In Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science, October 1990, pp. 106-114

The Computability and Complexity of Optical Beam Tracing

John H. Reif* J. D. Tygar† Akitoshi Yoshida*

Abstract

Consider optical systems consisting of a set of refractive or reflective surfaces. *The ray tracing problem* is, given an optical system and the position and direction of an initial light ray, to decide if a light ray reaches some given final position. We assume the position and the tangent of the incident angle of the initial light ray is rational. For many years, ray tracing has been used for designing and analyzing optical systems. Ray tracing is now used extensively in computer graphics to render scenes with complex curved objects.

We investigate the computability and complexity of the ray tracing problems over various optical models. Our results show that, depending on the optical model, ray tracing is sometimes undecidable, sometimes PSPACE-hard, and sometimes in PSPACE.

1 Introduction

We examine ray tracing problems in this paper. The history of ray tracing goes back at least to Archimedes, who examined images formed by a mirror to understand the law of reflections. In the 15th to 18th centuries, many scientists and astronomers in Europe worked on geometrical optics and invented optical instruments such as telescopes. In 1730, Newton published his book *Opticks* [17] in which he formally defined the reflective and refractive laws of optics, and first defined and investigated some ray tracing problems. These classical ray tracing problems are very important to the design of most optical systems which consist of a set of refractive or reflective surfaces, and involve tracing the path of rays to investigate the performance of the systems. Ray tracing also has important application in computer graphics, where ray tracing is used to render pictures which consist of objects with surfaces that reflect or refract light rays [9, 10, 19]. See [2, 3, 12, 13, 14] for physical theories and a brief history of ray tracing.

The *ray tracing problem* is a decision problem: given an optical system (namely, a finite set of reflective or refractive surfaces) and an initial position and direction of a light ray and some fixed point p, does the light ray eventually reach the point p? Our optical systems consist of a finite set of optical objects that may be totally reflective (we call these *mirrors*), partially reflective (we call these *half-silvered mirrors*), or totally refractive (we call these *lenses*). In this paper we restrict ourselves to optical systems constructed out of flat (e.g., planar sections) mirrors and half-silvered mirrors (throughout this paper, we assume that mirrors are rectangles, and we call their corners the mirrors' *endpoints*.) and out of lenses whose boundaries are quadratic curves. (We describe these lenses by the term *quadratic lenses*.) Do mirrors reflect if a light-beam is directed exactly at an endpoint? This might be important for the case when we form a corner out of two mirrors. What should happen when the light beam is directed exactly at the corner? For the purposes of our papers, we shall allow mirrors (and half-silvered mirrors) to have either open, closed, or mixed borders.

The positions of our mirrors, half-silvered mirrors, and lenses can be either *rational* or *irrational*. If the optical system consists only of mirrors or half-silvered mirrors with endpoints with rational coordinates, we say that the optical system is *rational*. If the optical system contains mirror or half-silvered mirrors with endpoints that have irrational coordinates, we say the optical system is *irrational*.

We are interested in determining if the light will reach a final certain position, and not in the intensity of the light at that position. Throughout this paper, we assume that the path taken by light rays are determined by the classical laws of optics: *the law of reflection* and *the law of refraction*. (The law of reflection states that the incident angle and the reflected angle are equal, and the law of refraction states that the angle of refraction depends on the incident angle and the index of refraction of the materials.) (Figure 1.)

We always assume that the initial position of the light ray has rational coordinates, that the tangent of the initial incident angle is rational, and that the test point p has rational coordinates. (In general, in our lower bound proofs, it suffices to let the light rays initially enter perpendicular to a window of the optical systems.) Our surprising discovery is that if the optical system is rational it may have high complexity, or even be undecidable. We generally denote n to be the number of bits in binary encoding of the optical system.

Our results of the computational complexity for ray tracing in various optical systems may be summarized as follows:

*Department of Computer Science, Duke University, Durham, NC 27706. Email: reif@cs.duke.edu and ay@cs.duke.edu. Research supported in part by Air Force contract AFOSR-87-0386, DARPA/ARO contract DAAL03-88-K-0195, DARPA/ISTO contract N00014-88-K-0458, and NASA subcontract 550-63 of contract NAS5-30428

†School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. Email: tygar@cs.cmu.edu. Research supported in part by a National Science Foundation Presidential Young Investigator Award, contract CCR-8858087.

