Hands-on Application Project
On Video Reflection Removal
HL

1. The goal: to design and implement Computer Vision algorithm for video reflection removal.

2. Total point: 20 (20% of the total semester score).

3. Input test video: provided by Professor in the class from the real testing environment. See reference photo provided in the class.

4. Rubrics for this assignment

4.1. (20 points) By the end of the project, the program implementation demonstrate successfully removal of the reflections from the test video;

4.2 (10 points) Finish implementation and testing of flood-filling algorithm to

(4.2.1.) Generate binary image from video sequence by proper color/gray-scale thresholding;

(4.2.2) Implement flood-fill algorithm to color each and every binary patterns from either the image or from the video in (4.2.1).

(4.2.3) Set threshold for the flood-fill algorithm to remove either binarized image or binarized video sequence;

4.3 (10 points) Implement contour computation algorithm to realize 19 feature extraction functions discussed in the class;

4.3.1 Use the sample program provided from the professor in the class to compute 19 features of the given 5-by-5 image;

4.3.2. Perform hand calculation to verify the programming implementation to match up the programming calculation.

4.3.3. Based on the programming implementation result, compute 19 contours features for each of every binary patterns from the image/video in (4.2.1).

4.3.4. Based on the attribute table given in the class, design and implement your attribute table.

4.4.5. Implement decision making function based on your attributes table.

4.4.6. Implement decision-making function.

4.4.7. Perform integration test by building flood-fill program as lib function as preprocessing stage for random noise removal.

4.4.8. Run integrated program on the test image/video demonstrate reflection removal result.

4.4.9. Make discussion and first submission (on-going work submission) by April 26 and submission on May 3rd.

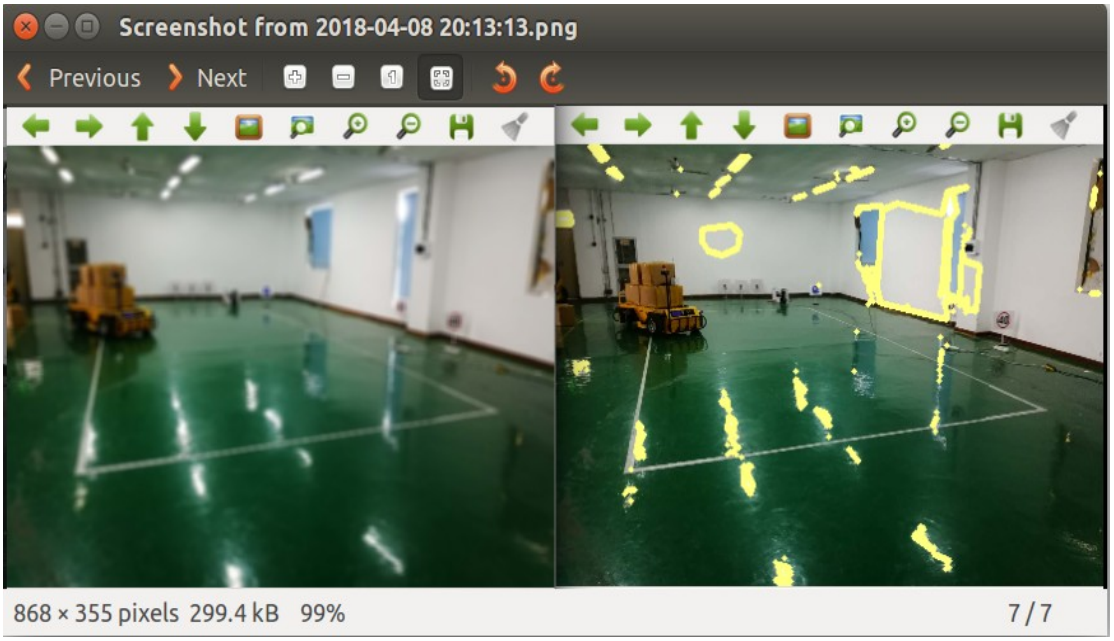

Figure 1. Test image (left), illustration of processed image (processed image from Harry Li).

(END)