Deep Learning for Computer Vision

NTU, Fall 2024, Homework4

電機所碩一　李彥璋　**R13921093**

Problem: 3D Novel View Synthesis

1. **(15%) Please explain :**
   a. **Try to explain 3D Gaussian Splatting in your own words.**

   3D Gaussian Splatting

   在3D Gaussian Splatting中，Gaussian指的是常見的高斯分布（Gaussian Distribution），又稱常態分布或鐘形分布。當高斯分布延伸到三維空間（XYZ三軸），若以涵蓋率99%為邊界截斷分布，其形狀會呈現橢圓球體的外觀。

   Splatting則是一種光柵化（Rasterization）的渲染技術，簡單來說，它是將三維空間中的物體投影到相機鏡頭上。可以形象地理解為將三維物體從近到遠「砸向」相機，就像雪球砸向平面，從而決定哪些像素被覆蓋。

   因此，3D Gaussian Splatting是一個3D重建架構，能夠從一組2D圖片中，透過光柵化技術學習該場景的3D顯示表示法。在這裡，顯示表示法是指以高斯橢球的形式來表徵三維場景，將其顏色、形狀、透明度等特徵編碼進去，並結合光柵化過程進行渲染，使其能有效地表現出場景的細節與深度資訊。

   b. **Compare 3D Gaussian Splatting with NeRF (pros & cons).**

   NeRF

   優點

   - 擅長處理複雜的幾何形狀和光照效果，能生成高度擬真的視覺效果。
   - 可從未捕捉的視角生成高品質影像，適合多角度場景探索。
   - 僅需儲存神經網路權重，無需顯式存儲幾何結構，因此儲存空間需求相對較小，特別是在處理大型場景時。

   缺點

   - 訓練時間較長（可能需要數天），渲染速度慢（單張影像可能需數秒），需要高性能硬體支持。
   - 不適合即時互動應用，對硬體要求高，且場景編輯困難。

   3D Gaussian Splatting

   優點

   - 訓練速度快（分鐘內），具備即時渲染能力（可達30+FPS）。
   - 特別適合VR/AR等需要快速開發的應用，尤其在大規模場景的即時渲

染中表現優秀。

- 顯式的場景表示便於修改，支持一定程度的場景編輯與動態場景處理。

缺點

- 需要存儲大量的高斯體參數，儲存空間需求比NeRF更高。
- 未觀察區域可能出現品質問題，例如視角跳躍的popping artifacts或高斯體拉伸現象，渲染效果依賴於點雲密度與演算法的優化程度。

c. **Which part of 3D Gaussian Splatting is the most important you think? Why?**

我認為「3D Gaussian Splatting」論文中最重要的部分是使用3D高斯函數來表示場景，從而實現實時的輻射場渲染。透過將場景表示為一系列具有位置、方向和其他屬性的3D高斯分布，這種方法能夠高效地結合神經輻射場的高品質渲染與傳統渲染的速度優勢。

其重要性在於，

1) 效率提升：使用3D高斯分布能夠充分利用GPU的並行計算能力，實現即時渲染，這對需要高幀率的應用（如虛擬現實和遊戲）尤為重要。
2) 品質保持：儘管渲染速度加快，但仍能保持高品質的視覺效果，因為高斯函數能夠平滑地表示光線和顏色的變化。
3) 靈活性：這種方法提供了一個結構化的數據表示，方便進行場景編輯、操作和優化，增加了實際應用的可行性。

因此，使用3D高斯分布來表示和渲染輻射場是該論文的核心創新，也是其最重要的部分，因為它在效率和品質之間取得了優秀的平衡，推動了實時渲染技術的進步。

2. **(15%) Describe the implementation details of your 3D Gaussian Spaltting for the given dataset. You need to explain your ideas completely.**

概述

3D Gaussian Splatting技術的實作包含多個步驟，旨在利用稀疏的點雲來生成場景的高品質表示，並透過自適應優化與差異化渲染達到即時視角合成的效果。首先，基於SfM（Structure-from-Motion）產生的點雲，將場景初始化為一組3D Gaussians，每個高斯由位置、協方差矩陣與不透明度等參數描述，且協方差矩陣以尺度和旋轉矩陣來表示以便於優化。渲染過程則透過將3D Gaussians投影至2D進行，可有效進行α-混合並生成最終渲染結果。在訓練中，會逐步提高球諧函數的次數，來更好適應場景中的光照變化。此外，採用適應性密度控制，對於低密度或不充分表示的區域進行克隆或分割，以精確地表示場景細節。在優化過程中，L1損失和SSIM作為主要的目標，同時也對深度進行正則化處理。差異化渲染器則利用tile為基礎進行渲染，藉由預排序的方式

