



Complexity Bounds for Ray Tracing and Illumination in 2D Using Spy Mirrors

Rosemary Adejoh, Andreas Jakoby, Sneha Mohanty and
Christian Schindelhauer

EasyChair preprints are intended for rapid
dissemination of research results and are
integrated with the rest of EasyChair.

February 13, 2024

COMPLEXITY BOUNDS FOR RAY TRACING AND ILLUMINATION IN 2D USING SPY MIRRORS

TECHNICAL REPORT

• **Rosemary U. Adejoh**

Faculty of Media
Bauhaus University
Weimar, Germany

rosemary.utenwojo.adejoh@uni-weimar.de

• **Andreas Jakoby**

Faculty of Media
Bauhaus University
Weimar, Germany

andreas.jakoby@uni-weimar.de

• **Sneha Mohanty**

Computer Networks and Telematics
University of Freiburg
Freiburg, Germany

mohanty@informatik.uni-freiburg.de

• **Christian Schindelhauer**

Computer Networks and Telematics
University of Freiburg
Freiburg, Germany

schindel@informatik.uni-freiburg.de

February 12, 2024

ABSTRACT

Since the seminal work of Reif et al. from 1994, it is known that Ray Tracing in three dimensions is as hard as solving the Halting problem. However, the question of whether this result also holds for two dimensions remains open. In this paper, we address this problem and demonstrate that for two dimensions, Ray Tracing is \mathcal{NP} -complete when the number of reflections is linearly bounded, and when linear one-way mirrors and mirrors are allowed.

To achieve this, we examine various sub-classes of one-way mirrors. Our hardness result holds when we allow natural one-way mirrors that are half-transparent from both sides or when we permit only merging and splitting one-way mirrors. Merging mirrors possess one side that is totally reflective and another side that is perfectly transparent, while split mirrors have one side that is half-transparent and the other side that is fully transparent.

In the absence of half-mirrors, we establish a polynomial time bounded algorithm for polynomial reflections in any k -dimensional space, provided the mirrors are represented as k -simplexes with rational coordinates. When given merging mirrors and mirrors represented as line segments or parabolic shapes, we show that Ray Tracing becomes \mathcal{P} -hard in two dimensions.

Moreover, we present the first computational complexity results for the Illumination problem. Given a light source, we inquire whether a specific point is illuminated. We prove that this problem is \mathcal{NP} -hard for two dimensions, considering mirrors represented as line segments. On the other hand, for the upper bound, we demonstrate that Illumination in k -dimensions, with mirrors represented as k -simplexes with rational point coordinates and linearly bounded reflections, can be solved in \mathcal{NP} .

Keywords Illumination, Ray Tracing, Computational Complexity

1 Introduction and Motivation

We are studying the computational complexity of the Illumination and Ray Tracing Problem in various settings and contribute new insights after this field has not seen any progress for over a quarter of a century with the seminal works of Reif et al. (23) and Aronov et al. (4). For both problems we assume that light follows a straight line and is reflected according to the law of reflections on mirrored surfaces that are placed in k -dimensional space. For the shape of these mirrors we consider planar objects, i.e. line segments in 2D and polygons embedded in a plane for three dimensions

(as well as a generalization for higher dimensions). For some results we also allow mirrors shaped as parabolas as in (23). While Penrose et al. (22) introduced illumination for enclosed objects, we do not restrict mirrors to enclosing the light source.

As a major contribution, we have extended the techniques employed by Reif et al. (23) in 3D even to the 2D case. We show that one-way mirrors increase the complexity and in order to understand this in more detail, we distinguish their behavior of splitting and merging beams. So, on one side of a splitting one-way mirror, light rays pass through and produce a reflection, but on the opposite side, it is a perfect mirror. A merging mirror is fully transparent on one side and on the other side fully reflective, while a normal one-way-mirror is semi-transparent for both sides and produces reflections for both directions, see Figure 1¹.

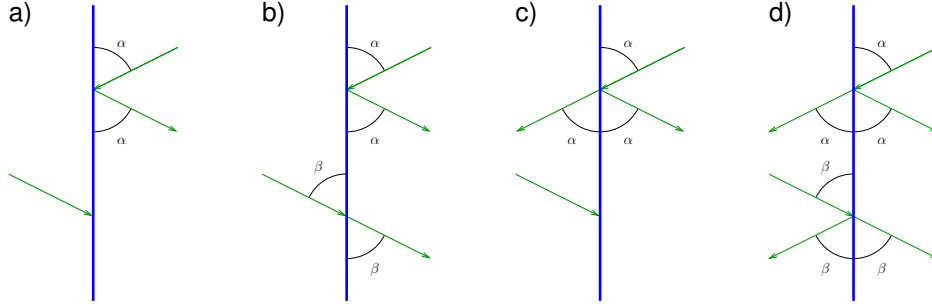


Figure 1: a) full, b) one-way merging, c) one-way splitting, and d) one-way mirror.

A well-studied problem in this area is the *Ray Tracing* problem which we define as: given a source point and the direction of a ray, one has to determine whether the ray hits a given target point for a given set of obstacles. While in other works these obstacles could be fully absorbing or diffuse reflectors, here we consider only mirrors and the above mentioned one-way mirrors. We also consider the *Illumination* problem as the question, whether for a given set of mirrors from a given source point, a ray exists that reaches a given target point. We may limit the number t of reflections the rays needs to reach the target for both problems and call this variant t -bounded Ray Tracing or t -bounded Illumination. Furthermore, we assume that the coordinates of all points are given as rational numbers, for which the binary representation is part of the input.

Contributions In this work we present the following results, where if not otherwise stated plane mirrors and unlimited number of reflections are assumed.

1. Polynomially bounded Ray Tracing is in \mathcal{P} , i.e. it can be computed in polynomial time by a deterministic Turing machine for full plane mirrors as well as one-way merging mirrors in k dimensions if k and the number of reflections is bound by a polynomial.

This observation uses the well known equations for reflections and shows that the exact calculation of the rational number representation of the reflected beams grows only linearly with the number of reflections.

2. Polynomially bounded Illumination and Ray Tracing is in \mathcal{NP} for polynomial dimensions in the context of full plane, one-way merging, one-way splitting as well as one-way mirrors for polynomial number of reflections.

This improves the result of Aronov et al. (4), who proves that there is an exponential time bounded algorithm which can resolve this question.

3. Polynomially bounded Ray Tracing with one-way mirrors is \mathcal{NP} -hard in two dimensions.

Reif et al. (23) showed that Ray Tracing is as hard as the Halting problem for unbounded reflections only for three dimensions. Our upper bound also holds for higher dimensions. We show our results for a construction involving one-way mirror or for one-way splitting and one-way merging mirrors in linear number of reflections.

4. Ray Tracing using full plane mirrors, parabolic mirrors or one-way merging mirrors is \mathcal{P} -hard for two dimensions for a exponential number of reflections. This shows that the techniques of Reif et al. (23) can be used in two dimensions, but only to a limited extent. By using parabola mirrors, we also implement a stack according to the position of the ray.

¹We remind the reader that a one-way mirror is the same as a semi-transparent mirror and the degree of reflectiveness is simply a function of the lighting where they are used.

| | Problem | Complexity Class | Dimensions | Type of Mirrors | Number of Reflections |
|-----------|--------------|----------------------|------------|-------------------|-----------------------|
| Theorem 1 | Ray Tracing | \mathcal{NP} | lin | a, b | pol |
| Theorem 1 | Ray Tracing | \mathcal{NP} | lin | a, b, c, d | pol |
| Theorem 4 | Ray Tracing | \mathcal{P} -Hard | 2 | a + b + parabolic | exp |
| Theorem 3 | Ray Tracing | \mathcal{NP} -hard | 2 | b + c | lin |
| Theorem 3 | Ray Tracing | \mathcal{NP} -hard | 2 | d | lin |
| Theorem 2 | Illumination | \mathcal{NP} | lin | a, b, c, d | pol |
| Theorem 5 | Illumination | \mathcal{NP} -Hard | 2 | a | exp |
| Theorem 6 | Illumination | \mathcal{NP} -Hard | 2 | a + b | lin |

Table 1: Summary of our contributions. The symbols '+' refer to 'and' and ',' for 'or'. The types of mirrors are depicted in Figure 1

5. Illumination is \mathcal{NP} -hard for two dimensions.

Aronov et al. (4) have proved that their version of the Illumination problem has an intrinsic exponential complexity for a linear number of reflection. There, they compute all illuminated regions with corresponding reflection paths. Here, we show with a different construction involving full plane mirrors that the question, whether a given point is illuminated is as hard as solving the subset-sum problem.

6. The Linear bounded Illumination problem using full plane mirrors and one-way merging mirrors is \mathcal{NP} -hard for two dimensions and linear number of reflections.

To our knowledge we give the first lower computational complexity bounds for the Illumination problem and for the Ray Tracing problem no lower bounds for two dimensions were known so far, although this field is a vivid area in theory and practice. In Table 1, we summarize all of our contributions.

2 Related Work

For the illumination problem Penrose et al. (22) proved that there exists a closed room with curved (elliptic and straight) surface, where a light source cannot illuminate all points within. For polygonal rooms Tokarsky (27) proved that there is a 27 sided polygonal where some internal points cannot be illuminated by a light source. A smaller construction of a 24 sided polygon was later provided by Castro (10). There is a qualitative difference between polygonal rooms and curved rooms, since Lelièvre et al. (18) proved that all inner points of a polygon except a finite set of points are illuminated.

The analysis of the complexity of the illumination problem has been studied as a special form of a *visibility problem*, see the chapter edited by O'Rourke in (28), the most prominent of which is the *art gallery problem*, which has been proven to be $\exists\mathbb{R}$ -complete (2). Researchers have replaced the walls with reflective mirrors and investigated the question how complex it is to calculate the illuminated area. For a single reflection Aronov et al. (3) showed that this region has a combinatorial complexity of $\Theta(n^2)$, and can be computed by a $\mathcal{O}(n^2 \log^2 n)$ time bounded algorithm. For k reflections the combinatorial complexity grows exponential with $\Omega((n/k) - \mathcal{O}(1))^{2k}$ for an n -sided polygon, i.e. there is a construction, where so many different regions with different reflection paths from the light source towards a given point exist. They also provide an $\mathcal{O}(n^{2k} \log n)$ -time bounded algorithm to compute all these paths in space $\mathcal{O}(n^{2k})$.

If the mirrors are replace with k diffuse mirrors, where the n sides of a polygon scatter the incoming light in all directions, Aronov et al. (5) prove that the complexity of the visibility problem is bounded by $\mathcal{O}(n^9)$. Such diffuse reflection paths can be computed in time $\mathcal{O}(n^2)$ with holes and in time $\mathcal{O}(n + k \log n)$ for k reflections (13). This bound can be improved if one considers constrained diffuse reflection paths (8). The calculation of the minimum-link diffuse reflection path is \mathcal{NP} -hard even for two-dimensional polygonal domains (17). In (1) a lower bound of $\lceil n/2 \rceil - 1$ on the number of diffuse reflections required to illuminate an orthogonal polygon is established, while (29) has already proved a lower bound of at least $\lceil (3n - 1)/2 \rceil - 4$.

Ray Tracing, proposed by Whitted (31), has been extensively studied by the Computer Graphics community producing photo realistic effects, e.g. as Metropolis Light Transport algorithm (30). To our knowledge the seminal work of Reif et al. (23) is the only one investigating its computational complexity so far. The main result is that the ray tracing problem is as hard as the Halting problem to much of the surprise of the community, which raises some paradigmatic questions (32). Reif et al. consider six different scenarios, which are all in three dimensions in contrast to our approach. If we put the cases concerning refraction aside, which is not of our concern here, they have proved the following results.

- *Ray Tracing is undecidable* in three dimensions for reflective objects represented by a system of rational quadratic inequalities;
- *Ray Tracing is undecidable* in three dimensions for reflective and partially reflective surfaces represented by a system of linear inequalities, where some of the inequalities are allowed to be irrational.
- *Ray tracing is PSPACE-hard* in three dimensions with full and one-way mirrors with curves described by rational linear inequalities.
- *Ray tracing is in PSPACE* for $k \geq 2$ dimensions for a finite set of parallel and perpendicular reflective surfaces represented by a system of rational linear equalities.

