415.773 - 2002 Assignment 1: Interactive Stereo Camera Calibration, Lines Extraction, 3D GUI

Due date: Tuesday, August 27, 2002, 5:00 pm

Project

This year project will allow you to control a car via hand signs recognition. Although it can be described in a few words, it will require a lot of works and achievements in different fields such as 2D (hand recognition) and 3D (objects positioning) Image Processing, Robotics (via Motion planning and AI: Reinforcement learning), Networking (for the car control) and much more.
Basically, for a randomly given field (including at least a bridge and a tunnel) the car will have to cruise from its starting position to the finish line. Full time control by hand gesture may appear to be really slow. We will rather set up check points where a list of actions will be offered (turning, going back, going forward, automatic mode, etc...), hand sign recognition carrying the next action.

The goal of this assignment is:

- To deal with the videoserver, the stereo cameras and their handling
- To perform the calibration of the cameras
- To explore colour detection techniques applied to stereo calibration
- Track the Hummer car, compute its position and velocity.

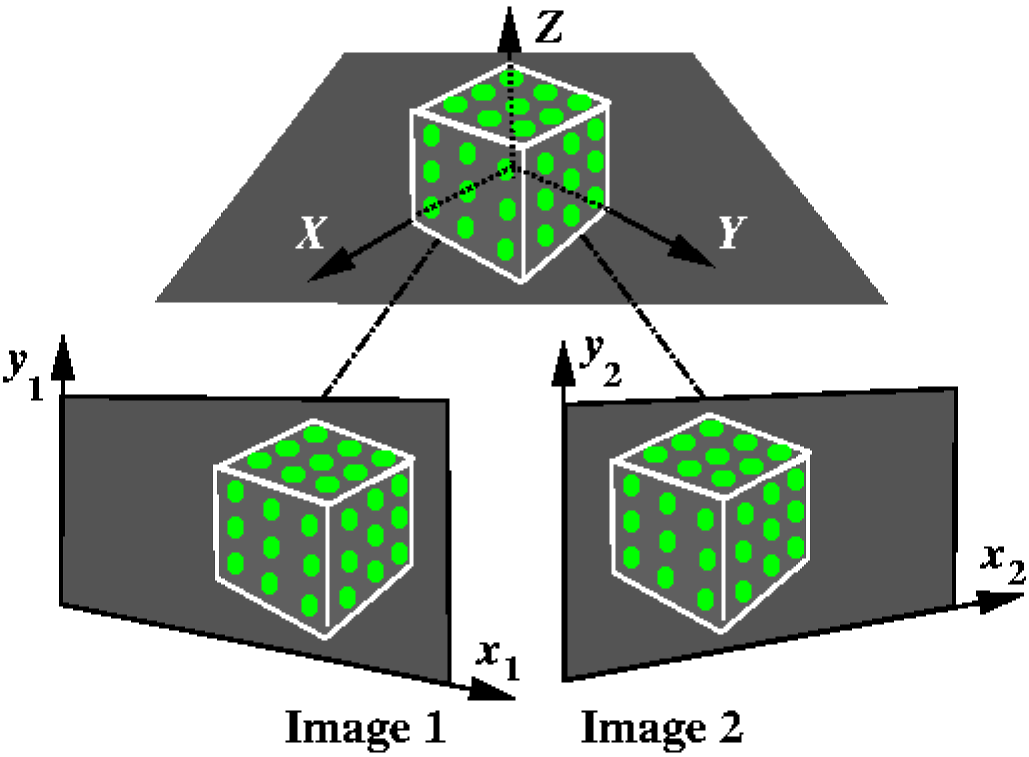
Comments:
For each part, hints and directions are provided. It is up to you (and strongly encouraged) to do some bibliography on each topic and propose a different solution which may solve the problem from a different perspective. Initiatives (when successful) will be rewarded. Still, well documented reports are expected to explain from your own point of view cons and pros of your very favorite approach.

