Technology
Arts Sciences
TH Köln

Projekt Ausarbeitung

Kamerakalibrierung anhand eines Punktgitters

geschrieben von

Vera Brockmeyer (Matrikelnr. 11077082)
Artjom Schwabski (Matrikelnr. 11113320)

Weiterführende Themen der Bildverarbeitung in SS 2017

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

Inhaltsverzeichnis

1 Abstrakt

XXXX

2 Einleitung

Vera

2.1 Motivation

Vera

3 Stand der Wissenschaft

???

3.1 Kamerakalibrierung mit Punktraster

XXXX

4 Materialien

XXX

4.1 Hardware

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 **??**. An extract from their data sheet is shown in Table 1 respectively Table 2.

Acer E5-571G	**Description**
Processor	Intel Core i7 CPU @ 2.40 GHz
RAM	8 GB
Graphic Card	NVIDIA GeForce 840M
Operating System	Windows 10 Education 64 bit

Tabelle 1: Extract from the Data Sheet of the Acer E5-571G

.

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

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

4.2 Software

4.3 Entwicklung

Das Plugin zur radialen Entzerrung von regelmäßigen Punktrastern wurde in der Programmiersprache Java (Version 1.8.031) in der Entwicklungsumgebung Eclipse IDE for Java Developers (Version 4.4.1 Luna) entwickelt. Java lief in der Laufzeitumgebung Java SE Runtime Environment (Version $1.8.031 - b13$), welche auf dem Betriebssystem Windows 7 Professional installiert war.

4.4 *ImageJ*

ImageJ ist eine plattformunabhängige Open-Source Bildbearbeitungssoftware, die in Java implementiert wurde. Mit *ImageJ* können die gängigen Bildformate, wie TIFF,

JPEG oder GIF mit unterschiedlichen Bittiefen angezeigt, analysiert und bearbeitet werden [1]. Es ist weiterführend ein weitverbreitetes Werkzeug zur Entwicklung von Methoden und Algorithmen, die zur Analyse von Bilddaten vorgesehen sind. Die Vielzahl der Funktionen wird stetig von den Nutzern in Form von Plugins erweitert und verbessert, wie es in Open-Source-Plattformen gängige Praxis ist.

4.4.1 ImageJ Makros

ImageJ Makros [3] sind kleine Programme, die eine Abfolge von *ImageJ* Kommandos ausführen. Zur Erstellung dieser Programme wurde der Recorder entwickelt, welche die ausgeführten Kommandos als Textdatei abspeichert. Diese Textdateien können editiert werden und in *ImageJ* beliebig oft ausgeführt werden. Sie stellen eine erhebliche Arbeitserleichterung beim Testen von Algorithmen mit mehreren Testdaten dar, da sie schnell implementiert sind. Vor allem werden sie genutzt um ein geeignetes Verfahren oder die richtige Parametrierung zu ermitteln. Ist ein richtiger Algorithmus gefunden, kann die Textdatei einfach in Java übersetzt werden.

4.5 UnwrapJ

Artjom

4.6 *Apache Common Math* Bibliothek

Artjom

4.6.1 Klassen

Artjom

Im folgenden werden die Methoden der einzelnen Klassen erläutert. Die vollständige UML zur besseren Verständlichkeit der Klassenbeziehungen ist der Abb. 1 zu entnehmen.

point_grid_radial_affin_distor_ Hauptklasse der Anwendung. Implementiert das Interface *PluginFilter* um über ImageJ aufgerufen werden zu können.

Die Klasse besitzt folgende Methoden und deren Funktion:

run	Main-Methode des PlugIns in der die Optimierung aufgerufen wird
setup	Konstruktor-Methode des PlugIns in dem die Bildreferenz gespeichert wird
readData	Liest aus einer in ImageJ geöffneten Textdatei Punkt-Paare ein für Start- und Ziel-Koordinten
computeDrawRadialTransformation	
drawTargets	Zeichnet Punkte an den übergebenen Ziel-Koordinaten in das übergebene Bild
computeDrawAffineTransformation	
computeRadius2Center	Berechnet anhand der Parameter den Abstand zum Gittermittelpunkt
compute_radial_dist_koeff	Berechnet mit dem LevenbergMarquadt Optimierer die Koeffizienten der Radialen Verzerrung der übergebenen Punkt und gibt die Koeffizienten zurück

Tabelle 3: Methoden der point_grid_radial_affin_distor_ Klasse

SimplePair Eine Einfache Klasse zum Speichern der Vorgabe- und Ziel Koordinaten und des Abstandes zum Mittelpunkt.