The Billiards problem is related to Ray Tracing and Illumination, see (14; 15; 19) for surveys. Gutkin (14) introduced the unfolding technique to analyze Billiard problems. Tokarsky et al. (26) has studied three different types of billiard tables, one formed of isosceles right triangles, another made of isosceles acute triangles and thereafter the one formed using obtuse angled triangles. He is using a mirroring technique to determine whether or not a point, is illuminable from a given point A in the room. Tabachnikov (25) surveyed several interesting results in this area highlighting the existence of caustics, the periodicity of billiard paths in polygons and chaotic billiards. The complexity of trajectories in rectangular billiards is the number of different reflection sequences (word complexity) of length n . In (6) the authors present how to compute this number. Billiards in polygons are not chaotic implying that nearby orbits do not diverge at any exponential speed. This fact and the growth rate of $\Omega(n^d)$ for an n -sided polygon in d -dimensions is explored in (24). A relationship between the word complexity and the number of generalized diagonals for polygonal billiard is established in (9). A new method for analyzing growth rates for geometric complexities and counting functions in polygonal billiards are introduced in (16). In (7) the word complexity of directional billiard in the cube some rational polyhedra is analyzed. Skripchenko and Troubetzkoy (24) prove that a polygonal billiard with one-sided mirrors has zero topological entropy and show sub-exponential or even polynomial estimates for the complexity in certain cases.

Motivated by the complexity of Ray Tracing and Illumination, it was recently proposed in (21) to use optical gates as basic building block of a symmetric encryption scheme. It turns out that the combination of non-optical, i.e. Boolean gates, and optical gates increases security and results in a novel fast encryption method. In a different direction Eppstein (11) identified a class of octagonal mirrors, that produce very simple predictable paths.

3 A pol-time Algorithm for Ray Tracing with Limited Reflections

To solve the ray tracing problem, one can trace the mirrored relative position of the source over a sequence of reflections. We assume that the inputs are given as rational numbers (numerators and denominators as binary integers); that is, the source and target positions as well as the end points of all mirrors.

Observation 3.1

Given $k + 1$ dimensional plane mirrors, the mirrored relative position of source over a sequence of reflections can be given by rational numbers such that the numerators and denominators are always polynomial in the input length and number of reflections.

Proof. To prove this, we proceed as follows: A k -dimensional simplex² is the convex hull determined by $k + 1$ points A_1, \dots, A_{k+1} where all vectors $\vec{v}_i = \overline{A_1 A_{i+1}}$ from A_1 to A_{i+1} ($i \in \{1, \dots, k\}$) are linearly independent. We assume that the k -dimensional simplices are located within a k -dimensional hyperplane of the $k + 1$ -dimensional space. For example, a segment as a one-dimensional mirror in 2d or a triangle as a 2-dimensional mirror in 3d.

Given such a simplex, we can determine an orthogonal vector \vec{u} , for the k -dimensional hyperplane given by the vectors \vec{v}_i of the simplex, by solving the k equations for $i \in \{1, \dots, k\}$ such that $\vec{v}_i \cdot \vec{u} = 0$. In the case of k -dimensional simple, merging or splitting mirrors, we assume that \vec{u} also determines the direction of the reflection.

Since we have to determine $k + 1$ coordinates by k equations, we will choose the value 1 for the first coordinate of \vec{u} if this is possible, and 0 otherwise. If the coordinates of the points A_1, \dots, A_{k+1} are given by quotients of binary or integer numbers then we can determine the coordinates of \vec{u} in polynomial time, and the representation of these coordinates require at most polynomial space. We can therefore assume without changing the computational complexity, that such an orthogonal vector \vec{u} is always given together with the $k + 1$ points A_1, \dots, A_{k+1} of the simplex.³

²Recall that a k -dimensional simplex is a generalisation of a triangle from 2 dimensions to k dimensions.

³Note that we do not use the normal, since it is usually scaled to have unit length. Hence the values of the coordinates of the normal might be an irrational number and cannot be written down by simple numbers in bounded time.

For a $k + 1$ -dimension we formally define the reflection on mirror given by a k -dimensional simplex by using the simple reflection on the corresponding k -dimension hyperplane.

Recall that a point A in the $k + 1$ -dimensional space also determines a vector from the point of origin to A . Let $\vec{u} \cdot \vec{v}$ denote the inner product of \vec{u} and \vec{v} . Then if we assume that the k -dimension hyperplane $L_i = (A_{i,1}, \dots, A_{i,k+1})$ with an orthogonal vector \vec{u}_i includes the origin, then the simple reflection S' of a point S can be determined by $S' = S - 2 \frac{\vec{u}_i \cdot S}{\vec{u}_i \cdot \vec{u}_i} \vec{u}_i$.

If the origin is not a point of the hyperplane then we have to shift the coordinate system before and after the reflection by a point of the simplex, e.g.

$$S' = \left((S - A_{i,1}) - 2 \cdot \frac{\vec{u}_i \cdot (S - A_{i,1})}{\vec{u}_i \cdot \vec{u}_i} \cdot \vec{u}_i \right) + A_{i,1} = S - 2 \cdot \frac{\vec{u}_i \cdot S}{\vec{u}_i \cdot \vec{u}_i} \cdot \vec{u}_i + 2 \cdot \frac{\vec{u}_i \cdot A_{i,1}}{\vec{u}_i \cdot \vec{u}_i} \cdot \vec{u}_i.$$

Let $\bar{u}_i = \frac{2}{\vec{u}_i \cdot \vec{u}_i} \cdot \vec{u}_i$ and $\hat{u}_i = \frac{2 \cdot \vec{u}_i \cdot A_{i,1}}{\{\vec{u}_i \cdot \vec{u}_i\}} \cdot \vec{u}_i$ denote two scaled version of the vector \vec{u}_i , and for simplification we assume that both vectors are also part of the simplex description, then

$$S' = S - (\bar{u}_i, S) \cdot \vec{u}_i + \hat{u}_i. \quad (1)$$

Analysing Equation 1 one can see that for every reflection, the number of bits in the numerators and denominators increases by a value at most linear in the input size. □

Hence, if the number of reflections is polynomial in the input size, the values remain polynomial in the input size. As such, the ray tracing problem can be solved deterministically in polynomial time, if the sequence of used mirrors can be determined deterministically by following the ray. Note that this is only possible with simple and merging mirrors. In the case of one-way and splitting mirrors, one has to distinguish between a reflection of a beam and the possibility that the beam might pass through the mirror. If we non-deterministically guess this behaviour, the algorithm described above can solve the problem in non-deterministic polynomial time. Hence we can conclude:

Theorem 1. *The $k + 1$ -dimensional ray tracing problem with rational input values and a polynomial number of reflections is in \mathcal{P} if we assume simple plane mirrors and merging mirrors. The problem is in \mathcal{NP} for a polynomial number of reflections and we allow simple plane mirrors, merging, splitting and one-way mirrors.*

For solving the illumination problem with a polynomial number of reflections, one can proceed analogously to the ray tracing problem. Note that even if we only consider simple plane mirrors, the sequence of used mirrors from the source to the target cannot be determined by following a ray through the collection of mirrors as we might have to investigate a cone of infinitely many rays. If however, we guess non-deterministically the sequence of used mirrors within a first phase, one can use a similar algorithm to verify that the target will be illuminated by using the given number of reflections. Hence we can conclude:

Theorem 2. *The $k + 1$ -dimensional illumination problem with rational input values and a polynomial number of reflections is in \mathcal{NP} if we allow simple plane mirrors, merging, splitting and one-way mirrors.*

4 Ray Tracing in 2D with One-Way Mirrors is \mathcal{NP} -Complete

In Section 3 we have seen that the ray tracing problem can be solved in \mathcal{P} for plane mirrors and with a polynomial number of reflections if the sequence of used mirrors can be determined deterministically in polynomial time. Using splitting and merging of one-way mirrors this seems to be a hard task. Since it is possible to guess this sequence non-deterministically, this problem is still in \mathcal{NP} (see Theorem 1). In the following we will show that this version of the ray tracing problem is \mathcal{NP} -hard. The hardness proof follows by a reduction of the subset sum problem (12):

Definition 1. *Given a finite set $X = \{1, \dots, n\}$, weights $x_i \in \mathbb{Z}^+$ for each $i \in X$, and a positive integer $B \in \mathbb{Z}^+$. Decide whether there exists a subset $X' \subseteq X$ such that $\sum_{i \in X'} x_i = B$.*

For the reduction we assume a beam starts at source $(0, 0)$ and follows the x -axis in positive direction. It then passes through n mirror modules constructed as in Figure 2. Each module represents one element $i \in X$ of weight x_i . Within each module the beam is duplicated by using a splitting mirror. The two beams leave the module in such a way that the original beam maintains its y -axis value while the second beam is shifted according to the y -axis by an offset of $-x_i$ for the corresponding element $i \in X$.

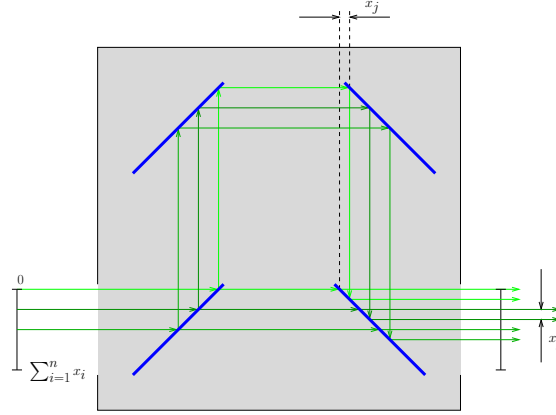


Figure 2: Representation of a weight x_j within the reduction of subset sum to ray tracing.

The target position will be at the coordinate $(h, -B)$ where h is a value large enough such that all the modules can be placed between the source and the target. Analysing this construction one can see, that the target will be reached by a beam iff the subset sum problem has a solution. Note that the reduction can be performed within polynomial time. Thus we can conclude that:

Theorem 3. *The subset sum problem is polynomial time reducible to the ray tracing problem with a linear number of reflections and if simple plane and one-way mirrors (or alternatively splitting and merging mirrors) can be used. Hence, this problem is \mathcal{NP} -hard.*

Combining Theorem 3 and 1 we can conclude that ray tracing with a linear number of reflections, and if simple plane and one-way mirrors (or alternatively splitting and merging mirrors) are used, is \mathcal{NP} -complete.

5 Ray Tracing in 2D is \mathcal{P} -Hard without Beam Duplication

In the previous section we have seen that one-way or splitting mirrors can be used to simulate non-deterministic guesses by beam duplication and sending the two resulting beams into different directions. In this section, we would like to investigate the computational power of a system of mirrors if the splitting and one-way mirrors are not available to perform the aforementioned duplication.

We show that circuits can be evaluated using plane mirrors, merging mirrors and parabolic mirrors such that the circuit value is computed by simulating a version of backtracking. The only available memory is a kind of stack represented by an offset of the single ray passing through a system of mirrors which represent the circuit.

For the offset of the beam position, we assume values $(0, 1)$. Given an offset of $s \in (0, 1)$ we can implement a push of a bit $b \in \{0, 1\}$ by $s' = (s + b) \cdot 2^{-1}$ and a pop by $s' = (2 \cdot s) - \lfloor 2 \cdot s \rfloor$ and determining $b = \lfloor 2 \cdot s \rfloor$. The multiplication by 2 or by 2^{-1} is a major tool for these operations. To implement this we use a combination of parabolic mirrors, e.g. for the multiplication with 2^{-1} (see Figure 3) $-2 \cdot x^2 + \frac{3}{8}$ for $x \in [-1, 0]$, and x^2 for $x \in [0, 2]$.

