中華大學資訊工程學系

100 學年度專題製作期末報告

- 利用 Kinect 製作 3D 攝影機 -

組　　員：莊勝富　林子雀
張登貴　楊曜彰

指導教授：鄭芳炫 教授

專題題號：**PRJ2011-CSIE-10005**

中 華 民 國 １０１ 年 8 月 １８ 日

目錄

專題計劃摘要 2

目的及背景 3

專題研究方法 5

Kinect 簡介
Kinect 官方系統需求
研究方法

專題開發流程 9

專題進度圖
專題分工

專題主要內容 10

對於深度影象的修改
遭遇困難 i
遭遇困難 ii:
解決方式
缺點

評估與展望 23

銘謝 24

參考文獻 25

專題計劃摘要

隨著科技的進步，電腦的應用推陳出新。其中 3D 技術正架構著虛擬與真實世界的無縫聯結，近年微軟 Microsoft 公司推出了支援 XBOX360 的體感遊戲攝影機 Kinect，銷售至今以在全世界共賣 5000 多萬套! 由此可見，Kinect 已經成功地獲得消費者的青睞。伴隨著 Kinect 的熱銷，微軟推出了 Kinect 連接電腦專用的應用程式，其名為 KinectSDK。在此之後，有如雨後春筍般 Kinect 的應用在網路上大量的出現，讓網友們大開眼界!也希望能藉此打造出更有趣、更實用的電腦體驗，而我們的目標為先了解 KinectSDK 如何在電腦上操作，以及裝置如何連接上電腦來使用，之後再利用 Kinect 攝影機得到的彩色及深度影像進而合成為 3D 影片來完成我們這次的專題製作。

目的及背景

目的

我們希望可以透過這個程式擷取 Kinect 所拍下的彩色及深度畫面來完成 3D 影片的製作，讓使用者能透過 Kinect 做出自己喜愛的 3D 影片。

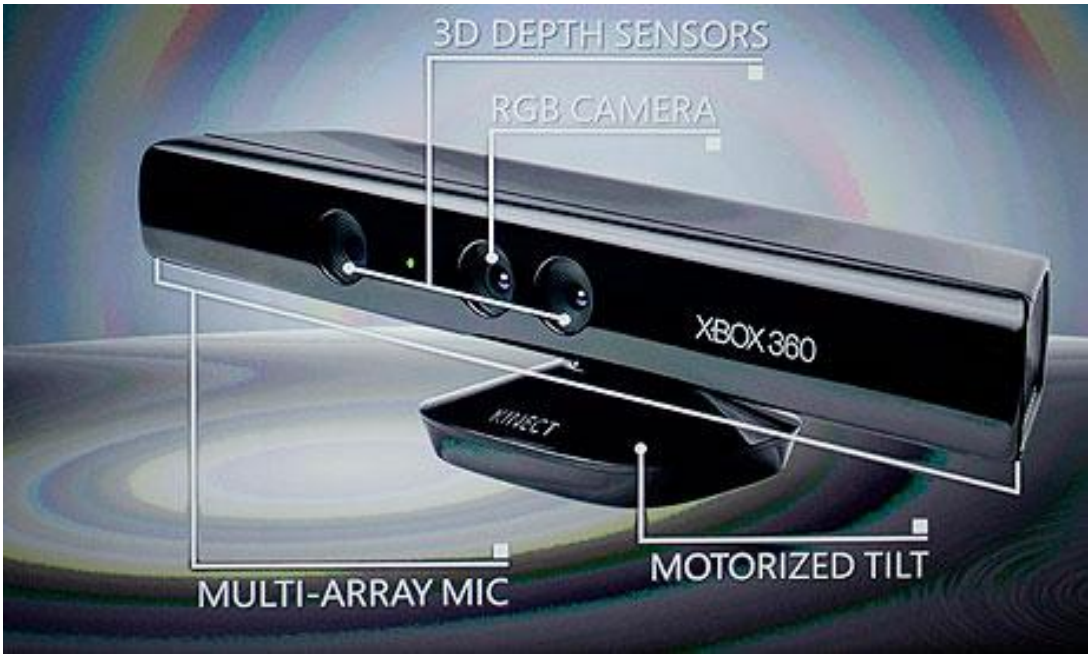
背景

從 XBOX360 Kinect 開賣以來，一般玩家在意的可能是遊戲好不好玩，而公司在意的當然就是全球的銷售量囉，對於沉醉於 Kinect 的技術狂來說，便是如何想出更好玩更有趣的運用囉!隨著技術的提升，我們的小組也上網看了許多 Kinect 運用的影片，像是電腦皮影戲、機器人導航與控制、利用 Kinect 來彈奏虛擬鋼琴、利用 Kinect 來操作 Windows 7 作業系統以及還有日

本網友可以藉著 Kinect 變身成鹹蛋超人發射光線保護地球，又或者像是七龍珠裡的孫悟空發射龜派氣功...等，都令我們感覺到非常的震撼!當然我們決定了目標之後，大家都希望我們可以藉由這些技術做出 Kinect 的應用。

