Kinect App Development

KINECT™
for XBOX 360.

By
Rishabh Maheshwari

Objective of today's lecture

Play Angry Birds in 3D

What’s an image ?

An image is simply a collection of pixels, each of which contains some data.

(A pixel is characterized by (x,y))

Let us see some types of images…

Binary Image

Each Pixel has either 1 (White) or
0 (Black)

Each pixel has 1 bit information

(Binary images are seldom used)

0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Grayscale

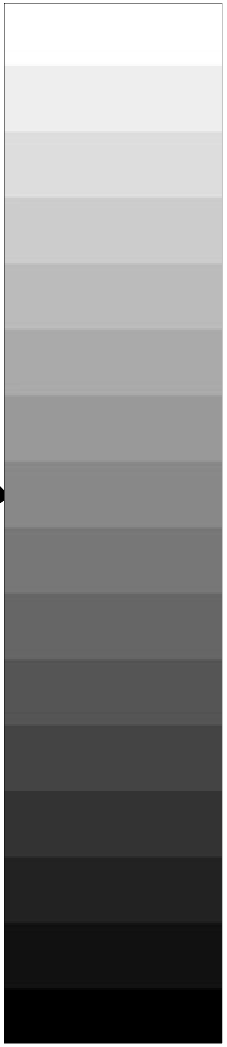
Each Pixel has a value from 0 to 255

0 : black   and   255 : White

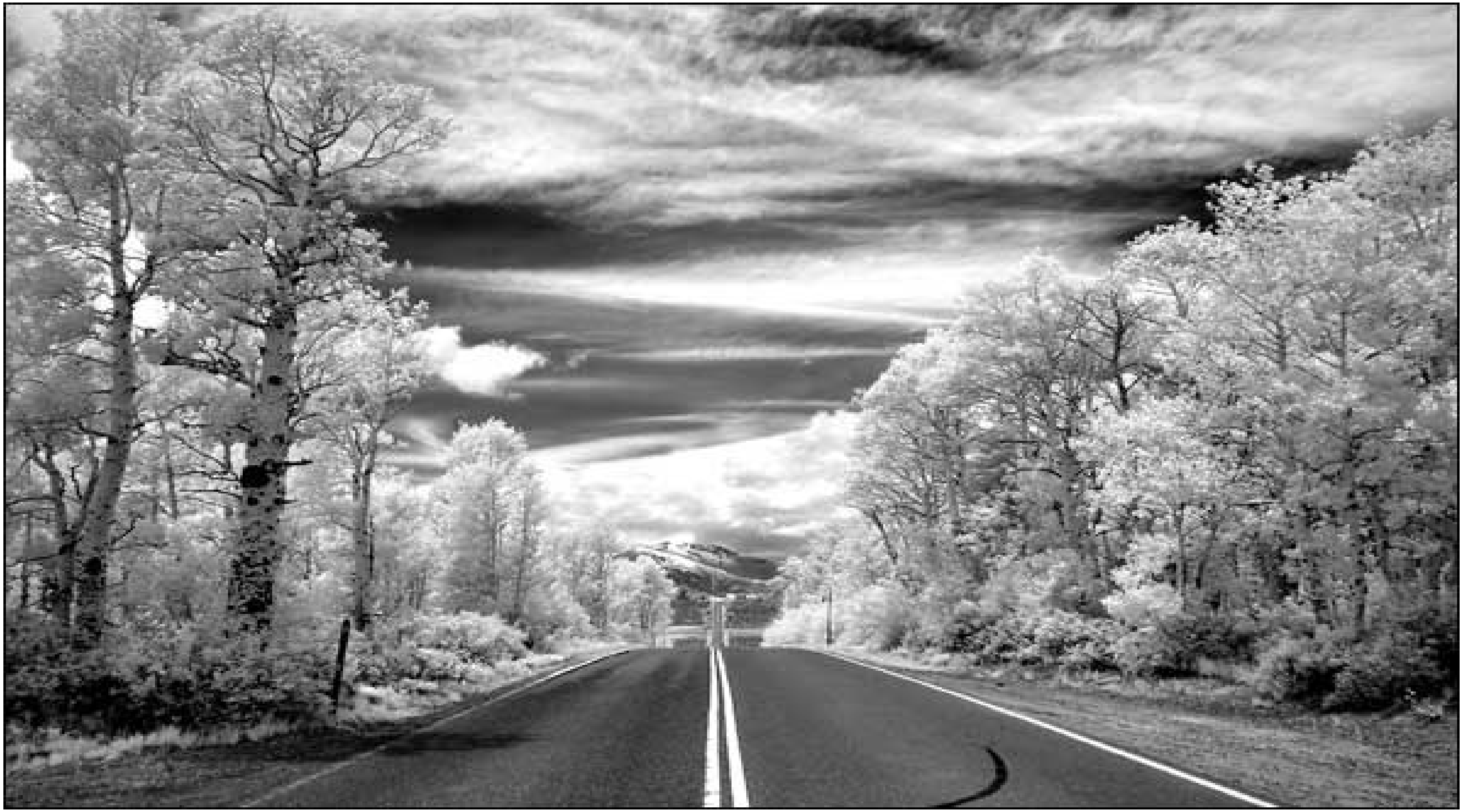
Between 0 and 255 are shades of b&

Each pixel has 1 byte information

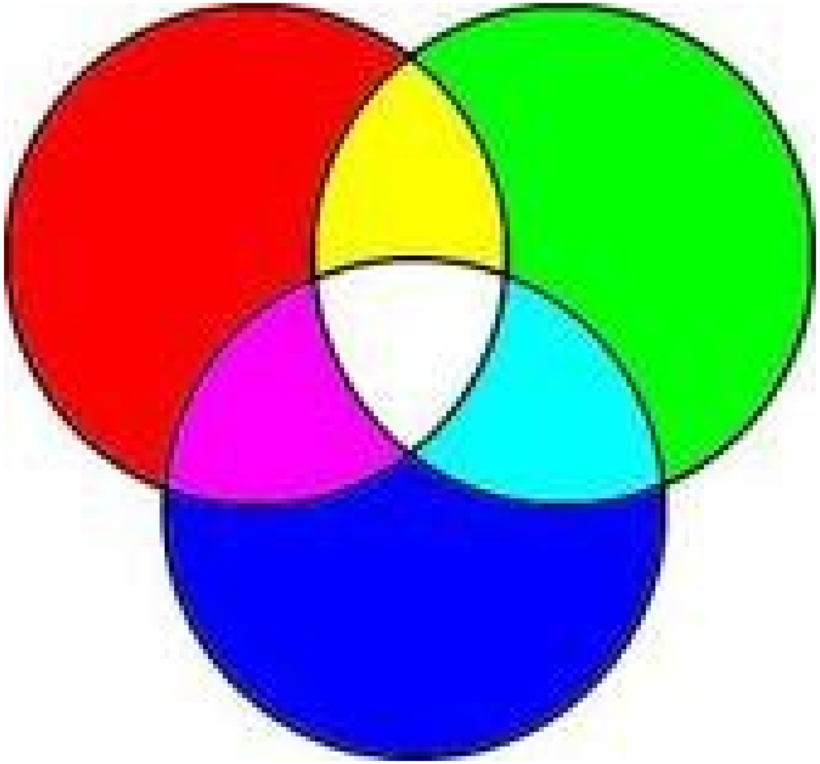
It is stored as an **array of bytes**.



