Technology
Arts Sciences
TH Köln

Project Documentation

Interactive Lighting Detector

written by

Vera Brockmeyer (Matrikelnr. 11077082)
Laura Anger (Matrikelnr. 11086356)

Image Processing in SS 2017

Supervisor:
Prof. Dr. Dietmar Kunz
Institute for Media- and Phototechnology

Contents

1 Introduction

Vera

1.1 Motivation

Vera

1.2 Usage Context

Vera

1.3 Project Goal

Vera

2 State of the Art

Laura

2.1 Image Forensic

Laura

2.2 Light Vectors

Laura

3 Materials

Laura

The following sections describe the resources and tools required for the completion of the project. Furthermore, the test images are presented in chapter 3.3.

3.1 Hardware

Laura

During the implementation phase, the application was run on two computers, which are described in the following two sections. Both computers needed to be able to deal with the software components described in section 3.2. An extract from your data sheet is shown in table 1 respectively table 2.

3.2 Software

Laura

In order to develop the *Interactive Lighting Detector Qt* was used (compare section 3.2.1). To take advantage of already existing functionalities the *OpenCV*-library, which is described in section 3.2.2, was taken advantage of.

3.2.1 QT

Laura

The *QT Creator* was invented by *The Qt Company*. It is an integrated software development environment in the programming language C++. More functionality can be added by using the Qt project's library, which is called *Qt*. As a cross-platform tool, the *QT Creator* can be used on all common operating systems [4].
Besides extensive database functions and XML-support the software can build graphic user interfaces (GUI).
For this project the algorithm was transcribed in source code using the *Qt Creator* and the GUI was designed in the *Qt Designer* [3];

NAME?	Description
Processor	??
RAM	??
Graphic Card	??
Operating System	??

Table 1: Extract from the Data Sheet of the NAME?

.

Acer Aspire 5820TG	Description
Processor	Intel Core i3 CPU @ 2.40 GHz
RAM	4 GB
Graphic Card 1	AMD Mobilty Radeon HD 5000 Series
Graphic Card 2	Intel(R) HD Graphics
Operating System	Windows 10 Education 64 bit

Table 2: Extract from the Data Sheet of the Acer Aspire 5820TG Notebook.

3.2.2 OpenCV

Laura

The *Open Source Computer Vision* (OpenCV) is an open source library for image- and video processing, which is among others available in the programming language *C++*. It has been introduced ten years ago and is developed by various programmers since then. This library offers the most common algorithms, as well as current developments in image processing [1].
On the case of the implementation of the *Light Detector* the library was mainly used for the detecting of the contours (compare section 4.2) and solving the minimization problem (compare section 4.4) introduced by Johnson and Farid [2].

3.3 Testimages

Laura

Due to the assumption that the objects shown on the test images described in section 3.3.1 have a too complicated shape, a second batch of images was made (compare section 3.3.2). Images of both batches were used to test the functionality of the the algorithms used for the lighting detection. All images have in common that besides the actual object they show a sundial to simplify the determination of the light direction for the user.

3.3.1 First Batch

Laura

Four examples of the first batch of test images are shown on figure 1. Next to the mandatory sundial there are different objects depicted, like a helmet, a handbag, a bucket or a hot-water bottle. Those objects differ in their surface texture, as well as their size. They are shot from different angles to produce different light directions. it is necessary that the objects are not trimmed at the boarders of the image, because the algorithm requires a full contour of the selected object.

Figure 1: Examples of the Test Images of the first Batch.

3.3.2 Second Batch

Laura

In contrast to the first batch, the test images described in this section show easier objects with a round surface. As depicted in figure 2 all images show the madatory sundial and one or two table tennis balls in yellow and pink, which have a matt texture. For the actual algorithm of the *Light Detector* only one of this balls is taken into consideration (compare section 4). Properties like size of the object, the camera angle and the lighting direction differs in each image.

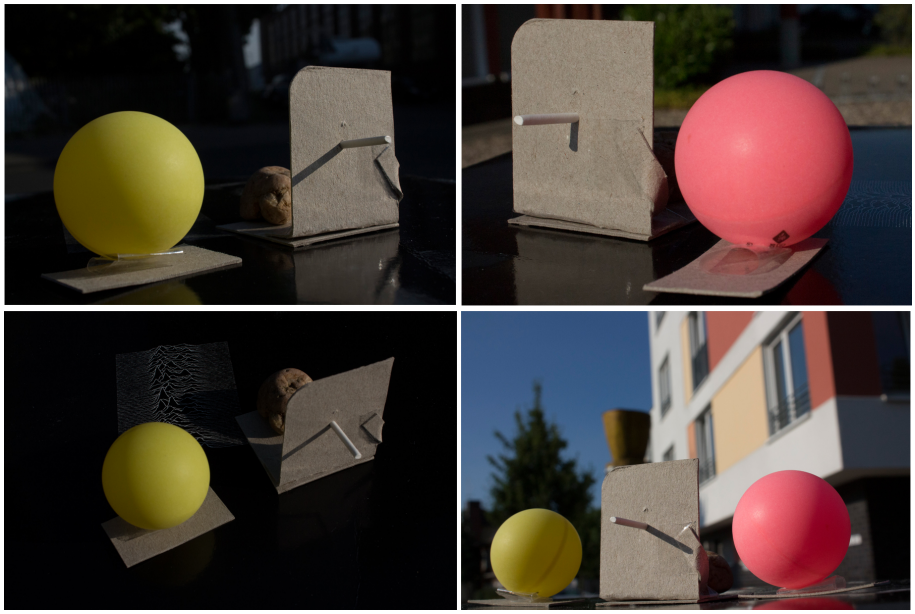

Figure 2: Examples of the Test Images of the second Batch.

