Computational Geometry in Minecraft: Integer Block Ray Shooting Problem

ANEP Research

May 30, 2020

Contents

1 Introduction

Definition 1. ***Block*** *in block space* $\mathbb{R}^3$ *is a integer polytope cube and length of each edges is* 1*.*

Definition 2. ***Block space*** $\mathcal{B}(w, h, r)$ *is a* $w \times h \times r$ *integer polytope cube in euclidean space* $\mathbb{R}^3$*.*

Problem 1. ***Integer block ray shooting problem*** *in block space* $\mathcal{B}(w, h, r)$ *is decide if there exists a block* $B \in \mathcal{B}(w, h, r)$ *which intersects in* $intB$ *by a vector* $\vec{l} = (x, \frac{h}{w}x, \frac{r}{w}x))$*.*

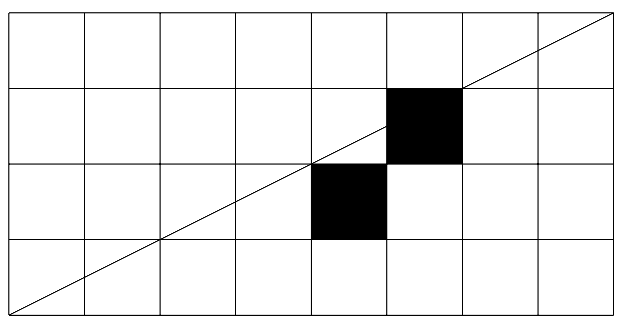
We shall introduce an efficient algorithm to solve problem 1.

2 Efficient algorithm: IBRS Problem

Algorithm 1 IBRS Problem in $O(n)$ time

1: **procedure** IBRS($\mathcal{B}(w,h,r)$) ▷ A block space $\mathcal{B}(w,h,r)$
2:     **for** $i \leftarrow 1$ to w **do**
3:         $\vec{l_i} \leftarrow (i, \lfloor \frac{h}{w} i \rfloor, \lfloor \frac{r}{w} i \rfloor)$
4:         **if** CheckBlock($\vec{l_i}$) **then** ▷ Procedure CheckBlock($\vec{v}$) is checking if there is an block at position $\vec{v} = (x,y,z)$ to $(x+1, y+1, z+1)$ in the block space $\mathcal{B}$
5:             **return** $false$
6:         **end if**
7:     **end for**
8:     **for** $j \leftarrow 1$ to h **do**
9:         $\vec{l_j} \leftarrow (\lfloor \frac{w}{h} j \rfloor, j, \lfloor \frac{r}{h} j \rfloor)$
10:         **if** CheckBlock($\vec{l_j}$) **then**
11:             **return** $false$
12:         **end if**
13:     **end for**
14:     **return** true
15: **end procedure**

For example, we consider the 2D block space.



And we may process algorithm 1 in this 2D block space. For the first loop in line 2, there is nothing happended until $i = 5$. We consider $i = 5$, $\vec{l_i} = (5, 0, 2)$ and the result of $CheckBlock(\vec{l_i})$ is *true*. Thus, the result of algorithm is $false$.

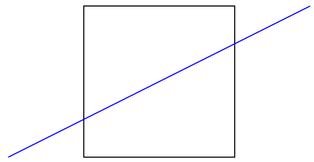
And we are going to proof the correctness of algorithm 1.

Lemma 1. *$(i, \frac{h}{w}i, \frac{r}{w}i)$ and $(\frac{w}{h}j, j, \frac{r}{h}j)$ are same straight line in $\mathbb{R}^3$.*

Proof. If $i = 0$ then $(i, \frac{h}{w}i, \frac{r}{w}i) = (0,0,0)$ and $i = w$ then (w,h,r). And if $j = 0$ then $(\frac{w}{h}j, j, \frac{r}{h}j) = (0,0,0)$ and $j = h$ then (w,h,r). Thus both straight line pass two points. Since first axiom of euclidean geomtry, Both straight lines are same. □

Theorem 1 (Correctness of algorithm 1)**.** *An algorithm 1 correctly decides problem 1.*

Proof. Firstly, We consider lines 2-7 of algorithm 1.



If a line and a square meet as shown in the photo, the point where it first meets is (x_0, y_0, z_0), then it's trivial the point left-bottom vertex of square is $(x_0, \lfloor y_0 \rfloor, \lfloor z_0 \rfloor)$. And it is $(x, \lfloor \frac{h}{w} x \rfloor, \lfloor \frac{r}{w} x \rfloor)$. When $i = x$ in the loop lines 2-7, Since line 4, It checks whether there is block which is intersected by a line. That is, If $\vec{l_i}$ implies left-bottom vertex of square, A line intersects with a square. If it is not true, which is contradicting to first axiom of euclidean geometry. And lines 8-13 is the case when y-coordinate is always integer.
Since lemma 1, It considers same straight line in lines 2-7 and lines 8-13. It can also be proved in the same way.
Then it is all possible cases about the position of block in a block space $\mathcal{B}$.
Thus algorithm 1 correctly decides problem 1. $\square$

Theorem 2 (Complexity of algorithm 1)**.** *An algorithm 1 has $O(\max(w, h))$ time complexity and $O(1)$ space complexity.*

Proof. Straightfoward. $\square$