הבינתחומי הרצליה
IDC HERZLIYA

Assignment 3
Computer Vision
Fall 2018

Yael Moses

Motion – OF & Change Detection

Submission Date: 24.12.18
Submit to moodle
You are encourages to submit this assignment in pairs.

Note: You should work on grey level images

Part A: Computing OF using Lucas-Kanade

1. **Write a matlab function:**
   [U,V]=OF(F1,F2, Sigma_S*, Region)*
   F1,F2: two frames from a sequence.
   Region is the local neighborhood window for computing the matrix *A*.
   Sigma_S = spatial Gaussian `smoothing` parameter
   (see matlab tips below).
   Output are 2 matrices:
   U is the x-component flow
   V is the y-component flow

2. Display the results using the matlab function quiver (see below matlab tips).

3. To test your method on real data download the following two sequences from the data folder. You can choose 2 sequences, and you are also welcome to choose your own sequence.
   Use rgb2gray to convert a color image to a gray level image.

4. **Run:**
   a. Choose several pairs of frames from each of the sequences.
   b. Play with the algorithm parameters.
   c. Play with the distance between the pair of frames (successive, jumps of k frames for several different values of k).
   d. Find examples that demonstrate the effect of the parameters and the `jump' size. Keep representative examples of the

results in a script file.
NOTE: change one parameter at a time to test its effect.

5. **Answer:** write your observations from (4). Show in the text file the examples that demonstrate your observations.

6. **Evaluation:**
   Use wrapping as in HW2, to evaluate the results.
   Present the absolute difference between the wrapped images and the goal image.
   Write the sum of the differences over all pixels.
   Compare for different parameters.
   For visualization:
   Use imshowpair(I1,I2) before warping and after warping.

7. **Run:** resize the image to see if you obtain different optical flow for different scales. Again, consider small and large `jumps' between frames.

8. **Answer**:
   a. Write your observations from 7.
   b. Assume you compute the OF in 3 with different scales. What is the right order of computing the OF in the different scales and how it should be used to overcome large motion?

9. **Bonus:** Compute OF using pyramid as we learned in class. You can fill in the missing parts of the algorithm by yourself. You can use matlab *imwrap()* function.

Part B: Segmenting OF

Note: you can wait for class 8, where we will talk on Segmentation.

10. **Write a matlab function:** *seg_OF_M= seg_OF_magnitude(U,V,th)*
    Input: U,V – the optical flow
    Th: is a threshold
    Output: a binary map.
    ***Output*:** segmented image – each segment in a different color.
    Detect pixels with large optical flow as foreground and low optical flow as background (|M(x,y)|>th), where M is the magnitude of the optical flow.
    Note that any segmentation should be a connected component.
    BWlabel – finds connected component in an image.
    Display the results – each connected component in a different color on a pair of frames and on the entire sequence (see matlab tips).

11. **Write a matlab function:** *seg_OF_O=seg_OF_orientation(U,V,th)*
    Input: U,V -- optical flow
    Th: is a threshold
    Output: a binary map.
    Detect pixels that move to the same direction. You should use threshold on the angles. You can use a set of thresholds rather than one threshold. It is similar to split the directions into several bins rather than just two.

    Output: as in 10.

Part C: Change Detection

12. **Write a matlab function**: *seg_CD=change_detection(seq,th)*
    Input: seq is an $w_x h_x F$ matrix where $w_x h$ is the frame size
    and *F is the number of frames.*
    You need to convert the vid into such a matrix.
    t*h* is a threshold.

    Output: a binary map.

    Compute a simple change detection algorithm on the input sequence. Use the median of a set of frames as a background, on a partial set of frames (you can decide how many).
    You can (you do not have to) apply some post-processing to remove noise (e.g., small foreground regions).

13. **Apply:** apply your function to the sequence SLIDE.avi

14. **Answer:** Compare the segmented results of the segmentation of moving regions using the OF algorithm and the change_detection algorithm. In which region each of them workd better? What may be the reason? When would you like to use each of the algorithms?

