

視訊影像相關類別與 主要方法介紹

王泓權

臺科大人工智慧研究中心 助理教授



010010111

彩色影像顯示專案

- ✔ Step 01: 建立新專案-`RGBImageDisplay`
- ✔ Step 02: 建立使用者介面，新增顯示元件-`Image`
- ✔ Step 03: 登錄 `StatusChanged` 事件及定義其事件處理常式 `KinectSensors_StatusChanged` 以及登錄 `Loaded` 事件的處理常式



彩色影像顯示專案

```
KinectSensor _myKinect;  
public MainWindow()  
{  
    InitializeComponent();  
    KinectSensor.KinectSensors.StatusChanged +=  
        KinectSensors_StatusChanged;  
}
```



彩色影像顯示專案

```
// KinectSensor.KinectSensors.StatusChanged事件處理函式
private void KinectSensors_StatusChanged(object sender, StatusChangedEventArgs e)
{
    switch (e.Status) //依感應器目前狀態啟用或停止串流
    {
        case KinectStatus.Connected:
            //如果KinectSense物件未參考至任何感應器，則將KinectSensor物件參考至觸發事件的
            //並呼叫InitialStream函式啟用串流，if條件式是為了確保應用程式只參考一個Kinect感
            if (this._myKinect == null)
            {
                this._myKinect = e.Sensor;
                InitialStream();
            }
            break;
        case KinectStatus.Disconnected:
            //如果Kinect感應器斷線，呼叫UninitialStream停止串流相關工作
            UninitialStream();
            break;
    }
}
```

感應器

應器



彩色影像顯示專案

```
//Loaded事件處理函式
private void Window_Loaded_1(object sender, RoutedEventArgs e)
{
    if (KinectSensor.KinectSensors.Count == 0)
    {
        MessageBox.Show("請將Kinect接上電腦");
        //Application.Current.Shutdown();
    }
    //如果已連至電腦感應器集合中第一個感應器的狀態為Connected,則將
    KinectSensor物件參考至此感應器，並
    //呼叫InitialStream函式啟用串流
    else if (KinectSensor.KinectSensors[0].Status == KinectStatus.Connected)
    {
        this._myKinect = KinectSensor.KinectSensors[0];
        InitialStream();
    }
}
```



彩色影像顯示專案

- ✔ Step 04: InitialStream 函式使用 ColorStream 物件的 Enabled 方法告訴 Kinect 感應器提供 RGB 資料串流。

```
//InitialStream函式
private void InitialStream()
{
    this._myKinect.ColorStream.Enable(); //要求Kinect感應器產生RGB
    資料串流
    //註冊ColorFrameReady事件與Kinect_ColorFrameReady事件處理函
    式
    this._myKinect.ColorFrameReady += Kinect_ColorFrameReady;
    this._myKinect.Start(); //啟動Kinect感應器硬體
}
```

彩色影像顯示專案

- ✔ Step 05: UninitialStream 函式的作用則是取消對 ColorFrameReady 事件登錄

```
//UninitialStream函式
private void UninitialStream()
{
    //取消ColorFrameReady事件註冊
    this._myKinect.ColorFrameReady -= Kinect_ColorFrameReady;
    this._myKinect.Stop(); //停止Kinect感應器硬體工作
    this._myKinect = null; //取消對Kinect感應器硬體參考
}
```



彩色影像顯示專案

- ✔ Step 06: 用 Kinect_ColorFrameReady 事件處理常式取得影像資料並顯示。
ColorImageFrame 物件生命期只在 using 區塊內。

```
//取得傳遞的影格資料
using (ColorImageFrame frameData = e.OpenColorImageFrame())
{
    //如果影格資料不存在,直接離開事件處理函式
    if (frameData == null)
    {
        return;
    }
    ....
}
```


彩色影像顯示專案

- ✔ Step 07: 建立一個用以暫時存放 `ColorImageFrame` 物件像素的位元組陣列

```
byte[] imageDataByte = new byte[frameData.PixelDataLength];
```

- ✔ Step 08: 將影格資料從 `ColorImageFrame` 物件移到此位元組陣列

```
frameData.CopyPixelDataTo(imageDataByte);
```



彩色影像顯示專案

- ✔ Step 08: 最後用 BitmapSoure.Create 方法產生 Image 控制影像的來源，就能將 Kinect 感應器傳來的 RGB 影像真實的呈現出來。

```
RGBImage.Source = BitmapSource.Create(  
    frameData.Width, frameData.Height,  
    96, 96,  
    PixelFormats.Bgr32,  
    null,  
    imageDataByte,  
    frameData.Width * frameData.BytesPerPixel  
);
```

//點陣圖寬、高
//點陣圖解析度
//點陣圖像素格式
//調色盤
//位元組陣列
//點陣圖步距



彩色影像顯示專案

