

Penrose World

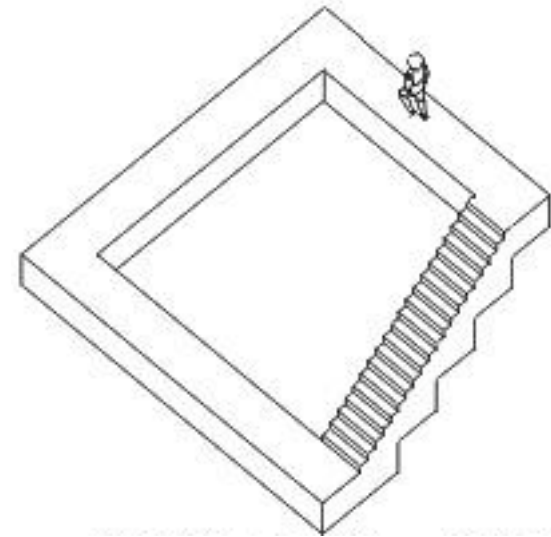
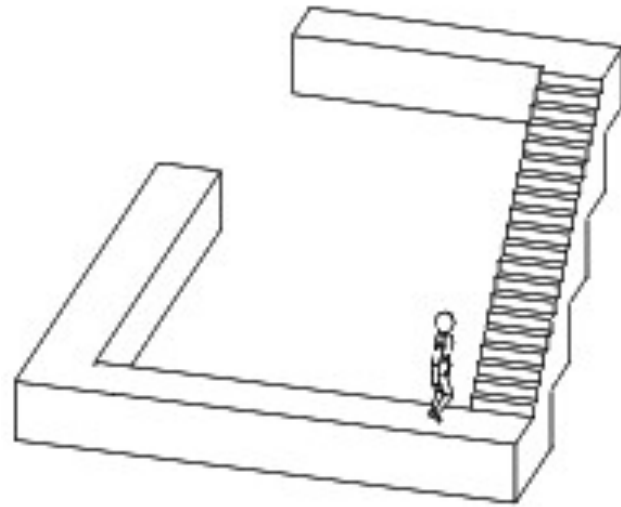
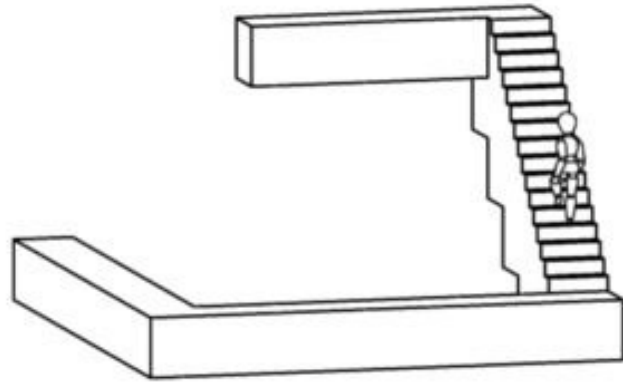
W4840 Embedded System Design
Final Project

Cong ZHU

Yuanhui LUO

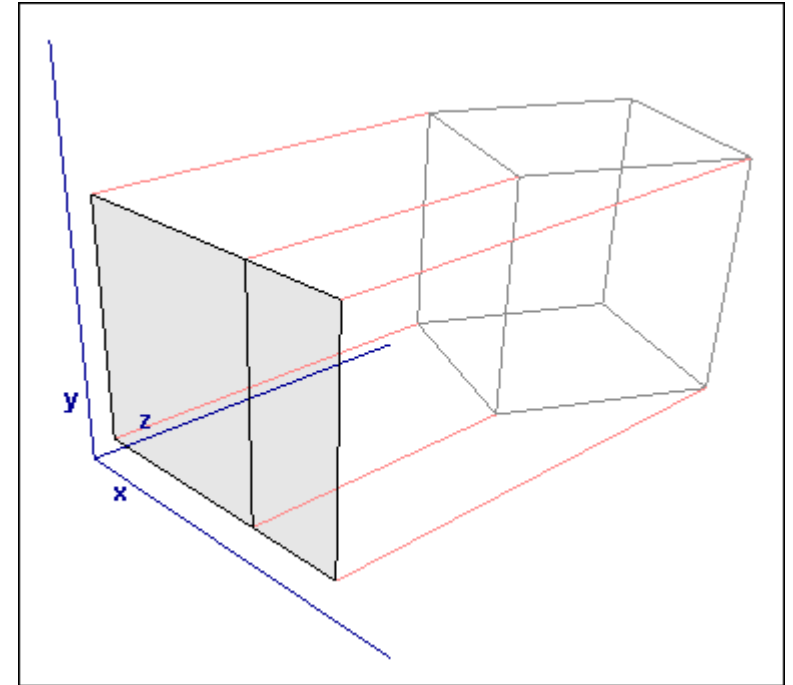
Yao LUO

Basic idea



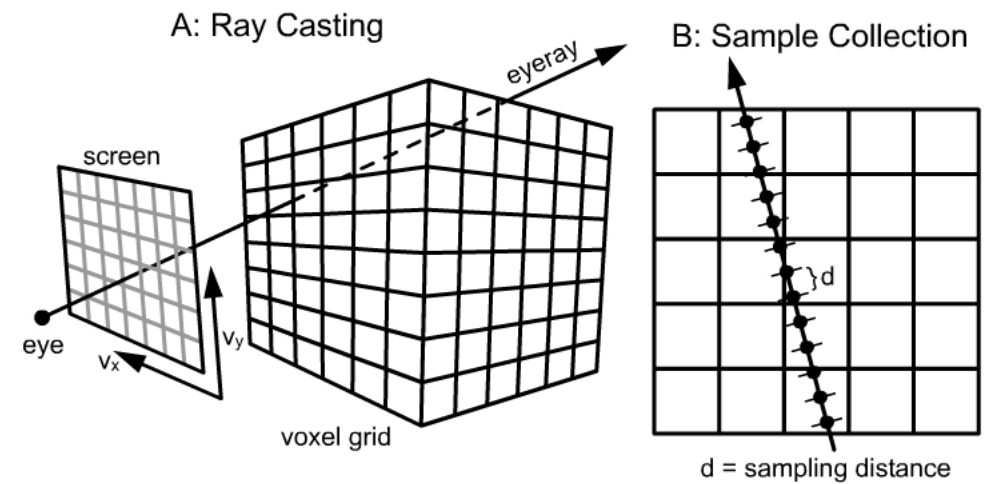
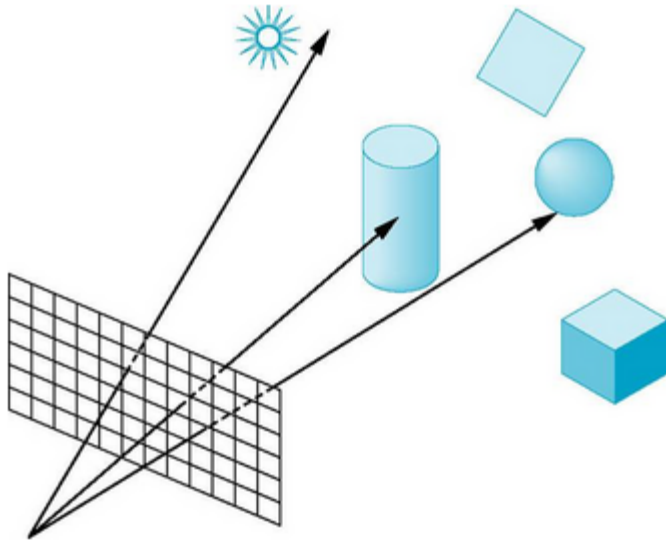
Ray-Casting Algorithm