Frame-grabber and calibration of cameras

To be able to control our cars correctly, we must know their current positions and orientations as well as the current position of other objects in the same world 3D coordinate system, or reference frame. In our lab, the two pan-tilt video-cameras form a stereo system which provide the required information. However, each video camera only records pixel coordinates of objects so that we have to link these 2D coordinates to the desired 3D world coordinates.

The problem of mapping pixel coordinates to real world coordinates is called **camera calibration**, or **orientation** in computer vision and analytic photogrammetry. Stereo calibration includes the body of techniques by which, from measurements of two 2D-perspective projections of a 3D object, we can make inferences about the 3D position, orientation, and length of the observed 3D object in a world reference frame. If a particular parametric computational model of the camera is specified, an algorithm can be developed to estimate these parameters of the cameras, given the image coordinates and 3D world coordinates of same observed 3D points.
The simplest camera model is a pin-hole camera that forms the precise central projection of an observed 3D optical surface. In robotics, the pin-hole model that takes account of additional non-linear image distortions due to non-ideal lenses has been developed by Tsai and is usually referred to as the Tsai camera model.

Part 1: Stereo calibration



Your first task is to use the 3D calibration cube (**you may design a new calibration object**) to compute the intrinsic and extrinsic parameters of the calibration matrix for each camera. You will write an interactive program that collects data from your calibration object, the image plane, perform the calibration and output the calibration matrices and parameters. You must also evaluate the accuracy of the calibration.

To use the binary machine vision techniques for image processing, the calibration cube should be placed onto a plain background so that the desired disks can be separated from background by thresholding. During the calibration, both cameras must mostly see the same faces of the calibration cube.
Position of the disk centers used for the calibration process should be mainly determined automatically. You could, for example, click on one disk (using the videoserver) to specify the desired colour boundaries for the binarisation process. The other disks positions and centers should be determined automatically with some interactive process to correct the wrong matches.

The interactive program has to do the following operations.

- Input and visualize the two images of the calibration object forming a stereo pair.
- Compute points in the two images that correspond to the centers of all visible disks of the calibration pattern.
- Visualize the stereo pair with the overlaid corresponding points.
  - Correct interactively, if necessary, the wrong matches.
- Output the coordinates of the corresponding 3D and 2D points in the format needed by the available calibration software.
- Evaluate the accuracy of the calibration data by comparing the true and reconstructed 3D coordinates of the disks in the calibration pattern.

You should use the cameras to take some stereo pairs of the calibration object and store them as they may be used again to develop and test your program. The input image files should be in the **ppm** format. You can use **fgrab** application software to take the images.

Matching the Points

The true 3D coordinates of the calibration points (centers of disks) should be input as a simple list of X,Y,Z triples in the ASCII text format. The triples should be listed in such an order that allows to easily determine the correspondence between the 3D points and 2D image points.

You have the following two options.

1. *Simplified option*: to interactively determine the correspondence between a current 3D point and its positions in both images. You should only point out the corresponding disks in the images, and your program has to compute the 2D coordinates of their centers (the average *x*- and *y-coordinates* of all the pixels of a disk) and store the resulting seven input and output coordinates ($X,Y,Z,x1, y1,x2,y2$) in the same order as the input 3D coordinates.

1. The disks should be detected by segmenting the binary image obtained by thresholding the initial greyscale one. The found blobs should be ordered by their relative positions to establish one--to--one correspondence between the blobs and 3D disks.

The output should be a simple list of the corresponding 3D world coordinates and 2D coordinates in the images in the ASCII text format:

Number	X	Y	Z	x_1	y_1	x_2	y_2
1	*15.0*	*8.0*	*10.0*	*125.4*	*75.4*	*150.0*	*36.8*
2	*10.0*	*5.0*	*10.0*	*105.4*	*52.5*	*175.5*	*49.6*
...	...	...	...	...	...	...	...

Also, you need to output two files that contain 5-tuples of the same 2D and 3D coordinates in the ASCII text format but separately for each camera:

$$X, Y, Z, x_1, y_1$$

for the first camera and

$$X, Y, Z, x_2, y_2$$

for the second camera. These two files are needed by the calibration software for the Tsai camera model to do the calibration.

