Assignment 3

Basic Image Processing Algorithms
Fall 2022

General rules

This is the third Assignment. There will be one more Assignment during the semester.

Point value: **30 points**, which is 30% of the total assignment points.

Deadline: **November 21, 2021 23:59:59** (grace period ends on Nov. 23)

This is a not-guided exercise. The description of this assignment is general and does not focus on the details as in case of the Lab exercises.

The main task is to provide a good, reasonable solution. You may code “freely” (only minimal restrictions on file names and outputs are given).

Problem formulation

You have to write a custom image recovery and texture identification program that can remove optical artefacts from a CCTV image and can identify various food-objects based on their appearance.

Input: an image showing a distorted image captured by a camera

Output: a categorized (segmented) map of the captured foods

Tasks to do

Write an algorithm that can process each captured image as follows:

1. loads the image and the assumed motion blur kernel
2. uses an iterative deconvolution method (Richardson-Lucy)
3. applies local contrast enhancement (Wallis filter)
4. identifies the regions using a texture matching algorithm (Laws filter)
5. filters/enhances the result based on majority voting
6. returns a segmented image showing the clustered region map
7. Calculates (based on area size) the price of the food.

Key results to be presented:

You may code freely, as there are not so many restrictions on what to use.

However, you should create a script which solves the exercise and presents the following outputs:

Figure 1 shows the degraded (original) and restored image

Figure 2 shows the result of the Wallis filtering

Figure 3 shows the segmented image

Console output that shows for each food item (rice, fries, salad, meat) their detected area and calculated price