- Basic module
- Rotation Calculation



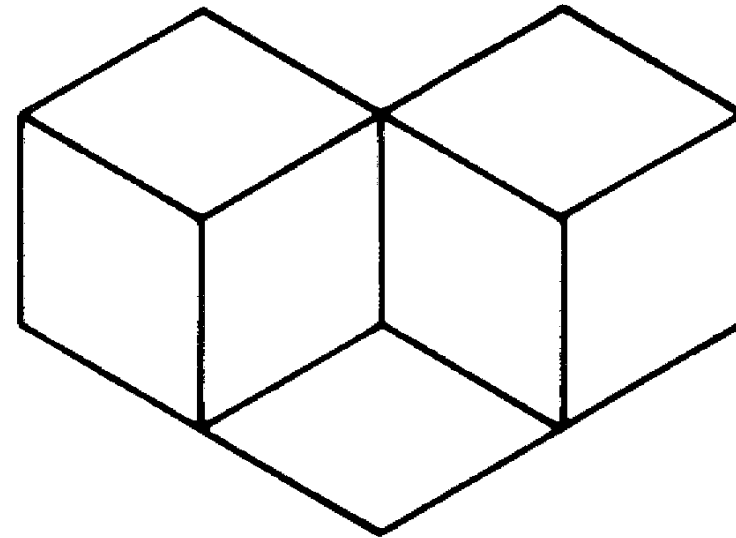
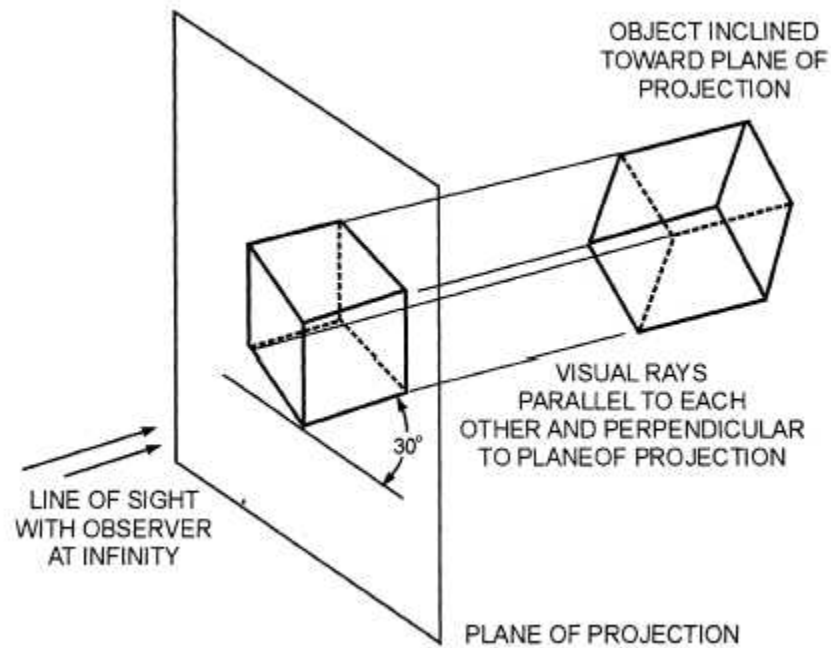
Ray-Casting Algorithm

- Original Version



Ray-Casting Algorithm

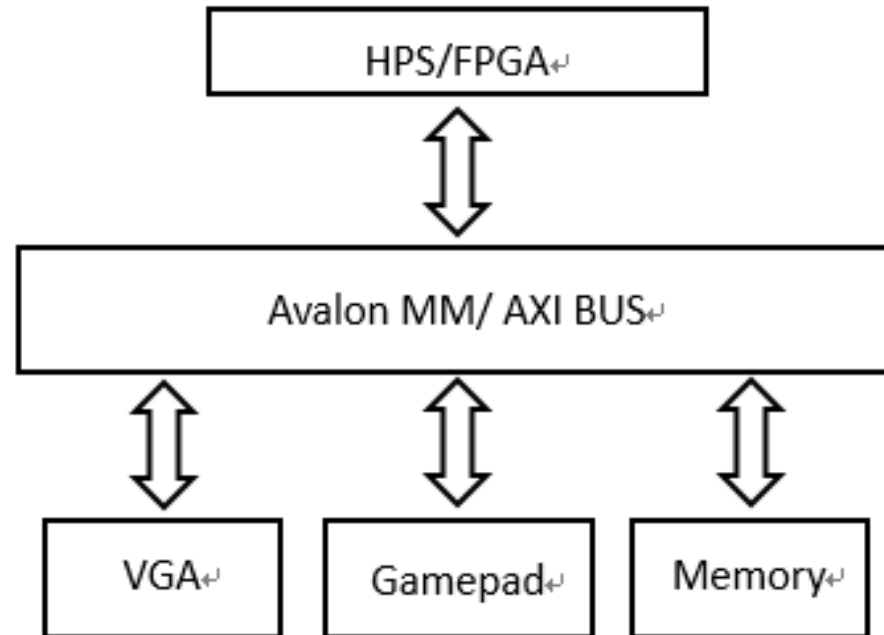
- Revised Version



Ray-Casting Algorithm

- Projection onto the Screen

System Architecture



HSP/FPGA

- HPS: 925 MHz, dual-core ARM[®] Cortex[™]-A9 MPCore[™] processor
- FPGA: Cyclone V SoC 5CSXFC6D6F31 Device
- Interface: ARM AMBA[®] AXITM bus bridges
- Usage: Use C-generated codes to control FPGA and other peripherals such as Gamepad.

VGA

- Doing projection of the object
- Plot 2D image of the Objects with respect of the location of 3D objects.

Gamepad

- Control the rotation angle when playing.
- Control the location of cubes when forming map
- Other control functions (speed up, fast turn, etc)

