1 Introduction

2 Background

3 Traversal Methods

Algorithm 1 Ordered Depth First Traversal

```
1:  procedure ODF_Traverse(b, r)
2:      s ← {b}
3:      c ← ∞
4:      while s ≠ ∅ do
5:          n ← pop(s)
6:          if n ∈ branches(b) then
7:              if intersect(r, n.left) < c then
8:                  push(s, n.left)
9:              end if
10:             if intersect(r, n.right) < c then
11:                 push(s, n.right)
12:             end if
13:         else if n ∈ leaves(b) then
14:             if intersect(r, n.prim) < c then
15:                 t ← n.prim
16:                 c ← intersect(r, t)
17:             end if
18:         end if
19:     end while
20:     if c = ∞ then
21:         return null
22:     end if
23:     return t
24: end procedure
```

Algorithm 2 Ray Order Traversal

```
1:  procedure RayOrder_Traverse(b, r)
2:      q ← {}
3:      push(q, n.left, intersect(r, n.left))
4:      while q ≠ ∅ do
5:          n ← pop(q)
6:          if n ∈ branches(b) then
7:              push(q, n.left, intersect(r, n.left))
8:              push(q, n.right, intersect(r, n.right))
9:          else if n ∈ leaves(b) then
10:             if intersect(r, n.prim) ≠ ∞ then
11:                 push(q, n.prim, intersect(r, n.prim))
12:             end if
13:         else if n ∈ primitives(b) then
14:             return n
15:         end if
16:     end while
17:     return null
18: end procedure
```

4 Experimental Setup

5 Results

6 Conclusion and Future Work

Figure 1: Not bad

Sphere	Plane	Hemisphere	Tube
image1	image2	image3	image4