

INTO THE ROS

ADVANCED ROS NETWORK INTROSPECTION

Praxis der Softwareentwicklung
Sommersemester 2014

P f l i c h t e n h e f t



Auftraggeber

KIT - Karlsruher Institut für Technologie
Fakultät für Informatik
Institut für Anthropomatik und Robotik (IAR)
Intelligente Prozessautomation und Robotik (IPR)

Betreuer: Andreas Bihlmaier
andreas.bihlmaier@gmx.net

Auftragnehmer

Name	E-Mail-Adresse
Alex Weber	alex.weber3@gmx.net
Matthias Hadlich	matthiashadlich@yahoo.de
Matthias Klatte	matthias.klatte@go4more.de
Micha Wetzel	micha.wetzel@student.kit.edu
Sebastian Kneipp	sebastian.kneipp@gmx.net

Inhaltsverzeichnis

1	Produktfunktionen	4
2	Produkteinsatz	5
2.1	Anwendungsbereiche	5
2.1.1	Zielgruppen	5
2.1.2	Betriebsbedingungen	5
3	Produktumgebung	6
3.1	Software	6
3.2	Hardware	6
3.3	Orgware	6
3.4	Produktschnittstellen	6
4	Funktionale Anforderungen	7
4.1	API <i>Pflicht</i>	7
4.1.1	Allgemein	7
4.1.2	Unterstützung folgender Operationen von OpenCV	7
4.2	API <i>Optional</i>	8
4.2.1	Allgemein	8
4.2.2	Unterstützung folgender Operationen	8
4.3	GUI <i>Pflicht</i>	8
4.4	GUI <i>Optional</i>	8
4.5	Mögliche Visualisierungen	9
4.5.1	Allgemein	9
4.5.2	Visualisierungen von Matches	9
4.5.3	Visualisierungen für Filter	10
5	Nichtfunktionale Anforderungen	11
5.1	Produktleistungen <i>Pflicht</i>	11
5.2	Produktleistungen <i>Optional</i>	11
5.3	Qualitätsanforderungen <i>Pflicht</i>	11
5.4	Qualitätsanforderungen <i>Optional</i>	11
6	Produktdaten	12

7 Systemmodell	13
8 Benutzeroberfläche	14
8.1 Skizzen bzw. Bilder	15
8.1.1 name des bildes	15
8.1.2 name des bildes	16
9 Qualitätsanforderungen	17
10 Abgrenzungskriterien	18
11 Glossar	19
11.1 Allgemein	19
11.2 OpenCV	21
12 Literatur	23

1 Produktfunktionen

Produktfunktionen hier

also was kann unser produkt kleine beschreibung

2 Produkteinsatz

Produkteinsatz
wie der name schon sagt wo es eingesetzt werden soll ;)

2.1 Anwendungsbereiche

ob wir die subsections brauchen weiß ich nicht hab sie mal drinn gelassen

2.1.1 Zielgruppen

same here

2.1.2 Betriebsbedingungen

same here

3 Produktumgebung

Produktumgebung selbe wie davor kp ob wir die subsections brauchen ich hab sie mal drinn gelassen

ja der name sagt ja auch schon alles

3.1 Software

..

3.2 Hardware

..

3.3 Orgware

..

3.4 Produktschnittstellen

..

4 Funktionale Anforderungen

das sind die von der letzten gruppe hab den inhalt mal drinn gelassen damit wir unser zeug einfach nur einfügen müssen

Erklärungen zu den OpenCV spezifischen Begriffen werden im Glossar gegeben.

4.1 API *Pflicht*

4.1.1 Allgemein

/FA0100/ Globale Auswahl zwischen Debug- und Release-Modus

/FA0200/ Auswahl der Visualisierung für jeden Operationstyp

4.1.2 Unterstützung folgender Operationen von OpenCV

/FA0210/ dilate

/FA0220/ erode

/FA0230/ morphologyEx

/FA0240/ Sobel

/FA0250/ threshold

/FA0260/ adaptiveThreshold

/FA0270/ floodFill

/FA0280/ KeyPoint

/FA0290/ DMatch

4.2 API *Optional*

4.2.1 Allgemein

/FA0300/ Optionale Parameter für Einstellungen der Visualisierungen

/FA0400/ Lokale Auswahl Debug/Release-Modus

/FA0500/ Optionale nicht-blockierende Aufrufe für Streaming

4.2.2 Unterstützung folgender Operationen

/FA0600/ stitching

/FA0610/ ocl (OpenCL)

/FA0620/ calcHist

/FA0630/ Canny

/FA0640/ HoughCircles

4.3 GUI *Pflicht*

/FA0700/ Eine Visualisierung pro oben gelisteter Operation (siehe API Kriterien)