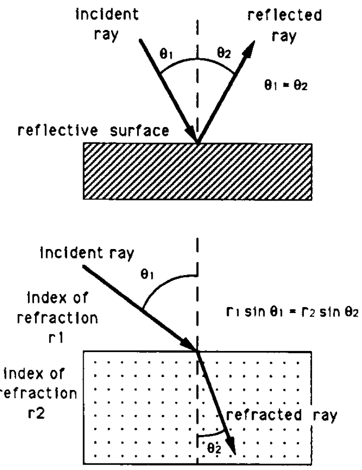

Figure 1: Law of reflection and law of refraction with indices of refraction r_1 and r_2.

1. Ray tracing in three dimensional optical systems which consist of a finite set of mirrors, half-silvered mirrors, and quadratic lenses is undecidable, even if the endpoints of the objects in the optical system all have rational coordinates. However, the problem is recursively enumerable.

2. Ray tracing in three dimensional optical systems which consist of a finite set of mirrors is undecidable, if the mirrors' endpoints are allowed to have irrational coordinates. However, the ray tracing problem is PSPACE-hard, if we restrict ourselves to mirrors with endpoints that are rational coordinates.

3. For any $d \geq 2$, ray tracing of d dimensional optical systems which consist of a finite set of mirrors surfaces lies in PSPACE, if the positions of all the surfaces are rational, and they lie perpendicular to each other. For $d \geq 3$, the problem is PSPACE-complete.

We consider three optical models in this paper:

In optical model (1), each optical system consists of a finite set of quadratic lenses, mirrors, and half-silvered mirrors. A light ray travels through the system with reflections or refractions. We show that the problem of deciding if the light ray will reach a given final position in this system is undecidable. In order to show this, we simulate some universal Turing machine with this optical model. It is perhaps surprising that our optical system has a fixed number of optical lenses and mirrors, and yet the ray tracing problem for it simulates any recursive enumerable computation, where the input is given by the initial position of the light ray.

In optical model (2), each optical system consists of a finite set of mirrors and half-silvered mirrors in three dimensional space. We again show that the problem of deciding is undecidable. To show this, we simulate a 2-counter machine with this optical model. Next, we consider the computational complexity when we restrict ourselves to rational optical systems. In this case, we show that the problem is PSPACE-hard. To show this, we first define a new type of automaton: augmented bounded 2-counter machines. Then, we simulate augmented bounded 2-counter machines with an optical system in this model. By showing augmented bounded 2-counter machines can compute an arbitrary polynomial space problems, we conclude that the problem of deciding if the light ray reach a given final position in this system is in PSPACE-hard. (Although we show that the problem is PSPACE-hard, we do not even know if this restricted problem is decidable.)

Optical model (3) is a restriction of optical model (2). In optical model (3), each optical system occurs in a unit-sized d dimensional hypercube. The hypercube contains a rational optical system of mirrors. Each of the mirrors lies perpendicular to every other mirror. We show that the problem of deciding if the light ray will reach a given final position has a non-deterministic polynomial space algorithm, thus showing the problem is in PSPACE.

Theoretically, these optical systems can be viewed as general optical computing machines, if our constructions can be carried out with infinite precision, or perfect accuracy. However, these systems may not be practical, since the above assumption may not hold in physical world. Moreover, at the atomic level of distances, the diffraction and interference properties which yield the classical optical laws begin to fail [5]. However, in contemporary computer science practice, only the classical laws are considered. Our work has immediate relevance for describing the complexity and computability of ray tracing as currently practiced.

Abstract ray tracing problems have been independently investigated by Fiume [6]. Fiume's work does not describe actual optical systems, but rather considers symbolic systems which transform intensity of rays without any consideration of the geometry of the optical systems. He showed that his abstract ray tracing problem was PSPACE-complete, if the amplification of light intensity was allowed. If the amplification is not allowed, then he showed the problem became NP-hard. However, his transformations on the intensity of optical rays would require electro-optical devices or photorefractive crystals, which are not purely optical devices. In contrast, our results do not depend on any intensity of the rays. Instead, we encode problems by the position of optical beams. Manipulation of the positions is carried out by use of pure geometrical optics. Thus, our models give results more appropriate to computational geometry. In particular, they answer the ray tracing problems in the models which are described by Newton in his *Opticks,* and which form the backbone of modern ray-tracing theory.

In the following sections (2.1, 2.2, 2.3), we describe the computability and complexity of ray tracing problems in optical models (1), (2), and (3). In section (2.4), we discuss an interesting phenomenon which leads to the conclusion that there exists an optical system such that if a ray enters from a single fixed point, it visits an edge of unit length densely and uniformly. In section 3, we give some interesting and challenging further open problems concerning optical systems.