提升渲染速度。通過這些技術的結合，能在多視角數據集上實現即時、高質量的視角合成。

訓練

python train.py \
--source_path hw4_data/dataset/train \
--model_path output/best_model \
--iterations 60000 \
--densification_interval 900 \
--opacity_reset_interval 100000 \
--densify_from_iter 1000 \
--densify_until_iter 40000

其中，

iterations表Number of total iterations to train for；

densification_interval表How frequently to densify；

opacity_reset_interval表How frequently to reset opacity；

densify_from_iter表Iteration where densification starts；

densify_until_iter表Iteration where densification stops。

渲染

因為一開始hw4_data/dataset/public_test/sparse/0當中，並沒有points3D.ply或points3D.txt，所以要更改scene/dataset_readers.py檔案內檔案讀取的部份程式碼，完成後就可以進行渲染。

執行bash hw4.sh ./hw4_data/dataset/public_test ./evaluate

其中，hw4.sh的內容為python3 render.py --source_path $1 --output_path $2

照片結果會儲存到指定的資料夾，在這裡是./evaluate，最後可以使用助教提供的grade.py進行評分。

3. **(15%) Given novel view camera pose, your 3D gaussians should be able to render novel view images. Please evaluate your generated images and ground truth images with the following three metrics (mentioned in the 3DGS paper). Try to use at least three different hyperparameter settings and discuss/analyze the results.**
    - **Please report the PSNR/SSIM/LPIPS on the public testing set.**
    - **Also report the number of 3D gaussians.**
    - **You also need to explain the meaning of these metrics.**

Setting	PSNR	SSIM	LPIPS (vgg)	Number of 3D gaussians

--iterations 10000	31.6046	0.9568	0.1533	347867
--iterations 30000	32.7667	0.9621	0.1304	380146
--iterations 50000	33.4590	0.9625	0.1299	397956

Discuss/Analyze

1) 訓練時間對模型性能的影響:
    - 隨著訓練迭代次數的增加(10000→30000→50000),所有評估指標都有明顯改善。
    - PSNR從31.6046提升到33.4590,顯示整體重建質量隨訓練時間提升。
    - SSIM從0.9568提升到0.9625,表示結構相似度也有所改善。
    - LPIPS持續下降(0.1533→0.1299),代表生成圖像的感知質量越來越好。
2) 高斯點數量的變化:
    - 高斯點數量從347,867增加到397,756。
    - 顯示模型在訓練過程中持續透過密度化操作增加點數。
    - 點數增加幅度在後期趨緩,從30000到50000迭代時僅增加約17,610個點。
3) 效率與效益分析:
    - 10000到30000迭代的改善最明顯,PSNR提升約1.16。
    - 30000到50000迭代的改善較小,PSNR僅提升約0.69。
    - 顯示訓練在後期的效益遞減,可能達到收斂。
4) 各指標間的關聯:
    - PSNR和SSIM的改善趨勢一致,但SSIM在後期改善較不明顯。
    - LPIPS持續改善,表示即使PSNR和SSIM改善減緩,感知質量仍在提升。
    - 高斯點數量的增加與性能改善呈正相關,但並非線性關係。

Meaning of 3 metrics

PSNR

PSNR(峰值信噪比,Peak Signal-to-Noise Ratio)是衡量圖像或影片品質的重要指標,主要用於評估原始影像與處理後(如壓縮或渲染)影像之間的差異程度。**PSNR**數值越高,代表處理後的影像越接近原始影像,失真程度越低。

計算方式:

1) 首先計算MSE(均方誤差),即兩張影像對應像素差異的平方和平均值。
2) 再依公式:PSNR = 10 * log10((MAX_I)^2 / MSE) 計算,

其中MAX_I為影像可能的最大像素值（如8位元影像為255）。

PSNR以分貝（dB）為單位，一般而言30dB以上即代表影像具有良好品質。這個指標在電腦圖學與影像處理領域被廣泛應用於評估各種演算法的成像效果。

SSIM

SSIM（Structural Similarity Index，結構相似性指數）是一種用於評估兩幅圖像之間相似度的客觀指標。它主要從人類視覺系統角度出發，考慮圖像的亮度（Luminance）、對比度（Contrast）和結構（Structure）三個方面，從而更貼近人眼對圖像品質的感知。