RadialDistFunction Klasse zum Erzeugen der Funktionen für den Optimierer.

RadialDistFunction	Konstruktor der Klasse. Es wird ein SimplePoint Array erwartet welcher Koordinaten-Paare für Start- und Ziel-Koordinaten enthält.
realTargetPoints	Gibt ein Array aus welches nur die Ziel-Koordinaten enthält. Dieses wird für den Optimierer benötigt.
retMVF	Funktion zur Modellierung der Radialen Verzerrung für den Optimierer. BErechnet zu den Vorgegeben Koeffizienten und einer Start-Koordinate die Ziel-Koordinate
retMMF	Jacobi-Matrix-Funktion zur Berechnung der Ableitung nach den einzelnen vom Optimierer vorgegebenen Koeffizienten

Tabelle 4: Methoden der RadialDistFunction Klasse

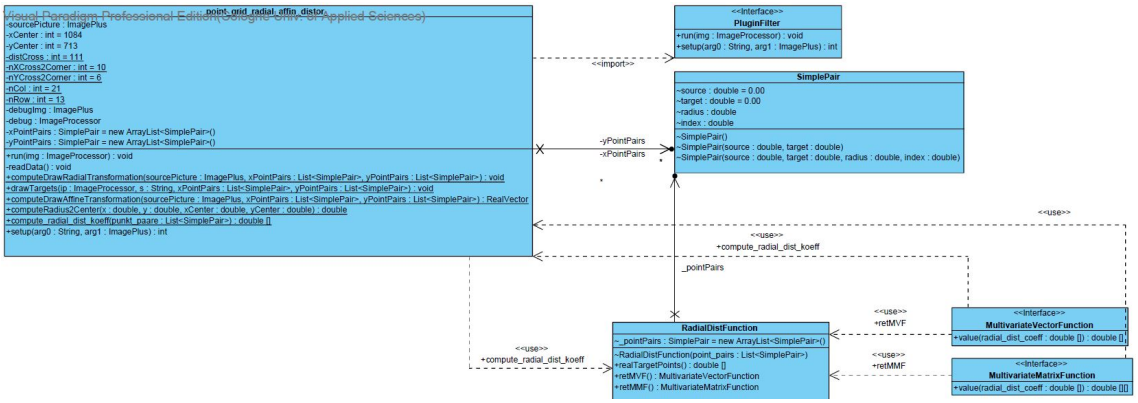

Abbildung 1: UML Klassendiagramm

UnwrapJ ist ein für ImageJ entwickeltes Plugin, das die elastische Registrierung von zwei Bildern ermöglicht, indem es ein Quellbild verformt, so dass es einem Zielbild ähnelt. Es stehen drei Betriebsarten zur Verfügung:

1. ein vollautomatischer Modus;
2. ein vollständig interaktiver Modus, bei dem die Verformung durch die Position einer beliebigen Anzahl von Landmarken eindeutig bestimmt ist;
3. ein gemischter Modus, bei dem interaktive Landmarken nur verwendet werden, um eine ansonsten automatische Registrierungsprozedur anzuzeigen.

Das Deformationsmodell besteht aus kubischen Splines, die Glätte und Vielseitigkeit gewährleisten. Das Registrierungskriterium enthält einen Vektor-Spline-Regularisierungstermin, um die Deformation physisch realistisch zu beschränken.[2]

In dieser Anwendung wird es jedoch nicht zur Registrierung sondern nur zum erzeugen von Landmarken genutzt. Diese werden in eine Textdatei gespeichert, welche im programmierten Plug-In eingelesen und verwendet wird.

5 Methode

5.0.1 Levenberg-Marquard-Approximation

Vera

5.0.2 Radial Distance Function (RDF)

Vera

5.0.3 Kamerakalibrierung

Artjom

6 Auswertung

XXXX

7 Reflexion

Vera

8 Zusammenfassung

Vera

Literatur

[1] Tony J Collins. Imagej for microscopy. *BioTechniques*, 43(1 Suppl):25—30, July 2007.

[2] Biomedical Imaging Group. Unwarpj: An imagej plugin that performs a spline-based elastic registration of two images. http://bigwww.epfl.ch/thevenaz/UnwarpJ/. Besucht: 19. September 2017.

[3] Jerome Mutterer and Wayne Rasband. Imagej macro language programmer's reference guide v1.46d. https://imagej.nih.gov/ij/docs/macro_reference_guide.pdf. Besucht: 19. September 2017.