專題研究方法

Kinect 簡介

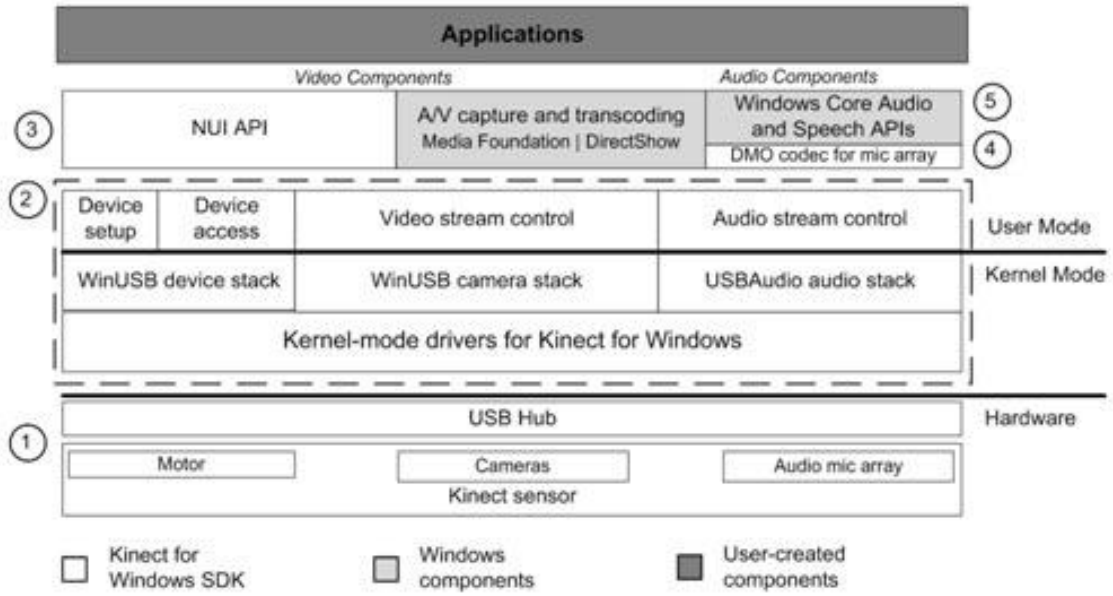


Kinect 可以取得以下三種資訊

- 彩色影像 (使用中間的鏡頭)
- 3D 深度影像 (透過圖中左右兩個鏡頭)
- Sound (透過陣列式麥克風)

底部馬達也可以根據使用者的要求而轉動

Kinect 方向 (左右各 28 度) !



1. Kinect 硬體

底座的馬達

三個攝影機 (RGB 攝影機、紅外線 CMOS 攝影機、紅外線發射器)

以及陣列式麥克風。

2. Kinect 驅動程式

核心模式下包含了以下驅動程式：

Microsoft Kinect Audio Array Control

Microsoft Kinect Camera

Microsoft Kinect Device

Kinect USB Audio

3. NUI API

4. 麥克風陣列 DMO 編碼器

5. Windows 7 內建影音處理

也就是架構圖中灰色底的部分，這就是為什麼 Kinect for Windows SDK 只支援 Windows 7 的主要原因了。

Kinect SDK 開發系統需求

- 作業系統 : Windows 7
- 軟體需求 : Visual studio 2010
  .NET Framework 4.0
- 及其他網路上所提供的 Kinect 套件
  (視使用者需求而定)

