Wilhelm Büchner Hochschule
Darmstadt
Fachbereich Informatik

Wilhelm Büchner Hochschule
Private Fernhochschule Darmstadt

Diplomarbeit

Arbeitstitel:

Erkennung von nix im Nullraum

Betreuer:	Dr.-Dr. Mustermann
Eingereicht am:	18. März 2015
Eingereicht von:	MUSTERMANN
Matrikel-Nummer:	666
Anschrift:	MUSTERWEG 6, 19061 MUSTERSTADT
Studiengang:	Informatik

MUSTERMANN

Erkennung von nix im Nullraum

Diplomarbeit
Studiengang Informatik
Wilhelm Büchner Hochschule Darmstadt

Bearbeitungszeitraum: 01. Oktober 2014 - 18. März 2015

Abstract

...

Inhaltsverzeichnis

Abkürzungsverzeichnis

GCC GNU Compiler Collection, Sammlung von OpenSource Compiler Programmen für Programmiersprachen wie C, C++, Java u.v.m.

Qt Plattformunabhängiges SDK für C++ basierte Anwendungen

1 Einleitung

...

2 Problemstellung

...

2.1 Ist-Zustand

...

2.2 Zielsetzung

...

2.2.1 Zielgruppe

...

2.2.2 Abgrenzung

...

2.3 Technische Produktumgebung

...

2.3.1 Software

Die Softwareentwicklung und Ausführung der Bildverarbeitungsapplikationen erfolgt bei ASinteg auf Linux-basierten Systemen. Hierfür wird als Basis ein OPENSUSE 12.3 Betriebssystem mit Kernel 3.4 im 64 Bit-Modus eingesetzt. Als Entwicklungsumgebung kommt das C++ Toolkit QT 5.3 mit dem GCC 4.7 Compiler zum Einsatz. Dabei liefert QT neben den plattformunabhängigen C++ Bibliotheken für Datei-, Netzwerk- und

GUI-Funktionalität auch eine plattformunabhängig verfügbare grafische Entwicklungsumgebung (IDE) namens QT CREATOR.

...

Für die Produktionsumgebung sind die Plattformen openSUSE Linux ab Version 12.3 (64 Bit) sowie Windows ab Version 7 SP1 (64 Bit) maßgeblich. Nur Softwareprojekte, die auf diesen Plattformen uneingeschränkt funktional sind, werden in realen Projekten der ASinteg eingesetzt.

...

2.3.2 Hardware

Neben der Festlegung der Softwareumgebung ist insbesondere die Spezifikation der Mindestanforderungen der Hardwareumgebung wesentlich für die Beurteilung der Leistungsfähigkeit und den Einsatz in Produktionsumgebungen.

2.3.2.1 Entwicklungssystem

...

Für die Entwicklung und Evaluierung der Softwarelösung wird ein ASinteg Entwicklungssystem mit folgenden Ausstattungsmerkmalen eingesetzt (es werden nur für die Bildverarbeitung relevante Merkmale aufgeführt):

Komponente	Beschreibung
Gerätebezeichnung	DELL Latitude E6520
Prozessor	Intel Core i7-2760QM 2.4 GHz
CPU-Kerne	4 physische, 4 logische (HyperThreading)
CPU-Cache	6 MB L3-Cache
Arbeitsspeicher	4 GB DDR3
Netzwerkanschluss	Ethernet 1000Base-T mit Unterstützung für Jumbo Frames[1]
Peripherie	3x USB 2.0

Tabelle 2.3.1: Hardwareausstattung Entwicklungssystem

Die *Gigabit Ethernet*-Schnittstelle ist Grundvoraussetzung für den Einsatz moderner industrieller Kamerasysteme, um die großen Datenmengen bei kontinuierlicher Aufnahme verarbeiten zu können.

...

Trotz der vorgenannten Hardwareanforderungen zum Entwicklungssystem soll die zu entwickelnde Lösung prinzipiell auf jedem x86 kompatiblen PC-System ausführbar sein.

...

2.3.2.2 Bildaufnahmesysteme

...

3 Grundlagen

...

3.1 Verarbeitungskette

...

3.2 Bildaufnahme

...

3.2.1 Beleuchtung

...

3.2.1.1 LED-Beleuchtung

...

3.2.1.2 Beleuchtungswinkel

Hellfeld-Beleuchtung ...

Dunkelfeld-Beleuchtung ...

...

3.2.2 Optik

...

3.2.2.1 Abbildungsgesetze

...

3.2.2.2 Schärfentiefe

...

3.2.2.3 Verzeichnungen

...

