



ENTWICKLUNG EINER (SMARTPHONE-)APP ZUR EINFACHEN KOMPRESSION VON BILDERN - DOKUMENTATION -

Studienmodul *ICT*
der Hochschule für Telekommunikation
Leipzig

Prüfungsvorleistung - Bilddatenkompression

vorgelegt von

Stefan Czogalla, Maik Lorenz, Jan Suttmöller, Stephan Kaden

11. Dezember 2015

Dozent: Prof. Dr. Tilo Strutz

INHALTSVERZEICHNIS

1	Einführung	2
1.1	Zweck	2
1.2	Hintergründe und Ziele des Projekts	2
1.3	Produktumfang	2
1.4	Musskriterien	2
1.5	Abgrenzungskriterien	2
1.5.1	Kostenrahmen	3
1.6	Definitionen, Akronyme, Abkürzungen	3
2	Allgemeine Übersicht	3
2.1	Beschreibung der Ausgangssituation (Ist-Zustand)	3
2.1.1	Vergleich der vorhandenen drei APP's	3
2.2	Beispielbilder	5
2.2.1	PhotoQuant	5
2.3	Produkteinsatz	5
2.3.1	Anwendungsbereiche	5
2.3.2	Zielgruppen, Qualifikationsniveau	5
2.4	Produktfunktionalität	5
2.5	Randbedingungen	5
2.6	Annahmen und Abhängigkeiten	6
3	Anforderungen	6
3.1	Fachkonzept	6
3.1.1	Verwendete Bibliotheken von Drittanbietern	6
3.2	Anfordeungen für Inbetriebnahme und Einsatz	6
3.2.1	Installationsprozedur	6
3.3	Qualitätsanforderungen	6
3.3.1	Qualitätsmerkmale	6
3.4	Anforderung an die Entwicklung	7
3.4.1	Entwicklungs-Umgebung	7
3.4.2	Änderungsmanagement	7
4	Anhang	7
4.1	PVL Aufgabenbeschreibung	8

1 EINFÜHRUNG

1.1 ZWECK

Dieses Dokument dient als Dokumentation des berufsbegleitenden Studienganges, Kommunikations- und Medieninformatik des Matrikel 13, mit der Programmierung einer APP zur Kompression von Bilddaten. Es setzt dabei die Rahmenbedingungen fest.

1.2 HINTERGRÜNDE UND ZIELE DES PROJEKTS

Die Hochschule für Telekommunikation Leipzig ([HfTL](#)) ist eine private, staatlich anerkannte Fachhochschule. Träger der [HfTL](#) ist die [HfTL-Trägersgesellschaft mbH](#), eine Beteiligungsgesellschaft der Deutschen Telekom AG. Die Hochschule befindet sich im Leipziger Stadtteil Connewitz. Es werden sowohl Direkt- als auch duale Studiengänge und berufsbegleitende Studiengänge angeboten.

1.3 PRODUKTUMFANG

Der Betrieb der Smartphone-APP muss auf allen gängigen Android-Smartphones ab Version 4.4.2 möglich sein.

Durch die APP wird den Studenten der HFTL ermöglicht:

- Fotos aufnehmen
- Bilder mit einer skalaren Quantisierung komprimieren
- Quantisierungsintervall soll einstellbar sein
- Ablage der komprimierten Bilder in passend zum Verfahren benannten Ordnern

1.4 MUSSKRITERIEN

Zunächst müssen zwingend folgende Punkte des Umfangs erfüllt werden:

- Bildaufnahme
- Quantisierung
- Auswahl Quantisierungsintervall
- Abspeichern mit passendem Dateinamen

1.5 ABGRENZUNGSKRITERIEN

Die APP soll später auch um zusätzliche Funktionen, wie Vektorquantisierung oder andere Verfahren erweiterbar sein. Versionen für andere Betriebssysteme müssen in einem separaten Projekt bearbeitet werden und sind nicht Bestandteil dieses Projektes.

1.5.1 KOSTENRAHMEN

Für die Entwicklung der **APP** soll auf kostenfreie Opensource-Programme oder auf vordefinierte Klassen für die Programmierung zurückgegriffen werden.

1.6 DEFINITIONEN, AKRONYME, ABKÜRZUNGEN

HfTL Hochschule für Telekommunikation Leipzig

APP Kurzform für Applikation

mbH mit beschränkter Haftung

XML Extensible Markup Language

GUI Graphical User Interface

QuantiPig quantisiertes Picture

2 ALLGEMEINE ÜBERSICHT

2.1 BESCHREIBUNG DER AUSGANGSSITUATION (IST-ZUSTAND)

Als Ausgangssituation wurden der Studentengruppe drei **APP's** aus vorherigen Matrikeln vorgelegt, die ebenfalls Bilddaten komprimieren. Diese drei **APP's** gilt es zu vergleichen und die Stärken herauszuarbeiten. Anhand dieser Ergebnisse gilt es eine neue **APP** zu entwickeln.