We assume that all circuit gates have fan-in and fan-out at most 2, and that the gate types are restricted to input literals, constants, negations, identity gates (for increasing the number of successors which can use an intermediate result), AND and OR gates.

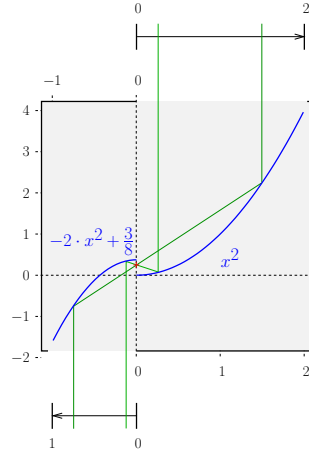


Figure 3: The example construction of multiplying the offset by 2^{-1} .

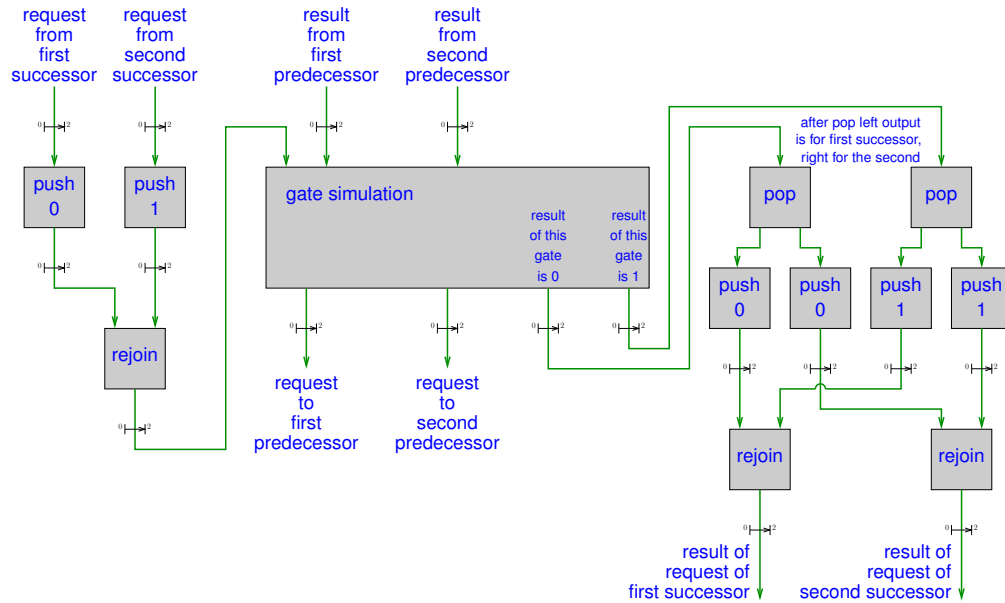


Figure 4: General stack behaviour for the evaluation of a gate.

The main strategy for the circuit evaluation works are as follows (see Figure 4):

1. Assume we would like to evaluate a gate u . Then the beam which implements the stack is within the mirror system of u and will be passed via specific mirrors as a result of a request from one of its predecessors v .
2. If u is the first successor of v we push a 0 to our stack, otherwise we push a 1. We assume that this value will not be removed from the stack by a mirror system of another gate.
3. By merging mirrors we combine the possible offsets to a single interval of offsets $(0, 1)$.
4. We determine the value of v by sending a request to its first predecessor if it exists. We assume that the top element of the stack is the result of this predecessor when the beam is coming back. If necessary, we also send a request to the second predecessor after the beam has arrived from the first predecessor.
5. We remove the results by popping from the stack and, depending on the predecessor results, we send the beam along the appropriate path through the rest of the mirror stack.
6. By popping, we can now determine the asking successor (i.e. u) of v , push the result of v to the stack and route the beam to the requesting successor of v .

To illustrate how a gate is evaluated see Figure 5 for the mirror system of an AND gate.

This construction shows that the circuit value problem can be solved by a system of simple plane mirrors, merging mirrors, and parabolic mirrors. We conclude that:

Theorem 4. *The ray tracing problem in 2D for simple plane mirrors, merging mirrors, and parabolic mirrors for an exponential number of reflections is \mathcal{P} -hard.*

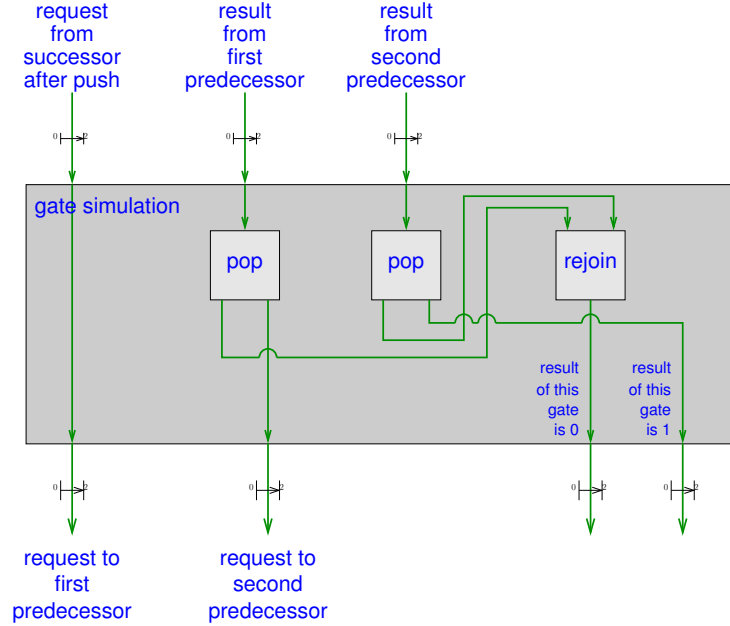


Figure 5: Mirror system of an AND gate.

6 Illumination in 2D is \mathcal{NP} -Complete

As we have seen in Section 3, we can solve the k -dimensional illumination problem with plane mirrors (simple, merging, splitting, and one-way mirrors) in \mathcal{NP} , if we bound the number of reflections by a polynomial. In this section, we would like to investigate a lower bound for this problem. We would like to show that the problem is \mathcal{NP} -hard even in 2D and if we restrict ourselves to simple plane mirrors. We start by proving that this problem is \mathcal{NP} -hard if we allow an exponential number of reflections. Based on this proof, we then show that the problem remains \mathcal{NP} -hard, if we restrict ourselves to a linear number of reflections.

Theorem 5. *The illumination problem restricted to simple plane mirrors in 2D is \mathcal{NP} -hard, if an exponential number of reflections is allowed.*

The main parts of the construction of the reduction of Theorem 5 from subset sum to the illumination problem is illustrated in Figure 6.

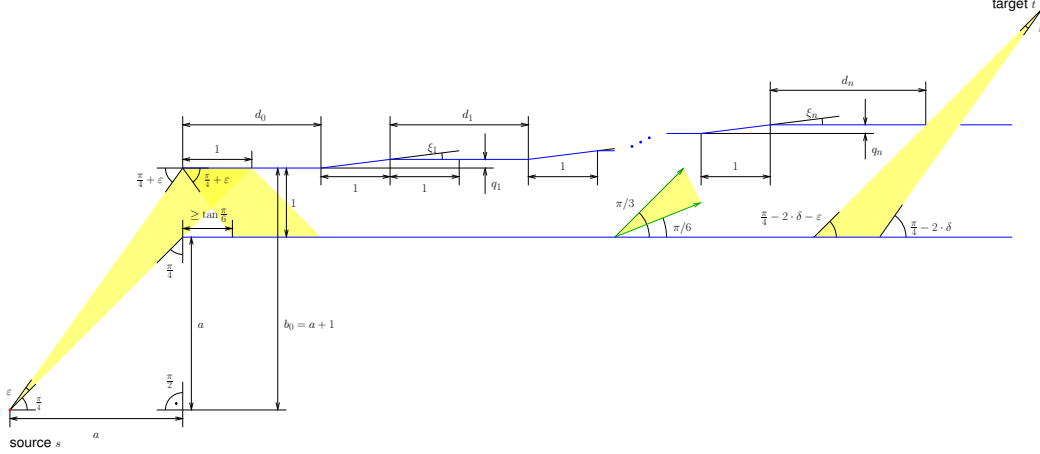


Figure 6: Overview of the construction of the reduction used in Theorem 5.

Recall that for an instance of the subset sum problem finite set $X = \{1, \dots, n\}$ and weights $x_i \in \mathbb{Z}^+$ for each $i \in X$ as well as a bound $B \in \mathbb{Z}^+$ is given. We would like to decide whether there exists a subset $X' \subseteq X$ such that $\sum_{i \in X'} x_i = B$. For the following reduction, we translate the values x_i into angles $\xi_i = \frac{\pi}{24} \cdot \frac{x_i}{\sum_i x_i}$ and ask whether we can find a collection $X' \subseteq X$ of these angles such that $\sum_{i \in X'} \xi_i = \delta = \frac{\pi}{24} \cdot \frac{B}{\sum_i x_i}$. To implement a selection process for the different angles, we have to use cones which are partially reflected by a tilted mirror with angle ξ_i and partially reflected by a horizontal mirror. For each of these selection processes the cone is subdivided into two sub-cones, one cone with the angle of the included beams unchanged, and one where the angle of the included beams are decreased by $2 \cdot \xi_i$ for some values i . Initially we assume that the cone has angle widths of $\varepsilon = \min_i \xi_i \leq \frac{\pi}{24n} \leq \frac{\pi}{48}$, i.e. initial differences of the directions of the beams is at most ε .

For the reduction, we construct a tube of mirrors, where the bottom consists of a horizontal mirror and the ceiling consists of a sequence of alternating horizontal and tilted mirrors. We start by a horizontal mirror of length d_0 , followed by a tilted mirror of horizontal length 1 and angle ξ_1 , then by a horizontal mirror of length d_1 , then by a tilted mirror of horizontal length 1 and angle ξ_1 , and so on. Within the final horizontal ceiling mirror at position d_n we include a gap which allows the target to be reached by a beam, only if the beam belongs to a sub-cone which consists only of beams which original angles are modified by $2 \cdot \delta$.

The location of the source guarantees that the initial cone which enters the tube has an horizontal width of 1 and it consists only of beams of angle from the range of $\frac{\pi}{4}$ to $\frac{\pi}{4} + \varepsilon$ (according to a horizontal line). In the following the angle values always describe the angle of a beam after it is reflected by the bottom mirror of the tube. If a beam might be reflected by a tilted mirror at most once, the angle of every beam and at every position within the tube will be within the range of $\frac{\pi}{4} - 2 \cdot \sum_i \xi_i$ to $\frac{\pi}{4} + \varepsilon$. Note that $\frac{\pi}{6} = \frac{\pi}{4} - 2 \cdot \sum_i \xi_i = \frac{\pi}{4} - \frac{2 \cdot \pi}{24}$ and $\frac{\pi}{4} + \varepsilon \leq \frac{\pi}{4} + \frac{\pi}{48} < \frac{\pi}{3}$. If we start with a tube of height 1 this guarantees that all tilted mirror will be hidden by a beam at most once, more precisely since the angle is at most $\frac{\pi}{4} + \varepsilon < \frac{\pi}{3}$ for every reflection in the tube each beam moves at least $\frac{1}{\sqrt{3}} > \frac{1}{2}$ units to the right and for each double reflection each beam moves for more than 1 to the right. In addition this verifies that every beam which enters the tube in the described way will always move up and down and to the right.

Note that the height of the tube will be increased by every tilted mirror of angle ξ_i by $q_i = \tan \xi_i$. Hence the final height of the tube will be $h_{\max} = 1 + \sum_i q_i$. Based on the angle sum rule for the tangents function one can show that $\sum_i q_i = \sum_i \tan(\xi_i) \leq \tan(\sum_i \xi_i) \leq \tan(\frac{\pi}{24}) < \frac{1}{3}$. Hence $h_{\max} \leq \frac{4}{3}$. We can conclude that the distance of the most narrow beam (with angle $> \pi/6$) moves by reflections at most $2 \cdot h_{\max} \cdot \cot(\frac{\pi}{6}) = 2 \cdot \sqrt{3} \cdot h_{\max} \leq \frac{8 \cdot \sqrt{3}}{3} \leq 5$.

