DIIP PRACTICAL ASSIGNMENT

Lappeenranta-Lahti University of Technology LUT

BM40A1201 Digital Imaging and Image Pre-Processing, Practical assignment

2024

0566770 Alex Karonen

TABLE OF CONTENTS

1 Introduction

The given task was to create an imaging system which can detect coins and their counts from given image using basic image processing methods. The task setup consists of a fixed camera, a varying number of coins (of Euro currency) and a fixed checkerboard in the background. By using the known properties of the coins, the setup and some extra reference images, it is possible to fairly accurately determine the composition of coins in each given measurement image. For completing this task we were given 12 measurement images (see figure 1), 5 bias images, 5 dark images and 5 flat images. Also the size of the checkerboard squares, the diameters of each coin [3] as well as the ground truth about the counts of the coins in each image were given.

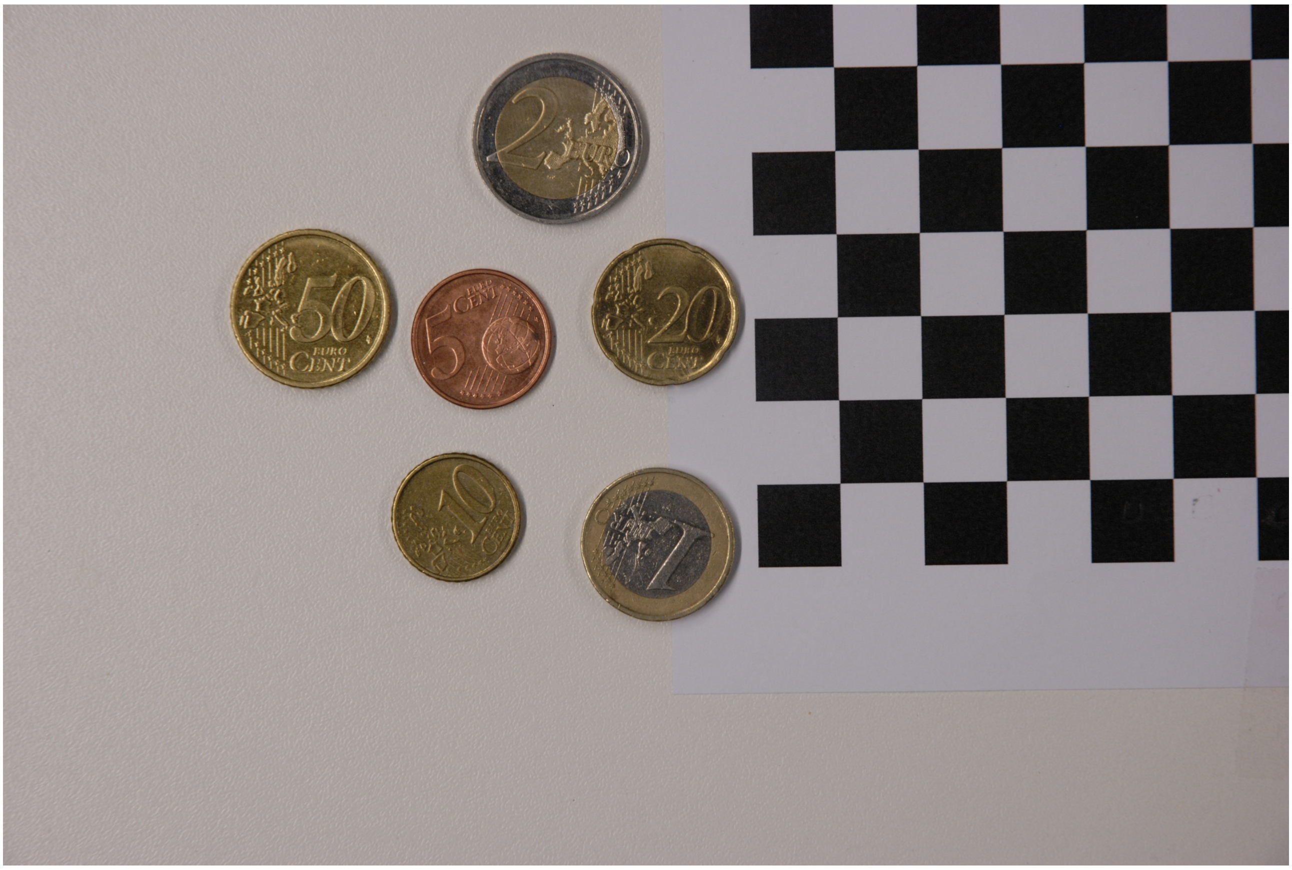

Figure 1. Example of a measurement image (DSC1772.jpg)

2 Method

2.1 Setup

The measurement setup used in the task consists of a camera which is fixed to a stand, a light background (a table), a checkerboard which is in a fixed position in the region of interest (top right corner of the images) and the coins. The lighting in the setup is not fixed which leads to a intensity variance in different images which is why intensity calibration is needed [2]. The bias images have been taken with some solid object blocking the camera sensor completely so that minimal amount of light from the environment is captured. These images give us the noise in the measurement images that is created by the camera system itself. The dark images have been taken without a blockage in the camera sensor but in a completely dark setup room. These images tell us what is the noise in the images