Final Project- Machine Learning for Robotics (RBE 577)

Monocular Depth Estimation from Drone Camera Images

Overview:
The goal of this project is to estimate depth from images captured by a monocular camera onboard a drone. Monocular depth estimation is a challenging problem in computer vision, which this project will address using deep learning methods.

You will reproduce the results from the paper:
M. Folder et al., "Parallax Inference for Robust Temporal Monocular Depth Estimation in Unstructured Environments", 2022.

This technique is unique because it fuses sequential camera images with corresponding camera translation and rotation measurements (from GPS/IMU) to produce metric depth estimates. The authors have open-sourced their code, which is implemented in TensorFlow.

The project has two main phases:

1. **Synthetic Data Training:** You will first follow the instructions in the official GitHub repository to train the network on the **MidAir** dataset, a synthetic dataset. The objective is to replicate the performance reported in the paper.
2. **Real-World Data Fine-Tuning:** Next, you will use the pre-trained model and fine-tune it on the **UseGeo** dataset, which consists of real-world images. The final objective is to predict depth for images in the UseGeo dataset and compare your predictions to the ground-truth depth maps.
   A key learning objective is to investigate whether pre-training on synthetic data improves performance on real images or if the domain shift is too significant.

Team Collaboration:

- Students may work in teams of two.

Note on Grading:
Grades will be based on the quality of the submission, with an emphasis on effort, methodology, and problem-solving, not just on achieving perfect results. Submissions will be evaluated based on the work produced. We will fairly adjust our expectations for individual versus team submissions.

Final Deliverables

1. Final Presentation:

- A 10-minute presentation describing the algorithm, highlighting implementation challenges, and presenting your results.

2. Submission Package:

- Presentation slides.
- A video recording of your presentation.
- Well-documented and organized code in a shared repository (e.g., GitHub).

3. Phase 1: MidAir Dataset Results
a. Provide a plot of the training and validation loss as a function of epochs, similar to Figure A1 in the paper (page 19).
b. Provide 10 visual examples comparing the predicted depth maps to the ground-truth depth maps.

4. Phase 2: UseGeo Dataset Results
a. Write a training script and a custom DataLoader for the UseGeo dataset.
b. Provide a plot of the training and validation loss as a function of epochs.
c. Provide 10 visual examples comparing the predicted depth maps to the ground-truth depth maps.

Note:

The model you pre-train on the synthetic MidAir dataset may or may not generalize well to the real-world UseGeo dataset. A key part of your analysis should be to evaluate and discuss the effect of this sim-to-real transfer. Does pre-training on synthetic data help, hurt, or have no significant effect on your results on real images? You are expected to address this question in your presentation and report.

Bonus Objectives (Optional)

1. **Framework Conversion:** Convert the original TensorFlow repository to PyTorch.
2. **Architectural Innovation:** Modify the model's architecture (e.g., by incorporating transformers) and demonstrate improved performance on the task.
