Projects

within the framework of the course:

image processing & computer vision

121091

date: 9 March 2016

F. van der Heijden

mailto:F.vanderHeijden@utwente.nl

Introduction

The project is the concluding part of the course in which students show their academic competence and skills with respect to image processing and computer vision.

Groups

A project is done with groups of two persons. A service will be offered for coupling students who have difficulties to find a partner. Working individually is allowed, but cannot be used as an excuse for delivering half of the work.

Subjects

	Title	Possible ingredients
1	Virtual advertising	• line/corner detection • vanishing point detection • auto camera calibration • perspective projection
2	Man over board	• camera calibration • image stabilization • tracking and prediction • point detection/localization
3	Augmented reality: placement of sEMG electrodes	• image registration • face tracking • real-time video processing
4	Visual navigation	• camera calibration • key point detection • robust key point matching • Hough transform • visual odometry • land mark detection • visual navigation – dead reckoning
5	3D Motion estimation of wings of a Robotic Bird: stereoscopic marker tracking in high-speed video	• camera calibration • point detection • point tracking • 3D reconstruction • tracking and prediction

Deadline report:

April 17, 2016

Repository of resources

Resources (templates for report, data, and additional software) will be made available via:

http://1drv.ms/17zJnTs

Report

- Use one of the templates (word, latex) provided in the repository is **mandatory**.
- Final format: pdf. (**don't** send word documents).
- Maximum 8 pages (without the references and the annexes)
- Describe your algorithm such that the principle of operation becomes clear. For that, use 'mathematical style' pseudo code. See: https://en.wikipedia.org/wiki/Pseudocode and pseudo-code examples provided in

https://en.wikipedia.org/wiki/Category:Articles_with_example_pseudocode. The report itself does **not** contain Matlab code (you should add your listing as an annex).

Structure of the report:

1. Intro:
   - Context
   - Statement of the problem, perhaps with research question and subquestions.
2. Methods and Materials
   - Materials used
   - Analysis of the problem and the strategy for the solution (i.e. the explanation of the algorithm)
   - Description of the experimental set-up including the evaluation method
3. Results
   - Here, you give the results of what is described in Section 2: tables, images and/or graphs.
   - The accompanying text of these tables, images and/or graphs clarifies how they are related to the methods described in Section 2.
   - If you have any remarkable observations, mention and describe them here, but without an interpretation, explanation, or meaning.
   - Do not introduce new methods in this section.
   - Do not give an interpretation or a judgement of these results.
4. Discussion/Conclusion
   - Give an interpretation and judgement of the results.
   - If you had any remarkable observations, discuss them here to give them meaning (interpretation, explanation, implication).
   - Describe limitations of the study.
   - If applicable, compare your results with results from literature.
   - Conclusion: describe the overall implication of the results to the original problem statement (or research questions).

   Literature list
   Matlab listings

Supervision

- on demand: max 1 times

Grading

After the report is finished, you make an appointment for a discussion with F. van der Heijden via a doodle: . The intent of the discussion is to evaluate the student's understanding of the subject. As a consequence, questions will appear that are not directly considered in the practical work, i.e. topics addressed in the lectures and the exercise. The final mark is given directly after the discussion. The grading is based on the reports of the short exercises and the project, and on the student's convincingness during the discussion. As such, the final mark is the result of the overall impression of the teacher.

Criteria on which the project and the report are graded are derived from the general criteria for academic competence and skills. This includes one or more of the following aspects:

- A critical attitude
- Being able to find relative literature
- Being able to read a scientific paper
- Being able to design systematically:
  - Defining the functionality of a system by dividing the system into interacting subfunctions.
  - Decisions during the design are made on explicitly defined criteria.
  - Being able to analyse a (sub)function.
  - Being able to experimentally evaluate the design.

Project 1: Virtual advertising

Many sport events are broadcast on television with an overlay providing an extra advertising space. An example is shown in the adjoining image which shows advertising for ‘AFAB’ as an overly next to the goal in a football match.

The placement and projection of such a virtual advertisement is based on a couple of aspects:

1. The 3D localization of the side lines of the game with respect to the arbitrarily chosen world coordinate system.
2. The position and orientation of the camera with respect to this world coordinate system.
3. The intrinsic parameters of the camera.

There is no specific calibration object in the scene. Instead, the court lines of the sport, and its known lengths, should be used. In computer vision, this falls under the umbrella of auto-calibration (self-calibration). See, for instance, [1].

