Visual Odometry via Contrastive Learning

Zeel S Bhatt

September 30, 2023

1 Introduction

Visual Odometry is an essential component in a Visual SLAM system. The aim of Visual SLAM (Simultaneous Localization and Mapping) is to create a 3D map of the world while simultaneously estimating the camera's position in the created map. Visual odometry is a building block to achieve this. Through visual odometry, we attempt to estimate the camera's motion by analyzing the changes in images due to camera movement. This element is pivotal in various fields, including SLAM, 3D reconstruction, augmented reality, and more. A classic pipeline for visual odometry consists of camera calibration, feature detection, feature matching, triangulation, and local optimization (Bundle Adjustment) Figure 1. This geometry based method has been broadly implemented in various SLAM algorithms.

On the other hand, deep learning-based methods have dominated many computer vision tasks, learning-based visual odometry is not on par with strong geometric methods. The community argues that this is due to insufficient diversity of data and scale ambiguity in triangulation[1]. These issues can explicitly be dealt with using Contrastive Learning. Contrastive learning utilise data augmentation to improve learning performance. In this way, contrastive learning can help derive benefits from sparsely diverse data available for visual odometry tasks through valid augmentations. Contrastive loss pulls feature vectors of negative pair far apart while keeping positive pairs of vector pushed together on latent space. This control of stretching over latent space can be very useful for visual odometry problem.

I am proposing a contrastive learning based method which can improve the capabilities of visual odometry.

- Standard contrastive learning tasks typically involve classification[2][3]. However, these loss functions are not suitable for our case because the output label of visual odometry is position and rotation of the camera in $\mathbb{R}^3$ space, making it a regression problem. Therefore, implementing a standard contrastive loss is not possible. Instead, I need to employ a custom regression loss function tailored for the visual odometry task. This custom loss function will operate over latent space.

- Having an appropriate loss function is not enough, as contrastive learning relies on data augmentations. Regular image augmentations, such as random rotation and flipping, are not suitable for our situation because any rotation in the images influences the output. Color jittering is not possible because visual odometry algorithms rely solely

on grayscale images. We require reasonable augmentations to effectively employ the contrastive loss function.

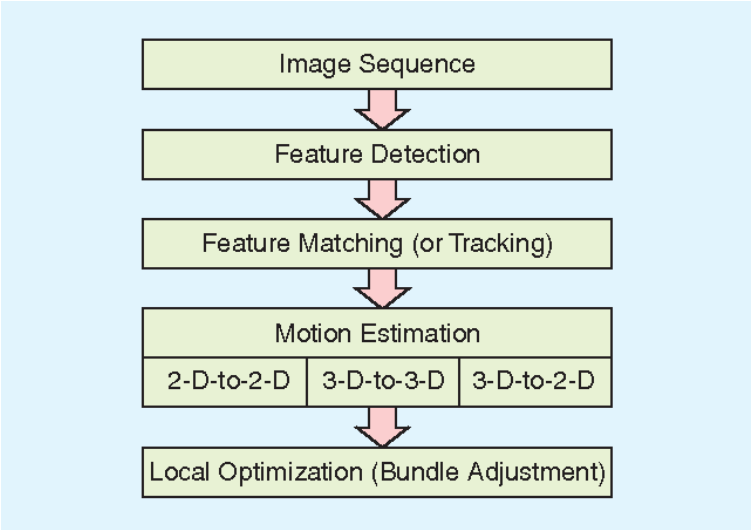

Figure 1: Visual odometry pipline, diagram credit [4]

2 Approach

Monocular visual odometry takes two consecutive images $\{I_t, I_{t+1}\}$, and estimates the relative camera motion $\delta_t^{t+1} = (T, R)$, where T is translation and R is rotation. As shown in figure 3 We initially define most basic CNN network as an encoder, we pass input pair to encoder model and the output of the encode is considered feature space. We call output features given by the encoder $f = E(.)$ we then calculate contrastive loss L_c at feature space. The features are fed to multilayer perceptron and then calculate the loss on the label space. We will be using (MSE) mean square error loss L_p. Both Losses are added to together $L_c + \lambda L_p$ and then we perform back propagation. Here λ is a hyperparameter.

It is obvious that positive pair indicates the data points belong to the same cluster and negative pairs means different cluster. However there is no notion of positive-negative pairing in regression task, since there are no classes. Generally, contrastive loss functions find cosine similarity between pair of inputs. The higher the cosine similarity between negative pairs, the higher the loss would be.

3 Dataset

For this project we will be using KITTI dataset [5]. It is a widely used computer vision dataset for training and evaluating algorithms related to autonomous driving and scene understanding. It was created by researchers from the Karlsruhe Institute of Technology and the Toyota Technological Institute in 2012 and has since become a benchmark dataset

in the field of computer vision and autonomous driving. For training the neural network I will be using PyTorch library in Python.

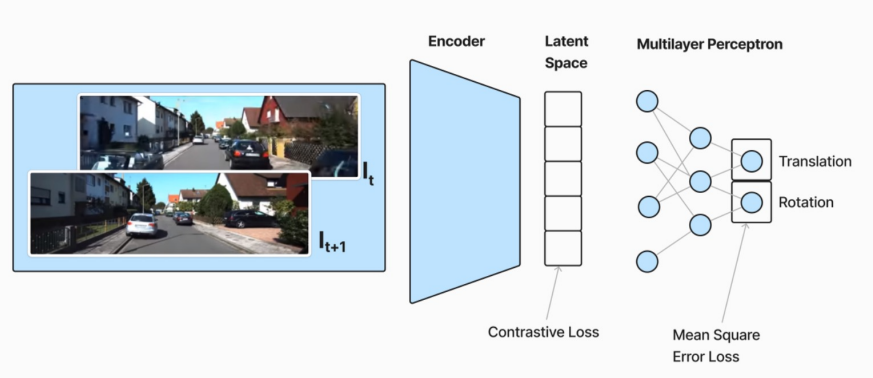

Figure 2: Network architecture for visual odometry. Camera image credit:KITTI dataset [5]

4 Experiments

There are various experiments we can perform here.

- We can use multiple augmentations, such as image cropping, and find optical flow from the consecutive image frames. Optical flow is particularly helpful in understanding how specific objects in the image are moving relative to each other.
- We can use results of road segmentation and feed it as a augmented dataset.
- Currently, I use ResNet50 as an image encoder to calculate the features. There are many DNN architectures available, such as VGGNet [6] or GoogLeNet [7]. I will use these various DNN architectures for my experiments.

5 Deliverables

- Midterm (Last week of October): Will be performing literature review and I will be having the custom loss function implemented our case. I will also prepare valid data augmentation.
- End of the semester: Runnig various experiments to compare between them and improve the performance.

This research is a part of my MS thesis. I have taken permission from the faculty advisor Dr Yezhou Yang to include this project as course project.

References

[1] W. Wang, Y. Hu, S. Scherer, Tartanvo: A generalizable learning-based vo (2020).

[2] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International conference on machine learning, PMLR, pp. 1597–1607.

[3] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, Advances in neural information processing systems 33 (2020) 18661–18673.

[4] D. Scaramuzza, F. Fraundorfer, Visual odometry [tutorial], IEEE robotics & automation magazine 18 (2011) 80–92.

[5] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: Conference on Computer Vision and Pattern Recognition (CVPR).

[6] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.