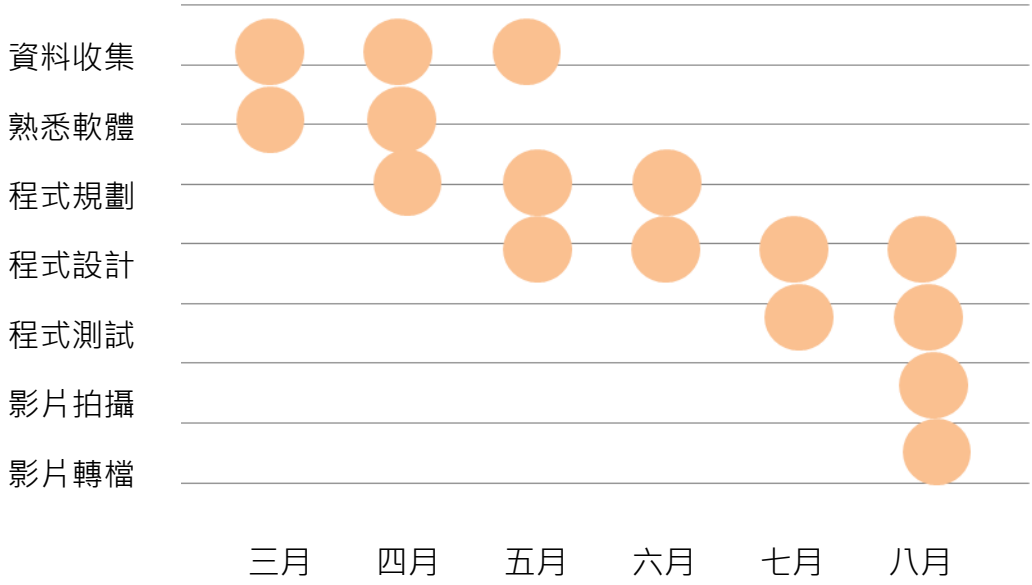
研究方法

根據官方所提供之 kinect 程式使用手冊，及網路上的資料利用 kinectSDK 官方套件進行程式的開發。

了解 kinect 的影像顯示方式後，再思考如何寫出錄影方面的程式，找出錄影方式後透過程式即可製作出 3D 影片，進而達成我們的目標。

專題開發流程

專題進度圖



專題分工

莊勝富 – 程式設計、PPT 製作、報告書製作

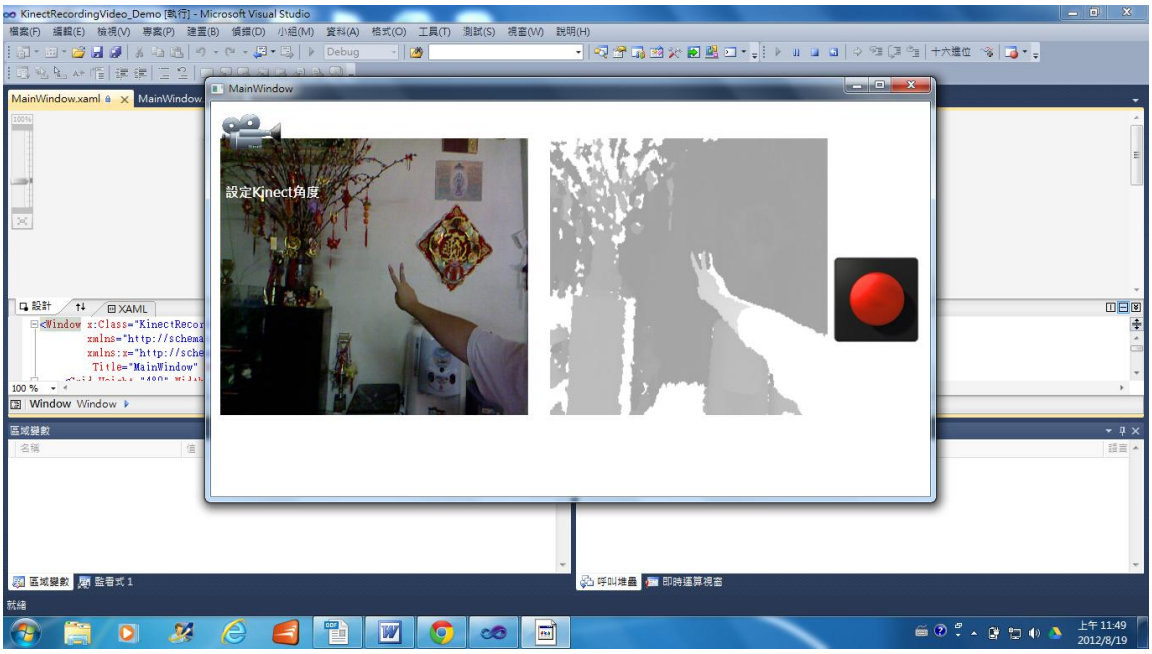
林子崔 – 資料收集、程式測試、影片拍攝製作

楊曜彰 – 資料收集、程式測試、影片拍攝製作

張登貴 – 資料收集、海報製作、影片拍攝製作

專題主要內容

下面這張圖是我們程式的執行畫面，大家可以在圖中看到左邊為彩色影像，右邊及為深度影像的顯示結果。



