series of orthogonal edges which have vertices that are strictly increasing/decreasing (as applies to the orientation). Aicholzer et al. show that a stair polygon may be fully illuminated by a k-modem placed on a point on the extreme face opposite the stair.

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 is instead contextualized as a terrain. The goal is to place towers at such positions and heights that they illuminate the entire terrain. This can be likened to watchtowers along a mountain range, or lighthouses along a coastline. Within this work, Bahoo et al ([3]) make the assumption that the given boundary is a x monotone chain. This assumption provides a terrain which does not overhang itself and thus create regions that are 0-visible from within a significantly constrained region. There are two constraint definitions for the problem 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 along the terrain. In addition, the space the problem takes place in is assumed to be 2D. This assumption simplifies the problem, but means that the solution may not scale to higher dimensions.

The continuous algorithm works by bounding the plane 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 given as 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. The possible region is searched linearly, and the minimum distance is maintained. At each step of the search, the distance is represented by a line from the terrain to the bottom of the k-kernel region. If the distance is ever 0, then the algorithm may terminate early, since this is the optimal solution 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 set of overlaps of these visibility regions is then 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 each of these regions 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 is interested in bridging the gap between the previous algorithms for computing a k-visibility region with $\Theta(nlogn)$ (when k > 0) time 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 takes in a simple polygon of n vertices, and works by first ensuring that no vertex is co-linear with the query point q. If a vertex is co-linear, 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 ±∞ (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 either a boundary or each other. This results in what is known as a trapezoidal decomposition. Then the inverse transformations of the query point and coordinate system 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 decomposition 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 since there is little we have to add to the discussion. It is worth note that since most of the work considered is theoretic, criticisms must remain fine grain since a proof is either correct or not, but why it's correct or not is more nuanced.

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 to this algorithm.

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.

Fig. 1. Caption

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 dimensional space is being considered. 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 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. One criticism that could be found for this work is that the method for deciding where to close the regions in the early part of the algorithm does not have any proof or guarantee that it would not result in a polygon with holes. This could result in falsely under-reporting a region, since this would add boundaries which are not present in the original polygon.

4 SYNTHESIS

In this section we discuss the possible directions research could continue in.

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. Finally, to better represent wireless communications signal attenuation could be brought into the context. This could lead to k representing the signal transmission power of a transmitter, rather than just its tolerance for obstacles. Such a case could result in a mapping to fractional k values as well. This is in itself a potential venue for future work, as it is unclear what this would represent.

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

[6] D. T. Lee and Arthur K. Lin. 1986. Computational Complexity of Art Gallery Problems. *IEEE Transactions on Information Theory* 32 (1986), 276–282. Issue 2. https://doi.org/10.1109/TIT.1986.1057165