Thus we are looking for the number of reflection and the required length d_i such that the cones which occur in the corresponding sub-tube covers at least an area of length 5 at the bottom horizontal mirror.

To determine the required length d_i and the number of required reflection to get a horizontal widths of the cones, we have to determine the vertical distance of the virtual (relative reflected) source b_i where the cone reaches the required horizontal widths. For the initial value b_0 we can choose for example $b_0 = a + 1 = 2 \cdot h_{\max}$. Based on b_i we can determine d_i by $d_i = \sqrt{3} \cdot b_i$ and the number of reflection by $(b_i - b_{i-1}) / (1 + \sum_{j=1}^{i-1} q_j)$. Recall that a tilted mirror reduces the vertical distances b_i of the virtual (relative reflected) source compared to using only horizontal mirrors. If

we are looking for upper bounds on the length d_i and the number of reflections we could investigate b_i for using only horizontal mirrors, and bound the number of reflections by b_i .

To determine b_i we use Thales's theorem. Whenever a cone starts its way through the horizontal mirror tube (ceiling mirror is horizontal and has length d_i), we assume that the horizontal widths is 1 and the vertical distance is b_{i-1} (plus a small value upper bounded by a constant c). At the end of this area we assume that the cone has horizontal widths of 5 and the vertical distance is the unknown value b_i . By Thales's theorem we can conclude that $b_i = 5 \cdot b_{i-1} + 5 \cdot c \in b_0 \cdot 2^{\mathcal{O}(n)}$. Note that the start and end positions of the concrete mirrors as well as the target point as determined above by the values ξ_i , b_i , and d_i result in irrational numbers. To approximate these values with rational numbers we search for values such that the sum of angular distances is small compared to ε , e.g. $\varepsilon/100$.

To reduce the number of reflections to $\mathcal{O}(n)$ we replace the selection areas (starting shortly after a tilted mirror and ending after the following tilted mirror) of the construction above by the construction presented in Figure 7.

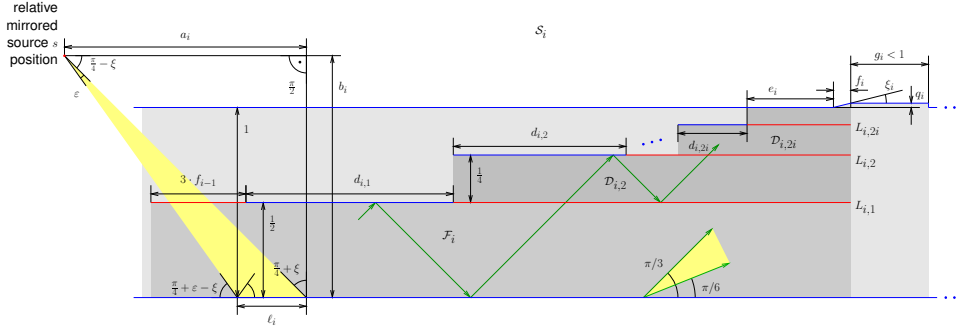


Figure 7: Overview of the construction of the selection area used in the reduction in Theorem 6.

Theorem 6. *The illumination problem restricted to simple plane mirrors and merging mirrors in 2D is \mathcal{NP} -hard, even if only a linear number of reflections is allowed.*

The construction of the \mathcal{NP} -hardness proof for the linear number of reflections follow our main approach of the construction within the proof of Theorem 5. For reducing the number of reflections let us investigate the question what causes the exponential number of reflections. The main reason is given by the required expansion of the horizontal widths of every cone from 1 to 5. This was necessary to guarantee that the tilted mirrors plus a part of the following horizontal mirrors are covered by any cone. For this we assumed that

1. horizontal widths of every cone is at least 2 and
2. the bottom mirror in this area is completely covered by every cone.

To address the first task we reduce the horizontal widths f_i of the tilted mirrors for ξ_i from 1 to 2^{-2i} . Hence it is required that horizontal widths of every cone at the tilted mirror with angle ξ_i is at least $2 \cdot 2^{-2i}$ and therefore every cone directly after the tilted mirror with angle ξ_i has widths 2^{-2i} which is already $4 \cdot 2^{-2(i+1)}$ and therefore large enough to fulfil this requirement for the next sub-construction with a tilted mirror with angle ξ_{i+1} and horizontal widths of $2^{-2(i+1)}$. Note, that this reduces $q_i = \frac{1}{2^{2i}} \cdot \tan(\xi_i)$.

For the second task we would like to reduce the height of the tube close to the tilted mirror with angle ξ_i to 2^{-2i} . In this case the distance of the most narrow ray (with angle $> \pi/6$) after two reflections within the upper sub-tube is at most $2 \cdot 2^{-2i} \cdot \cot \pi/6 = 2 \cdot 2^{-2i} \cdot \sqrt{3} < 4 \cdot 2^{-2i} = 2^{-2(i-1)}$. As we have seen above this is already the case.

Within this setting we have that within this area (a sub-tube of height 2^{-2i}) for every reflection each beam moves at least $\frac{1}{\sqrt{3}} \cdot 2^{-2i} > \frac{1}{2} \cdot 2^{-2i}$ units to the right, since the angle of a beam is at most $\pi/4 + \varepsilon \leq \pi/3$ we get for each double reflection that each beam moves for more than 2^{-2i} . This implies that a beam hits an interval of length 2^{-2i} at most once.

It remains to address the question of how we can reduce the height in such a way that the distances between all the cones are relatively small. For this we use a system of simple mirrors and merging mirrors as presented in Figure 7 and 8.

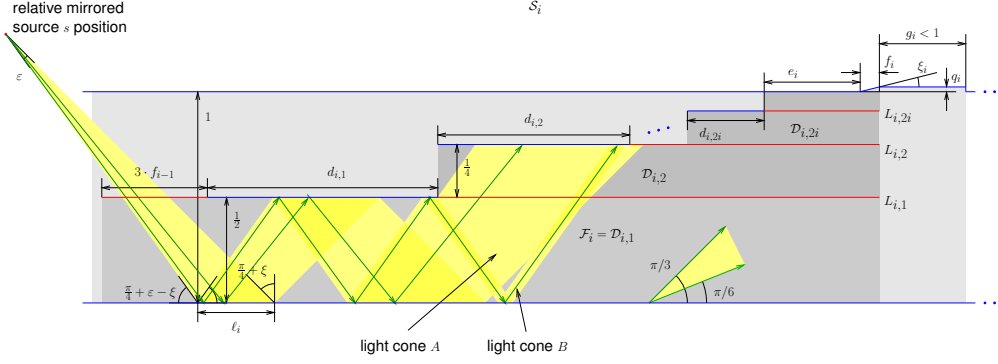


Figure 8: At the point, where a light cone within an area $\mathcal{F}_i = \mathcal{D}_{i,1}$, or $\mathcal{D}_{i,j}$ respectively, reaches the merging mirror of $L_{i,j}$ it might be split into 2 parts, a part A which passes through the merging mirror and enters $\mathcal{D}_{i,j+1}$, and a part B which stays in $\mathcal{D}_{i,j}$. The later part B will enter $\mathcal{D}_{i,j+1}$ after one additional reflection at the bottom of $\mathcal{D}_{i,j}$. This corresponds with 3 reflections within $\mathcal{D}_{i,j+1}$ of part A . Thus at the fourth reflection of part A it will be recombined with part A to the original cone.

The construction \mathcal{S}_i which leads the cones from the tilted mirror with angle ξ_{i-1} to the tilted mirror with angle ξ_i consist out of $2i$ mirror lines $L_{i,j}$, with $j \in \{1, \dots, 2i\}$ where each line reduces the height of the active sub-tube by a factor of $\frac{1}{2}$. The lines $L_{i,j}$ start with an initial simple mirror of length $d_{i,j}$ (starting at the end of the simple mirror of lines $L_{i,j-1}$ in a vertical distance of 2^{-j} above $L_{i,j-1}$). After the initial simple mirror $L_{i,j}$ is complemented by a merging mirror which allows upgoing beam to pass. The merging mirror end if the (horizontal) end position of the tilted mirror with angle ξ_i is reached. In addition we add before the initial simple mirror of $L_{i,1}$ a merging mirror of length $3 \cdot 2^{-2(i-1)}$, which allows downgoing beams to pass, for trapping all cones of the previous sub-construction \mathcal{S}_{i-1} .

We now have to investigate the construction and determine the conditions of the different values such that the tube fulfils our requirements of the reduction. We start with the analysis of $d_{i,j}$ where $j \geq 2$. Note that whenever a cone might pass through a merging mirror of $L_{i,j-1}$ the first time, it might be split into two parts which can be recombined after one reflection (respectively three reflections), and it is required that $d_{i,j} \geq 4 \cdot 2^{-j} \cdot \cot \pi/6 = 2^{-j} \cdot \sqrt{3}$ or $d_{i,j} \geq 2^{3-j}$. In the case of a large horizontal width of the cone, it might be possible that the part of the cone which stays inside area below $L_{i,j-1}$ might be split again. But since we one need a cone of horizontal width $4 \cdot 2^{-2i}$ which is 4 times larger than the smallest sub-tube of this construction, and since each cone moves at least the height of a sub-tube for each double reflection to the right we have to consider up to 5 splitting, which results in $d_{i,j} \geq 10 \cdot 2 \cdot 2^{-j} \cdot \cot \pi/6 = 5 \cdot 2^{-j} \cdot \sqrt{3}$ or $d_{i,j} \geq 2^{6-j}$. If we assume that a beam should have at least c reflections at the simple mirror of $L_{i,j}$, i.e. that the most narrowed ray from $\mathcal{D}_{i,j-1}$ can enter $\mathcal{D}_{i,j}$ and has at least c double reflection within $\mathcal{D}_{i,j}$ even if the ray enters $\mathcal{D}_{i,j}$ as far as possible to the right, then we have to choose $d_{i,j} \geq 2 \cdot \cot(\pi/6) \cdot 2^{-i+1} + c \cdot \cot(\pi/6) \cdot 2^{-j} = (4+c) \cdot \sqrt{3} \cdot 2^{-j}$. For our construction $c \leq 32 < \frac{2^6}{\sqrt{(3)}} - 4$ it is sufficient.

This allows us to choose $d_{i,j} = 2^{6-j}$. Choosing the length of the merging mirror part of $L_{i,2i}$ by $2 \cdot d_{i,2i}$ guarantees that the upper surface of $L_{i,2i}$ will be completely illuminated by every cone before any cone reaches the tilted mirror of this sub-construction.

For $L_{i,1}$ we have to investigate $3 \cdot f_{i-1} + d_{i,1}$. This length has to guarantee that every light cone has entered \mathcal{F}_i and is reflected at least 3 times before it leaves the area. Hence we choose $3 \cdot f_{i-1} + d_{i,1} \geq 4 \cdot \cot(\pi/6) \cdot 2^{-1} = \sqrt{3} \cdot 2^1$, and in the worst case, $3 \cdot f_{i-1}$ is at most 3. To have equivalent values at all levels we choose $3 \cdot f_{i-1} + d_{i,1} = 2^{6-1}$.

Finally, and before counting the number of reflections, we have to show that the most and the least narrowed cones can enter (the same) \mathcal{S}_i correctly. For this we assume that the most narrowed cone (or better beam) leaves \mathcal{S}_{i-1} reflected by the horizontal ceiling mirror f_{i-1} units behind the tilted mirror of \mathcal{S}_{i-1} , and that the least narrowed cone (or beam) leaves \mathcal{S}_{i-1} (or better reflected that last time within \mathcal{S}_{i-1}) at the beginning of the tilted mirror, that is f_{i-1} units before the end of \mathcal{S}_{i-1} . To show that both beams enter \mathcal{S}_i correctly we have to verify that $\frac{3}{2} \cdot \cot(\pi/3) - f_{i-1} + 3 \cdot f_{i-1} > \frac{1}{2} \cdot \cot(\pi/6) + f_{i-1}$ for all i , i.e. $\frac{\sqrt{3}}{2} + f_{i-1} > \frac{\sqrt{3}}{2}$. To combine the sub-constructions \mathcal{S}_i we choose $g_i = \frac{1}{2}$ and assume that \mathcal{S}_{i+1} start directly after this tiny corner at the ceiling, which reduces the height of the total tube back to 1.

