Can Dense Object Representations help Task Modeling?

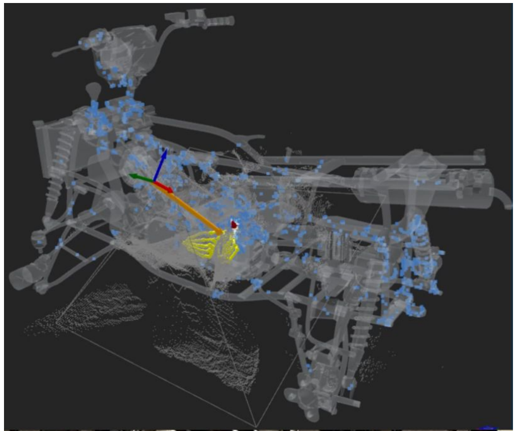
Goal: Estimate dense object representations for multi-modal task understanding in procedural videos

Description:

We aim to explore the efficacy of dense object representations for multi-modal task understanding in procedural videos. Current methods for action recognition, object state change identification, and mistake detection primarily rely on inferences drawn from video sequences. However, these approaches often overlook the significance of object pose and actionable parts, which are critical for accurately estimating actions, object states, and human errors.

The project will address two primary problems. Firstly, we will investigate the performance of existing methods like OnePose++, which have demonstrated high accuracy in tracking static and dynamic objects, under conditions of occlusion that occur naturally in hand-object interactions. Secondly, the project will focus on encoding information such as object poses, part labels, and hand tracks into a cohesive framework multiple downstream tasks.

We will use an existing multi-modal dataset of procedural tasks to implement the algorithms and evaluate the improvements over state of the art methods.



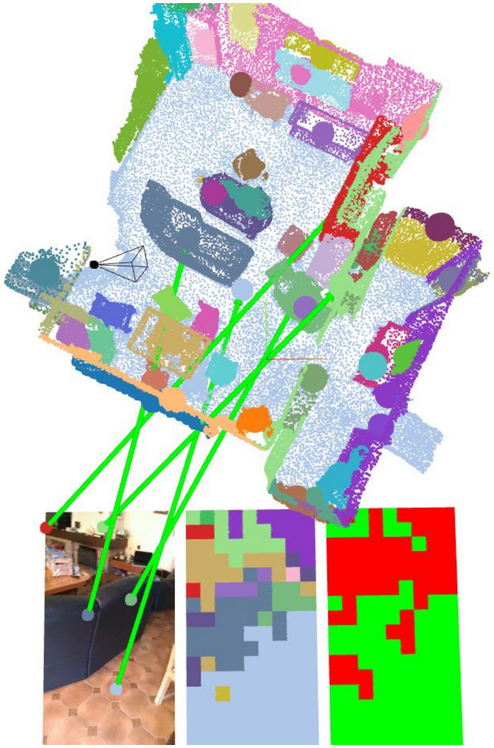
Requirements / Tools:

Knowledge of 3D object pose estimation
Knowledge of Transformer and other auto-regressive algorithms

Supervisor:

Ishani Chakraborty (ischakra@microsoft.com)

Coarse Visual Localization of Sequences in 3D Scene Graphs



Goal: Localize an image sequence in a 3D scene graph in a course manner

Description:

The project is about finding potential locations (e.g., room or building) of an input image sequence given a prebuilt map of the environment represented by 3D scene graph. The scene graph is a graph where the nodes represent object instances (e.g., chair) or large semantic classes (e.g., wall) and the edges relationships (e.g., "nearby"). The benefit of this representation is that it is light-weight and it enables simultaneously exploiting multiple modalities (e.g., point cloud, image). Potential main steps of the project:

- Downloading and setting up an existing dataset with prebuilt scene graphs and various data modalities.
- Designing algorithms that matches objects appearing in the images to objects in the graph.
- Designing methods for selecting the room/building where the sequence was recorded or notifying the user of there is no such location in the map.

Requirements / Tools:

Python

Supervisor:

Olga Vysotska <ovysotska@ethz.ch>
Daniel Barath <dbarath@ethz.ch>

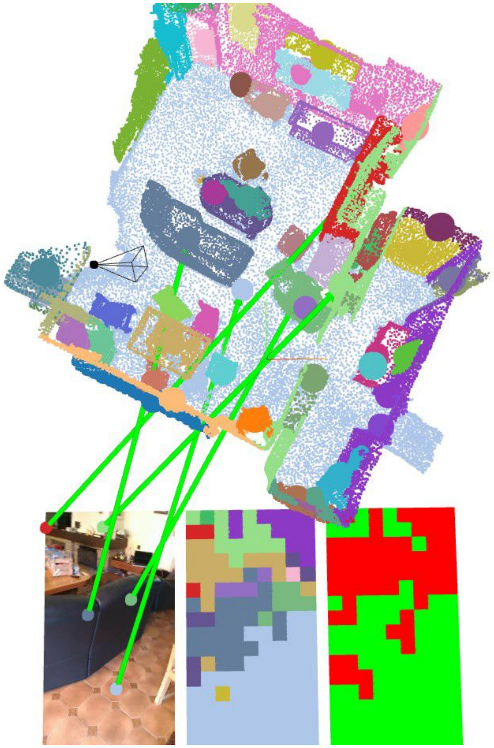
Accurate Visual Localization of Sequences in 3D Scene Graphs

Goal: Localize an image sequence in a 3D scene graph

Description:

The project is about finding the exact 6D pose (rotation and translation) of an input map of the environment represented by 3D scene graph. The scene graph is a g object instances (e.g., chair) or large semantic classes (e.g., wall) and the edges r benefit of this representation is that it is light-weight and it enables simultaneou (e.g., point cloud, image). Potential main steps of the project:

- Downloading and setting up an existing dataset with prebuilt scene graphs ar
- Designing algorithms that matches objects appearing in the images to objects
- Designing methods for estimating the sequence's pose given the object-to-ob



Requirements / Tools:

Python

Supervisor:

Daniel Barath <dbarath@ethz.ch>
Olga Vysotska <ovysotska@ethz.ch>

Human-Scene Interactions by observing Hand Poses from Meta Quest 3

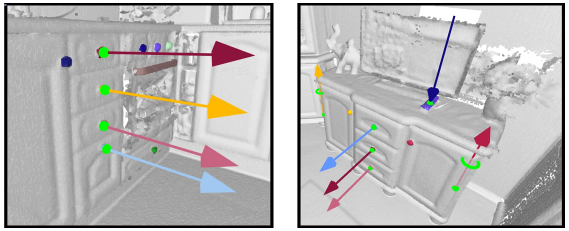
Goal: Representing human-scene interactions in a 3D scene, by recording human-scene interactions using a Meta Quest 3.

Description:

Input: Ego-centric video and hand tracking obtained from the Meta Quest 3.

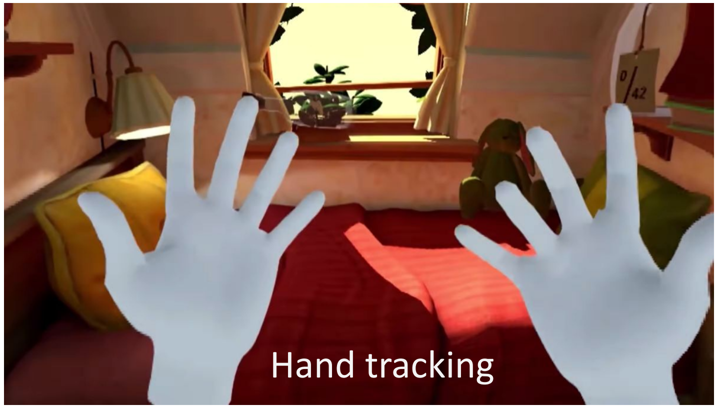
Output: Contact areas within the pre-scanned 3D room. This means ***where*** in the 3D scene the users touches, and ***how*** the user interacts with the scene. For example:

- ***grabbing*** a ***door handle***
- ***pulling*** open a ***drawer***
- ***pressing*** a ***light switch***



The goal is to represent interactive scene elements in a prerecorded 3D scene (scanned with an iPhone) towards so that a robot (Spot) can replicate those actions in the real scene.

Detlitzas et al. "SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes" CVPR'24



Requirements / Tools:

PyTorch

Supervisor:

Francis Engelmann (francis.engelmann@ai.ethz.ch)

3D Scene Understanding on a City-Scale Level using Foundation Models

Goal: Obtaining high-level knowledge from 3D scenes at a city scale by using foundation models, such as VLM (e.g., CLIP).

Description:

Input: An areal 3D reconstruction of a city and the corresponding posed areal RGB image frames.

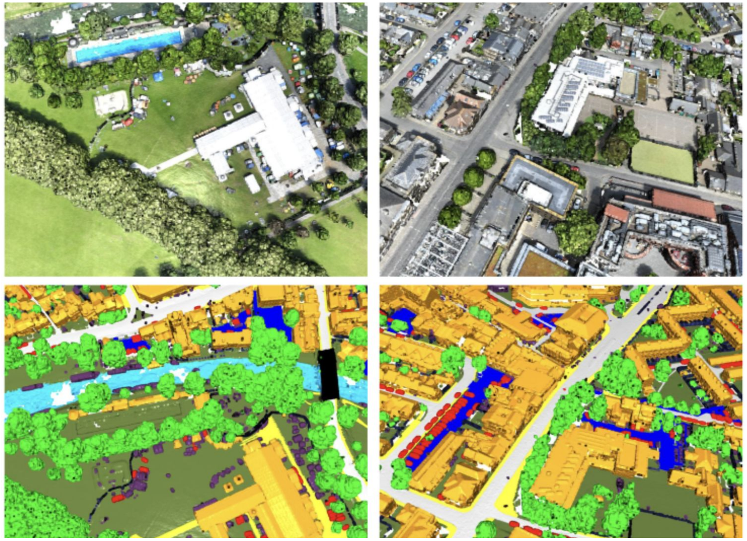
Output: A city-scale 3D representation that allows to query the city for arbitrary ***objects*** (e.g., pools, cars, train stations, schools) and ***higher-level concepts*** (housing prices, noisy areas).

The goal of this project is a scene representation at a city-scale to enable automatic inventory (for example of street furniture, benches, signs etc.) as well as extracting knowledge on a socio-economical level (high-income neighborhoods, crimes rates, voting behavior).

The key challenge in this work is to understand the capabilities of foundation models trained on street-level imagery and transfer their abilities to areas images or renderings of 3D cities.

Peng et al. “OpenScene: 3D Scene Understanding with Open Vocabularies” CVPR’23
Takmaz et al. “OpenMask3D: Open-Vocabulary 3D Instance Segmentation” NeurIPS’23

Areal 3D City Reconstruction



Semantic Understanding

Requirements / Tools:

PyTorch / Foundation models (Visual-Language Models, e.g., CLIP)

Supervisor:

Francis Engelmann (francis.engelmann@ai.ethz.ch)

Pixel accurate hologram placement with AR devices

Goal: Pixel accurate hologram placement with AR devices such as Hololens.

