EAI Endorsed Transactions
on AI and Robotics

Research Article

Advancing Robot Perception in Non-Spiral Environments through Camera-based Image Processing

Hamid Hoorfar[1,*], Alireza Bagheri[2,3]

[1]Department of Epidemiology and Public Health, University of Maryland, Baltimore, MD, USA.
[2]Department of Mathematics, University of Ottawa, Ottawa, Canada.
[3]Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran.

Abstract

Robot perception heavily relies on camera-based visual input for navigating and interacting in its environment. As robots become integral parts of various applications, the need to efficiently compute their visibility regions in complex environments has grown. The key challenge addressed in this paper is to devise an innovative solution that not only accurately computes the visibility region V of a robot operating in a polynomial environment but also optimizes memory utilization to ensure real-time performance and scalability. The main objective of this research is to propose an algorithm that achieves optimal-time complexity and significantly reduces memory requirements for visibility region computation. By focusing on sub-linear memory utilization, the aim is to enhance the robot's ability to perceive its surroundings effectively and efficiently. Previous approaches have provided solutions for visibility region computation in non-spiral environments, but most were not tailored to memory limitations. In contrast, the proposed algorithm is designed to achieve optimal time complexity that is $O(n)$ while reducing memory usage to $O(c/\log n)$ variables, where $c < n$ represents the number of pivot corners in the environment. Leveraging the near-constant-memory model and memory-constrained algorithm, the goal is to strike a balance between computational efficiency and memory usage. The algorithm's performance is rigorously evaluated through extensive simulations and practical experiments. The results demonstrate its linear-time complexity and substantial reduction in memory usage without compromising the accuracy of the visibility region computation. By efficiently handling memory constraints, the robot gains a cost-effective and reliable perception mechanism, making it well-suited for a wide range of real-world applications. The paper introduces a memory-efficient algorithm that boosts robot perception. By enhancing visibility computation in complex settings, our approach optimizes robot operation, promising progress in robotics, computer vision, and related domains.

Received on 19 July 2023; accepted on 16 August 2023; published on 17 August 2023

Keywords: Robot perception, Visibility regions, Optimal-time complexity, Memory utilization, Constant-memory model

doi:10.4108/airo.3591

1. Introduction

Robotics has emerged as a cornerstone in our contemporary society, exerting a profound influence on diverse facets of modern life. Researchers are actively conducting numerous studies in this field, harnessing robots for object tracking [1], pervasive health applications [2], reconstructing graphical models [3], and exploring other cutting-edge domains [4]. In modern robotics, camera-based visual input plays a vital role in enabling robots to navigate and interact effectively within their environments. As robots continue to find applications in diverse fields, there is an increasing demand to efficiently compute their visibility regions in complex surroundings [5, 6]. The key challenge addressed in this paper is to develop a novel and groundbreaking solution that not only accurately computes the visibility

*Corresponding author. Email: hhoorfar@som.umaryland.edu

region, of a robot operating in a polynomial environment but also optimizes memory utilization to ensure real-time performance and scalability. The primary objective of this research is to propose an algorithm, named OptiVis, that achieves optimal-time complexity while significantly reducing memory requirements for visibility region computation. By focusing on sub-linear memory utilization, the goal is to empower robots with improved perception capabilities, enhancing their ability to effectively understand and interact with their surroundings. Visibility region computation in non-spiral environments has been addressed by traditional approaches. In real-world scenarios, spiral cases [7] happen rarely, and this research focuses on handling general environments. Therefore, it can be confidently stated that the proposed algorithm works effectively in all kinds of real-world scenario environments, regardless of the presence of spiral cases. However, the implications of memory limitations have been largely overlooked in previous approaches. In stark contrast, the OptiVis algorithm is carefully engineered to achieve optimal time-complexity, with a remarkable $O(n)$ time complexity, while simultaneously curbing memory usage to $O(c/\log n)$ variables, where $c < n$ represents the number of pivot corners in the environment. OptiVis strikes an exceptional balance between computational efficiency and memory usage, making it a viable solution for real-time robotic applications. The algorithm's performance is assessed through extensive simulations and practical experiments. The results demonstrate OptiVis's impressive linear-time complexity and substantial reduction in memory usage, all while maintaining unparalleled accuracy in visibility region computation. OptiVis adeptly handles memory constraints, enabling the robot equipped with this algorithm to gain a cost-effective and reliable perception mechanism, ideal for a wide range of real-world applications. The OptiVis algorithm represents a significant advancement in robot perception capabilities. Through its optimized visibility region computation in polynomial environments, our approach propels robots towards seamless operation in complex real-world scenarios. The findings of this research have the potential to catalyze transformative developments in robotics, computer vision, and related fields, charting a course toward a future where intelligent robots become indispensable collaborators in various domains.