2 Optical Models

2.1 Three dimensional optical systems with curved reflective or refractive surfaces

Optical System

In our first optical model, we assume that each optical system consists of a finite set of quadratic lenses, mirrors, and half-silvered mirrors at rational positions. Here, all the surfaces are represented by rational quadratic equations. A light ray travels through the system with reflections or refractions. The initial position of the light ray has rational coordinates and the tangent of the initial incident angle is rational. (Note every reflection and refraction takes place at a rational position. Also, the angle of reflected or refracted light rays have rational tangents.)

The ray tracing problem is recursively enumerable

The total path which the light ray travels can be partitioned into simple straight subpaths (i.e., line segments) between reflective or refractive surfaces. We assume both the position and the tangent of the angle of the light ray at the endpoint of a subpath are rational. Then, by the laws of reflection and refraction, the tangent of an angle which the subsequent subpath makes is rational, and also the end points of the subsequent subpath are rational. Therefore, they can be represented by rational equations, and hence by induction, the total path consisting of a finite number of subpaths can be traced, to see if it ever reaches the specified target final point.

Turing machines

We show that the problem of deciding if the light ray will reach a given final position in this system is undecidable. We show how to simulate any Turing machine as an optical system. In particular, this implies that some optical system simulates some universal Turing machine. The basic model of a Turing machine [11] has a finite control, a tape which contains cells and input, and a tape head that reads one cell on the tape at a time. The tape can be infinite in both directions.

Our Turing tape simulation will assume the tape always contains only 0's and 1's. We assume that the set of states in the finite control is $Q = \{q_1, q_2, \ldots, q_s\}$; that δ is the transition function; that q_1 is the initial state; and that q_s is the (only) final state.

Simulation

First, we show the relation between this optical system and the Turing machine. We view this optical system as a set of complex optical boxes, each of which has a set of basic boxes with mirrors, half-silvered mirrors, and quadratic lenses. Each complex box has a unit square through which the light ray enters (always entering perpendicular to this surface), and one or two unit squares from which the light ray exits (always exiting perpendicular from this surface). These unit squares are called the *input windows* and the *output windows*. Our Turing tape is encoded by the (x, y) coordinates of the light ray relative to the input and output windows. We organize these complex boxes so that the whole system simulates a Turing machine.

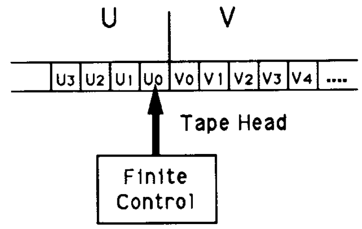

Figure 2: A Turing machine and two real numbers $U = 0.u_0u_1\ldots$ and $V = 0.v_0v_1\ldots$ which represent the tape value.

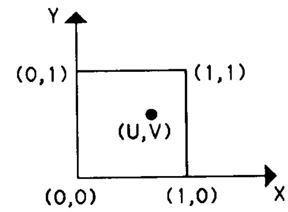

Figure 3: The representation of (U, V) in a unit square.

Each complex box will correspond to one state of the Turing machine's finite control. Each complex box implements the transition function on the tape defined for a particular state; and the light beam exits out of one of the two exit windows depending on which state the Turming machine will enter next. The system then projects the light ray onto the next complex box (preserving the coordinates of the light beam relative to the window), and thus simulates the transition of states defined on the Turing machine. This is the general idea of how to simulate a Turing machine by an optical system in this optical model. Next, we describe this idea in more detail.

Representation of Operations

We represent the storage tape of M by using two binary fractions U, and V. Let u_0 be the symbol which the tape head is scanning. Let $u_1, u_2, u_3, \ldots$ be the successive symbols on the left of u_0, and $v_0, v_1, v_2, \ldots$ be the successive symbols on the right of u_0. This is shown in Figure 2. Then, we can represent the storage tape by using two numbers U, and V:

$$U = \sum_{i=0}^{\infty} u_i/2^{i+1} \tag{1}$$

$$V = \sum_{i=0}^{\infty} v_i/2^{i+1} \tag{2}$$

U and V form coordinates on a unit square, since $0 \leq U \leq 1$, and $0 \leq V \leq 1$. (Figure 3.)

Next, we consider the mapping from the transition function δ of M into the transition operation of this optical system. Turing machine transitions can be divided into two cases, *left moves*, and *right moves*.