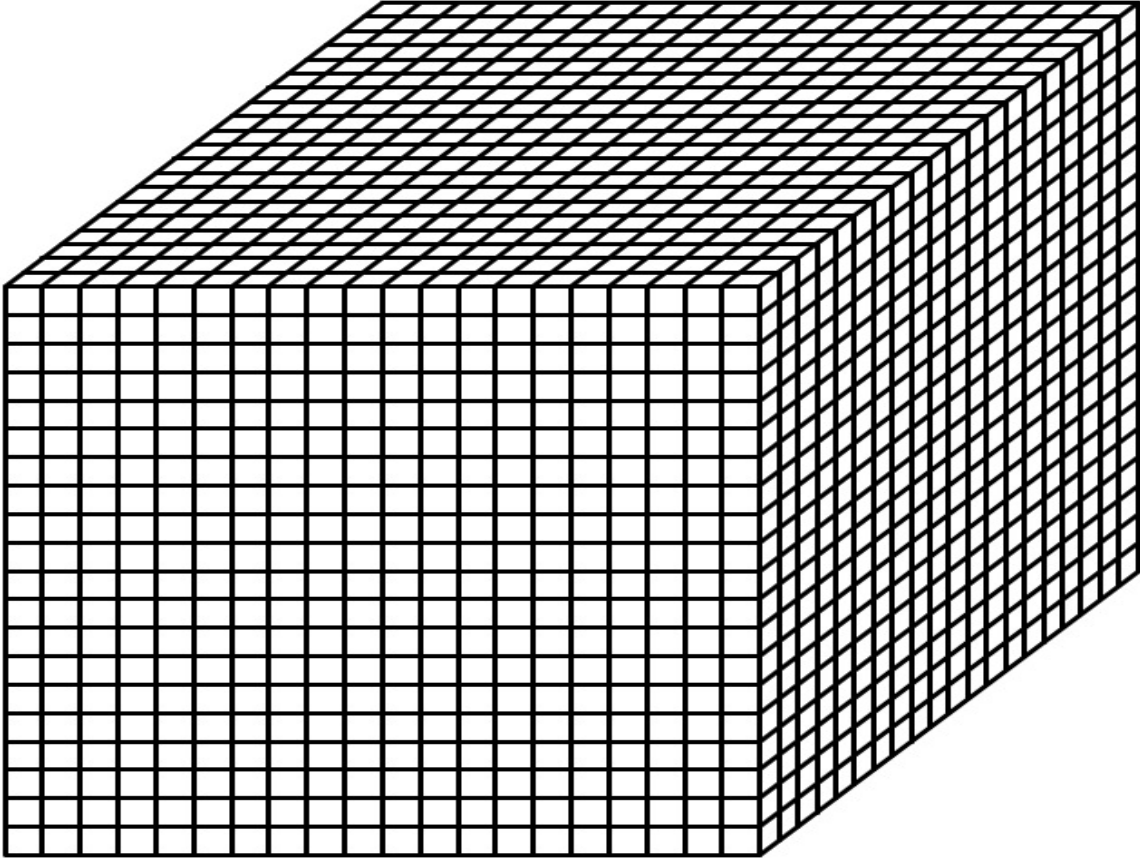
Memory

- Store Sin/Cos look up table
- Buffer projection information of cubes
- Store images used in the game

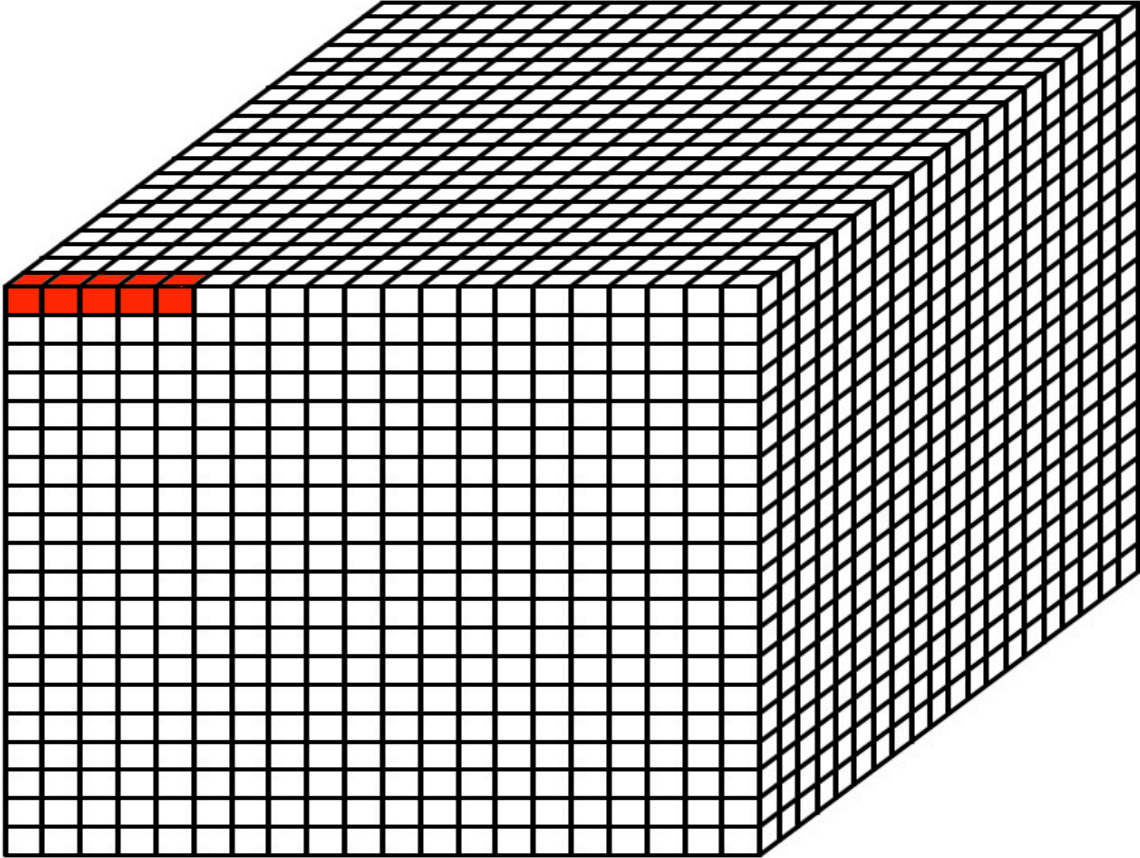
Hardware implementation

- Acceleration by using parallel casting
- Tradeoff between resolution and refresh rate
- Ping pang frame buffer
- Separate one object into four part