For the number of reflection within the different areas we have to examine the number of reflections within $\mathcal{D}_{i,j}$. This value can be upper bounded by

$$\frac{d_{i,j}}{\frac{1}{\sqrt{3}} \cdot 2^{-j}} + 12 \leq 2^6 \cdot \sqrt{3} + 12 < 123 \quad \text{resp. by} \quad \frac{3 \cdot f_{i-1} + d_{i,1}}{\frac{1}{\sqrt{3}} \cdot 2^{-j}} + 12 \leq 2^6 \cdot \sqrt{3} + 12 < 123$$

In addition we get for the top most tube $2 \cdot 123 + 1$. To sum up all the reflections of the areas $\mathcal{D}_{i,j}$ for one value i we get $123 \cdot (2i + 2)$ reflections for \mathcal{S}_i . And in total for all \mathcal{S}_i (we start with \mathcal{S}_1 , then we use \mathcal{S}_2 , and so on, and for the final step, to add the hole which leads to the target we use a similar construction to \mathcal{S}_n) we get $\Theta(n^2)$ reflections. Since we use n^2 mirrors, the number of reflections is linear within the number of used mirrors.

If we only work with one-way merging mirrors and not full mirrors, and do the right type of combining them, then we can make a further improvement using just merging mirrors for Theorem 6.

7 Conclusions

We introduce multiple scenarios of the Ray Tracing and the Illumination problems. We show that Ray tracing with a linear number of reflections and if simple plain mirrors and one-way mirrors (or alternatively splitting and merging mirrors) are used, is \mathcal{NP} -complete. Furthermore, we show that the circuit value problem can be solved by a system of simple plain mirrors, merging mirrors, and parabolic mirrors in the case that splitting and one-way mirrors are not available to perform duplication of the light rays, sending the two resulting beams into different directions. We also build a stack-based mirror system to represent different boolean logic gates. Finally in the section 6, we infer that Illumination in 2D is \mathcal{NP} -Complete.

Open Problems The question whether the Ray Tracing in 2D is as hard as in 3D remains open. The center point of the construction of Reif et al. (5) is a two-dimensional offset of a bundle of possible parallel ray positions, where the decimal places in binary representation of the cartesian coordinates encode two counters. These two counters are the very ones from the famous two-counter machine that Minsky (20) proved to be undecidable. A straight-forward transformation to 2D leaves only one counter, which results in a Push-Down-Automaton, which is in fact a decidable class. So far, it is open whether another construction can prove the undecidability for Ray Tracing in 2D.

We have shown that Illumination is already \mathcal{NP} -hard for only plane mirrors, indicating that it may be harder than Ray Tracing. However, the question whether Illumination is \mathcal{NP} -hard for a polynomial number of reflection is open, albeit the work of (4) suggests that exponential description complexity exists.

References

- [1] Mridul Aanjaneya, Arijit Bishnu, and Sudebkumar Prasant Pal. Directly visible pairs and illumination by reflections in orthogonal polygons. In *24th European Workshop on Comput. Geom.*, pages 241–244, 2008.
- [2] Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. The art gallery problem is $\exists R$ -complete. *ACM Journal of the ACM (JACM)*, 69(1):1–70, 2021.
- [3] Boris Aronov, Alan R. Davis, T. K. Dey, S. P. Pal, and D. C. Prasad. Visibility with one reflection. *Discrete & Computational Geometry*, 19(4):553–574, 1998. URL: <https://doi.org/10.1007/PL00009368>, doi: 10.1007/PL00009368.
- [4] Boris Aronov, Alan R. Davis, Tamal K. Dey, Sudebkumar Prasant Pal, and D Chithra Prasad. Visibility with multiple reflections. *Discrete & Computational Geometry*, 20(1):61–78, 1998. URL: <https://doi.org/10.1007/PL00009378>, doi: 10.1007/PL00009378.
- [5] Boris Aronov, Alan R. Davis, John Iacono, and Albert Siu Cheong Yu. The complexity of diffuse reflections in a simple polygon. In José R. Correa, Alejandro Hevia, and Marcos Kiwi, editors, *LATIN 2006: Theoretical Informatics*, pages 93–104, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [6] Yu. Baryshnikov. Complexity of trajectories in rectangular billiards. *Communications in Mathematical Physics*, 174(1):43–56, 1995. URL: <https://doi.org/10.1007/BF02099463>, doi: 10.1007/BF02099463.
- [7] Nicolas Bedaride. Billiard complexity in rational polyhedra. *Regular and Chaotic Dynamics*, 8(1):97, 2003. URL: <https://doi.org/10.1070%2Frd2003v008n01abeh000228>, doi: 10.1070/rd2003v008n01abeh000228.
- [8] Arijit Bishnu, Subir Kumar Ghosh, Partha Pratim Goswami, Sudebkumar Prasant Pal, and Swami Sarvattomananda. An algorithm for computing constrained reflection paths in simple polygon. *arXiv preprint arXiv:1304.4320*, 2013.
- [9] Julien Cassaigne, Pascal Hubert, and Serge Troubetzkoy. Complexity and growth for polygonal billiards. In *Annales de l’institut Fourier*, volume 52, pages 835–847, 2002.
- [10] David Castro. Corrections. *Quantum*, 7(42), Jan. 1997.
- [11] David Eppstein. Reflections in an octagonal mirror maze, 2022. [arXiv:2206.11413](https://arxiv.org/abs/2206.11413).
- [12] Michael R. Garey and David S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-completeness*. Freeman, 1979.
- [13] Subir Kumar Ghosh, Partha Pratim Goswami, Anil Maheshwari, Subhas Chandra Nandy, Sudebkumar Prasant Pal, and Swami Sarvattomananda. Algorithms for computing diffuse reflection paths in polygons. *The Visual Computer*, 28(12):1229–1237, 2012. URL: <https://doi.org/10.1007/s00371-011-0670-z>, doi: 10.1007/s00371-011-0670-z.
- [14] Eugene Gutkin. Billiards in polygons. *Physica D: Nonlinear Phenomena*, 19(3):311–333, 1986. URL: <https://www.sciencedirect.com/science/article/pii/016727898690062X>, doi: [https://doi.org/10.1016/0167-2789\(86\)90062-X](https://doi.org/10.1016/0167-2789(86)90062-X).
- [15] Eugene Gutkin. Billiard dynamics: An updated survey with the emphasis on open problems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(2), 2012.
- [16] Eugene Gutkin and Michał Rams. Growth rates for geometric complexities and counting functions in polygonal billiards. *Ergodic Theory and Dynamical Systems*, 29(4):1163–1183, 2009. doi: 10.1017/S0143385708080620.
- [17] Irina Kostitsyna, Maarten Löffler, Valentin Polishchuk, and Frank Staals. On the complexity of minimum-link path problems. *arXiv preprint arXiv:1603.06972*, 2016.
- [18] Samuel Lelievre, Thierry Monteil, and Barak Weiss. Everything is illuminated. *Geometry & Topology*, 20(3):1737–1762, 2016.
- [19] Howard Masur and Serge Tabachnikov. Chapter 13 rational billiards and flat structures. volume 1 of *Handbook of Dynamical Systems*, pages 1015–1089. Elsevier Science, 2002. URL: <https://www.sciencedirect.com/science/article/pii/S1874575X02800157>, doi: [https://doi.org/10.1016/S1874-575X\(02\)80015-7](https://doi.org/10.1016/S1874-575X(02)80015-7).
- [20] Marvin Lee Minsky. *Computation*. Prentice-Hall Englewood Cliffs, 1967.
- [21] Sneha Mohanty, Eric Peairs, and Christian Schindelhauer. *Introducing Gate Based Ray Tracing Cryptography*. EasyChair, 2023.
- [22] L Penrose and R Penrose. Puzzles for christmas. *New Scientist*, 25:1580–1581, 1958.

- [23] John H. Reif, J. Doug Tygar, and A. Yoshida. Computability and complexity of ray tracing. *Discrete & Computational Geometry*, 11:265–288, 1994.
- [24] Alexandra Skripchenko and Serge Troubetzkoy. Entropy and complexity of polygonal billiards with spy mirrors. *Nonlinearity*, 28(9):3443–3456, aug 2015. URL: <https://doi.org/10.1088/0951-7715/28/9/3443>, doi:10.1088/0951-7715/28/9/3443.
- [25] Serge Tabachnikov. *Geometry and billiards*, volume 30. American Mathematical Soc., 2005.
- [26] G. Tokarsky. An impossible pool shot? *SIAM Review*, 37(1):107–109, 1995. URL: <https://doi.org/10.1137/1037016>, arXiv:<https://doi.org/10.1137/1037016>, doi:10.1137/1037016.
- [27] George W Tokarsky. Polygonal rooms not illuminable from every point. *The American mathematical monthly*, 102(10):867–879, 1995.
- [28] Csaba D Toth, Joseph O’Rourke, and Jacob E Goodman. *Handbook of discrete and computational geometry*. CRC press, 2017.
- [29] Jorge Urrutia. On the number of internal and external visibility edges of polygons. In *CCCG*, 1997.
- [30] Eric Veach and Leonidas J Guibas. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76, 1997.
- [31] Turner Whitted. An improved illumination model for shaded display. In *Proceedings of the 6th annual conference on Computer graphics and interactive techniques*, page 14, 1979.
- [32] Martin Ziegler. Physically-relativized church–turing hypotheses: Physical foundations of computing and complexity theory of computational physics. *Applied Mathematics and Computation*, 215(4):1431–1447, 2009.

A Appendix

A.1 Algorithms for Solving Ray Tracing and Illumination with Plain Mirrors

There are plenty of algorithm for the ray tracing problem. Most of the algorithms are efficient if we assume that one can perform operation on arbitrary numbers, e.g. on real or rational values. To see whether the algorithms also run efficiently on Turing Machines (or if we count the bit operations) one has to examine the length of the binary representations of the values which occurs while running the algorithm. If we assume that all input values (initial direction of the ray, source position, target position, endpoints of the mirrors, orthogonal vectors off the mirrors) are rational numbers of the form, p/q (given by binary representations of an integer numerator, p and integer denominator, q), one could see that for plain mirrors only rational intermediate values can occur. Therefore we assume that all intermediate values are represented by their binary integer numerator and denominator.

To simplify the input and the analysis we assume that all mirrors are k -dimensional simplices. Note that a k -dimensional simplex⁴ is the convex hull determined by $k + 1$ points A_1, \dots, A_{k+1} where all vectors $\vec{v}_i = \overline{A_1 A_{i+1}}$ from A_1 to A_{i+1} ($i \in \{1, \dots, k\}$) are linearly independent. We assume that the k -dimensional simplices are located within a k -dimensional hyperplane of the $k + 1$ -dimensional space, for example, a segment as a one-dimensional mirror in 2d or a triangle as a 2-dimensional mirror in 3d.

Given such a k -dimensional simplex in $k + 1$ -dimensional space we can determine an orthogonal vector \vec{u} , for the k -dimensional hyperplane given by the vectors \vec{v}_i of the simplex, by solving the k equations for $i \in \{1, \dots, k\}$

$$\vec{v}_i \cdot \vec{u} = 0$$

where $\vec{v}_i \cdot \vec{u}$ denote the inner product of \vec{u} and \vec{v}_i . In the case of k -dimensional simple, merging or splitting mirrors, we assume that \vec{u} also determines the direction in which the reflection occurs.

