ex9

May 8, 2019

0.1 TODO Nuts and bolts

In this little project you will design and test a program that can recognize various nuts and bolts in an image using Matlab's morphological functions and a bit of statistics.

The image can be seen in Figure 4. Matlab contains an excellent tutorial segmenting and counting rice in an image, which you can work through as preparation. The principle steps that need to be done are the following: - Segment the foreground which contains all parts, from the background. You can use 4 http://www.mathworks.ch/ch/help/images/image-enhancement-and-analysis.html morphological opening, e.g. imopen to ascertain background statistics or use the so called Otsi's method implemented by Matlab graythresh. - Use morphology to remove any noise from the image - Select all individual items using Matlab's bwlabel and bwconncomp. - To gather statistics deploy Matlab's regionprops function. It is capable of collecting a vast amount of information on binary objects which in term can be used to distinguish the various parts from each other. - Find a combination of metrics to separate the different parts as best as possible.

0.1.1 Exercise 9. ex

1. Implement the image segmentation and statistics gathering functions

[ ]:

```
# otsu method from skimage
import matplotlib.pyplot as plt
from skimage import data
from skimage.filters import threshold_otsu
image = data.camera()
thresh = threshold_otsu(image)
binary = image > thresh

fig, axes = plt.subplots(ncols=3, figsize=(8, 2.5))
ax = axes.ravel()
ax[0] = plt.subplot(1, 3, 1)
ax[1] = plt.subplot(1, 3, 2)
ax[2] = plt.subplot(1, 3, 3, sharex=ax[0], sharey=ax[0])

ax[0].imshow(image, cmap=plt.cm.gray)
ax[0].set_title('Original')
ax[0].axis('off')

ax[1].hist(image.ravel(), bins=256)
```

```python
ax[1].set_title('Histogram')
ax[1].axvline(thresh, color='r')

ax[2].imshow(binary, cmap=plt.cm.gray)
ax[2].set_title('Thresholded')
ax[2].axis('off')

plt.show()
```

2. Report on what statistics work and why (not).