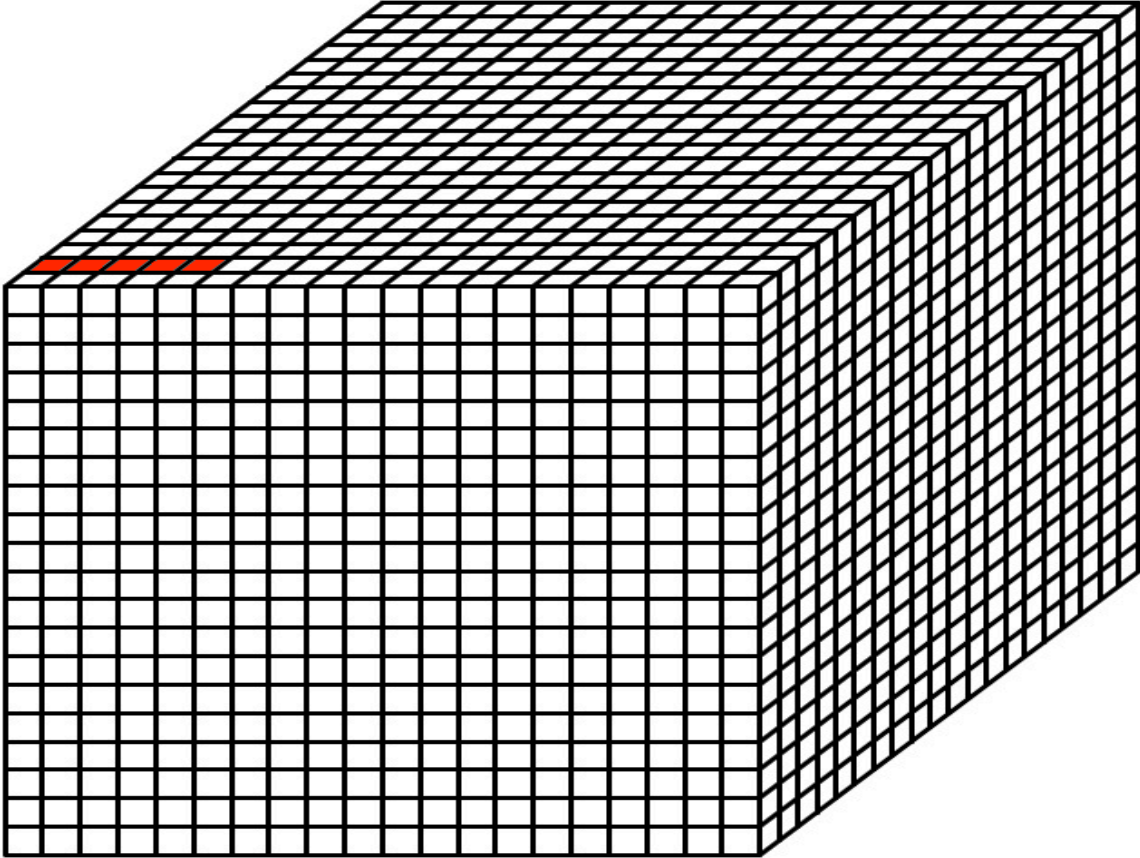
Acceleration by using parallel casting



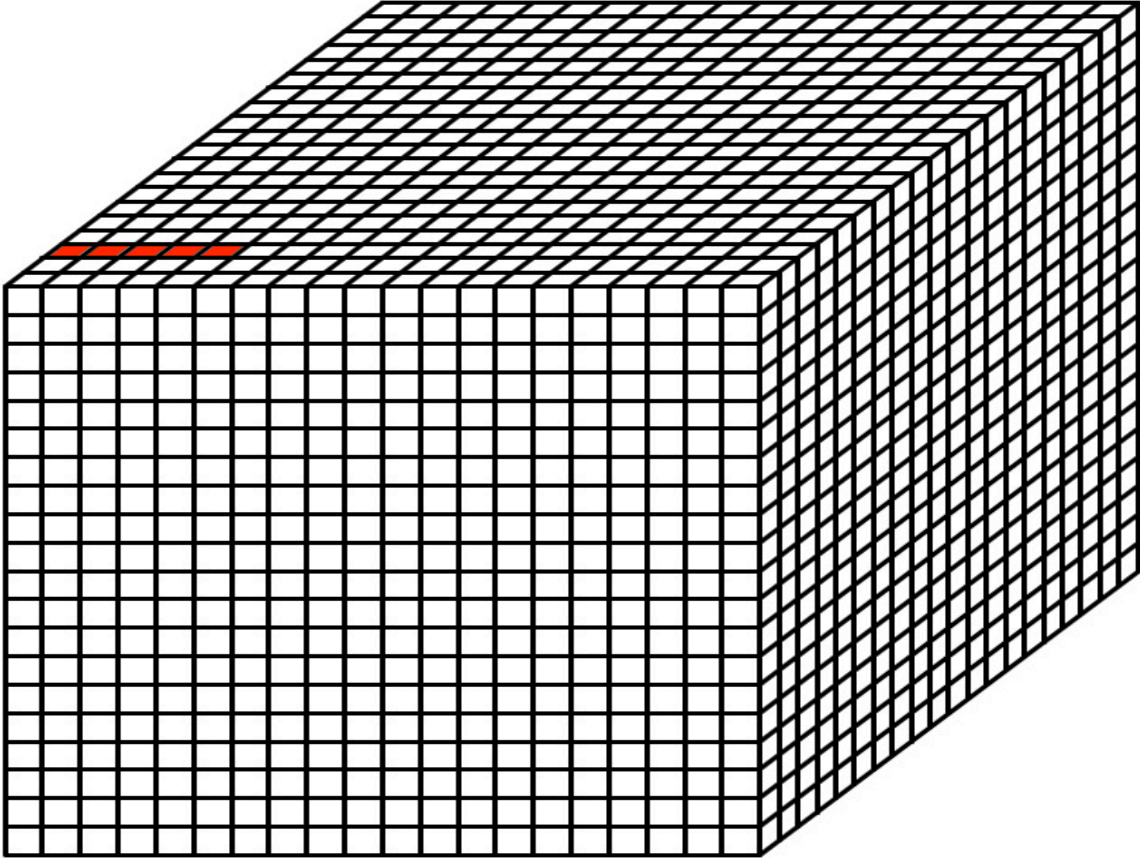
Acceleration by using parallel casting



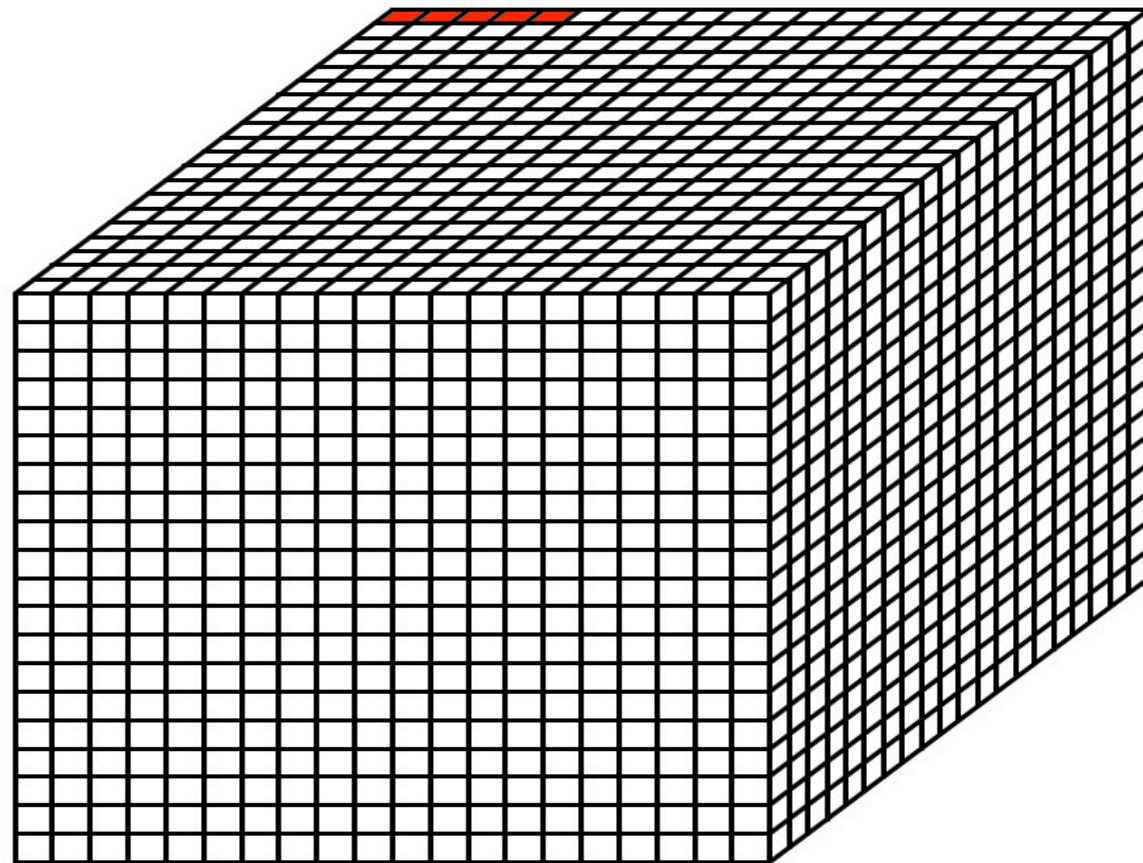
Acceleration by using parallel casting



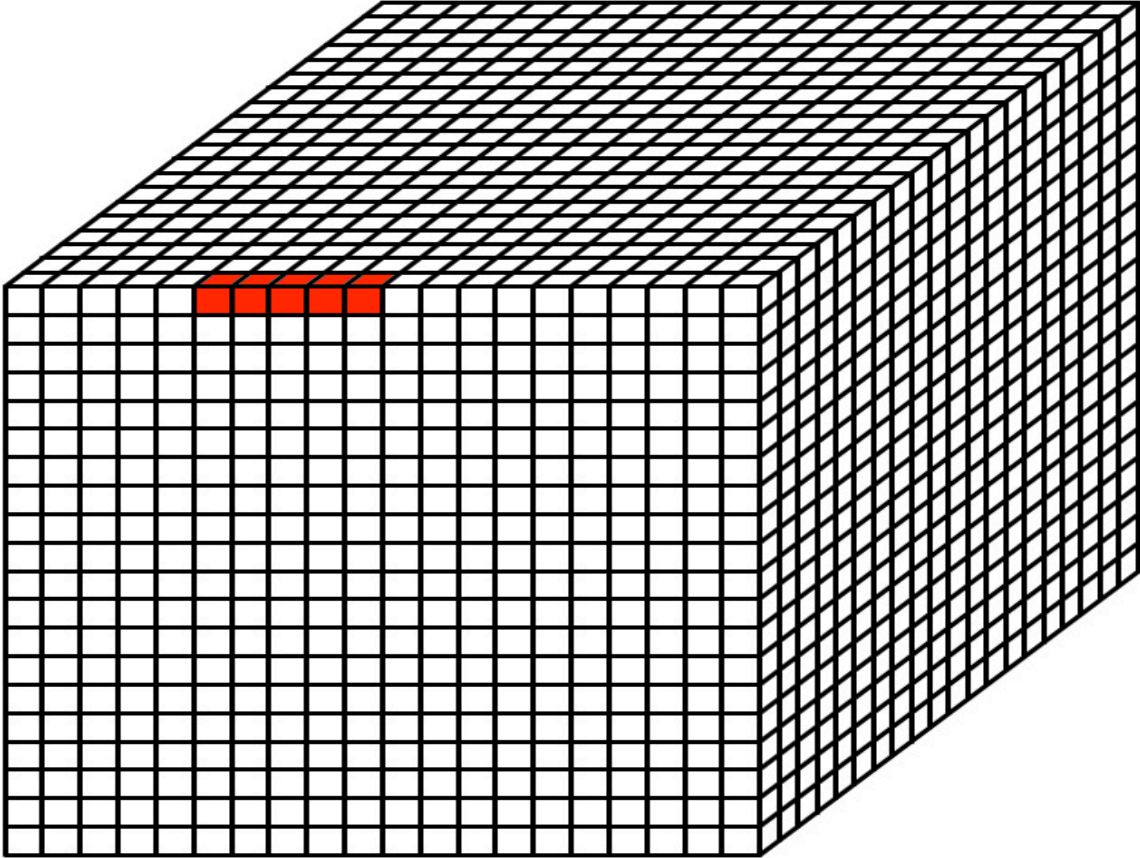
Acceleration by using parallel casting



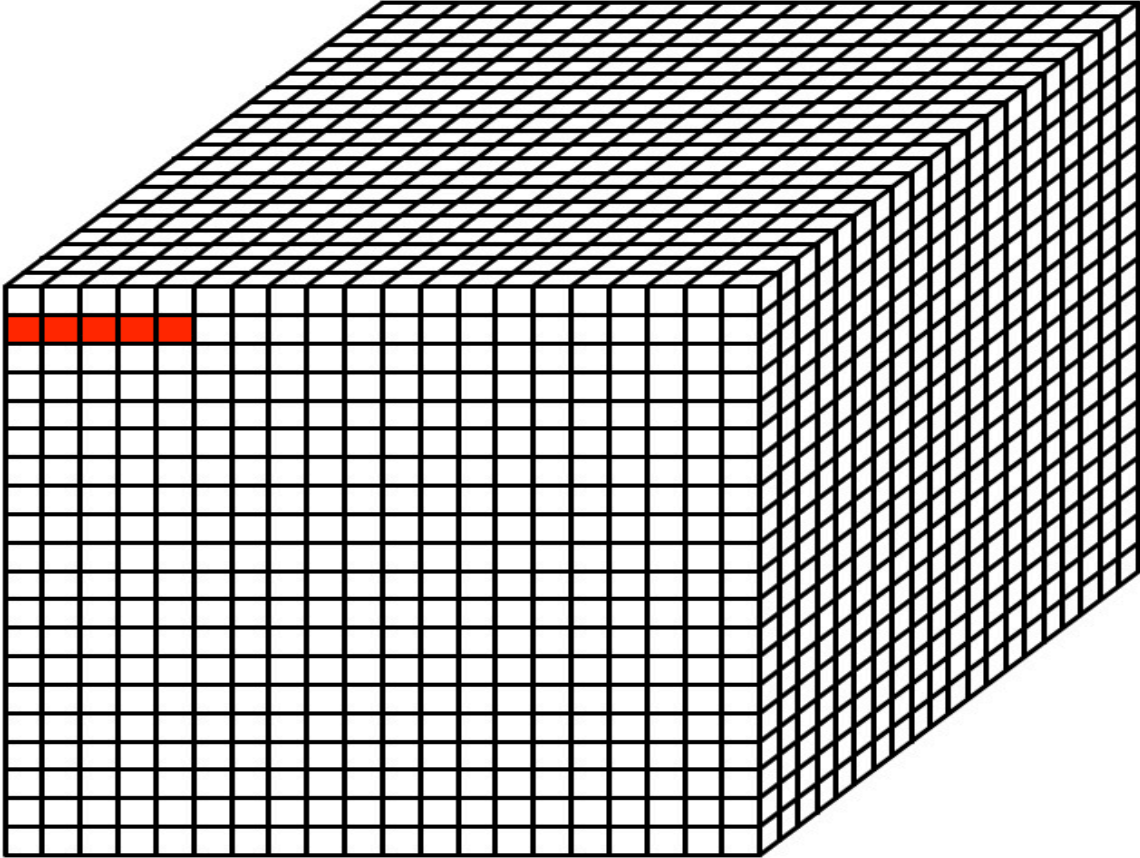
Acceleration by using parallel casting



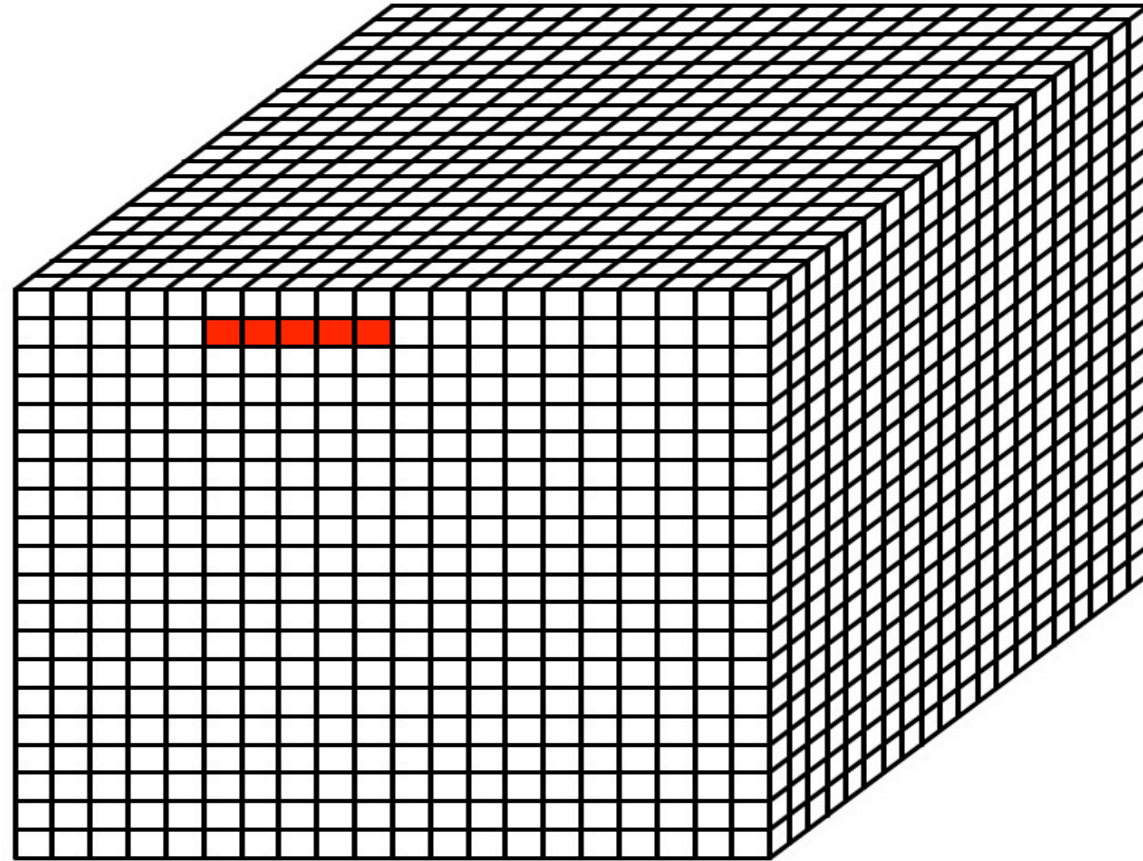
Acceleration by using parallel casting



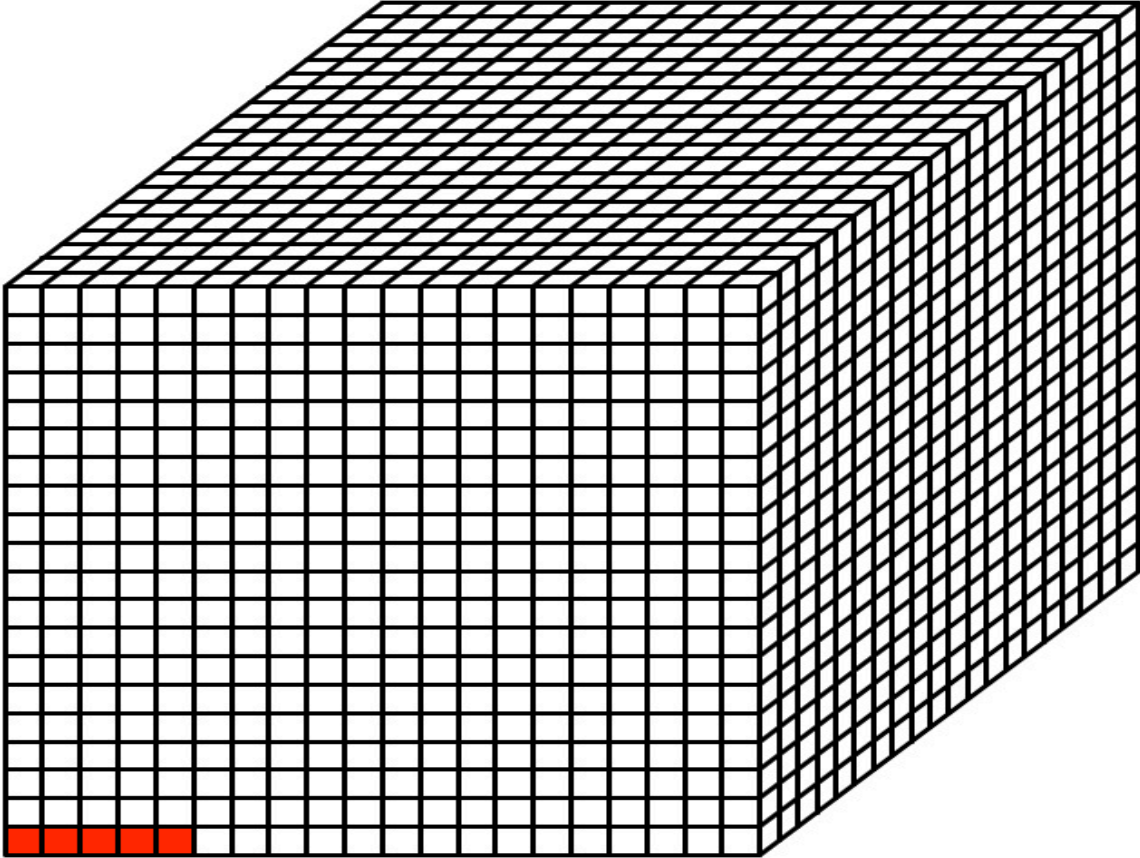