created by the dark current in the imaging sensor system. The flat images on the other hand are taken in varying lighting with only the background and the checkerboard visible. These can be used in the geometrical calibration of the images as well as the measurement image intensity calibration. Calibration of the camera system has been done correctly which can be determined from the fact that the images have minimal lens distortion. Thus no lens calibration is needed in this method.

2.2 Preprocessing

The measurement image calibration is done using the mean bias, flat and dark images taken beforehand. Using the mean flat image, we can also determine the geometric calibration of the system i.e. find out how large is one pixel in a image compared to real life units (millimeters). This is done by choosing an arbitrary row and a column of pixels (the 500th row and 500th last column in this case) from the mean flat image which contain the checkerboard. Segmenting these lines of pixels by finding the pixels which have a mean value under 100 over the RGB channels, we can find the pixels which coincide the black squares in the checkerboard. Thus by analyzing the lengths of these segmented pixels i.e. finding the maximum number of ones in a row in these segmented lines of pixels (see figure 2). From the segmentation we can roughly determine the length of the squares in pixels and thus determine the conversion ratio of millimeters to pixels.

In addition to the geometric calibration, a image intensity calibration was done. The intensity calibration is done using the scaled mean flat, mean bias and mean dark images. The purpose of the intensity calibration is to minimize the effect of the differing environment lighting which in turn creates some differences in the luminous intensity in the measurement images. The calibration is done by using the equation 1.