Description:

Hologram placement in the 3D world is a weak link in applications that require pixel accurate / cm level accuracy in hologram rendering. Since Hololens like devices are used to place and manipulate content directly in the 3D world, this is a commonly used method of content placement. However, this mode does not capture a user's intent accurately, leading to device framework related errors in perceived holograms. Placement directly in the 3D world represents a hologram in "abstract" anchors that are opaque to the user. A user may attach a hologram to a button or to corners. A user places the hologram where they perceive is the correct location and orientation. The system represents the hologram not with respect to the intended locations but in a internal anchors. Invariably, the anchors drift to create inaccurate hologram perception. We will employ explicit user intent by not simply using the 3D placement but also explicit image inputs to express placement intent. Image inputs will be used to accurately render holograms aligned to the expressed user intent thus leading to high accuracy in rendering. Key project steps: (1) Play with holograms on Hololens2. (2) Use simple interfaces to capture user intent. (3) Build algorithms for alignment of captured images to online data from Hololens2. (4) Experiment with and evaluate the algorithms in real-time.

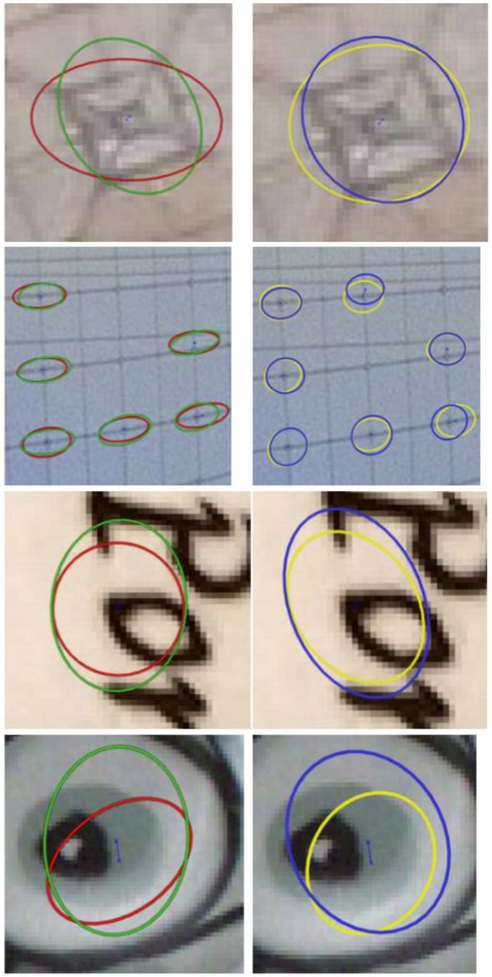
Requirements / Tools:

Hololens2. Python for showing offline experiments. C++ for online experiments. (The project is doable without real-time on HL2. With HL2 the experience is real-time.)

Supervisor:

harpreet.sawhney@microsoft.com

Learning Affine Features Consistent with the Relative Pose



Goal: The goal is to improve HesAffNet including affine consistency w.r.t. the relative pose into the loss function minimized.

Description:

The project is about improving affine shapes of affine covariant features detected by the HesAffNet algorithm [1]. This is done by directly considering the epipolar geometry (i.e., relative pose of an image pair) in the loss minimized when learning the affine frames. This can be done by either

- calculating the loss from the two geometric constraints that the epipolar geometry imply on affine correspondences, or
- by using differentiable affine-based relative pose solvers and use the pose error as the loss.

The improved affine frames are useful in a number of applications, e.g., feature matching, homography and epipolar geometry estimation.

[1] Mishkin, Dmytro, Filip Radenovic, and Jiri Matas. "Repeatability is not enough: Learning affine regions via discriminability." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

Requirements / Tools:

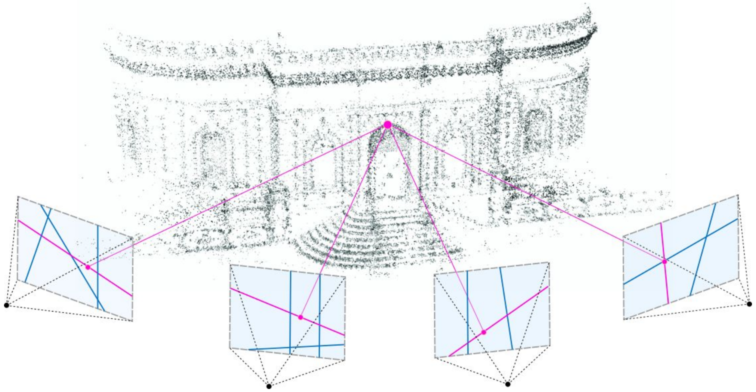
C++
Knowledge about 3D geometry

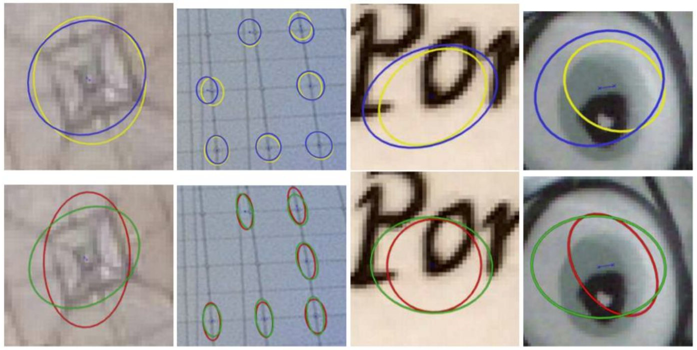
Supervisor:

Daniel Barath <dbarath@ethz.ch>
Simone Schaub-Meyer
<simone.schaub@visinf.tu-darmstadt.de>

Global Structure-from-Motion with Affine Correspondences

Goal: The goal is to create a global structure-from-motion algorithm that leverages local affine features.





Description:

The project is about implementing a global structure-from-motion algorithm that uses affine correspondences for various tasks, e.g.,

- Feature matching
- Relative pose estimation between pairs of images
- Rotation and translation averaging
- Triangulating oriented 3D points
- Bundle adjustment to get the final reconstruction

Requirements / Tools:

C++

Knowledge about 3D geometry

Supervisor:

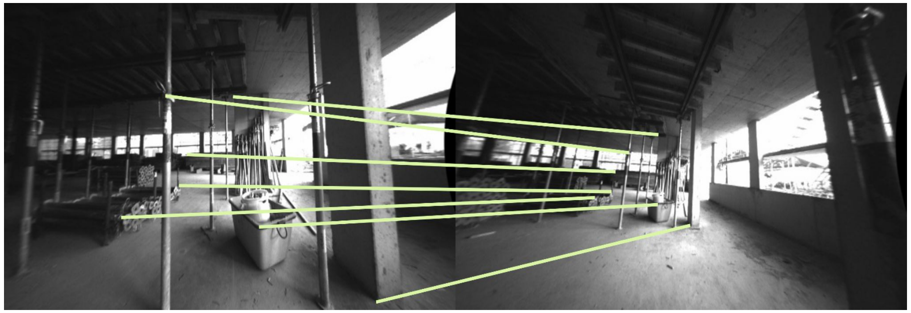
Daniel Barath <dbarath@ethz.ch>

Online Feature Selection for Visual Localization in Construction Sites

Goal: Introduce a method for online descriptor selection in construction site

Description:

This project focuses on **visual localization** by utilizing local features in construction sites. **Construction sites** are dynamic environment where localization can be challenging if the map was collected a long time before.



The goal of this project is to come up with an algorithm that would decide which features to consider for pose estimation in online fashion. We draw inspiration from Pautrat *et al* [1].

[1] Pautrat, R., Larsson, V., Oswald, M. R., & Pollefeys, M. (2020). Online invariance selection for local feature descriptors. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16 (pp. 707-724). Springer International Publishing.

Requirements / Tools:

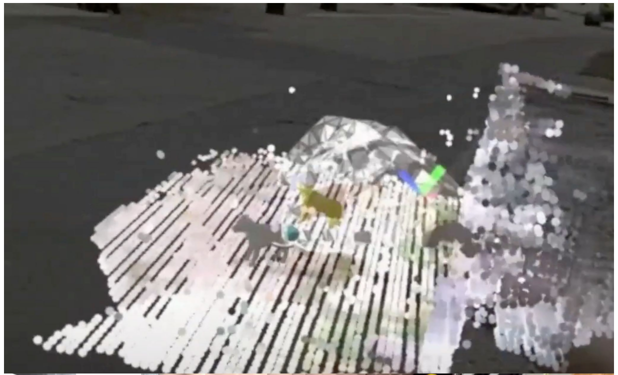
Python

Supervisor:

Olga Vysotska <ovysotska@ethz.ch>
Daniel Barath <dbarath@ethz.ch>

3D Mixed Reality Object Drag-and-Drop

Goal: Control a robot to pick and place an object by performing drag-and-drop of the object with a HoloLens.



Description:

This project combines 2 earlier works that

- control a robot in 3D through drag-and-drop
- perform pick-and-place with spot

We will use an iPhone scan of the environment to perform instance segmentation with Mask3D and send one mesh per instance to the HoloLens, where the user can move the instances through drag-and-drop.

Then, we use AnyGrasp to find a suitable grasp and plan a robot trajectory through python-rrt.

To make this interface even more useful, there is potential to add e.g.

- Smart checking of the object goal position such that it will fit
- Highlighting object categories through voice command

Requirements / Tools:

Open3d, Spot API, ROS, python

Supervisor:

Julia Chen <jiaqi.chen@inf.ethz.ch>,
Zuria Bauer <zuria.bauer@inf.ethz.ch>,
Hermann Blum <hermann.blum@inf.ethz.ch>
Francis Engelmann <francis.engelmann@ai.ethz.ch>

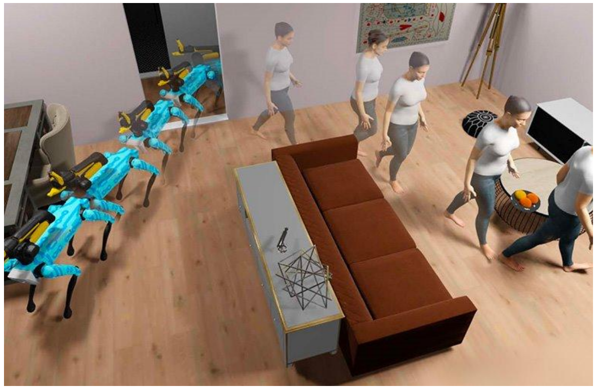
VR in Habitat 3.0

Goal: Explore the newly improved Habitat 3.0 simulator with a special focus on the Virtual Reality Features

Description:

This project is meant to be an exploration task on the Habitat 3.0 [1] simulator, exploring all the newly introduced features, focusing specifically on the implementation of virtual reality tools for scene navigation. The idea is to extend these features to self created environments in Unreal Engine that build upon Habitat.

+



[1] Puig, X., Undersander, E., Szot, A., Cote, M.D., Yang, T.Y., Partsey, R., Desai, R., Clegg, A.W., Hlavac, M., Min, S.Y. and Vondruš, V., 2023. Habitat 3.0: A co-habitat for humans, avatars and robots.

Requirements / Tools:

Unity, Python

Supervisor:

Hermann Blum <blumh@ethz.ch>
Zuria Bauer <zbauer@ethz.ch>
Boyang Sun <boysun@ethz.ch>

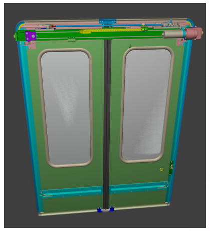
Improving Object Pose Estimation with Line Features

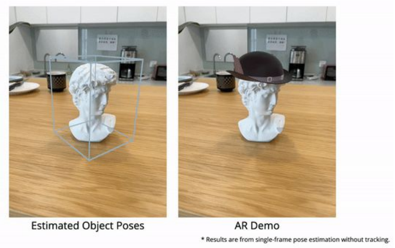
Goal: Use line features to improve 6 DoF visual object pose estimation, in particular on low-textured objects.