We consider two cases of δ:
Case 1:

$$\delta(q, c) = (q', w, L)$$

Case 2:

$$\delta(q, c) = (q', w, R)$$

Here, q is the current state, q' is the next state, $c \in \{0, 1\}$ is the symbol which the tape head scanned, and $w \in \{0, 1\}$ is the symbol which the tape head writes on the tape. L represents a left move, and R represents a right move. Consider the corresponding operations of this optical system.

Case 1: (Left move) $\delta(q, c) = (q', w, L)$

Let U_{new}, V_{new} be the values of U and V respectively after this transition. Then, U_{new}, V_{new} can be written as:

$$U_{new} = 2\sum_{i=1}^{\infty} u_i/2^{i+1} = 2(U - u_0/2) \quad (3)$$

$$\begin{aligned} V_{new} &= w/2 + (1/2)\sum_{i=0}^{\infty} v_i/2^{i+1} \\ &= w/2 + V/2 \quad (4) \end{aligned}$$

Case 2: (Right move) $\delta(q, c) = (q', w, R)$

In this case, U_{new}, V_{new} can be written as:

$$\begin{aligned} U_{new} &= v_0/2 + w/4 + (1/2)\sum_{i=1}^{\infty} u_i/2^{i+1} \\ &= v_0/2 + w/4 + (U - u_0/2)/2 \quad (5) \end{aligned}$$

$$V_{new} = 2\sum_{i=1}^{\infty} v_i/2^{i+1} = 2(V - v_0/2) \quad (6)$$

If we can implement complex optical boxes which can "read" the light beam entering the window, perform the transformations listed above, and then redirect the light beam to a new complex optical box corresponding to the new state, we will have succeeded in simulating our Turing machine. Next, we describe how to build these complex boxes.

Basic Boxes

First, we describe basic boxes which can be used by each complex box.

- Readout box
  We assume U, and V are represented as a position on a unit square lying on the x-y plane, and we use the z axis which is normal to this plane. The readout box uses two flat mirrors (reflective surfaces) making an angle $\pi/2$ at $x = 1/2$. (Figure 4.)

  The figure shows several views from different angles. The light ray entering a box from the unit square hits one of two flat reflective surfaces in the box. We call the unit square which the light enters *input window*, and a window of size 1 by 1/2 from which the light exits the box *output window*. Then, if $U < 1/2$, the ray will be reflected through the left output window. If $U \geq 1/2$, the ray will be reflected through the right output window. Here, we observe that V maintains its value along the y axis from the entrance window to one of the output windows. U loses its value along the x axis, but the value $U - u_0/2$ is obtained along the z axis at the output window.

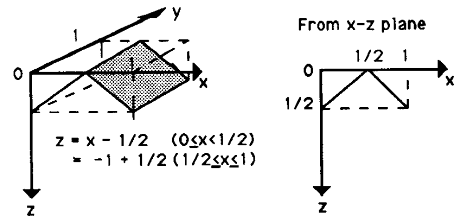

Figure 4: Readout box.

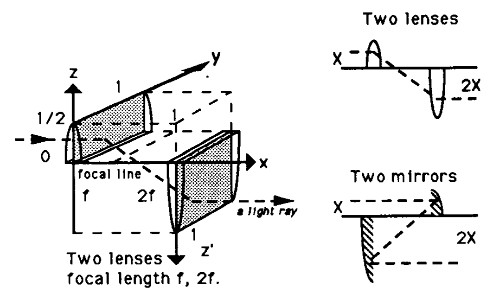

Figure 5: Multiply2 (divide2) box.

- Multiply2 (Divide2) box
  We need a box which performs the multiplication by two (division by two) operations used in the equations. This can be done by using a pair of cylindrical quadratic lenses (refractive surfaces) placed with rational endpoints. (Figure 5.)

  Similarly, we use terms, *input window* and *output window* to denote the window through which the light enters, and the window from which the light exits.

- Beam splitter (mixer) box
  We need this box in order to allow light rays to travel through one of several possible paths to enter the next entrance window. This box is merely a half-silvered mirror oriented at a $\pi/4$ angle.

- Beam turner box
  We need this box in order to change the direction of the light ray by $\pi/2$. This box is merely a mirror oriented at a $\pi/4$ angle.

We combine these basic optical boxes to construct complex boxes which implements transition function δ. Here, we retain the above notation.

