國立台北大學資訊工程學系專題報告

以影片為輸入之虛擬角色替換系統

專題編號:PRJ-NTPUCSIE-109-010

執行期間:2020 年 07 月 至 2021 年 05 月

1. 摘要

VTuber 指的是以虛擬角色的身分進行影音創作的創作者。通常在虛擬的環境背景中以直播的方式進行。

而此專題研究 VTuber 之技術應用在現有的影片中之成效，以一人之人臉為鎖定目標進行替換。

藉由頭部姿態估計演算法、UNITY 以及 opencv 函示庫進行影像處理以及讀寫影片。

這套系統成功在單人的座談影片、單人舞蹈影片、多人的談話影片完成了替換。

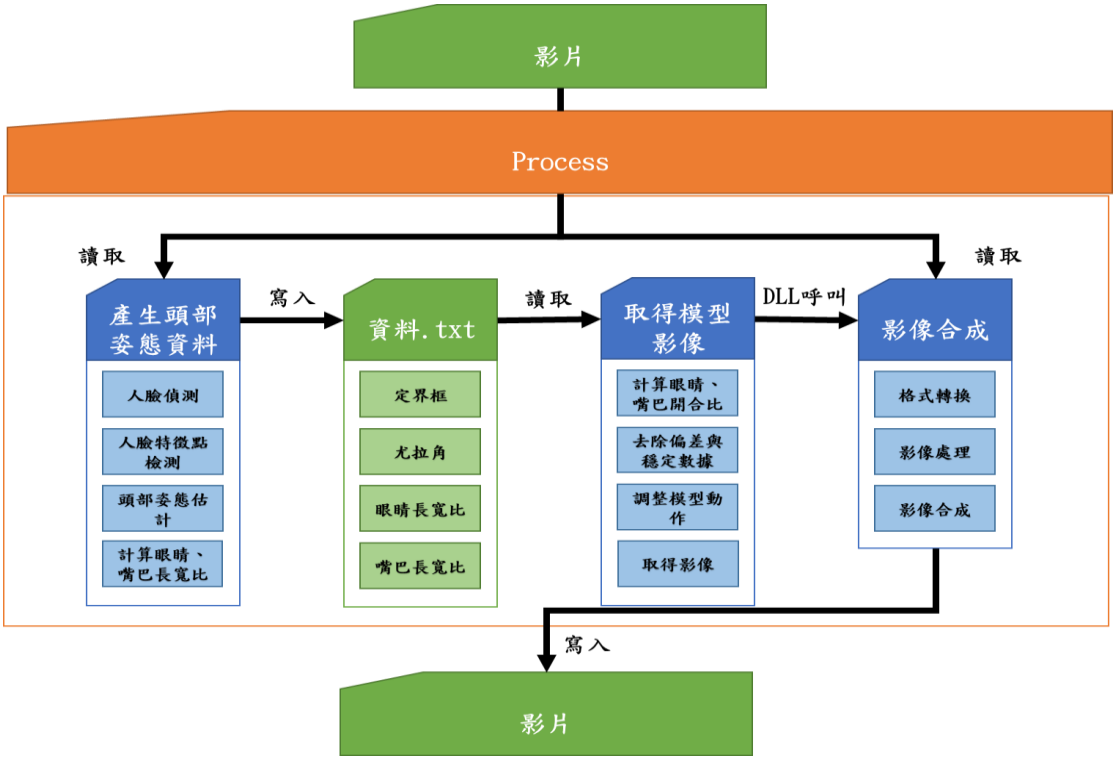
系統仍有許多發展的方向，如全身的替換、前景的保留、人臉識別等。

2. 簡介

Virtual youtuber 簡稱 Vtuber 是指以虛擬的角色進行 Youtube 相關的影音創作的創作者。在日本動漫文化的影響下，VTuber 受到了一定程度的迴響。粉絲們能與生動的動漫人物進行交流，帶給人能真實與動漫人物互動的感覺。因此有部分的實況主也加入到 Vtuber 的行列，體驗扮演虛擬角色。而 Vtuber 中多半以直播的方式進行，將虛擬的人物形象去背放在影片中一角，或是放置於一虛擬環境中，以此來跟粉絲們互動。隨著 Vtuber 逐漸火紅，在世界各地打起知名度，也吸引了更多人想用虛擬的人物形象上傳影片。此專題研究 VTuber 的技術如果應用在現有的影片中，替換影片中的人體頭部，將有如何的效果，能應用在哪些影片類型中，讓有興趣想成為 Vtuber 的人可以透過本專題研究，在不購買過多設備的狀況下達到目的。而此專題之系統目標為以影片中一人的頭部進行替換。