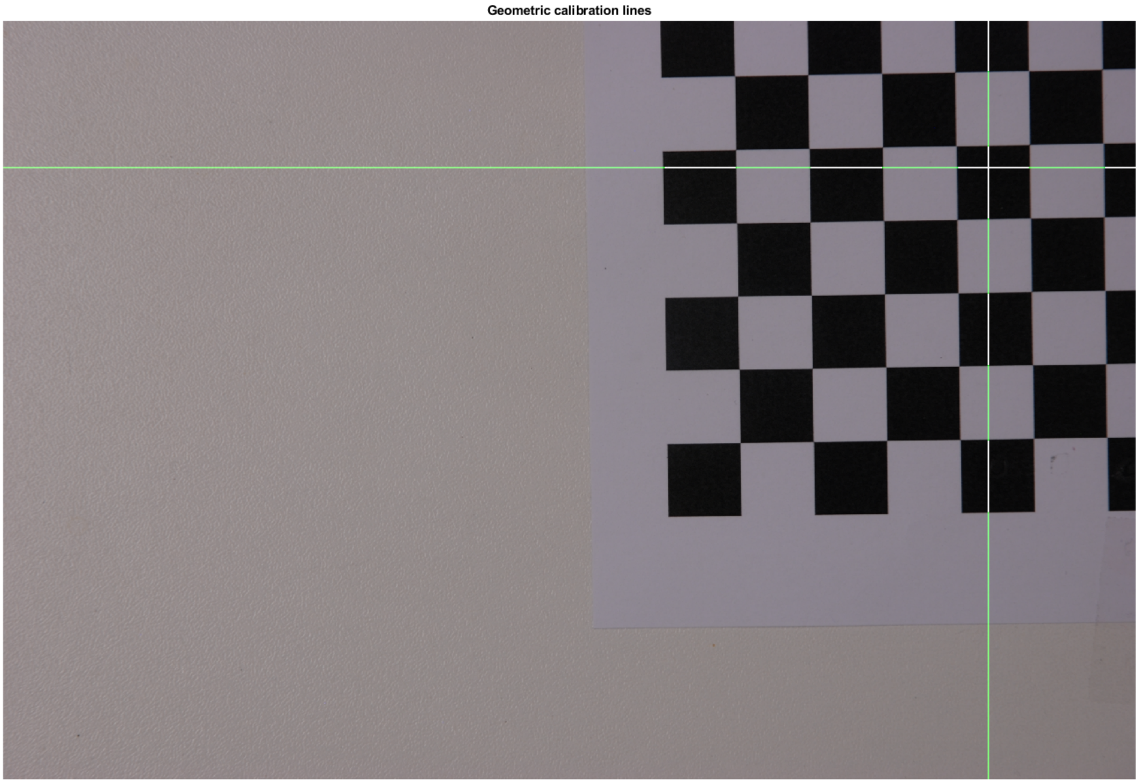

Figure 2. Pixels used in the geometric calibration (green = chosen pixels, white = segmented pixels).

$$R_{calibrated} = \frac{R_{raw} - D_{mean} - B_{mean}}{F_{scaled}}, \tag{1}$$

where R_{raw} is the original measurement image, D_{mean} is the mean of dark images, B_{mean} is the mean of bias images and F_{scaled} is the mean of flat images scaled to have mean 1. After the intensity calibration, the images are turned into grayscale for the segmentation.

2.3 Segmentation

After the image preprocessing, the goal is to segment the coins from the measurement images. In this solution the segmentation is done in four steps (see figure 3):

1. Binarization & inversion
2. Dilation & filling
3. Erosion

4. Finding of the circles/coins

Step 1 is done by finding a grayscale threshold value for the binarization using the MATLAB [1] function *graythresh* and binarizing using this threshold with the MATLAB function *imbinarize*. The resulting binarization is then inverted with the NOT-operator.

Step 2 uses a disk-shaped neighbourhood element of radius 21px to dilate each white pixel in the image using the MATLAB function *imdilate*. The resulting blobs of white regions is then filled using *imfill* with the option 'holes'.

Step 3 uses a square-shaped pixel neighbourhood element of sidelength 28px to trim and smooth out some of the unnecessary dilation artifacts using the MATLAB function *imerode*.

Step 4 uses the MATLAB function *imfindcircles* with object polarity of bright and sensitivity of 0.98 to find circles from the processed image. From the found circles we choose strongest candidates by thresholding the metric returned by *imfindcircles* with a value of 0.047. Then we compare these found circles and their diameters to those of the real coin diameters by using the conversion rate of mm to pixels. Lastly we assign labels to the circles by finding the closest match from the comparison i.e. finding the smallest difference in diameter compared to the real diameters.