Case 1: (Left move) $\delta(q, c) = (q', w, L)$
We must configure boxes to project the light ray at U_{new}, V_{new}:

$$U_{new} = 2(U - u_0/2)$$

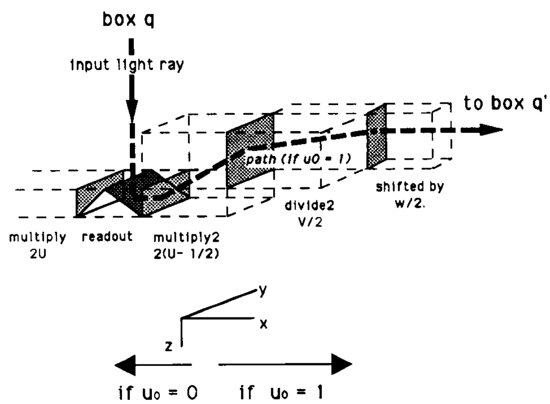

Figure 6: $\delta(q,c) = (q', w, L)$.

$$V_{new} = w/2 + V/2$$

Figure 6 shows an implementation of this function. For each state q, a separate complex of boxes, such as shown in Figure 6, is constructed. The box in the middle is a readout box which reads u_0. Now, suppose $u_0 = 1$, then the light enters the multiply2 box on right. At this point, V has not changed its value, but the light ray comes out of the face along the $z - y$ plane, and the z coordinate of the intersection is the value for U_{new}. The light ray which exits the multiply2 box enters the divide2 box. This halves the value of V. The light ray which exits the divide2 box enters the readout box of state q' ($c = 1$) with shifting by the value $w/2$. The path which would be taken if $u_0 = 0$ can be similarly organized, and the light ray can enter the readout box of state q' ($c = 0$).

Case 2: (Right move) $\delta(q,c) = (q', w, R)$
In this case, we must project the light at U_{new}, V_{new}:

$$U_{new} = v_0/2 + w/4 + (U - u_0/2)/2$$
$$V_{new} = 2(V - v_0/2)$$

Figure 7 shows an implementation of this transition.

Undecidability Proof

We have shown our system and its relation to the Turing machine. By simulating each δ function as a complex box and routing the rays between the boxes (using half-silvered mirrors to merge paths), we can simulate any Turing machine, and in particular, some universal Turing machine. We immediately have

Theorem 2.1 *The ray tracing of 3-D optical systems consisting of quadratic lenses, half-silvered mirrors, and mirrors is undecidable, even if the endpoints of every object in the optical system has rational coordinates.*

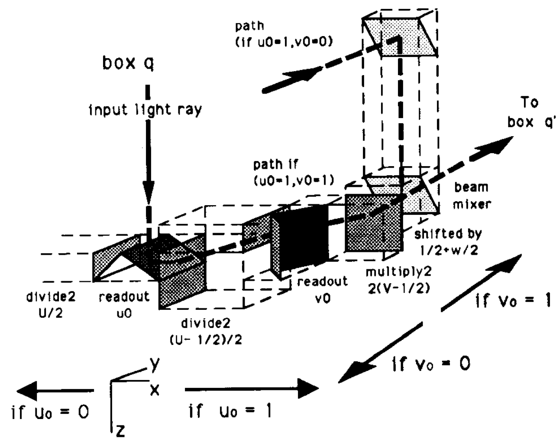

Figure 7: $\delta(q,c) = (q', w, R)$.

2.2 3-D optical systems with flat reflective or refractive surfaces

2.2.1 The general case

Optical System

In this model, each optical system consists of a finite set of mirrors and half-silvered mirrors in three dimensional space. We shall show this system is undecidable by using mirrors with irrational coordinates as endpoints.

2-counter Machine

In order to show that the problem of deciding in this model is undecidable, we will show that an optical system can simulate a 2-counter machine. Since a 2-counter machine can simulate an arbitrary Turing machine, we can conclude that the previous problem is undecidable.[11] A 2-counter machine has a finite control, and two counters which can assume arbitrarily large values. The counters can be incremented, decremented, or tested for zero.

Simulation

First, we show the relation between this optical system and the 2-counter machine. We view this optical system as a set of optical boxes which shift light rays by an irrational distance along either the x or y axis by using flat reflective surfaces. Each box has a unit square through which the light ray enters normally, and has a unit square from which the light ray exits normally. Similarly, we call the unit square which the light ray enters the *input window*, and the unit square from which the light exits the *output window*. The values of the counters (m, m') can be encoded as a position $(m\xi \bmod 1, m'\xi \bmod 1)$ relative to the unit square where ξ is some irrational value. The 2-counter machine has a finite number of states, and each state can be represented by a box. Both incrementing and decrementing a counter can be

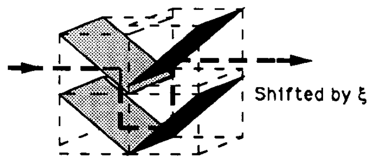

Figure 8: Shift operation.