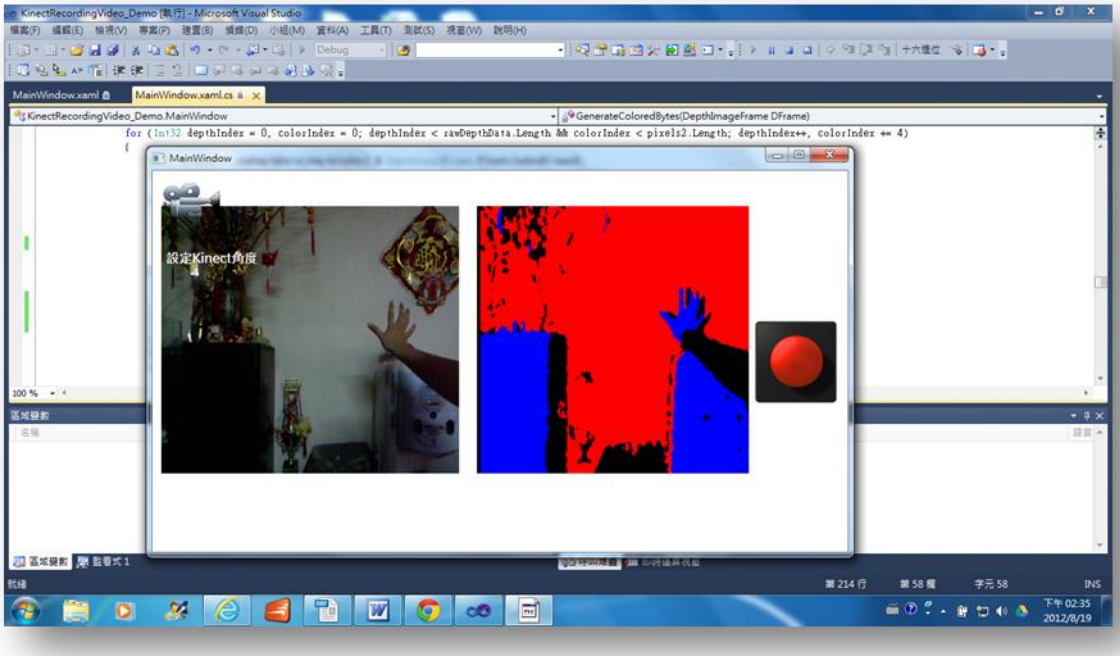
而顯示結果都是由 Kinect 官方所提供的電腦套件 - KinectSDK 加上 Visual studio 2010 所進行開發的。

對於深度影像的修改

我們將深度影像資料透GenerateColoredBytes方法轉成byte陣列便為彩色影像。

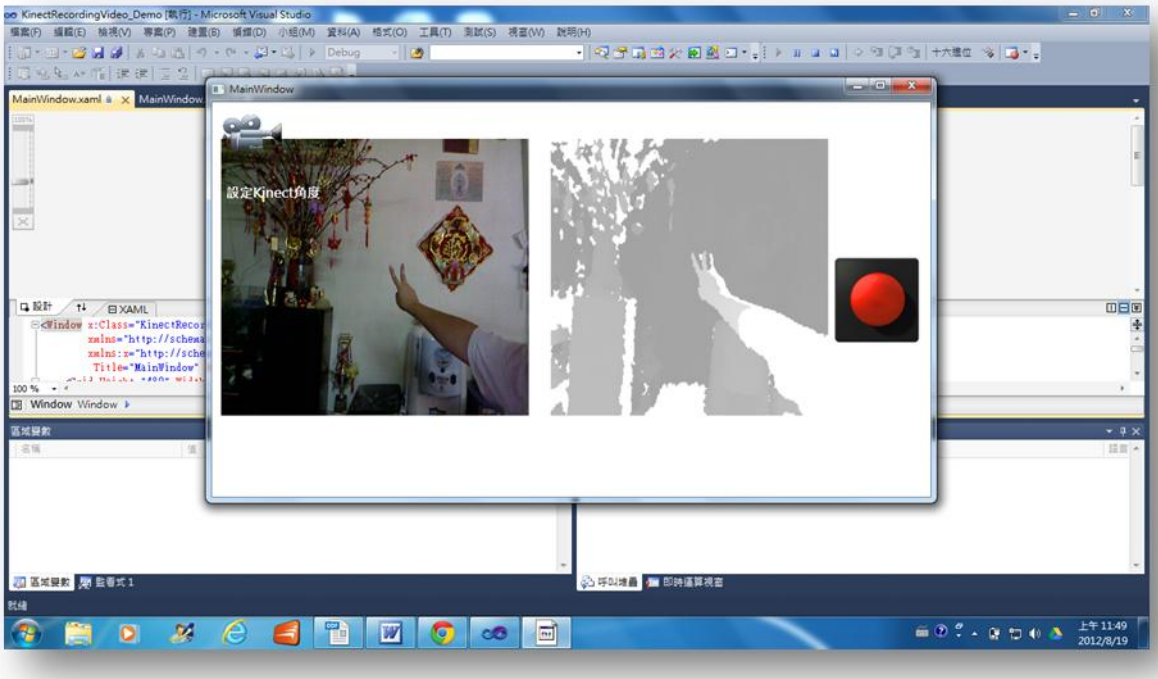
透過 GenerateColoredBytes 方法，將深度原本的 16 位元灰階圖改為 32 位元的 RGB 圖像，設定資料的 RGB 值並由一串迴圈來決定影像中像素點的顏色。

而深度影像使用CalculateIntensityFromDepth將所獲取的深度值傳入，並計算出應該回傳的色彩值(由0~255來表現由黑到白的值)顯示在深度影像上，也可以經由距離來改變某段距離的顏色。