Calibration

The output of calibration should be a simple list of the interior and exterior orientation parameters in the ASCII text format for each camera.

Evaluation of Accuracy

For the computed orientation parameters, determine the accuracy of these parameters with respect to the calibration cube. An error is given by the difference between the true 3D coordinate and the coordinate computed using the image coordinates and the calibration parameters. You should use the least square technique to find the 3D intersection between two projective rays from each pair of the corresponding pixels in the images because the rays may not exactly intersect in the 3D space due to imprecise camera parameters.

You should compute the average and maximum absolute error for the 3D positions of the centers of the disks in the calibration cube with respect to the same positions reconstructed by intersecting the corresponding projective rays.

Your report should include theoretical aspects, experimentation and conclusions.

Part 2: Design of a Tunnel, line detection

Design

To make full use of the stereo vision system, you need to create obstacles which will render the car path planning dependent on accurate 3D positioning. A bridge designed by last year students is already available. this year you will have to design a tunnel. It should be large enough to allow car-like robots movements. You may also create a few thin vertical objects which will be used as check points by the car. These objects may consist on a flat round base, a thin pod and a top round base which may include a color patch as the videoserver already includes color patches center extraction. Therefore, several patches cleverly positioned may provide information on 3D geometry of an object. Objects such as the tunnel may be localized through line detection. The stereo field is also delimited by red tapes vertical and horizontal lines.

Line detection
Extract lines from the field and the different objects designed with lines to automatically compute their positions. Hough transform may be a good tool for that task Line detection should be performed once before the start of the car control sequence. Lines extracted should be displayed in the 3D reconstructed view of the field which you will have to create for part 4.

Part 3: Tracking the car

Your aim is to detect and track the car on a predefine track by computing its position and velocity. The car should go on the four corners of the field, pass the bridge, go through the tunnel and the different obstacles (colored plastic glasses or anything else you may design) when you control it manually. Your program should display the 3D position of the car and its velocity at every moment during this journey. The videoserver gives the real world 3D coordinates of an object by intersecting one ray with a plane with already known Z coordinates (the field plane). You will have to:

- use the videoserver with both cameras to obtain rays pointing to the desired object (the car).
  - You should use the config file to specify the color and shape of the object you wish to follow.
- Intersect these 2 rays to find the 3D position of the desired object.

Evaluation
Your interface will display the position and the velocity of the car at any moment. You should be able to display a graph of the path followed by the car from one robot to the other one and print out 3D coordinates of characteristic points specifying this path.

Submission

You should submit your assignment as a printed report (one per group) before the due date (it may be handed in either after the lecture or in the CITR Robotics Lab).

Title page of each report should have names and signatures of all the members of the corresponding group. The report itself should include the following parts.

Part 4: The GUI

Your interface will display on line the 3D position and the velocity of the car at any moment. You should be able to display a graph
of the path followed by the car from the moment it started. You should also display a 3D window where the positions of the car, obstacles, tunnel and bridge are available during your experiment.

The interface may "control" all the different parts of this assignment:

- Clicking one button should perform the calibration.
- Clicking another one should detect limits of the field and 3D position of the objects.
- As soon as the car is moving it should update its position, direction and velocity.

Forthcoming events:

In the next assignment (assignment 2) the GUI will include car control (through a set of buttons controlling forward, backward, turning control) and display the recognized hand signs. In the final assignment it should link hand signs recognition to car control. It should also display the minimum (motion planning) path for a given position of the different obstacles...

Minimum achievement:

- 3D calibration with reasonable accuracy
- Object detection with color patches
- GUI with 3D positioning of the car

Final report

Your report MUST include individual report (one per student) and group report. Group report may deal with introduction, description of your overall achievements in regard with the proposed work and conclusion. Each group member should write an individual report regarding his/her contribution to the project. **Each** group member will have a short talk with me or Georgy before the end of the 773 paper to assess his/her contribution. Still your report should be a scientific contribution rather than a daily diary account of your 773 journey. Front page should clearly indicate who did what.
Report structure may include:

- Description of algorithms used for each parts.
- Description of your programs.
- Description of your experiments.
- Conclusions and/or comments.