2.1.1 VERGLEICH DER VORHANDENEN DREI **APP's**

Vergleich der drei vorhandenen APP's

	Photo Quant	QuantiPic	QuanPic
Merkmale	drei Modi (Lloyd-Algo, Midtread, Original)	drei Modi (Original, Farbwerte, Helligkeit)	zwei Modi (Original, MedianBlur)
		"Farbwerte stellt das Bild mit den	
		jeweiligen Komplementärfarben dar	
Speicherformat	JPEG	PNG	PNG
Auflösung	2688x1520 px	1456x832 px	1456x832 px
Dateigrößen	Original: 2,13 MB Lloyd: 3,5 MB Midtread: 2,8 MB	Original: 2,24 MB Helligkeit: 1,88 MB Farbwerte: 126 Kbyte	Original: 2,47 MB MedianBlur: 1,94 MB
Vorteile	<ul style="list-style-type: none"> • APP speichert Bilder je nach Verfahren in einem spezifischen Ordner • setzt Zeitstempel bei Bildern 	<ul style="list-style-type: none"> • Livebild bei Originalbild flüssig • Livebild bei "Farbwerte" flüssig • "Verarbeitete" Bilder sind kleiner als das Original (komprimiert) 	<ul style="list-style-type: none"> • hat OpenCV bereits integriert • zeigt fps an • zeigt Auflösung des Bildes live an • zeigt Livebild flüssig an (14fps)
Nachteile	<ul style="list-style-type: none"> • sehr langsame Verarbeitung in Echtzeit • langsame Verarbeitung bei Originalbild • verarbeitete Bilder sind größer als Orig. • bei Midtread wird nur die obere Hälfte des Bildes quantisiert • OpenCV muss nachträglich auf dem Smartphone installiert werden 	<ul style="list-style-type: none"> • OpenCV muss nachträglich installiert werden • Livebild bei "Helligkeit" stockend 	<ul style="list-style-type: none"> • ruckelt stark bei "MedianBlur (2fps)"
Usability	<ul style="list-style-type: none"> • Quantisierung ist über Menü auswählbar • Auslösebutton • Hilfefunktion mit Beschreibung der einzelnen Menüpunkte 	<ul style="list-style-type: none"> • intuitive Bedienung • Verfahren ist über ein Menü auswählbar 	<ul style="list-style-type: none"> • keine intuitive Bedienung • keine Erklärung der Funktionen

2.2 BEISPIELBILDER

2.2.1 PHOTOQUANT



2.3 PRODUKTEINSATZ

2.3.1 ANWENDUNGSBEREICHE

Aktuell soll die APP nur für die Studenten der [HfTL](#) zugänglich sein, welche ein Android-Smartphone besitzen.

2.3.2 ZIELGRUPPEN, QUALIFIKATIONSNIVEAU

Da bei der Nutzergruppe von Studenten mit Erfahrung im Umgang mit solchen APP's ausgegangen werden kann, wird auch die Oberfläche dementsprechend gestaltet.

2.4 PRODUKTFUNKTIONALITÄT

Auszug aus der Aufgabenbeschreibung:

„Auf Basis der Analyse ist eine neue App zu programmieren, welche die positiven Eigenschaften der vorhandenen Apps vereint und als Verarbeitungsfunktion das Bild gleichmäßig mit einem einstellbaren Quantisierungsintervall quantisiert.“

2.5 RANDBEDINGUNGEN

Der zeitliche Rahmen für die Entwicklung und Programmierung dieser APP endet mit der 4. Kalenderwoche 2016.

Durch das Projektteam wird es nach Ende des Projektes keine weitere Softwarebetreuung, Wartung oder der gleichen geben. Es finden ebenfalls keine Schulungen oder Einweisungen statt.

2.6 ANNAHMEN UND ABHÄNGIGKEITEN

Die APP wird für Android-Geräte ab Version 4.0.3 zur Verfügung gestellt. Entsprechend der Vorgaben der Deutschen Telekom AG muss bei der Programmierung der APP, explizit beim Design, auf die Konzernrichtlinien geachtet werden. Es soll zusätzlich auf die Designempfehlungen für Androidgeräte geachtet werden.

