Tampere University

Roosa Kuusivaara & Väinö-Waltteri Granat

Digital Image Formation And Processing - Report

Faculty of Information Technology and Communication Sciences (ITC)
Laboratory Report

Abstract

Roosa Kuusivaara & Väinö-Waltteri Granat: Digital Image Formation And Processing - Report
Laboratory Report
Tampere University
Master's Degree Programme in Signal Processing
December 2023

This report documents the work done in the Digital Image Formation And Processing assignment as a part of the Advanced signal processing laboratory course. In this assignment we familiarized ourselves with basics of digital image formation from raw sensor data as well as image processing.

Keywords: Image processing, Image formation.

Contents

1 Introduction

In this report we describe our work done in the 'Digital Image Formation and Processing' laboratory assignment for the Advanced Signal Processing Laboratory course. In this assignment we implemented a basic raw image formation and processing pipeline in Matlab to construcut RGB images from raw sensor data.

2 Methodology

In this section we will present our implementation of the image processing pipeline, by going over the main parts of the Matlab code as well as explaining the the decisions we took when requirements we ambigious.

2.1 Overview of the pipeline

Figure 2.1 shows the complete image processing and formation pipeline that we are going to introduce in this report. The following sections will go over the different parts of the pipeline. The pipeline takes in raw sensor data of Bayer arrays in RGGB format and produces RGB images. In addition to image formation the pipeline also performs focusing, white balancing and contrast and saturation correction.

2.2 Reading images and converting to doubles

2.3 Image visualization and Bayer mosaic

2.4 Sliding window

2.5 Scatterplots and regression

2.6 Transformation and reverse transformation

2.7 DCT denoising

```
1  function [denoised] = DCTImageDenoising(image, lambda,
       transformBlockSize)
2
3      % Create a custom function to be applied to each block
4      fun = @(block_struct) idct2(thresholdDCT(block_struct.data,
           lambda));
5
6      % Apply the function to each block
7      denoised = blockproc(image, transformBlockSize, fun);
8  end
9
10 function denoised = thresholdDCT(input, lambda)
11
12     % Apply DCT to the block
```

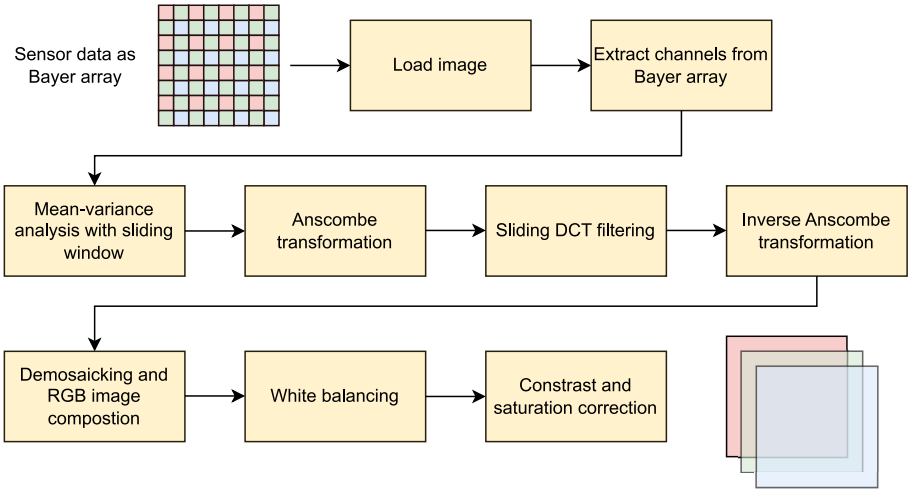

Figure 2.1 *Our image processing and formation pipeline*

```
13      dctBlock = dct2(input);
14
15      % Threshold the DCT coefficients
16      dctBlock(abs(dctBlock) < lambda) = 0;
17      denoised = dctBlock;
18  end
```

2.8 Demosaicking

2.9 White balancing

2.10 Contrast and saturation correction

3 Results

In this section we will present the results from our image processing pipeline and also do some analysis on why we got the kinds of results that we did.

4 Conclusions