The assignment

The problem statement is how to find and to test a suitable method to accomplish virtual advertising. Subtasks are:

- Select 4 videos clips of a play in which the (orthogonal) court lines (field lines) are sufficiently visible during the panning and rotation (possibly, but more challenging, also zooming) of the camera.
- In the first frame of each clip: locate some of these court lines.
- In the next frames, track (the parameters of) these lines.
- Use (the parameters of) the detected lines to calibrate the camera. The extrinsic parameters describe the panning and rotation of the camera and may vary from frame to frame. The intrinsic parameters of the camera are constant except if the camera is also zooming. You are allowed to neglect nonlinear lens distortion.
- Using the calibration parameters: project a virtual banner which would have a rectangular shape in the real world in the image near a court line.
- Finally, create demo movies of the clips which show the found court lines and the banner.

You are allowed to adapt your line finding algorithm to the type of sport and the colours that have been used. For instance, a football play might need a different strategy than the volleyball play. Make sure that at least two parallel lines are visible in two orthogonal directions (so, at least 2 times 2 lines should be visible).

Grading:

This project has different levels of challenge. The dependence of the grade on the implemented level is:

a) max 6 — The lines are detected correctly, and a banner is placed horizontally flattened on the surface of the field.

b) max 7 — The lines are detected correctly, and a banner is placed near a court line and vertically aligned with respect to the surface of the field. One of the vanishing points of the projected banner is correct.

c) max 8 — As in b), but also auto calibration has been correctly implemented. The principal point is excluded from the calibration by assuming that it is located in the middle of the image. The ratio between the width and height of the banner in 3D is correctly represented in the projected banner, and both vanishing points of the projected banner are correct.

d) max 9 — As in c), but now the principal point is also included in the calibration.

e) max 10 — As in d), but now the banner is placed at an angle of 60 degrees w.r.t. the field. A shadow of the banner is also projected on the field as in the figure.

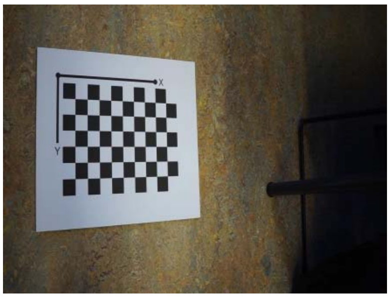
Other factors, e.g., the quality of the report, the quality of the result (stability), also count.

[1] Z. Zhang, "Camera Calibration," in *Emerging Topics in Computer Vision*, G. Medioni and S. B. Kang, Eds., ed: Prentice Hall Professional Technical Reference, 2004, pp. 4-43.

Project 2: Man over board

Finding someone lost at sea is a difficult task. Even when a "man over board" situation is timely detected, and the person is spotted nearby a vessel, keeping track of that person is difficult.

In this project, the possibility of tracking such a person in a wavy sea must be examined. Videos are available of a test buoy that is a stand-in for someone at sea. The first task of analysing such a video would be the detection of the object. Once detected and localized in the video, the next task is to keep track of the object so that a rescue vessel can approach it. The third task would be to estimate the distance of the test buoy to the camera, which would be useful information of a rescue operation in a real situation.



Assignment: This project focuses on the second and the third task. Given the position of the test buoy in the first frame, track the buoy in consecutive frames, and estimate its distance in meters.

Hints:
For tracking the object, different choices must be made.

- As a pre-processing step, consider stabilizing the images first. There are camera rotations along three orthogonal axes, i.e. tilting (rotations around the optical axis), panning (rotations around the vertical axis), and pitching (rotations up and down). To compensate, you need reference points/lines in the images. For which of these rotations is such a reference available? In which order of rotations do you apply these compensations? Note: the videos are not acquired at open sea. So, objects near the horizon are visible. But in a real situation, we may assume that the horizon is still visible, but we cannot assume that we have locatable objects there. Thus, the preferred solution does not use the row of trees at the horizon.
- Having estimated the position of the object in one frame, the localization of the object in the next frame starts with defining a search area, i.e. a region of interest (ROI), based on the previous estimate. What shape, size and position should this search area have? This becomes especially important since in some frames the object is not always visible due to the heavy waves. In these cases, we should move on to following frames until the object is detected again. To do that, you need motion models.
- There are three different sources of motion in the image data: the camera motion, the travelling see waves, and the motion of the buoy. We may safely assume that the buoy doesn't move fast. Relative to the see waves, it is almost static. The motion of the camera consists of rotations of which the effect is (perhaps partly) compensated by the pre-processing step. There is also a linear motion, e.g. up and down, but at a larger distance the effect of that in the image is similar to rotations. The motion of the travelling sea waves is more complex due to the perspective projection. Waves nearby seems to travel much faster than waves at a larger distance. However, at a larger distance, the optical flow within a small ROI, can be modelled as constant. The effect of rotations of the camera within this ROI can also be approximated as a uniform optical flow field. You can perhaps measure this optical flow field vector by means of a key point detection and matching, or otherwise by template matching using this blockwise motion model.
- A ROI holds travelling waves together with the buoy. The latter only if the buoy is visible. How do we detect whether the buoy is visible? Knowing the uniform optical flow of the sea waves, you might be able to apply motion compensation and image subtraction to nullify the waves.