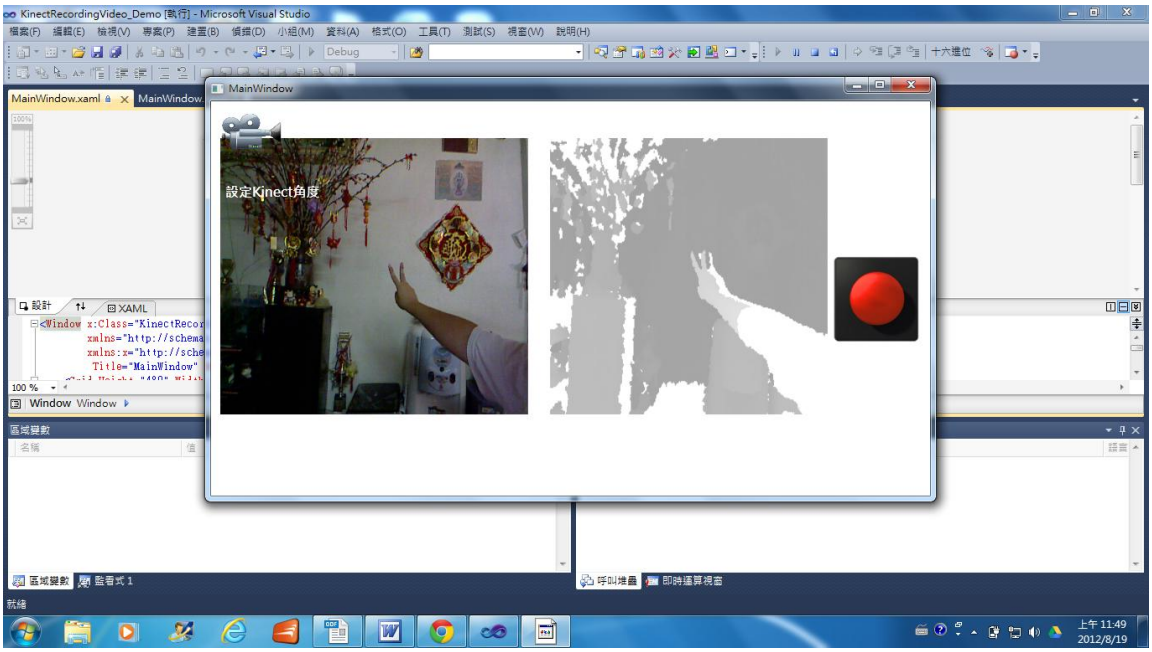
上圖可以看到經過修改過的深度影像，紅色的值為距離大於 1 公尺到 1.5 公尺，藍色的值為距離較近的位置所拍下的，距離約為 60 公分到 1 公尺，而較近的值及無法辨識的值則以黑色表示。

遭遇困難ｉ



在圖中大家可以看到 Kinect 所擷取的深度影像中，因為太近所以會造成在手臂的部分為白色，所以深度攝影機再拍攝上無法拍攝太近的東西，當然他的深度最遠距離也是有所限制。

再來，由深度圖中各位可以看到除了太近的深度無法辨識之外，較遠距離的物體也有可能因為光影問題造成深度影像異常。



另外假如物體是屬於會反光的，因為 Kinect 深度攝像鏡頭是以紅外線來測定物體的的距離，所以在遇到鏡子、玻璃或是稍微會反光的物體，他都會回傳一個無法辨識的值，像圖中的白色區塊。

彩色及深度影像錄製

在錄製影像的部分，我們原先是採用 Emgu 的套件來進行錄製影像的開發，但是使用了之後對於 Emgu 套件不熟的我只好作罷。

之後我們經過同學的幫助，使用了 system.drawing 命名空間中的其中一項類別-Bitmap。

這個類別封裝 GDI+點陣圖，而點陣圖是由圖形影像的像素資料及屬性所組成。而在 Bitmap 類別中 Bitmap.LockBits 方法可以將點陣圖鎖在記憶體內，使用 LockBits 方法鎖定系統記憶體內現有的點陣圖，便可以容易的以程式方式變更達到使用者的要求。

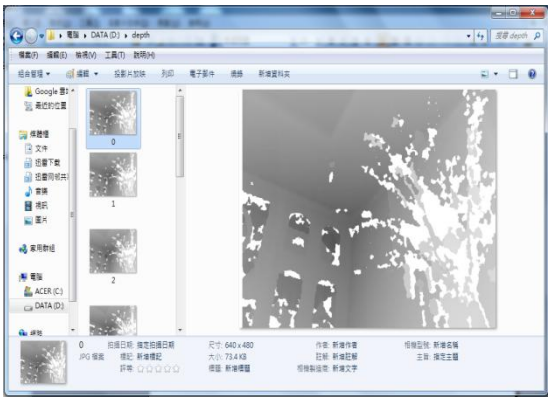
```
private void LockUnlockBitsExample(PaintEventArgs e)
        {

            Rectangle rect = new Rectangle(0, 0, bmp.Width, bmp.He
ight);
            System.Drawing.Imaging.BitmapData bmpData =
                bmp.LockBits(rect, System.Drawing.Imaging.ImageLoc
kMode.ReadWrite,
                bmp.PixelFormat);

            IntPtr ptr = bmpData.Scan0;

            int bytes  = Math.Abs(bmpData.Stride) * bmp.Height;
            byte[] rgbValues = new byte[bytes];

            System.Runtime.InteropServices.Marshal.Copy(ptr, rgbV
alues, 0, bytes);

            for (int counter = 2; counter < rgbValues.Length; coun
ter += 3)
                rgbValues[counter] = 255;

            System.Runtime.InteropServices.Marshal.Copy(rgbValues,
 0, ptr, bytes);

            bmp.UnlockBits(bmpData);

        }
```

