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

WS 18/19
2018-12-14

Exercises Computer/Robot Vision - 09

PROGRAMMING (7 points)

In this exercise sheet we will implement the optical flow algorithm, which has been introduced in the lecture. This will be done by four steps: Learning how to visualize a vector field in MATLAB, writing the actual optical flow function, writing a function to visualize the optical flow and finally writing a script to apply the optical flow on three image sequences.

19. **Visualizing a vector field** (1 points)
In this exercise we will learn how to visualize a vector field in MATLAB. Therefore, we will define very simple (constant) vectors on a small set of points and visualize these vectors. We will also divide a image into square local areas and visualize a vector in the middle of each area. This will be a good preparation for visualizing optical flow.

    a) Read the documentation of the function `quiver` [1]. Have a look at the quiver properties AutoScale and AutoScaleFactor as well [2]

    b) Create a script named

    `CRV_19.m`

    The first eight lines should look like:

```
%% CRV_19_QuiverPlot
% name: John Doe
% student number: 11235813

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

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

    c) Visualize the vector field

$$f_1 : \Omega_1 \to \mathbb{R}^2, (x, y) \mapsto f_1(x, y) := (0, 0.5)$$

    where

$$\Omega_1 := \{1, 2, \ldots, 10\}^2 := \{(1, 1), (1, 2), \ldots, (1, 10), (2, 1), \ldots, (10, 10)\}.$$

Useful commands: `meshgrid`, `quiver`

d) Load the image 'cameraman.tif'. Divide the image in square areas of 15×15 pixels. In the middle of each square define the vector $(0.5, 0.5)$, if the middle point is part of the image. Visualize this vector field $f_2 : \Omega_2 \to \mathbb{R}^2, (x, y) \mapsto f_2(x, y) := (0.5, 0.5)$ where

$$\Omega_2 := \{8, 23, \dots, 248\}^2 := \{(8, 8), (8, 23), \dots, (8, 248), (23, 8), \dots, (248, 248)\}.$$

on top of the image.

e) Combine both visualizations in one figure and save the figure as an image named 'vectorfields.png'. The result should look similiar like this:

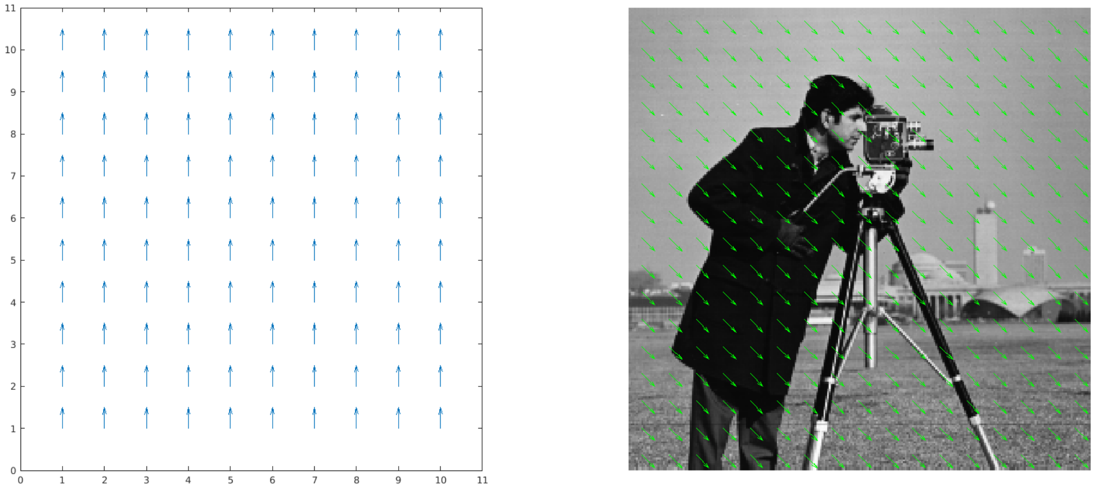

Figure 1: Visualization of the vectorfields f_1 and f_2.

20. **MyOpticalFlow** (3 points)
In this exercise we will write a function that calculates the optical flow.

a) Create a function file named

`MyOpticalFlow.m`

The header should look like:

```
function [ x, y, vx, vy ] = MyOpticalFlow( Ic, Il, N )
%MYOPTICALFLOW calculates the optical flow
%   [ x, y, vx, vy ] = MyOpticalFlow( Ic, Il, N )
%   x, y are row vector contain the coordinates of the domain of the flow
%   vector field
%   vx, vy are matrices containing the flow vector components
%   The inputs are the current grayscale image Ic, the last grayscale
%   image Il and the side length N of the local area Q.
```

This file can be found in the moodle course.

b) Calculate the vectors x and y containing the coordinates of the middle points of the local areas. They depend on the size of the image and the local area size. Divide the image in as many local areas as possible. This means the last few rows and columns might not be considered.
Useful commands: `size`, `floor`

c) Initialize the matrices vx and vy with zeros.
Useful commands: `zeros`, `length`

d) Calculate the spatial gradients of the current image using the Prewitt operator.
Useful commands: `imgradientxy`

e) Estimate the temporal gradient by subtracting the last image from the current one.

f) For every local area:

   i. Create the matrix A containing the spatial gradients of the current local area Q.
   Useful commands: `reshape`

   ii. Create the vector h containing the temporal gradients of the current local area Q.
   Useful commands: `reshape`

   iii. Determine the flow vector v using the pseudoinverse and save its components in the according positions of the matrices vx and vy.
   Useful commands: `pinv`

21. **MyOpticalFlowVisualization** (1 points)
In this exercise we will write a function that visualizes the optical flow on top of the image and saves the result.

a) Create a function file named

`MyOpticalFlowVisualization.m`

The header should look like:

```
function [ ] = MyOpticalFlowVisualization( I, x, y, vx, vy, sF, fN)
%MYOPTICALFLOWVISUALIZATION saves visualization of optical flow
%   MyOpticalFlowVisualization( I, x, y, vx, vy, fN, sF) shows the image I
%   and visualizes the optical flow with components vx and vy and the
%   coordinates x and y. The plot of the flow vectors is auto scaled with
%   the scale factor sF. The resulting figure is saved as a png image file
%   named fN.
```

This file can be found in the moodle course.

b) Make the function create a figure and show the image.

c) Plot the flow vectors using the scaling factor sF.
Useful commands: `meshgrid`, `quiver`

d) Save the figure using the filename fN and close it.
Useful commands: `print`, `close`

22. **Optical flow** (2 points)
In this exercise we will apply our functions `MyOpticalFlow` and `MyOpticalFlowVisualizations` on three image sequences.

   a) Create a script named

      `CRV_22.m`

      The first eight lines should look like:

```
%% CRV_22_OpticalFlow
% name: John Doe
% student number: 11235813

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

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

   b) Download the zip-file 'sequencesOpticalFlow.zip' and unzip it.

   c) For every image in the first sequence, load the image and convert it to double.

   d) For the second till the last image call `MyOpticalFlow` and `MyOpticalFlowVisualization`. Choose a suitable file name and scaling factor.

   e) Try out different sizes for the local area Q in the range from 3 to 50. Write down your observations as comments in the script.

   f) Extend the script for processing the second and third image sequence. Write down your observations as comment in the script.

References

[1] MathWorks Matlab Help. *quiver: Quiver or velocity plot.* URL: `https://de.mathworks.com/help/matlab/ref/quiver.html`.

[2] MathWorks Matlab Help. *Quiver Properties: Control quiver chart appearance and behavior.* URL: `https://de.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.quiver-properties.html`.