Description:

Line features are particularly useful for structured low-textured objects, such as the doors of the SBB trains. This project aims to integrate lines into the existing point-alone object pose estimation pipeline, which should ideally improve upon state-of-the-art practices.

The first step is to reproduce a 2d-2d hybrid pipeline as in limap [3] that is better than the 2d-2d pipeline as in hloc [1] and OnePose++ [2]. Then we move on to develop 2d-3d matching of points and lines that resembles OnePose++ [2]. The minimal goal is to make the pipeline consistently better than OnePose++ (this is natural since it just adds lines in addition to points). Bonus developments include acquiring 3D map directly from CAD models (generally available) and generalizing the method to ratio changes (e.g. the same product but a different size).




Estimated Object Poses
AR Demo
* Results are from single-frame pose estimation without tracking.

[1] hloc: https://github.com/cvg/Hierarchical-Localization
[2] He et al. OnePose++: Keypoint-Free One-Shot Object Pose Estimation without CAD Models, NeurIPS 2022.
[3] LIMAP: https://github.com/cvg/limap

Requirements / Tools:

Python, Deep Learning
Basic 3D geometry

Supervisor:

Shaohui Liu, Julia Chen, Hermann Blum

Point-Line Feature Detector and Descriptor

Goal: Design a feature detector and descriptor jointly predicting points and lines from images.

Description:

Recent 3D vision pipelines (e.g. SfM, SLAM, pose estimation, etc) are not only leveraging feature points but also other kind of local features such as line segments. Obtaining all these features is currently inefficient, as they are detected and described with different methods (e.g. one to detect + one to describe, and one for points + one for lines), while these tasks share a lot of similar computations.

In this project, we aim to unify all these methods into a single network to jointly detect and describe both point and line features. The goal is to make this step as efficient as possible, by sharing computations between points and lines, sharing common descriptors, and optimizing the backbone network with modern tools. This unified approach will be integrated into existing point-line pipelines [1][2] and has the potential to become the de-facto standard to obtain feature points and lines. Also, this project will benefit from our existing codebase/framework and preliminary developments for training joint feature detectors and descriptors.

[1] Pautrat Rémi*, Suárez Iago*, Yu Yifan, Pollefeys Marc, Larsson Viktor. GlueStick: Robust Image Matching by Sticking Points and Lines Together. ICCV 2023.

[2] Liu Shaohui, Yu Yifan, Pautrat Rémi, Pollefeys Marc, Larsson Viktor. 3D Line Mapping Revisited. CVPR 2023.

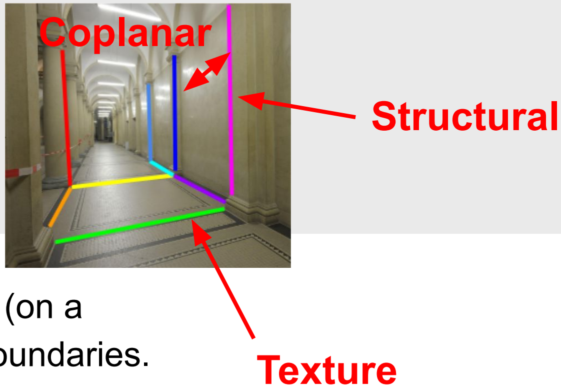
Requirements / Tools:

- Basic knowledge of computer vision and deep learning
- Python + Pytorch

Supervisor:

Shaohui Liu <shaohui.liu@inf.ethz.ch>
Rémi Pautrat <pautratrmi@microsoft.com>

Line Segment Understanding



Goal: Given an image and 2D line segments in it, infer which lines are co-planar (on a surface) and understand whether they are pure textural or are on object boundaries.

Description:

Line segments provide insights about the 3D structure of a scene, even from images only. For instance, they can be used to retrieve vanishing points and the orientation of the camera, as well as directly give the layout of rooms and buildings. However, current line detectors mainly detect all kinds of lines, including noisy ones, lines present on textured areas, and structural lines present on object boundaries.

In this project, you will implement a method to classify the 2D detected lines into either texture / structural lines, as well as to understand which lines are co-planar (on a surface), from a single image. This will be done by training a graph neural network (GNN) performing the classification for every line segment as well as pairs of line segments.

This method can then be applied to an existing 3D line reconstruction software [1], to implement an interactive demo showing only texture / structural lines, as well as all the different planes of the scene. With a successful classification of lines, this project would open the door to exciting applications such as room layout extraction, CAD model reconstruction from images only, etc, and could lead to a publication.

[1] Liu Shaohui, Yu Yifan, Pautrat Rémi, Pollefeys Marc, Larsson Viktor. 3D Line Mapping Revisited. CVPR 2023.

Requirements / Tools:

- Basic knowledge of computer vision and deep learning
- Python + Pytorch

Supervisor:

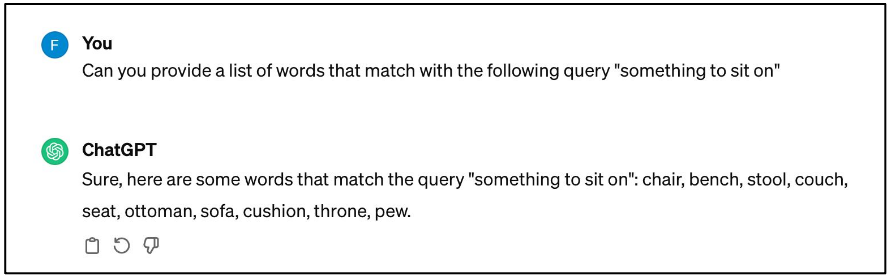
Shaohui Liu <shaohui.liu@inf.ethz.ch>
Rémi Pautrat <pautratrmi@microsoft.com>

3D Language Embedding for Improved Open-Vocabulary Scene Understanding

Goal: Infer affordances, functionalities and interactions in 3D scenes by combining a large-language-model (LLM e.g. ChatGPT) with visual language-models (VLM)

Description:

Existing open-vocabulary 3D scene understanding methods rely only on the knowledge embedded in VLMs such as CLIP and encode it all into a single feature vector. The goal if this project is to investigate, via the use of LLM (ChatGPT) how arbitrary user queries can be rephrased to have more explicit, dedicated feature representations that are query-dependent.



You
Can you provide a list of words that match with the following query "something to sit on"

ChatGPT
Sure, here are some words that match the query "something to sit on": chair, bench, stool, couch, seat, ottoman, sofa, cushion, throne, pew.

Detlitzas et al. "SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes" CVPR'24

Requirements / Tools:

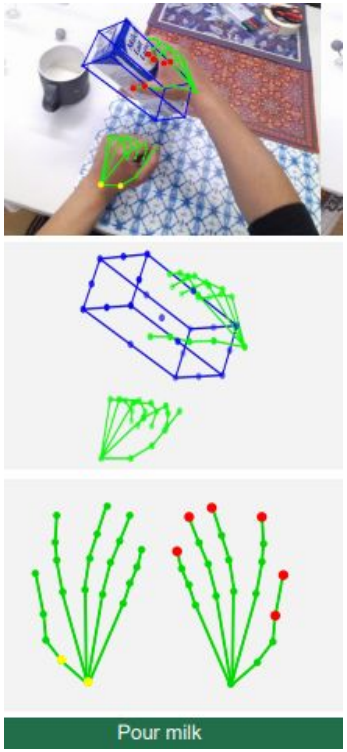
PyTorch / Foundation models (Visual-Language Models, e.g., CLIP)

Supervisor:

Francis Engelmann
Hermann Blum

Action Recognition Using 3D Hand-Object Contact-Map

Goal: The primary objective of this project is to use an enhance representation of 3D hand and object interaction to improve action recognition accuracy.



Description:

Action recognition is an essential task in computer vision and has numerous applications in various fields, including robotics, surveillance, and healthcare. The recognition of actions involves the analysis of temporal and spatial information within a video sequence. Current state-of-the-art methods use 3D hand and object poses for action recognition, where the object's corners are commonly used for representation. However, this approach has limitations in accurately modeling the hand-object interaction. In [1], we show leveraging hand-object contact-map representation helps to improve action recognition. However, this representation can be learned implicitly for the task of action recognition. We aim to achieve this objective through the following tasks:

1. Use existing datasets which has annotated 3D hand and object pose with action labels.
2. Investigate features extracted from RGB and/or 3D Skeletal streams.
3. Investigate different architecture (e.g., transformer) for implicit contact-map prediction.
4. Compare the performance of the proposed method with state-of-the-art methods.

[1] https://arxiv.org/pdf/2309.10001.pdf

Requirements / Tools:

Python, Deep Learning
Basic 3D geometry

Supervisor:

Mahdi Rad <mahdirad@microsoft.com>
Taein Kwon <taein.kwon@inf.ethz.ch>

Action Label Correction with LLMs

Goal: The primary objective of this project is to leverage LLMs to improve the action recognition accuracy.

Description:

The recent development of LLMs (Large Language Models), such as ChatGPT and Llama, opens up new possibilities for understanding procedural actions. In the past, action recognition was restricted to the classification of visual frames. However, with LLMs, the model can observe the whole action sequence in a more effective way and even predict the future actions [1]. In this project, students will explore how LLMs can improve action recognition in procedural tasks. Specifically, given a high-level procedural task (e.g., making coffee, copying a paper), students will use existing pretrained action recognition models to predict the top 5 actions for each clip and feed them into the LLMs to refine and correct the predicted actions. As a comparison, students will also establish a baseline using simple machine learning and statistical methods to correct actions.

[1] Palm: Predicting Actions through Language Models @ Ego4D Long-Term Action Anticipation Challenge 2023, CVPR'23 workshop

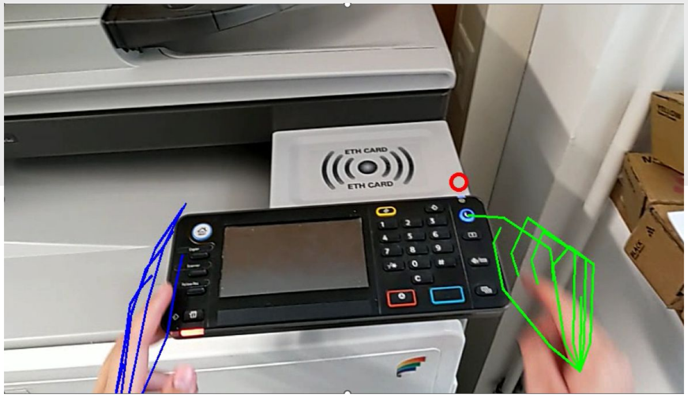
Requirements / Tools:

Python, Deep Learning

Supervisor:

Taein Kwon <taein.kwon@inf.ethz.ch>
Mahdi Rad <mahdirad@microsoft.com>

Holographic Guidance App in MR



Goal: Implement an app that can guide 3D hand pose

Description:

Reading text manuals to set up and manipulate devices not only takes a lot of time but also is not intuitive when it comes to 3D instruction. Despite the advent of Mixed Reality (MR) devices, 3D instruction is still limited and expensive to set up. In this project, we will develop an app, an adaptive 3D hand guidance system that projects instructional 3D hand poses in MR devices with pre-recorded instructional videos using MR devices.