Grayscale Image



RGB Image



Each Pixel stores 3 val

R : 0- 255

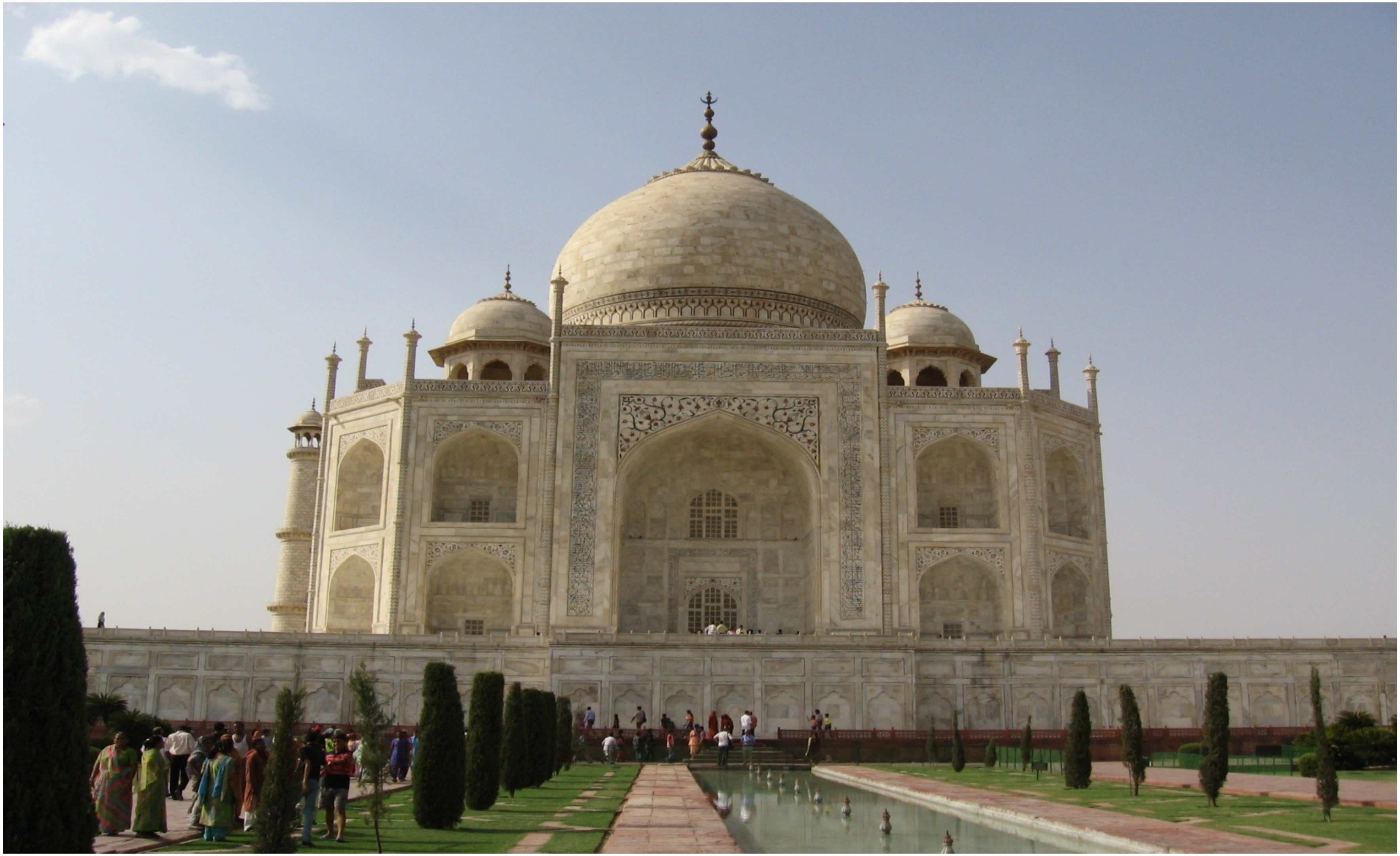
G: 0 -255

B : 0-255

Each pixel has 3 bytes of information

It is also stored as an **array of bytes**.