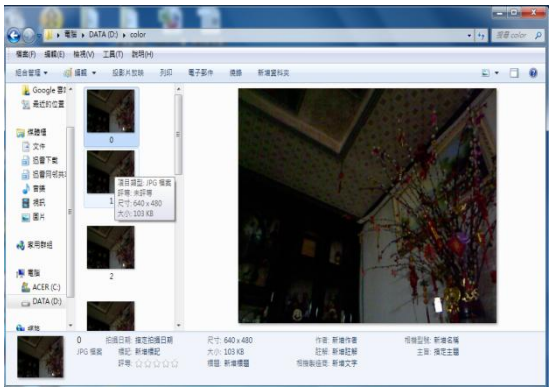
以上為 msdn 示範如何使用 PixelFormat、Height、Width 及 Scan0 屬性；LockBits 和 UnlockBits 方法；以及 ImageLockMode 列舉型別。

之後再使用 Emgu.Image 的方法來存取 Bitmap。

```
Emgu.CV.Image<Emgu.CV.Structure.Bgr, Byte> ColorFrame = new
Emgu.CV.Image<Emgu.CV.Structure.Bgr, Byte>(bmp);
```

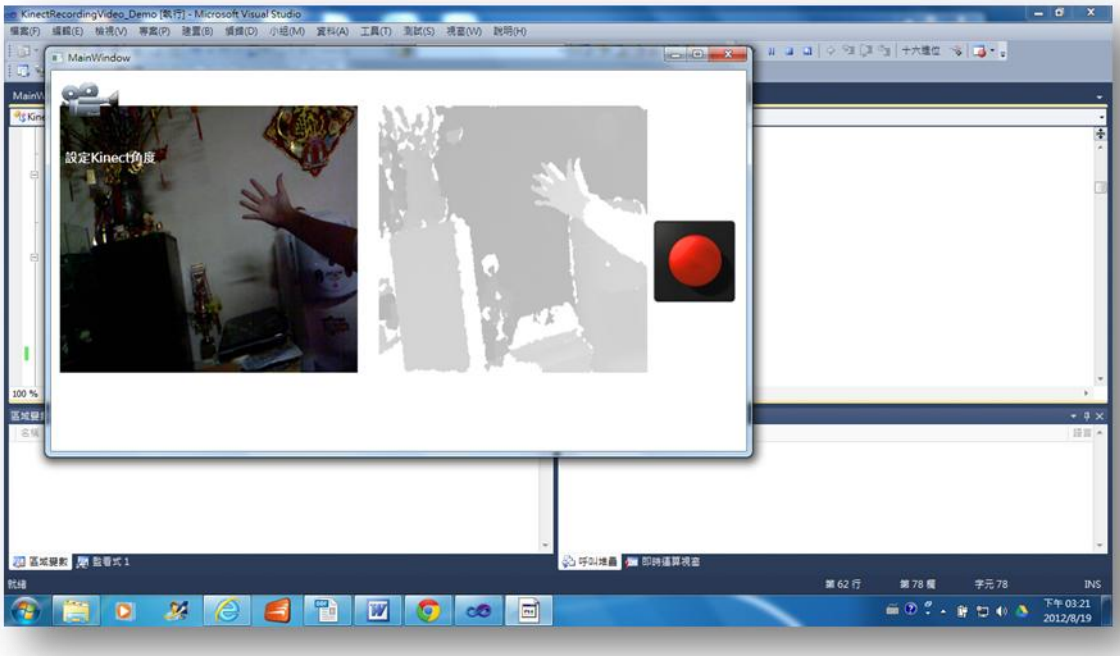
經過以上的處理，ColorFrame 就可以透過 save 的方法以我們所先前所設定影像的解析度及 fps 來進行拍攝，以每秒拍下 30 張的速率存下 640x480 解析度的圖片，之後再使用影片軟體將圖片串聯製作成影片。





遭遇困難 ii:

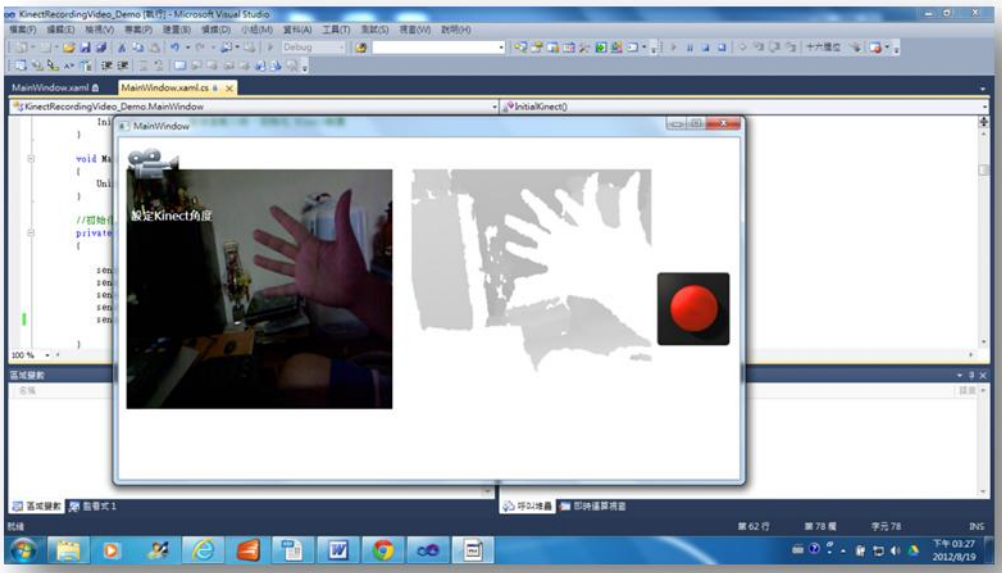
當初我們製作好錄影方面的功能之後，將所結取之圖片拿給學長看看，但是學長認為我們的深度圖有異常，在 2D+Depth 轉 3D 可能會造成錯誤，轉出不正常的影片。



由上面這張圖片所示，在我的手掌及在後面的音響部分大家都可以很清楚的看見在物體的右方會出現一塊像是影子般的白色區域。

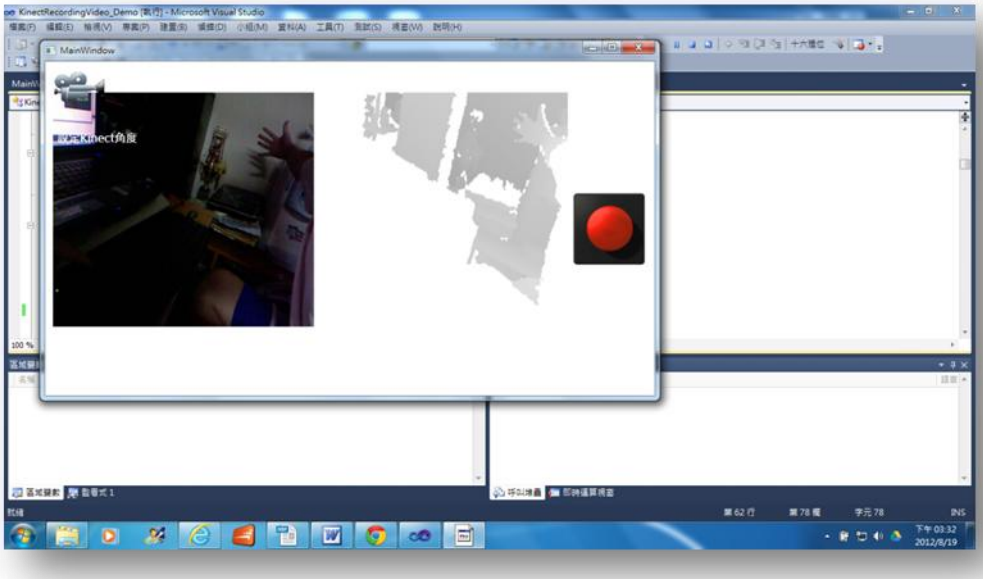
這塊白色區域跟我們之前所提到的無法辨識的區

域是一樣的，根據了解這似乎是因為 Kinect 深度攝影鏡頭在發射紅外線與接收的鏡頭設計上是分隔再左右兩邊，因此可能就會造成這種現象。而當前方的物體與後面的物體相距越遠時這個情況會非常明顯。



