Elsevier Editorial System(tm) for
Theoretical Computer Science
Manuscript Draft

Manuscript Number: TCS-D-18-00624R1

Title: Visibility Testing and Counting for Uncertain Segments

Article Type: Regular Paper (10 - 40 pages)

Section/Category: A - Algorithms, automata, complexity and games

Keywords: computational geometry; visibility; randomized algorithm; approximation algorithm; probabilistic segments

Corresponding Author: Miss sharareh alipour,

Corresponding Author's Institution: Institute for research in fundamental science

First Author: mohammad ali abam

Order of Authors: mohammad ali abam; sharareh alipour; mohammad ghodsi; mohammad mahdian

We really appreciate the referees during the revision process. We believe that their comments improved the quality of our paper more than what we expected.
In the following we answer each of the comments separately. Thanks again.

Reviewer #1:

-The authors presented algorithms for the Probabilistic Visibility Testing Problem (PVTP) and the Probabilistic Visibility Counting Problem (PVCP). The probability is on whether a segment exists with some probability, which is not explained by any valid application.
The main results are standard: (1) it is #P-complete (a reduction from #perfect-matching), (2) when z=2 (i.e., whether a segment exists nor not exists
with certain probability), PVTP can be solved in O(n\log n) time with dynamic programming. The remaining parts are just a combination of (2) with some
standard tricks. I am not sure whether TCS is the best venue for this paper, somewhere like Computational Geometry or IPL might be more suitable.

The paper is in general well-written.

-Minor Comments: Page 5, fig 3, caption: "I'_1,I'_2,I'_3 are the intervals that can cover [a,b]", I guess "[a,b]" should be a'.

-You are right, Corrected

Also, sth like "p_isol(-)" looks weird, add a "\cdot" between p_i and sol(-).

-Corrected.

Reviewer #2: In this paper, authors study on the visibility testing and counting for uncertain segments. This paper presents the definitions of Probabilistic Visibility Testing Problem(PVTP) and Probabilistic Visibility Counting Problem(PVCP). Then, they show the PVTP is #P-complete. Meanwhile, they show that the PVTP can be answered in O(nlogn) time when the uncertainty is only about whether segments exist and not about

their location. And the PVCP can be solved in O(n2logn) time. This paper has excellent work about the Visibility Testing and Counting Problem for unique kind of uncertain segments.
However, there's some doubt in my mind.

1. The authors only study the special case where the uncertainty is only about the existence of the segments and not about their location. It is very unique.

-For the PVTP we proved that when uncertainty is about the location of segments the problem is #p complete.
For PVCP , when the uncertainty is about the location of segments, we gave two approximation algorithms.

2. The authors claimed "we can show that one can preprocess S in O(n5logn) time into a data structure of size O(n4), so that PVTP queries can be answered in O(logn) time. Our algorithm for PVTP combined with linearity of expectation gives an O(n2logn) time algorithm for PVCP." in abstract. It is inaccurate and mislead. The "preprocess" is a part of the process of PVTP or PVCP. If the ENTIRE process of the PVTP takes O(n5logn) time, the application of algorithm will be limited. The authors need to explain the circumstances under which the "preprocess" part can be processed separately from PVTP and PVCP in Section.1.

-You are right. We changed that part of the abstract to explain the running times accurately.

3. The authors have proved their theories very well. However, the authors should also provide appropriate code to help understand the process of the algorithm.

-We added the code as you said.

Visibility Testing and Counting for Uncertain Segments

Mohammad Ali Abam * Sharareh Alipour† Mohammad Ghodsi ‡
Mohammad Mahdian §

December 11, 2018

Abstract

We study two well-known planar visibility problems, namely visibility testing and visibility counting, in a model where there is uncertainty about the input data. The standard versions of these problems are defined as follows: we are given a set $\mathcal{S}$ of n segments in $\mathbb{R}^2$, and we would like to preprocess $\mathcal{S}$ so that we can quickly answer queries of the form: is the given query segment $s \in \mathcal{S}$ visible from the given query point $q \in \mathbb{R}^2$ (for visibility testing) and how many segments in $\mathcal{S}$ are visible from the given query point $q \in \mathbb{R}^2$ (for visibility counting).