In this project, the students will

- Collect instruction videos using the existing recording app (link)
- Develop an app that takes hand pose inputs and show the possible hand poses to guide an user by calculating the similarity scores between hand inputs and pre-recorded hand poses.
- Explore methods to guide hand adaptively, for example, filtering methods, K-Nearest Neighbor (KNN) or Dynamic Time Warping (DTW)

Requirements / Tools:

3D Vision, C# , Unity

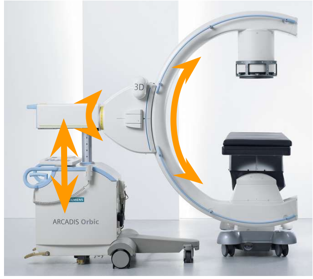
Supervisor:

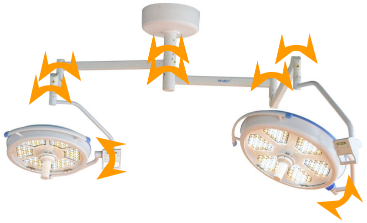
Taein Kwon (taein.kwon@inf.ethz.ch)
Jonas Hein (jonas.hein@inf.ethz.ch)

6DoF Pose Estimation for Articulated Objects

Goal:

Develop a method to estimate the 6DoF pose of articulated objects.





Description:

Surgery digitization aims to create a virtual replica of a surgery, often referred to as a digital twin. In order to merge the information from multiple devices in a shared representation, their spatial relations need to be estimated.

A potential approach extends the SurfEmb model [1] with a part segmentation and a refined sampling and optimization process that maximises the consistency of the estimated pose and configuration with the estimated 2D-3D correspondences. A virtual environment for the generation of realistic training data is available.

The goal of this project is to develop a 6DoF pose estimation method for articulated objects, such as a mobile C arm, OR lamps, -displays, and -tables. Optionally, the developed method can also be applied to articulated surgical instruments like scissors, forceps, or rasps.

[1] Haugaard, R. L., & Buch, A. G. (2022). Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 6749-6758).

Requirements / Tools:

Python, PyTorch
Blender

Supervisor:

Jonas Hein <jonas.hein@inf.ethz.ch>

Reconstructing Material Properties of Surgical Instruments

Goal: Develop a method to 3D reconstruct objects with accurate material properties.

Description:

Training in simulation or on synthetic data can significantly reduce the need to capture and annotate real data. However, the quality of the synthetic data greatly determines the performance of the trained model due to the synth-real domain gap. Thus, an accurate reconstruction of the material properties [1,2] is highly desirable.
The goal of this project is to reconstruct the material properties of surgical instruments to improve the quality of generated synthetic training data. A commercial 3D scanner serves as a baseline. While its spatial resolution and accuracy are great, the computed textures often suffer from baked-in reflections and related artifacts.

A potential approach to this problem may include the following steps:

- Estimate the material properties of an object from several (un-)calibrated photographs.
- Compute a mapping from the estimated texture and materials onto the 3D object surface.

[1] Ikehata, S. (2023). Scalable, Detailed and Mask-Free Universal Photometric Stereo. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13198-13207).
[2] Rodriguez-Pardo, C., Dominguez-Elvira, H., Pascual-Hernandez, D., & Garces, E. (2023). UMat: Uncertainty-Aware Single Image High Resolution Material Capture. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5764-5774).

Requirements / Tools:

Python, PyTorch
Blender

Supervisor:

Jonas Hein <jonas.hein@inf.ethz.ch>
Lilian Calvet <Lilian.Calvet@balgrist.ch>

Improving Multi-Modal and Multi-View Camera Calibration

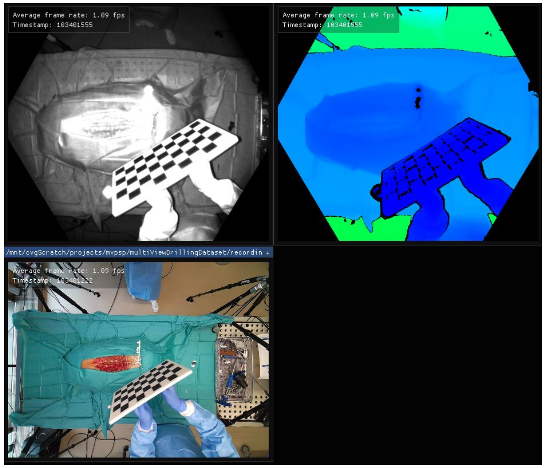
Goal: Develop an improved method to calibrate a multi-camera system that utilizes multiple modalities.

Description:

Our current multi-camera calibration method optimizes the camera extrinsics independently for each camera, while using a stereo IR tracking system as a reference.

The goal of this project is to improve the calibration baseline. Potential improvements include:

- Integration of a bundle adjustment step
- Integration of checkerboard detections on the IR images
- Detecting IR-reflective fiducials in the IR images



```
Average frame rate: 1.09 fps
Timestamp: 103401555
```

```
Average frame rate: 1.09 fps
Timestamp: 103401555
```

```
/mnt/cvgScratch/projects/mvpsp/multiViewDrillingDataset/recordin +
Average frame rate: 1.09 fps
Timestamp: 103401222
```

Requirements / Tools:

C++ or Python

Supervisor:

Jonas Hein <jonas.hein@inf.ethz.ch>

Multi-View 6DoF Object Pose Estimation on HMDs

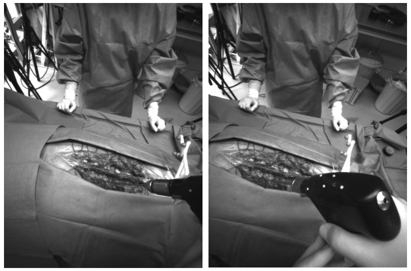
Goal: Develop a multi-view 6DoF object pose estimation method for HMDs.

Description:

Single-view pose estimation methods generally suffer from depth ambiguities. In contrast, stereo or multi-view methods can solve these depth ambiguities by implicit or explicit triangulation. Recent HMDs are usually equipped with multiple RGB, grayscale, or IR cameras to sense their environment.

The goal of this project is to implement a object pose estimation method that utilizes multiple or all of these available sensors to improve the accuracy of the 6DoF pose estimation. Ideally, single-view [1], stereo, and multi-view [2] methods are compared. A synthetic and real training dataset including camera parameters are provided.

[1] Haugaard, R. L., & Buch, A. G. (2022). Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 6749-6758).
[2] Haugaard, R. L., & Iversen, T. M. (2023, May). Multi-view object pose estimation from correspondence distributions and epipolar geometry. In 2023 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1786-1792). IEEE.



Requirements / Tools:

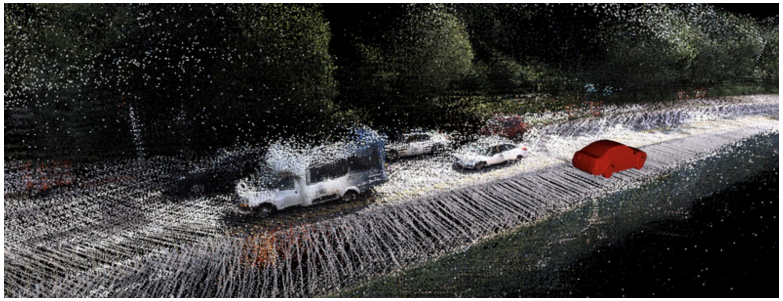
C++ or Python

Supervisor:

Jonas Hein <jonas.hein@inf.ethz.ch>

Dense Point Tracking for Visual SLAM

Goal: Improve the efficacy and efficiency of dense visual SLAM approaches with dense point tracking.



Description:

Dense SLAM is a key challenge in autonomous driving and robotics. Compared to the complex, multi-modal systems deployed today, camera-based systems provide a simpler, low-cost alternative. Yet, camera-based dense SLAM of complex dynamic scenes has proven extremely difficult. While previous works rely on costly and incoherent pair-wise optical flow estimates, the **goals** of this project are:

- **Integrate** a **dense point tracking** approach into a **visual SLAM** framework
- Show its effectiveness for improving performance and runtime of the framework

References:

[1] De Moing et al., “Dense Optical Tracking: Connecting the Dots”, arXiv 2023.

[2] Schmied et al., “R3D3: Dense 3D Reconstruction of Dynamic Scenes from Multiple Cameras”, ICCV 2023.

Requirements / Tools:

Python, PyTorch

Multi-view Geometry

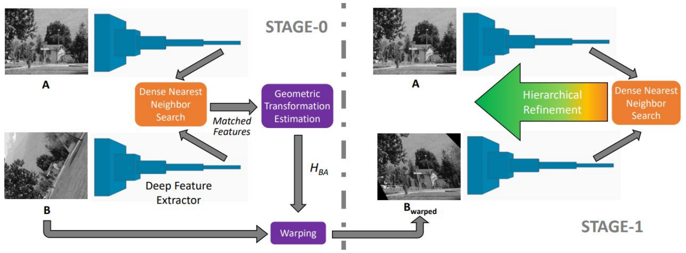
Supervisor:

Tobias Fischer <tobias.fischer@inf.ethz.ch>

Geometry Guided Dense Feature Matching

Goal: Use epipolar geometry to guide dense matching before / during SfM

Description:



Establishing correspondences between two images is a fundamental task for many Computer Vision systems such as SLAM, Structure-from-Motion or Visual Localization.

The typical pipeline involves detecting keypoints and descriptors in each image, and then match these either with a NN-search in descriptor space or by utilizing Graph Neural Networks. Recently, Efe et al.[1] have shown that utilizing basic geometry (in their case homographies) can help finding better correspondences. More general, epipolar or projective geometry could be used to guide the matching process.

The goal of this project is to design a **matching algorithm** that takes **dense feature maps**[2] of each image as input, and yields correspondences between images **guided by epipolar / projective geometry**.

[1] Efe et al., "DFM: A Performance Baseline for Deep Feature Matching", CVPRW 2021
[2] Germain et al., "S2DNet: Learning Accurate Correspondences for Sparse-to-Dense Feature Matching", ECCV 2020

Requirements / Tools:

Python, PyTorch, glue-factory

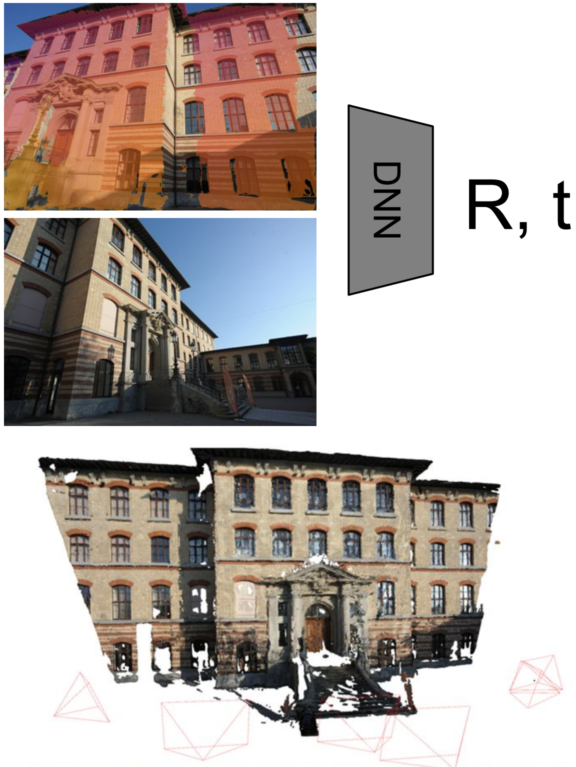
Basic 3D Geometry

Supervisor:

Philipp Lindenberger<philipp.lindenberger@inf.ethz.ch>

Correspondence-free Visual Localization from RGB to RGB-D images

Goal: Train a network that regresses the absolute pose of an RGB image given a RGB-D image



Description:

Visual Localization typically relies on 2D-3D correspondences between a query and a reference image to estimate the 6DoF pose of an image. Typically, 2D-2D correspondences between images are estimated first and then lifted to 3D via a sparse 3D reconstruction [1]. Recent deep methods [2] directly regress the pose with both priors (image + sparse 3D model), but do not achieve SotA performance on popular benchmarks.

