Intelligent Systems Group
Universität Duisburg-Essen
Prof. Dr. Josef Pauli / Stefan Lörcks

WS 18/19
2017-12-07

Exercises Computer/Robot Vision - 08

PROGRAMMING (7 points)

In this exercise sheet we will implement the estimation of the time-to-collision.

18. **Time to collision** (7 points)
In this exercise the approach to estimate the time-to-collision, which has been introduced in the lecture, will be implemented. Three image sequences are provided. A robot was driving towards a solid object with constant velocity. This situation can be modeled the same way as the situation explained in the lecture. There, a fixed camera and a object moving towards the camera has been assumed. Each image sequence consists of 100 images. The images are taken a discrete point in time. The points are not equidistant, though. This means the time in between the pictures varies. Therefore, a vector containing the points in time, when the images of the sequence have been taken, is also provided.

    a) Download the zip-file 'sequences.zip' and unzip it.

    b) Create a script named

    `CRV_18.m`

    The first eight lines should look like:

```
%% CRV_18_TimeToCollision
% name: John Doe
% student number: 11235813

%% clean up
clear all;
close all;
clc;
```

    Of course again, fill in your name and student number!

    c) Start with the first image sequence. Assume it is in a subfolder '1'.

        i. Load the workspace file

        `WS_time.mat`

        It contains one column vector with the points in time in milliseconds when the images of the sequence have been taken.

ii. We will extract the height of the object in every image. Therefore, create a vector to save the height of the object in every image.

iii. For every image in the image sequence:

  A. Load the image, convert it to gray-scale and extract the edges.
  Useful commands: `imread`, `int2str`, `rgb2gray`, `edge`

  B. Determine the height of the object the robot approaches using the hough transform. Apply the hough transform for straight lines. Make sure to set a small range of possible angles. Find the highest peak in the hough image and extract the finite line segment. If at least one line has been found, use the endpoints of the longest line to determine the height of the object.
  Useful commands: `hough`, `houghpeaks`, `houghlines`, `isempty`
  Hint: For choosing suitable parameters (especially FillGap and MinLength), it is a good idea to visualize the extracted line segment.

  C. After the height of the object has been extracted in every image use the approach introduced in the lecture to estimate the time-to-collision for each image.

  D. Estimate the time of collision (NOT time-to-collision). This is the actual point in time (in milliseconds), when the collision takes place. Use the estimations of the time-to-collision for the different images to come up with the estimation for the time of collision. Display/print your estimation in the command window. For the first sequence this should be roughly 24 s.

  E. OPTIONAL: Which images of the sequence are useless? How can they be disregarded?

d) Repeat c) for the second and the third image sequence! Estimate the time of collision for both sequences!