RGB image



Before moving to **depth image**, we must familiarize ourselves with the basics of kinect.

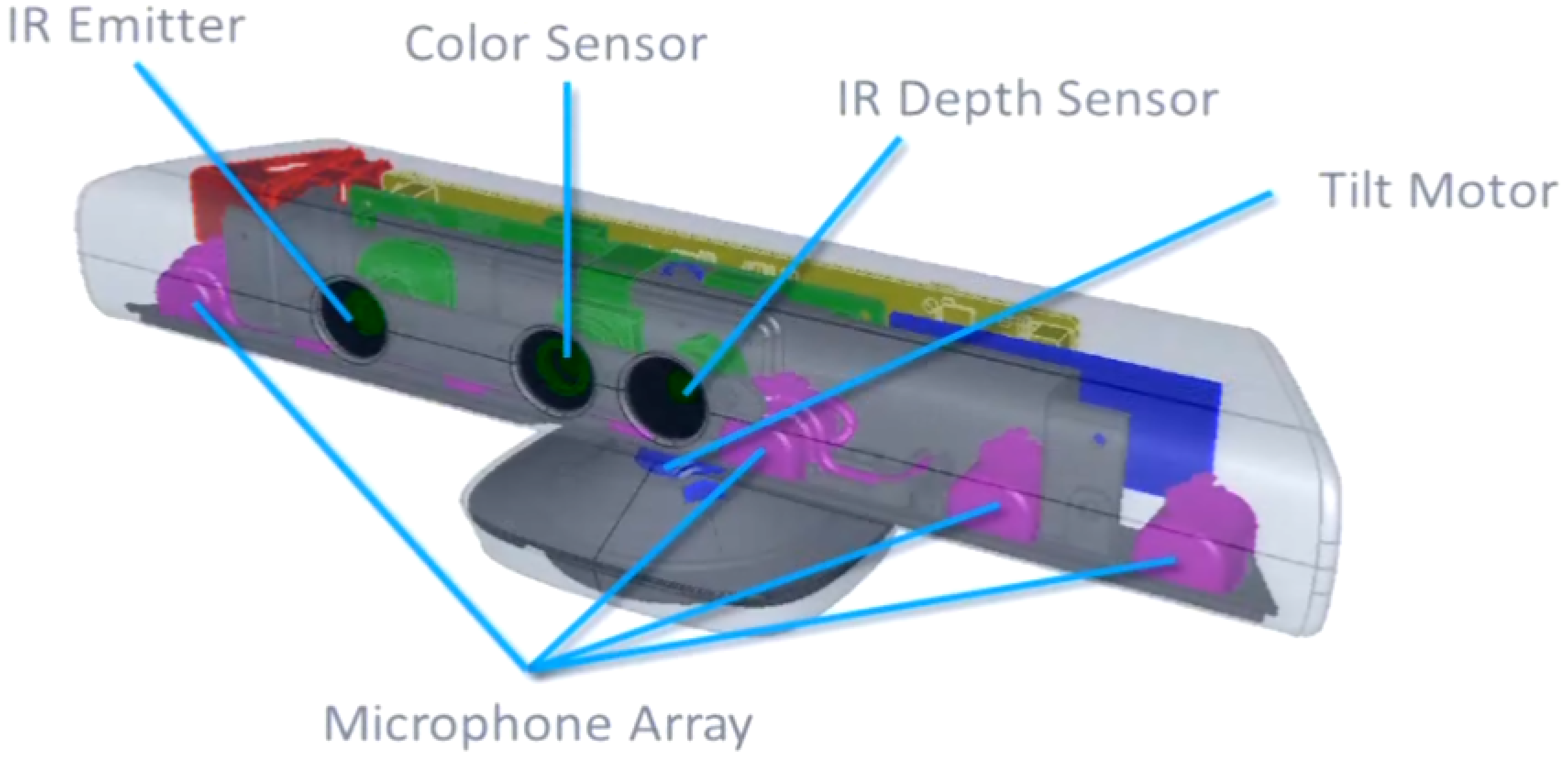
What is a kinect camera ?

