

方案設計書

隊伍名稱：w

模型介紹：

首先我們確認了比賽的執行時間限制：本次比賽共有 100 個場景，並且要求在 48 小時之內訓練完成，圖像生成也要在 1 小時完成。因此，我們決定採用三維重建及生成都很快的方法：instant-ngp。

原代碼採用 cuda 編寫，並不容易直接套用到本次競賽中，因此我們找到了 pytorch 版的復現 ngp_pl (https://github.com/kwea123/ngp_pl)，並以此作為 baseline。

實現細節：

基於 baseline，我們總共做了 3 個對於本次比賽的改進。

1. 使用 Matting 提取人物領域
2. 優化相機外參
3. 優化不同相機所造成的色差

以下將對各點進行詳細介紹。

1. 使用 Matting 提取人物領域

因為比賽只要求重建人物，因此背景不重要，我們使用 PaddleSeg 來提取人物領域，並只對人物領域進行訓練。

Matting 前後的結果如下圖（左：原圖、右：Matting 結果）：



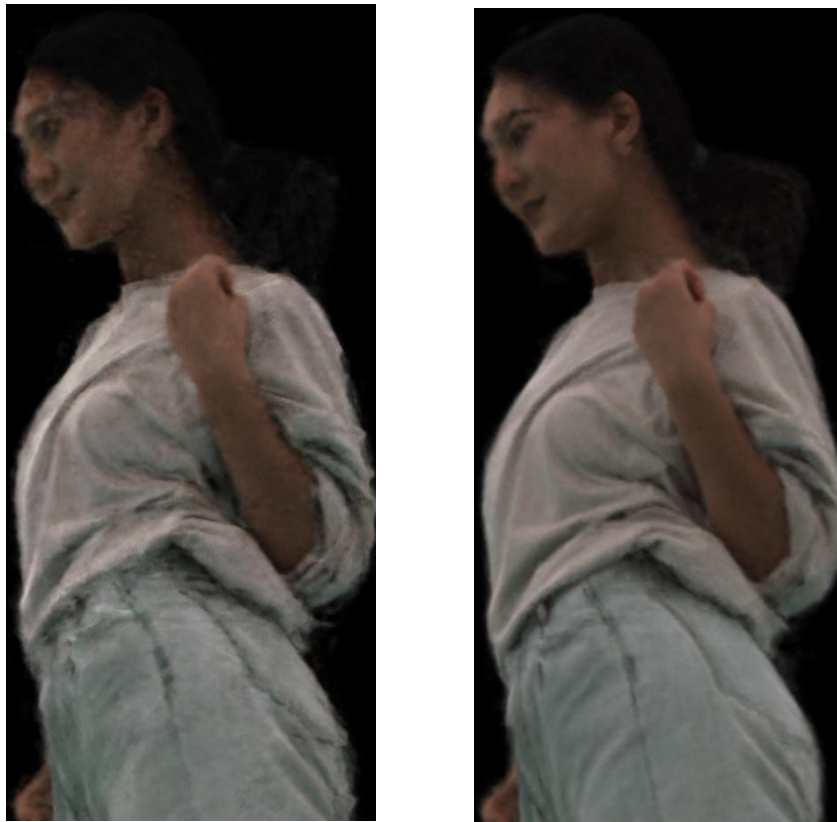
空白的部份為 $\alpha=0$ 。在大部分的場景中，Matting 效果非常好，然而在有些場景（例如有人戴帽子、反光）等等的惡性條件下，Matting 會出錯。但我們並未對這些場景進行手動調整。

訓練時，我們如原 NeRF 一樣對所有像素進行 rgb loss 計算，另外為了正確訓練背景的 α 為 0，我們另外加入了一個 alpha loss，讓每個像素的 α 值也接近 matting 出來的 α 值（請參照 losses.py）。對於有人物的地方($\alpha>0$)，我們把 rgb loss 乘以 5 倍，讓模型更有效率重建這個領域（因為背景占的領域太大）。

2. 優化相機外參

基於 1 的訓練結果，我們發現有些場景(F1_06 及 F2_07)訓練出來的結果相較其他場景特別模糊，我們推測是因為外部參數沒有很正確的被校準出來。因此，對於這些場景，我們同時優化重建以及外部參數（參照 train.py 中的一optimize_ext）。以下給出一個優化外部參數前後的例子：

（左：無優化外參、右：有優化外參）



可見有優化外參的視覺效果比無優化來的好。

3. 優化不同相機所造成的色差

最後，我們還發現每個相機的回應函數(Camera response function)相差很大。即一個相同顏色的物體，在不同相機裡照出來會呈現不同的顏色。如下圖這個訓練場景，同樣的衣服在不同相機中，左邊的圖呈現帶紅色、右邊的圖呈現帶綠色。



這種顏色的不一致並非是由視角(NeRF 當中的 view direction, \mathbf{d})造成的，而是因為**照相機特性**不同造成的。我們發現如果不做任何處理直接訓練的話，會造成新視角的生成圖像顏色非常不一致（有些地方紅有些地方綠）。因此，我們導入了 NeRF in the wild 中提到的 appearance embedding vector，即我們對於每台照相機，都給它指定一個 feature vector，然後把這個 feature vector 也輸入到 rgb_net 中生成最後的顏色。在生成新視點的時候，我們指定一個相機（1 號），並始終以這個相機的 feature vector 去生成顏色。我們發現這樣做不僅不同視點的顏色變得一致，同時也少了不同顏色混雜的情形。

代碼運行流程：

請參照 https://github.com/kwea123/ngp_pl/tree/mgtv