

# 改善深度影像品質

王泓權

臺科大人工智慧研究中心 助理教授



010010111

# 改善深度影像品質

- ✔ Kinect 感應器原始的深度影像，離感應器越近的物體顏色越暗，這與我們一般的感覺越近越亮剛好相反
- ✔ 深度量測有問題的情況有3種
  - UnKnowDepth (無法決定深度)
  - TooFarDepth(深度超過最大感測範圍)
  - TooNearDepth(深度小於最近感測範圍)



# 改善深度影像品質

- ✔ Kinect 感應器正常模式下能正常感測的深度範圍是0.8~4公尺，而0.4~0.8公尺的範圍定義為 TooNearDepth，4~8公尺定義為 TooFarDepth 的範圍，0.4公尺以內與8公尺以外則是 UnKnowDepth 代表的範圍





# 改善深度影像品質

- ✔ 在此專案中，將以從黑到白共255個亮度變化表示不同的深度值，並順從一般對距離的感覺，離 Kinect 感應器越近則越亮
- ✔ 另外深度影像的格式將從16位元灰階，擴大成32位元的RGB格式，只要將RGB 3個顏色設為相同值，就可以得到0~255的亮度變化，其中為了方便區別3種有問題的深度像素，這裡設計成可以讓使用者依照自己的喜好設定問題像素所呈現的顏色