3.2.2.4 Objektivarten

...

3.2.3 Kamera

...

3.2.3.1 CCD-Sensoren

...

3.2.3.2 CMOS-Sensoren

...

3.2.3.3 Zeilenkameras

-> GigE Vision Standard

-> eindimensionale vs. zweidimensionale Kameras

[[SKIZZE: Matrix + Zeilenkamera]] ...evtl STEMMER S.224 oder Schematisch Matrix + 1-Zeile + 3-Zeilen

Bild 3.2.1: Schematischer Aufbau Matrix- und Zeilenkameras

...

3.3 Bildvorverarbeitung

Die Hauptaufgaben der Bildvorverarbeitung sind:

- Korrektur von Sensoreinflüssen und nichtlinearen Charakteristiken (Rauschen)
- Optimierung von Helligkeit und Kontrast
- geometrische Entzerrung zum Eliminieren von Objektiveinflüssen

...

3.4 Bildauswertung

...

3.5 Entwicklungsumgebung

In den folgenden Abschnitten wird die Entwicklungsumgebung beschrieben, die für die Durchführung der Evaluierung der Lösungswege und Implementierung der Lösung eingesetzt wird. Da sämtliche Softwareprojekte bei ASinteg in der Programmiersprache C++ umgesetzt werden, beginnt die Beschreibung mit dem bei ASinteg breit eingesetzten C++-Framework QT.

3.5.1 C++-Framework Qt

Das QT Framework ist ein weit verbreitetes C++-Framework zur Entwicklung von plattformunabhängigen Anwendungen. Ursprünglich von der norwegischen Firma TROLLTECH für die Entwicklung von grafischen Oberflächen ins Leben gerufen, hat sich QT zu einem auf Servern, Desktops, Mobilgeräten, Embedded- und Automotive-Systemen verbreiteten Framework entwickelt. Seine weite Verbreitung begann mit dem Linux Fenstermanager KDE, mittlerweile nutzen Unternehmen wie Google, Adobe und Samsung das Framework ebenfalls für ihre Softwareentwicklung.

...

3.5.2 ASinteg ImageManager Framework

Der ASINTEG IMAGEMANAGER ist ein C++-Framework zur Erstellung komplexer Applikationen im Bereich industrieller Bildverarbeitung. Das Framework basiert dabei auf QT und richtet sich in puncto Schnittstellen und Handhabung streng nach den Vorgaben des C++-Toolkits.

Das IMAGEMANAGER Framework ist hoch modular aufgebaut und nutzt ähnlich Qt eine logische wie physische Unterteilung in Module wie z.B. Core, Network und Userinterface. Hierzu sind alle Schnittstellen möglichst generisch[1] konzipiert und direkte Abhängigkeiten zwischen Modulen auf ein Minimum beschränkt.

So nutzen alle Module das Qt Property-Framework (dynamische Properties) mit seinen generischen Schnittstellen die für Konfiguration und die Persistenz.

...

[1] „generisch“ steht in diesem Zusammenhang für Datentyp-unabhängig

4 Lösungsansatz

...

4.1 Evaluierungsumgebung

4.1.1 Ablauf

...Ziel und Ablauf der Validierung

...

4.1.2 Parameter

Aus den Erfahrungen mit den bestehenden Separierungsverfahren und den Anforderungen aus der Aufgabenstellung wurden die für einen objektiven Vergleich notwendigen Kriterien einer Validierung ermittelt.

Die Validierung erfolgt auf einheitlicher Hard- und Software für die folgenden Validierungsparameter:

- Performanz (Verarbeitungszeit in Millisekunden)
- Störungsunempfindlichkeit (Fehlerrate bzgl. nicht oder falsch verarbeitete Bilder)
- Qualität (prozentuale Größenabweichung der Ergebnisbilder)

...

4.1.3 Testsoftware

-> Programmablaufplan

-> Qt Unit-Test Framework „QTestLib“

...

4.2 Evaluierung von Bildverarbeitungsverfahren

...

4.3 Evaluierung von Algorithmen und Toolkits

...Validierung verschiedener Implementationen für folgende Validierungsparameter:

- Performanz
- Zielplattformen
- Kosten

Im Zuge der Validierung werden folgende Implementierungen untersucht:

- C++ Algorithmus
- ...

...