Since we have to determine $k + 1$ coordinates by k equations, we will choose the value 1 for the first coordinate of \vec{u} if this is possible, and 0 otherwise. In a realistic setting, we take the representation of coordinates as floating point numbers. However, to emphasize more precisely on the space complexity, in our setting, the coordinates of the points A_1, \dots, A_{k+1} are given by quotients of binary or integer numbers (for both the numerator and denominator), then we can determine the coordinates of \vec{u} in polynomial time, and the representation of these coordinates require at most polynomial space. We can therefore assume without changing the computational complexity, that such an orthogonal vector \vec{u} is always given together with the $k + 1$ points A_1, \dots, A_{k+1} of the simplex.⁵

We now formally define a reflection of a point S . For a $k + 1$ -dimension we define the reflection on mirror given by a k -dimensional simplex by using the simple reflection on the corresponding k -dimension hyperplane.

Recall that a point S in the $k + 1$ -dimensional space also determines a vector from the point of origin to S . Then if we assume that the k -dimension hyperplane $L_i = (A_{i,1}, \dots, A_{i,k+1})$ with an orthogonal vector \vec{u}_i includes the origin, then the simple reflection S' of a point S can be determined by

$$S' = S - 2 \frac{\vec{u}_i \cdot S}{\vec{u}_i \cdot \vec{u}_i} \vec{u}_i.$$

If the origin is not a point of the hyperplane then we have to shift the coordinate system before and after the reflection by a point of the simplex, e.g.

$$S' = \left((S - A_{i,1}) - 2 \cdot \frac{\vec{u}_i \cdot (S - A_{i,1})}{\vec{u}_i \cdot \vec{u}_i} \cdot \vec{u}_i \right) + A_{i,1} = S - 2 \cdot \frac{\vec{u}_i \cdot S}{\vec{u}_i \cdot \vec{u}_i} \cdot \vec{u}_i + 2 \cdot \frac{\vec{u}_i \cdot A_{i,1}}{\vec{u}_i \cdot \vec{u}_i} \cdot \vec{u}_i.$$

Let $\bar{u}_i = \frac{2}{\vec{u}_i \cdot \vec{u}_i} \cdot \vec{u}_i$ and $\hat{u}_i = \frac{2 \cdot \vec{u}_i \cdot A_{i,1}}{\vec{u}_i \cdot \vec{u}_i} \cdot \vec{u}_i$ denote two scaled version of the vector \vec{u}_i , and for simplifications we will assume that both vectors are also part of the simplex description, then

$$S' = S - (\bar{u}_i \cdot S) \cdot \vec{u}_i + \hat{u}_i. \quad (2)$$

Recall that we assume that for k -dimensional simple, merging or splitting mirrors, \vec{u}_i also determines the direction in which the reflection of the mirror occurs. To determine whether the reflection of the point S is possible, i.e. it is located at the correct side of the hyperplane defined by the mirror, we have only to verify whether

$$\overline{S' S} \cdot \bar{u}_i > 0. \quad (3)$$

⁴Recall that a k -dimensional simplex is a generalisation of a triangle from 2 dimensions to k dimensions.

⁵Note that we do not use the normal, since it is usually scaled to have unit length and therefore the values of the coordinates of the normal might irrational number and cannot be written down by simple numbers in bounded time.

To get the new direction of the ray after the reflection, we can proceed as follows. First we determine an additional point on the ray, e.g. $H = S + \vec{r}$, then we determine the reflected position H' of H by using Equation 2. Finally the new direction \vec{r}' of the ray after the reflection will be

$$\vec{r}' = \overline{S'H'}. \quad (4)$$

In some applications it might be useful to determine also the intersection point P of the ray with the mirror. For this we have to find the values c_1, \dots, c_k, c_{k+1} such that

$$\sum_{j=1}^k c_j \cdot \overline{A_{i,1}A_{i,j}} - c_{k+1} \cdot \vec{r} = 0. \quad (5)$$

Then we get the intersection point

$$P = c_{k+1} \cdot \vec{r}. \quad (6)$$

To test whether the intersection point is within the mirror, we have to verify that

$$\forall j \in \{1, \dots, k\} : c_j \in [0..1]. \quad (7)$$

Within our algorithms we would like to determine the relative position of the source and the ray direction after a given sequence of reflection. Analysing this formula one can see that for every reflection the number of bits of the numerators and denominators increase by a value at most linear in the input size. Hence, if the number of reflections is polynomial in the input size, the values remain polynomial in the input size. If we replace the relative source positions by the intersection points we might double the length of the binary representation of the numerator and the denominator with each reflection, resulting in exponential large binary representations for a polynomial number of reflections. As a consequence, we should not iterate over the intersection points.

Given a point that is used as source S' and a ray direction \vec{r}' as well as an intersection point P on a ray we can, by using Formula 6 determine all the intersection point of the ray with any mirror of the system. Furthermore the values c_{k+1} of the Formula 6 act as a kind of distance between the source and the intersection points with the corresponding mirrors. If we sort mirrors according to these values then the list of the mirrors with higher distance value as P represent the potential future reflection points after P . Note that this list can be determined based on S' , \vec{r}' , and P deterministically in polynomial time. Based on these observations we can now present algorithms for solving the ray tracing problem for a bounded number of reflections. To solve the ray tracing problem one can trace the mirrored relative position of the source over a sequence of reflections.

Algorithm 1: Bounded Ray Tracing without Ray Duplication

input : initial direction of the ray \vec{r} , source S , the number of allowed reflections t , a destination T , and set of n mirrors given by n k -dimensional simplices $L_i = (A_{i,1}, \dots, A_{i,k+1}, \vec{u}_i, \bar{u}_i, \hat{u}_i)$ (either simple plain or merging mirrors)

output : accept, if there is a ray from S to T using at most t reflections

```

1 let  $P_0 = S, S_0 = S, \vec{r}_0 = \vec{r}$  and  $i = 0$ 
2 while  $i \leq t$  do
3   determine the sequence of simplices having an intersection with the line starting at  $S_i$  following the direction
   given by  $\vec{r}_i$  sorted by the distance from  $S_i$ 
4   delete from the list all mirrors with distance smaller or equal to the distance of  $P_i$ 
5   delete from this list all merging mirrors, where the ray does not go to the reflecting side of the mirror
6   let  $L_j$  denote the first mirror on this list
7   if the ray reaches the target without reflection then
8     | accept
9   if the ray does not reach  $L_j$  at the reflecting side or  $L_j$  does not exists then
10    | reject
11   determine the reflected source  $S_{i+1}$  based on  $L_j, S_i, \vec{r}_i$  (Equation 2)
12   determine the reflected ray direction  $\vec{r}_{i+1}$  based on  $L_j, S_i, S_{i+1}, \vec{r}_i$  (Equation 4)
13   determine the intersection point of the old ray  $P_{i+1}$  with  $L_j$  based on  $L_j, S_i, S_{i+1}, \vec{r}_i$  (Equation 6)
14   let  $i = i + 1$ 
15 reject
```

Analysing this formula one can see that for every reflection the number of bits of the numerators and denominators increase by a value at most linear in the input size. Hence, if the number of reflections is polynomial in the input size,

the values remain polynomial in the input size, and the ray tracing problem can be solved deterministically in polynomial time, if the sequence of used mirrors can be determined deterministically by following the ray (see Algorithm 1). Note that this is possible if we examine only simple and merging mirrors. In the case of one-way and splitting mirrors one has to distinguish between a reflection of a beam and the possibility that the beam might pass through the mirror. If we non-deterministically guess this behaviour (see Algorithm 2 Line 12), the described algorithm above can solve the problem in non-deterministic polynomial time. Hence we can conclude:

Theorem 1. *The $k + 1$ -dimensional ray tracing problem with rational input values and a polynomial number of reflections is in \mathcal{P} if we assume simple plain mirrors and plain merging mirrors. The problem is in \mathcal{NP} for a polynomial number of reflections and we allow simple plain mirrors, plain merging, plain splitting and one-way mirrors.*

For solving the illumination problem with a polynomial number of reflections, one can proceed analogously to the ray tracing problem. Note that even if we only consider simple plain mirrors, the sequence of used mirrors from the source to the target cannot be determined by following a ray through the collection of mirrors, since we might have to investigate a cone of infinitely many rays. But if we guess non-deterministically the sequence of used mirrors within a first phase, one can use similar algorithm to verify that the target will be illuminated by using the given number of reflections.

Algorithm 2: Bounded Ray Tracing with Ray Duplication

input : initial direction of the ray \vec{r} , source S , the number of allowed reflections t , a destination T , and set of n mirrors given by n k -dimensional simplices $L_i = (A_{i,1}, \dots, A_{i,k+1}, \vec{u}_i, \bar{u}_i, \hat{u}_i)$ (either simple plain, merging, splitting, or one-way mirrors)

output : accept, if there is a ray from S to T using at most t reflections

```

1 let  $P_0 = S, S_0 = S, \vec{r}_0 = \vec{r}$  and  $i = 0$ 
2 while  $i \leq t$  do
3   determine the sequence of simplices having an intersection with the line starting at  $S_i$  following the direction
   given by  $\vec{r}_i$  sorted by the distance from  $S_i$ 
4   delete from the list all mirrors with distance smaller or equal to the distance of  $P_i$ 
5   delete from this list all merging mirrors, where the ray does not go to the reflecting side of the mirror
6   delete from this list all mirrors after the first simple or merging mirror
7   let  $L_{j_1}, \dots, L_{j_\ell}$  denote the resulting list
8   if the ray reaches the target without reflection then
9     | accept
10  if the list is empty then
11    | reject
12  non-deterministically choose a mirror  $L_j$  from  $L_{j_1}, \dots, L_{j_\ell}$ 
13  if the ray does not reach from  $L_j$  a reflecting side then
14    | reject
15  determine the reflected source  $S_{i+1}$  based on  $L_j, S_i, \vec{r}_i$  (Equation 2)
16  determine the reflected ray direction  $\vec{r}_{i+1}$  based on  $L_j, S_i, S_{i+1}, \vec{r}_i$  (Equation 4)
17  determine the intersection point of the old ray  $P_{i+1}$  with  $L_j$  based on  $L_j, S_i, S_{i+1}, \vec{r}_i$  (Equation 6)
18  | let  $i = i + 1$ 
19 reject
```

We start with an algorithm (Algorithm 3) for simple plain double sided mirrors, which can simply be modified to work for all types of mirrors used in this work. The points which are determined within this algorithm are illustrated in Figure 9.

The correctness of the Algorithm 3 follows by Lemma 7.

Lemma 7. *Algorithm 3 accepts iff there exists a ray from S to T using at most t reflections.*

Proof. If there exists a ray from S to T using $\ell \leq t$ reflections then within Step 1 of Algorithm 3 will non-deterministically guess the corresponding sequence of mirror for these reflections. The corresponding sequence of positions $P_0, \dots, P_{\ell-1}$ represent the positions at the sequence of mirrors, where the ray will hit the mirrors. Since the initial ray will reach T starting from S , there will be no object on this path which will intersect the ray, except of the chosen mirrors where the ray will be reflected. Thus, our algorithm will accept the input.