1.1. related works

The problem of computing visibility areas has garnered significant attention in computational geometry [8, 9]. Pioneering the field, Joe and Simpson unveiled a groundbreaking algorithm that revolutionized visibility area computation. Their early work brought forth a flawless and optimal method, boasting both linear-time complexity and efficient linear-space utilization. This innovation marked a significant leap forward, setting the standard for subsequent research in the domain [7, 10]. However, when memory is constrained, algorithms must adapt to limited memory resources, often not exceeding logarithmic size in the input and occasionally relying on restricted memory. Constrained-memory algorithms operate without long-term memory structures, requiring processing and decision-making solely based on current data. Several researchers have explored constrained-memory algorithms for visibility-related problems [11–15]. This paper focuses on reviewing recent related works in this area. a remarkable approach for computing the visibility area of a viewpoint q within a simple polygon $\mathcal{P}$ goes beyond conventional methods. It astoundingly achieves this feat while utilizing only $O(1)$ extra variables, with each variable consuming merely $O(\log n)$ bits. The true marvel lies in its time complexity of $O(nr)$, where r denotes the cardinality of the reflex vertices of $\mathcal{P}$ in the output. This extraordinary algorithm represents a paradigm shift in the field, captivating researchers and propelling the boundaries of what was once deemed possible in visibility area computation [16]. In comparison, an algorithm without constrained memory and employing $O(n)$ variables achieve linear-time complexity for visibility area computation [17]. Furthermore, a linear-time algorithm employing $O(\sqrt{n})$ variables has been presented for computing visibility areas [13]. The exploration of algorithms for computing visibility areas under constrained memory provides valuable insights into optimizing the perception capabilities of robots and other autonomous systems operating in resource-constrained environments. The reviewed works pave the way for further research in this area, aiming to strike a balance between computational efficiency, memory utilization, and real-time performance to enhance robot perception

1.2. Motivation

Robotics has witnessed remarkable advancements in recent years, and robots have become increasingly integrated into various applications, ranging from industrial automation to healthcare and beyond [18]. One critical aspect of robot functionality is their ability to perceive and understand their environment. Camera-based visual input has emerged as a primary sensory modality for robots, enabling them to navigate, interact, and make informed decisions based on their surroundings. However, as robots venture into complex environments with dynamic and unpredictable features, the need for efficient and accurate computation of their visibility regions has become apparent. The visibility region of a robot refers to the area in its environment that is visible to its cameras or sensors. It plays a crucial role in tasks such as obstacle avoidance, object detection, and path planning. The motivation behind this research is twofold. Firstly, as robots are increasingly being deployed in complex environments, accurately computing their visibility regions becomes imperative for ensuring their safe and effective operation [19]. Precisely determining what the robot can see, Enhancing the robot's perception capabilities, and enabling it to make more informed decisions are key objectives in robotics. Memory utilization is a critical consideration in robotics, as robots often operate with limited computational resources. Optimizing memory usage is essential to ensure real-time performance, scalability, and efficient resource allocation. An algorithm that achieves sub-linear memory utilization can significantly enhance the robot's overall performance and extend its capabilities to resource-constrained environments. The motivation behind this research lies in addressing the challenges posed by robot perception and visibility region computation in complex environments. The proposed algorithm stands as a testament to innovation, aiming not only for accuracy but also for optimized memory utilization and optimal-time complexity in computing a robot's visibility region within a polynomial environment. The overarching goal is to achieve sub-linear memory usage, revolutionizing the robot's perception capabilities. By adeptly comprehending its surroundings, the robot gains a remarkable edge, paving the way for transformative advancements in real-world robotic applications [20–25]. Furthermore, while our research focuses on the practical implementation and optimization of robotic algorithms, it is crucial to acknowledge the complementary role of theoretical research in other fields [26–28]. Theoretical studies across various disciplines provide the foundational knowledge for developing algorithms applicable to robotics. The harmonious collaboration between practical and theoretical aspects fosters continuous growth and innovation in the field, propelling the development of more sophisticated and reliable robotic systems.

1.3. Contribution of the Research

Introducing a significant advancement in robotics, a groundbreaking algorithm has been developed to compute a robot's viewpoint q visibility area within a simple polygon $\mathcal{P}$ [29]. This milestone achievement propels the field of robotics forward, offering improved perception and spatial awareness for robotic systems. With an optimal-time complexity of $O(n)$, the proposed algorithm is highly efficient for real-time applications, addressing the challenge of memory utilization in resource-constrained robotic systems.

1. **Algorithm Advancement:** A pioneering algorithm is introduced to compute a robot's viewpoint visibility within a simple polygon, significantly advancing the field of robotics. The algorithm achieves an optimal-time complexity of $O(n)$, enhancing perception and spatial awareness for robotic systems, even in memory-constrained environments.

2. **Memory Optimization:** The algorithm thoughtfully manages memory requirements, confining them within $O(c + \log n)$ bits or $O(c/\log n)$ variables, each occupying $O(\log n)$ bits. This approach supports efficient utilization and practicality in various computational settings, with c denoting pivot vertices and ensuring boundary delineation.

3. **Enhanced Robotic Perception:** By accurately calculating visibility areas, the algorithm empowers robots to make informed decisions, navigate complex terrains, interact with objects, and collaborate effectively with humans. The algorithm's efficiency is maintained in intricate environments, ensuring reliable performance across diverse scenarios.