For the estimation of the distance, a geometrical model should be applied incorporating:

- The average height of the camera above sea level. This height is given: 2.5 m.
- A spherical model of the earth surface, or an approximation of that.

- A perspective projection model of the camera. The intrinsic parameters of the camera should be obtained from calibration. For that, images of a chessboard are available. See the example. The size of a square is 50 x 50 mm^2. These images are suitable for usage in the calibration app of matlab.

Grading: the grade depends on the quality (robustness, accuracy, etc) and elegance of the solution, the thoroughness of the evaluation, and the quality of the report.

Project 3: Augmented reality: placement of EMG electrodes

In cooperation with the Department of Head and Neck Surgery
Antoni van Leeuwenhoek Hospital
Amsterdam

The Netherlands Cancer Institute, together with the University of Twente, performs research on modelling and simulating the functional tasks of the oral and oropharyngeal cavity, i.e. speech, swallowing and mastication. See www.virtualtherapy.nl. One of the projects is the development of an EMG based prediction model of the lip motions. For that purpose, EMG electrodes are attached to the facial muscles of a subject, and concurrently EMGs and videos are recorded. In a second session, about two weeks after the first session, this procedure is repeated to test the predictability. During this second session, the electrodes should be placed exactly on the same location as during the first session. This is difficult to achieve.

The assignment – During the attachment of the electrodes live video is available. This offers the opportunity to show the laboratory assistant a live image of the subject with an overlay of the correct positions of the electrodes. This will provide him the feedback to place the electrodes at the correction place. The assignment is to develop methods:

- To find the positions of the electrodes in an image that is taken from the first session. This part is done only once, so there is no strict need to do this fully automated.
- To track the face in the live stream in the second session, so that the image of the first session can be easily registered to the live images. Initialization of this tracking can also be done with some manual assistance.
- To overlay the live images with markers of the electrodes from the first session.

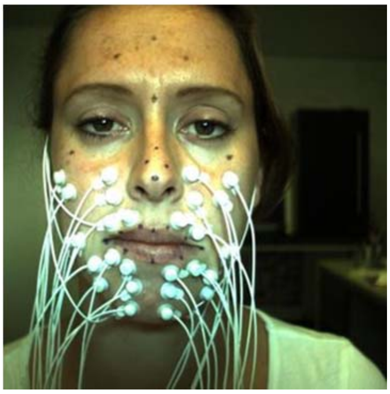


Figure 1. Electrode placement in the 1st session (day 1)

For development of the software, and for testing, movies are available that are acquired just after the first session on day1 and during electrode placement on day 14.

Grading:

a) max 7 Movies are provided of the electrode placement on day 14 with the overlay of the markers showing the locations where the electrodes should have been placed.
b) max 8 The results of a) have been quantitatively evaluated.
c) max 9 As in b), but also a turnkey Matlab application is provided. Not necessarily realtime (25 fps), but with the possibility to have a live input video stream[1]. You may want to create a GUI, but this is not mandatory. You can test and demonstrate your design using skin markers instead of the electrodes.
d) max 10 As in c), but also a realtime application is provided, not necessarily in Matlab. Other platforms (Java, C++, C#, Python) are also allowed, but the implementation should run on a Windows 64bit platform.

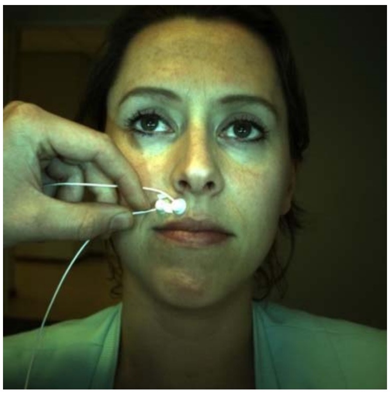


Figure 2. Electrode placement for the 2nd session (day 14)

The grade also depends on the quality (robustness, accuracy, etc) and elegance of the solution, and the quality of the report.

[1] Perhaps you want to use Matlab's image acquisition toolbox. You can test your software with a webcam.

Project 4: Visual navigation

Two frames from the "ceiling" movie

In mobile robotics, a well-known paradigm is the so-called SLAM, simultaneous localization and mapping. In other words, exploring an uncharted environment resulting in a map, and at the same time locating yourself in that map. Often, the input of SLAM consists of camera images possibly augmented by complementary sensory data delivered for instance by accelerometers, gyroscopes, and magnetic field sensors.