Acceleration by using parallel casting



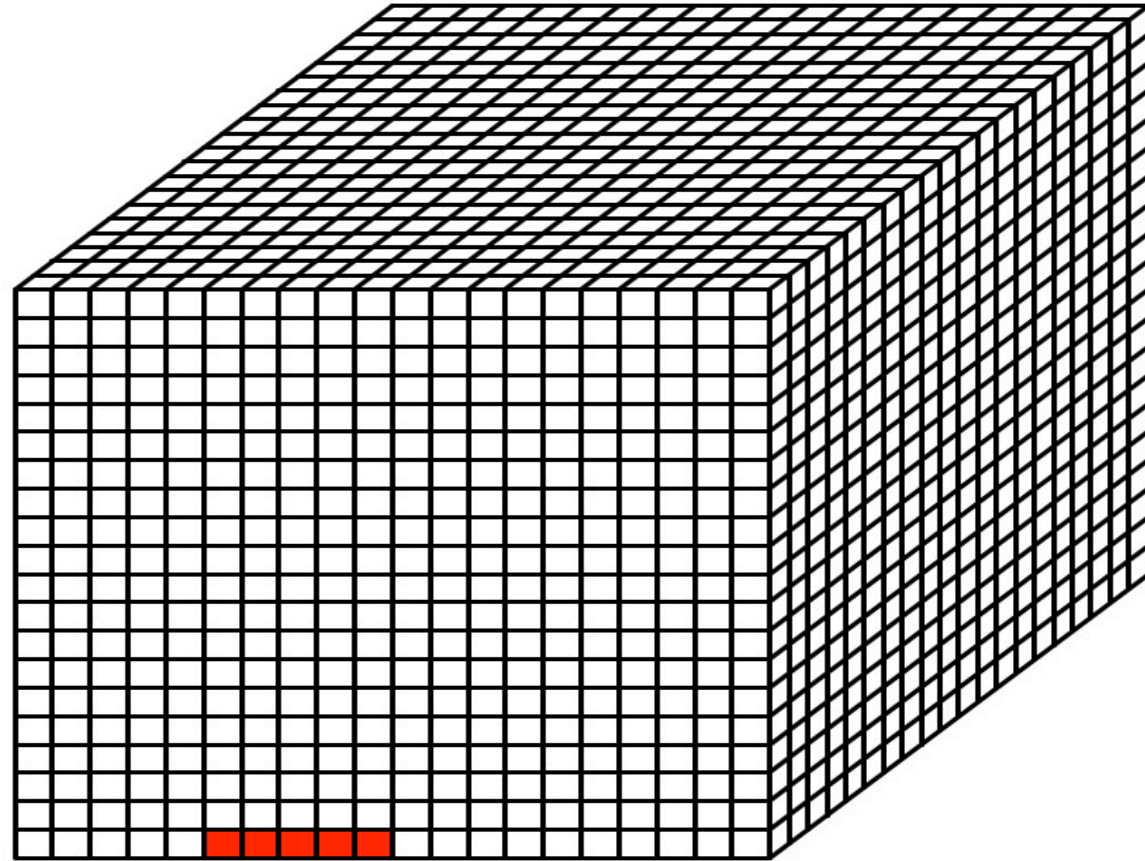
Acceleration by using parallel casting



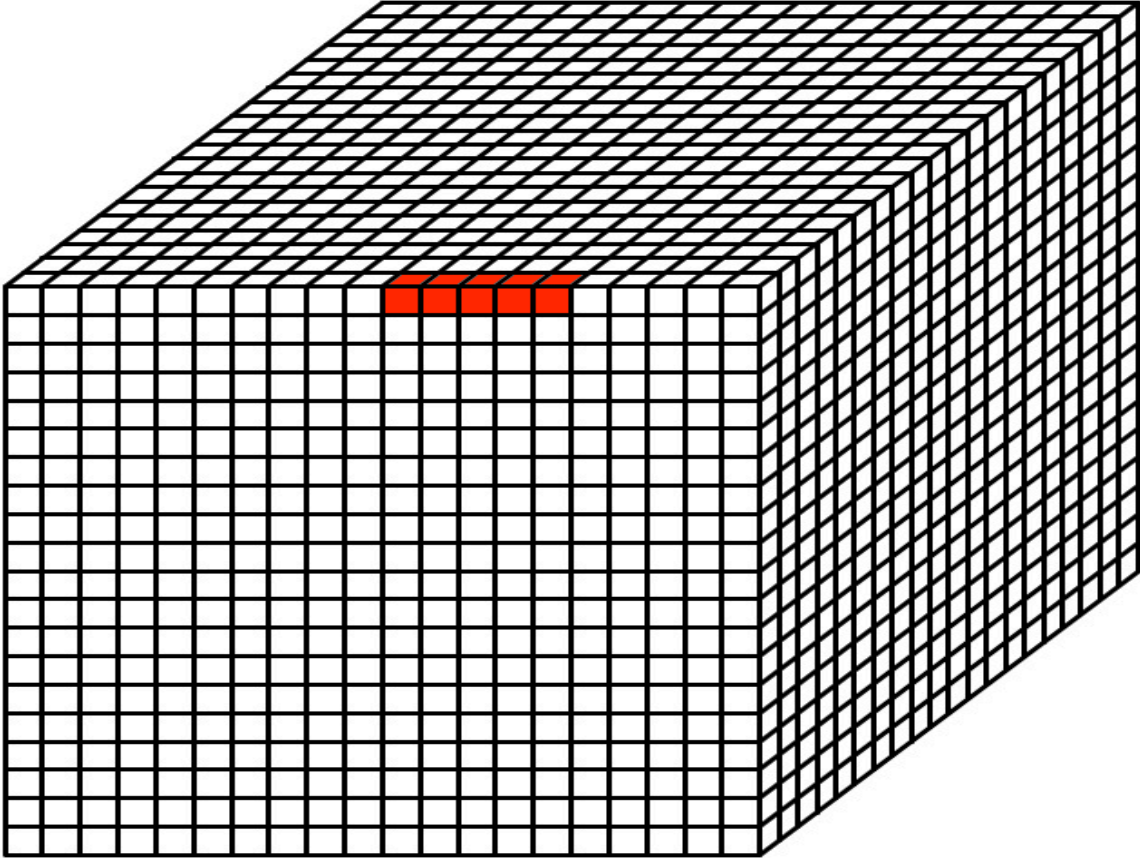
Acceleration by using parallel casting



Acceleration by using parallel casting



Acceleration by using parallel casting



Tradeoff between resolution and refresh rate

- $N*N \rightarrow (N/2)*(N/2)$
- Pixels need to be calculated $\rightarrow 1/4$
- Depth need to be cast $\rightarrow 1/2$
- Total calculation $\rightarrow 1/8$
- Scan 16 times per refresh \rightarrow Scan 2 times per refresh