The goal of this project is to design and train a network that utilizes **dense geometry** in the **reference image (RGB-D)**, and **directly aligns the RGB image** without estimating correspondences.

References:

[3] Sarlin, Paul-Edouard, et al. "From Coarse to Fine: Robust Hierarchical Localization at Scale", CVPR 2019
[2] Sarlin, Paul-Edouard, et al. "Back to the Feature: Learning Robust Camera Localization from Pixels to Pose", CVPR 2021

Requirements / Tools:

Python, PyTorch, glue-factory

Basic 3D Geometry

Supervisor:

Philipp Lindenberger<philipp.lindenberger@inf.ethz.ch>

Real-Time Deep Multi-View Stereo

Goal: Develop an efficient deep learning-based multi-view stereo network, ideally in real time

Description:

Given multiple 2D RGB observations and camera parameters, Multi-View Stereo (MVS) aims to reconstruct the dense geometry of the scene. MVS is a fundamental task in 3D computer vision, with applications ranging from autonomous navigation to virtual/augmented reality. Despite the remarkable progress in the last few years, existing deep MVS models are usually computational expensive, which hinders their application and deployment in real-world applications like robotics that typically requires real-time feedback. This project aims to develop an efficient MVS network, ideally in real time.

This project will start with being familiar with several representative baselines (e.g., IterMVS [1] and Effi-MVS [2]) and implementing the full pipeline with pytorch. We will then carefully evaluate and compare the speed/accuracy trade-offs of different components in the full pipeline, and propose methodological and/or engineering improvements to achieve real-time speed while maintaining competitive accuracy. Finally we will show some 3D reconstruction demos with the developed network.

References:

[1] Wang et al. IterMVS: Iterative Probability Estimation for Efficient Multi-View Stereo. CVPR 2022
[2] Wang et al. Efficient Multi-view Stereo by Iterative Dynamic Cost Volume. CVPR 2022

Requirements / Tools:

Python, PyTorch

Basic Multi-View Geometry

Supervisor:

Haofei Xu <haofei.xu@inf.ethz.ch>
Fangjinhua Wang <fangjinhua.wang@inf.ethz.ch>

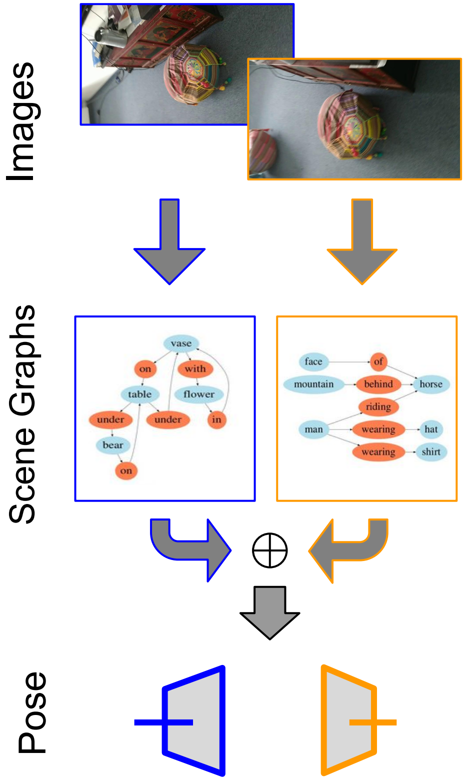
Estimating Relative Pose using Scene Graphs

Goal: Construct Scene Graphs for two images. Use the correspondences between these scene graphs to find relative pose between the images.

Description:

Relative pose estimation is essential for many computer vision applications, i.e. 3D reconstruction, NeRF/Gaussian Splatting initialization, and SLAM. Typically, relative position estimation relies on point correspondences. It is notoriously difficult to estimate the point correspondences between images with large viewpoint or illumination changes. Scene Graph is a modern, lightweight scene representation that leverages objects in the scene and the relations between them. Utilizing these, higher-level features helps to overcome the limitation of the point features. The Scene Graph representation has found applications in various computer vision tasks.

The goal of the project is to construct a scene graph for each image, and to establish correspondences between the scene graphs. Finally, these correspondences will be leveraged for relative pose estimation.



Requirements / Tools:

Python, Basic 3D Geometry

Supervisor:

Petr Hruby <petr.hruby@inf.ethz.ch>
Zuria Bauer <zuria.bauer@inf.ethz.ch>

Using Language for 3D Geometry Estimation

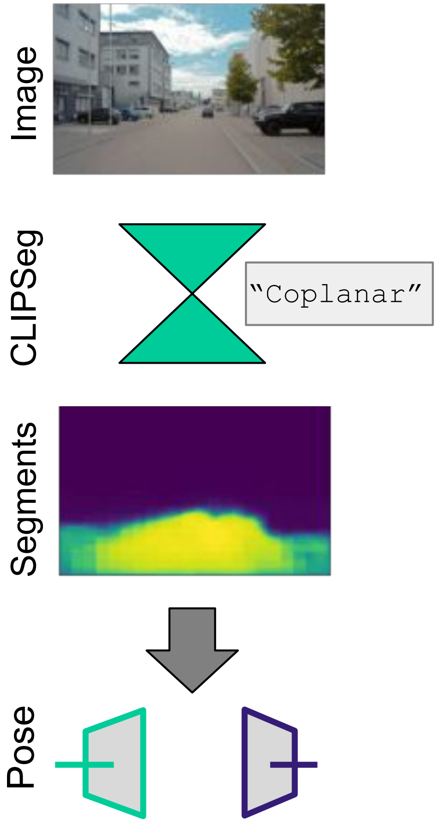
Goal: Utilize CLIP features for extracting regions with a special geometry to improve performance in geometric tasks.

Description:

CLIPSeg [1] is a neural network able to segment out image regions based on a text query. It leverages CLIP features, which are able to capture the relations between image and text.

Geometric tasks (i.e. pose estimation), are crucial for many computer vision applications, such as 3D reconstruction, initialization of NeRf/Gaussian Splatting, and SLAM. Solving these geometric tasks becomes much easier, if we know, which parts of the scene have special geometric properties (coplanar, horizontal, vertical, near, far).

The goal of the project is to explore the ability of the CLIPSeg method to segment out regions of images having special geometric properties, and to utilize this knowledge for solving geometric problems. All minimal solvers will be provided by the supervisors.



[1] T. Lüddecke, A. Ecker: Image Segmentation Using Text and Image Prompts

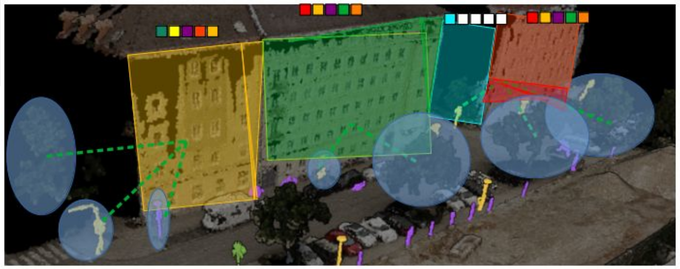
Requirements / Tools:

Python, Basic 3D Geometry

Supervisor:

Petr Hruby <petr.hruby@inf.ethz.ch>
Zuria Bauer <zuria.bauer@inf.ethz.ch>

Scene Graphs for Outdoor Localization



Goal: Adapt the Scene Graph representation to outdoor scenes, while maintaining its suitability for visual localization.

Description:

Scene Graph is a modern, lightweight scene representation leveraging objects in the scene and the relations between them. Scene Graphs have been successfully used for localization in indoor scenes, surpassing standard point-based methods on challenging queries with blur or large changes of illumination or viewpoint.

The goal of this project is to examine the possibility of adaptation of the Scene Graph representation to outdoor scenes, while maintaining its suitability for visual localization.

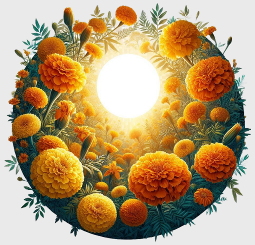
Since the number of object classes in outdoor scenes is lower than that in indoor scenes, it may be helpful to encode the properties of particular objects in the scene. For instance, for a house, we may encode the material of the facade, shape of the windows, type of the roof, any text on the house, etc. A part of the task will be to explore the abilities of the foundational models to extract this type of information from images.

Requirements / Tools:

Python, Basic 3D Geometry

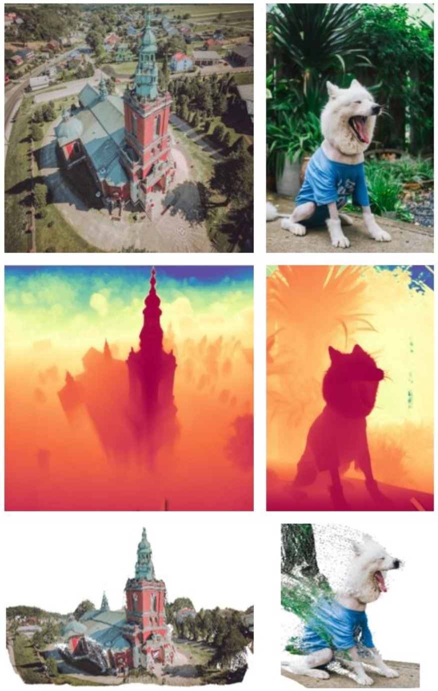
Supervisor:

Petr Hruby <petr.hruby@inf.ethz.ch>
Zuria Bauer <zuria.bauer@inf.ethz.ch>



Is it all sunshine and Marigolds?

Goal: Research the usability of the depth predictions from Marigold for the purpose of image alignment



Description:

Recently, works to improve monocular depth predictions in the wild were introduced by the research community [1, 2]. However, the interest of this project relies on the study of the usability of these networks for 3D reconstruction tasks.
Possible steps for this project:

- Detailed understanding of the Marigold paper and pipeline.
- In depth analysis of single view performance and failure cases.
- Transform the output of Marigold into metric depth.
- Investigate alignment protocols between different Marigold point clouds.

[1] Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C. and Schindler, K., 2023. Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation. Webpage: https://marigoldmonodepth.github.io/
[2] Saxena, S., Hur, J., Herrmann, C., Sun, D. and Fleet, D.J., 2023. Zero-Shot Metric Depth with a Field-of-View Conditioned Diffusion Model.

