UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning

Bolaños T.

Kilkkilä M.

Kurki L.

Rehn A.

Zaka A.

Zhao Z.

October 14, 2020

Outline

- Introduction
- System Overview
- Objective Losses
- Experimental Evaluation
- Conclusion
- 6 Contributors

Introduction UnDeepVO

A monocular visual odometry system

Introduction

Visual odometry

- Goal
 - Robot localization using only visual information

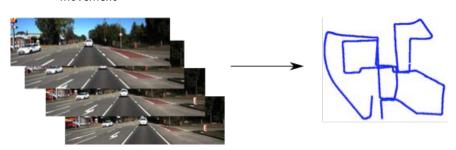




Introduction

Visual odometry

- Goal
 - Use consecutive monocular images to construct a path of robot movement



Introduction UnDeepVO

- Based on deep learning
- Unsupervised
 - No need for labeled training data
- Pose estimation
- Depth estimation

System Overview

Architecture

• Maybe that figure on the paper ...

System Overview

Training Scheme

• ..

Objective Losses

Spatial Losses

The spatial losses are based on the fact that, given the structure of stereo cameras, for a pixel $p_l(u_l, v_l)$ on the left image and $p_r(u_r, v_r)$ on the left image:

$$u_l = u_r$$
 and $v_l = v_r + D_p$

Photometric Consistency Loss (Image reconstruction)

$$L_{pho} = \lambda_s L^{SSIM}(I, I') + (1 - \lambda_s) L^{I_1}(I, I')$$

Disparity Consistency Loss (Depth)

$$L_{dis} = L^{l_1}(D_{dis}, D'_{dis})$$

Pose Consistency Loss (Camera orientation)

$$L_{pos} = \lambda_p L^{l_1}(t_l, t_r) + \lambda_o L^{l_1}(R_l, R_r)$$

Objective Losses

Temporal Losses

This is based on the reconstruction of pixels on time k and (k+1) as

$$p_{k+1} = KT_{k,k+1}D_{dep}K^{-1}p_k$$

Photometric Consistency Loss (Image reconstruction)

$$L_{pho} = \lambda_s L^{SSIM}(I, I') + (1 - \lambda_s) L^{I_1}(I, I')$$

• 3D Geometric Registration Loss (Adding depth with P(x, y, z))

$$L_{geo} = L^{l_1}(P, P')$$

Evaluation

Trajectory

...

Evaluation

Depth

• ..

Conclusions

• UnDeepVo ...

Contributors

- Bolaños Tlahui
 - Objective Losses
- Kilkkilä Miikka
 - ...
- Kurki Lauri
 - ...
- Rehn Aki
 - Slide organization, introduction, conclusions
- Zaka Ayesha
 - UnDeep VO Key Contributions
- Zhao Zhao
 - System Overview