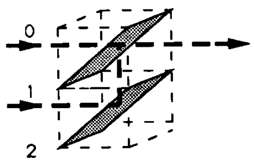

Figure 9: Modulo 1 operation.

done in this optical system by shifting the light ray by ξ modulo 1. Checking if a count is zero can be done by testing to see if a light beam strikes the rim of a unit square.

Representation of Operations

Use U and V to denote the values of two counters. In this optical system, U and V are encoded as a position on a unit square. Let ξ be an irrational number such that $0 < \xi < 1$. The operations for a 2-counter machine are:

Operation 1: Increment (or decrement) a counter by 1.

Operation 2: Check if a counter is 0.

Now, we describe the corresponding operations for this optical system.

Operation 1: Shift the light ray along the x axis by ξ (or $-\xi$) in modulo 1 space. The modulo 1 operation is simple, and is implemented by flat reflective surfaces. These operations can be done by using flat reflective surfaces as in Figures 8 and 9.

Operation 2: In order to check if the U counter is zero, we check whether the light ray passing through $x = 0$ from $x > 0$. (We are using the fact that we have mirrors with open edges here.) The case for the V counter is handled similarly.

Since $m\xi = 0$ if and only if $m = 0$, we have

Lemma 2.1 *This optical counter can simulate a counter.*

Optical Boxes

First, we describe optical boxes which are used in this system.

- Shift box
  This box shifts light rays by an irrational distance ξ. The shift box uses four flat mirrors as shown in Figure 8.

- Modulo box
  The modulo operation uses a mirror and a half-silvered mirror, as in Figure 9.

Since a two-counter machines can simulate universal Turing machines, we have:

Theorem 2.2 *Ray tracing of 3-D optical systems consisting of half-silvered mirrors and mirrors is undecidable.*

2.2.2 The rational case

We now restrict ourselves to rational optical systems. We shall show that this system is PSPACE-hard, although we still do not know if it is decidable.

Augmented Bounded 2-Counter Machine

First, we define a new machine type:

Definition 2.1 *A 2^n augmented bounded 2-counter machine has two counters which count up to $2^n - 1$. Furthermore, it can add 2^{n-1} to a counter, and it can read the 2^{n-1} bit of its counters.*

We will simulate a 2^n augmented bounded 2-counter machine with a rational optical system.

Representation of Operations

We describe the operations which we need to simulate a 2^n augmented bounded 2-counter machine. Counting operations are handled as before, only ξ is set to the rational value 2^{-n}. We use U, and V to denote the values of two counters. Let ξ be a rational number such that $\xi = 2^{-n}$. We consider n to be the input size, since the optical system can be described in a polynomial in n number of bits.

Once again, we represent the counters (m, m') as $(m \bmod 1, m' \bmod 1)$.

Lemma 2.2 *This optical counter can count up to $2^n - 1$.*

Proof: Since ξ is a rational number, for any integer m we have $m\xi \bmod 1 = 0$ if and only if $m = k2^n$, where k is an integer. Hence the counter can count up to $2^n - 1$.

Next, we introduce the additional operations.

- Add 2^{n-1} to a counter
  This can be done by shifting the light ray along the x or y axis by $1/2$.

- Read the 2^{n-1} bit
  We do this by using the "readout box" introduced in Section 2.1.

Using these boxes, we can build an optical system that simulates a 2^n augmented bounded 2-counter machine.

Decidability

Lemma 2.3 *Augmented bounded 2-counter machines can decide any PSPACE problem.*

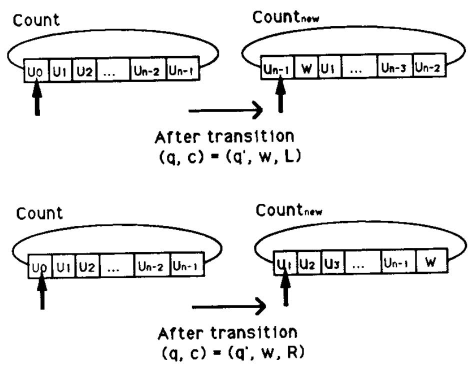

Figure 10: Cyclic Turing machine tapes.