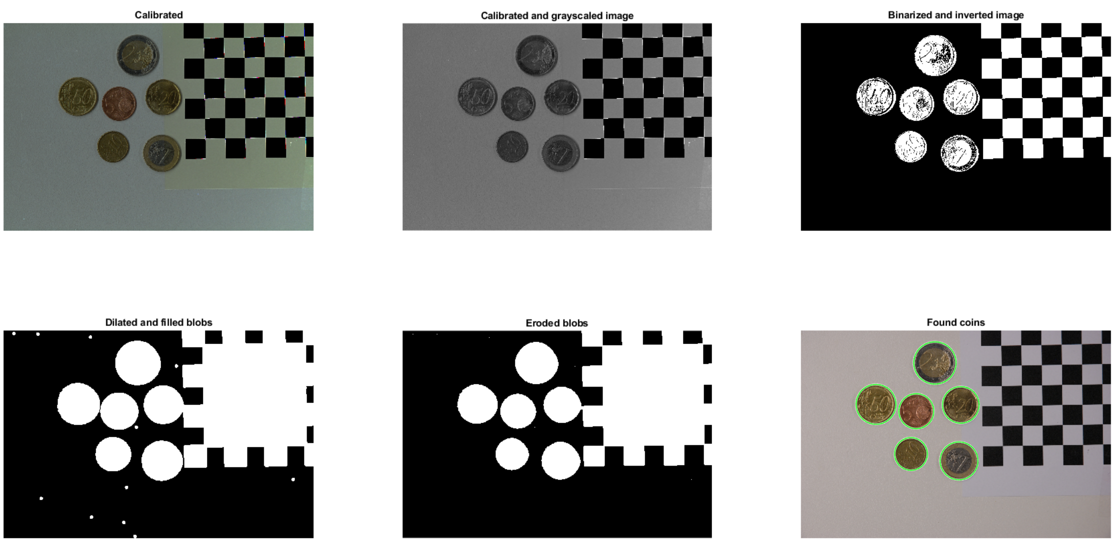

Figure 3. Illustration of the segmentation process on the measurement image DSC1772.

3 Results

Using the methods described in this report, we are able to achieve a total labelling accuracy of 96.05% and a found labelling accuracy of 98.65% over the given measurement images. In the table 1 are shown the statistics of each individual measurement image estimation.

Table 1. Table of the coin estimations (total accuracy = estim. vs. ground truth, found accuracy = found coins vs. ground truth).

Image	GT	Estim.	Total Acc.	Found Acc.
DSC1772	[1, 1, 1, 1, 1, 1]	[1, 1, 1, 1, 1, 1]	100%	100%
DSC1773	[3, 1, 0, 1, 0, 0]	[2, 0, 0, 1, 0, 0]	60%	100%
DSC1774	[1, 0, 0, 5, 1, 1]	[1, 0, 0, 5, 1, 1]	100%	100%
DSC1775	[0, 0, 0, 3, 1, 3]	[0, 1, 0, 2, 1, 3]	86%	86%
DSC1776	[0, 1, 0, 4, 1, 3]	[0, 1, 0, 4, 1, 3]	100%	100%
DSC1777	[0, 3, 0, 1, 0, 2]	[0, 3, 0, 1, 0, 2]	100%	100%
DSC1778	[0, 1, 0, 3, 0, 0]	[0, 1, 0, 3, 0, 0]	100%	100%
DSC1779	[0, 0, 1, 4, 0, 3]	[0, 0, 1, 4, 0, 3]	100%	100%
DSC1780	[0, 0, 0, 0, 1, 3]	[0, 0, 0, 0, 1, 3]	100%	100%
DSC1781	[0, 0, 1, 4, 0, 0]	[0, 0, 1, 4, 0, 0]	100%	100%
DSC1782	[0, 3, 1, 5, 0, 0]	[0, 3, 1, 5, 0, 0]	100%	100%
DSC1783	[0, 3, 1, 1, 0, 0]	[0, 3, 1, 1, 0, 0]	100%	100%

From the results we can determine that the method seems to work quite well with the provided images. Only with the image DSC1773 does the method fall off a bit, probably due to harsh illumination changes compared to the rest of the images which in turn leads into not so good binarization of the image and thus the segmentation fails to find all of the coins in the image.

SOURCES

[1] MathWorks ®. Last accessed 12 January 2024. URL: https://se.mathworks.com/help/matlab/.

[2] Assoc. Prof. Xin Liu. Last accessed 12 January 2024 through Moodle.

[3] Suomen Pankki. Last accessed 12 January 2024. URL: https://www.suomenpankki.fi/en/money-and-payments/euro-coins/euro-coins-common-sides/.