

3D Gaussian Splatting for Real-Time Radiance Field Rendering

論文ソース

- [3D Gaussian Splatting for Real-Time Radiance Field Rendering](#)

qiita

- 論文読むよりこの2つ読んだ方がよさそう
 - <https://qiita.com/scomup/items/d5790da25a846e645de1>
 - <https://qiita.com/scomup/items/92716342a3ef0b915e0c>

概要

- 以下の3点
 - 3D Gaussianを導入
 - 3D Gaussianのpropertiesの最適化とdensity controlを組み合わせた
 - 高速で微分可能なGPUレンダリング手法

前提知識

NVS(Novel View Synthesis)

- 複数の視点から撮影された画像を手掛かりに新たな視点の画像を合成する技術のこと

Rasterization

- 3Dシーンの情報を2D平面に投影して画像としてレンダリングするプロセス

SfM

- Structure from Motion
- ある対象を撮影した複数枚の写真から対象の形状を復元する技術の総称

- つまり、複数の2D画像から3Dシーンの構造を推定する
- カメラが異なる位置や角度から撮影した複数の画像を解析し、視点の変化に基づいてシーン内の特徴点の3D座標とカメラの位置・姿勢を同時に推定

Camera Calibration

- カメラ内部および外部のパラメータを求めるプロセス
- カメラの内部パラメータは行列 \mathbf{K} であり以下で定まる

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

- ただし、 f_x, f_y は焦点距離、 c_x, c_y は画像平面の主点(光学中心)の座標
- 外部パラメータは回転行列 \mathbf{R} と平行移動ベクトル \mathbf{t} でカメラの位置を表す
- 外部パラメータと内部パラメータを用いてワールド座標系の3D点 (X, Y, Z) を2D画像平面上の点 (u, v) に投影する式は以下のように書ける

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} (\mathbf{R} | \mathbf{t}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

MVS

- Multi View Stereo

spherical harmonic

- 球面調和関数

クォータニオン

- 任意軸回転を表現する
- 3次元空間内の任意の回転を表現できる
- 3次元空間において方向ベクトル $(\lambda_x, \lambda_y, \lambda_z)$ を回転軸として角度 θ だけ回転するときクォータニオンは4次元ベクトルとして以下のようにあらわされる

$$\mathbf{q} = \left(\lambda_x \sin \frac{\theta}{2}, \lambda_y \sin \frac{\theta}{2}, \lambda_z \sin \frac{\theta}{2}, \cos \frac{\theta}{2} \right)$$

- 逆に、クォータニオン $\mathbf{q} = (q_x, q_y, q_z, q_w)$ が得られたときに回転行列は以下で定まる

$$R = \begin{pmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_z q_w) & 2(q_x q_z + q_y q_w) \\ 2(q_x q_y + q_z q_w) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_x q_w) \\ 2(q_x q_z - q_y q_w) & 2(q_y q_z + q_x q_w) & 1 - 2(q_x^2 + q_y^2) \end{pmatrix}$$

3DGS

Gaussianの導入

- ワールド空間で定義された完全な3D共分散行列 Σ で定義
- 平均 μ を中心とする
- よって以下のように書ける

$$G(\mathbf{x}) = \exp\left(-\frac{1}{2}\mathbf{x}^\top \Sigma^{-1} \mathbf{x}\right)$$

- このガウシアンはblending processで α と掛け算される
- この α が不透明度
- 3Dガウシアンを2Dに投影することを考える
- 視点変換 W が与えられたときにカメラ座標系での共分散行列 Σ' は

$$\Sigma' = JW\Sigma W^\top J^\top$$

- ここで J は射影変換のアフィン近似のヤコビアン
- この Σ' の第3行と第3列を無視して得られる 2×2 行列がカメラ座標系での共分散行列

Initialization

- 点群からGaussianを構築
- 共分散行列は以下、未理解
 - We estimate the initial covariance matrix as an isotropic Gaussian with axes equal to the mean of the distance to the closest three points.

Gaussianの最適化の準備

- 共分散行列 Σ を直接最適化するとGaussianではなくなる可能性がある
 - 共分散行列 Σ は半正定値行列(positive semidefinite matrix)という制約をもつ
 - ただ最適化するだけだと当然この制約を満たすとは限らない
- そこで以下のように分解する

$$\Sigma' = RSS^\top R^\top$$

- S はスケーリング行列、 R は回転行列
- S をスケーリングのための3Dベクトル \mathbf{s} で表現
- R を4Dベクトルクォータニオン \mathbf{q} で表現
- この \mathbf{s}, \mathbf{q} を最適化する

