3DCV-Final: Depth Estimation of Dynamic Objects

Group 19
R10922042 柯宏穎 R10922077 蔡秉辰 R10922095 黃秉迦

Introduction

深度預測已被研究多年，但至今仍只有少部份的是研究在動態物體上的。在現實社會中，大多數的物體都是會移動的。舉例來說，現在常會出現在家裡的掃地機器人，通常放下去後主人也不一定會離開家門，機器人也不能當作沒看到就撞上去。因此，我們試圖來解決這類的問題，有幾篇論文已有不錯的成果，但需要的限制較多，有許多我們無法直接得到的參數，如相機內外參等，為了實現在只給予單一影片的情況下能得到我們想要的深度圖，我們合併了兩種模型，用來獲取一個初步的深度圖與內外參，再逐步去用得到的 optical flow 與 scene flow 來調整最後的結果，來得到更加穩定且方便的 pipeline。

Baseline Model

我們將 Google 2021 年的論文 - Consistent Depth of Moving Objects in Video 作為我們的 baseline model。這篇 work 由兩個重要的 components 組成，分別是 pretrained single depth model 和 train from scratch 的 scene flow model。在已知 optical flow 、內外參和 motion segmentation 的條件下，藉由兩個 consistency 進行 self-supervised training 以在深度預測達到更好的表現。

Components

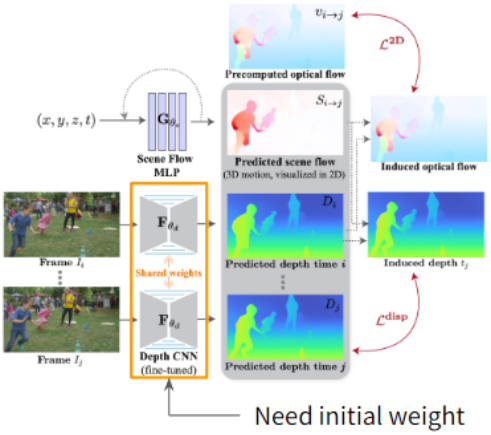
- Depth model: 對於每張輸入的圖片輸出預測的深度，會使用已經在 MiDaS-v2 上 pretrained 後的 weight 進行 fine-tuning
- Scene flow model: 對於每個輸入的三維點座標和時間戳輸出其短時間內的位移。由於是 train from scratch ，所以訓練時會先把 depth model freeze 住進行 5 個 epoch 進行 warmup。

Consistency

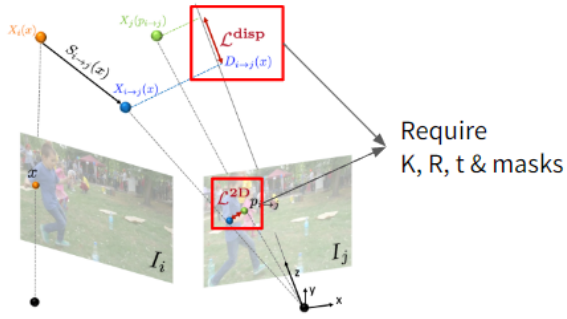
給定不同時間點的兩張 frame F_i 和 F_j。令 F_i 上的一點為 x，透過 depth model 我們可以得到 x 的深度，並透過 F_i 的內外參將 x 投影至三維坐標 $X_i(x)$。透過 optical flow 我們可以得知 x 在 F_j 上的對應點 $P_{i\to j}$，並利用前面的方法將 $P_{i\to j}$ 投影至三維坐標 $X_j(P_{i\to j})$。最後，透過 scene flow model 我們可以推測 $X_i(x)$ 經過一段時間後的位置 $X_{i\to j}(x)$。