In our model of uncertainty, each segment may or may not exist, and if it does, it is located in one of finitely many possible locations, given by a discrete probability distribution. In this setting, the probabilistic visibility testing problem (PVTP, for short) is to compute the probability that a given segment $s \in \mathcal{S}$ is visible from a given query point q and the probabilistic visibility counting problem (PVCP, for short) is to compute the expected number of segments in $\mathcal{S}$ that are visible from a query point q. We first show that PVTP is $\#P$-complete. In the special case where uncertainty is only about whether segments exist and not about their location, we show that PVTP is solvable in $O(n \log n)$ time. Our algorithm for PVTP combined with linearity of expectation gives an $O(n^2 \log n)$ time algorithm for PVCP.

Using the algorithm for PVTP, together with a few old tricks, we can show that one can preprocess $\mathcal{S}$ in $O(n^5 \log n)$ time into a data structure of size $O(n^4)$, so that each PVTP query for a fixed segment s can be answered in $O(\log n)$ time.

We also give a faster 2-approximation algorithm for this problem. At the end, we improve the approximation factor of the algorithm.

Keywords. computational geometry, visibility, randomized algorithm, approximation algorithm, probabilistic segments.

1 Introduction

Background. Visibility testing and visibility counting are basic problems in computational geometry. Visibility plays an important role in robotics and computer graphics. In robotics, for example, the efficient exploration of an unknown environment requires computing the visibility polygon of the robot or the number of visible objects from the robot or test whether the robot sees a specific object. In some computer graphics applications, also, it is important to identify the objects in a scene that are illuminated by a light source.

*Computer Engineering Department, Sharif University of Technology
†School of Computer Science, Institute for Research in Fundamental Sciences (IPM)
‡Computer Engineering Department, Sharif University of Technology and School of Computer Science, Institute for Research in Fundamental Sciences(IPM)
§Google research

Two points $p, q \in \mathbb{R}^2$ are visible from each other with respect to $\mathcal{S}$, if there exists no segment $s \in \mathcal{S}$ intersecting line segment $\overline{pq}$. We say that a segment $\overline{st} \in \mathcal{S}$ is visible from a point p, if a point $q \in \overline{st}$ can be found from which p is visible. In this paper, we consider two planar visibility problems; visibility testing and visibility counting. For a set $\mathcal{S}$ of n segments in $\mathbb{R}^2$ and a point q, in the visibility testing problem, we want to test whether q sees a given segment $s \in \mathcal{S}$. In the visibility counting problem we want to count the number of segments in $\mathcal{S}$ that are visible from q. For simplicity we assume all the segments are contained in an arbitrary bounding box, denoted by $\mathbf{B}$.

Uncertain data. It is not surprising that in many real-world applications we face uncertainty about the data. For geometric problems like visibility, this means uncertainty about the location of the input set. There are multiple ways to model such uncertainty. For example, we can assume each object lies inside some region, but not exactly where in that region, and use this assumption to prove bounds on the quantity of interest. Such a model is used in [15]. Alternatively, we can use a discrete probability distribution to model uncertainty. This "stochastic" approach is used in [1, 12]. We choose the latter approach in this paper. In particular, our model of uncertainty is very similar to the model used in [12].

Related work. There is significant prior work on the non-stochastic version of the problems studied in this paper. There are some work dedicated not only to the exact computing [6, 13, 16] of the problem but also to approximate computing [3, 4, 5, 10, 13]. In both, time-space trade-offs haven been considered.

In real application there are situations where we need to model the problems based on uncertain data (See [1, 15, 11]). In [7], Buchin and et.al computed the visibility between imprecise points among obstacles. For example in robotics there are situations where obstacles also are moving. We can model the movements of obstacle by considering uncertainty in their locations. This leads us to define the uncertain model of two visibility problems and propose algorithms to compute them.

Problem statement. Suppose we are given a set $\mathcal{S}$ of n uncertain segments. More precisely, we are given a discrete probability distribution for each $s_i \in \mathcal{S}$, that is, we have a set $\mathcal{D}_i = \{s_{i,1}, \cdots, s_{i,m_i}\} \cup \{s_{i,0} = \perp\}$ of possible locations with associated probabilities $p_{i,j}$ such that $\mathbf{Pr}(s_i = s_{i,j}) = p_{i,j}$ and $\sum_j p_{i,j} = 1$. The special segment $\perp$ indicates that the segment s_i does not exist in $\mathcal{S}$. In this setting, the set $\mathcal{S}$ can be seen as a random variable (or random set) as it consists of probabilistic segments. This random variable gets its value from a sample space of size $\Pi_i(m_i + 1)$ with the probability being equal to $\Pi_{s \in \mathcal{S}} \mathbf{Pr}(s) \Pi_{s \notin \mathcal{S}} \mathbf{Pr}(s = \perp)$. Assume $z = \max\{1 + m_i\}$, i.e., z denotes the maximum size of the given distributions. A special case that we will pay special attention to is when $z = 2$. This is the case where the uncertainty is only about the existence of the segments, and not about their location.