Requirements / Tools:

3D Geometry Understanding, PyTorch

Supervisor:

Mihai Dusmanu,
Zuria Bauer <zuria.bauer@inf.ethz.ch>

Map-aware 3D Multiple Object Tracking

Goal: Utilize city map information with Bird-Eye-View feature to perform camera-based 3D multiple object tracking

Description:

Recently, works to lift image features to bird-eye-viewed space to perform 3D object detection has achieved a huge success [1]. Moreover, some work further utilize the map information to help the overall result [2]. On the other hand, how to conduct the appearance-based tracking approach directly on the BEV space lacks of exploring. To this end, this project aims to utilize the map information not just for detection but also for betting tracking. We will focus on how the lift the correct appearance feature to BEV grid but also use map to help the motion model.

[1] Huang, Junjie, et al. "Bevdet: High-performance multi-camera 3d object detection in bird-eye-view." arXiv preprint arXiv:2112.11790 (2021).

[2] Chang, Mincheol, et al. "BEVMap: Map-Aware BEV Modeling for 3D Perception." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024.

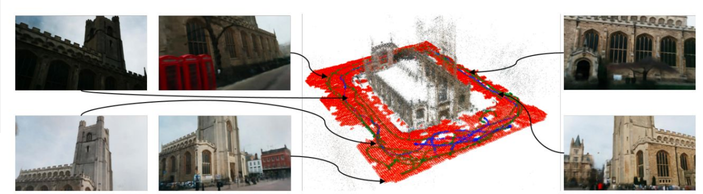
Requirements / Tools:

PyTorch

Supervisor:

Yung-Hsu Yang (yangyun@ethz.ch)

Enhancing Visual Localization with Novel View Synthesis



Goal:

Synthesize new database images with novel view synthesis;

Improve visual localization quality for scene coordinate regression methods in sparse-view setting;

Description:

Visual localization describes the task of estimating the camera position and orientation for a query image in a known scene. Scene coordinate regression (SCR) methods [1] are a family of visual localization methods that directly regress 2D-3D matches for camera pose estimation. They are effective in small-scale scenes with small map size and fast training. In large-scale scenes, SCR methods usually separate the scene space and assign a network for each part, i.e. ensembles. Recently, we propose GLACE, a new SCR method that is able to perform localization in large-scale scenes with a single model. However, same as other SCR methods, GLACE cannot handle sparse input images well since the training needs dense images to triangulate well. Recently, NeRF-based methods [2,3,4] and gaussian-splatting-based methods [5] are popular in novel view synthesis and are able to synthesis novel viewpoints with high quality.

In this project, the goal is:

- densify the database images with novel view synthesis,
- train scene coordinate regression methods with both real and synthesized images to improve performance.

[1] Brachmann et al. Accelerated Coordinate Encoding: Learning to Relocalize in Minutes using RGB and Poses. CVPR 2023.
[2] Mildenhall et al. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. ECCV 2020.
[3] Barron et al. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. ICCV 2023.
[4] Mueller et al. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. SIGGRAPH 2022.
[5] Kerbl et al. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. SIGGRAPH 2023.
[6] Moreau et al. LENS: Localization enhanced by NeRF synthesis.

Requirements / Tools:

Python, PyTorch

Basic Multi-View Geometry

Supervisor:

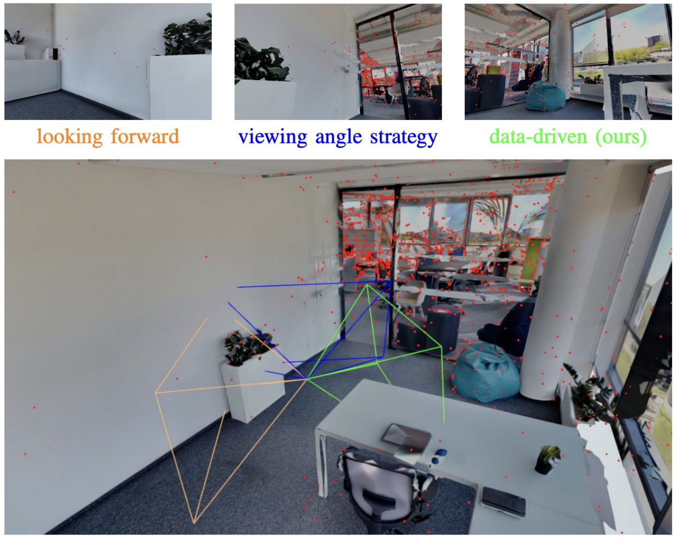
Fangjinhua Wang (fangjinhua.wang@inf.ethz.ch)
Xudong Jiang (xujiang@student.ethz.ch)

Learning-based Active Visual Localization

Goal: Build a viewpoint-selection model for mobile robots to select best viewpoint for visual localization

Description:

Visual localization describes the task of estimating the camera position and orientation for a query image in a known scene.This task has garnered significant focus, especially on enhancing the accuracy of localization from a specific viewpoint. Unlike static methods, a mobile robot has the capability to choose its viewing angle actively to refine the localization process. Our prior research[1] introduced a method for assessing viewpoints using the scene's Structure-from-Motion model, effectively ranking each sampled viewpoint. In this project, we want to move forward from the "sampling-and-grading" scheme, to an approach that directly predicts a "good" viewpoint at a given 3D location.



References:

[1] Hanlon et al. Active Visual Localization for Multi-Agent Collaboration: A Data-Driven Approach. ICRA 2024

[2] Zhou et al. Is Geometry Enough for Matching in Visual Localization? ECCV 2022

Requirements / Tools:

Python, PyTorch

Basic knowledge of SfM and deep learning

Supervisor:

Boyang Sun(boyang.sun@inf.ethz.ch)
Hermann Blum (blumh@ethz.ch)

Modeling Monocular Depth Uncertainties for 3D Reconstruction

Goal: Research what kinds of monocular depth uncertainties we need for 3D reconstruction and how to model them

Description:

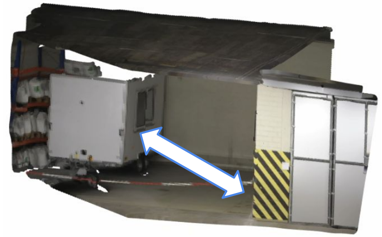
Recent works (e.g., [1]) have shown that monocular depth estimation networks provide priors that can be useful for 3D reconstruction tasks. Furthermore, in recent years, monocular estimators on depth and surface normals have been booming. However, the contribution of these priors in 3D reconstruction is still limited due to the unreliability of these networks. To fully leverage the potential of depth estimators, we need to quantify when the estimates are reliable.

Project steps:

- Build a pipeline for benchmarking the quality of depth uncertainty estimates
- Explore what kinds of uncertainties are the most relevant to the reliability of the monocular depth prediction (see figures [2])
- model depth uncertainties accordingly in the most decent way

Bonus: Integrate the depth uncertainty in existing Structure-from-Motion pipeline and investigate how much we can gain.

Is the absolute scale of our depth estimate accurate?



Is the relative depth estimate accurate?

Is the local geometry estimated accurate?

Requirements / Tools:

PyTorch, Basic 3D Geometry

[1] Liu et al. "Depth-guided sparse structure-from-motion for movies and tv shows." CVPR, 2022.
[2] Yin, et al. "Metric3d: Towards zero-shot metric 3d prediction from a single image." CVPR. 2023.

Supervisor:

Zador Pataki <zador.pataki@inf.ethz.ch>
Shaohui Liu <shaohui.liu@inf.ethz.ch>

Monocular Priors for the improvement of Virtual Object Placement

Goal: This project aims to investigate how we can exploit priors like depth or normals to improve technologies like Scandit's MatrixScan

Description:

MatrixScan is a core product of Scandit, where the user is able to detect multiple codes, place virtual objects in the scene, and track their movement. MatrixScan provides an intuitive pipeline to significantly improve the workflow and productivity of our customers.

We require the pipeline to run efficiently on very low compute mobile devices. This poses a large number of constraints, for example, rendering most advances in localization and mapping computationally intractable. This project aims to explore how we can utilize monocular priors, such as depth, to further improve the robustness of our solution.

This project will allow you to gain industrial research experience while tackling real-life problems.

Requirements / Tools:

PyTorch, and Basic computer vision / 3D vision

Supervisor:

Menelaos Kanakis <menelaos.kanakis@scandit.com>
Zador Pataki <zador.pataki@inf.ethz.ch>

Faster Training of Scene Landmark Detectors to enable Fast Visual Localization

Goal: Explore faster training of CNN-based keypoint detection architectures to recognize salient landmarks (3D points) within pre-mapped environments.

Description:

The visual localization task requires estimating the camera position and orientation for a query image in a known scene, i.e. a scene which has been pre-mapped using structure-from-motion (SfM) and/or SLAM. Scene landmark detection (SLD) proposed recently [1,3] proposes a new approach for visual localization that involves detecting a set of pre-specified scene landmarks within the scene whose 3D coordinates are known. Detecting the landmarks in query images provides 2D--3D point correspondences which can be utilized to calculate the camera pose of the query image.

However, the heatmap-based CNN architectures used for scene landmark detection are currently trained from scratch on each scene, and the training stage can take hours depending on the size of the scene and the number of scene landmarks. The goal in this project is to investigate alternative ways to train the CNN architectures that can significantly reduce the training time. Another related goal would be to demonstrate that SLD could be trained on a smaller amount of training data. Perhaps, this can be achieved by extending the SLD architecture to one where there is a universal scene-agnostic backbone which is pretrained and fixed, and a scene-specific layer (or head), which must be trained for landmarks from each specific scene.

Faster training was recently shown to be possible for scene coordinate regression (SCR) methods. The goal here is to achieve similar improvements for SLD-based methods which can scale to much larger scenes compared to SCR-based methods.

[1] Do et al. Learning to Detect Scene Landmarks for Camera Localization. CVPR 2022.
[2] Brachmann et al. Accelerated Coordinate Encoding: Learning to Relocalize in Minutes using RGB and Poses. CVPR 2023.
[3] Do and Sinha, Improved Scene Landmark Detection for Camera Localization. 3DV 2024.

Requirements / Tools:

- Basic 3D Computer Vision fundamentals.
- PyTorch, COLMAP,
- https://github.com/microsoft/SceneLandmarkLocalization

Supervisor:

Sudipta N. Sinha (sudipta.sinha@microsoft.com)

(https://snsinha.github.io/)

Learning Salient Scene Features for Robust Visual Localization

Goal: Train a neural net to predict a saliency or attention map to filter local feature candidates in a query image, and retain scene-specific features that are highly discriminative and stable over time, that will lead to accurate and robust camera localization within the pre-mapped scene.

Description:

The visual localization task requires estimating the camera position and orientation for a query image in a known scene, i.e. a scene which has been pre-mapped using structure-from-motion (SfM) and/or SLAM. Scene landmark detection (SLD) proposed recently [1,3] proposes a new approach for visual localization that involves detecting a set of pre-specified scene landmarks within the scene whose 3D coordinates are known. Detecting the landmarks in query images provides 2D--3D point correspondences which can be utilized to calculate the camera pose of the query image.

However, the SLD approach trains a CNN architecture to distinguish between thousands of scene landmarks which can be a challenging task. The goal in this project is to explore a simpler approach to learning to extract scene-specific salient features, that can scale to larger and complex time-varying scenes. Given a SfM reconstruction of the scene, posed images, and a filtered 3D point cloud, the idea is to generate a saliency or attention map for each posed image, where the saliency score at a pixel in an image reflects how useful that feature is for solving the downstream camera localization task.