4.3.1 OpenCV Bibliothek

Die OPEN SOURCE COMPUTING VISION LIBRARY, kurz OPENCV, ist eine weit verbreitete Programmbibliothek für die Implementierung von Anwendungen zur Bildverarbeitung und für maschinelles Sehen. Ursprünglich von Intel im Jahre 1998 initiiert, wird OpenCV heute unter der freien BSD-Lizenz durch die Unternehmen Willow Garage und Itseez sowie einer großen Gemeinschaft von ehrenamtlichen Entwicklern gepflegt. Ziel von OpenCV ist es, Bildverarbeitung...

[[GRAFIK Struktur OpenCV Lib]]

...

4.3.2 MVTec HALCON Toolkit

Das weltweit erfolgreiche und besonders in der Industrie weit verbreitete Bildverarbeitungs-Toolkit HALCON des Münchener Unternehmens MVTEC kann auf eine Historie von mehr als einem Jahrzehnt verweisen. Die MVTEC SOFTWARE GMBH wurde 1996 durch Dr. Eckstein und Dr. Munkelt als Spin-off der Technischen Universität München und dem Bayerischen Forschungszentrum für wissensbasierte Systeme (FORWISS) gegründet.

HALCON ist eine in C und C++ entwickelte universelle Software für industrielle Bildverarbeitung, welche vorwiegend für den Einsatz durch Systemintegratoren und Entwicklern von Bildverarbeitungs-, optischen Inspektions- und Qualitätskontrollsystemen konzipiert wurde. Die Stärke von HALCON ist die hochperformante Implementierung in C/C++, die im Vergleich zu Wettbewerbslösungen große Anzahl verfügbarer Bildverarbeitungsalgorithmen, die Unterstützung vielfältiger Framegrabber und Kamerasysteme sowie die grafische Entwicklung mit Quelltextgenerierung.

Mittels der grafischen Entwicklungsumgebung HDEVELOPER können komplexe Bildverarbeitungsanwendungen durch Assistenten und mittels einer Skriptsprache erstellt werden. Dabei unterstützen grafische Analysewerkzeuge die Einbindung und Parametrierung von Bildverarbeitungsalgorithmen, bei HALCON „Operatoren“ genannt. Nach Erstellung der Skripte können diese in verschiedene Programmiersprachen übersetzt oder mittels der HALCON C/C++ Bibliothek zur Laufzeit interpretiert werden.

Die HALCON Bibliotheken und die HALCON Laufzeitumgebung steht dabei für verschiedene Plattformen und Prozessorarchitekturen bereit, was den Einsatz insbesondere in industriellen Embedded-Systemen ermöglicht, z.B. zur Entwicklung intelligenter Kamerasysteme und Robotersteuerungen.

Neben dem hohen Einsatzspektrum und der im Wettbewerb sehr hohen Performanz zeichnet sich HALCON allerdings auch durch sein vergleichsweise hohes Kostenniveau aus. Dabei sind neben den Kosten für Entwicklerlizenzen auch Laufzeitlizenzen pro Installation zu entrichten, welche nach Funktionsumfang gestaffelt sind. Im Gegensatz zu anderen Herstellern gewährt MVTEC dabei kaum Preisnachlässe und vertreibt die Software nur über ein Netzwerk zertifizierter Partner, was einen Preiswettbewerb wirksam unterminiert.

Basierend auf HALCON entwickelt ASinteg seit 9 Jahren u.a. internationale OCR-basierte Adresslesesysteme, komplexe Identifikations- und Inspektionslösungen im Logistikbereich sowie Systeme zur Qualitätskontrolle in der Pharma- und Lebensmittelindustrie. Hierfür wird auf die C++ API von HALCON aufgesetzt, welche für die zeitkritischen und komplexen Bildverarbeitungsaufgaben die effizienteste Implementierung darstellt. Der C++ Quelltext wird dabei aus im HDeveloper erstellten Projekten erzeugt, was die Wartbarkeit und die Fehlerdiagnose mittels der grafischen Entwicklungsumgebung erheblich vereinfacht.

Aufgrund der Performanzvorteile und der seitens MVTec verwendeten Prozessor-spezifischen Optimierungen kommen bei ASinteg ausschließlich Systeme mit Intel Prozessoren und überwiegend Linux-Betriebssysteme zum Einsatz. Für reine Desktop-Lösungen

wird zudem Windows als Betriebssystem eingesetzt, um die Integration in die kundenseitige IT-Landschaft zu ermöglichen.

[[GRAFIK Halcon + Extensions]]

...

4.3.3 C++ Algorithmus

...

4.4 Lösungsentscheidung

...

Die folgende Tabelle 4.4.1 stellt das eingesetzte Kennzahlenschema und dessen Gewichtungen dar.