It is natural to define the probabilistic version of visibility testing and visibility counting problems in the above setting where $\mathcal{S}$ is a random set:

- Probabilistic Visibility Testing Problem (PVTP): compute the probability that a given segment $s \in \mathcal{S}$ is visible from a given query point q, denoted by PVTP(q).
- Probabilistic Visibility Counting Problem (PVCP): compute the expected number of segments in $\mathcal{S}$ being visible from q, denoted by PVCP(q).

Our results. We first show that PVTP is $\#P$-complete. We then turn our attention to the special case where $z = 2$. We present an algorithm running $O(n \log n)$ time that answers PVTP. Then, we present a simple way of putting n uncertain segments into a data structure of size $O(n^4)$ such that queries can be answered in $O(\log n)$ time. Finally, we focus our attention to PVCP. Here, we present a polynomial-time 2-approximation algorithm that approximately solves PVCP. We then show how to preprocess $\mathcal{S}$ into a data structure of size $O(n^4)$ in order to approximately answer each query in $O(\log n)$ time. At the end, by using a result of [3], we improve the approximation factor from 2 to 1.5.

2 Probabilistic visibility testing

We start by a simple polynomial-time reduction from #perfect-matching problem to PVTP in order to show PVTP is $\#P$-complete. The #perfect-matching problem of computing the number of perfect matchings in a given bipartite graph, is known to be $\#P$-complete [14] even for 3-regular bipartite graphs [9]. We next explain the details.

Suppose a bipartite graph $G = (U, V, E)$ is input to #perfect-matching problem where $U = \{u_1, \cdots, u_n\}$ and $V = \{v_1, \cdots, v_n\}$ are vertex parts of G and E is the edge set of G. For the given bipartite graph, we construct an instance of PVTP and introduce a query point q and a query segment s such that each perfect matching uniquely corresponds to one element of the sample space of uncertain segments in which s is not visible from q. Consider n intervals $[i, i+1]$ on the x-axis where i changes from 0 to $n-1$. Imagine the interval $[i, i+1]$ corresponds to the vertex v_i; denoted by $I(v_i)$. For each vertex $u_i \in U$, we define an uncertain segment $\mathcal{D}_i = \{I(v_j) | \{u_i, v_j\} \in E\}$ with the uniform distribution—note that in this instance each uncertain segment always exists. We add one more uncertain segment s consisting of one segment with probability 1 whose endpoints are $(0, -1)$ and $(n, -1)$. Finally, let q be the point in $(n/2, n)$ (See figure 1).

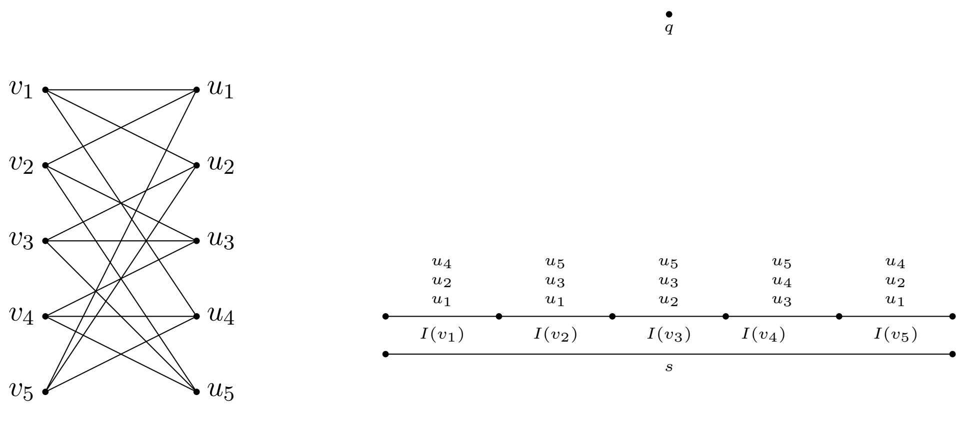

Figure 1: Each matching in the left side corresponds to a set of segments that cover s in the right side and each set of segments that cover s corresponds to a matching.

