

Übungen zur Vorlesung: Praktische Parallelprogrammierung

Blatt 3 (Bildverarbeitung: Sobel, VCD)

Aufgabe 3 (Projekt Bildverarbeitung)

[4+6+5+10+5 Punkte]

Wir wollen in diesem Projekt den Sobel-Operator und den VCD-Operator zur Bearbeitung von Graustufenbildern implementieren. Der Sobel-Operator erkennt Kanten in Bildern und der VCD-Operator entfernt fleckenartiges Rauschen, ohne die Kantenschärfe stark zu beeinträchtigen.

- (a) Implementieren Sie den Sobel-Operator sequentiell.
- (b) Implementieren Sie den Sobel-Operator parallel mit OpenMP und MPI.
- (c) Implementieren Sie den VCD-Operator sequentiell.
- (d) Implementieren Sie den VCD-Operator parallel mit OpenMP und MPI.
- (e) Kombinieren Sie die Implementierung von Sobel und VCD zu einem Programm, das es gestattet, zuerst den VCD-Operator und dann den Sobel-Operator anzuwenden.

Sie können alle 5 Teilaufgaben in einem gemeinsamen Programm implementieren; sehen Sie dann z. B. Kommandozeilen-Optionen vor für: Sobel ja/nein, VCD ja/nein, parallel ja/nein. Laden und Speichern Sie jeweils Bilder im PGM-Format (vgl. Blatt 2). Testen Sie Ihre Programme mit den Bildern `bill.pgm` (für Sobel) und `schaedel.pgm` (für VCD+Sobel) in `/home/hpcuser/ppp2010-data/images`. Wählen Sie dazu $c = 0.9$ für den Sobel-Operator und $N = 40, \epsilon = 0.005, \kappa = 30, \delta t = 0.1$ für den VCD-Operator. Messen Sie jeweils die Ausführungszeiten und geben Sie den relativen Speedup für 2, 4, 8, 16 Cores an.

Schicken Sie den Quelltext Ihrer Programme per E-Mail an Armin Größlinger (groessli@fim.uni-passau.de) und geben Sie einen Ausdruck in der Vorlesung am 7.12.2010 ab. Bitte beachten Sie die Deadline! Aus prüfungsrechtlichen Gründen können verspätete Abgaben nicht akzeptiert werden!

Hinweise zu C:

Verwenden Sie `#include <math.h>` um mathematische Funktionen (z. B. `sqrt`, `exp`, `hypot`) in Ihr Programm einzubinden. Das Programm muss dann mit `libm` gelinkt werden, d. h. geben Sie beim Aufruf von `gcc` die Option `-lm` an. Beachten Sie, dass für manche Funktionen (z. B. `lrint`) mit der Option `-std=c99` (oder `-std=gnu99`) kompiliert werden muss.

Achten Sie bei Ihren Benchmarks darauf, dass das sequentielle und das parallele Programm mit konsistenten Optionen für die Code-Optimierung (etwa `-O3 -fomit-frame-pointer`) übersetzt werden und keine der beiden Versionen „zufällig“ besser optimiert werden kann.

Sobel-Operator:

Der Sobel-Operator mit Parameter c ordnet jedem Bildpunkt $s(x, y)$ den neuen Wert $t(x, y)$ nach folgender Formel zu:

$$\begin{aligned}s_x(x, y) &= s(x-1, y-1) + 2 \cdot s(x, y-1) + s(x+1, y-1) \\ &\quad - s(x-1, y+1) - 2 \cdot s(x, y+1) - s(x+1, y+1) \\ s_y(x, y) &= s(x-1, y-1) + 2 \cdot s(x-1, y) + s(x-1, y+1) \\ &\quad - s(x+1, y-1) - 2 \cdot s(x+1, y) - s(x+1, y+1) \\ t(x, y) &= c \cdot \sqrt{s_x(x, y)^2 + s_y(x, y)^2}\end{aligned}$$

Für Punkte außerhalb des Bildes wird $s(x, y) = 0$ gesetzt.

VCD-Operator (VCD=Variable Conductance Diffusion):

Der VCD-Operator hat die Parameter $N, \epsilon, \kappa, \delta t$. Die Operation (am ganzen Bild) ist vollendet, wenn die unten angegebene Vorschrift N -mal angewendet wurde, oder an jedem inneren Bildpunkt $|\Delta(x, y)| \leq \epsilon$ gilt. In einem einzelnen Schritt wird jedem Bildpunkt $s(x, y)$ der neue Wert $t(x, y)$ nach folgender Formel zugeordnet:

$$\begin{aligned}t(x, y) &= s(x, y) + \kappa \cdot \delta t \cdot \Delta(x, y) \\ \Delta(x, y) &= \phi(s(x+1, y) - s(x, y)) - \phi(s(x, y) - s(x-1, y)) \\ &\quad + \phi(s(x, y+1) - s(x, y)) - \phi(s(x, y) - s(x, y-1)) \\ &\quad + \xi(s(x+1, y+1) - s(x, y)) - \xi(s(x, y) - s(x-1, y-1)) \\ &\quad + \xi(s(x-1, y+1) - s(x, y)) - \xi(s(x, y) - s(x+1, y-1)) \\ \phi(\nu) &= \chi \cdot e^{-\frac{\chi^2}{2}} \quad \text{mit } \chi = \frac{\nu}{\kappa} \\ \xi(\nu) &= \frac{1}{\sqrt{2}} \cdot \psi \cdot e^{-\frac{\psi^2}{2}} \quad \text{mit } \psi = \frac{\nu}{\sqrt{2} \cdot \kappa}\end{aligned}$$

Um Rundungsfehler zu verringern, sollten die Graustufenwerte der Pixels als `doubles` repräsentiert werden, damit nicht nach jedem Schritt des VCD-Operators jeder Grauwert auf eine ganze Zahl gerundet werden muss.

Beachten Sie, dass sowohl bei Sobel als auch bei VCD der Bildwert am Ende der Berechnung außerhalb des Intervalls `[0, maxcolor]` liegen kann. Außerhalb liegende Werte werden einfach auf 0 bzw. `maxcolor` gesetzt.