Ping pang frame buffer

- Frame buffer odd & Frame buffer even
- Read from Frame buffer odd => write to Frame buffer even
- Read from Frame buffer even => write to Frame buffer odd

Separate one object into four part

- That is an illusion
- We actually have four objects, but each one only show 1/4 surface
- This 1/4 surface is dynamic
- 4 hollow objects can be merge into an integrated one
- If 4 hollow objects have different colors, the integrated one looks like a magic cubic
- Achieve it by banking the calculation of object surface

Software in C

- **What software do**
- 1. read signal of gamepad from USB in software, and transfers information messages to hardware
- 2. calculate rotation information of space
- 3. calculate the pass situation
- 4. control the movement of balls
- 5. new graph creation & level select

Part 1

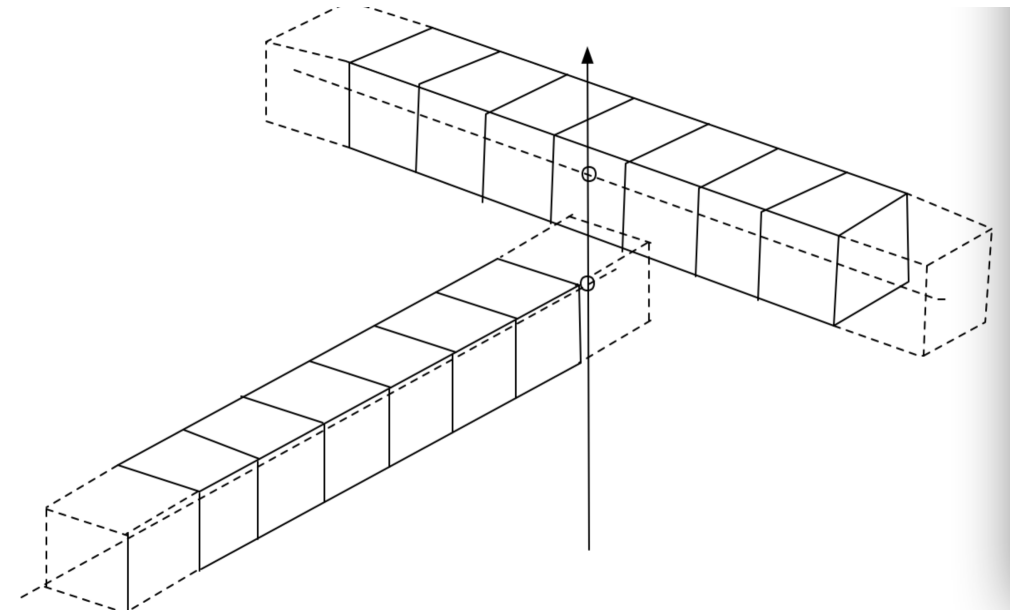
1. Press down key of gamepad, it will generate different signal. Pass it to hello.c
2. When we finish each computation of a time slot, we will transfer computed message to hardware

