Lecture 01: Introduction

1 Introduction

Definition 1. ***Computer Vision*** is the automatic extraction of meaningful information from images and videos. Among related disciplines, one can find graphics, artificial intelligence, machine learning, cognitive sciences, robotics and image processing. One can generally define two types of information:

- *Semantic information:* objects are related with their meaning (e.g. car, person, european streets).
- *Geometric information:* e.g. shapes, bounding boxes.

In this course, we will rather focus on this second type of information.

1.1 Vision in Humans

In humans, vision is the most powerful sense. In fact, the retina (approximately 1000 mm^2) contains 130 million photoreceptors (120 million rods and 10 million cones, for colour sampling). This results in an enormous amount of information transmitted every second (approximately 3GB/s), which occupies half of the primate cerebral cortex to perform visual processing. To match the same resolution as the human eye, we would need a 500 Megapixel camera. However, in practice, the acuity of an eye is 8 Megapixel within the 15-degree field of view. But is the amount of data the only problem of computer vision? Computer vision is hard because the visualized objects, the viewpoint variations, the changes of illumination and its motion, are represented through numbers.

Remark. If you are interested in the origin of Computer Vision, check `tadada` (L.G. Roberts, MIT, 1963 , Solids Perception).

1.2 Visual Odometry (VO)

Definition 2. ***Visual Odometry (VO)*** is the process of incrementally estimating the pose of a vehicle by examining the changes that motion induces on the images of its onboard cameras.

1.2.1 Why Visual Odometry?

- Contrary to Wheel Odometry (WO), Visual Odometry is **not affected** by wheel slippage and adverse conditions in general.
- Visual Odometry is much more accurate, i.e. the general relative position error is 0.1-2 %.
- Visual Odometry can be used as a complement to wheel encorders, GPS, IMUs, etc.
- Visual Odometry becomes crucial for flying, walking and underwater swimming.

1.2.2 Assumptions for Visual Odometry

We assume

- **Sufficient illumination**,
- **dominance of static scene** over moving objects,
- **enough texture** to allow apparent motion,
- **sufficient scene overlap** between consecutive frames.

1.2.3 History

Table 1 reports a simplified timeline for the history of computer vision.

1980	First known VO real-time implementation for Mars rovers (Moravec, NASA/JPL).
1988-2000	VO research dominated by NASA/JPL for the mission to Mars (2004).
2004	VO used on Mars: Spirit and Opportunity
2004	VO revived in academic environment (David Nister, Visual *Odometry*).

Table 1: History of Computer Vision.

Remark. Some facts about the first known VO implementation of Moravec:

- The Mars rover had a single camera sliding on a rail horizontally, taking 9 pictures at equidistant intervals.
- Corners were detected with Moravec's algorithm (more later) and normalized using correlation.
- Outliers were removed by checking depth inconsistencies.
- Although only one camera was used, this was a case of stereo vision, becuase of the triangulation between the different positions of the robot.
- A single camera without performing stereo vision has the disadvantage that motion cab be recovered up to a scale factor (more later).

1.3 VO vs VSLAM vs SFM

It holds

$$\text{VO} \subseteq \text{VSLAM} \subseteq \text{SFM}, \tag{1.1}$$

i.e. Visual Odometry (VO) is a particular case of Visual SLAM (VSLAM), which is a particular case of Structure from Motion (SFM).

1.3.1 Structure From Motion (SFM)

Structure Motion is more general than Visual Odometry: it tackles the problem of 3D reconstruction and 6 degrees of freedom (DOF) pose estimation from **unordered image sets** (e.g. reconstruction through flickr pictures). Visual Odometry, instead, focuses on estimating the 3D motion **sequentially** and in **real time**.

1.3.2 Visual Simultaneous Localization And Mapping (VSLAM)

Visual SLAM (Symultaneous Localization and Mapping) focuses on obtaining a globally consistent estimation. Essentialy, VSLAM can be resumed into Visual Odometry, loop detection and graph optimization. Depending on the application, one has to find a tradeoff between performance and consistency of the solution. In this sense, Visual Odometry is extremely useful. In fact, it does not require to keep track of all previous history of the camera and hence allows real-time implementations.

1.3.3 Working Principle:

1. Compute the relative motion T_k from images I_{k-1} to image I_k

$$T_k = \begin{pmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{pmatrix}, \tag{1.2}$$

   where $R_{k,k-1}$ and $t_{k,k-1}$ represent the rotation and the translation matrix from I_{k-1} to I_k.

2. Concatenate them to recover full trajectory

$$C_n = C_{n-1} \cdot T_n. \tag{1.3}$$

3. An optimization over the last m poses can be done to refine locally the trajectory (Pose-Graph or Bundle Adjustment).

Direct Image Alignment

How can we estimate the relative motion T_k? First of all, there is a difference between **direct** and **indirect** methods.

- Indirect methods first extract features and then use them for localization and building of the map.
- Direct methods instead, try to recover the environment depth, the structure and the camera pose through an optimisation on the map and camera parameters together (affected from light changes).

The whole **direct** problem is a **minimization** of the **per-pixel intensity difference**, i.e.

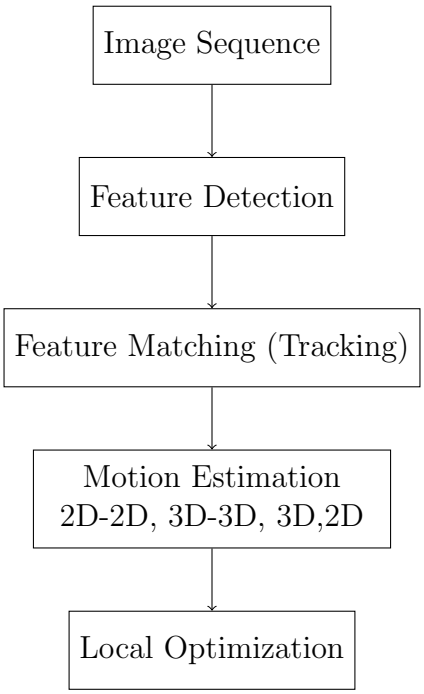
$$T_{k,k-1} = \underset{T}{\operatorname{argmin}} \sum_i \|I_k(u_i') - I_{k-1}(u_i)\|_\sigma^2. \tag{1.4}$$

Depending on the number of used pixels, one can define:

- Dense Methods: $\approx$ 300'000 pixels (all). Require powerful hardware (GPU).
- Semi-Dense Methods: $\approx$ 10'000 pixels (only edges).
- Sparse Methods: $\approx$ 2'000 pixels. (100-200 point features per 4x4 patch). Basically pointclouds, used mainly to do localization (camera pose estimation).

1.3.4 VO Flowchart

VO computes the camera path incremental, pose per pose



1.4 Understanding Check

Are you able to:

- *Provide a definition of Visual Odometry?*
- *Explain the most important differences between VO, VSLAM and SFM?*
- *Describe the needed assumptions for VO?*
- *Explain the working principle of VO? Illustrate its building blocks?*
- *Explain the difference between Dense, Semi-Dense and Sparse Methods?*