具體而言，SSIM通過以下方式評估兩幅圖像的相似性：

1) 亮度比較：評估兩幅圖像整體亮度的差異。
2) 對比度比較：評估圖像局部對比度的變化。
3) 結構比較：評估圖像結構（如邊緣、紋理）的相似性。

SSIM的取值範圍通常在**0**到**1**之間，值越接近**1**表示兩幅圖像越相似。與傳統的評估方法（如均方誤差MSE或峰值信噪比PSNR）相比，SSIM更能反映人類對圖像品質的主觀感受，因此在圖像處理、壓縮和傳輸等領域被廣泛應用。

LPIPS (vgg)

LPIPS（Learned Perceptual Image Patch Similarity，學習的感知圖像塊相似度）是一種衡量兩幅圖像之間感知相似性的指標。它旨在更準確地反映人類視覺系統對圖像差異的感知，比傳統的評估方法（如MSE、PSNR、SSIM）更貼近人眼對圖像品質的主觀感受。

當提到LPIPS（VGG）時，表示在計算LPIPS時使用了VGG網絡作為特徵提取器。具體而言：

1) 特徵提取：LPIPS首先使用預訓練的深度卷積神經網絡（如VGG）提取圖像的深度特徵。這些特徵能夠捕捉圖像的高層次語義信息和結構特徵。
2) 特徵比較：將兩幅圖像的深度特徵進行比較，計算它們之間的距離。這個距離反映了兩幅圖像在高層次語義和感知上的差異。
3) 感知加權：LPIPS還會對這些特徵差異進行加權，這些權重是通過學習得到的，進一步優化了與人類感知的相關性。

LPIPS的取值範圍通常在**0**到**1**之間，值越小表示兩幅圖像在感知上越相似。由於使用了深度學習模型的特徵，LPIPS能夠更準確地捕捉到人眼對於紋理、顏色和結構等方面的敏感度，因此在圖像質量評估、圖像生成、超分辨率等領域被廣泛應用。

4. **(15%) Instead of initializing from SFM points [dataset/sparse/points3D.ply], please try to train your 3D gaussians with random initializing points.**
    - **Describe how you initialize 3D gaussians.**

- **Compare the performance with that in the previous question.**

How

- 在arguments/__init__.py增加一個參數random_init（初始設定為False），代表是否random initialize。
- 在scene/__init__.py，原本使用self.gaussians.create_from_pcd()（也就是initialize from SFM points）前先判斷是否random_init為True，若是，則改用自行新增的self.gaussians.random_init()，表示random initialize。
- 在scene/gaussian_model.py，
  新增
  def random_init(self, num_pts : int, cam_infos : int, spatial_lr_scale : float),
  這個函式主要是複製原本的
  def create_from_pcd(self, pcd : BasicPointCloud, cam_infos : int, spatial_lr_scale : float)，主要改動僅有傳進的參數num_pts（表初始點數）及以下三行，
  - # 生成隨機點
    fused_point_cloud = torch.rand((num_pts, 3), device="cuda")
  - # 生成隨機顏色
    fused_color = torch.rand((num_pts, 3), device="cuda")
  - # 為每個點計算最近鄰距離
    dist2 = torch.clamp_min(distCUDA2(fused_point_cloud), 0.0000001)

Compare

Setting	PSNR	SSIM	LPIPS (vgg)	Number of 3D gaussians
--iterations 10000 --random_init	19.9501 (差)	0.7151 （差）	0.4476 （差）	756175 （多）
--iterations 30000 --random_init	17.4367 (差)	0.6129 (差)	0.5145 (差)	1615156 （多）
--iterations 50000 --random_init	17.7273 (差)	0.6325 (差)	0.4820 (差)	1166989 （多）

PSNR, SSIM, LPIPS (vgg)皆較initialize from SFM points差；
Number of 3D gaussians則較initialize from SFM points多，
有關iterations的影響沒有明顯趨勢。

Reference

1. 3D建模零門檻！以照片生成場景的3DGS技術解析
   https://www.find.org.tw/index/tech_obser/browse/7eaa3e8ffc7876e74509befd5ed50b8a/
2. Revolution in 3D Imaging: Photogrammetry vs. NeRF vs. Gaussian Splatting
   https://www.toolify.ai/ai-news/revolution-in-3d-imaging-photogrammetry-vs-nerf-vs-gaussian-splatting-79775
3. claude
   https://claude.ai/new
4. chatgpt
   https://chatgpt.com/
5. 3D Gaussian Splatting for Real-Time Radiance Field Rendering
   https://arxiv.org/abs/2308.04079
6. AI 甘安捏 - 3D 重建技術 (三): 3D Gaussian Splatting, 讓 3D 重建從學術走向實際應用的革新技術！用彩色橢圓球球表示場景！
   https://www.youtube.com/watch?v=UxP1ruyFOAQ