4 System

Vera

4.1 Lighting Model

Vera

4.2 Contours

Vera

4.2.1 Find Contours

Vera

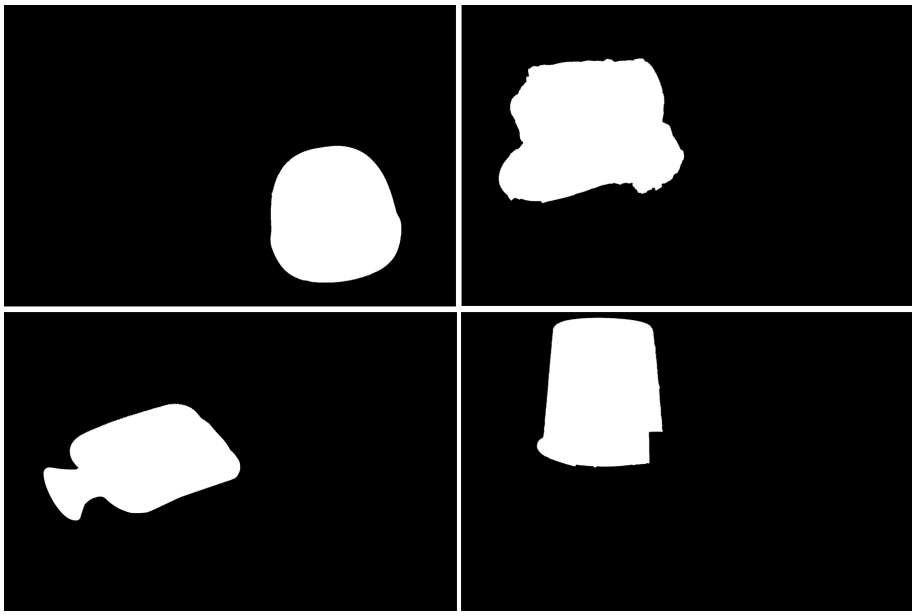

Figure 3: Bildunterschrift.

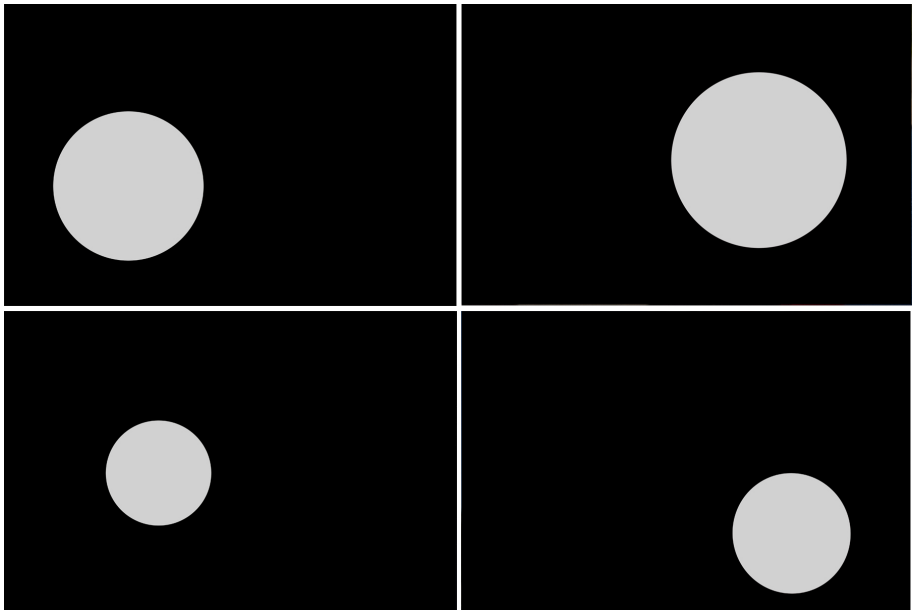

Figure 4: Bildunterschrift.

4.3 Subcontours

Vera

4.4 Different Approaches

Laura

4.4.1 1. Approach: One Lightvector

Laura

4.4.2 2. Approach: Averaging Lightvectors

Laura

4.4.3 3. Approach: Lightvector with highest Intensity

Laura

5 Evaluation

Vera und Laura: Stichpunkte

Vera: Ausformulierung

6 Project Management

Laura

6.1 Project Definition

Laura

6.2 Project Planning

Laura

6.3 Project Execution

Laura

6.4 Project Completion

Laura

7 Conclusion

Vera und Laura

References

[1] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek. A brief introduction to opencv. pages 1725–1730, May 2012.

[2] Micah K. Johnson and Hany Farid. Exposing digital forgeries by detecting inconsistencies in lighting. In *Proceedings of the 7th Workshop on Multimedia and Security*, pages 1–10, New York, NY, USA, 2005. ACM.

[3] Qt. Qt Designer Manual. http://doc.qt.io/qt-5/qtdesigner-manual.html. Aufgerufen: 13. Juli 2017.

[4] Ray Rischpater. *Application Development with Qt Creator - Second Edition*, volume 2. Packt Publishing, 2014.