Optimization with adaptive density control

- ガウシアンのパラメータの最適化とガウシアンの密度のコントロールが交互に行われる
- 最適化するガウシアンのパラメータは以下
 - \mathbf{p} : 位置
 - α : 不透明度
 - Σ : 共分散行列
 - 実際はスケール \mathbf{s} とクォータニオン \mathbf{q}
 - \mathbf{c} : 色(SH係数が色を表す)
 - 実際は球面調和パラメータ \mathbf{h}
- よってガウシアンのパラメータは5つのベクトル(世界座標におけるガウシアンの分布点、クォータニオン(回転)、スケール、球面調和パラメータ、不透明度)
 - これら5つのベクトルがガウシアンの個数分だけ存在する
- 不透明度 α にはシグモイド関数を活性化関数として適用して $[0, 1]$ にする
- 共分散行列には指数関数を活性化関数として適用
- 損失関数は最終的に得られた画像に対してL1ロスとD-SSIMで比較

$$\mathcal{L} = (1 - \lambda) \mathcal{L}_1 + \lambda \mathcal{L}_{\text{D-SSIM}}$$

- Adaptive density Controlについては以下
 - 100回の反復ごとに透明なガウシアンを削除する
 - 不透明度 α が閾値 ϵ_α より小さいもの
 - 視点空間における位置勾配の平均大きさ(average magnitude of view-space position gradients)が閾値 τ_{pos} を上回るものを探す
 - このaverage magnitude of view-space position gradientsは $\frac{\partial \mathcal{L}}{\partial \mathbf{u}}$ のこと
 - $\mathbf{u} = \mathbf{K}(\mathbf{R}|\mathbf{t}) \mathbf{p}$ (ワールド座標系からカメラ座標系への変換)
 - under-reconstructed regionsにあるような小さいガウシアンは自身と同じサイズのガウシアンを複製する(Clone)
 - over-reconstructed regionsにあるような大きいガウシアンは自身を2分割してスケールを小さくする(Split)

Fast Differentiable Rasterizer for Gaussians

- section 6

- 未理解

全体の流れ

- これまでの内容を簡潔にまとめる
- 学習するパラメータは N 個のガウシアンに対してそれぞれ以下の5つ
 - \mathbf{p}_{wi} : 3D Gaussian分布の中心点(世界座標)
 - \mathbf{q}_i : 3D Gaussianの回転行列をクォータニオンで表現したもの
 - \mathbf{s}_i : 3D Gaussianのスケール
 - \mathbf{h}_i : 3D Gaussianの球面調和パラメータ(色情報)
 - α_i : 3D Gaussianの不透明度
- 以下が設定値(値を変えると投影される画像が変わるが3D世界には関係なく学習対象ではない)
 - \mathbf{R}_{cw} : カメラの回転
 - \mathbf{t}_{cw} : カメラの併進
 - \mathbf{K} : カメラの内部パラメータ
- 全体の流れは以下
 - 3D Gaussianの中心点を2D画面に投影するときの座標値計算
 - 3D共分散の計算
 - 2D共分散の計算
 - 色を計算
 - 2D画像生成
- 1についてはカメラの内部パラメータと回転と併進を用いて \mathbf{p}_w から \mathbf{u} を得る
- 2については上述の通りスケールベクトルとクォータニオンから共分散行列を計算
- 3については以下のようにして3D共分散行列 Σ から Σ' を得る

$$\Sigma' = \mathbf{J} \mathbf{R}_{cw} \Sigma \mathbf{R}_{cw}^\top \mathbf{J}^\top$$

- \mathbf{J} はヤコビ行列で以下

$$\mathbf{J} = \begin{pmatrix} \frac{f_x}{z} & 0 & -\frac{f_x x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{f_y y}{z^2} \end{pmatrix}$$

- 4については以下

$$c(r, h) = \sum_{l=0}^{l_{max}} \sum_{m=-l}^l h_{lm} Y_l^m(r)$$

- r はカメラと3D Gaussianの間の単位方向ベクトルで、 Y_l^m は l 次元 m 番の球面基底

- 5についてはカメラの方向から見た深度(Depth)に基づいて3D Gaussianをソートした後に以下を計算することで j 番目のピクセルのRGB値 γ_j を得る

$$\gamma_j = \sum_{i=1}^N \alpha'_{ij} (\alpha_i, \Sigma_i, u_i, x_j) c_i \tau_{ij}$$

- α'_{ij} は i 番目の3D Gaussianがピクセル j における不透明度を表していて以下で定まる

$$\alpha'_{ij} (\alpha_i, \Sigma_i, u_i, x_j) = \exp \left(-\frac{1}{2} (u_i - x_j) \Sigma_i'^{-1} (u_i - x_j)^\top \right) \alpha_i$$

- τ_{ij} は i 番目の3D Gaussianが前方の3D Gaussianを透過してピクセル j に到達した際の減衰を示す係数である
 - これを計算するためにdepth sortすること

$$\tau_{ij} = \prod_{k=1}^{i-1} (1 - \alpha'_{kj}) \tau_{1j} = 1$$

英単語メモ

- Radiance Field methods : NeRF(Neural Radiance Field)のこと
- interleave : 交互配置する
- anisotropic : 異方性の
- opacity : 不透明度
- spherical : 球状の
- harmonic : 調和した
- flickering : ちらつき
- drawback : 欠点
- regime : 政権、体制