Occupancy Networks:
Learning 3D Reconstruction in Function Space

Andreas Geiger

Autonomous Vision Group
University of Tübingen / MPI for Intelligent Systems
Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision

[Niemeyer, Mescheder, Oechsle & Geiger, In Review]
Architecture

Forward Pass

(Rendering)
Forward Pass:

- For all pixels $u$
- Find surface point $\hat{p}$ along ray $w$
  via ray marching and root finding
- Evaluate texture field $t_\theta(\hat{p})$ at $\hat{p}$
- Insert color $t_\theta(\hat{p})$ at pixel $u$
Backward Pass
(Differentiation)
Differentiable Volumetric Rendering

Backward Pass:

- Image Observation $\mathbf{I}$
- Loss $\mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}) = \sum_u \|\hat{\mathbf{I}}_u - \mathbf{I}_u\|
- Gradient of loss function:
  \[
  \frac{\partial \mathcal{L}}{\partial \theta} = \sum_u \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{I}}_u} \cdot \frac{\partial \hat{\mathbf{I}}_u}{\partial \theta}
  \]
  \[
  \frac{\partial \hat{\mathbf{I}}_u}{\partial \theta} = \frac{\partial \mathbf{t}_\theta(\hat{\mathbf{p}})}{\partial \theta} + \frac{\partial \mathbf{t}_\theta(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} \cdot \frac{\partial \hat{\mathbf{p}}}{\partial \theta}
  \]
- Differentiation of $f_\theta(\hat{\mathbf{p}}) = \tau$ yields:
  \[
  \frac{\partial \hat{\mathbf{p}}}{\partial \theta} = -\mathbf{w} \left( \frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} \cdot \mathbf{w} \right)^{-1} \frac{\partial f_\theta(\hat{\mathbf{p}})}{\partial \theta}
  \]

⇒ Analytic solution and no need for storing intermediate results

Results

Thank you!

http://autonomousvision.github.io