3. 專題進行方式

(1) 系統流程



此系統運行在 Windows 電腦上，使用 python、github、Unity、openCV、c++等語言以及工具開發，須以 pyhon 環境執行。

2.2 產生頭部姿態資料

資料處理步驟為(1)讀取影片 (2)人臉偵測 (3)68 個 2D 人臉特徵點檢測 (4) 頭部姿態估計 (5)資料寫入。程式碼主要參考 kweal23 的開源程式碼 VTuber_Unity[1]

2.2.1 讀取影片

透過 opencv[2]的函式庫的 VideoCapture 逐格讀取影片，一次取得一張影像

2.2.2 人臉偵測(Face detection)

使用開源程式碼 Face alignment[3] 提供的 SFD face detector 實現，使用函式 detect_from_image()將影像傳入後會回傳多個定界框(bounding box)，每一個定界框由左上及右下兩個點表示。系統中只選擇一個定界框使用，第一個人臉的定界框是選擇最接近影像中間的定界框。第二個之後的定界框則是選取和上一個定界框有最短曼哈頓距離(Manhattan distance)的定界框。

2.2.3 人臉特徵點檢測(Facial landmark detection)

使用開源程式碼 Face alignment[3] 提供的類神經網路 face alignment network 實現，使用函式 get_landmarks_from_image()輸入影像以及人臉的定界框，輸出 68 個 2D 人臉特徵點(圖 1)[4]

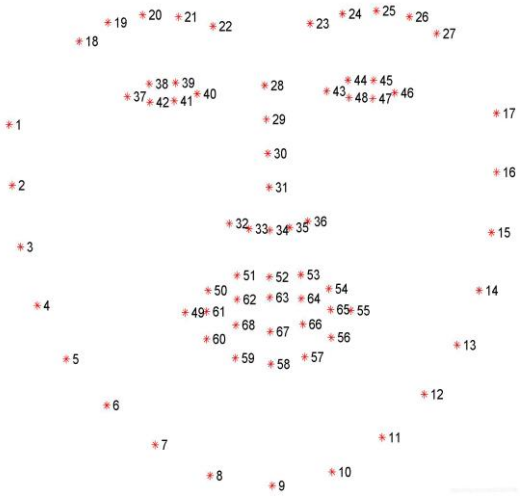


圖 1

2.2.4 頭部姿態估計(Head pose estimation)

透過 opencv[2]函式庫的 solvePnP()來完成，由 68 個 2D 人臉特徵點以及 Head pose estimation[5] 提供的 68 個 3D 人臉特徵點經由疊代算法推算頭部在世界坐標系中的旋轉與平移，其中旋轉以三個尤拉角(Euler angle)(圖 2)[6]，偏航(yaw)、俯仰 (pitch)、翻滾(roll)表示。

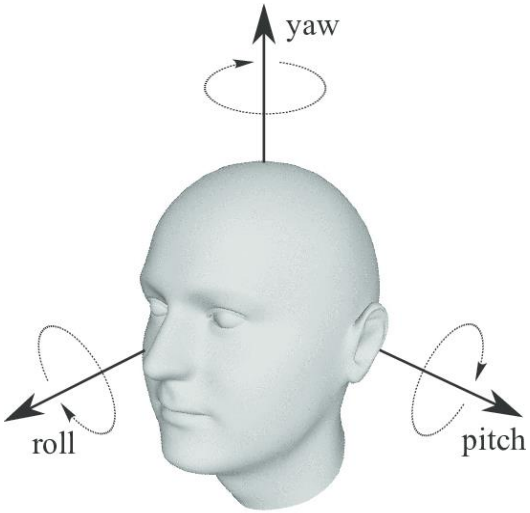
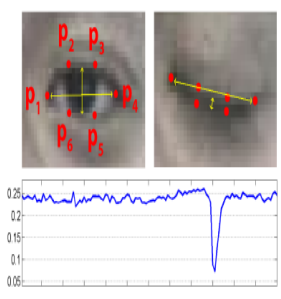


圖 2

2.2.5 資料寫入

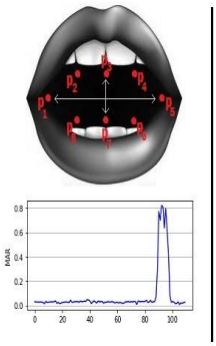
最後將人臉的定界框、3 個尤拉角、眼部長寬比、嘴部長寬比寫入到 txt 檔中，其中眼睛長寬比(Eye aspect ratio)(圖 3)[7]、嘴巴長寬比(Mouth aspect ratio)(圖 4)[8]由 2D 人臉特徵點計算。之後交由透過 Unity[9]開發的程式完成影片中頭像的替換。



$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|},$$

Source: Tereza Soukupova´ and Jan C˘ech. *Real-Time Eye Blink Detection using Facial Landmarks* at https://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf

圖 3



$$MAR = \frac{\|p_2 - p_8\| + \|p_3 - p_7\| + \|p_4 - p_6\|}{2\|p_1 - p_5\|}$$

圖 4

2.3 取得模型影像

將 txt 檔中的資料讀入 Unity 後，會先產生三個類似的模型(一個有整顆頭、一個包含脖子、一個只有臉)，提供後面合成影像時所需的資訊，並分別進行以下步驟: (1) 計算眼睛、嘴巴開合比 (2) 去除偏差並穩定數據 (3) 調整模型動作 (4) 取得模型影像

2.3.1 計算眼睛、嘴巴開合比

計算眼睛開合比跟嘴巴開合比，由於直接套用嘴巴長寬比效果不佳，因此做點調整

眼睛開合比:

```
eye_ratio = -100 / (eye_open - eye_close) * (min_ear - eye_open)
```

※(eye_open = 0.5 , eye_close = 0)

嘴巴開合比:

```
mouth_ratio = 100 / (mouth_open - mouth_close) * (mar - mouth_close)
```

※(mouth_open = 0.4 , mouth_close = 0)

2.3.2 去除偏差並穩定數據

由於在前面 2.2.4 頭部姿態估計 (Head pose estimation) 所推算的尤拉角與 Unity 中使用的存在偏差，因此對尤拉角進行調整:

```
adjusted_yaw = 0.8 * yaw
adjusted_roll = roll + 1
```

去除偏差後，利用 2.3.1 算出的開合比，與上一幀的資料比較，分別對眼睛嘴巴進行數據穩定

眼睛:

```
ratio_dif = |ratio - pre_ratio|
if(ratio_dif > 15)
    pre_ratio = ratio
```

嘴巴:

```
ratio_dif = |ratio - pre_ratio|
if(ratio_dif > 12)
    ratio = (ratio + pre_ratio) / 2
pre_ratio = ratio
```

2.3.3 調整模型動作

模型的動作調整分為

頭部旋轉:

將處理過後的尤拉角與上一幀模型使用的角度取插值

pitch = pitch * 0.4 + pre_pitch * 0.6

yaw = yaw * 0.4 + pre_yaw * 0.6

roll = roll * 0.4 + pre_roll * 0.6

，再乘上模型頭部的旋轉軸(neck_quat)，以取得頭部最終的旋轉:

neck.rotation = (-pitch, yaw, -roll) * neck_quat

眼睛、嘴巴開合:

將 2.3.2 穩定後的資料套用到眼睛、嘴巴上

2.3.4 取得模型影像

為了解決手中只有臉部定界框資訊的問題，我們需要取得三張模型的影像。第一張只有臉的影像，用來了解模型臉的大小;第二張是整個頭的影像，用來替換影片中頭部的影像;第三張是頭部包含脖子的影像，為第二張影像去掉一些不必要的部分。

2.4 影像合成

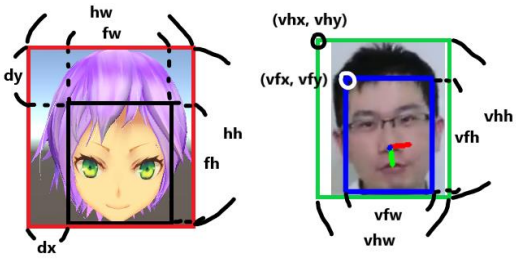
在取得 Unity[9]中模型的影像後，需要幾個步驟做最後的影像合成。(1) 呼叫 dll 並轉換格式 (2) 影像作處理 (3) 合成影像並輸出影片

2.4.1 呼叫 dll 並轉換格式

從這裡開始，影像處理、影片寫入都由 c++所寫的 dll 透過 opencv[2] 函式庫實現。首先在 Unity 中的 C# script 中呼叫 dll 中的函式，透過 pointer 將三張模型影像傳遞給 dll，使用參數中傳遞的長寬創建 Mat 後再使用 memcpy()將影像存入 Mat 中，最後在透過將影像從 RGBA 轉換成 BGRA、上下翻轉，從 Unity 的 Texture2D 的格式轉換成 opencv[2]的格式。

2.4.2 影像處理

首先先將整個頭的影像以及包含脖子的影像透過交集運算，並找出模型頭部的定界框裁切。再來在只有臉部的影像中找出模型臉部的定界框。透過頭部的定界框、臉部的定界框、影片中人臉的定界框以比例計算影片中大概的頭部之定界框，



vhx = vfx - dx * vfw / fw

vhy = vfy - dy * vfh / fh

vhw = vfw * hw / fw

vhh = vfh * hh / fh

並將裁切下的模型頭部影像調整大小成影片中頭部之定界框的大小。

2.4.3 影像合成與影片寫入

由 opencv[2]函式庫的 VideoCapture 取得影片的影像後，取出模型頭部影像的 alpha 通道作為遮罩，根據頭部之定界框以及遮罩影像覆蓋到影片的影像中。最後再透過 opencv[2]函式庫的 VideoWriter 將影像寫入成影片。