Kinect is a camera which gives R , G , B and depth information of each pixel.

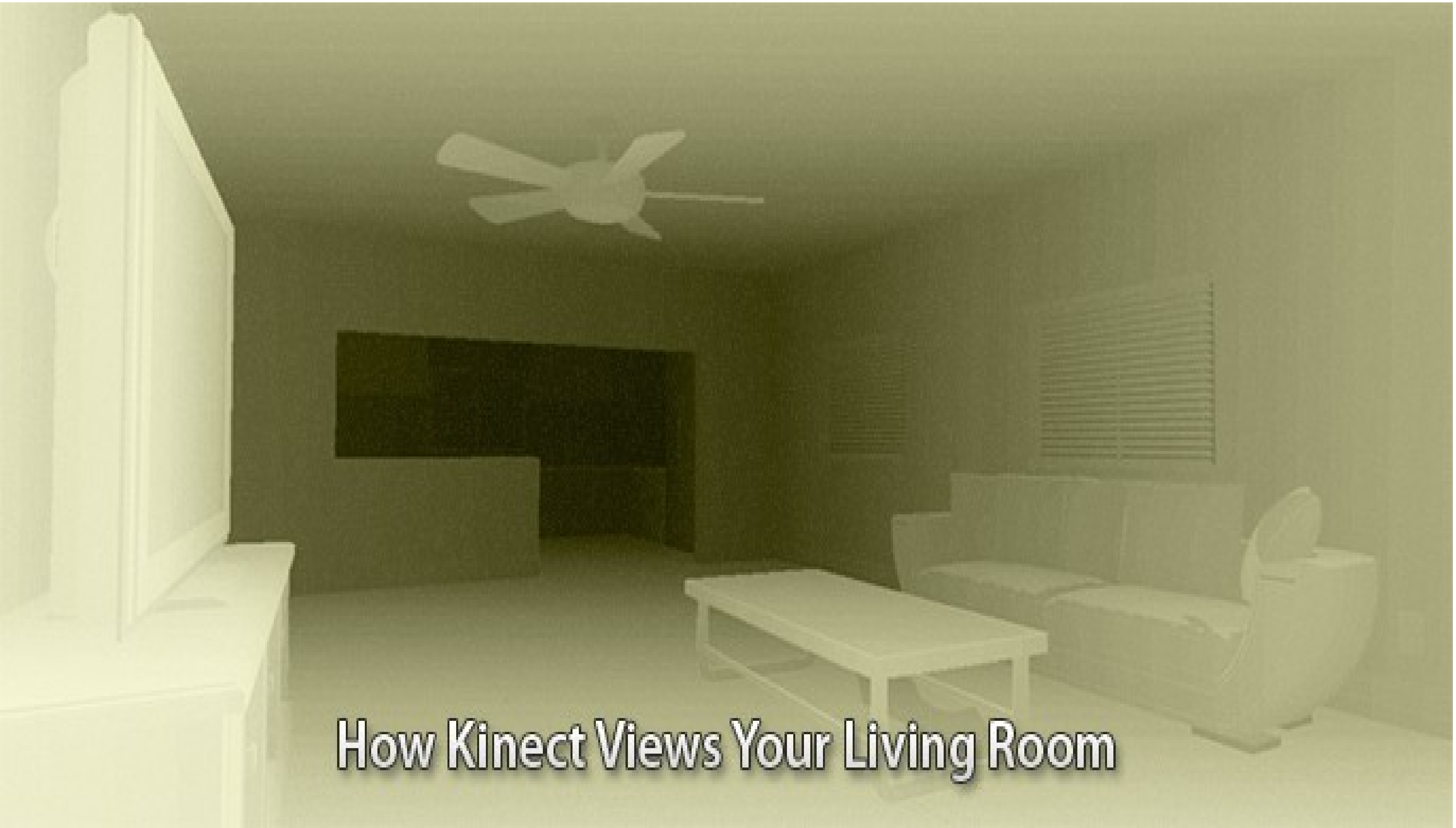
How does Kinect work?