Proof: It is suffice to show that the 2^n augmented bounded 2-counter machine can simulate a space n Turing machine. Any linear space Turing machine has a storage tape of length n, where n is the input size. We show that the 2^n bounded 2-counter machine can simulate this Turing machine. We consider the Turing tape to be cyclic. (Figure 10.) We can view the storage tape as a binary number, where the 1's digit lies under the tape head. Let *Count* be the binary representation of the storage tape. Then, *Count* satisfies $0 \leq Count < 2^n$. One of the counters of the 2^n augmented bounded 2-counter machine stores this value, *Count*. Each time when the tape head moves left or right, we must compute the new *Count* which represents the storage tape at the next step. In order to compute the new *Count* for a right move, we first drop the 2^{n-1} bit off the input (if it is set), then we multiply by two (using the second counter) and then we add one if the original counter had the 2^{n-1} bit set. For a left move, we first divide *Count* by 2 (using the second counter), and then add 2^{n-1} if 1 is to be written on the tape.

The simulation is straightforward, completing the proof.

We immediately have

Theorem 2.3 *Ray tracing of 3-D rational optical systems is PSPACE-hard.*

2.3 *d* dimensional optical systems with perpendicular surfaces

Optical System

In this model, we consider rational optical systems where the mirrors lie perpendicular to each other in *d* dimensions.

Decidability Proof

We show that the problem of deciding if the light will reach a certain position in this system is in PSPACE. First, we show the problem is in PSPACE if the angle which the initial light ray makes with the system is $\pi/4$.

Lemma 2.4 *The problem of deciding if the light ray will reach a certain position is in PSPACE, if the angle which the initial light ray makes with the system is* $\pi/4$.

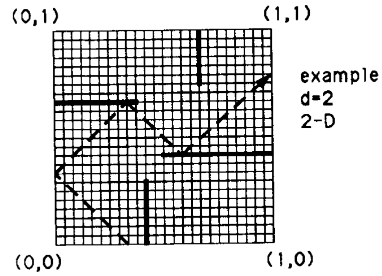

Figure 11: Grid lines.

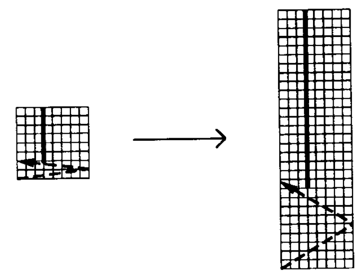

Figure 12: Expanding grid lines.

Proof: Since the input to the optical system is encoded in n bits, we can construct grid lines at every 2^{-n^h} interval such that the light ray always intersects grid lines at the grid points, where h is a constant. (Figure 11.) The number of the grid points in the system is 2^{dn^h}. We give a non-deterministic polynomial space algorithm for the ray tracing problem. Let M_n be a non-deterministic polynomial space Turing machine which compute this problem. M_n has two vectors. Let *CUR*, and *NEXT* be these two vectors. *CUR* represents the current position and the direction of the light ray in the system. *NEXT* represents the position and the direction where the light ray will be reflected after leaving the current position. M_n non-deterministically guess *NEXT* by using *CUR*, and determines the right reflection position in a polynomial space. Initially, *CUR* stores the position and the direction of the initial light ray. We assume inductively that *CUR* stores the current position and the direction of the light ray. M_n halts if it reaches the specified position. Since these vectors representing the position and the direction of the light ray can be encoded in a polynomial of n, we can construct such a non-deterministic polynomial space Turing machine to compute this problem. By Savitch's theorem, the problem is in PSPACE.

Next, we consider the case in which the incident angle α is not $\pi/4$. We simply give a sketch of the proof. We simply note that the system can be reduced to the $\pi/4$ case by multiplying one of the axes by the tangent of α. (See Figure 12.) The proof proceeds similarly.

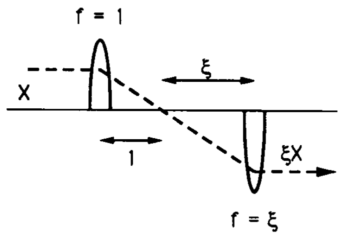

Figure 13: Multiplication by ξ.

We hence have

Theorem 2.4 *The ray tracing of d dimensional rational optical systems which consist of a finite set of perpendicular mirrors is in PSPACE.*

2.4 Other Optical Ray Tracing Phenomena

By using linear iterative maps which generate ∞-distributed sequences [16], we show that an optical system exists such that if a ray enters from a single fixed point, it visits an edge of unit length at every point densely and uniformly in the range $[0, 1)$.