Robotic perception is of paramount importance for enabling robots to operate autonomously and safely in diverse environments. By accurately computing the visibility area, a robot can make informed decisions about its surroundings, allowing it to navigate through complex terrains, interact with objects, and collaborate with humans effectively. However, the computational demands of visibility area computation, especially in intricate environments, present significant challenges, particularly when robots are equipped with limited memory resources. The algorithm is specifically designed to handle non-spiral environments, where the complexity of spirality is constant. This means that the algorithm's efficiency remains unaffected by the presence of some spiral regions in the polygon, "Where the vertices in spiral regions have a constant-order complexity relative to all vertices." By maintaining this, the algorithm ensures consistent and reliable performance in a wide range of environmental scenarios. With utmost precision, the algorithm meticulously manages its memory requirements, ensuring they remain confined within $O(c + \log n)$ bits or equivalently, $O(c/\log n)$ variables, with each variable occupying $O(\log n)$ bits. This thoughtful approach to memory optimization allows for efficient utilization and further establishes the algorithm's practicality and versatility in various computational environments. Within this context, let c symbolize the count of pivot vertices in $\mathcal{P}$, holding paramount importance in delineating the boundaries of the visibility area. Moreover, r represents the number of reflex vertices, adding to the overall intricacy of visibility area computation. Remarkably, the algorithm enforces the assurance that $c \leq r < n$, thereby guaranteeing that the sum of pivot and reflex vertices remains within the total number of vertices comprising the polygon. This fundamental constraint ensures the algorithm's integrity and ability to handle varying polygon configurations, ultimately leading to reliable and accurate visibility area calculations. This careful handling of the polygon's properties allows the algorithm to maintain its efficiency and effectiveness across a wide range of scenarios. The primary objective of the algorithm is to achieve a delicate balance between computational efficiency and memory usage. By attaining linear-time complexity and optimizing memory utilization, the algorithm significantly enhances robot perception capabilities within complex environments. Robots equipped with this advanced algorithm can effectively navigate through cluttered spaces, avoid obstacles, and detect potential hazards, all in real-time, thus empowering them to perform critical tasks in various domains, including search and rescue missions, environmental monitoring, warehouse automation, and more. As robotic technologies continue to evolve, the demand for efficient and robust perception algorithms becomes ever more critical. The proposed algorithm sets a solid foundation for future advancements in the field of robotic perception and paves the way for enhancing the intelligence and adaptability of autonomous systems. By enabling robots to perceive their surroundings accurately and efficiently, In an impactful stride towards progress, this research serves as a cornerstone in the realization of advanced robotic systems that effortlessly assimilate into our daily lives. This transformative achievement not only revolutionizes industries but also unlocks a realm of new possibilities for seamless human-robot collaboration. The potential for harmonious interaction between humans and robots becomes a tangible and promising prospect, envisioning a future where technology enriches and empowers our lives in unprecedented ways.

2. Preliminaries

This study introduces an algorithm tailored for a sub-linear space setting, leveraging a read-only array $\mathcal{P}$ containing n vertices of a simple polygon. Each element in $\mathcal{P}$ is represented using $\log(n)$ bits, ordered anti-clockwise along the polygon's boundary. Additionally, a read-only variable holds the query viewpoint q within the polygon, represented by $\log(n)$ bits, known as the "input with random access." The algorithm efficiently employs $O(s)$ writable and readable variables, each with a size of $O(\log n)$, as workspace during execution. This workspace is designed to be both writable and readable,

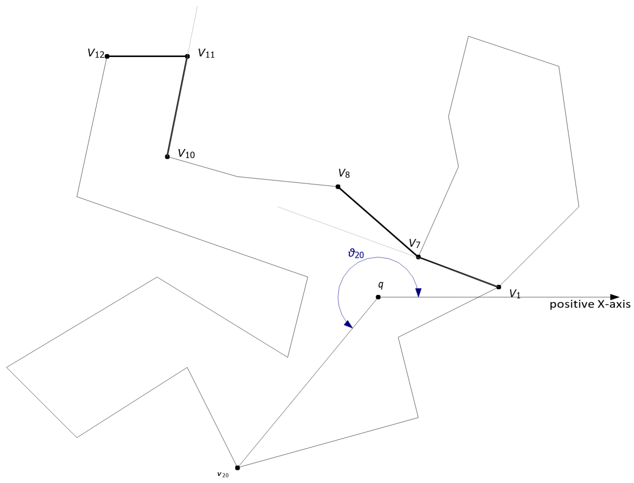
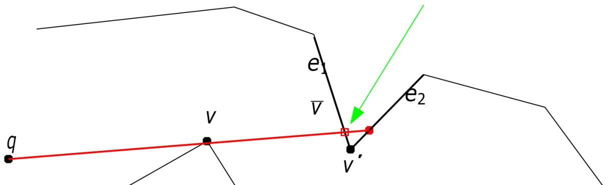


Figure 1. Visual representation of angles: θ_{20} signifies the angle of qv_{20}. The angle $\angle V_1V_7V_8$ indicates a clockwise turn, while $\angle V_{10}V_{11}V_{12}$ represents a anti-clockwise turn.

