

LEHRSTUHL FÜR RECHNERARCHITEKTUR UND PARALLELE SYSTEME

Praktikum Rechnerarchitektur

Gammakorrektur (A208)

Projektaufgabe – Aufgabenbereich Bildverarbeitung

1 Organisatorisches

Auf den folgenden Seiten finden Sie die Aufgabenstellung zu Ihrer Projektaufgabe für das Praktikum. Die Rahmenbedingungen für die Bearbeitung werden in der Praktikumsordnung festgesetzt, die Sie über die Praktikumshomepage¹ aufrufen können.

Wie in der Praktikumsordnung beschrieben, sind die Aufgaben relativ offen gestellt. Besprechen Sie diese innerhalb Ihrer Gruppe und konkretisieren Sie die Aufgabenstellung. Die Teile der Aufgabe, in denen Assembler-Code anzufertigen ist, sind für die 64-Bit x86-Architektur (x86-64) unter Verwendung der SSE-Erweiterungen zu schreiben.

Der **Abgabetermin** ist der **24. Juli 2020, 11:59 Uhr (MESZ)**. Die Abgabe erfolgt per Git in das für Ihre Gruppe eingerichtete Projektrepository. Bitte beachten Sie die in der `README.md` angegebene Liste von abzugebenden Dateien.

Die **Abschlusspräsentationen** finden in der Zeit vom **10.08.2020 – 28.08.2020** statt. Weitere Informationen werden noch bekannt gegeben. Beachten Sie, dass die Folien für die Präsentation am obigen Abgabetermin im PDF-Format abzugeben sind. Geben Sie *außerdem* eine Datei mit Notizen zum Vortrag ab, mit deren Hilfe die Folien auch ohne den gehaltenen Vortrag nachvollziehbar sind. Der Kurzvortrag entfällt ersatzlos.

Sofern die Rahmenbedingungen (Hygiene-, Abstandsregeln etc.) für eine Präsenzprüfung nicht vorliegen, kann gemäß §13a APSO die geplante Prüfungsform auf eine virtuelle Präsentation (Videokonferenz) oder eine kurze schriftliche Fernprüfung umgestellt werden. Die Entscheidung über diesen Wechsel wird möglichst zeitnah, spätestens jedoch 14 Tage vor dem Prüfungstermin nach Abstimmung mit dem zuständigen Prüfungsausschuss bekannt gegeben.

Bei Fragen/Unklarheiten in Bezug auf den Ablauf und die Aufgabenstellung wenden Sie sich bitte an Ihren Tutor.

Wir wünschen Ihnen viel Erfolg und Freude bei der Bearbeitung Ihrer Aufgabe!

Mit freundlichen Grüßen
Die Praktikumsleitung

PS: Vergessen Sie nicht, sich rechtzeitig in TUMonline zur Prüfung anzumelden. Dies ist Voraussetzung für eine erfolgreiche Teilnahme am Praktikum im laufenden Semester.

¹<https://gepasp.in.tum.de/eraweb/>

2 Gammakorrektur

2.1 Überblick

Im Zuge Ihrer Projektaufgabe werden Sie theoretisches Wissen aus der Mathematik im Anwendungszusammenhang verwenden, um einen Algorithmus in Assembler zu implementieren. Sie konzentrieren sich dabei auf das Feld des *Image Processing*, in welchem Pixelbilder, wie sie typischerweise Digitalkameras produzieren, als Eingabe für bestimmte Algorithmen verwendet werden und mathematische Überlegungen dadurch sichtbar gemacht werden.

2.2 Funktionsweise

Die zu implementierenden Algorithmen werden verwendet, um ein farbiges Bild in Graustufen zu konvertieren und anschließend eine Gammakorrektur durchzuführen. Wir definieren ein Pixel an der Position (x, y) $p_{x,y}$ als einen Vektor mit drei Elementen für die Farbkanäle Rot (R), Grün (G) und Blau (B):

$$p_{x,y} = \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Wir betrachten zunächst den Graustufenfilter: Die Konvertierung eines Pixels in Graustufen geschieht durch das Berechnen eines gewichteten Durchschnitts D :

$$D = \frac{a \cdot R + b \cdot G + c \cdot B}{a + b + c} \quad (1)$$

Das Pixel in Graustufen ist dann gegeben durch

$$p_{x,y} = \begin{pmatrix} D \\ D \\ D \end{pmatrix}$$

Das Graustufenbild soll anschließend noch einer Gammakorrektur unterzogen werden. Hierzu weist man jedem Pixel (Graustufe) im Bild einen neuen Wert zu, der sich berechnet aus:

$$p_{neu} = \left(\frac{p_{alt}}{255} \right)^\gamma \cdot 255 \quad (2)$$

Die Konstante γ ist dabei eine vom Benutzer zu spezifizierende Fließkommazahl.

