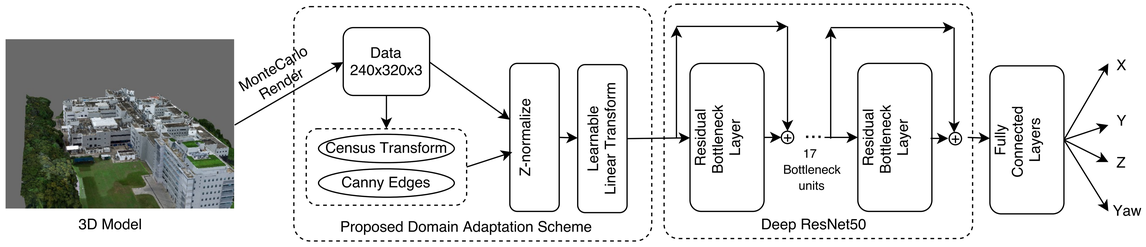
Deep MapNets

A Residual Network for 3D Environment Representation

***Manohar Kuse, Sunil Prasad Jaiswal & *Shaojie Shen**
*HKUST Robotics Institute
Dept. of Electronics and Computer Engineering
Hong Kong University of Science and Technology
Hong Kong
mpkuse@connect.ust.hk



Objective

- Ability to fly small drones in a large known environments.
- Drone has only a monocular camera for sensing.
- In this work, **Localization in Model's Co-ordinate system.**

Introduction

- For navigation and planning, robot needs a) 3D point cloud b) Current position.
- LiDARs are accurate but heavy and expensive.
- RGBD cannot be used outdoors, stereo needs high computational power.
- However, offline creation of 3D models from videos a mature technology.
- If we can localize in co-ordinates of 3D model we effectively know the structure of environment for path planning.

Related Works

- PoseNet [3]. Camera relocalization using 6-DOF pose regression on SfM data.
- Qiu *et al.* [4]. Edge Alignment for model to image registration.

Contributions

- Residual learning for 3D environment representation.
- Use of 3D model to provide training data.
- Data augmentation with census transform for faster learning.

Proposed Technique

View Rendering

- 3D polygonal mesh created using external tools.
- Sample a 4-DOF pose (X,Y,Z,yaw) of a down looking camera.
- Render images using a game-engine to feed into deep neural nets.

Network Architecture

- $f_{\boldsymbol{\theta}} \; : \; \Re^{c \times n \times m} \; \mapsto \; \Re^{4}$
- Input: Image and Augmented Data.
- Learnable Parameters : θ.
- f is modelled as a Residual net[2], followed by fully connected layers
- loss $= \sum_{i=0}^{b} \left\| \mathbf{W}(\mathbf{f}_{\theta}(\mathbf{x_i}, \tilde{\mathbf{x}}_\mathbf{i}) - \hat{\boldsymbol{\kappa}}) \right\|_2 + \lambda \left\| \boldsymbol{\theta} \right\|_2$. $\mathbf{W}$ is diagonal matrix.
- Loss function minimized with Stocastic Gradient Descent (SGD)

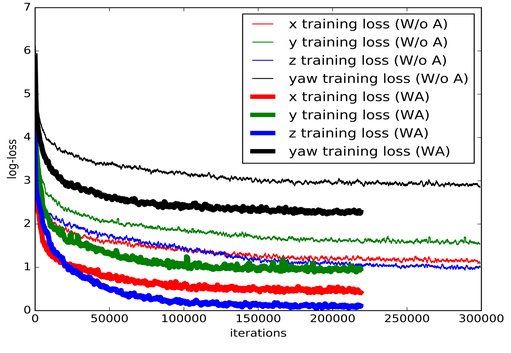
Feature Augmentation

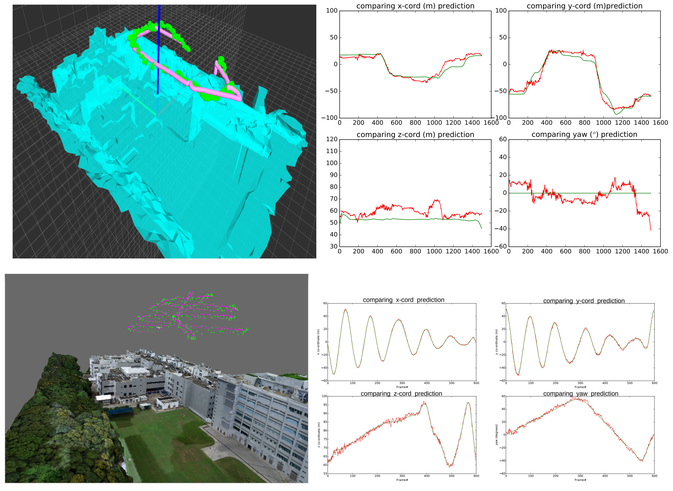
- Cross domain generalization is a major challenge.
- Use case of frustratingly easy domain adaptation [1].
- Augment input space with textural and structural information.
- Census transform for texture information.
- Canny edges for structural information.

Implementation Details

- Created 3D model with *Altizure* system, using manually piloted drone video.
- *Panda3D* game engine for rendering views from 3D model.
- *caffe* for defining and training neural nets.
- *Nvidia TitanX (Pascal)* with 12GB memory for training.
- *DJI-Onboard SDK* on *DJI Matrice100* for logging camera and GPS data.
- 17 Bottleneck ResNet units. 51 layers in all.
- Residual nets produces 2048-D vector which is fed into 2-layer fully connected network.
- $\lambda = 0.0006$, iterations $200k$, batch size $= 16$ etc. Training GPU memory required $11GB$. Test time $400MB$.
- Adding random noise, intensity changes and perturbation in pitch and roll to rendered images.

Experiments





References

[1] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, 2015. cite arxiv:1512.03385Comment: Tech report.

[3] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[4] Kejie Qiu, Tianbo Liu, and Shaojie Shen. Model-Based Global Localization for Aerial Robots Using Edge Alignment. *IEEE Robotics and Automation Letters*, 2(3):1256–1263, 2017.