3 ANFORDERUNGEN

3.1 FACHKONZEPT

Die [QuantiPig](#)-APP wird in Java programmiert, um durch Verwendung bestehender Klassen die Erweiterbarkeit und Realisierbarkeit zu vereinfachen. Für das Design werden [XML](#)-Stylesheets verwendet.

3.1.1 VERWENDETE BIBLIOTHEKEN VON DRITTANBIETEREN

- OpenCV

3.2 ANFORDEUNGEN FÜR INBETRIEBNAHME UND EINSATZ

3.2.1 INSTALLATIONSPROZEDUR

Die [APP](#) wird als .apk Datei ausgeliefert und kann somit manuell auf Android-Smartphones ab Version 4.4.2 installiert werden. Dabei muss das Installieren von Software mit unbekannter Herkunft erlaubt werden.

3.3 QUALITÄTSANFORDERUNGEN

3.3.1 QUALITÄTSMERKMALE

Folgende Qualitätsansprüche werden gestellt:

- Hohe Zuverlässigkeit der Software
- schnelle und zuverlässige Verarbeitung der gewünschten Daten
- Fehler werden mit einer entsprechenden Fehlermeldung beantwortet
- Intuitiv benutzbar
- Leicht zu warten und zu erweitern
- Vollständige Dokumentation des Projektes

3.4 ANFORDERUNG AN DIE ENTWICKLUNG

3.4.1 ENTWICKLUNGS-UMGEBUNG

Für die Entwicklung wird Android Studio inkl. Gradle in der Version 1.x genutzt. Für die Dokumentation und Projektkoordination wird GitHub verwendet. Die Dokumentation wird mittels \LaTeX erstellt.

3.4.2 ÄNDERUNGSMANAGEMENT

Zur Versionsverwaltung wurde Git eingesetzt. Als Hosting-Anbieter wurde dabei auf GitHub gesetzt, welcher einen kostenfreien Zugang für nicht kommerzielle Projekte bereitstellt. Ein Graphical User Interface ([GUI](#)) oder ein Konsolenprogramm für Windows und Linux übernehmen dabei die Steuerung der Versionsverwaltung. Konflikte in den einzelnen Versionen können nur über die Konsole behoben werden. Auf der Webseite von GitHub können Milestones erstellt werden und an die jeweiligen Mitarbeiter zugeteilt werden. In den Milestones werden einzelne Aufgaben, sogenannte Issues angelegt und wiederum den Bearbeitern zugeordnet, somit ist der Bearbeitungsstand zu jeder Zeit des Projektes ersichtlich und es kann schnell auf sich ergebende Probleme reagiert werden.

4 ANHANG

4.1 PVL AUFGABENBESCHREIBUNG

Hochschule für Telekommunikation Leipzig
Institut für Kommunikationstechnik
Gustav-Freytag-Straße 43-45
04277 Leipzig

Themen zur PVL IKT-KMI-13

Hinweise:

- ein Thema auswählen
- Mitstreiter benennen (inklusive Emailadresse) und deren (Teil-)Aufgabe(n)
- Name des Teams wählen (Ort der Dienststelle, wenn möglich)
 - Emails ohne Angabe „PVL, StudiengangXX, Team NameXX“ werden von mir ignoriert!!
- Max. 3 Teams pro Thema (first come first serve)
- Bearbeitungszeitraum bis 4. KW, Verzögerung führt zu Punktabzug
- Formatierung der Quelltexte, siehe: <http://www1.hft-leipzig.de/ice/Files/c-quell.txt>
- strukturelle und inhaltliche Gestaltung der Dokumentation gemäß
 - <http://www1.hft-leipzig.de/ice/Files/ThesisTemplate.zip>
- Für Programmieraufgaben darf **keine neuere** Version als Visual C++ 2008 Express verwendet werden.
- Genauere Hinweise (+Präzisierung der Aufgabenstellung, Unterlagen, Daten) gibt es nach Wahl eines Themas.

"Modifikation eines Prädiktors (MED) in Anwendung auf Farbbilder"

Es ist eine Software zur Bilddatenkompression vorhanden (TSIPcoder). Über eine grafische Nutzerschnittstelle (GUI) kann Einfluss auf die Verarbeitungskette genommen werden. Soll ein Farbbild komprimiert werden, so werden die RGB-Komponenten typischer Weise in einen anderen Farbraum (YUV) konvertiert. Dadurch werden die drei Komponenten dekorreliert. Trotzdem enthalten die Komponenten Y, U und V noch ähnliche Strukturen, insbesondere hinsichtlich der Richtung der Kanten.