Kinect has 3 components :-

- color camera (takes RGB values)
- IR camera (takes depth data)
- Microphone array (for speech recognition)



Depth Image

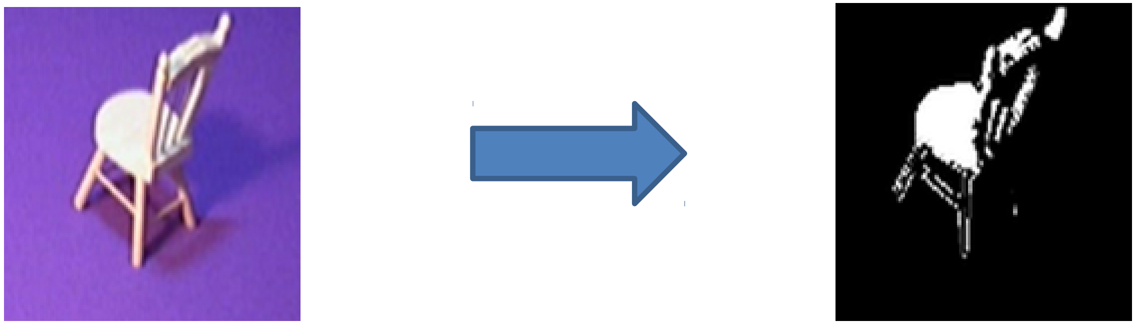


But what’s the use of Depth ??

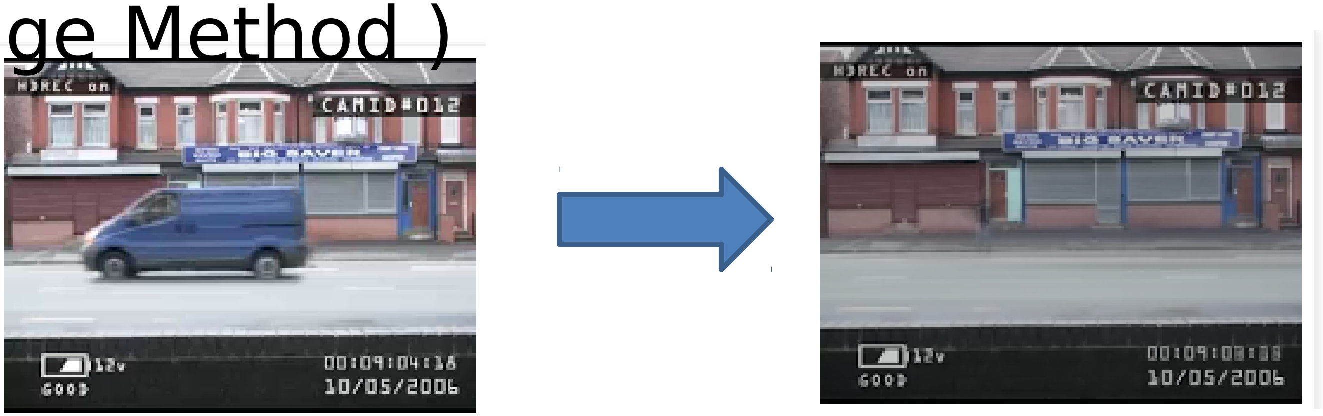
For this, Lets Discuss some *Image Processing*