Segment s is not visible from q iff the interval $[0, n]$ is completely covered by the uncertain segments defined on the x-axis. There are n such uncertain segments and each covers exactly 1 unit of $[0, n]$. Therefore, each uncertain segment must cover exactly one of n unit intervals. So, the number of perfect matchings is equal to the number of ways that s is covered by the uncertain segments. Therefore, we conclude the following theorem.

Theorem 2.1. *PVTP is $\#P$-complete.*

In the remainder of this section, we restrict ourselves to the special case where $z = 2$, i.e., each uncertain segment either does not exist or exists in only one possible location. Suppose we are given n uncertain segments $s_1, \cdots, s_n$. Let $\mathbf{Pr}(s_i \in \mathcal{S}) = p_i$ which of course implies $\mathbf{Pr}(s_i \notin \mathcal{S}) = 1 - p_i$.

Next, we explain how to compute $\mathbf{Pr}(q \text{ sees } s)$ for the given segment s and point q. If $s \notin \mathcal{S}$, q of course cannot see s. Therefore, $\mathbf{Pr}(q \text{ sees } s) = \mathbf{Pr}(q \text{ sees } s | s \in \mathcal{S})\mathbf{Pr}(s \in \mathcal{S})$. This reduces our task to computing of $\mathbf{Pr}(q \text{ sees } s | s \in \mathcal{S})$. Let Δ be a triangle with vertex q and side s. Every other uncertain segment that does not intersect Δ, cannot prevent q from seeing s. Therefore, we can restrict ourselves to uncertain segments intersecting Δ. We project these uncertain segments to s with respect to q. Now, as the main ingredient, we must solve the following problem (See figure 2):

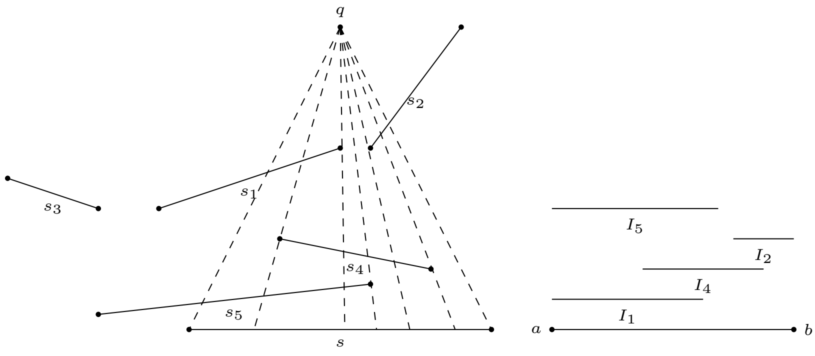

Figure 2: The projection of uncertain segments on s according to q defines four uncertain intervals.

- Suppose we are given n uncertain intervals $I = \{I_1, \cdots, I_n\}$ on the real line; each I_i exists with probability p_i. Compute the probability that the given interval $[a, b]$ is covered by the uncertain intervals, denoted by $\mathbf{Pr}([a, b] \text{ is covered})$.

Computing the desired probability seems to need $\Theta(2^n)$ time as the size of the sample space can be $\Theta(2^n)$ in the worst case. But, we next show how the dynamic programming paradigm helps us to perform the computation in $O(n \log n)$ time. For simplicity, we can assume the intervals have been sorted by their right endpoints and each I_i covers some part of $[a, b]$ i.e, $I_i \cap [a, b] \neq \emptyset$. Let $r(I_i)$ $(l(I_i))$ be the right (left) endpoint of I_i. We present the following recursive formula.

For each point $a' \in [a, b]$, let $sol(a')$ be the probability that $[a', b]$ is covered. So, $sol(a)$ is the probability that $[a, b]$ is covered. Let $S(a') = \{I'_1, ..., I'_l\}$ be the set of intervals that cover a' and they are sorted according to their right endpoints (See figure 3).

