Computer Vision & Image Analysis / Digital Image Processing

EXERCISE 5 – Image Segmentation

This problem asks you to separate foreground (a walking pedestrian) from background (everything else) in a video sequence, and to create a few sample frames of a new video with a new background (a green meadow under a blue sky), but the same foreground. If you are curious, you can watch the video sequence at http://www.youtube.com/watch?v=Jwi7uboNlTs. I found it fun to create the entire output video, which you can watch at http://www.youtube.com/watch?v=hS0U1q4XkWw. As you see from the output, results are far from perfect, but then the technique used is very simple. For this problem, it will be useful to work with both medians and median filtering.

Consider the following assignments over the problem

- You will work with two versions of the input video: The original, color version is used to create the foreground of the output video. A gray-valued version of the input video, in which a pixel is an integer between 0 and 255, will be used to select which pixels belong to the foreground and which to the background. Write code to convert color video to gray, without overwriting the color video, which you will need later. Since perceptual subtlety is unimportant for this application, a simple conversion is enough: a gray pixel value in the output video gray is just the average of the corresponding red, green, blue values. Hand in your code snippet for this conversion.
- The pedestrian in the video is small relative to the size of a frame, and walks through the entire frame over the course of the video. So the pedestrian only spends a small number of frames at any one pixel. In addition, the camera does not move. Explain how you can use these two observations to write a simple algorithm that estimates the value of each of the 160×240 pixels of a gray-valued image that contains only the background, without the pedestrian.
- Hand in the snippet of your code that generates the background image. The input is the gray-valued video gray you created earlier, and the output is a 160 ×240 gray-valued image named 'background'.
- Explain briefly why the task of generating the background image would be much harder if the camera were moving.
- Let frame be one of the frames in the gray-valued video, and background the gray-valued background image you generated above. Write code that produces a 160 ×240 binary image named mask that takes values in {true, false} and is true for pixels on the pedestrian body in frame, and false at background pixels. This is hard to do precisely, just think of an approximate method. You may have to do some trial and error. Do not expect perfect results. Provide a brief explanation and a code snippet.
- Show images of your mask for frames number 25, 50, 75, 100 (here and elsewhere, you are free to start counting frames at 0 or at 1, whatever is most convenient). I recommend encoding true with black and false with white to save ink.
- If everything went well, you will still notice many errors in the mask. Many of these errors are in the form of individual wrong pixels surrounded by many correct pixels. What is a good way to clean up the latter type of errors?

- Do that, and show your cleaned-up mask for sample frames number 25, 50, 75, 100. Do not expect perfect results.
- Identify an image region in one of your sample frames where errors are still significant, and explain briefly why they occur.
- Use your masks to superimpose the pedestrian onto a new background image given as file meadow.png on the homework web site. The result (both foreground and background) should be in color. Hand in the relevant code snippet, where color is the original color video, mask is the cleaned-up binary mask computed earlier, the variable newBackground is the new 160 ×240 ×3 color background image, and cframe is the resulting 160 ×240 ×3 color frame.
- Show your result for frames number 25, 50, 75, 100.

Access the supplementary files from https://courses.cs.duke.edu/fall11/cps274/homework/2/