(3) 主要困難與解決之道

影像輸出

為了輸出影片，我們首先找到的方法是 Unity[9]的套件 Unity Recorder，雖然成功的輸出了影片，但是它只能在 Unity 的編輯模式下執行，因此卻導致我們無法建立執行檔。所以後來嘗試透過 c++的 opencv[2]函式庫提供的 VideoWriter 實現，但是因為 Unity 的 script 是以 c#撰寫，所以要透過動態程式連結庫(dll)的技術呼叫 c++的函式。

影像合成

影像合成的目標是頭部影像，但因為 Face detection 提供的只有臉部的定界框，因此需要依靠模型臉部的定界框以及模型頭部的定界框為比例，推算出影片頭部的定界框。這個方法對於短髮的頭部算是解決了，但是長髮與戴帽子的頭部卻還未解決，仍會有遮擋不住的問題。另外為了去除模型脖子後方不該出現的頭髮，需要另外取得含有脖子的影像，與沒有脖子的影像作交集運算。

數據浮動

因為人臉定界框的數值會上下浮動，造成畫面上的頭部會顫抖。為了解決此問題，使用向前比對的方式穩定。

```
dif = Σ abs(box[i] - pre_box[i])
if(dif < 10) box = pre_box
elif(dif < 15) box = (box + pre_box)/2
```

人臉追蹤

在多人性質的影片中，需要鎖定某一張人臉，起初的構想是在連續環境下，因此選擇了簡易的算法，透過定界框在影像中與前一張影像中的定界框之距離進行簡易追蹤。在連續環境下確實可行，但是如綜藝節目，常常會有鏡頭切換，導致透過單純影象空間上的距離判斷無法準確的跟蹤。可能可以透過人臉辨識系統解決，但系統尚未導入人臉辨識系統，因此還未解決。

4. 主要成果與評估

第一部測試用的影片為蒼藍鴿的蒼藍鴿聊醫學[10]，影片是以座談的方式進行，影片中的人臉全程皆能被人臉偵測系統偵測到，使系統能鎖定人臉，因而不被後續出現的人臉干擾。而人臉偵測的定界框數據浮動的影響透過向前比對穩定了許多。算是有一個不錯的成果。

第二部影片是從網路上截錄的韓國舞者跳舞影片[11]，相較於第一部，這部影片裡的人有更大的移動幅度及轉身，系統也能捕捉到人臉並進行替換。

第三部影片是木曜 4 超玩的下班去吃飯[12]的一部份片段，內容為主持人與兩位來賓的談話。這段影片顯示在多人情況下，系統將鎖定在中間的人臉作為替換對象。

5. 結語與展望

這個專題中透過 python、Unity、c++ dll 等多個程式語言以及開發工具互相結合，建構出一套系統，藉由 UI 整合，一鍵執行完成。

此系統仍有許可以更精進的部分，希望能加入人臉辨識，能更良好的應用在多人的影片。加上去動態模糊的處理，使系統在動態影片中有更好的效果。或是全身姿態的偵測能使虛擬角色完整的走到現實。以及前景的偵測與保留，使系統能更完善的應用在物品介紹的影片中。

6. 銘謝

感謝指導教授在專題中一步步為我們指引方向，引導我們研究。並在我們做不好時提出指正，讓我們能夠知道如何修正。

感謝 AI 葵提供的 VTuber 教學影片，讓我們了解如何實現 VTuber 的系統。

7. 參考文獻

[1] kwea123, VTuber_Unity
[2] Intel Corporation, OpenCV
[3] Adrian Bulat, face-alignment
[4] Adrian Rosebrock, Facial landmarks with dlib, OpenCV, and Python, 2017
[5] Yinguobing, head-pose-estimation
[6] Alberto Fernández Villán, Driver Distraction Using Visual-Based Sensors and Algorithms, 2016
[7] Luka Cehovin, Rok Mandeljc, Vitomir Struc (eds.), Real-Time Eye Blink Detection using Facial Landmarks, 2016

[8] Akshay L Chandra, Mouse Cursor Control Using Facial Movements — An HCI Application, 2018
[9] Unity Technologies, Unity
[10] 倉藍鴿的醫學天地, 減肥、抗老化必看!「間歇性斷食」可望成為人類救星? | 蒼藍鴿聊醫學 EP116
[11] 韓國舞者跳舞影片, 190810

처음처럼 - 달라달라 & 뿜뿜 (유진)

직캠 by 수원촌놈

[동대문구구민회관 청춘음악회]

[12] 木曜 4 超玩, 《下班去吃飯第十二集》到底～哪個東西才是可以吃的啦？！feat.動力火車