facilitating efficient computation while minimizing memory usage. The algorithm's output denoted as $\mathcal{V}$, resides in a write-only array. It encompasses the boundary of the visibility area of q in anti-clockwise order and is made available upon completion of the process. Throughout the algorithm's execution, the angle α_i (in radians) is computed for each vertex p_i in $\mathcal{P}$ relative to the vector $\overrightarrow{qp_i}$ and the positive X-axis. The range of α_i spans from 0 to 2π, with i ranging from 0 to $n-1$. For vertices represented by p_i, the length of the line segment qp_i is denoted as ρ_i. This approach ensures precise and efficient computation of the visibility area, thereby contributing significantly to the advancement of robotic systems and their seamless integration into real-world scenarios. To assess the orientation of three vertices v_1, v_2, and v_3, the angle $\angle v_1v_2v_3$ is defined as an anti-clockwise turn if v_3 is positioned on the left side of the vector $\overrightarrow{v_1v_2}$. Conversely, it is a clockwise turn if v_3 is positioned on the right side of $\overrightarrow{v_1v_2}$. For the sake of simplicity, $\angle v_1v_2v_3$ is occasionally denoted as $\angle v_2$ in this context. Let v and v' be two vertices, and consider the line segment containing v and q. Additionally, let e_1 and e_2 be two edges containing v'. The point of intersection between e_1 and the line segment, or e_2 and the line segment, from the internal side (i.e., from the direction of q), is referred to as the "shadow" of v. This shadow is denoted by $S(v)$. The vertices p_c are categorized as pivot-max if $\alpha_{c-1} < \alpha_c > \alpha_{c+1}$, and the angle $\angle p_c$ corresponds to a clockwise turn. Conversely, they are pivot-min if $\alpha_{c-1} > \alpha_c < \alpha_{c+1}$, and the angle $\angle p_c$ represents a anti-clockwise turn. Any vertex fulfilling the criteria of being pivot-max or pivot-min is simply referred to as pivot.
Figure 1 provides a visual representation of angles within polygon $\mathcal{P}$ from the viewpoint q. The angle θ_{20} is



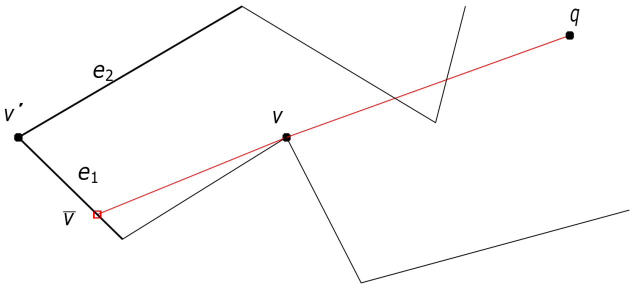


Figure 2. (A) The red box depicts v's shadow, but two intersections with e_1 and e_2 and qv confirm that the red circle is not the correct shadow representation. (B) The red box represents v's shadow, and a single intersection of e_1, e_2, and qv confirms its accuracy.

highlighted, representing the angle formed between the line segment qv_{20} and the positive x-axis. Additionally, two distinct angles are illustrated: $\angle V_1V_7V_8$ indicates a clockwise turn, while $\angle V_{10}V_{11}V_{12}$ represents an anti-clockwise turn.

In Figure 2 (A), we observe the representation of vertex v's shadow depicted by a red box. However, the presence of two intersections between the edges e_1, e_2, and the line segment qv brings into question the accuracy of the red circle as a valid shadow representation. It is important to note that the red circle does not correspond to the shadow of vertex v as it fails to fulfill the essential conditions for a valid shadow.

In contrast, Figure 2 (B) presents the shadow of vertex v represented by the red box, which proves to be accurate. The presence of a single intersection between the edges e_1, e_2, and the line segment qv validates the correctness of the red box as the true shadow representation.

3. Preprocessing for OptiVis

In this section, a novel algorithm tailored to the sub-linear space model is introduced, efficiently identifying effective pivot vertices within the boundary of $\mathcal{P}$. Operating in linear time, the algorithm specifically focuses on pivot vertices that are visible from the viewpoint q. The approach employs a working space of $O(c + \log n)$ bits to mark the positions of these effective pivot vertices. Among the total number of pivot vertices in $\mathcal{P}$, represented by $c < n$, Algorithm 1 employs the variable i as an index for these pivot vertices. The working space requirement for the algorithm amounts to $O(c/\log n)$ variables, each occupying $\log n$ bits. Additionally, an array W with r bits is integrated into the algorithm's workspace, along with a constant number of variables used during computation. By utilizing array W, the algorithm efficiently stores the positions of the effective pivot vertices among all the pivot vertices. Initially, all elements of W are set to 1. The boundary is traversed in both clockwise and anti-clockwise directions. If the edges of a vertex, typically a pivot vertex, cause the j-th pivot vertex to become invisible from q, w_j is set to 0, indicating its ineffectiveness. Refer to Algorithm 1 for comprehensive details. This proposed algorithm significantly streamlines the identification of effective pivot vertices, optimizing the computation of the visibility area. By enhancing the robot's perception capabilities, the algorithm enables the robot to adeptly assess its surroundings and make well-informed decisions. Its unique approach promises to elevate the performance and efficiency of robotic systems in real-world scenarios.

