

Group 10

R09921075 賴正諺

R09921081 趙天易

R09525125 林品均

3DCV Final Report

Reconstruct 2D to 3D from singal RBG hair image

一、 Motivation

目前 3D 人臉重建已經擁有不錯的效率及結果，然而，大多數的 3D 重建主要是注重於人臉的構造，細部的紋路及表情等等，如牙齒，頭髮等部分則較少著墨。因此我們決定以頭髮重建為這次的主題。

二、 Method

我們主要的方法使用 Single-View Hair Reconstruction using Convolutional Neural Network^[1] 這篇於 ECCV2018 發表的 paper 的內容。給定一張大頭照，輸出 3D 的頭髮模型。整個方法可以分為四個部分：準備 Dataset、Preprocessing、Model、Loss。

Dataset：

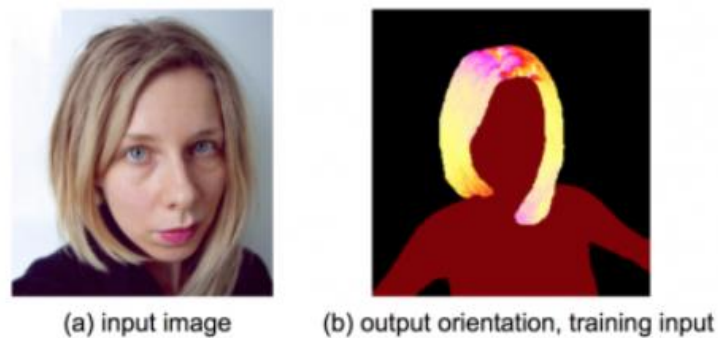
首先以 USC-HairSalon^[2] 提供的 343 個基礎頭髮模型作為整個 Dataset 的基底，這個頭髮模型是以髮絲為單位，每個模型有 10000 個髮絲，每條髮絲取樣 100 個點。接著隨機混合兩種髮型的髮絲，產生新的髮型，最終產生 2400 個髮型，每個髮型再產生五種不同的 view。

Preprocessing :

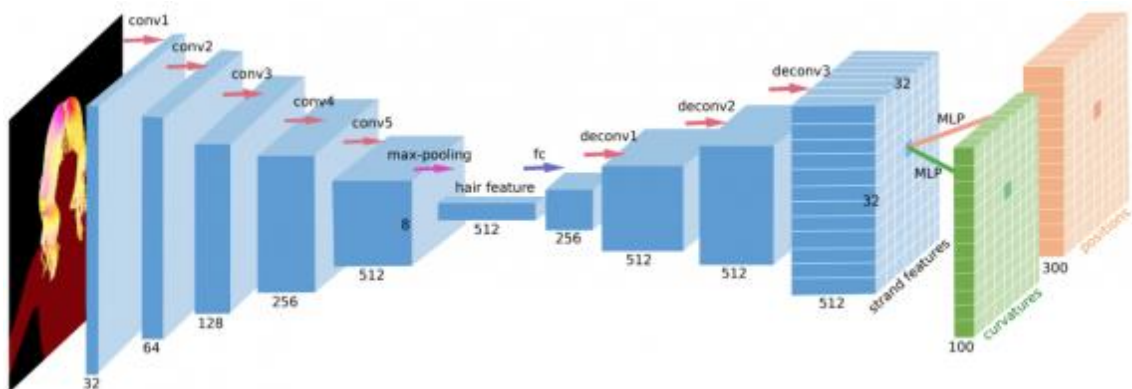
給定一張大頭照後，首先需要將頭髮部分獨立出來，這個部分可以使用將物體邊框分別獨立的 PSPNet_[3]來達成。分離後，針對頭髮的部分取得

Orientation(color coded, 如附圖所示)、Visibility map 與 Curvature。

Orientation image 將作為模型的輸入，而 Visibility map 與 Curvature 將於訓練時計算 Loss 時使用。



Model :



如附圖所示，orientation image 作為 Model input。Encoder 由五層 convolutional layer 組成，最後透過 Max pooling 得到一 512 大小的 hair

feature，然後經過由一層 FC 及 3 層 deconvolution 組成的 Decoder，輸出一個 1000x512 的 Tensor，每一個 512 長度的 vector 代表一條髮絲。最後分別經過兩個 MLP，output 出每條髮絲 100 個 sample points 的位置與 curvature。