Figure 1

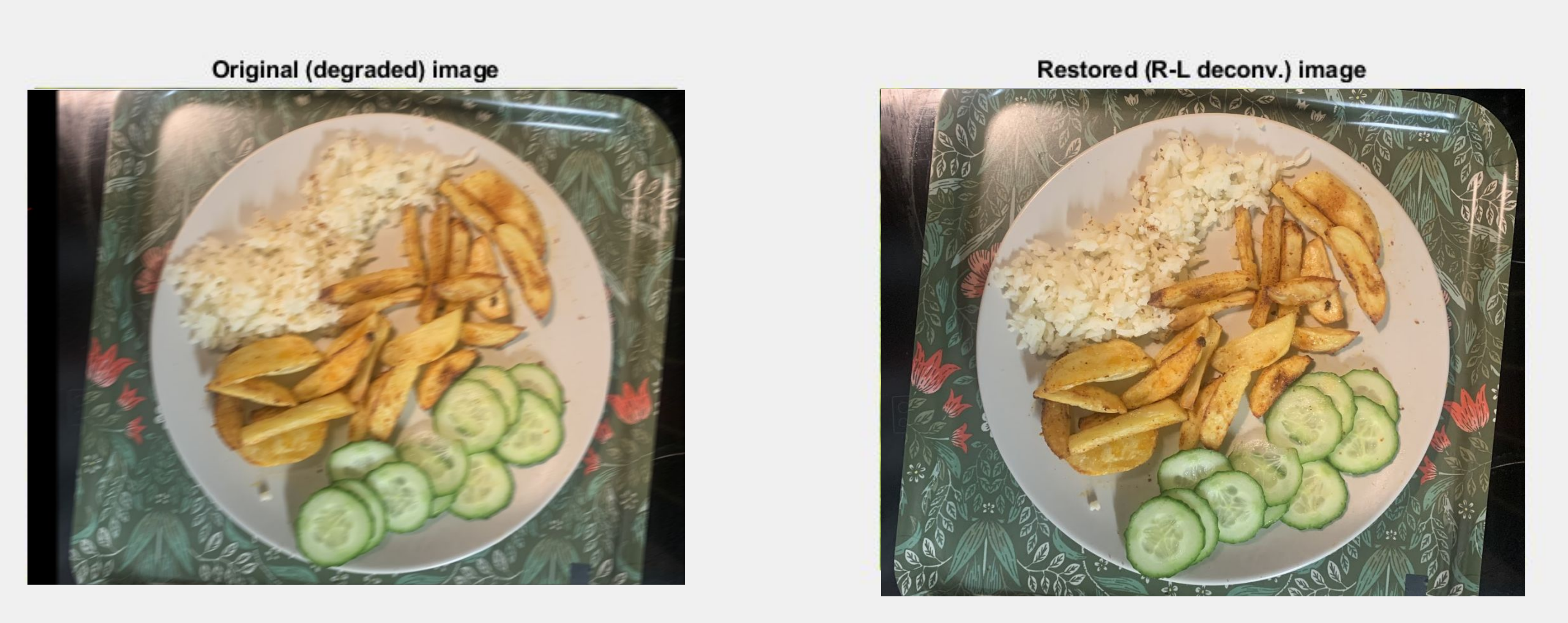


Figure 2

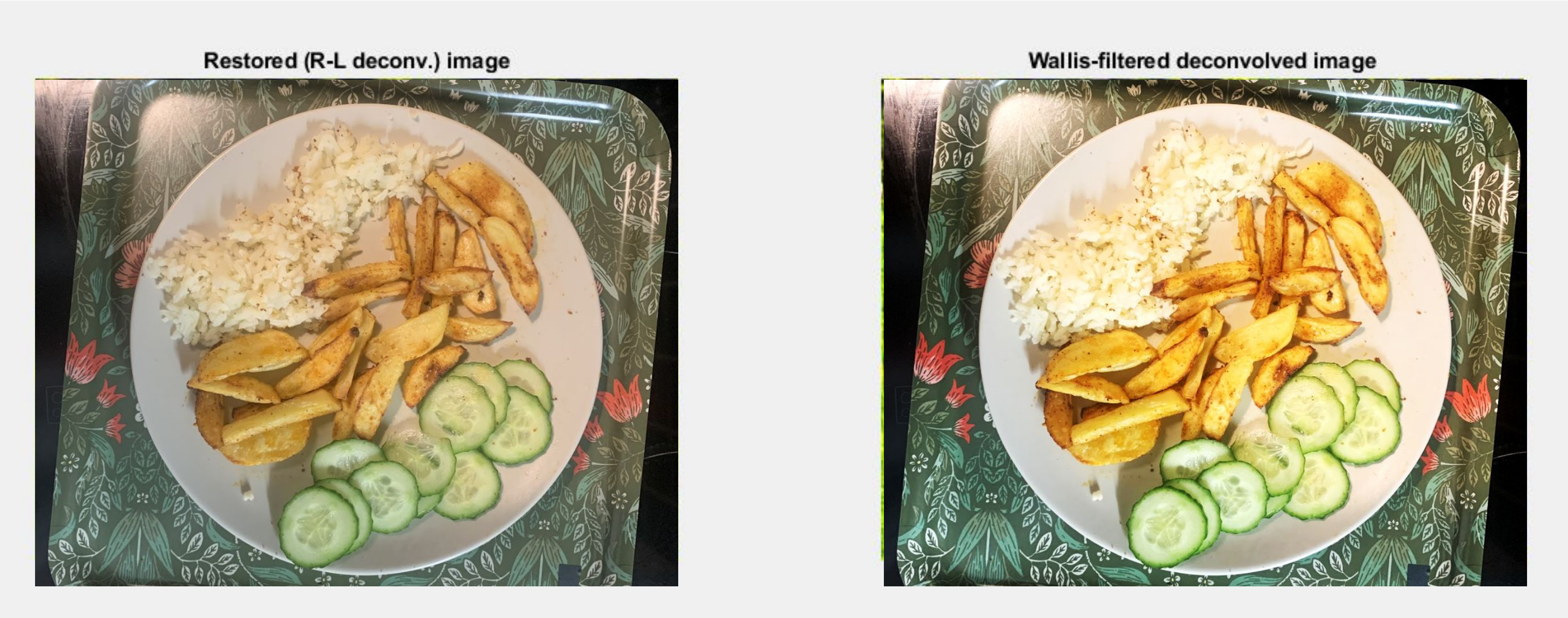
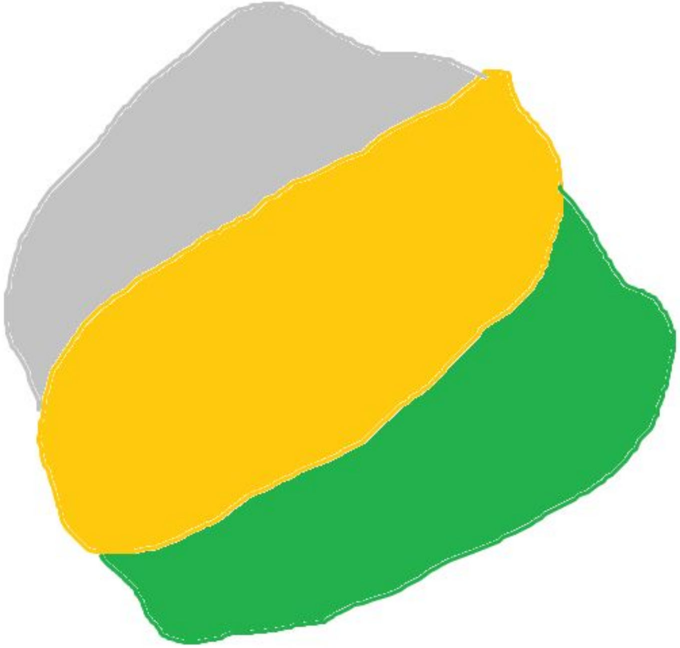

Restored (R-L deconv.) image
Wallis-filtered deconvolved image

Figure 3



There is no code package for this assignment.

All scripts and functions must be written entirely by you.

Download the image to be processed from here:
https://beta.dev.itk.ppke.hu/bipa/assignment_03

Submission & hints

You should create a script named `a03_NEPTUN.m` where the NEPTUN part is your Neptun ID. This has to be the main script; running that must be able to solve the problem.

You are allowed to create other files (e.g. additional functions) too, if necessary. **Usage of any built-in high level restoration functions** (e.g. `wallis`, `deconvlucy` etc.) **is prohibited.**

Please submit the source files only (**WITHOUT THE TEST IMAGES**) in a **ZIP** file via the Moodle system.