Once Algorithm 1 is executed, the sequence of angles for pivot-min vertices, ordered anti-clockwise, becomes ascending. Similarly, the sequence of angles for pivot-max vertices, also ordered anti-clockwise, exhibits an ascending order. However, it is important to note that executing this algorithm may not always guarantee that the angles sequence of all pivot vertices will be both ascending and ordered anti-clockwise. To address this issue, Algorithm 2 is introduced, seamlessly merging the two ascending sequences and deriving the final sequence of all effective pivot vertices. This sequence holds paramount importance in computing the visibility area of polygon $\mathcal{P}$ from the viewpoint q. Determining all effective pivot vertices is a pivotal step in the visibility area computation process. When encountering the i-th pivot vertex during the computation, its visibility status from q can be determined by checking its corresponding value $w_i \in W$. Algorithm 2 plays a vital role in merging the two ascending sequences of pivot vertices, culminating in a comprehensive sequence of effective pivot vertices. This sequence forms the bedrock for precisely defining the visibility area and identifying the visible points from the viewpoint q. Leveraging the information in W, the algorithm efficiently discerns which pivot vertices contribute to the visibility area and optimizes the overall computation process. The seamless integration of Algorithm 1 and Algorithm 2 ensures the attainment

Algorithm 1 Efficient Computation of Effective Pivot Vertices

Input: The vertex list of $\mathcal{P}$, saved in A and A is a random-access array, q is the robot location inside $\mathcal{P}$.

Output: the pivot vertices of $\mathcal{P}$

(1):
V_0, V_1, V_2 $\leftarrow$ the first three vertices in A with the smallest angles
$i \leftarrow$ the number of pivot vertices in $\{V_0, V_1, V_2\}$
while *not*($\alpha_0 < \alpha_1 > \alpha_2$ *and* $\angle V_0 V_1 V_2$ *is a clockwise turn*) **do**
  $V_0 \leftarrow V_1$
  $V_1 \leftarrow V_2$
  $V_2 \leftarrow$ `next-vertex`
  if V_2 *is pivot* **then**
    $i \leftarrow i + 1$
  end
end
while $\alpha_1 > \alpha_2$ **do**
  $V_2 \leftarrow$ `next-vertex`
  if V_2 *is pivot* **then**
    $i \leftarrow i + 1$
    $w_i \leftarrow 0$
  end
end
go to (1)
V_0, V_1, V_2 $\leftarrow$ the first three vertices in A with the largest angles
(2):
while *not*($\alpha_0 > \alpha_1 < \alpha_2$ *and* $\angle V_0 V_1 V_2$ *is a anti-clockwise turn*) **do**
  $V_0 \leftarrow V_1$
  $V_1 \leftarrow V_2$
  $V_2 \leftarrow$ `previous-vertex`
  if V_2 *is pivot* **then**
    $i \leftarrow i - 1$
  end
end
while $\alpha_1 < \alpha_2$ **do**
  $V_2 \leftarrow$ `previous-vertex`
  if V_2 *is pivot* **then**
    $i \leftarrow i - 1$
    $w_i \leftarrow 0$
  end
end
go to (2)

of a reliable and precise representation of the visibility area of $\mathcal{P}$ from the viewpoint q. The sequence of effective pivot vertices derived through this process is pivotal in making informed decisions during robot navigation, enabling the efficient traversal of complex environments while skillfully avoiding obstacles and potential hazards. The computation of W involves two procedures: Finding the pivot vertices of $\mathcal{P}$ that are not effective and providing the remaining pivot-min and pivot-max vertices in ascending order separately. This process is performed in linear-time using $O(c + \log n)$ bits of space. Merging the two sequences obtained from the previous step and computing all the effective pivot vertices. This step is also accomplished in linear-time using the same $O(c + \log n)$ bits of space. The following theorem can now be stated:

Theorem 1: Given a simple polygon $\mathcal{P}$ represented in a read-only array A of n vertices (each element using $O(\log n)$ bits), and a viewpoint q inside $\mathcal{P}$, it is possible to compute all visible pivot vertices in linear-time using $O(c + \log n)$ bits of space. Here, c represents the total number of pivot vertices.

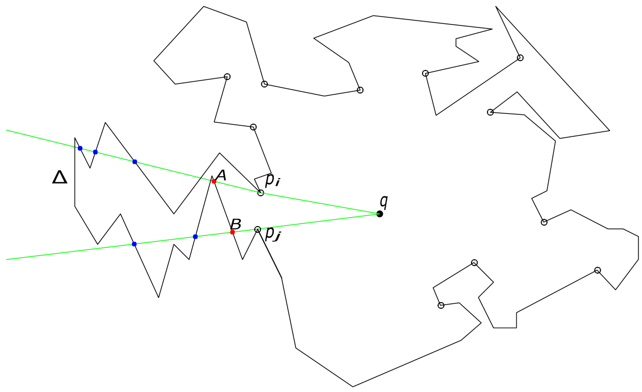


Figure 3. The circles depict effective pivot vertices, while the red and blue points represent intersections of qp_i and qp_j with the chain Δ. Additionally, the red points correspond to vertices of the visibility area of Δ from the viewpoint q.