Part 2

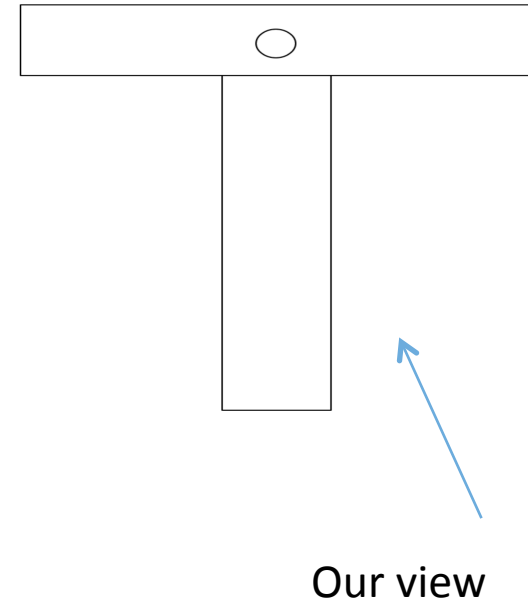
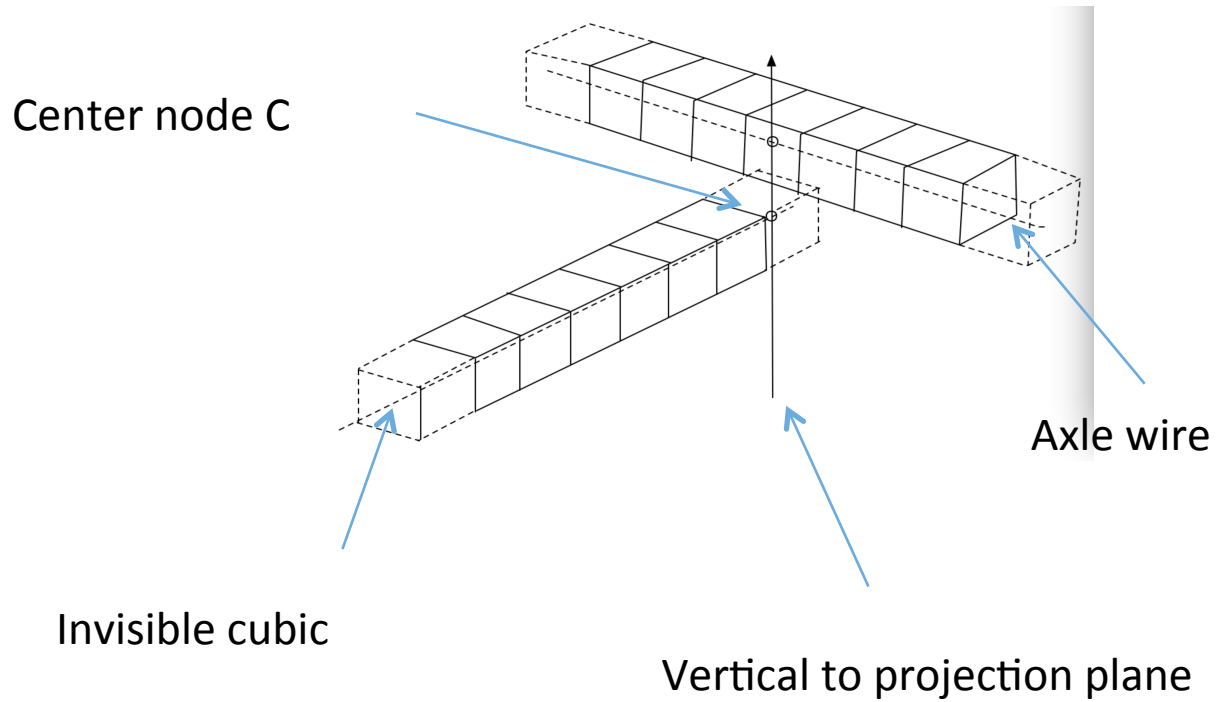
1. Read key value from gamepad, generate the operation number, which will change rotation value from 0 – 4095 for x-coordinate and y-coordinate. As we divide 360 degree into 4096 parts. We will send number from 0-4095 to hardware once it changes.
2. More function
 - control the rotation rate
 - complete a 90 degree rotation for once

Part 3

Our goal is to decide whether two cubic can connect or no in special rotation angles of projection plane.



Part 3



Part 3

implement:

Step1: Record the vectors from nodes on axle wire to all center nodes on other cubic.(initial step)

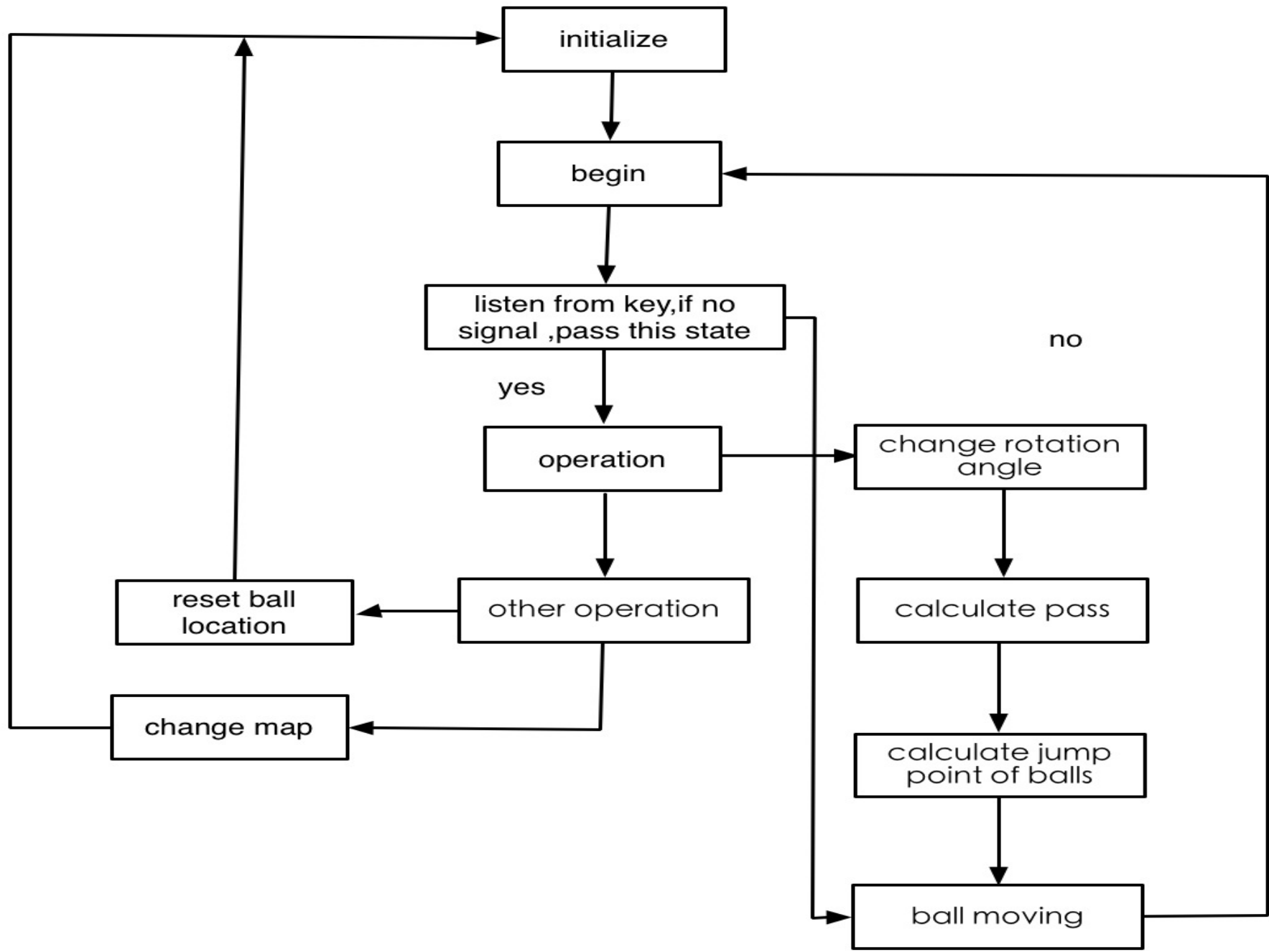
Step2: As in our algorithm, we change the rotation angles of screen, so we only need to change the Normal vector of projection plane. Then compare it to every pair in step1, if their direction are same in same degree. The two cubic are connected.

Part 4

1. Initialize balls – initialize_ball()
2. Normal operation – ball_moving()
3. Decide jump node - calculate_jump_point()
4. Combine two ball when meet – combine()
5. recovery

Part 5

- create new graph
- Select levels



Thank you!