Check the upcoming slides for hints!

Hint 1

For each image there is a very well noticeable marker region on the tray. This pattern (marker) is a white dot in a black “quiet” zone.

This marker can be used to determine the point spread function of the motion blur effect.

Find a marker on a distorted image and measure the length and angle of the distortion by hand.

You can assume that the kernel is the same for all of the images.

The marker on a non-distorted image

Distorted marker

Hint 2

It is a very good idea to find the plate on the tray and crop that round region before applying the next operations.

For cropping, you may want to find large white objects on a binary mask and use any built-in high level circle-cropping algorithm that you can find in Matlab.

It is allowed to subtract the empty plate image from all of the images and/or create a mask and apply it to the image before further processing steps.

Hint 3

The Richardson-Lucy reconstruction method is described in details in multiple articles [1,2]. The key idea is to apply the following iteration rule:

$$\hat{u}^{(t+1)} = \hat{u}^{(t)} \cdot \left(\frac{d}{\hat{u}^{(t)} \otimes P} \otimes P^* \right)$$

Where d is the degraded image, u is the reconstructed image and P is the PSF of the imaging system.

Also, it might be beneficial to read the following article:
https://en.wikipedia.org/wiki/Richardson%E2%80%93Lucy_deconvolution

Hint 4

There should be a stopping criteria for the iteration. It can be hard-coded or it can be based on some measures (e.g. the maximum pixel intensity in the result of the Laplacian-of-Gaussian filtering is above a threshold).

Hint 5

The R-L deconvolution uses convolution. Please DO NOT use convolution in the spatial domain (which is a slow operation). Transform everything to the frequency domain (DFT) where the convolution is just a multiplication.

Hint 6

The Wallis filter [3] was described in detail at the beginning of Lecture 4:

$$y(n_1,n_2) = [x(n_1,n_2) - \bar{x}(n_1,n_2)]\frac{A_{max}\sigma_d}{A_{max}\sigma_l(n_1,n_2)+\sigma_d} + [p\bar{x}_d + (1-p)\bar{x}(n_1,n_2)]$$

Please check the Lecture slides (Slide 4).

Suggestions on what parameter values to use:

Desired local mean: ½ of the top value of the dynamic range

Desired local contrast: ⅕ of the top value of the dynamic range

Amax: from the [1, 5] interval p: something small (e.g. 0.2)

Hint 7

Texture samples of the Laws filtering should be fabricated by hand using small samples from each region of the input (restored, filtered) image.

Since the images are color, one can implement an RGB Laws filter which incorporates color info in the decision making. You can implement this by using 3D texture samples or by applying Laws filter on each color layer separately.

Please be smart and creative!

Hint 8

The regions on the segmented image should be represented by their corresponding cluster index (1, 2, 3, 4).

Use the built-in function `imagesc` to visualize the result.

Hint 9

To be able to generate the console output, create a “Look-up-table” in which you store the names (class labels) and unit prices for the different food items. See next hint for detailed description.

Hint 10

The plate is 25 cm in diameter.

The food unit prices (by area in 100 cm²) are:

salad

€0.77 / 100 cm²

fries

€0.60 / 100 cm²

schnitzel

€2.00 / 100 cm²

rice

€0.55 / 100 cm²

Grading

The final score of this assignment is the sum of the following points:

The script filename is correct, it’s a script	2 points
Script loads the image	1 point
PSF assumption is based on the marker at the corner, kernel is OK	1 point
Figure 1 exists and looks good	1 point
Plate is masked / cropped form the image before further processing steps	2 points
R-L deconvolution idea is understood, formula is OK	2 points
R-L deconvolution stopping criteria is OK	1 point
R-L deconvolution parameters are appropriate	1 point
Figure 2 exists and looks good	1 point

Continues on next slide...

Grading

Wallis filter local average and contrast calculation is OK	2 points
Wallis operator implementation is OK	2 points
Wallis parameters are appropriate	1 point
Figure 3 exists and looks good	1 point
Laws kernels are OK	1 point
Laws sample textures are OK	1 point
Laws implementation is working, result is acceptable	3 points
Some kind of voting / post-processing is applied and makes sense	2 points
Area size detection working, results are OK	1 point
Console output exists and looks good	1 point
Images with missing textures (e.g. only rice) are processed correctly	1 point
Code quality (readability, understandability, good comments and structure)	2 points
TOTAL:	**30 points**

Contact

If you have any further questions regarding this assignment, contact

Márton Bese NASZLADY

via **Teams** (in private chat) or write an email to

naszlady@itk.ppke.hu

References

[1] Richardson, William Hadley (1972). "Bayesian-Based Iterative Method of Image Restoration". JOSA. 62 (1): 55–59. doi:10.1364/JOSA.62.000055

[2] Lucy, L. B. (1974). "An iterative technique for the rectification of observed distributions". Astronomical Journal. 79 (6): 745–754. doi:10.1086/111605

[3] Wallis, R., (1976). "An approach to the space variant restoration and enhancement of images". In: Proc. IEEE Conference on Computer Vision and Pattern, Naval Postgraduate School, Monterey, CA, USA

THE END