- Depth consistency($\mathcal{L}^{disp}$): $X_{i\to j}(x)$ 和 $X_j(P_{i\to j})$ 是我們利用不同方式推測出 x 在時間點 j 的三維座標，我們希望兩者在時間點 j 的視角中 depth 越接近越好。
- Scene flow consistency($\mathcal{L}^{2D}$): 我們希望 $X_{i\to j}(x)$ 利用 F_j 的內外參投影到 F_j 上後與 $P_{i\to j}$ 越接近越好
- Motion segmentation: 幫助我們對動態和靜態的物體進行不一樣的約束，藉以讓 model 對不同狀態的物品有更深的認識

Model architechture:



Loss illustration:



Our Method

Motivation

雖然 Consistent Depth of Moving Objects in Video 作為 SOTA 已經有很好的表現了，但我們認為他仍有許多可以進步的空間

1. Single-frame depth model 使用了 pretrained 過的 weight，所以最直覺的想法是如果我們用更好的 pretrained model，相信整體的 performance 會變得更好
2. 內外參的取得在現實情形中是非常困難的事，因此我們希望能利用一些方式獲得不錯的近似，而非直接使用 ground truth。
3. Motion segmentation 的取得在現實情形中也是非常困難的事，因此我們希望能利用一些方式獲得不錯的近似，而非直接使用 ground truth。

Pipeline

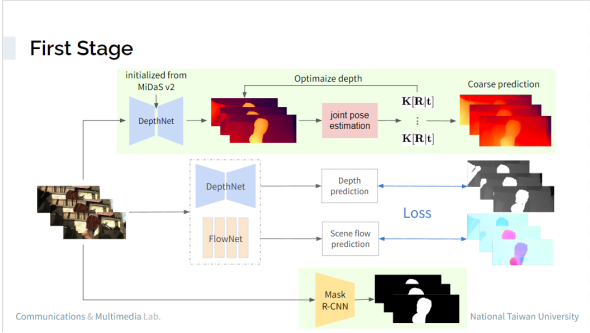
Stage 1

我們 follow Facebook 2021 年的論文 - Robust Consistent Video Depth Estimation。

1. 將 video 中的每個 frame 利用 pretrained 過的 single-frame depth model 得到每個 frame 的 depth estimation。

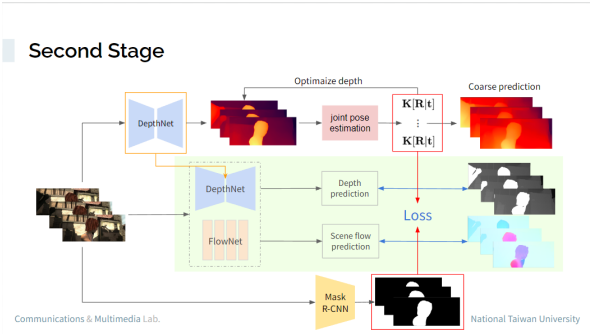
2. 在假設物體都不會移動的情況下，利用與 Consistent Depth of Moving Objects in Video 計算 depth consistency 差不多的方式計算 reprojection consistency (差別是直接計算兩組三維座標的相似性而非 depth 的相似性)
3. 利用 reprojection consistency 學習每個 frame 的內外參
4. 利用內外參修正 depth estimation model
5. 重複的 refine depth estimation 和內外參直到收斂

另外，我們可以利用 Mask R-CNN 預測出每個 frame 的 motion segmentation



Stage 2

在 Stage 1 中我們可以只根據每個 frame 的圖片就得到其對應的內外參以及 motion segmentation。另外，經過 Stage 1，我們也可以得到一個更好的 pretrained single-frame depth model。
有了這些 components 後，便可以根據 baseline model (Consistent Depth of Moving Objects in Video) 的 pipeline 進行 self-supervised training