Background/Foreground Subtraction

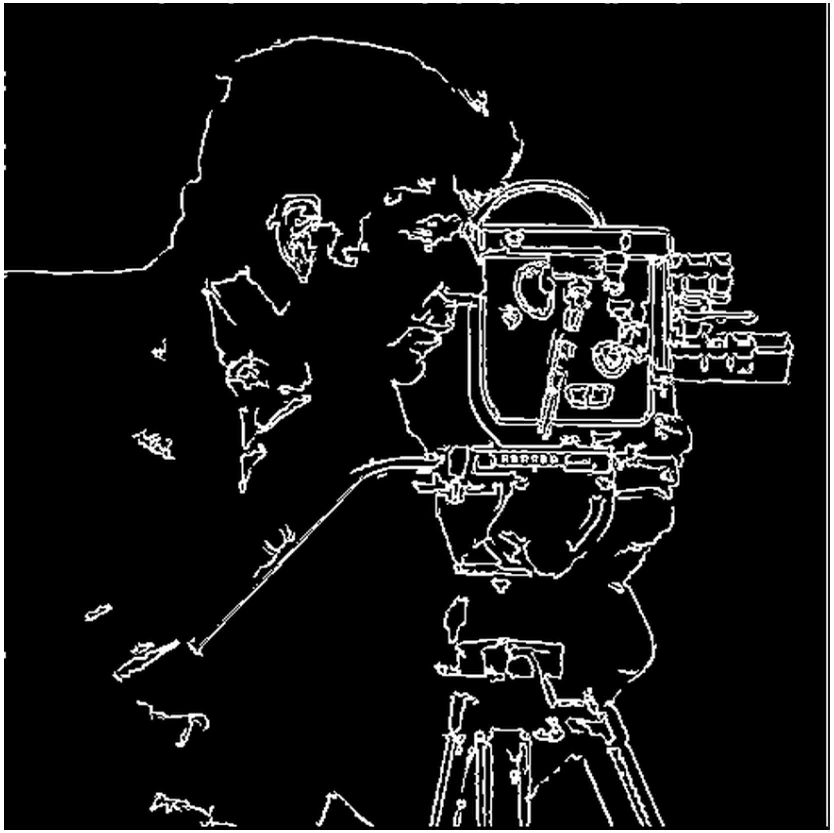
1. On an ***Image*** (Pixel Select Method)



2. On a ***Running Video*** (Running Average Method)



Edge Detection (the gradient)



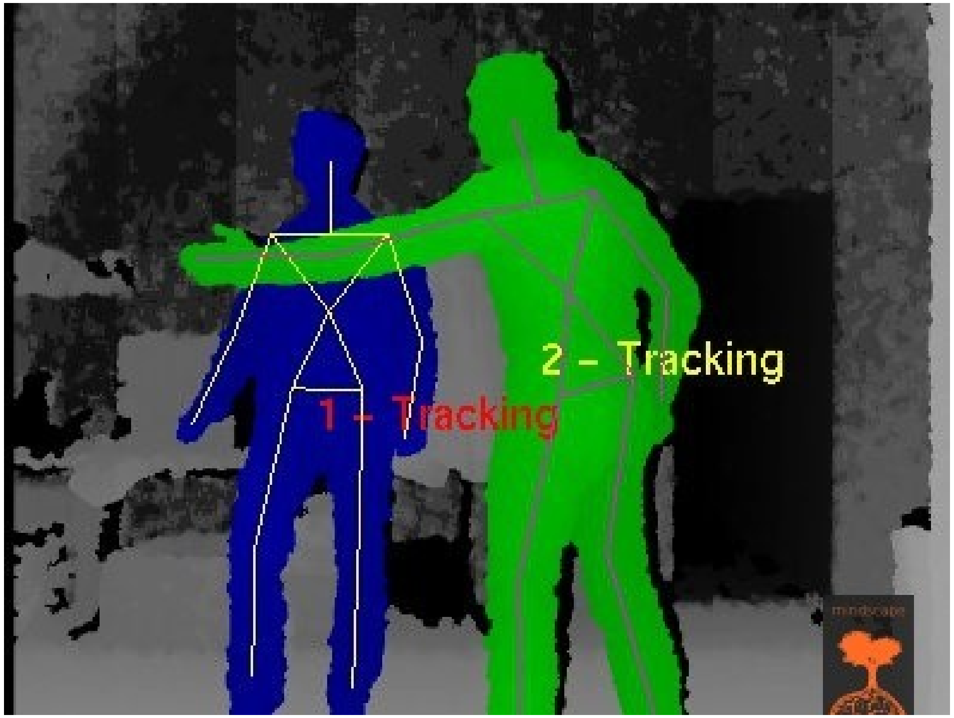
How can a Depth Image help in the above two ??