2.3 Aufgabenstellungen

Ihre Aufgaben lassen sich in die Bereiche Konzeption (theoretisch) und Implementierung (praktisch) aufteilen. Sie können (müssen aber nicht) dies bei der Verteilung der Aufgaben innerhalb Ihrer Arbeitsgruppe ausnutzen. Antworten auf konzeptionelle Fragen sollten an den passenden Stellen in Ihrer Ausarbeitung in angemessenem Umfang erscheinen. Entscheiden Sie nach eigenem Ermessen, ob Sie im Rahmen Ihres Abschlussvortrags auch auf konzeptionelle Fragen eingehen. Die Antworten auf die Implementierungsaufgaben werden durch Ihrem Code reflektiert.

2.3.1 Theoretischer Teil

- Machen Sie sich mit dem PPM-Bildformat vertraut. Die Verwendung des PPM-Formats erleichtert die I/O-Operationen für Ihr Rahmenprogramm.
- Bestimmen Sie Werte für a , b und c in Gleichung 1 zunächst durch intuitives Festlegen und passen Sie sie später anhand der entstehenden Graustufenbilder weiter an, sobald Sie die Aufgabe gelöst haben. Berücksichtigen Sie hierbei die Eigenschaften des Menschlichen Visuellen Systems (HVS) und ziehen Sie, wenn nötig, geeignete Sekundärliteratur hinzu.
- Entwickeln Sie einen Algorithmus, welcher ein Bild erst in Graustufen konvertiert und anschließend die Gammakorrektur auf allen Pixeln ausführt. Besprechen Sie Ihren Pseudocode mit Ihrem Tutor, bevor Sie mit der Implementierung beginnen, falls nötig.

2.3.2 Praktischer Teil

- Implementieren Sie im Rahmenprogramm I/O-Operationen in C, mit welchen Sie Ihrer Assemblerroutine einen Pointer auf die sequentiellen Bilddaten sowie Höhe und Breite des Bildes übergeben können. Das Rahmenprogramm sollte dann ein neues Bitmap erstellen und die von der Assemblerfunktion berechneten Werte nach dem Aufruf dort hineinschreiben. Das entstehende Bild sollten Sie nach der Verarbeitung wieder im PPM-Format abspeichern, sodass Sie die Ergebnisse in einem gängigen Bildbetrachter (GIMP, ...) überprüfen können.
 - Implementieren Sie in der Datei mit dem Assemblercode Ihren Algorithmus. Die Assemblerfunktion nimmt dabei einen Pointer auf die Pixelwerte des Bildes als Argument und liefert einen Pointer auf ein Array mit den berechneten Ausgabewerten zurück. Sie haben freie Wahl, ob Sie im Assemblercode freien Speicher allokalieren oder einen Pointer auf das Ergebnisarray by-reference übergeben. Zusätzlich nimmt die Funktion die Breite und die Höhe des Bildes, sowie eine Konstante γ für die Gammakorrektur als Argument.
-

2.4 Allgemeine Bewertungshinweise

Die folgende Liste soll Ihnen als Gedächtnisstütze beim Bearbeiten der Aufgaben dienen. Beachten Sie ebenfalls die in der Praktikumsordnung angegebenen Hinweise.

- Stellen Sie unbedingt sicher, dass Ihre Abgabe auf der Referenzplattform des Praktikums (1xhalle) kompiliert und funktionsfähig ist.
 - Fügen Sie Ihrem Projekt ein funktionierendes `Makefile` hinzu, welches durch den Aufruf von `make` Ihr Projekt kompiliert.
 - Verwenden Sie keinen Inline-Assembler.
 - Verwenden Sie SIMD-Befehle, wenn möglich.
 - Verwenden Sie keine x87-FPU- oder MMX-Instruktionen. Sie dürfen alle SSE-Erweiterungen bis SSE4.2 benutzen. AVX-Instruktionen dürfen Sie benutzen, sofern Ihre Implementierung auch auf Prozessoren ohne AVX-Erweiterungen lauffähig ist.
 - Sie dürfen die Signatur der in Assembler zu implementierenden Funktion nur dann ändern, wenn Sie dies (in Ihrer Ausarbeitung) rechtfertigen können.
 - I/O-Operationen dürfen grundsätzlich in C implementiert werden.
 - Denken Sie daran, das Laufzeitverhalten Ihres Codes zu testen (Sichere Programmierung, Performanz) und behandeln Sie alle möglichen Eingaben, auch Randfälle. Ziehen Sie ggf. alternative Implementierungen als Vergleich heran.
 - Eingabedateien, welche Sie generieren, um Ihre Implementierungen zu testen, sollten mit abgegeben werden.
 - Verwenden Sie für die Ausarbeitung die bereitgestellte \LaTeX -Vorlage und legen Sie sowohl die PDF-Datei als auch sämtliche \LaTeX -Quellen in das Repository.
 - Stellen Sie Performanz-Ergebnisse nach Möglichkeit grafisch dar.
 - Vermeiden Sie unscharfe Grafiken und Screenshots von Code.
 - Geben Sie die Folien für Ihre Abschlusspräsentation im PDF-Format ab. Achten Sie auf hinreichenden Kontrast (schwarzer Text auf weißem Grund!) und eine angemessene Schriftgröße. Verwenden Sie 4:3 als Folien-Format.
 - Zusatzaufgaben (sofern vorhanden) müssen nicht implementiert werden. Es gibt keine Bonuspunkte.
-