/FA0800/ Drei Visualisierungen für features2d/DMatch (drei der möglichen Wunsch-Visualisierungen, siehe unten)

/FA0900/ Zoomfunktion

4.4 GUI *Optional*

/FA1000/ Permanente GUI mit Historie

/FA1200/ Möglichkeit eine Filteroperation mit geänderten Parametern erneut anzuwenden

/FA1300/ Hohe Zoomstufen mit Zusatzinformationen (z.B. Pixelwerte)

/FA1400/ Optionale Ausnutzung von mehreren Bildschirmen durch Fenstermodus

/FA1500/ Interaktive Überlagerung der Bilder durch Zusatzinformationen (Mouse over)

/FA1600/ Flexibler Umgang mit unterschiedlichen Bildschirm- und Bildauflösungen

/FA1700/ Suchleiste für alle Tabellen (z.B. jener der Übersichtsseite oder der Rohdatendatenanzeige)

/FA1710/ Spezielle Syntax zum Beispiel zur Gruppierung von Datensätzen

4.5 Mögliche Visualisierungen

4.5.1 Allgemein

/FA1800/ Darstellung von Rohdaten

/FA1810/ Abmessungen der Bilder

/FA1820/ Farbraum der Bilder (der in OpenCV genutzte Datentyp)

/FA1830/ Tabellarische Darstellung, z.B. der Matches, mit Filtermöglichkeit

/FA1840/ Diagramme (wie Histogramme)

/FA1900/ Darstellung der Bilder nebeneinander

4.5.2 Visualisierungen von Matches

/FA2010/ Einzeichnen der Keypoints in die Bilder

/FA2020/ Verbinden der Matches durch Linien oder Pfeile

/FA2030/ Einfärben der Linien, Pfeile oder Punkte mit Falschfarben

/FA2040/ Ausblenden der Keypoints ohne Matches

/FA2050/ Auswahl von Matches anhand bestimmter Kriterien (z.B. via Histogramm)

/FA2060/ Manuelle Auswahl von Matches

/FA2070/ Automatische Zusammenfassung von Matches zu Gruppen

/FA2110/ Einzeichnen von Linien / Formen

/FA2120/ Auswählen von zugehörigen Matches

/FA2130/ Die Linien / Formen werden auf das zweite Bild projiziert

/FA2140/ Automatische Gruppierung der Matches zu Flächen

/FA2210/ Pfeillänge und Richtung entsprechen der jeweiligen Translation

/FA2310/ Pixelfarbwerte entsprechen den jeweiligen Tiefenwerten

4.5.3 Visualisierungen für Filter

/FA2400/ Differenzbilder

/FA2500/ Überlagerungen

/FA2600/ Direkte Anwendung von Filtern auf ein oder zwei Bilder Beispiel: Anwendung eines Kantenfilters um die Auswirkungen z.B. einer Kantenglättung zu visualisieren

/FA2700/ Visualisierung über die Auswirkungen auf bestimmte Bildmetriken Beispiel: Überlagerung von Histogrammen beider Bilder oder Vergleich der Kontrastwerte von bestimmten Bildbereichen

5 Nichtfunktionale Anforderungen

5.1 Produktleistungen *Pflicht*

/NF0100/ Die GUI soll schnell starten und interaktiv bedienbar sein

/NF0200/ Möglichst kein Overhead im Release-Modus

5.2 Produktleistungen *Optional*

/NF0300/ Flexibler Umgang mit unterschiedlichen Bildschirm- und Bildauflösungen

/NF0400/ Integration in das OpenCV Test Framework

5.3 Qualitätsanforderungen *Pflicht*

/NF0500/ Keine signifikanten Speicherlecks

/NF0600/ Erweiterbarkeit um zusätzliche OpenCV Operationen und Visualisierungen

/NF0700/ Modularer Aufbau (API und GUI)

/NF0900/ Einhaltung der OpenCV und Qt Konventionen

/NF1000/ Ausführliche Dokumentation der API und des GUI

5.4 Qualitätsanforderungen *Optional*

/NF1100/ Dokumentation des internen Codes mit Werkzeug

/NF1200/ OpenCV geeigneter Aufbau des Build-Systems

/NF1300/ Abdeckung durch Tests

/NF1400/ Keine Resource-Leaks

/NF1500/ Threadsafety *Im C++11-Modus*

/NF1600/ Toleranz gegenüber fehlerhaften API-Aufrufen

/NF1700/ Kein undefiniertes Verhalten

6 Produktdaten

hier bin ich mir nicht ganz so sicher was genau rein muss

7 Systemmodell

das selbe wie im kapitel davor nicht sicher was wir da genau reinschreiben sollen