ㅆ

Back to kinect ….

Player

A player is the (human) skeleton which is detected by kinect sdk. There can be multiple players. Each pixel stores the corresponding “player index”.



🟦 Player index = 1

🟩 Player index = 2

By default:-
Player index =

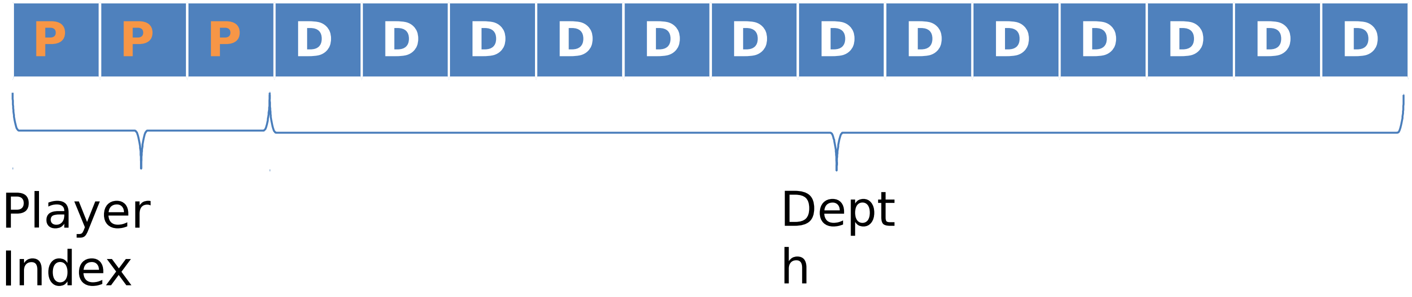
Depth Image (Specific To Kinect sdk v1)

Each pixel stores :-

- Player index : 0 – 7   (3 bits)
- Depth(in mm) : 0 – 8192 (13 bits)

It is stored as an **array of shorts**.

(A short is a 16 bit data type)



Some important datatypes:-

- Kinect is defines as a datatype (same as int or char)

```
 KinectSensor _kinect;
```

- Kinect sdk can handle multiple kinects at same time and treats these kinects as an array of kinect datatype :-

```
 _kinect = KinectSensor.KinectSensors[0];
```

- DepthImagePoint is a struct which stores X ,Y and Depth of a point :-

```
 DepthImagePoint xyz;
```

You can use : xyz.X xyz.Y xyz.Depth

Kinect has 3 streams

- ColorStream : contains RGB data as byte array
- DepthStream : contains depth data as short array
- SkeletonStream : a template (What ??)

You need to enable these streams in the beginning as per your requirements.

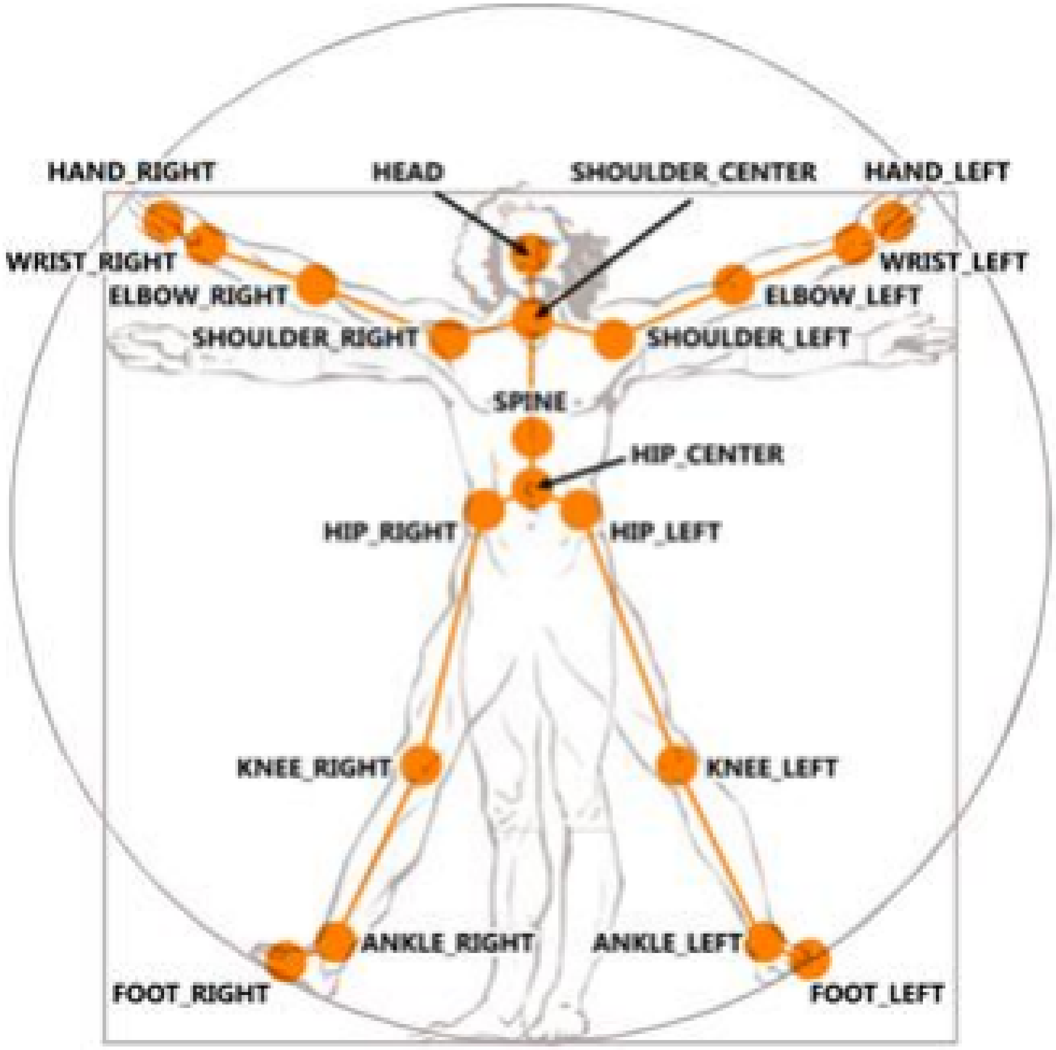
What is a SkeletonStream ?