Validierungsparameter	Einheit	Gewichtungsfaktor
minimale Ausführungszeit	Millisekunden	1,0
maximale Ausführungszeit	Millisekunden	1,5
durchschnittliche Ausführungszeit	Millisekunden	1,5
nicht separierte Bilder	Anzahl	1,3
zu stark beschnittene Bilder	Anzahl	1,0
gemittelte Gut-Rate	Prozent	1,4
durchschnittliche Größenabweichung	Prozent	1,4

Tabelle 4.4.1: Kennzahlenschema für Validierungsparameter

Für die Bestimmung der einzelnen Gewichtungsfaktoren wurde der Kunde (die ASinteg GmbH) nach der Bedeutung und den Auswirkungen der einzelnen Parameter auf die Produktion befragt.

Die Testergebnisse sind in der Tabelle A.0.1 aufgeführt, welche dem Anhang entnommen werden kann.

...

5 Lösungsentwurf

...Konzeption und Feinplanung der Software mittels Methoden des Objektorientierten Design (OOD) in

...Einbindung in das ASinteg ImageManager Framework und *Qt*

...

5.1 Planung und Konzeption

...Planung und Realisierung von Softwareprojekten und -komponenten bei ASinteg GmbH nach festgelegten Planungsphasen...

...

5.1.1 Zeitplanung

... Gantt-Diagramm aus Redmine

5.1.2 Business-Case

... Use-Case-Diagramm

5.1.3 Ablaufplan

... Sequenz-Diagramm

5.1.4 Klassenstruktur

... Klassendiagramm

5.2 Systementwurf

... Bibliothek + Klassen + UnitTest + ConsoleApp / GUI-App für Parametrierung

6 Implementierung

6.1 Bildaufnahme

...

6.2 Filterung

TODO: u.a. Kontrastspreizung und Mean-Filter

...

6.3 Separierung

...

7 Nachweis der Funktionalität

Im vorangegangenen Kapitel wurde die Implementierung der Lösung gemäß der Test-Driven-Development Methode beschrieben, deren Umsetzung auf umfangreiche Testfunktionen beruht. Nachdem bereits Implementierungsdetails erläutert wurden, widmet sich dieses Kapitel der Beschreibung der Testverfahren, Testdaten und den Testaufbauten.

...

7.1 Testverfahren und -umgebung

TODO: Beschreibung QS-Verfahren, wie z.B. UnitTests, indiziertes Testdeck, Benchmarkumgebung/-Tools

-> Positiv-/Negativ-Tests mit synthetischen und realen Daten/Bildern

...

7.1.1 Testdaten

Wie im Kapitel Evaluierung beschrieben, wird für alle Tests und Validierungen ein einheitlicher Satz von Bilddateien verwendet.

-> Format Bilddaten

-> Verzeichnisstruktur

-> Dateiformat Indizierung + Testergebnisse

-> Tabelle mit Testdaten (Kamera, System, Paket/Brief, Anzahl)

7.1.2 Unit-Tests

...

7.1.3 Benchmarks

Für die Testung der Verfahren auf ihre Performanz erfolgen sog. Benchmark-Tests. Diese dienen zur Ermittlung nachvollziehbarer Kennzahlen der Ausführungsgeschwindigkeit von Funktionen.

...

Während der Testläufe dürfen keine weiteren Anwendungen ausgeführt werden, die durch ihre Prozessorlast die Zeitmessungen verfälschen könnten. Auch sind die Tests nur auf dem spezifizierten Testsystem auszuführen, um eine vergleichbare Datenbasis zu gewährleisten.

7.2 Testergebnisse

TODO: Tabelle mit empirischen Daten der Benchmarks und UnitTests

...

Die Testergebnisse gliedern sich dabei gemäß den Validierungsparametern wie folgt:

- Performanz
  - minimale Ausführungszeit in Millisekunden
  - maximale Ausführungszeit in Millisekunden
  - durchschnittliche Ausführungszeit in Millisekunden
- Störungsunempfindlichkeit
  - nicht separierte Bilder
  - zu stark separierte Bilder
- Qualität
  - prozentuale Größenabweichung der Ergebnisbilder vom indizierten Soll-Wert

...

	Schwellwert-Cropper	**Neuer Cropper**
minimale Ausführungszeit (ms)		
maximale Ausführungszeit (ms)		
durchschnittliche Ausführungszeit (m)		
nicht separierte Bilder		
zu stark beschnittene Bilder		
gemittelte Gut-Rate (%)		
durchschnittliche Größenabweichung (%)		

Tabelle 7.2.1: Testergebnisse finale Lösung

8 Bewertung und Ausblick

-> Schwierigkeiten

-> Änderungen an Verfahren + Konzept

-> Erfahrungen aus der Arbeit

-> Auswirkungen auf ASinteg

-> Einsatz der Lösung

-> Vorteile

CameraLink

...