8 Benutzeroberfläche

hier kommen dann ein paar bilder und dazugehörige beschreibungenen unserer gui

- blasdgs
- sdfg
- was man halt alles so allgemeines reinschreiben kann

8.1 Skizzen bzw. Bilder

8.1.1 name des bildes



Abbildung 8.1: bildunterschrift

Ein ganz ganz langer text zum bild
kdfnakgflkadshfkjsadfköadshfölkdf
hsgkjöldhflkghsfdlkghödklfhglkdsjglödshgkdsfhögh
sdfkengkjhdskföghadfsopfghaoifhaigtfuiaerhgiuag

8.1.2 name des bildes



Abbildung 8.2: bildunterschrift

Ein ganz ganz langer text zum bild
kdfnakgflkadshfkjsfkljsadfköadshfölkdf
hsgkjöldhflkghsfdlkghödklfhglkdsjglödshgkdsfhögh
sdflkgkjhdsköghadfsopfghaoifhaigtfuiaerhgiuag

9 Qualitätsanforderungen

Testfälle und testszenarien kommen dann hier rein

am besten dann noch sections einfügen

10 Abgrenzungskriterien

wie der name schon sagt hier dann die abgrenzungskriterien

11 Glossar

des ist jetzt mal der glossar von letztem jahr den können wir ja ganz schnell bearbeiten und unsere sachen einfügen

11.1 Allgemein

API Application Programming Interface. Eine Schnittstelle (s. Interface), über welche andere Programme auf der Quelltextebene auf dahinter verborgene Funktionalität zugreifen können

Augmented Reality „Erweiterung der Realität“ durch einen Computer, etwa bei der Einblendung von Informationen in ein Bild der Umgebung, das auf einem Smartphone angezeigt wird.

Bug Fehlerhaftes Verhalten eines Programmes.

Binärform Hier das Programm in für den Computer direkt verwendbarer Form im Gegensatz zum Quellcode. Anders als dort ist hier kaum sichtbar, wie das Programm genau arbeitet, weshalb bei OpenSource-Projekten gerade der Quelltext offen liegt (vgl. OpenSource).

Datenstrom Daten, die beispielsweise während der Ausführung eines Programms fließen, wobei das Ende dieses Flusses nicht absehbar ist.

Debug-Modus Modus, bei dem zusätzliche Informationen angezeigt werden, um dem Programmierer das Auffinden und Beheben von Bugs (kurz *Debugging* oder *Debuggen*), hier insbesondere Programmierfehlern, zu erleichtern. Vgl. Release-Modus.

Debug-Visualisierung Hier eine Visualisierung, die den Benutzer beim Debuggen unterstützt, indem sie relevante Daten zu den übergebenen Bildern anzeigt und diese damit leicht verständlich darstellt.

FAQ Engl. für „Häufig gestellte Fragen“ enthält sie viele Fragen, die besonders neue Benutzer sich stellen, wenn sie anfangen ein Projekt zu nutzen.

Falschfarben Verwendung von Farben in einem Bild, die sich von den natürlichen, erwarteten Farben stark unterscheiden, um etwa Details hervorzuheben.

Filter In der Bildverarbeitung die Veränderung eines Bildbereiches mithilfe eines bestimmten Algorithmus.

GNU/Linux Das GNU-Betriebssystem in Kombination mit einem Linux-Kern

GNU-Projekt Das Projekt zur Erstellung von GNU-Betriebssystem und Software.

GUI Graphical User Interface, zu deutsch Graphische Benutzeroberfläche. Stellt Funktionen graphisch dar, sodass der Benutzer beispielsweise per Mausklick damit interagieren kann; im Gegensatz zu textbasierten Benutzerschnittstellen (vgl. Interface).

Kompilieren Umwandlung eines Quellcodes in eine für den Computer verständliche Form, mithilfe eines Kompilers genannten Programms.

Matches Durch OpenCV erzeugte Verknüpfungen zwischen zwei Bildbereichen bzw. Bildpixeln, welche vom Benutzer an die API übergeben werden.

Mouse over Information über das Element einer GUI, auf dem der Mauszeiger ruht, wird angezeigt.

Parallelrechner Rechner, in dem mehrere Threads gleichzeitig nebeneinander ausgeführt werden können (vgl. Thread).

OpenCV Test Framework Stellt Funktionen zum Testen zur Verfügung, etwa Überprüfungen, ob zwei Matrizen gleich sind.

Open Source Software, bei welcher der Quellcode frei zugänglich gemacht wird. Dies erlaubt unter anderem die Weiterverwendung und -entwicklung durch andere.

Overhead Zusätzlicher Speicher- oder Zeitaufwand.

proprietär In diesem Zusammenhang Software, die nicht unter einer freien Lizenz steht.

Release-Modus Veröffentlichungs-Modus, in dem keine zusätzlichen Debug-Informationen angezeigt werden, also der Modus, in dem das fertige Programm läuft.