Loss：

Loss 一共分為兩個部分，由 Position loss 與 curvature loss 組成。Position loss 即為模型 output 中的 sample points 與實際上的距離，如下式所示，此處的 N 為髮絲總數，此處為 1000；M 為每條髮絲的 sample point 數，此處為 100。此外，權重 ω 會根據 visibility map 的值而有所變化，被頭擋住的髮絲就算有偏差也較不容易注意到，因此 weight 較小，而看的見的髮絲有偏差就很容易察覺，因此將 weight 設大。

$$L_{pos} = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} w_{i,j} \|\mathbf{p}_{i,j} - \mathbf{p}_{i,j}^*\|_2^2$$

$$w_{i,j} = \begin{cases} 10.0 & s_{i,j} \text{ is visible} \\ 0.1 & \text{otherwise} \end{cases}$$

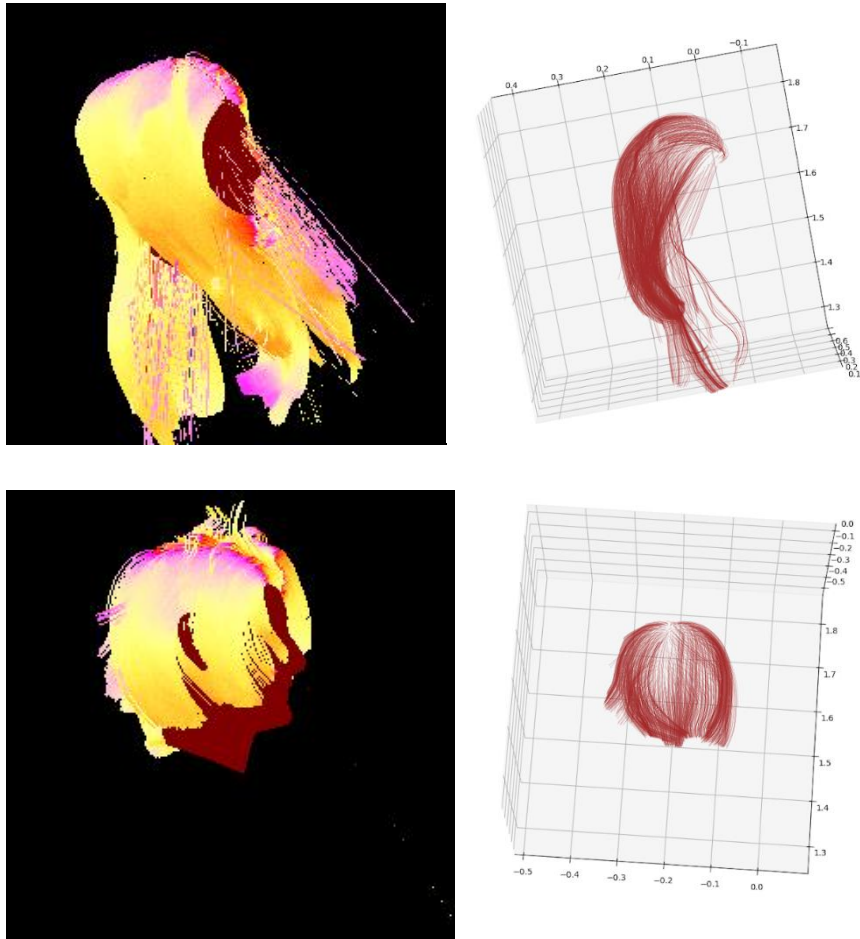
而 Curvature Loss 則如下式所示，我們希望髮絲的 curvature 能夠越接近越好，如此才能使產生的髮型形狀更相近。

$$L_{curv} = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} w_{i,j} (c_{i,j} - c_{i,j}^*)^2$$

而 Total Loss 即是 Position Loss 與 Curvature Loss 的加總。

三、 Result

如圖所示，左圖為經過 preprocess 後的 input orientation image，右圖則為 model 產生的頭髮模型。



訓練是在 RTX 3070 Mobile 上進行，一共訓練 25 個 epoch，總訓練時數約為 40 小時左右。

Postion error : 0.366174

Curvature error : 8.415472e-05

四、 Discussion & Future Works

產生的模型外觀上看起來還算可以，調成相近角度後形狀也大致上算接近。我們原先計畫將產生 model 後的頭髮模型與其他人臉重建(例如: Learning an Animatable Detailed 3D Face Model from In-The-Wild Images_[4] 中的模型)後組合起來，但遇到了幾個困難，包含：頭髮位置對不起來、沒有考慮人臉本身的位置導致頭髮會與人臉在同一位置，訓練時間過久(這也是更換主題的主因)很難重複嘗試等等，因此最後並沒有順利完成這個部分。針對以上的問題，我們認為可以透過在 data 中增加起始位置資訊，增加 Loss 的設計(是否與人臉的範圍衝突)等等來改善。此外，本次並沒有考慮到模型大小或者 render 效率的問題，是否能壓縮模型，改善處理效率，甚至達到 real time 等等，是值得繼續研究的方向。