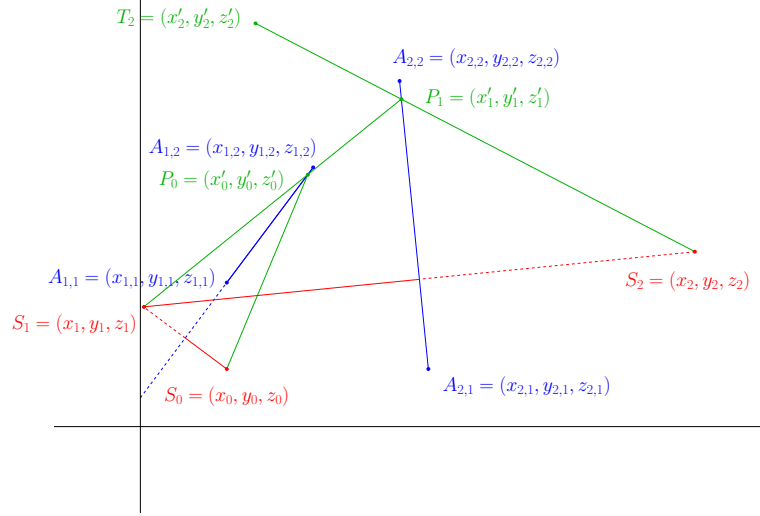


Figure 9: Illustration of the individual steps of Algorithm 3

Let us now assume that Algorithm 3 accepts an input. Then the chosen sequence of mirrors together with the sequence of intersection points $P_0, \dots, P_{\ell-1}$ describe a path of a ray from S to T with at most t reflections. \square

Algorithm 3: Bounded Illumination

input : light source S , the number of allowed reflections t , a destination T , and set of n mirrors given by n k -dimensional simplices $L_i = (A_{i,1}, \dots, A_{i,k+1}, \vec{u}_i, \bar{u}_i, \hat{u}_i)$

output : accept, if there is a ray from S to T using at most t reflections

- 1 non-deterministically guess the sequence of mirrors L_1, \dots, L_ℓ as the sequence of reflections on the path from S to T with $\ell \leq t$
 - 2 let $S_0 = S$ and $P_\ell = T$
 - 3 **for** $i = 1$ **to** ℓ **do**
 - 4 let S_i be the reflection of S_{i-1} at mirror L_i by using Equation 2
 - 5 **for** $i = \ell$ **to** 1 **do**
 - 6 let P_{i-1} be the point at L_i where the light source will be seen, i.e. P_{i-1} is the cut point of the line from S_{i-1} to P_i within L_i
 - 7 **if** P_{i-1} does not exist within L_i **then**
 - 8 reject
 - 9 **if** there exists a simplex L_j intersecting the line segment from P_{i-1} to P_i **then**
 - 10 reject
 - 11 **if** there exists a simplex L_j intersecting the line segment from S_0 to P_0 **then**
 - 12 reject
 - 13 **else**
 - 14 accept
-

If we determine the intermediate point using Formula 5 and 6 might double the number of bits required for the numerator and denominator within each iteration. To simplify the analysis of the required number of bits we will introduce a slightly improved version (Algorithm 4) of Algorithm 3.

Analogously to our observation for the ray tracing algorithms one can see that the binary representation of the mirrored sources S_1, \dots, S_ℓ and the mirrored targets $T_0, \dots, T_{\ell-1}$ require only a polynomial number of bits if the value ℓ is polynomial in the input length. The intermediate points $P_i \in \{P_0, \dots, P_{\ell-1}\}$ are computed based on L_i and the line $\overline{S_i T_i}$, hence they require only a polynomial number of bits, too. Moreover all the test within Algorithm 4 can be

Algorithm 4: Bounded Illumination II

input : light source S , the number of allowed reflections t , a destination T , and set of n mirrors given by n k -dimensional simplices $L_i = (A_{i,1}, \dots, A_{i,k+1}, \vec{u}_i, \bar{u}_i, \hat{u}_i)$

output : accept, if there is a ray from S to T using at most t reflections

- 1 non-deterministically guess the sequence of mirrors L_1, \dots, L_ℓ as the sequence of reflections on the path from S to T with $\ell \leq t$
- 2 let $S_0 = S$ and $T_\ell = T$
- 3 **for** $i = 1$ **to** ℓ **do**
- 4 let S_i be the reflection of S_{i-1} at mirror L_i by using Equation 2
- 5 **if** S_{i-1} is on the non-reflective side of L_i **then**
- 6 | reject
- 7 **for** $i = \ell - 1$ **down to** 0 **do**
- 8 let T_i be the reflection of T_{i+1} at mirror L_{i+1} by using Equation 2
- 9 **if** T_{i+1} is on the non-reflective side of L_{i+1} **then**
- 10 | reject
- 11 let $P_\ell = T$
- 12 **for** $i = \ell$ **to** 1 **do**
- 13 let P_{i-1} be the intersection point of L_i with the line $\overline{S_i T_i}$
- 14 **if** P_{i-1} does not exist within L_i **then**
- 15 | reject
- 16 **if** there exists a simplex L_j intersecting the line segment from P_{i-1} to P_i such that this simplex blocks the direct connection from P_{i-1} to P_i **then**
- 17 | reject
- 18 **if** there exists a simplex L_j intersecting the line segment from S_0 to P_0 such that this simplex blocks the direct connection from S_0 to P_0 **then**
- 19 | reject
- 20 **else**
- 21 | accept

performed analogously to corresponding test for the ray tracing problem. Hence, the algorithm requires only polynomial time in the input length and the value of the bound t . The point determined by Algorithm 4 are illustrated in Figure 10.

Finally one can show that the two points P_i (for all i) of both algorithms are equal. Hence also Algorithm 4 solves the illumination problem. Summarizing we get:

Theorem 2. *The $k + 1$ -dimensional illumination problem with rational input values and a polynomial number of reflections is in \mathcal{NP} if we allow simple plain mirrors, plain merging, plain splitting and one-way mirrors.*

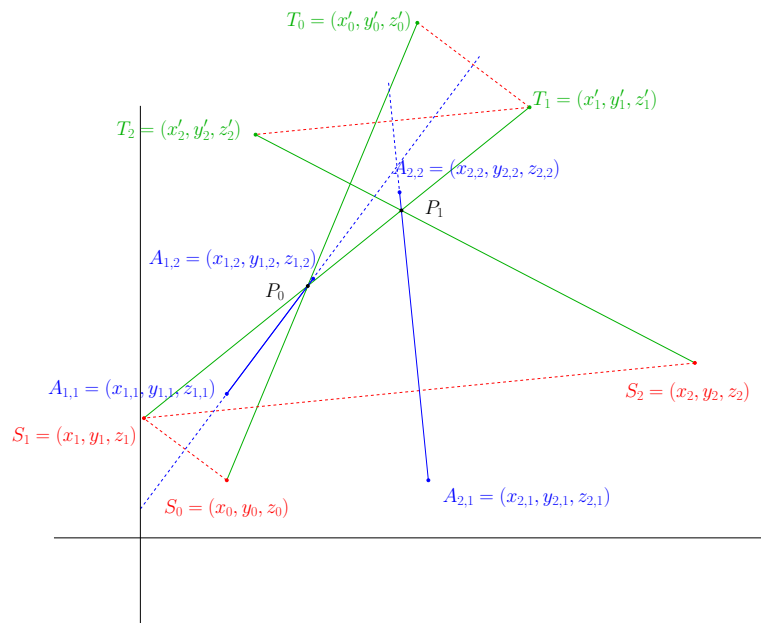


Figure 10: Illustration of the individual steps of Algorithm 4

A.2 Ray Tracing in 2D without Beam Duplication

In the following we would like to show how we can implement a binary stack by some kind of an offset of the position of the position of a single ray. For implementing pop and push we use systems of plain mirrors, merging mirrors, and parabola mirrors. Based on this constructions it is possible to evaluate a circuits for a given fixed input where modules of mirrors represent the gates of the circuit.

Before going into the details it should be mentioned that for every parabolic mirror given by the formula $a \cdot x^2 + b$ with $a, b \in \mathbb{R}$ and $a > 0$ every ray coming from above the mirror and going parallel to the x -axis which hits the mirror has to pass through the focus of the parabola given by $(0, \frac{1}{4a} + b)$, see e.g. Figure 11.

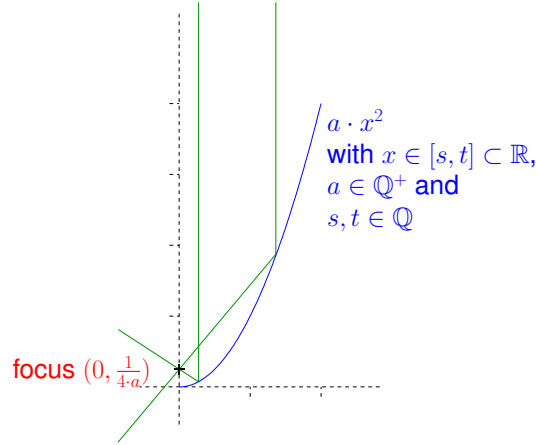


Figure 11: Focus of a parabolic mirror.

We assume that a ray without a offset passes a specific horizontal line at a specific position. The (horizontal) distance between this position and the actual position where the ray passes the line will be called offset. For the offset of the ray we assume in the following usually values between 0 and 1 (excluding the extreme values). Given an offset of $s \in (0, 1)$ we can implement a push of a bit $b \in \{0, 1\}$ by

$$s' = (s + b) \cdot 2^{-1}$$

and a pop by

$$s' = (2 \cdot s) - \lfloor 2 \cdot s \rfloor \quad \text{and determining} \quad b = \lfloor 2 \cdot s \rfloor$$

One can see that the multiplication by 2 or by 2^{-1} is a major tool for these operations. To implement this we use a combination of parabola mirrors. For the multiplication with 2^{-1} (see Figure 3)

$$-2 \cdot x^2 + \frac{3}{8} \quad \text{for } x \in [-1, 0] \quad \text{and} \quad x^2 \quad \text{for } x \in [0, 2].$$

For the multiplication with 2 (see Figure 12)

$$-\frac{1}{2} \cdot x^2 + \frac{3}{4} \quad \text{for } x \in [-4, 0] \quad \text{and} \quad x^2 \quad \text{for } x \in [0, 2].$$

In our figures we have used the possibility of having larger offsets. This is based on the requirement that we have to add to the offset in some scenarios a 1, especially if we would like to push a 1 to our stack, or we have to pop a 1 from the stack. In this cases the construction takes care about this larger offset, and either divides the value of the offset by 2 or splits the interval of possible offset values into to interval, each going from 0 to 1.

Note that after using one of these constructions the offset interval is reversed. If on the input the offset was assumed to move the ray to the right of the 0-line, it is at the output moving the ray to the right of the 0-line. To correct this we can use to extra parabolic mirrors (see Figure 13)

$$-x^2 + \frac{1}{2} \quad \text{for } x \in [-2, 0] \quad \text{and} \quad x^2 \quad \text{for } x \in [0, 2].$$

Using this modules we can now implement the rewired modules for the stack operations, see Figure 14, 15, and 16. For push 0 we simply divide the offset by 2. For push 1 we first shift the ray, implement an addition of 1 and afterwards

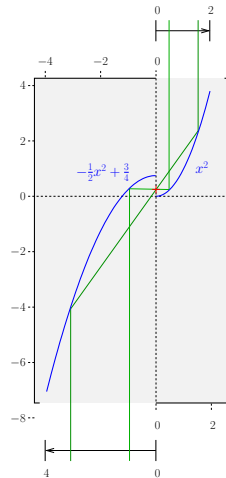


Figure 12: The example construction of multiplying the offset by 2.

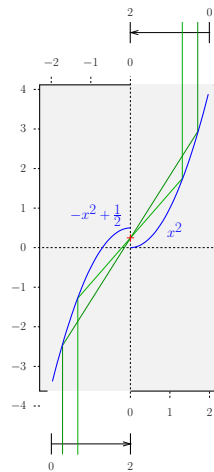


Figure 13: After performing an multiplication step the order of the offsets is reversed. This module reverses the order of the offset again, with out any additional multiplication.

divide the offset by 2. The pop module has two possible output intervals for the offset, the first is used if the pop results in a 0, the second if the pop results in a 1. To get these two different intervals we multiply the offset first with 2.

As we have seen in the construction of the pop operation it might be possible the offset interval is split into two intervals. After performing some interval dependent operation it might be possible, that the resulting offset intervals have to be joint again. To join the intervals merging mirrors are used as presented in Figure 17.

Using these modules we can now describe how a circuit can be evaluated by a system of mirrors. W.l.o.g. we assume that all gates of the circuit which we would like to evaluate have fanin and fanout at most 2, and that the gate types are restricted to input literals, constants, negations, identity gates (for increasing the number of successors which can use an intermediate result), AND and OR gates.