The idea then is to train a simpler neural net that given a query image predicts this saliency or attention map. The predicted attention map could then be combined with an existing structure-based visual localization method such as hloc [2]. Specifically, the attention map would be used to filter the underlying SuperPoint (or another existing) feature map so that uniformative features in the query image can be pruned or suppressed. If the idea works well, it could be used to recognize more informative, stable-over-time features in the specific scene, leading to more accurate 2D–3D correspondences which will then be used to calculate the 6-DofF camera pose.

[1] Do et al. Learning to Detect Scene Landmarks for Camera Localization. CVPR 2022.
[2] From Coarse to Fine: Robust Hierarchical Localization at Large Scale, CVPR 2019.
[3] Do and Sinha, Improved Scene Landmark Detection for Camera Localization. 3DV 2024.

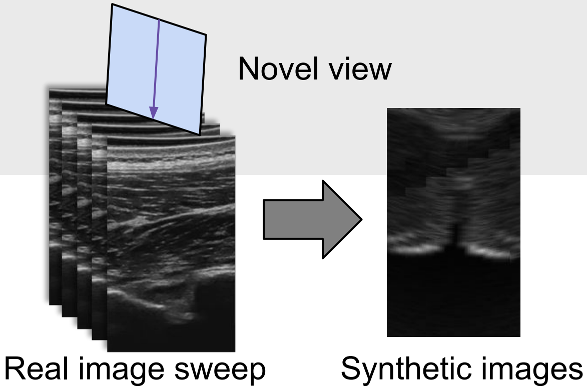
Requirements / Tools:

- Basic 3D Computer Vision fundamentals, PyTorch, COLMAP, hloc
- https://github.com/cvg/Hierarchical-Localization
- https://github.com/microsoft/SceneLandmarkLocalization

Supervisor:

Sudipta N. Sinha (sudipta.sinha@microsoft.com)
(https://snsinha.github.io/)

Gaussian Splatting for Ultrasounds



Goal: Adapt Gaussian splatting for novel view synthetic ultrasound imaging generation, and compare with Ultra-NeRF

Description:

Ultrasound (US) imaging has been widely used in the medical domain, including diagnosis and surgery. Compared to other medical modalities, its advantages include high frequency and non-invasiveness. Recent progress of deep learning techniques facilitates more comprehensive diagnosis and analysis based on US. But for some applications, available US dataset are often too sparse to train neural networks.

In this sense, realistic synthetic US imaging generation technique is particularly useful for augmenting the training dataset or validating the methods. However, it is often challenging for traditional model-based approach because of complex noise patterns. In this project, we intend to investigate the recent high performance rendering approaches such as NeRF [1] and Gaussian Splatting [2] for novel-view synthetic US imaging generation based on real images.

[1] Wysocki et. al, Ultra-NeRF: Neural Radiance Fields for Ultrasound Imaging, MIDL 2023
[2] Kerbl et. al, 3D Gaussian Splatting for Real-Time Radiance Field Rendering, ToG 2023

Requirements / Tools:

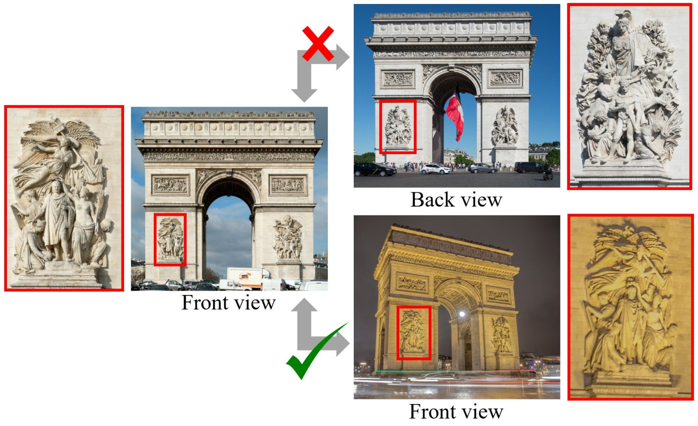
PyTorch, Basic 3D Geometry

Supervisor:

Linfei Pan <linfei.pan@inf.ethz.ch>
Yunke Ao <yunke.ao@ai.ethz.ch>

Global Structure-from-Motion with Symmetry

Goal: Tackling symmetry in global structure-from-motion pipelines



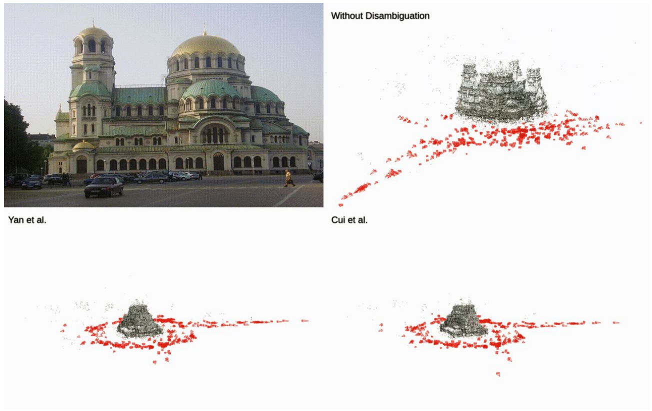
Description:

Symmetry generally poses large problem to Structure-from-Motion. When scenes are highly similar, incremental SfM (eg. COLMAP [1]) can results in reconstructions with “ghost” structures. In this project, we aim at addressing such problem in the global SfM pipelines.

Global SfM pipelines fundamentally differ than incremental ones by considering all images simultaneously while incremental pipelines register image one by one.

In this regard, symmetry in global SfM can be addressed following two paths:

1. Doppelgangers [2] presents a new learning-based method for disambiguate matches. Following a similar strategy, it should be possible to develop symmetry-aware matching schemes
2. As global SfM pipelines have global information, ambiguous matches would show a clustered pattern. Analyzing such patterns could reveal the underlying repetitive structure.



[1] Schönberger, et. al, Structure-From-Motion Revisited, CVPR 2016
[2] Cai et. al, Doppelgangers: Learning to Disambiguate Images of Similar Structures, ICCV 2023

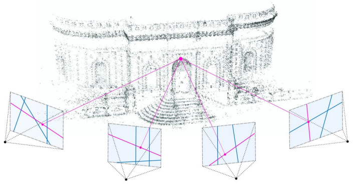
Requirements / Tools:

- PyTorch, Basic C++, Basic 3D Geometry
- https://github.com/cvg/sfm-disambiguation-colmap

Supervisor:

Philipp Lindenberger <philipp.lindenberger@inf.ethz.ch>
Linfei Pan <linfei.pan@inf.ethz.ch>

SfM / Tracking from 2D Vertical Lines

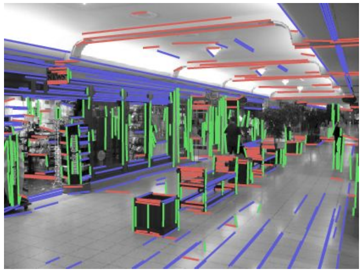


Goal: Develop a SfM system (up-to-height) from 2D vertical line detection given known gravity direction

Description:

In man-made scenes vertical lines are omnipresent. With known vertical (gravity/IMU) those lines can easily be identified and triangulated from two views. These vertical lines alone can be directly used for up-to-height structure-from-motion (i.e. recovering the 2D location of each image), and has potential to be robust to low-light condition (where robust tracking is generally very difficult). Specifically, SfM initialization can be obtained from 3 views of 5 (calibrated) or 7 (uncalibrated) lines [1]. At localization, the 3D pose (1-DoF rotation + 2-DoF translation) can be recovered from three previously reconstructed lines. Relevant steps may include:

1. 2D Vertical line detection. This can be done on top of either the existing line detections or the gradient map from existing line detector. Different from general line detection, the developed vertical line detector should be robust to low-light changes.
2. Develop and integrate vertical line SfM initialization and localization from [1].
3. Develop basic track building and integrate into a whole SfM system over vertical lines.
4. Benchmark the results on public or captured datasets.
5. (Optional) develop a good way to detect and deal with degeneracy from coplanar lines (e.g. with homography estimation).



Requirements / Tools:

Python, C++, Basic 3D Geometry

[1] Geppert et al. “Privacy Preserving Structure-from-Motion”, ECCV 2020.

Supervisor:

Marc Pollefeys <marc.pollefeys@inf.ethz.ch>,
Shaohui Liu <shaohui.liu@inf.ethz.ch>
Daniel Barath <dbarath@ethz.ch>

Pose with far-away points + one local

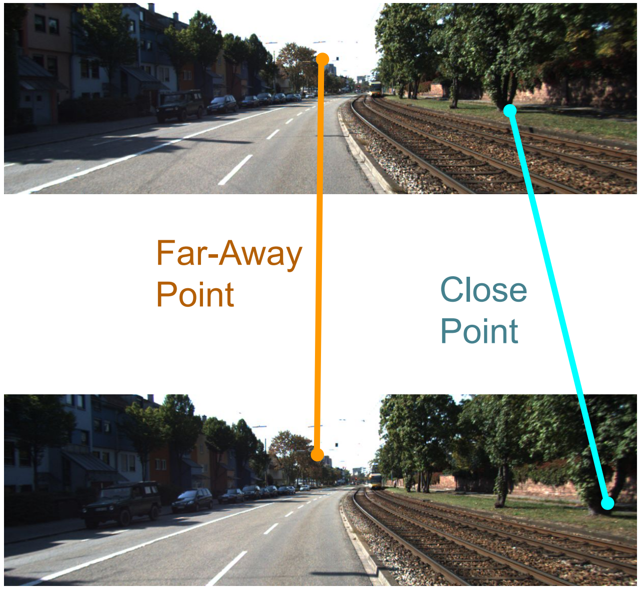
Goal: Develop a scheme for relative pose estimation utilizing far-away points to estimate rotation and near points for translation.

Description:

Far-away points cannot be used to determine translation. However, they determine rotation more accurately than the close points, and they are often easier to match.

Assuming known vertical, then a single far away point correspondence determines the remaining rotation. Without known vertical, two far away point correspondences determine the orientation.

The idea for this project is to first determine the orientation accurately using far-away points, and then perform a 1.5 point RANSAC to determine the translation by sampling explicitly points that are not far away (do not satisfy the homography for far away points).



Requirements / Tools:

Python, C++, Basic 3D Geometry

Supervisor:

Marc Pollefeys <marc.pollefeys@inf.ethz.ch>,
Petr Hruby <petr.hruby@inf.ethz.ch>
Daniel Barath <dbarath@ethz.ch>

Monocular Video Depth Prediction with Diffusion Model

Goal: Developing a method to predict consistent monocular video depth utilizing a diffusion model.