五、 References & Links

[1] [Single-View Hair Reconstruction using Convolutional Neural Networks](#), Yi Zhou , Liwen Hu , Jun Xing , Weikai Chen , Han-Wei Kung , Xin Tong , and Hao Li, ECCV 2018

[2] [USC-HairSalon: A 3D Hairstyle Database for Hair Modeling](#)

- [3] [Pyramid Scene Parsing Network](#), Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, Jiaya Jia, The Chinese University of Hong Kong, SenseTime Group Limited
- [4] [Learning an Animatable Detailed 3D Face Model from In-The-Wild Images](#), YAO FENG, HAIWEN FENG, MICHAEL J. BLACK, TIMO Bolkart
- [5] [HairNet DataSetGeneration](#)
- [6] [HairNet](#)
- [Model weight](#)
- [Dataset](#)(pswd: ezh6)

Optical flow-motion

1.Motivation

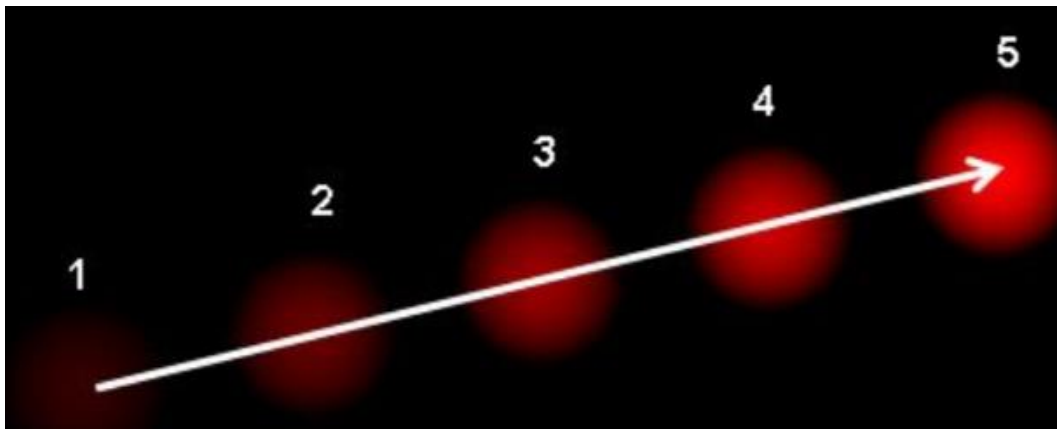
隨者顯示器設備的發展，高幀數的畫面能夠提升使用者的感受，已是非常的重要。

尤其是在各種慢動作的畫面中，高幀數能夠使畫面更為的流暢。那麼此次研究主要

是在如何提升幀數、在畫面中補齊缺失的幀數，以及評估各式方法對於補幀的影響。

2.Method

再進行補幀之前首先要能偵測光流的移動，後才能得知畫面中物體的移動方向，如圖(1)所示。



圖(1)光流移動位置

經由像素 1，和像素 5，可推知 forward flow 後先推估像素 3 的位置後，進行 backward warp，再以(1,3),及(3,5)推估(2,4)的位置。