The main strategy for the circuit evaluation works as follows (see Figure 4):

1. Assume we would like to evaluate a gate u . Then the beam which implements the stack is within the mirror system of u and will be passed via specific mirrors as a result request to one of its predecessors v .
2. If u is the first successor of v we push a 0 to our stack, otherwise we push a 1. We assume that this value will not be removed from the stack by a mirror system of an other gate.
3. By merging mirrors we combine the possible offsets to a single interval of offsets $[0, 1]$.

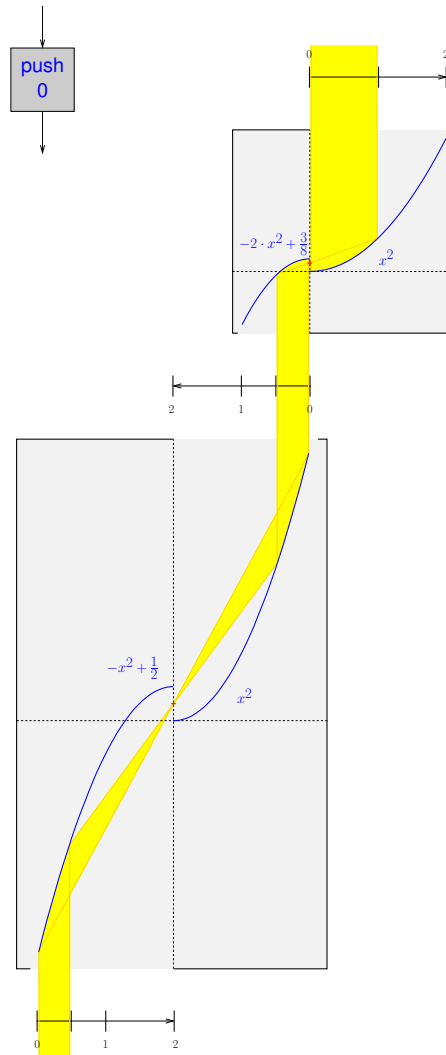


Figure 14: Multiplying the offset from the interval $[0, 1]$ by $\frac{1}{2}$ implements a push of the Boolean value 0 to the stack. The yellow area represents area of possible offsets.

4. We start to determine the value of v by sending a request to its first predecessor (if it exists). We assume that the top element of the stack is the result of this predecessor when the beam is coming back. If necessary we send an request also to the second predecessor after the beam has arrived from the first predecessor.
5. We remove the results from the predecessors by pop from the stack and depending on the predecessor results we send the the beam along a result depending paths through the later mirrors.
6. By a pop we can now determine the asking successor (i.e. u) of v , push the result of v to the stack, and route the beam to the requesting successor of v .

It remains to describe how the different gates functionalities can be implemented by a system of mirrors. The construction of the AND-gate is illustrated Figure 5, the OR-gate in Figure 18, the ID-gate in Figure 19, and the NOT-gate in Figure 20. For constants or inputs we can use the construction of Figure 21 to represent the value 0 and the construction of Figure 22 to represent the value 1.

This construction shows that the circuit value problem can be solved by system of simple plain mirrors, merging mirrors, and parabola mirrors. We conclude

Theorem 4. *The ray tracing problem in 2d for simple plain mirrors, merging mirrors, and parabola mirrors for an exponential number of reflections is \mathcal{P} -hard.*

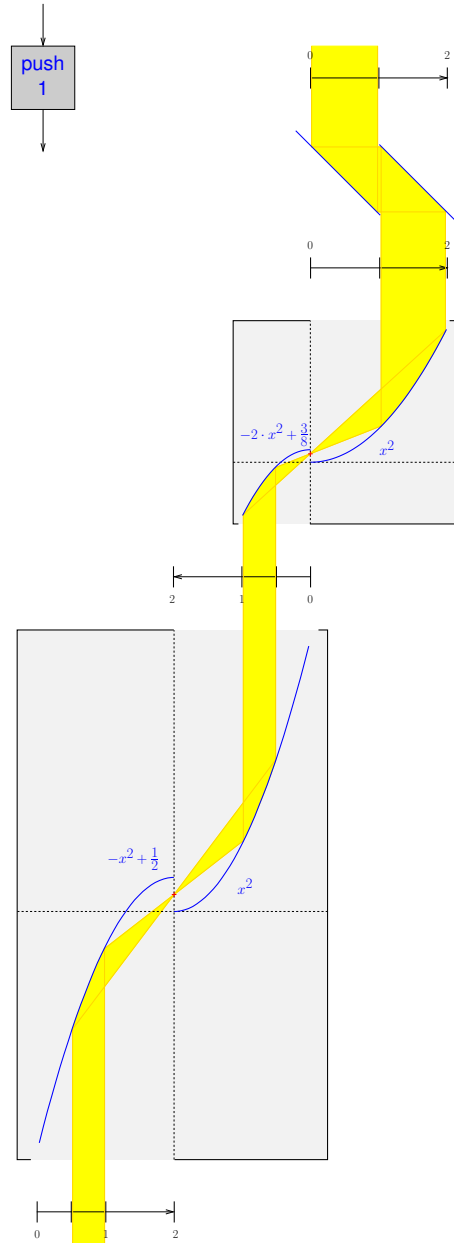


Figure 15: Shifting the offset from the interval $[0, 1]$ to the interval $[1, 2]$ and multiplying the resulting offset by $\frac{1}{2}$ implements a push of the Boolean value 1 to the stack. The yellow area represents area of possible offsets.

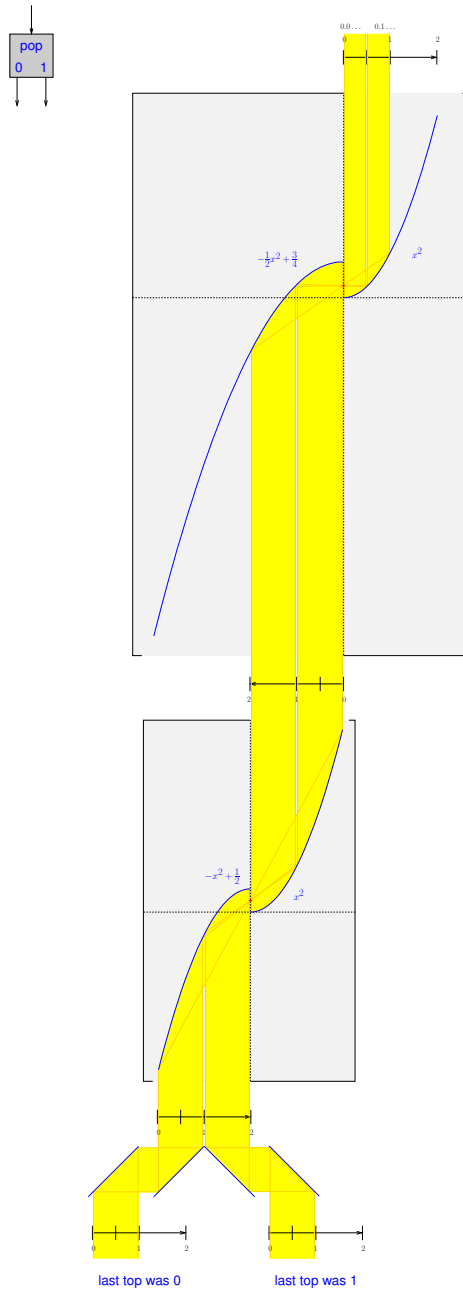


Figure 16: Multiply the offset by 2 is a basic part for implementing the pop operation. If the top entry of the stack in 0 the beam with modified offset (the top element is removed) will accrue on the left exit. In the case it is 1 the beam with modified offset will accrue on the right exit. If we assume that to bottom of the stack always contain at least two 1 digits, then our offset can never assume the values $0, \frac{1}{2},$ and 1. The yellow area represents area of possible offsets.

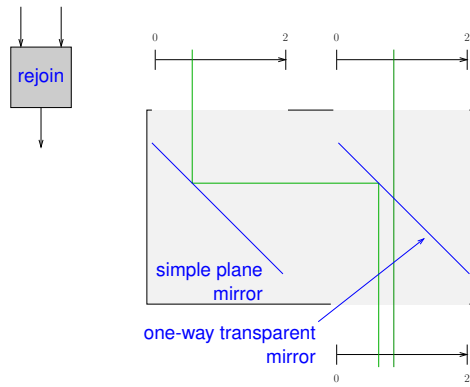


Figure 17: Based on the pop operation one offset interval is split into two intervals. To merge the intervals one can use the rejoin tool which consists of a simple plane mirror and a one-way transparent mirror.

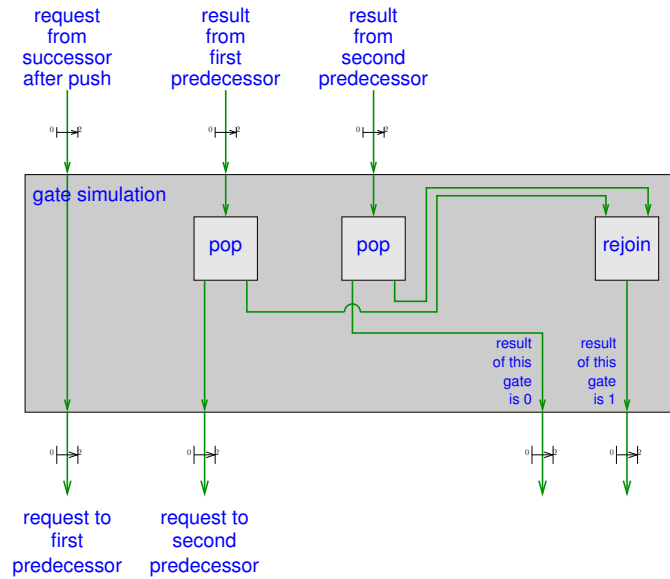


Figure 18: Mirror system of an OR gate.

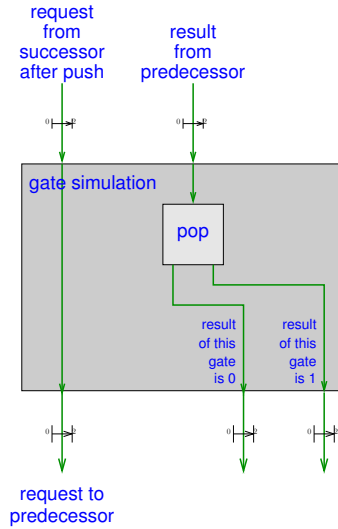


Figure 19: ID gate have usually only one predecessor. The main goal of such a gate is to serve requests from two successors.

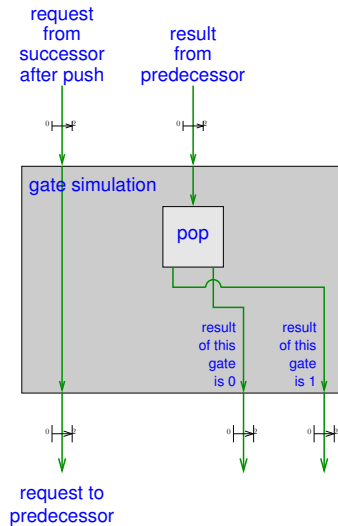


Figure 20: Like ID gates a NOT gate has only one predecessor. The main goal of such a gate after permuting the result intervals to serve requests from two successors.

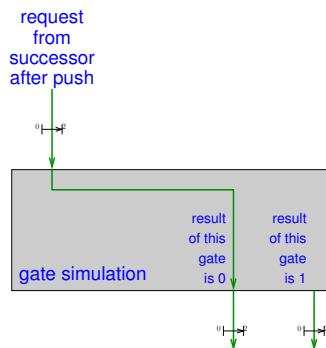


Figure 21: Gate simulation for constant gates and input gates with value 0.

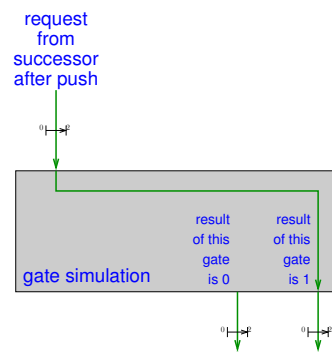


Figure 22: Gate simulation for constant gates and input gates with value 1.