Lemma 2.1. *We define $sol(b) = 1$, then we have*

$$sol(a') = \textstyle\sum_{j=1}^{l} p'_j (\prod_{i=1}^{j-1}(1 - p'_i)) \cdot sol(r(I'_j)).$$

Proof. Suppose that $a' \in [a, b]$, so if $[a', b]$ is covered, then at least one of the segments in $S(a')$ should be chosen. There are l segments that cover a'. Since the segments in $S(a')$ are sorted according to their right endpoints then, the probability that I'_j is the first segment that covers a' is $p'_j \prod_{i'=1}^{j-1}(1 - p'_i)$. Recursively $[a', b]$ is covered with the

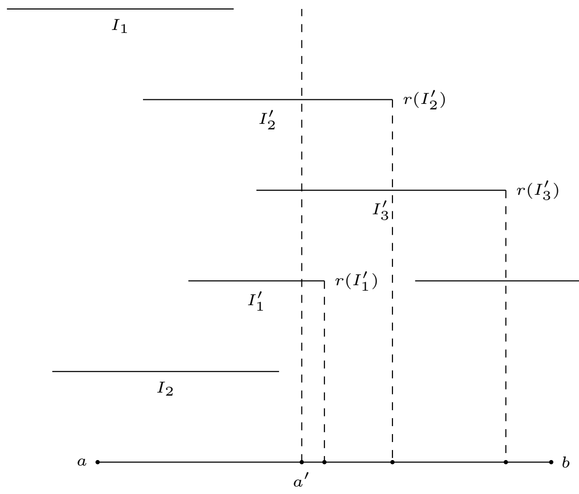

Figure 3: I'_1, I'_2 and I'_3 are the intervals that can cover a', so we have $sol(a') = p'_1 \cdot sol(r(I'_1)) + p'_2(1-p'_1) \cdot sol(r(I'_2)) + p'_3(1-p'_2)(1-p'_1) \cdot sol(r(I'_3))$.

probability of $sol(r(I'_j))$. So, we have

$$sol(a') = \sum_{j=1}^{l} p'_j \Big(\prod_{i'=1}^{j-1}(1-p'_i)\Big) \cdot sol(r(I'_j)).$$

■

The steps of our algorithm is stated in Pseudocode 1. Each right endpoint of the intervals can be covered by $O(n)$ of the intervals. In the recursive formula, we call each right endpoint at most once. For each $sol(r(I'_j))$ we have to compute $\prod_{i'=1}^{j-1}(1-p'_i)$, since the segments are sorted according to their right endpoint, for each $sol(r(I'_j))$ we multiply $\prod_{i'=1}^{j-2}(1-p'_i)$(the value of previous step) by $1-p'_j$, which means we can compute $sol(a)$ in $O(n^2)$ time. Next we propose a faster algorithm.

To fill the array sol, we sweep the endpoints from right to left and keep track of all intervals intersecting the sweep line in a binary search tree (BST, for short) over the right endpoint of intervals supporting insertion/deletion in $O(\log n)$ time. We augment each node of the BST with extra values in order to expedite our computation as we explain next.

Upon processing a right endpoint, say $r(I_i)$, we compute $sol(r(I_i))$, which is the sum of all the nodes of tree. This can be computed in $O(\log n)$ time. Then, we implicitly multiply all the nodes by $(1-p_i)$ and then add $r(I_i)$ to the tree with the value of $p_i \cdot sol(r(I_i))$. For the left endpoint of an interval, $l(I_i)$, we delete I_i, from the tree and implicitly divide all the right endpoints greater than $r(I_i)$ by $(1-p_i)$. This also can be done in $O(\log n)$ time. There are $O(n)$ endpoints, so the running time is $O(n \log n)$.

Theorem 2.2. *Given a point and a segment, PVTP can be answered in $O(n \log n)$ time when $z = 2$.*

Now, we preprocess the segments such that for any given query point q, PVTP can be answered in $O(\log n)$ time. First, connect each pair of the endpoints by a line and extend it until it hits the bounding box. These lines will partition the bounding box into $O(n^4)$ regions. For a fixed segment $s \in \mathcal{S}$, the answer to PVTP for all the points in a given

Algorithm 1 Algorithm for PVTP

Let Δ be the triangle with side s and vertex q.
For each $s' \in \mathcal{S}$, compute $s'(\Delta)$, the part of s' being inside Δ.
Project all $s'(\Delta)$ into s with respect to q, and store all projected intervals on s in the set I.
Let s be the interval $[a, b]$, and let the endpoints of intervals in I be real numbers in $[a, b]$.
Let $sol(a')$ be the probability that $[a', b]$ is covered by I. Set $sol(b) = 1$
Process the right endpoints of intervals in I in the decreasing order as follows.
Upon reaching at the right endpoint a', Let $S(a') = \{I'_1, ..., I'_l\}$ be the set of intervals that cover a' and they are sorted according to their right endpoints
Then, $sol(a') = \sum_{j=1}^{l} p'_j (\prod_{i'=1}^{j-1}(1 - p'_i)) \cdot sol(r(I'_j))$.