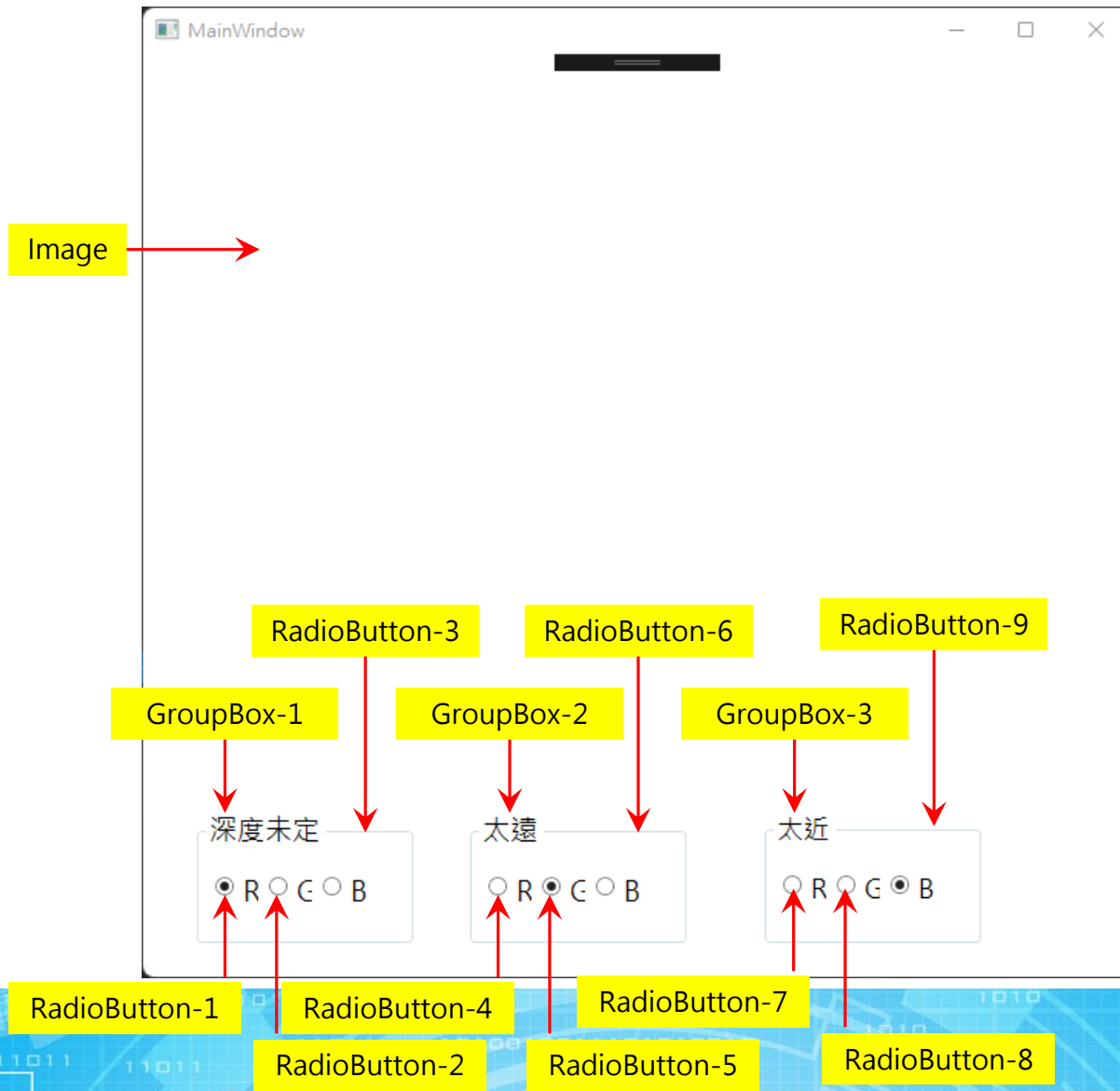


# 改善深度影像品質

- ✔ Step 01: 建立新專案-EnhancedImageTest，在專案中增加對 Kinect SDK 參考
- ✔ Step 02: 建立使用者介面，新增1個 Image 控制項、3個 GroupBox 控制項，每一個 GroupBox 控制項內放置3個 RadioButton 控制項



# 改善深度影像品質



# 改善深度影像品質

- ✔ Step 03: 初始化作業，登錄  
StatusChanged 事件及定義其事件處理常式  
KinectSensor\_StatusChanged，新增3個控制深度有問題的像素色彩欄位

```
private byte[] _colorDataByte; //定義影像暫存資料陣列
private int _unknowCode=0; //控制深度未定像素顏色
private int _tooFarCode = 1; //控制深度太遠像素顏色
private int _tooNearCode = 2; //控制深度太近像素顏色
```



# 改善深度影像品質

## 🟢 Step 04: 串流初始化、啟動與停止作業

```
//InitialStream函式
private void InitialStream()
{
    this._myKinect.DepthStream.Enable(); //要求Kinect感應器產生深度資料串流
    //建立WriteableBitmap物件及影像區域
    this._writeableBitmap = new WriteableBitmap(_myKinect.DepthStream.FrameWidth,
_myKinect.DepthStream.FrameHeight, 96, 96, PixelFormats.Bgr32, null);
    this._imageRect = new Int32Rect(0, 0, _myKinect.DepthStream.FrameWidth,
_myKinect.DepthStream.FrameHeight);
    //指定Image控制項影像來源為WriteableBitmap物件,使Image控制項顯示內容隨
WriteableBitmap物件內容改變
    DepthImage.Source = this._writeableBitmap;
    //用影格資料像素資料大小的4倍定義暫存資料陣列長度
    _colorDataByte = new byte[this._myKinect.DepthStream.FramePixelDataLength*4];
    //註冊Kinect_DepthFrameReady事件處理函式
    this._myKinect.DepthFrameReady += Kinect_DepthFrameReady;
    this._myKinect.Start(); //啟動Kinect感應器硬體
}
```



# 改善深度影像品質

## 🟢 Step 05: 深度影格備妥事件處理常式

```
//影格備妥事件處理常式-- Code 4-15
private void Kinect_DepthFrameReady(object sender, DepthImageFrameReadyEventArgs e)
{
    using (DepthImageFrame frameData = e.OpenDepthImageFrame()) //取得傳遞的影格資料
    {
        if (frameData == null) //如果影格資料不存在,直接離開事件處理函式
        {
            return;
        }
        short[] depthDataArray=new short[this._myKinect.DepthStream.FramePixelDataLength]; //定義
        影像暫存資料陣列
        //將影格資料複製到資料陣列
        frameData.CopyPixelDataTo(depthDataArray);
        int brightPosition = 0; //設定處理第1個像素
        for (int i = 0; i < depthDataArray.Length; i++)
        {
            int depthValue = depthDataArray[i] >> DepthImageFrame.PlayerIndexBitmaskWidth; //取得目前像
            素深度資料
        }
    }
}
```