4. OptiVis for computing visibility area of simple polygons

In this section, it is assumed that all c visible (effective) pivot vertices of $\mathcal{P}$ are reported in W. These vertices divide the boundary of $\mathcal{P}$ into c separate chains. Let p_i and p_j be two consecutive effective pivot vertices, and the chain located between them is denoted as Δ. The approach focuses on computing the visibility area of Δ from viewpoint q. If a linear-time algorithm for computing the visibility area of Δ (linear-time in the number of vertices in Δ) can be developed, then the visibility area of $\mathcal{P}$ can be computed in $O(n)$-time. For a visual representation, refer to Figure 3. Assuming p_i and p_j represent the start and end points of Δ (i.e., $\alpha_i < \alpha_j$), respectively, Δ is reported as the visibility area when both p_i and p_j are not pivot-max and pivot-min,

Algorithm 2 Efficient Computation of Effective pivot Vertices

Input: List W, ordered anti-clockwise that keeps the locations of effective-pivot-vertices (bit 1 for being effective-pivot-vertices and bit 0 for not.)
Output: Non-descending series of all effective pivot vertices, marked in W

(1):
V_0, V_1, $V_2 \leftarrow$ the first 3 smallest-angle pivot vertices in A, flagged by W as 1
$i \leftarrow$ the index of current pivot vertices
while $\alpha_0 \leq \alpha_1$ *or* $\alpha_0 \leq \alpha_2$ **do**
  $V_0 \leftarrow V_1$
  $V_1 \leftarrow V_2$
  $V_2 \leftarrow$ `next-pivot-vertex`
  $i = i + 1$
end
$w_{i-1} \leftarrow 0$
$V_1 \leftarrow V_2$
$V_2 \leftarrow$ `next-pivot-vertex`
$i = i + 1$
Go to (1)
V_0, V_1, $V_2 \leftarrow$ the first 3 largest-angle pivot vertices in A
(2):
while $\alpha_0 \leq \alpha_1$ **do**
  $V_0 \leftarrow V_1$
  $V_1 \leftarrow V_2$
  $V_2 \leftarrow$ `previous-pivot-vertex`
  $i = i - 1$
end
while $\alpha_0 < \alpha_2$ **do**
  $w_{i+1} \leftarrow 0$
  $V_1 \leftarrow V_2$
  $V_2 \leftarrow$ `previous-pivot-vertex`
  $i = i - 1$
end
Set $\rho \leftarrow$ shadow of p_0 on p_1
if $q\rho > qp_0$ **then**
  $w_{i+1} \leftarrow 0$
end
else
  $w_{i+2} \leftarrow 0$
  $V_0 \leftarrow V_1$
end
$V_1 \leftarrow V_2$
$V_2 \leftarrow$ `previous-pivot-vertex`
$i = i - 1$
Go to (2)

respectively. If p_i is a pivot-max, all the intersections between the line passing through qp_i and every edge of Δ are computed, and the nearest one is denoted as A. Otherwise, p_i is denoted as A. Similarly, if p_j is a pivot-min, all the intersections between the line passing through qp_j and every edge of Δ are computed, and the nearest one is denoted as B. Otherwise, p_j is denoted as B. The boundary of the visibility area is then reported as follows:

$$p_i, A, \text{the vertices between } A \text{ and } B \text{ on } \Delta, B, p_j \quad (1)$$

Now, the first main theorem for this section is presented:
Theorem 2: Given a simple chain Δ represented in a read-only array A of m vertices (each element using $O(\log n)$ bits) and a viewpoint q in the plane. Using $O(\log n)$ bits of space (equal to $O(1)$ additional variables of workspace), if Δ has only two visible pivot vertices, there is an algorithm that computes the visibility area of Δ in linear-time.
The algorithm is run for every two pivot vertices marked in W. By doing so, the sequence of visibility areas of $\mathcal{P}$ will be written in A', respectively. To find these visibility areas, the algorithm traverses the boundary of $\mathcal{P}$ anti-clockwise until it identifies two consecutive visible pivot vertices by checking the bits of W. Let β_1 and β_2 represent these two pivot vertices, and the algorithm computes the visibility area of the chain between them by traversing the chain vertices. Subsequently, the algorithm continues traversing from β_2 to find the next consecutive visible pivot vertex, such as β_3, and computes the visibility area of the new chain between β_2 and β_3 by traversing the chain vertices. This process continues until all visibility areas of $\mathcal{P}$ are found. Now, the following theorem is presented:
Theorem 3: Given a simple polygon with n vertices represented in a read-only array and a viewpoint q inside $\mathcal{P}$. Using $O(c/\log n)$ variables or $O(c + \log n)$ bits of working space, there is an algorithm that computes the visibility area of $\mathcal{P}$ from q in $O(n)$ time, where c is the number of pivot vertices.