Der Median-Edge-Detection Prädiktor (MED) nutzt Kanteninformation aus, um zwischen drei verschiedene Prädiktoren umzuschalten. In der vorliegenden Version wird das Umschalten für alle drei Komponenten separat durchgeführt. Leider wird die Entscheidung manchmal durch Rauschen im Bild ungünstig beeinflusst.

Aufgabe ist es, eine zweite Version des MED-Prädiktors zu implementieren, bei der die Prädiktorauswahl für U und V auf Basis von Informationen aus Y erfolgt. Wenn die Y-Komponente bereits verarbeitet ist (dem Decoder steht die Information über Y zur Verfügung), dann kann nachträglich geprüft werden, ob für einen konkreten Bildpunkt der ausgewählte Prädiktor der beste war oder einer der beiden anderen evtl. einen kleineren Schätzfehler ergeben hätte. Die beste Variante wird dann auch für U und V eingesetzt.

Für einen vorgegebenen Satz an verschiedenen Bildern ist die neue Variante zu testen und die Ergebnisse mit dem normalen MED-Prädiktor zu vergleichen.

Der Quellcode ist klar zu strukturieren, mit ausreichenden Kommentaren zu versehen und gemäß

den Richtlinien zu formatieren. Variablennamen sollten selbsterklärend sein. Alle Untersuchungen sind schriftlich zu dokumentieren. Neben der schriftlichen Arbeit sind alle Quellen (Programmcode, Texte, Testbilder) und Tools abzugeben, damit eine Reproduktion der Ergebnisse möglich ist.

Teilaufgaben:

- Koordination
- Programmierung
- Dokumentation (Grundlagen, Änderungen am Quellcode, Kompressionsergebnisse)

Max. 4 Personen,

Max 2 Zusatzpunkte für Klausur

"Programmieren einer (Smartphone-)App zur einfachen Kompression von Bildern"

Es existieren drei Android-App, welche ein Bild von der Smartphone-Kamera im Roh-Format aufnehmen, verarbeiten und speichern. Diese drei Programme sind zu vergleichen und Vor- und Nachteile hinsichtlich verschiedener Eigenschaften (z.B. Bedienkomfort, Speicherbedarf, Bildfolgefrequenz, ...) zur ermitteln. Auf Basis der Analyse ist eine neue App zu programmieren, welche die positiven Eigenschaften der vorhandenen Apps vereint und als Verarbeitungsfunktion das Bild gleichmäßig mit einem einstellbaren Quantisierungsintervall quantisiert. Um die Bildfolgefrequenz zu erhöhen ist ggf. die Ausgabe-Bildgröße zu verringern.

Alle Untersuchungen sind schriftlich zu dokumentieren. Neben der schriftlichen Arbeit sind alle Quellen (Programmcode, Texte, Testbilder) und Tools abzugeben, damit eine Reproduktion der Ergebnisse möglich ist.

Teilaufgaben:

- Koordination
- Recherche
- Programmierung
- Dokumentation (Grundlagen, Methode, Änderungen am Quellcode, Kompressionsergebnisse)

Max. 4 Personen

Max. 2 Zusatzpunkte für Klausur

"Analyse Template-Matching-Prädiktion im CoBaLP2-Coder"

CoBaLP2 ist ein Verfahren für die kontextbasierte lineare Prädiktion von Bilddaten auf Basis von Differenzwerten. Die Gewichte für die Berechnung der Schätzwerte werden für automatisch und signalangepasst festgelegte Kontexte sukzessive optimiert. Für manche Kontexte ist die Streuung der Schätzfehler jedoch relativ hoch. Hier kann evtl. eine Template-Matching-Prädiktion erfolgreich sein. Als Parameter sind Template-Größe, Größe des Suchraumes und vor allem die Verknüpfungsmethode von verschiedenen Matches einzustellen. Je nach Bildinhalte

können verschiedene Einstellungen günstig sein

Aufgabe ist es, systematisch zu testen, welche Einstellungen für welches Bild optimal sind. Daraus ist abzuleiten, wie die optimalen Einstellungen automatisch gewählt werden können.

Der Quellcode ist klar zu strukturieren, mit ausreichenden Kommentaren zu versehen (inkl. Tags zum Markieren der Änderungen am originalen Quellcode) und gemäß den Richtlinien zu formatieren. Variablennamen sollten selbsterklärend sein. Alle Untersuchungen sind schriftlich zu dokumentieren. Neben der schriftlichen Arbeit sind alle Quellen (Programmcode, Texte, Testbilder) und Tools abzugeben, damit eine Reproduktion der Ergebnisse möglich ist.