region is the same, because the combinatorial order of segments that cover s is the same for all the points inside that region. Therefore, in the preprocessing time we choose a point q_i from each region r_i and compute $\mathbf{Pr}(q_i \text{ sees } s)$ in $O(n \log n)$ time. So, for a given set of segments $\mathcal{S}$ and a segment $s \in \mathcal{S}$, we preprocess the segments in $O(n^5 \log n)$ time and $O(n^4)$ space such that for any given query point q, we locate the region r_i containing q in $O(\log n)$ time and return $\mathbf{Pr}(q_i \text{ sees } s) = \mathbf{Pr}(q \text{ sees } s)$.

3 Probabilistic visibility counting

In this section we study the probabilistic visibility counting problem. We start with some notation. For each subset $T \subset \mathcal{S}$, let $m_q(T)$ be the number of segments visible from q when the set of segments is T. So, the expected number of segments visible from q can be written as: $E(m_q) = \sum_{T \subseteq \mathcal{S}} \mathbf{Pr}(T) m_q(T)$, where $\mathbf{Pr}(T)$ denotes the probability that the set of realized segments is T. Another way to compute $E(m_q)$ is using linearity of expectations: $E(m_q) = \sum_{i=1}^{n} \mathbf{Pr}(q \text{ sees } s_i)$.

For the case $z = 2$, we can use the above identity and the algorithm in the previous section to compute $E(m_q)$ in $O(n^2 \log n)$ time with no preprocessing. Also as in the previous section, we can use preprocessing to reduce query time: the answer of $PVCP$ is the same for all the points in each region in the space partition. So, we can compute this number for all the regions in $O(n^6 \log n)$ preprocessing time and $O(n^4)$ space, such that for any query point q, $E(m_q)$ can be answered in $O(\log n)$ time. Now, we show how to approximately solve this problem more efficiently.

3.1 2-approximation of PVCP

In this section we propose a 2-approximation solution for PVCP. First, we present the following theorem

Theorem 3.1. [5] *Suppose that we are given a set S of n pairwise disjoint line segments in the plane. Let m_q be the number of visible segments from q. And let ve_q be the number of visible endpoints of the segments from q, then we have*

$$m_q \leq ve_q \leq 2m_q$$

Now, we use Theorem 3.1 to approximate PVCP.

Theorem 3.2. *Let $\mathcal{S}$ be a set of n uncertain segments where each segment $s_i \in \mathcal{S}$ has z possible locations. There is a 2-approximation algorithm for PVCP which runs in $O(n^2z^2)$ time.*

Proof. Let $T \subset \mathcal{S}$. Suppose that $m_q(T)$ and $ve_q(T)$ are the number of visible segments and visible endpoints in T w.r.t T, so we have $m_q(T) \leq ve_q(T) \leq 2m_q(T)$. So, we can conclude that,

$$\sum_{T \subset \mathcal{S}} \mathbf{Pr}(\mathcal{S} = T) m_q(T) \leq \sum_{T \subset \mathcal{S}} \mathbf{Pr}(\mathcal{S} = T) ve_q(T) \leq \sum_{T \subset \mathcal{S}} \mathbf{Pr}(\mathcal{S} = T) 2m_q(T).$$

In other words,

$$E(m_q) \leq E(ve_q) \leq 2E(m_q).$$

So, we compute

$$E(ve_q) = \sum_{i=1}^{n} \mathbf{Pr}(r(s_i) \text{ sees } q) + \mathbf{Pr}(l(s_i) \text{ sees } q).$$

We have

$$\mathbf{Pr}(r(s_i) \text{ sees } q)) = \textstyle\sum_{j=1}^{z} p_{i,j} \mathbf{Pr}(r(s_{i,j}) \text{ sees } q).$$

Let $s_{k,1'}, s_{k,2'}, ..., s_{k,l'}$ be the possible locations of s_k in $\mathcal{D}_k$ that cross $\overline{r(s_{i,j})q}$, the probability that s_k does not intersect $\overline{r(s_{i,j})q}$ is $p_k^{i,j} = (1 - p_{k,1'} - p_{k,2'} - ... - p_{k,l'})$.

$$\mathbf{Pr}(q \text{ sees } r(s_i)) = \textstyle\sum_{j=1}^{z} p_{i,j} p_1^{i,j} p_2^{i,j} ....p_n^{i,j}$$