# 改善深度影像品質

```
//如果像素問題是深度未定，依unknowCode欄位值設定像素顏色
if (depthValue == this._myKinect.DepthStream.UnknownDepth)
{
    switch (_unknowCode)
    {
        case 0:
            this._colorDataByte[brightPosition] = 0; //設定藍色頻道
            this._colorDataByte[brightPosition+1] = 0; //設定綠色頻道
            this._colorDataByte[brightPosition+2] = 255; //設定紅色頻道
            break;
        case 1:
            this._colorDataByte[brightPosition] = 0; //設定藍色頻道
            this._colorDataByte[brightPosition+1] = 255; //設定綠色頻道
            this._colorDataByte[brightPosition+2] = 0; //設定紅色頻道
            break;
        case 2:
            this._colorDataByte[brightPosition] = 255; //設定藍色頻道
            this._colorDataByte[brightPosition+1] = 0; //設定綠色頻道
            this._colorDataByte[brightPosition+2] = 0; //設定紅色頻道
            break;
    }
}
```



# 改善深度影像品質

```
//如果像素問題是深度太遠，依tooFarCode欄位值設定像素顏色
else if (depthValue == this._myKinect.DepthStream.TooFarDepth)
{
    switch (_tooFarCode)
    {
        case 0:
            this._colorDataByte[brightPosition] = 0; //設定藍色頻道
            this._colorDataByte[brightPosition + 1] = 0; //設定綠色頻道
            this._colorDataByte[brightPosition + 2] = 255; //設定紅色頻道
            break;
        case 1:
            this._colorDataByte[brightPosition] = 0; //設定藍色頻道
            this._colorDataByte[brightPosition + 1] = 255; //設定綠色頻道
            this._colorDataByte[brightPosition + 2] = 0; //設定紅色頻道
            break;
        case 2:
            this._colorDataByte[brightPosition] = 255; //設定藍色頻道
            this._colorDataByte[brightPosition + 1] = 0; //設定綠色頻道
            this._colorDataByte[brightPosition + 2] = 0; //設定紅色頻道
            break;
    }
}
```



# 改善深度影像品質

```
//如果像素問題是深度太近，依tooNearCode欄位值設定像素顏色
else if (depthValue == this._myKinect.DepthStream.TooNearDepth)
{
    switch (_tooNearCode)
    {
        case 0:
            this._colorDataByte[brightPosition] = 0; //設定藍色頻道
            this._colorDataByte[brightPosition + 1] = 0; //設定綠色頻道
            this._colorDataByte[brightPosition + 2] = 255; //設定紅色頻道
            break;
        case 1:
            this._colorDataByte[brightPosition] = 0; //設定藍色頻道
            this._colorDataByte[brightPosition + 1] = 255; //設定綠色頻道
            this._colorDataByte[brightPosition + 2] = 0; //設定紅色頻道
            break;
        case 2:
            this._colorDataByte[brightPosition] = 255; //設定藍色頻道
            this._colorDataByte[brightPosition + 1] = 0; //設定綠色頻道
            this._colorDataByte[brightPosition + 2] = 0; //設定紅色頻道
            break;
    }
}
```





# 改善深度影像品質

```
else
{
    //先正規化像素的深度值，再取相反亮度
    //byte brightByte = (byte)(255 - (depthValue >> 4));
    byte brightByte = (byte)(255-255 * depthValue / 0xFFFF);
    this._colorDataByte[brightPosition] = brightByte; //設定藍色頻道
    this._colorDataByte[brightPosition + 1] = brightByte; //設定綠色頻道
    this._colorDataByte[brightPosition + 2] = brightByte; //設定紅色頻道
}
brightPosition += 4; //移至下一像素
}
this._writeableBitmap.WritePixels(this._imageRect, this._colorDataByte,
frameData.Width*4, 0);
}
```