In general, SLAM needs two types of sensory data. The first type of data allows the estimation of the vehicle's own pose (= position and orientation). As an example, an accelerometer and a gyroscope provide the acceleration and the turn rate of the vehicle. By integration, the displacement (linear and angular) can be estimate. By accumulating these displacements, the pose of the vehicle can be predicted. This is called *dead reckoning*.

The second type of data relates the position of distinct points in the environment relative to the vehicle's pose. Such a distinct point is called a landmark, and finding its position relative to the vehicle's coordinate system is called: *landmark detection*. A SLAM algorithm combines dead reckoning with the landmark detection to build the map, i.e. to place the landmarks in a world coordinate system, and to find its pose within this coordinate system.

Camera images can be used for both purposes: dead reckoning and landmark detection. Finding the needed information for dead reckoning is called *visual odometry*. Here, the displacement (a translation vector and a rotation angle) is estimated from two consecutive frames. By accumulative summations of this displacements the trajectory of the pose can be prediction, and the dead reckoning is accomplished. Landmark detection is accomplished by the following tasks: a) detection of landmarks that are visible in the current frame; b) localization of these landmarks, c) data association: checking to see if a detected landmark is never seen before, and if not, associating this landmark with one from the list of previously seen landmarks.

The assignment: Develop the image processing needed for the input of 2D SLAM. That is, given is a video taken from a moving vehicle that is equipped with a camera. See figure. The camera is looking upwards to the ceiling of a building. The movement of the vehicle is 2D. That is, it consists of translations and rotations in just one 2D plane. That plane is parallel to the image plane of the camera. The ceiling is populated with balloons that are now and then visible. The path of the vehicle is approximately a rectangle that traversed two times. Calibration images are available for calibrating the camera. The assignment is as follows: a) Perform the dead reckoning based on visual odometry, b) Implement the landmark detection, the data association included. c) Evaluate your algorithms. The tasks are as follows:

1. Perform camera calibration
2. Develop an algorithm to find the Euclidean transforms (rotation and translation) between two consecutive frames. The video has a frame rate of 25 fps, but you may consider to skip frames.
3. Use the output of 1. and 2. to perform dead reckoning.
4. Develop an algorithm to detect the balloons.
5. Extend this algorithm with data association
6. Create a map showing the predicted position trajectory with an indication of the orientation; the latter, for instance, by means of arrows. Include the landmarks in this map.
7. Performance evaluation.

Note: we are not asking to perform the actual SLAM[2], just the input that is needed for SLAM suffices.

Grading:

a)	max 7	Dead reckoning is successfully implemented: a map has been constructed showing the predicted trajectory.
b)	max 8	The detection and localization of balloons is done successfully.
c)	max 9	The data association of the landmarks is done successfully.
d)	max 10	The results are in some way quantitatively evaluated.

[2] If you are planning to follow the course 'Optimal Estimation in Dynamic Systems': you can continue this project by building the SLAM algorithm

Project 5: 3D Motion estimation of wings of a Robotic Bird

G.A. Folkertsma
Robotics and Mechatronics group
EEMCS-Utwente

Figure 1a The Robird peregrine falcon model.

The “Robird” is a robotic bird capable of flying by flapping its wings, at speeds comparable to the peregrine falcon after which it is modelled (Figure 1a). Both the structural and aerodynamics are rather complicated: it is not fully understood how the Robird can fly, even though it does.

In attempt to get more insight in the wing behaviour, a single wing of the Robird has been placed in the UT’s wind tunnel and in a vacuum chamber at ESA (Figure 1b). By means of a stroboscope, two GoPro cameras and reflective markers, high-speed (or slow-motion?) stereoscopic videos that show the motion and deformation of the wing have been made. (See this shared folder for two example videos.)

Figure 1b The wing in ESA’s vacuumchamber

The assignment

Given:

- A set of stereoscopic, synchronised video registrations of the moving wing,
- A set of calibration images of the stereo camera setup, and
- Physical dimensions of the wing,

The assignment is to find the 3-D trajectories of the 43 markers on the wing. The first task will be to find the reflective markers in the first frame of the video, and to associate markers in the left image to markers in the right image. Next, these markers must be tracked in consecutive frames. From each pair of corresponging markers the 3D position must be derived.

Tracking points implies that based on the position of a marker in the current frame, a prediction must be made of its position in the next frame. Tracking then proceeds by searching in a region surrounding this predicted position. This project involves some design decisions to make:

- Will the prediction be done in the image plane, or in the 3D space?
- What motion model will be applied to do the prediction?
- How large should the search region be?
- Will the prediction be done on each marker individually, or will the cloud of markers be treated as one entity?
- Is image rectification useful in this application, or could it be done without?

For more information, contact: g.a.folkertsma@utwente.nl