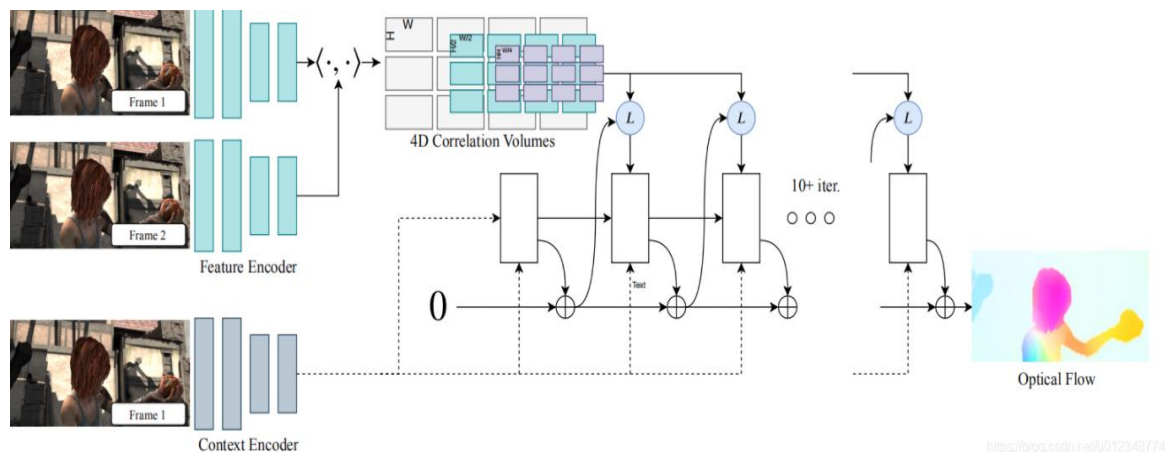
我們將主要分為三個部分:1. 以 opencv-gunner farneback 算法來得之光流的流向後進行 backward warp 補幀。 2. 以 raft(**Recurrent All-Pairs Field Transforms**)深度神經網路推估光流後，進行 backward warp 補幀 3. 由於經過 warp 後的補幀圖像會產生扭曲，故選擇以 GAN(**Generative Adversarial Network**) 網路來嘗試消除扭曲的現象。

2-1. Raft(Recurrent All-Pairs Field Transforms)

Raft 的主要架構如圖(2)所示

Raft 主要有 3 個部分：

- 1.特徵萃取: 將圖像 1 跟 2 的特徵萃取出來，以便進行 mapping, context encoder 則是將圖 1 的特徵提取出來，用來計算 loss 更新光流。

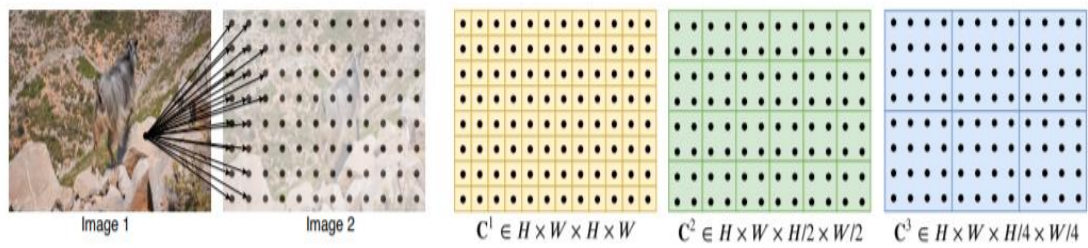


圖(2) Raft 架構圖

2. correlation volumn :主要就是將從 frame1 及 frame2 萃取出的特徵進行相似度匹配，這裡我們經過 4 層 pooling，並將每層都輸出，這樣確保了影像局部細微的運動，到大範圍的運動都可以偵測到。如圖(3)所示

$$C(g_{\theta}(I_1), g_{\theta}(I_2)) \in \mathbb{R}^{H \times W \times H \times W} \quad g_{\theta}$$

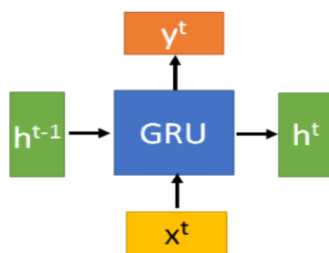
是匹配特徵, i_1 和 i_2 則是 frame1 和 frame2



圖(3) 影像特徵由 2D 映射到 4D 中

3. 則是通過 GRU (Recurrent Neural Network)RNN,來進行疊代

GRU 就是 RNN 的一種, 主要就是考慮前次網路的狀態。有一個輸入 x_t , 和上一個節點傳遞下來的狀態 (hidden state) ,結合 x_t 和 x_{t-1} , GRU 會得到節點



的輸出 y_t 和傳遞給下一個節點的狀態 h_t 。如下圖所示

2-2 GAN(Generative Adversarial Network)

由於 warp 後的影像會產生扭曲的現象, 故我們想以 GAN 網路來消除扭曲, 在此參考[1],DeblurGan 的方式, 並以 GROPO DATASET 以及自行創造的