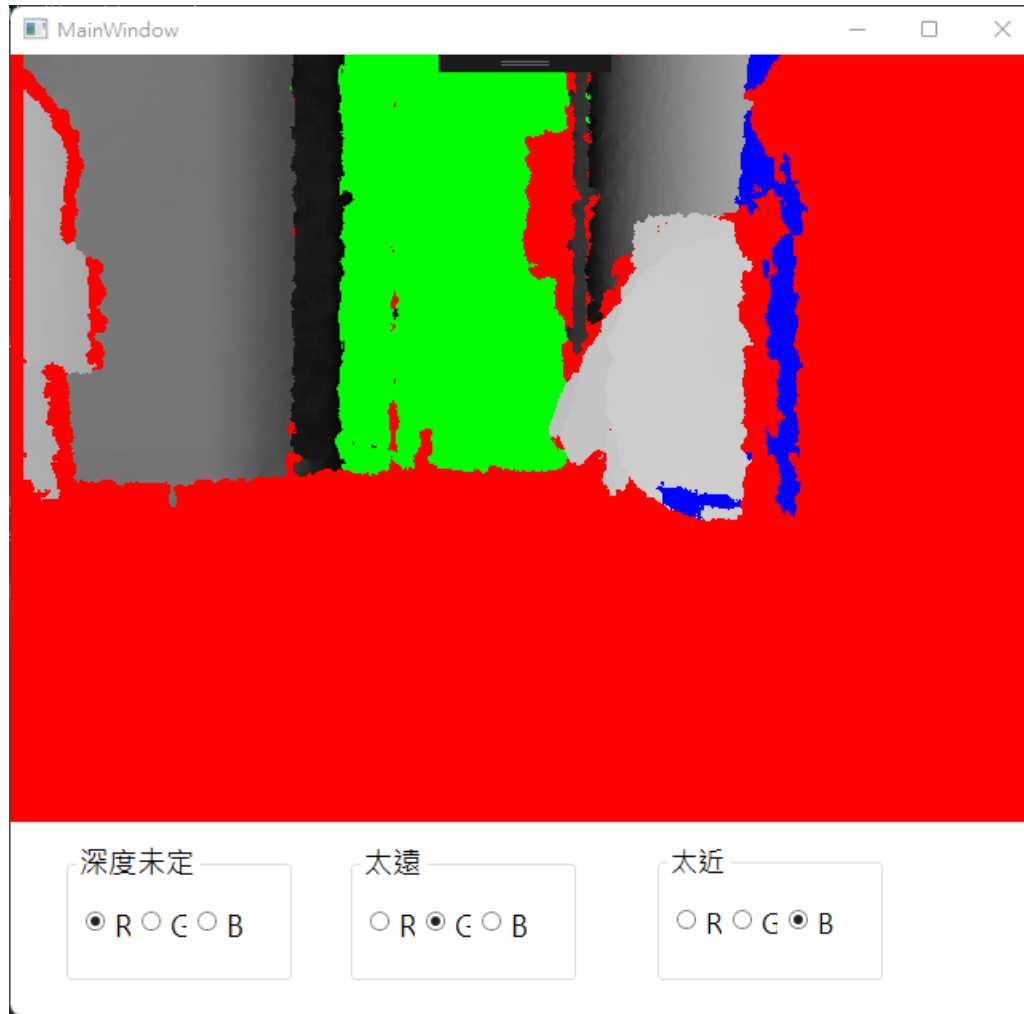
# 改善深度影像品質

## ✔ Step 06: 深度值有問題像素顏色指定

```
//RadioButton共用事件處理常式, 設定問題像素顏色  
private void Radio_Clicked(object sender, RoutedEventArgs e)  
{  
    //UnknowDepth像素顏色設定  
    if ((bool)UR.IsChecked) _unknowCode = 0;  
    if ((bool)UG.IsChecked) _unknowCode = 1;  
    if ((bool)UB.IsChecked) _unknowCode = 2;  
    //TooFarDepth像素顏色設定  
    if ((bool)FR.IsChecked) _tooFarCode = 0;  
    if ((bool)FG.IsChecked) _tooFarCode = 1;  
    if ((bool)FB.IsChecked) _tooFarCode = 2;  
    //TooNearDepth像素顏色設定  
    if ((bool)NR.IsChecked) _tooNearCode = 0;  
    if ((bool)NG.IsChecked) _tooNearCode = 1;  
    if ((bool)NB.IsChecked) _tooNearCode = 2;  
}
```



# 測試



000101