DIP - Final Project

Due on **Saturday, December 9** at or before 11:59 pm

For the final DIP project, students will work in teams of two on the implementation of image processing algorithms. The assignment of students to teams can be found on ICON (see ICON —> DIP —> People —> Groups). Each team member is expected to contribute equally to the project. The project consists of two parts as described below.

Part I
Consider the image shown in Fig. 1. Utilizing the Hough transformation in combination with suitable pre- and post-processing methods (e.g., edge detection, filtering, …), design and implement an algorithm that detects (Fig. 2) and counts all wood logs in a given image (e.g., Fig. 1). Each detected log must have a corresponding label mask in the output image (segmentation result) to indicate which pixels of the input image belong to it (i.e., one, unique label per log). Furthermore, generate a CSV file (name: "g?_part1.csv", replace "?" with your group number) containing the following information:

> "<mask number>, <row-coordinate of center>, <column-coordinate of center>, <radius>".

An example is given below.

> 1, 910, 1262, 121
> 2, 1324, 519, 102

Part II
a) Consider the image shown in Fig. 3. Design and implement an algorithm that detects (Fig. 4) and counts all price and barcode tags in an image (e.g., Fig. 3). Each detected tag must have a corresponding label mask in the output image (segmentation result) to indicate which pixels of the input image belong to it (i.e., one, unique label per price and barcode tag). Furthermore, classify tags into price (gray background) and barcode (white background). For implementation, Matlab functions like `bwlabel`, `regionprops`, and `ismember` might be helpful. Furthermore, generate a CSV file (name:
"g?_part2.csv", replace "?" with your group number) containing the following information:

> "<mask number>, <upper-left row index of bounding box>, <upper-left column index of bounding box>, <lower-right row index of bounding box>, <lower-right column index of bounding box>, <tag type>".

The tag type is 0 for barcode tags and 1 for price tags.

b) Generalize the algorithm developed in part II.a so that it can be applied to new image data. For this purpose, take 6 new images with a camera. Make sure that the images contain several instances of price or barcode tags and that you have 3 images with "simple" as well as 3 images with "challenging" background in your image set, each with a different scale.

Note that you must design your algorithm for parts I and II.a such that they can be quickly applied to new, previously unused test images. Your grade will depend on the performance of your algorithms. Furthermore, perform a quantitative performance analysis. In your report, show and discuss your results. Explain the rationale behind your approach.

Make sure that your code utilizes MATLAB efficiently, similarly, as discussed in lecture. Submit all files needed (including image data) to run all experiments that you have performed. Include a report describing your work. Discuss your approach as well as any problems that you have encountered. Also, discuss all experiments and results. Make sure to adequately include/display results (e.g., images). Specifically, address any questions given above. Include a "conclusions" section, in which you describe what you have learned from your

experiments. The report must be in PDF format. Collect all files, compress them using zip, and submit the compressed file on ICON.

Note that one submission/report per team is sufficient. Furthermore, be prepared to present, demonstrate, and discuss your implementation on **December 14 (5:30pm-7:30pm, 3315 SC)**. More details will be provided in December. Also, be prepared to apply your algorithm to new, unseen image data. Furthermore, it is important to not swap row and column coordinates as this will impact the automated performance assessment of your algorithms. Similarly, make sure that there is only one entry per row in the CSV file.

Fig. 1. Image “woodlogs_b.png”.

Fig. 2. Example of a log detection result.

Fig. 3. Image “ptag_b.png”.

Fig. 4. Example for price and barcode tag detection result.