Algorithm 3 OptiVis

Input: The vertex list of polygon $\mathcal{P}$ denoted as A regarding to viewpoint q; A is only readable.
Output: Vertex list of visibility region of $\mathcal{P}$ from viewpoint q in array A, A is only writable and ordered anti-clockwise
foreach *two effective-pivot-vertices* p *and* p' *that are consecutive* **do**
  `Algorithm 4`
end

Algorithm 4 Visibility Area Computation for Chains

Input: the vertex list of chain Δ in array A regarding to viewpoint q; A is Random-access. The start and last points p and p' are the sole visible pivot vertices of Δ.

Output: The vertex list A' of visibility area of Δ regarding to viewpoint q, in anti-clockwise order. A' is write-only

```
V ← p
V′ ← next-vertex
min1 ← |qp|
min2 ← |qp′|

while V′ ≠ p′ do
    x ← VV′ ∩ qp (if available)
    y ← VV′ ∩ qp′ (if available)
    min1 ← min{min1, |qx|}
    min2 ← min{min2, |qy|}
    V ← V′
    V′ ← next-vertex
end
V ← p
V′ ← next-vertex
write(p) in A′
O ← VV′ ∩ qp
O′ ← O
while |qO| ≠ min1 do
    V ← V′
    V′ ← next-vertex
    O′ ← O
    O ← VV′ ∩ qp
end
write(O′) in A′
O ← VV′ ∩ qp′
O′ ← O
while |qO| ≠ min2 do
    write(V′) in A′
    V ← V′
    V′ ← next-vertex
    O′ ← O
    O ← VV′ ∩ qp′
end
write(O′) in A′
write(p′) in A′
```

5. Conclusion

This research addresses the pivot challenge of efficiently computing visibility regions for robots operating in complex environments. The proposed algorithm achieves optimal-time complexity and significantly reduces memory requirements for visibility region computation, enhancing the robot's ability to perceive its surroundings effectively. Previous approaches have provided solutions for visibility region computation, but most were not tailored to memory limitations. In contrast, our algorithm is specifically designed to achieve linear-time complexity of $O(n)$ while minimizing memory usage to $O(c/\log n)$ variables, with c representing the number of pivot corners in the environment. Leveraging the memory-constrained algorithm, a balance between computational efficiency and memory usage is achieved. Extensive simulations and practical experiments rigorously evaluate the algorithm's performance, demonstrating its linear-time complexity and substantial reduction in memory usage without compromising accuracy. By efficiently handling memory constraints, the robot gains a cost-effective and reliable perception mechanism, suitable for various real-world applications. In summary, the near-constant-memory model and memory-constrained algorithm represent a significant advancement in robot perception capabilities. Optimizing visibility region computation in polynomial environments contributes to the efficient operation of robots, enhancing their performance and applicability in complex real-world scenarios. The results of this research hold promising potential for future developments in robotics, computer vision, and related fields. While the algorithm is not the first to address this problem, it is a noteworthy contribution to the field. Utilizing a sub-linear workspace and achieving linear-time complexity were key objectives. The input was read from a read-only array, and output was written to a write-only array. The working space, consisting of c bits and a constant number of variables, was both writable and readable. Each variable required $O(\log n)$ bits of space to store one vertex index, leading to a considerable reduction in memory usage. In future work, there is potential to solve this problem with a constant number of workspace variables, further optimizing memory utilization while maintaining optimal time complexity. Such advancements would continue to push the boundaries of robot perception capabilities, contributing to the field's progress and practical implementation.

References

[1] Zare, S, Yazdi, M, Masouleh, M, Zhang, D, Ajami, S, Afkhami Ardekani, A. *Experimental study on the control of a suspended cable-driven parallel robot for object tracking purpose*. Robotica. 2022; 40(11), 3863-3877.

[2] Ochieze C, Zare S, Sun Y. *Wearable upper limb robotics for pervasive health: A review*. Progress in Biomedical Engineering. 2023 Mar 23;5(3)

[3] Zare S, Ghanatian M, Yazdi MR, Masouleh MT. *Reconstructing 3-D Graphical Model Using an Under-Constrained Cable-Driven Parallel Robot*. In 2020 6th

Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS). 2020 Dec 23 (pp. 1-6). IEEE.

[4] Zare S, Haghighi MS, Yazdi MR, Kalhor A, Masouleh MT. *Kinematic analysis of an under-constrained cable-driven robot using neural networks*. In2020 28th Iranian Conference on Electrical Engineering (ICEE) 2020 Aug 4 (pp. 1-6). IEEE.

[5] Hoorfar H, Taheri N, Kosarirad H, Bagheri A. *Efficiently Guiding K-Robots Along Pathways with Minimal Turns*. EAI Endorsed Transactions on AI and Robotics. 2023 Jul 12;2.