Experiments

Settings

我們使用 MPI-Sintel dataset (以下簡稱 Sintel)。由於 testing set 需要上傳至網路去取得各項 metrics 的分數，對我們短時間跑多組實驗的期末報告而言，時間上會過於緊縮，故我們從有 ground truth 的資料中，隨機抽取部分資料當作 validation set。而後續的分數也都是在這個 validation set 上跑出來的結果 (validation set 包括 alley_1, ambush_5, bamboo_2, bandage_1, cave_2, market_6, shaman_2, sleeping_1, temple_2 這九個類別) 。原始 Sintel 影片大小為 436x1024，由於我們 gpu 運算資源有限，故我們會將影片縮放至 160x384 來進行後續的實驗。
在做訓練時，我們採用的是 test-time training。在 stage 1 產生 mask 時採用 Mask R-CNN 預訓練好的模型；產生相機內外參時，會訓練 3 個 epoch，stage 1 時間瓶頸主要跟 cpu 處理速度有關。在 stage 2 時，訓練時所使用的 optimizer 參數都跟原本論文一樣，總共會訓練 20 個 epoch；在 stage 2 中，跑完一個影片在單張 RTX 3090 上約需要 15 分鐘。
我們以下的實驗會跟 robust CVD 與 baseline model 這兩篇論文做比較。我們會使用跟上面相同的訓練參數去得到這兩篇方法的結果。而前面有提到，baseline model 需要使用相機的內外參，而我們沒辦法取得當初作者用的 ORB-SLAM2 來得到他們估測出的內外參。故我們在這裡使用 Sintel 提供的真實內外參。

Results on Sintel Dataset

我們遵從 robust CVD 的論文，使用 7 種不同的 metrics 去衡量每個方法的表現，並忽略 80 公尺以上的深度（通常為 dataset 本身的 outlier）。在算這些 metrics 之前，要先將兩部影片的 median depth 做 align。我們參照的兩篇論文分別是採用兩種不同的方式，一種是做 per-frame scaling，將每個 frame 的 median depth 做 align；另一種則是 sequence scaling，影片內的每個 frame 使用相同的縮放值。
以下我們將兩種不同 scaling 的方式所得出的結果都呈現出來：

- Sequence scaling:

Method	RMSE	log RMSE	Abs Rel	Sq Rel	acc-1.25	acc-1.25^2	acc-1.25^3
robust CVD	8.2574	0.9163	0.7616	5.2219	0.3564	0.5831	0.6887
baseline model	9.2782	0.7624	0.7583	9.2931	0.4933	0.6636	0.7738
full pipeline	**7.4069**	**0.5873**	**0.3918**	**3.0630**	**0.5396**	**0.7095**	**0.8160**

- Per-frame scaling:

Method	RMSE	log RMSE	Abs Rel	Sq Rel	acc-1.25	acc-1.25^2	acc-1.25^3
robust CVD	8.0897	0.8998	0.7758	5.8674	0.3763	0.5903	0.6918
baseline model	9.0596	0.7456	0.7682	9.9214	0.5088	0.6664	0.7792
full pipeline	**7.2700**	**0.5770**	**0.3792**	**2.9804**	**0.5536**	**0.7108**	**0.8224**

可以發現我們的方法的表現都相當的好。至於為何 baseline model 會表現得比想像中差這麼多，我們認為是因為估測出來的內外參和真實的有一定程度的落差，而在原始論文提供的 pretrained weight 在估測出來的內外參下去算 loss 可以得到合理的值，但若換成真實的內外參，就會使 loss 變得很大，進而導致模型往奇怪的地方學習。這點我們在檢驗深度預測的部分時，就能發現 baseline model 會在某些點預測極大的值，這也是為何在有平方的 metrics 中，baseline model 的表現都差上許多。
而關於兩種不同的縮放方式，可以發現在 metrics 的分數並不會差到非常多。而兩者之中，sequence scaling 才會跟 temporal consistency 有相關，我們認為這比較能代表一個影片深度預測的表現，故我們後續的 metrics 只放入 sequence scaling 的結果。

