1. Physically Embodied Gaussian Splatting: A Realtime Correctable World Model for Robotics

1. Links
    1. https://arxiv.org/pdf/2406.10788
    2. https://embodied-gaussians.github.io
2. Authors: Jad Abou-Chakra (lebanon surname),Krishan Rana, Feras Dayoub (middle-east, lebanon surname), Niko Sunderhauf
3. Affliations: Queensland University of Technology (QUT), University of Adelaide
4. TL;DR: An approach combines gaussian splatting with PBD, which enables a world model with both physical and visual representaion.

TODO

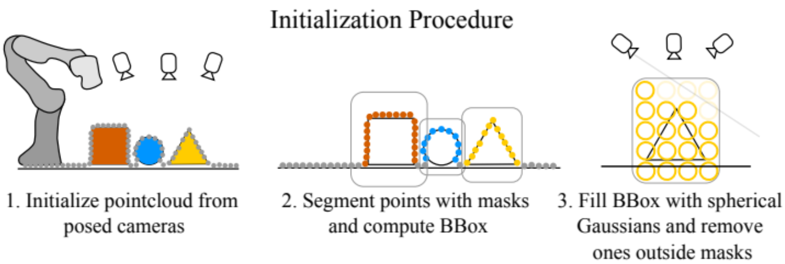
1. read PBD
2. try warp https://github.com/NVIDIA/warp

Contributions

1. A mechanism that combines gaussian seeds with particles from PBD, captures both visual and physical information about the scene
2. A real-time method to correct the particle states using visual feedback

Key concepts

1. Position-Based Dynamics (PBD)
    1. Resources:
        1. original paper: https://matthias-research.github.io/pages/publications/posBasedDyn.pdf

        2. tutorial from original author: https://matthias-research.github.io/pages/publications/PBDTutorial2017-CourseNotes.pdf
        3. useful blog: https://carmencincotti.com/2022-07-11/position-based-dynamics/
    2. Algorithm
        1.
```
for all vertices i
    initialize xi = xi_0, vi = vi_0, wi = 1/mi
while simulating
    for all particles i (pre-solve)
        vi ← vi + Δt*wifext(xi)
        pi ← xi + Δt*vi
    for solverIterationCount
        for all constraints C (solve)
            solve(C, Δt) (solve for xi)
    for all particles i (post-solve)
        xi ← xi + Δxi
        vi ← (xi - pi)/Δt
```
            1. Initialize each particle with position, rotation, mass, linear and angular velocities
            2. given applied force, solve uncorrected position for each particle
            3. solve contraints (nonlinear, inequality/equality)
            4. update positions
        2. These can be parallelized
2. Gaussian splatting
    1. 

    2. The authors transformed point clouds to gaussians with a similiar approach to GaussianGrasper(RAL'24).

Details

Initialization

Gaussians

1. use tools to get object instance label and 3D-BBOX
2. filled with Gaussians(r = 4-7mm, $\Sigma = I$)
3. solving positions, colors and opacities by PBD
4. training with photometric and segmentation reconstruction loss

$$L_{\text{rgb}} = \sum_{\boldsymbol{u}} |C_{\text{rgb}}(\mathbf{u}) - C_{\text{gt}}(\mathbf{u})| \quad \text{and} \quad L_{\text{seg}} = \sum_{\boldsymbol{u}} |S(\mathbf{u}) - S_{\text{gt}}(\mathbf{u})| \tag{8}$$

5. prune gaussians with $I_{\alpha_i \leq 0.3}$

Particles

1. init. with given position
2. align them with shape constraint

$$\boldsymbol{A}_S = \sum_{i \in S} \frac{1}{5} m_i \boldsymbol{R}_i + \boldsymbol{p}_i \bar{\boldsymbol{p}}_i^T - M \boldsymbol{c}_S \bar{\boldsymbol{c}}_S^T, \quad \boldsymbol{c}_S = \frac{\sum_{i \in S} m_i \boldsymbol{p}_i}{M}, \ \bar{\boldsymbol{c}}_S = \frac{\sum_{i \in S} m_i \bar{\boldsymbol{p}}_i}{M}, \ M = \sum_{i \in S} m_i \tag{4}$$

Refinement

1. optimize gaussians, add more gaussians
2. remove gaussians far from particles

Online Prediction

1. use FK to get positions of particles for the robot
2. use PBD to get predicted positions of particles

Online Correction

1. update gaussians those are only attached on objects by photometric reconstruction loss from given camera viewpoints.
2. reset the gaussian to original position
3. compute the imposed force of each particle

$$\boldsymbol{f}_i = K_p \sum_j o_j (\boldsymbol{g}_j - \boldsymbol{g}_j^0),$$