Description:

Recent research [1] demonstrates that directly fine-tuning a text-to-image diffusion model, trained on a large-scale dataset, exhibits remarkable generalizability for the depth prediction task. While this approach yields impressive results for single-image depth prediction, the current state-of-the-art (SOTA) method [1] encounters challenges in producing consistent results when applied to video clips, resulting in flickering outputs. This inconsistency may impede potential applications. Hence, this project aims to develop a method capable of predicting consistent video depth based on the baseline model [1].
Possible steps for this project:

- Detailed understanding of the Marigold paper and pipeline
- In depth analysis of performance on single frames and video clips
- Improve the performance on video clips
- Investigate its application in downstream tasks such as 3D reconstruction

[1] Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C. and Schindler, K., 2023. Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation. Webpage: https://marigoldmonodepth.github.io/

Requirements / Tools:

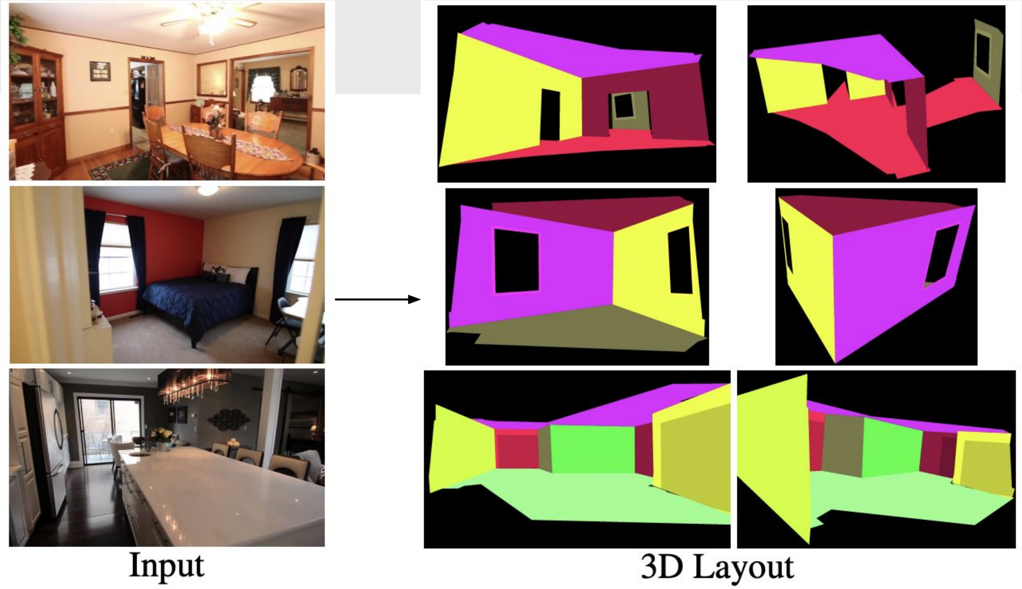
PyTorch, Deep Learning

Supervisor:

Botao Ye (botao.ye@inf.ethz.ch)

3D Room Structures from RGB Videos

Goal: Develop a baseline method to estimate 3D room structures from RGB videos.


Input

3D Layout

Description:

Recently, many datasets have been proposed that contain indoor room, e.g. RealEstate10k [1].

However, most methods to estimate 3D room structures assume inputs with depth maps, panoramic images, etc.

The task of this project is to design a method that is able to estimate 3D room structures or layouts (walls, floor, ceiling) from only RGB videos in the wild. For that, large-scale datasets [2,3] should be leveraged to train a deep learning based approach.

[1] Zhou et al. Stereo magnification: Learning view synthesis using multiplane images. SIGGRAPH 2018

[2] Maninis et al. CAD-Estate: Large-scale CAD Model Annotation in RGB Videos. ICCV 2023

[3] Rozumnyi et al. Estimating Generic 3D Room Structures from 2D Annotations. NeurIPS 2023

Requirements / Tools:

Python, Tensorflow or PyTorch, Deep Learning, Rendering pipelines, Blender

Supervisor:

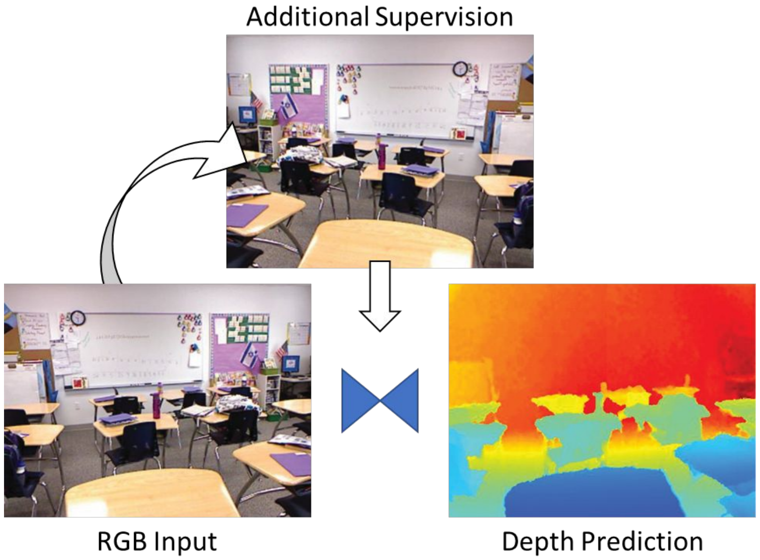
Denys Rozumnyi (denys.rozumnyi@inf.ethz.ch)

Test-time Augmentation for Monocular Depth Estimation

Goal: Providing additional constraints at test-time to improve monocular depth estimation

Description:

Monocular networks benefit from additional training constraints like additional views [Wiles2020, Bauer2021] to improve the depth predictions. In this project, we aim to provide a depth estimation network with additional constraints during training time as well as at test time via data augmentation like image mirroring [Hornauer2022]. While [Hornauer2022] only aim to estimate test-time confidence estimates, we aim to also improve the estimates based on the augmentation consistency constraints.



[Wiles2020] - O. Wiles et al. SynSin: End-to-end view synthesis from a single image. In CVPR, 2020.
[Bauer2021] - Z. Bauer et al. NVS-monodepth: improving monocular depth prediction with novel view synthesis, 3DV 2021.
[Hornauer2022] - J. Hornauer, V. Belagiannis, Gradient-based Uncertainty for Monocular Depth Estimation, ECCV 2022.

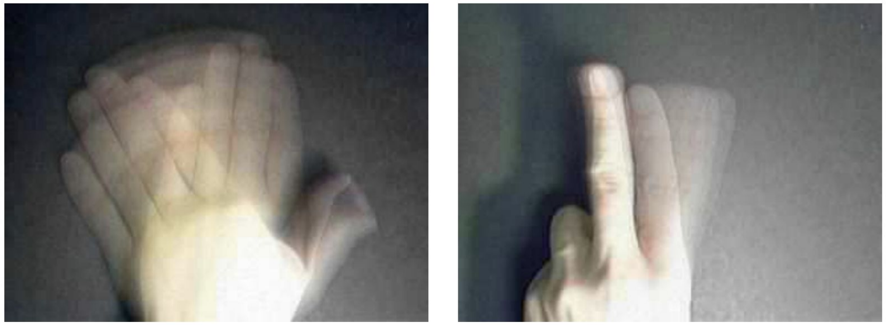
Requirements / Tools:

Python, Deep learning framework (PyTorch or TensorfFlow)

Supervisor:

Zuria Bauer <zbauer@ethz.ch>
Martin Oswald <martin.oswald@inf.ethz.ch>

Hand from Blur



Goal: Estimate a hand model from blurry input images.

Description:

The task is to design a hand pose estimation pipeline that works on blurry inputs [1], even from a single image. Output hand pose will have sub-frame precision, resulting in many poses from one input image [2].

The main idea is to use differentiable rendering with additional priors and learned hand models such as MANO [3]. Such an optimization-driven method was already shown to work well in practice [4].

[1] Sugimura et al.: *“Using Motion Blur to Recognize Hand Gestures in Low-light Scenes”*, VISIGRAPP 2016
[2] Oh et al.: *“Recovering 3D Hand Mesh Sequence from a Single Blurry Image: A New Dataset and Temporal Unfolding”*, CVPR 2023
[3] Taheri et al.: *“GRAB: A Dataset of Whole-Body Human Grasping of Objects”*, ECCV 2020
[4] Zhao et al.: *“Human from Blur: Human Pose Tracking from Blurry Images”*, ICCV 2023

Requirements / Tools:

Python

Supervisor:

Denys Rozumnyi <denys.rozumnyi@inf.ethz.ch>
Taein Kwon <taein.kwon@inf.ethz.ch>

Using Monocular Priors for Gaussian Splatting

Goal: **Use a pre-trained monocular predictor for Gaussian Splatting initialization**

Description:

- Gaussian Splatting [1] is a popular 3D reconstruction approach offering fast high-quality results. However, it relies on having a good input point cloud.
- Monocular depth priors potentially offer a good alternative to initialization as they seem to work well for the SDF representations [3].
- The objective of this project is to use trained monocular depth predictors [2] for Gaussian Splatting initialization.

References:
[1] Kerbl, B., et. al., 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ToG 2023
[2] Shao, S, et. al., NDDepth: Normal-Distance Assisted Monocular Depth Estimation and Completion, ICCV 2023
[3] Yu, Z., et. al. MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction, NeurIPS 2022

Requirements / Tools:

PyTorch / JAX

Supervisor:

Jonas Kulhanek <kulhanek.jonas@inf.ethz.ch>

Efficient underwater 3D reconstruction

Goal: **Enable efficient underwater or foggy 3D reconstruction**

Description:

- Current NeRFs [3] and Gaussian Splatting [2] methods are designed to work well with opaque surfaces but fail when the scene contains lots of semi-transparent content (fog/water).
- The goal of the project is to enable underwater/fog reconstruction [1] by fusing two fields (opaque and semi-transparent). For the semi-transparent field we can use the expected distance between the splatted Gaussians as input to semi-transparent field for faster rendering (instead of approximating the integral over the ray).
- The project will first use synthetic-modelled scenes and later switch to real-world collected data.

References:
[1] Levy, D., et. al. SeaThru-NeRF: Neural Radiance Fields in Scattering Media, CVPR 2023
[2] Kerbl, B., et. al., 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ToG 2023
[3] Barron, J., et. al., Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. ICCV 2023

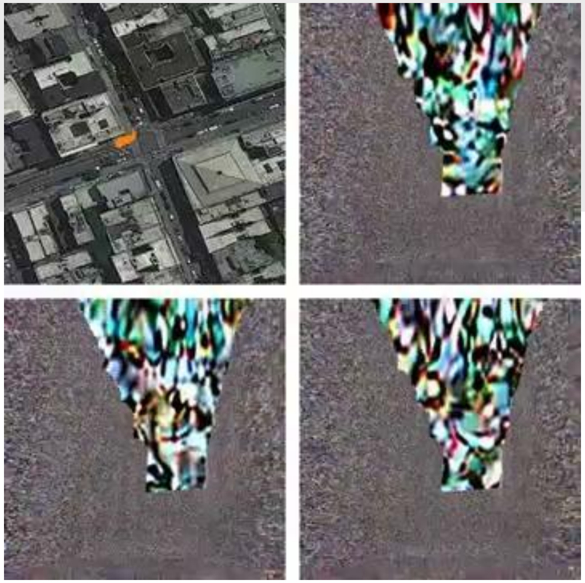
Requirements / Tools:

PyTorch / JAX
C++/CUDA

Supervisor:

Jonas Kulhanek <kulhanek.jonas@inf.ethz.ch>

3D-aware scene generation using diffusion models



Goal: To generate 3D scene representations that can be rendered to arbitrary views which hold very good consistency

Description:

Generalize diffusion models to 3D sparse space and perform scene generation on a given or predicted geometry, followed by neural rendering techniques to render arbitrary views with excellence in both single-frame quality and inter-frame consistency.

References:
[1] Sat2Scene: 3D Urban Scene Generation from Satellite Images with Diffusion (arXiv 2024)
[2] Video generation models as world simulators: https://openai.com/research/video-generation-models-as-world-simulators

Requirements / Tools:

Python, deep learning frameworks (PyTorch or TensorFlow)

Supervisor:

Zuoyue Li (li.zuoyue@inf.ethz.ch)