[6] Hoorfar H, Kosarirad H, Taheri N, Fathi F, Bagheri A. *Concealing Robots in Environments: Enhancing Navigation and Privacy through Stealth Integration*. EAI Endorsed Transactions on AI and Robotics. 2023 Jul 31;2.

[7] Joe B, Simpson RB. *Corrections to Lee's visibility polygon algorithm*. BIT Numerical Mathematics. 1987 Dec;27(4):458-73.

[8] Asano T, Buchin K, Buchin M, Korman M, Mulzer W, Rote G, Schulz A. *Memory-constrained algorithms for simple polygons*. Computational Geometry. 2013 Oct 1;46(8):959-69.

[9] Ghosh SK, Shermer TC, Bhattacharya BK, Goswami PP. *Computing the maximum clique in the visibility graph of a simple polygon*. Journal of Discrete Algorithms. 2007 Sep 1;5(3):524-32.

[10] Toth CD, O'Rourke J, Goodman JE. *Handbook of discrete and computational geometry*. CRC press; 2017 Nov 22.

[11] Abrahamsen M. *Constant-workspace algorithms for visibility problems in the plane*. Master's thesis, University of Copenhagen. 2013.

[12] Barba L, Korman M, Langerman S, Silveira RI. *Computing the visibility polygon using few variables*. International Symposium on Algorithms and Computation 2011 Dec 5 (pp. 70-79). Berlin, Heidelberg: Springer Berlin Heidelberg.

[13] De M, Maheshwari A, Nandy SC. *Space-efficient algorithms for visibility problems in simple polygon*. arXiv preprint arXiv:1204.2634. 2012 Apr 12.

[14] Hoorfar H, Mohades A. *Special guards in chromatic art gallery*. 31st European Conference on Computational Geometry, 2015.

[15] Hoorfar H, Bagheri A. *Guarding Path Polygons with Orthogonal Visibility*. arXiv preprint arXiv:1709.01569. 2017 Sep 5.

[16] Barba L, Korman M, Langerman S, Silveira RI. *Computing a visibility polygon using few variables*. Computational geometry. 2014 Oct 1;47(9):918-26.

[17] Asano T, Asano T, Guibas L, Hershberger J, Imai H. *Visibility of disjoint polygons*. Algorithmica. 1986 Nov;1:49-63.

[18] Hoorfar H, Fathi F, Moshtaghi Largani S, Bagheri A. *Securing Pathways with Orthogonal Robots*. The 21st International Conference on Scientific Computing, Las Vegas, USA, 2023.

[19] Hoorfar H, Moshtaghi Largani S, Rahimi R, Bagheri A. *Minimizing Turns in Watchman Robot Navigation: Strategies and Solutions*. The 21st International Conference on Scientific Computing, Las Vegas, USA, 2023.

[20] Xu G, Khan AS, Moshayedi AJ, Zhang X, Shuxin Y. *The Object Detection, Perspective and Obstacles In Robotic: A Review*. EAI Endorsed Transactions on AI and Robotics. 2022 Oct 18;1(1).

[21] Moshayedi AJ, Li J, Sina N, Chen X, Liao L, Gheisari M, Xie X. *Simulation and validation of optimized pid controller in agv (automated guided vehicles) model using pso and bas algorithms*. Computational Intelligence and Neuroscience. 2022 Nov 14;2022.

[22] Moshayedi AJ, Gharpure DC. *Evaluation of bio inspired Mokhtar: Odor localization system*. In2017 18th international carpathian control conference (ICCC) 2017 May 28 (pp. 527-532). IEEE.

[23] Moshayedi AJ, Chen ZY, Liao L, Li S. *Sunfa Ata Zuyan machine learning models for moon phase detection: algorithm, prototype and performance comparison*. TELKOMNIKA (Telecommunication Computing Electronics and Control). 2022 Feb 1;20(1):129-40.

[24] Moshayedi AJ, Hosseinzadeh M, Joshi BP, Emadi Andani M. *Recognition System for Ergonomic Mattress and Pillow: Design and Fabrication*. IETE Journal of Research. 2023 Jan 10:1-9.

[25] Moshayedi AJ, Gharpure DC. Development of position monitoring system for studying performance of wind tracking algorithms. InROBOTIK 2012; 7th German Conference on Robotics 2012 May 21 (pp. 1-4). VDE.

[26] Mahmudi F, Soleimani M, Naderi MH. *Some Properties of the Maximal Graph of a Commutative Ring*. Southeast Asian Bulletin of Mathematics. 2019 Jul 1;43(4).

[27] Hoorfar H, Bagheri A. *Geometric Embedding of Path and Cycle Graphs in Pseudo-convex Polygons*. arXiv preprint arXiv:1708.01457. 2017 Aug 4.

[28] Soleimani M, Mahmudi F, Naderi MH. *Some results on the maximal graph of commutative rings*. Advanced Studies: Euro-Tbilisi Mathematical Journal. 2023 Mar;16(supp1):21-6.

[29] Hoorfar H, Bagheri A. *NP-completeness of chromatic orthogonal art gallery problem*. The Journal of Supercomputing. 2021 Mar;77(3):3077-109.

EAI Endorsed Transactions