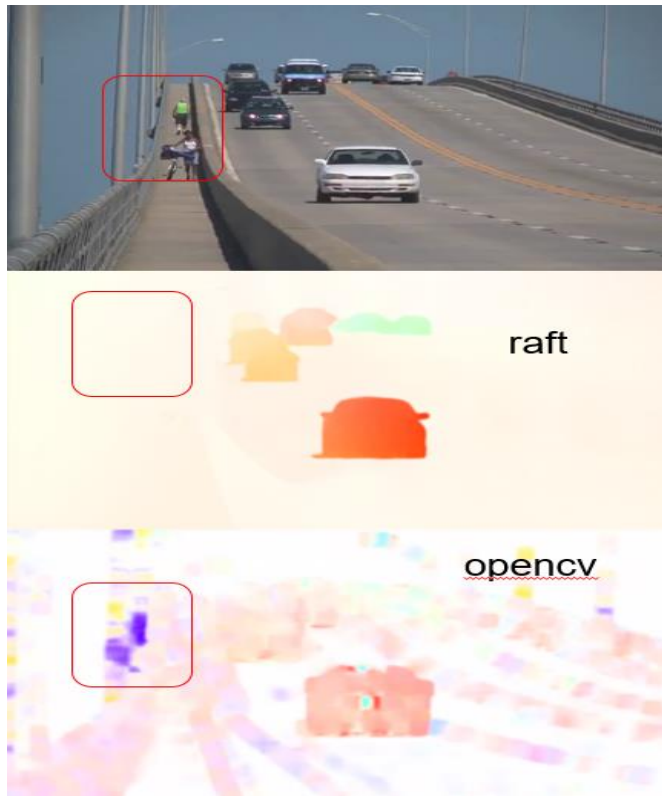
DATASET 約(13000 對影像)來訓練網路。

3 Result

	Opencv	Raft	Raft-Gan
PSNR	46.5	44.7	36.5
SSIM	0.986	0.93	0.92

	Opencv-Gan	Super-slomo
PSNR	38	35.9
SSIM	0.982	0.9899

我們評估了 dataset 中的車流影像，並進行補幀，總計在 2 張 frame 之間插入 7 張 frame 並使 FPS 提升四倍，後跟[2]進行評比，可以看出加入 Raft 跟 gan 會使得效能下降。



圖(4)光流圖

4. Conclusion

1. 可以發現到加 Raft 後會使得效果下降，那光流圖如圖(4)所示，可以發現到在這個車流資料中車道旁的行人會移動，但是 Raft 卻將它的光流給過濾掉了，所以導致 warp 後無法顯現人物像素移動的效果，故結果會較差。

2. 首先 deblur gan 會使得圖像的分辨率上升，而這個車流資料中的影像分辨率原始就比較低，第二 我們使用的別人的 pretrain model 在加一些自行處理的 data (將原始圖像的像素隨機移動後 warp)，而原本的 pretrain model dataset 是將圖片進行隨機 interpolation 來產生模糊影像，而這樣產生的模糊影像並沒有

warp 的效果，所以較無法使用在消除 warping 後扭曲的現象。如圖(5)所示，

紅圈圈選處產生扭曲殘影的現象，並無法有效地消除。



圖(5)Deblur Gan 前後對比圖

3.由結果可以發現，如果不已 end to end 的 model 來進行補幀，而是先求出 optical flow 後再分別進行 warp 補幀，那最終的效果，就有很大一部份取決於 warp 的算法，而此次研究我們只使用了單純的 forward flow 以及 backward warp。相較與多數 paper 不同的是，我們提出了以 GAN 的 model 來改善 warp 後扭曲的圖像，雖然結果不盡理想，原因可能也如同(2)所猜測。未來我們可以在 warp 的方式 以及解決 occlusion,和 GAN 的影像後處理地方多做琢磨

5.Reference

- [1] DeblurGAN-v2: Deblurring (Orders-of-Magnitude) Faster and Better, Orest Kupyn, Tetiana Martyniuk, Junru Wu, Zhangyang Wang, In ICCV 2019
- [2] Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan

Yang, Erik Learned-Miller, Jan Kautz, CVPR 2018

[3] RAFT: Recurrent All Pairs Field Transforms for Optical Flow, Zachary Teed
and Jia Deng ,ECCV 2020

[4] FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks ,
Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey
Dosovitskiy, Thomas Brox, CSCV

[5] Softmax Splatting for Video Frame Interpolation, Simon Niklaus, Feng
Liu,CVPR2020

分工:

R09921075 賴正諺 :報告、report、optical-flow-motion

R09921081 趙天易:收集 dataset、影像後處理

R09525125 林品均:report、整合 hairnet