First, we describe an ∞-distributed sequence which is dense and uniform in the range $[0, 1)$.

Lemma 2.5 $\{\xi^i \bmod 1 \mid i \text{ integer } \geq 0\}$ *is dense and uniform in the range* $[0, 1)$ *(∞-distributed sequence in the range* $[0, 1)$*) for almost all real numbers* $\xi > 1$.

Proof: The proof of this lemma is found in [7, 16].

This leads to

Theorem 2.5 *An optical system exists such that if a ray enters from a single fixed point, the set of points it visits on an edge of unit length is dense and uniform over the range* $[0, 1)$.

Proof: The idea is to construct an optical box which performs the multiplication by an irrational number ξ which satisfies Lemma 2.5. The implementation of modulo operations is already introduced in Section 2.2. To perform the multiplication by an irrational number ξ, we use two lenses, where one of the lenses has a focal length of one, and the other has an irrational focal length ξ. Then, by placing them with the distance $1 + \xi$, we can construct an optical box which implements the multiplication by ξ. (Figure 13.)

We can extend this type of problem by considering polygons whose sides form mirrors. We can then pose the *illumination problem*: to find an initial position and orientation which would cause a light beam to visit all the interior edges of the polygon densely? These problems are often called *billiard ball* problems, and the literature contains a number of "art gallery theorems," about these models. One particular result [1, 15, 18] states that for every polygon whose angles are rational multiples of π, the illimunation problem has a solution. We ask the following extension of the illumination problem: can the points on the polygon be visited densely and uniformly?

3 Conclusion and Further Open Problems

Our paper has classified a wide degree of ray tracing scenarios and given lower bounds for many of the problems. Unlike previous approaches (such as [6]) our work considers only geometric constructs. Our models give interesting results in terms of pure computational geometry.

There are many further interesting problems which remain open. Here are a few:

1. Find a lower bound for computations of ray tracing in *two* dimensional rational optical systems. Note *three* dimensional case is PSPACE-hard, but we have no lower bound in *two* dimensional case.

2. Are optical systems with only reflective flat surfaces always decidable?

References

[1] Boldrighini, C., M. Keane, and F. Marchetti, "Billiards in polygons," *Ann. Prob.* **6**, (1978) 532–540.

[2] Born, M., and E. Wolf, *Principles of optics.* Pergamon Press, Oxford, 1980.

[3] Driscoll, W., and W. Vaughan, *Handbook of optics.* McGraw-Hill, New York, 1978.

[4] Feitelson, D., *Optical computing: A survey for computer scientists,* MIT Press, Cambridge, 1988.

[5] Feynman, R., *QED: The strange theory of light and matter,* Princeton University Press, Princeton, 1985.

[6] Fiume, E., *The mathematical structure of raster graphics,* Academic Press, New York, 1989.

[7] Franklin, J., "Deterministic simulation of random processes", *Math. Comp.*, **17**, (1963) pp. 28–59.

[8] Glassner, A., ed., *Introduction to ray tracing,* Academic Press, New York, 1989.

[9] Greenberg, D., "Light reflection models for computer graphics," *Science,* **244** (1989), 166–173.

[10] Hanrahan, P., "Ray tracing algebraic surfaces," *Computer Graphics*, **17**, (1983) 83–90.

[11] Hopcroft, J., and J. Ullman, *Introduction to automata theory, languages, and computation,* Addison-Wesley, Reading, 1979.

[12] Horner, J., ed, *Optical signal processing,* Academic Press, New York, 1987.

[13] Iizuka, K., *Engineering optics,*, 2nd ed., Springer-Verlag, New York, 1987 .

[14] Klein, M., and T. Furtak, *Optics,* John Wiley & Sons, New York, 1986.

[15] Kerckhoff, S., H. Masur, and J. Smillie, "A rational billiard flow is uniquely ergodic in almost every direction," *Bull. AMS* **13** (1985) 141–142.

[16] Knuth, D., *Seminumerical algorithms,* 2nd ed., Addison-Wesley, Reading, 1981, 142–168.

[17] Newton, I., *Opticks,* 4th ed., London, 1730.

[18] O'Rourke, J., *Art gallery theorems and algorithms,* Oxford University Press, Oxford, 1987.

[19] Pokorny, C., and C. Gerald, *Computer graphics,* Franklin, Beedle & Associates, New York 1989.