We have $2nz$ possible locations for the endpoints and we can compute $P(q \text{ sees } r(s_i))$ in $O(zn)$, so $E(ve_q)$ is computed in $O(n^2z^2)$. ■

For $z = 2$ we present a faster algorithm. Suppose that $a \in s_i$ is an endpoint of s_i. Let $s'_1, s'_2, ..., s'_k$ be the set of segments that intersect $\overline{aq}$, since the probability of selection of the segments are independent, we have

$$\mathbf{Pr}(q \text{ sees } a) = p_i(1 - p'_1)(1 - p'_2)...(1 - p'_k).$$

Which yields: $E(ve_p) = \sum_{a \in s_i} \mathbf{Pr}(q \text{ sees } a)$.

So, for each endpoint, we need the segments that intersect $\overline{aq}$. We use the following theorem:

Theorem 3.3. [2, 8] *Let S be a set of n segments in the plane and $n \leq k \leq n^2$, we can preprocess the segments in $O_\epsilon(k)$ time such that for a given query segment s, the number of segments crossed by s can be computed in $O_\epsilon(n/\sqrt{k})$ time. Where $O_\epsilon(f(n)) = O(f(n)n^\epsilon)$ and $\epsilon > 0$ is a constant that can be made arbitrarily small.*

Note that in Theorem 3.3 if the segments are weighted, then in $O_\epsilon(n/\sqrt{k})$ time we can compute the product of the weights of segments that are crossed by s. Thus, by Theorem 3.3 we can compute $\mathbf{Pr}(q \text{ sees } a)$ in $O(n/\sqrt{k})$. So, for $2n$ endpoints, $E(ve_p)$ is computed in $n \cdot O(n/\sqrt{k})$. If $k = n^{\frac{4}{3}}$, then we have:

Theorem 3.4. *Let, S be a set of given segments and q be a given point. If each segment is chosen with probability p_i, then, the expected number of visible endpoints from q can be computed in $O_\epsilon(n^{\frac{4}{3}})$ which is a 2-approximation of $E(m_q)$.*

3.2 1.5-approximation of PVCP

In this section we use a result of [3] to improve the approximation factor of previous subsection. Here we assume that $z \geq 2$. For each point $a \in s_i$, let $\overrightarrow{qa}$ be the ray emanating from the query point q toward a and let $a' = ext_q(a)$ be the first intersection point of $\overrightarrow{qa}$ and a segment in S or the bounding box. We say that $a' = ext_q(a)$ is covered by a or the extension of a is a'. Let C_1 be the set of segments, s_i such that both their end-points are visible and $ext_q(r(s_i))$ and $ext_q(l(s_i))$ are on the bounding box or the same segment, i.e. there exists $s_j \in S$, such that ($ext_q(r(s_i)) \in s_j$ and $ext_q(l(s_i)) \in s_j$) or ($ext_q(r(s_i)) \in \mathbf{B}$ and $ext_q(l(s_i)) \in \mathbf{B}$). In [3], it is proved that

$$m_q \leq ve_q - |C_1| \leq 1.5m_q.$$

If the segments are uncertain, then by linearity of expectation we have

$$E(m_q) \leq E(ve_q) - E(|C_1|) \leq 1.5E(m_q).$$

In the previous section, we explained how to compute $E(ve_q)$. So, if we can compute $E(|C_1|)$, then we improve the approximation factor.

For each segment s_i, let $\mathbf{Pr}(s_i \in C_1)$ be the probability that $s_i \in C_1$. So, $E(|C_1|) = \sum_{i=1}^{n} \mathbf{Pr}(s_i \in C_1)$. Now, we explain how to compute $\mathbf{Pr}(s_i \in C_1)$.

For each realization $s_{i,j}$ of s_i, first, we want to compute the probability that $s_{u,v}$ is a segment that $ext_q(r(s_{i,j})) \in s_{u,v}$ and $ext_q(l(s_{i,j})) \in s_{u,v}$. Let $s_{k,1'}, s_{k,2'}, ..., s_{k,l'}$ be the possible locations of s_k that cross $\overline{qext_q(r(s_{i,j}))}$ or $\overline{qext_q(l(s_{i,j}))}$. Suppose that $P_k^{i,j,u,v} = (1 - p_{k,1'} - p_{k,2'} - ... - p_{k,l'})$.