Ablation Study

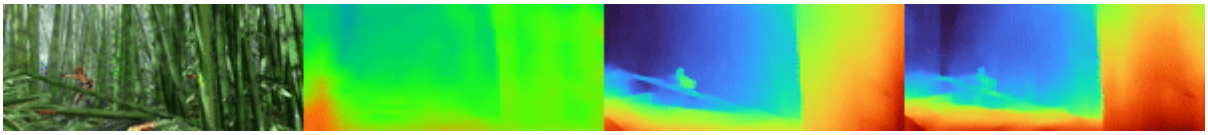
- Sequence Scaling

Method	RMSE	log RMSE	Abs Rel	Sq Rel	acc-1.25	acc-1.25^2	acc-1.25^3
Ours	**7.4069**	**0.5873**	**0.3918**	**3.0630**	0.5396	**0.7095**	**0.8160**
Ours w/ our init weight	7.7332	0.6979	0.4959	3.7593	0.5453	0.7001	0.7699
Ours w/o mask	7.7630	0.6985	0.4948	3.7922	**0.5534**	0.7028	0.7713

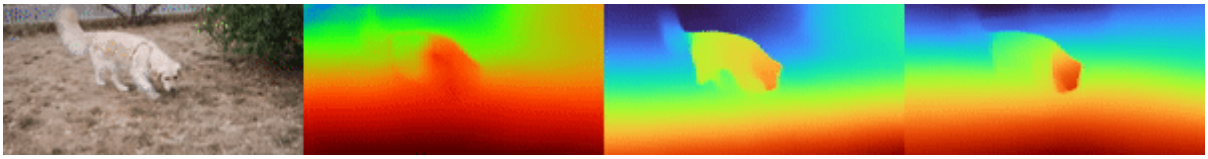
我們可以看到，不同的 initial weight 的確會造成影響，有在 stage 1 中 tune 過且整個 pipeline 都是使用同一個來源還是比中途更換來得好。Mask 的部份，也是影響表現的重要因素之一，能讓模型更加清楚我們所想針對的動態物件是位於哪裡，能更有效地去做調整。