上圖，使用者的手指因為距離很近所以這個現象很嚴重。

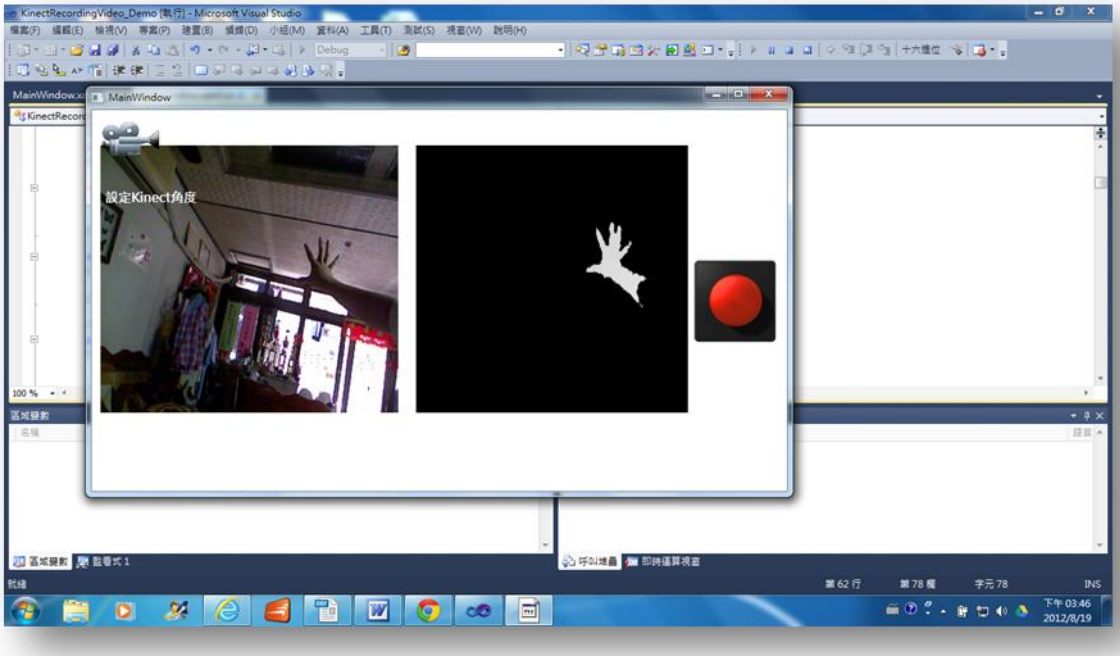
當你的手掌與背景較為接近時，也就是離鏡頭距離較遠時，這個情況會比較趨緩，但是依然存在。這塊白色區域在我們的轉圖程式中會造成很大的傷害，因為程式是利用深度資料來將原始圖片分為 9 個視角的圖，所以不該存在的區域會造成程式使用上的誤判。



反之，上圖為遠距離所拍攝結果。

解決方式

原本想說硬體上已經不能更改了，所以想從軟體著手，老師所提出的想法為我們確定他所代表的值之後看能不能將他改為無法辨識的值處理掉，但是發現他本來就是無法辨識的值所以改了依然存在，後來想說會不會在官方有提供方法解決這個問題，但在網路上遍尋不着只好無疾而終。學長也有提到若是找不到辦法只能使用影像處理的方式來修掉那塊區域，但對於程式不強的我們，實在是一項艱鉅的考驗...。



由上圖大家可以看到，此深度圖為處理過後的結果，因為我們的深度圖是根據距離來決定所顯示的顏色，所以我們決定將一定距離以上的背景設為黑色，然後一定距離的區段一樣是使用 0~255 的灰階圖表示，而那塊不能辨識的區域根據網路上的資料顯示他的距離為-1mm，因此我們也可以將他改變顏色。因此將白色區塊改為黑色後他便會跟背景值相同，不會再干擾我們的影像。

上圖為網路上擷取之深度距離計算程式片段。

評估與展望

關於這次的專題，最後大致完成了所定下的目標，但是在許多功能還是跟我們的預想有誤差!錄影程式的使用上其實對於我的電腦來說會造成不順暢的狀況，雖然說調低解析度能夠紓緩問題，但是這樣子再轉影片時解析度的不同會發生其他的問題...，還有最重要的一點就是最後製作出的成品，因為只剩下一定範圍內才有深度資訊，所以轉出的影片只有那一部分會有立體效果，算是美中不足的地方，希望在未來能將這些地方深入研究並且做改進。

銘　謝

在這個專題中我們學到了很多，學習到了在面對沒遇過的事物時，如果去應對他的方法，透過資訊的蒐集，一點一滴的慢慢研究並找出解決之道。專題製作過程中，教授的指導改變了很多我對專題製作的想法及方式 !

研究所的學長更是無怨無悔的幫助我們，犧牲了個人的時間來幫我們一起想出問題的解決之道，在這邊真的很感謝鄭芳炫教授以及研究所的 Aaron 學長以及大力相助的林建邑學長，程式方面的瓶頸解決之道以及程式設計的建議真的很謝謝教授及學長幫忙!

參考文獻

- MSDN Library 繁體中文站

  http://msdn.microsoft.com/zh-tw/library/ms123401.aspx

- TW-HKT 程式.城市.人

  http://tw-hkt.blogspot.tw/

- KinectSDK 開發入門

  http://www.cnblogs.com/eoocy/archive/2012/06/21/2557581.html

- Heresy’ s space

  http://kheresy.wordpress.com/tag/kinect/

- Kinect for windows 開發

  http://msdn.microsoft.com/zh-tw/hh367958