Processamento de Imagem e Visão (PIV)

Project description

The project is structured in 2 parts: Part 1 is common to all groups and Part 2 groups may choose between 2 subproblems. Students must develop and submit code for both parts and a report only for part 2.

Part 1 – Image stitching : Given a sequence of images, the goal is to compose the images into one single panoramic image, as illustrated below.

(in https://www.pyimagesearch.com/2018/12/17/image-stitching-with-opencv-and-python/)

This is a current tool in today's devices (smartphones, laptops) and is the basis of mapping apps like google maps/street view. It hinges in basic concepts that you will learn in class:

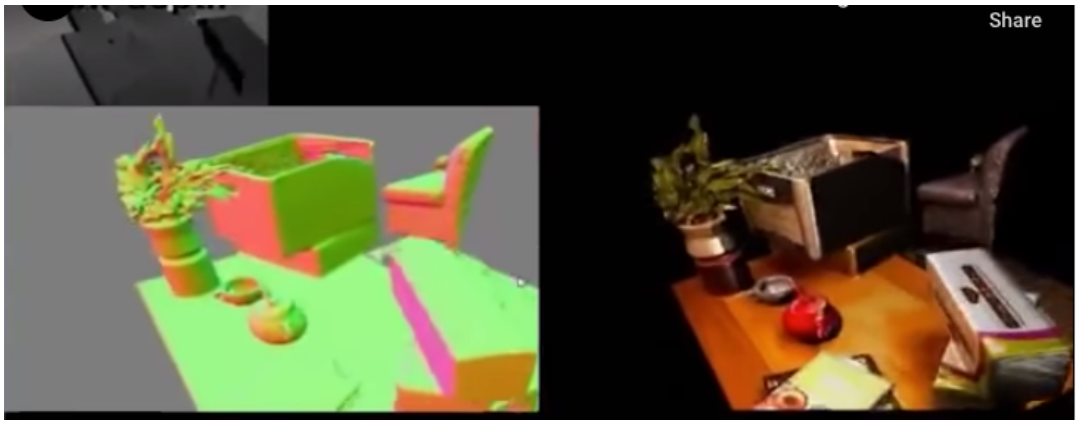
- Camera model (sections 1,2)
- Image transforms (sec 1,2,5)
- Image features (sec 4)
- point matching (sec 4)
- outlier removal (RANSAC sec 4,5)

Students are required to develop code for some tasks (to be defined later) but in general can use any available source (matlab toolboxes, open source code from the internet). This part must be demonstrated by each group in the week after the mid-term test (Nov. 4th).

Part 2 - Choose one : 2.1 – 3D point cloud registration 2.2 – Bowlling with the Kinect

2.1 – 3D point cloud registration

The goal of this project is to create a 3D reconstruction of a scene from a sequence of images from a depth camera (Kinect, Intel Realsense). This type of camera provide two types of images: a “regular” RGB image (right picture below) and a DEPTH image from which a 3D map can be computed (left picture below)



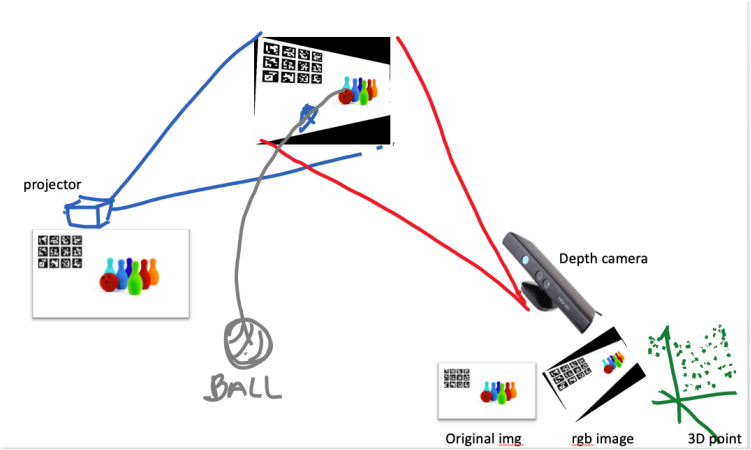
(from Kinect Fusion project)

This project is, fundamentally, an extension of the previous one in the sense that the same steps are required but for a different type of transformations (3D rigid motion). However the problems involved, unknowns and output differ significantly.

Students are required to develop code to implement the fusion of the 3D point clouds and can use code from all available resources. Details and datasets will be provided soon.

2.2 – Bowling with the Kinect

Consider the following setup:



An image is projected in the wall (using a slide projector). A depth camera is placed in an arbitrary position (not kown in advance) and acquires rgb and depth images (3D point clouds).

A ball is thrown against the wall. and the goal of the project is to detect if the ball hits the pins and which.

Notes (for all parts):

- The code is not required to run in real time. We will provide the sequences and your code will run offline without time limitations (within reasonable limits!).

- The decision about which part you will do can be taken after you conclude part 1. In other words, groups must be formed, start working in part 1 and then decide what to do for part2.

- Relevant methods and parts of code will be explained and constructed in the lab. In other words, lab classes will follow a “script” whereby if you pay minimum attention and use the supplied code should not be hard to do part 1 + part 2.1

- Part 2.2 is not more complex then 2.1 but has not been done before. There are several paths, we still need to acquire data and define the details. Since it is more risky the performance requirements are not so demanding (if you miss some pins!) and better rewarded if done right (with the proper techniques or if it works reasonably well).

-Part 2.1 is well established, it’s a project that run in PIV before, will be thoroughly detailed in the lab classes. Projects must run flawlessly, at least for “easy” datasets.

Specific details will be provided soon.