Visualize Results

從左至右分別為原始影片、Robust CVD、Baseline Model、我們 proposed 的方法：



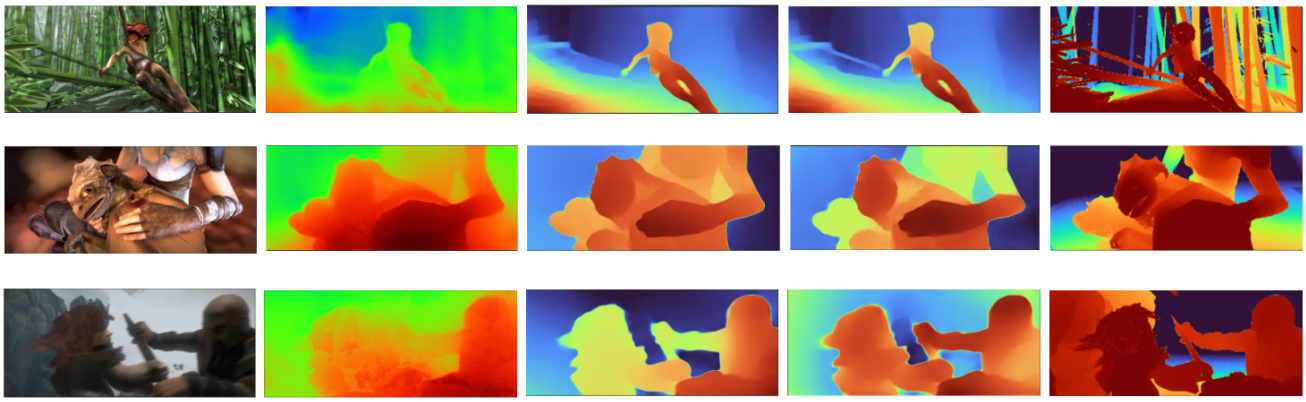
sintel example



real world example

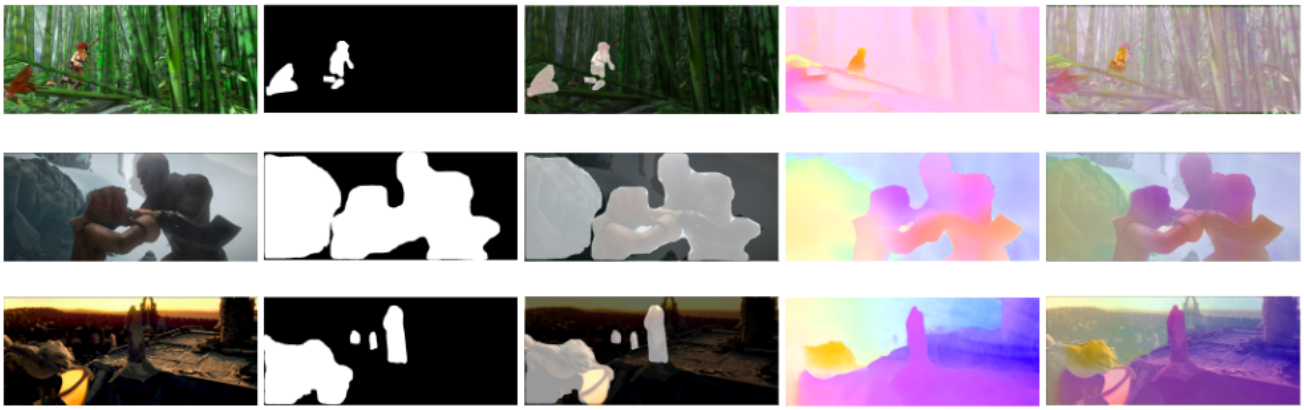
可以發現 baseline 及我們提出的方法無論是在合成的影片(Sintel)或是真實影片，表現上都相當的不錯。而 robust CVD 的結果雖然可以抓出物體的輪廓，但整體表現還是有點差強人意。

Qualitative Results



可以發現，robust CVD 只能大致上可以看出物體的輪廓。而 baseline method 跟我們的方法表現都還算不錯，物體間的輪廓都更加的清晰。我們的方法在某些細節（e.g. 竹子）甚至比 baseline model 表現得更好；在深度部分，也比 baseline method 來得更精確(e.g. 最下面打鬥的部分)。

Mask & Scene Flow



這邊可以看到我們無論在 mask 或 scene flow 都有良好的表現，透過疊圖，我們可以明顯地看到他準確地判斷動態物件是在何處，不會在背景上做過多的預判。

Conclusion

Consistent Depth of Moving Objects in Video 雖然在動態物體的深度預測上已經做得還不錯了，但他假設我們能夠取得每個 frame 的內外參以及 motion segmentation，這些條件不太符合實際的情況。因此，我們決定參照 Robust Consistent Video Depth Estimation 的方式，獲得不錯的內外參估計和 motion segmentation estimation 後，再利用這些結果依照 Consistent Depth of Moving Objects in Video 做 fine-tuning。如此一來，我們在預測影片中每一個 frame 的深度時，除了影像外並不需要任何額外的資訊，就可以做出很好的結果了。
從結果來看，在 quantitative result 中可以看出，我們在各項 metrics 的表現都優於之前的結果；除此之外，從 frame 的深度圖也顯示我們的深度預測結果表現得很好，甚至在一些細節上更勝於 baseline model。

Division of Works

- R10922042 柯宏穎: run code, combine pipeline, write report
- R10922077 蔡秉辰: run code, combine pipeline, write report
- R10922095 黃秉迦: run code, combine pipeline, write report

Reference

- ZhoutongZhang,ForresterCole,RichardTucker,WilliamT Freeman, and Tali Dekel. Consistent depth of moving objects in video. ACM Transactions on Graphics (TOG), 40(4):1– 12, 2021
- Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. ACM Transactions on Graphics (TOG), 39(4):71–1, 2020.
- Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In CVPR, 2021.
- D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, ECCV, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012.