15. **Think:** Can you use it on other sequences as well? If not why?

Part D: Answer the following questions

16. Under which assumptions thresholding the magnitude or the orientation of the optical flow can be used to define background and foreground regions?

17. Assume two pixels have the same optical flow. Does it necessarily implies that they are projections of two 3D points that move at the same speed and direction? If so explain why, and if not explain what may cause the identical optical flow.

18. Suggest a method to use the optical flow to determine whether the scene is planar. You can assume that the camera is moving.

19. Assume a video of a static scene is captured by a camera that is translated in the x direction (of the cameras coordinate system) in a fixed velocity.

    a. What is the expected optical flow?
    b. Can you determine which region in the image is closer to the camera and which one is far? If so, explain how, and if not, explain why.

Implementation notes:

- If smoothing in the spatial domains, try using a very gentle Gaussian filter (i.e., a small sigma).

- Estimate the gradients in the temporal domain using the difference between the intensities of pixels in successive frames.

- If the local neighborhood window contains out-of-bounds pixels, then consider only the pixels that are within the movie frame.

- For each pixel *(x,y)* in time *t*, test the rank of the matrix *G* and if it's not 2 set U(x,y,t)=0, V(x,y,t)=0.

Matlab tips:

1. *rgb2gray*(im) – returns a gray level image.
2. Look at the script-HW3 for reading a video and presenting the results on all the video. You can use just part of the video if it is too long.
3. You can also resize the images in order for the program to run faster: look at help('imresize');

4. To read an avi file in matlab use:
   MOV= VideoReader ('file_name');
   seq=read(MOV);
   Seq is a 4 dimensional matrix of the sequence of color images. In particular, *seq(:,:,:,k)* is the k frame and

seq(:,:,c,k) are the red , green and blue channels for c=1, 2 or 3, respectively.
Have a look at : *imshow(seq(:,:,:,100));*
For moving to gray image use: *rgb2gray(seq(:,:,:,k));*
Have a look at *: imshow(rgb2gray(seq(:,:,:,100)));*
You can also resize the images in order for the program to run faster. Look at help('imresize');

5. To write your results to an avi file you should look at help('VIDEOWRITER');
   For example, you can read the people sequence and create a video:
   for i=1:10, seq3(:,:,:,i)=imread(sprintf('people2_%d.jpg',i),'jpg');
   end;
   vidObj = VideoWriter('people.avi');
   open(vidObj);
   for i=1:10, writeVideo(vidObj,seq3(:,:,:,i)); end
   close(vidObj);
   You do not have to submit an avi file but just representative frames.

6. To compute the derivative in the x and y direction – you can use the functions from HW1.

7. Use the matlab function *find* for thresholding.

8. Use the matlab function quiver to plot the optical flow. Since you do not want to draw an arrow for each pixel, you should draw one of each neighborhood (e.g., 5x5).
   For example, let im1 be the first image and u12 and v12 the optical flow components. Then you can present the results overplayed on im1 by:

```
[X Y]=meshgrid(1:size(im1,2),1:size(im1,1));
nu12=medfilt2(u12,[5 5]);
nv12=medfilt2(v12,[5 5]);
figure; imshow(im1,[]);
hold on;

     quiver(X(1:5:end,1:5:end),Y(1:5:end,1:5:end),…
     nu12(1:5:end,1:5:end),nv12(1:5:end,1:5:end),5);
(the above two lines should be one code line)
Note: you can play with the last argument (here
it is 5). You can also decide to display only pixels
with OF higher than a given threshold.
```

Submit:

A zip file, which includes the following, files:

A. **All** the documented functions you wrote, and all the files required to run them (except for the input images and the toolbox).

B. HW3.doc :
   Your name and id.
   Answers to questions and all the relevant comments

C. Run3.m :
   A documented script file that runs the functions and shows its output.

Good luck