So, the probability that both end-points of $s_{i,j}$ are visible to q and $ext_q(r(s_{i,j})) \in s_{u,v}$ and $ext_q(l(s_{i,j})) \in s_{u,v}$ is $P^{i,j,u,v} = p_{i,j}p_{u,v}\Pi_{k \neq i,u} P_k^{i,j,u,v}$. So,

$$\Pr(s_{i,j} \in C_1) = \sum_{u \neq i} P^{i,j,u,v}.$$

And

$$\Pr(s_i \in C_1) = \sum_{j=1}^{z} \Pr(s_{i,j} \in C_1).$$

We can compute $P(s_i \in C_1)$ in $O(n^2z^2)$ time. So, overall we can compute $E(|C_1|)$ in $O(n^3z^3)$ which results the following theorem.

Theorem 3.5. *Let, S be a set of given n probabilistic segments and q be a given query point. We can compute $E(ve_q) - E(|C_1|)$ in $O(n^3z^3)$ which is a 1.5-approximation answer of PVCP(q).*

4 Conclusion

We introduced a probabilistic variant of two well known visibility problems: visibility testing and counting. We proved that visibility testing problem in general case is $\#P$-complete. Then, we proposed a polynomial time for a special case of these problems and then gave an approximation algorithm for the probabilistic visibility counting problem. In future we want to study the complexity of these problems in some other special cases. Also, we want to study algorithms to approximate the answer of probabilistic visibility testing problem.

Acknowledgments

We thank Mahdi Safarnejad for his comments and helps.

References

[1] M. A. Abam, M. de Berg, and A. Khosravi. Piecewise-linear approximations of uncertain functions. In *Algorithms and Data Structures - 12th International Symposium, WADS . Proceedings*, pages 1–12, 2011.

[2] P. K. Agarwal and M. Sharir. Applications of a new space-partitioning technique. *Discrete & Computational Geometry*, 9:11–38, 1993.

[3] S. Alipour, M. Ghodsi, and J. Amir. Randomized approximation algorithms for planar visibility counting problem. *Theor. Comput. Sci.*, To be published, 2017.

[4] S. Alipour, M. Ghodsi, A. Zarei, and M. Pourreza. Visibility testing and counting. *Inf. Process. Lett.*, 115(9):649–654, 2015.

[5] S. Alipour and A. Zarei. Visibility testing and counting. In *Proceedings of the 5th Joint International Frontiers in Algorithmics, and 7th International Conference on Algorithmic Aspects in Information and Management*, FAW-AAIM'11, pages 343–351, 2011.

[6] T. Asano. An efficient algorithm for finding the visibility polygon for a polygonal region with holes. *IEICE TRANSACTIONS (1976-1990)*, 68(9):557–589, 1985.

[7] K. Buchin, I. Kostitsyna, M. Löffler, and R. I. Silveira. Region-based approximation of probability distributions (for visibility between imprecise points among obstacles). *CoRR*, abs/1402.5681, 2014.

[8] S. Cheng and R. Janardan. Algorithms for ray-shooting and intersection searching. *J. Algorithms*, 13(4):670–692, 1992.

[9] P. Dagum, M. Luby, M. Mihail, and U. V. Vazirani. Polytopes, permanents and graphs with large factors. In *29th Annual Symposium on Foundations of Computer Science*, pages 412–421, 1988.

[10] J. Gudmundsson and P. Morin. Planar visibility: testing and counting. In *Proceedings of the 26th ACM Symposium on Computational Geometry*, pages 77–86, 2010.

[11] M. Löffler and M. J. van Kreveld. Largest bounding box, smallest diameter, and related problems on imprecise points. *Comput. Geom.*, 43(4):419–433, 2010.

[12] A. Munteanu, C. Sohler, and D. Feldman. Smallest enclosing ball for probabilistic data. In *Proceedings of the 30th ACM Symposium on Computational Geometry*, pages 214–223, 2014.

[13] S. Suri and J. O'Rourke. Worst-case optimal algorithms for constructing visibility polygons with holes. In *Proceedings of the 2nd ACM Symposium on Computational Geometry*, pages 14–23. ACM, 1986.

[14] L. G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

[15] M. J. van Kreveld and M. Löffler. Approximating largest convex hulls for imprecise points. *J. Discrete Algorithms*, 6(4):583–594, 2008.

[16] G. Vegter. The visibility diagram: a data structure for visibility problems and motion planning. In J. R. Gilbert and R. G. Karlsson, editors, *SWAT*, Lecture Notes in Computer Science, pages 97–110. Springer, 1990.