Lab 2 – Optical Flow, Mosaicking & Stabilization

Goal: to provide an introduction to optical flow and tracking algorithms using Computer Vision System Toolbox. In this lab we'll learn to compute optical flow, generate mosaicking (panorama) vides and generate stabilized video.

1. Optical Flow

Use the command **doc** in Matlab command window. Then select:

Computer Vision System Toolbox→Examples.

Then find the section **Tracking** and choose **Tracking Cars Using Optical Flow**.

Then press the button **Open this Example** (in the top-right corner). Then copy the contents of the file **videotrafficof.m** to the file in your local directory.

1. Describe the algorithm used for cars tracking.
2. What optical flow algorithm is used in the program (Lucas-Kanade or Horn-Schunck)? Describe this algorithm.
3. What is the size of the matrix containing optical vectors (velocities)?
4. What is running mean? Explain and provide a simple formula. (Hint: Test on small matrices the values of **hMean1 = vision.Mean;** and **hMean2 = vision.Mean('RunningMean', true);** .)
5. Write a recursive formula for running mean and prove this formula using induction.
6. Define the *median*.
7. Usually, the median is applied on vector of odd length. But in general case the median can be applied on any vector. How the median of vector of even length is defined?
8. Suppose you have a histogram of the matrix of size $M \times N$ (both M and N are odd). Suggest a simple algorithm for computation of median from the histogram.
9. What value of the parameter **'NeighborhoodSize'** is used in the command **hMedianFilt = vision.MedianFilter;**?
10. Explain the command **vision.MorphologicalClose**.
11. Explain the command **vision.MorphologicalErode**.
12. Explain the use of the command **vel_th = 0.5 * step(hMean2, step(hMean1, y1));**.
13. Explain the use of the command **segmentedObjects = step(hMedianFilt, y1 >= vel_th);**.
14. Find a bug in the line **Idx = bbox(:,1) > lineRow;**. (Also, note that **Idx** is not used in the code, so, actually, this line can be deleted.)
15. What is the definition of the parameter **extent** in the command **isCar = extent > 0.4;**? Provide a simple formula.
16. In the command **[Y X] = meshgrid(CV,RV);** what is the direction of **X** and **Y**? Draw the directions (coordinate axes) of **X** and **Y**.

2. Mosaicking

Use the command **doc** in Matlab command window. Then select:

Computer Vision System Toolbox→Examples.

Then find the section **Registration** and choose **Video Mosaicking**.

Then press the button **Open this Example** (in the top-right corner). Then copy the contents of the file **videomosaicking.m** to the file in your local directory.

In the command **hcornerdet = vision.CornerDetector(...)** change the parameter **'Method'** to **'Harris corner detection (Harris & Stephens)'.** (Note 1: this is the default method. Note 2 (for general knowledge): correct: **'Rosen & Drummond'→'Rosten & Drummond'**.)

1. Describe the algorithm used for video mosaicking.
2. Describe the Harris & Stephens corner detection method.
3. What is *Random Sample Consensus (RANSAC)*?
4. Suggest a better implementation of the command **roi = int32([2 2 size(I, 2)-2 size(I, 1)-2]);** (line 105). (Hint: we want to accelerate the program.)
5. Explain the command **xtform = xtform * [tform, [0 0 1]'];** (line 141). Why do we compute **xtform** and don't use **[tform, [0 0 1]']** directly?
6. Explain the command **mosaic = step(halphablender, mosaic, transformedImage, transformedImage(:,:,1)>0);**. What is the role of the parameter **transformedImage(:,:,1)>0**?

3. Stabilization

Use the command **doc** in Matlab command window. Then select:

Computer Vision System Toolbox→Examples.

Then find the section **Registration** and choose **Video Stabilization.**

Then press the button **Open this Example** (in the top-right corner). Then copy the contents of the file **videostabilize.m** to the file in your local directory.

1. Describe the algorithm used for video stabilization.
2. Explain the command **vision.TemplateMatcher**. Describe the principle of the work of this command.
3. What is the default metric in the command **vision.TemplateMatcher**?
4. What is the meaning of the format string **%+05.1f**?
5. In Question 4 of the Mosaicking Section we suggested the better implementation of the command **roi=…** . Will a similar trick help to accelerate the command **Idx = int32(pos.template_center_pos);** (line92)?
6. Suggest a method for better implementation of the command **ROI = [SearchRegion, pos.template_size+2*pos.search_border];** (line 98). (Hint 1: we want to accelerate the program. Hint 2: Initialize **ROI** in the first frame, then make use of **SearchRegionRect**.)
7. In the command **Idx = step(hTM, input, Target, ROI);** (line 99) how the variable **Idx** is related to **Target**? (Is **Idx** the index of corner or of center of **Target**?)
8. The variable **Target** is initialized using the command **Target = zeros(18,22);.** Does the first execution of the command **Idx = step(hTM, input, Target, ROI);** (line 99) generate a wrong **Idx**?

9. Explain the command **MotionVector = double(Idx-IdxPrev);** (line 101). Why do we compute the difference **double(Idx-IdxPrev)** and don't use **Idx** directly?
10. Suggest a method for better implementation of the command **SearchRegionRect = [SearchRegion, pos.template_size + 2*pos.search_border];** (line 117). (Hint: we want to accelerate the program.)