When *skeletonstream* is called , it recognizes skeletons and **populates pixels of *depthstream* with player index**.

*If *skeletonstream* is not enabled, player index of all pixels of *depthstream* will remain 0.

Joints

Using skeletonstream, kinect sdk provides us with 20 joints.



Joints

Eg:-

```
JointType.HandRight
JointType.FootLeft
JointType.ShoulderLeft
and so on….
```

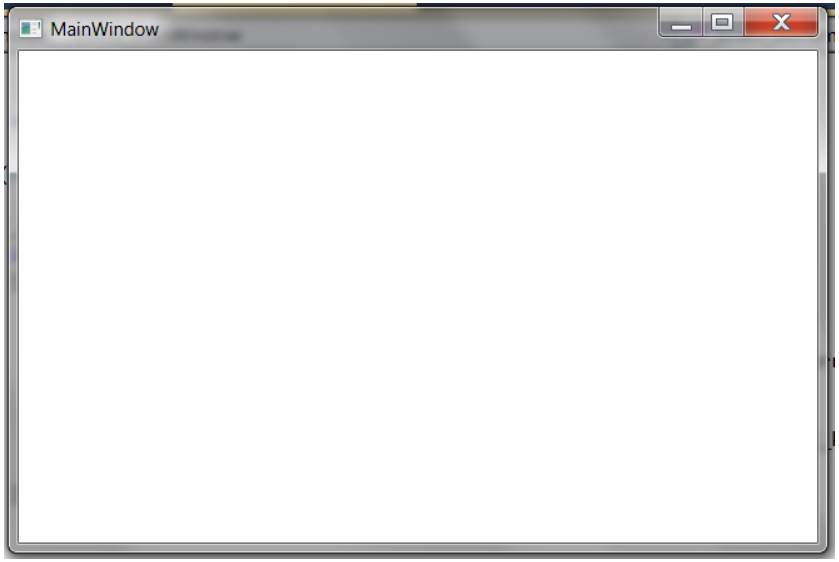
Psuedo Code :-

```
righthand = JointType.HandRight
float x = righthand.X
```

Let’s start with coding

1. Install visual studio.
2. Install kinect sdk for visual studio.
3. Select New Project
4. In C# projects, select WPF project
5. Add Microsoft.Kinect in reference of your project.
6. Write using Microsoft.Kinect;

As you open your new project, a default window is provided.



There are 2 events associated with this window:-

Window_Loaded() //when window loads

Window_Closing() //When X is pressed

The Final basic code :-

```
KinectSensor _kinect;

Window_Loaded()
{
   _kinect = KinectSensor.KinectSensors[0];
   _kinect.ColorStream .Enable();
   _kinect.DepthStream .Enable();
   _kinect.SkeletonStream .Enable();
   _kinect.Start();
}

Window_Closing()
{
   _kinect.Stop();
}
```

Lets see the code to understand more about “frame events”

Questions ?

For online video lectures :-

http://channel9.msdn.com/Series/KinectQuickstart