Resource-Leak Das Auftreten der Situation, dass Ressourcen irgendeiner Art (Speicher, Dateien, usw.) zwar alloziert werden, aber nach Verwendung nicht mehr an das System zurückgegeben werden.

Rohdaten Daten, welche direkt und ohne wirkliche Aufarbeitung, aus den vom Entwickler beim API-Aufruf übergebenen Datenstrukturen stammen.

Speicherleck *engl. memory leak* Vgl. Resource-Leak; die Ressource ist in diesem Fall Speicher.

Stand-Alone-Programm Programm, das für sich alleine funktioniert.

Streaming Hier das Weiterlaufen des Datenstroms.

Tab Hier ein registerkartenähnlicher Teil einer GUI.

Thread Ausführungsstrang in einem Programm. Der Begriff wird insbesondere im Zusammenhang mit Mehrfachkernsystemen, wo mehrere Threads gleichzeitig nebeneinander laufen können, oft verwendet.

thread-lokal Auf einen Thread beschränkt (vgl. Thread).

Threadsafety Es ist sichergestellt, dass mehrere Threads sich nicht gegenseitig stören, etwa beim Speichern von Daten.

Translation Unit Eine Einheit, die einzeln im Ganzen kompiliert wird (s. Kompilieren); ein Projekt teilt sich meist in mehrere solcher Einheiten auf.

Undefiniertes Verhalten Befehle, deren Verwendung dazu führt, dass der C++-Standard *keinerlei* Verhaltensgarantien irgendeiner Art für das gesamte Programm mehr gibt. Etwas Umgangssprachlich: Der Standard untersagt die Verwendung.

View Zusammengehörige Visualisierungen eines bestimmten OpenCV-Features (oder einer Featureart).

11.2 OpenCV

Weiterführende Informationen sind auf docs.opencv.org zu finden.

adaptiveThreshold OpenCV-Methode, die mittels eines adaptiven Threshold (s. unten) Graustufenbilder in (u.U. invertierte) Binärbilder umwandeln kann.

calcHist Berechnet ein Histogramm.

Canny Kantenerkennung (mithilfe des Canny86-Algorithmus).

Dilatation Berechnung des Bereiches in einem Bild, der von einer Maske abgedeckt wird, wenn sich deren Bezugspunkt (oft der Mittelpunkt) durch den ganzen zu untersuchenden Bildbereich bewegt.

DMatch Klasse für das Matching (vgl. Matches).

Erosion Prüft, ob eine Maske, etwa eine geometrische Figur, vollständig in ein Bild bzw. einen Bildbereich passt und gibt u.U. ein Bild zurück, in dem nur die überdeckten Teile erhalten sind. Bildet zusammen mit der Dilatation zwei der grundlegenden Bildverarbeitungsmethoden.

floodFill Bei dieser Methode wird ein zusammenhängender Bereich des Bildes mit der übergebenen Farbe ausgefüllt.

HoughCircles Findet Kreise in einem Graustufenbild (unter Benutzung der Hough-Transformation).

KeyPoint Klasse, die Daten eines Punktes (Koordinaten, Nachbarschaftsgröße etc.) enthält.

morphologyEx diese Methode von OpenCV erlaubt fortgeschrittene morphologische Transformationen unter Benutzung und mehrfacher Anwendung von Dilatation und Erosion.

ocl Das OCL-Modul stellt Implementierungen von OpenCV-Funktionalität für Geräte, welche die Open Computing Language (kurz OpenCL), ein Interface über Parallelrechner, benutzen, zur Verfügung.

Sobel-Operator Ein Kantenerkennungs-Filter.

Stitching Zusammenfügen mehrerer Bilder mit zusammenhängenden Bereichen an den Rändern zu einem großen Bild, etwa von Einzelfotografien zu einem Panorama.

threshold Diese Methode eröffnet verschiedene Möglichkeiten, die Elemente eines Arrays auf ein bestimmtes Niveau zu trimmen, auf binäre Werte herunterzubrechen und Ähnliches.

12 Literatur

same here das von letztem jahr

Designed for Use – Create Usable Interfaces for Applications and the Web / Lukas Mathis

Die vorliegenden GUI-Entwürfe sind stark von diesem Buch beeinflusst, da es wichtige Techniken, Methoden und Denkmuster der UI-Entwicklung vermittelt. *Und das in einer für Informatiker verständlichen Sprache.*

OpenCV Dokumentation Eine wichtige Quelle aus offensichtlichen Gründen.

Technisches Schreiben – (nicht nur) für Informatiker / Peter Rechenberg Ein anschaulich und verständlich geschriebenes Buch, das hilft den eigenen Schreibstil zu verbessern. Und damit die Verständlichkeit der geschriebenen Dokumentation (vgl. ??) zu erhöhen.