Glossar

CameraLink Schnittstellenstandard für industrielle Bildverarbeitungskomponenten über spezielle Kabel und Protokolle für hohe Datenraten.

Gigabit Ethernet Standard für die Kommunikation bei Übertragungsraten von bis zu 1.000 Megabit/s in Ethernet-kompatiblen Netzwerken.

Qt Plattformunabhängiges SDK für C++ basierte Anwendungen.

Literaturverzeichnis

[BEE+12] BAGGIO, Daniel L. ; EMAMI, Shervin ; ESCRIVÁ¡, David M. ; IEVGEN, Khvedchenia ; MAHMOOD, Naureen ; SARAGIH, Jason ; SHILKROT, Roy: *Mastering OpenCV with Practical Computer Vision Projects*. Birmingham : Packt Publishing Ltd, 2012. – ISBN 978–1–849–51783–6

[BK08] BRADSKI, Gary ; KAEHLER, Adrian: *Learning OpenCV - Computer Vision with the OpenCV Library*. Sebastopol : Ö'Reilly Media, Inc.", 2008. – ISBN 978–0–596–55404–0

[DG13] DEITELHOFF, Fabian ; GEISLER, Christof: *Erkennen und Auswerten von rotierten Datamatrix-Codes mit Perspektive*. München : GRIN Verlag GmbH, 2013. – ISBN 978–3–656–37353–7

[Jäh05] JÄHNE, Bernd: *Digitale Bildverarbeitung*. 6. Auflage. Berlin, Heidelberg : Springer, 2005. – ISBN 978–3–540–24999–3

[Lag14] LAGANIÈRE, Robert: *OpenCV Computer Vision Application Programming Cookbook, 2nd Edition*. 2nd New edition. Birmingham : Packt Publishing, Limited, 2014. – ISBN 978–1–782–16148–6

[Ste08] STEINMULLER, Johannes: *Bildanalyse - Von der Bildverarbeitung zur räumlichen Interpretation von Bildern*. 2008. Auflage. Wiesbaden : Springer Berlin Heidelberg, 2008. – ISBN 978–3–540–79742–5

[STE13] STEMMER IMAGING GMBH: *Das Handbuch der Bildverarbeitung*. Puchheim : STEMMER IMAGING GmbH, 2013. – ISBN 978–3–00–039674–8

[SUW08] STEGER, Carsten ; ULRICH, Markus ; WIEDEMANN, Christian: *Machine Vision Algorithms and Applications*. 1. Auflage. New York : Wiley, 2008. – ISBN 978–3–527–40734–7

[Wik08] WIKIPEDIA: *BibTeX — Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=BibTeX&oldid=44468732. Version: 2008, Abruf: 6. April 2008

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang

Tabellen

Validierungsparameter	Referenz	Kontrast	Median	Schärfung	Schwellwertseg.	Regionenseg.	Kantenseg.
min. Ausführungszeit (ms)	1.0						
max. Ausführungszeit (ms)	7.0						
durchschn. Ausführungszeit (ms)	4.0						
nicht separiert	1						
zu stark beschnitten	2						
gemittelte Gut-Rate (%)	99.0						
durchschn. Größenabweichung (%)	10.0						

Tabelle A.0.1: Kennzahlenschema für Validierungsparameter

Bilder

....
....
....
....

Bild A.0.1: Bild xy

Programm-Listings

Eine Möglichkeit Quelltext aufzulisten.

```
#include <stdio.h>

int main(void)
{
    printf("Hello World"); // print message

    return 0;

}
```

Listing A.1: Ein 'Hallo Welt' Programm

```
// Noch eine Möglichkeit Quelltext einzufügen - hier mit Bunti Klicki:
// "Einfügen" -> "Programmlisting", dann in letzte Listing-Zeile gehen und "
     Einfügen" -> "Legende"

cv::Mat src, dest;
src = cv::imread("test.png");
cv::meanFilter(src, dest, 5);
src = dest;
cv::threshold(src, dest, 0, 140, cv::THRESH_BINARY);
cv::imshow("cropped image", dest);
```

Listing A.2: Mean-Filter und Schwellwert-Segmentierer in OpenCV

Eidesstattliche Erklärung

MUSTERMANN, Matrikelnummer 666

Hiermit erkläre ich, dass ich diese Arbeit selbständig abgefasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Abgabedatum Unterschrift (Vor- und Zuname)