Teilaufgaben:

- Koordination
- Recherche
- Programmierung
- Dokumentation (Grundlagen, Methode, Änderungen am Quellcode, Kompressionsergebnisse)

Max. 4 Personen,

Max. 2 Zusatzpunkte für Klausur

"Restauration von synthetischen Bilddaten nach DCT-basierter JPEG-Kompression durch near-lossless Codierung"

Synthetische Bilddaten werden aus Unkenntnis häufig mit JPEG (DCT-basierter Modus) komprimiert. Dadurch entstehen störende rauschartige Strukturen im Bild, insbesondere an Helligkeits- oder Farbkanten.

Eine vorhandene Software zur verlustlosen Bilddatenkompression (TSIPcoder) soll so modifiziert werden, dass auf Basis einer gewissen Toleranz das Rauschen im Bild reduziert wird. Dadurch wird die Kompression verlustbehaftet. Ein entsprechender Parameter ist in die grafische Benutzeroberfläche (GUI) aufzunehmen.

Der Quellcode ist klar zu strukturieren, mit ausreichenden Kommentaren zu versehen (inkl. Tags zum Markieren der Änderungen am originalen Quellcode) und gemäß den Richtlinien zu formatieren. Variablennamen sollten selbsterklärend sein. Alle Untersuchungen sind schriftlich zu dokumentieren. Neben der schriftlichen Arbeit sind alle Quellen (Programmcode, Texte, Testbilder) und Tools abzugeben, damit eine Reproduktion der Ergebnisse möglich ist.

Teilaufgaben:

- Koordination
- Programmierung
- Dokumentation (Grundlagen, Methode, Änderungen am Quellcode, Kompressionsergebnisse)

Max. 4 Personen,
Max. 2 Zusatzpunkte für Klausur

"Optimierung von Parametern zur Kompression von Farbbildern"

Eine vorhandene Software (TSIP) zur Kompression von Bilddaten analysiert das geladene Bild und wählt günstige Parameter zur Kompression aus. Die Auswahl ist jedoch nicht in jedem Fall optimal.

Für ein gegebenes Set von Bilddaten sind manuell die Einstellungen zu variieren um eine bessere Kompression zu erzielen. Auch das Decodieren der komprimierten Bilder ist zu testen. Das TSIP-Programm kann über ein Batch-File aufgerufen, sodass ein systematischer Test mit einer Vielzahl von verschiedenen Einstellungen erfolgen kann.

Die Zusatzpunkte ergeben sich wie folgt:

- besseres Kompressionsergebnis als mit automatisch gewählten (oder bereits bekannten) Parametern: 0.05 Punkte (pro Team)
- "-" und besseres Ergebnis als alle andere Teams: +0.2 Punkte (pro Team)
- Entdecken eines Bugs: 0.1 Punkte (pro Team)

Alle Untersuchungen sind schriftlich zu dokumentieren. Neben der schriftlichen Arbeit sind alle Quellen (Programmcode, Texte, Testbilder) und Tools abzugeben, damit eine Reproduktion der Ergebnisse möglich ist.

Teilaufgaben:

- Recherche
- Dokumentation (Grundlagen, Änderungen am Quellcode, Kompressionsergebnisse)

Max. 2 Personen, (maximal 3 Teams mit diesem Thema möglich)
Max 1.5 Zusatzpunkte für Klausur

"Automatische Wahl von Parametern zur Kompression von Farbbildern"

Eine vorhandene Software (TSIP) zur Kompression von Bilddaten analysiert das geladene Bild und wählt günstige Parameter zur Kompression aus. Die Auswahl ist jedoch nicht in jedem Fall optimal.

Auf Basis einer vorhandenen Datenbank, welche numerische Eigenschaften von Bildern und die besten Kompressionseinstellungen enthält, sind Zusammenhänge (Korrelationen) zu ermitteln

und die Frage zu beantworten, welche Einstellungen (automatisch) vorgenommen werden müssen, damit die komprimierte Datei möglichst klein ist.

Die herausgefundenen Zusammenhänge sind anhand vorhandener Testbilder zu prüfen.

Alle Untersuchungen sind schriftlich zu dokumentieren. Neben der schriftlichen Arbeit sind alle Quellen (Programmcode, Texte, Testbilder) und Tools abzugeben, damit eine Reproduktion der Ergebnisse möglich ist.

Teilaufgaben:

- Recherche
- Dokumentation (Grundlagen, Änderungen am Quellcode, Kompressionsergebnisse)

Max. 2 Personen, (maximal 3 Teams mit diesem Thema möglich)

Max 1.5 Zusatzpunkte für Klausur