

Projekat iz predmeta Paralelno programiranje

Detekcija ivica unutar slike

U osnovi digitalne obrade slike nalazi se operacija dvodimenzionalne diskretne konvolucije. Uprošćeno, ova operacija se može svesti na množenje odgovarajućih elemenata dveju matrica i sumiranje rezultata. Prva matrica u operaciji konvolucije je jezgro digitalnog filtra (kvadratna matrica dimenzija 3x3, 5x5, 7x7 itd.). Druga matrica je podmatrica slike koja sadrži podatke o vrednostima boja pojedinih tačaka (piksela) slike, i istih je dimenzija kao odabrani filter. Drugim rečima, cela operacija se može posmatrati kao klizanje matrice filtra preko svih podmatrica slike i obavljanje aritmetičkih operacija nad svakim ovako stvorenim parom.

Sljedeća formula opisuje filtriranje slike koja se nalazi u matrici X, filtrom jezgra F, i smeštanje rezultata u matricu Y.

$$Y[x, y] = \sum_{n=0}^2 \sum_{m=0}^2 X[x - 1 + m, y - 1 + n] * F[m, n]$$

Treba primetiti da krajnji indeksi matrice X izlaze izvan opsega same matrice kada [x,y] indeksira celu matricu. Ovakve ivične efekte ćemo, zarad jednostavnosti, tretirati smanjivanjem izlazne matrice za po jednu tačku (piksel) sa svake strane tj. izlazna slika će biti manja (ako je početna slika bila dimenzija 1000x1000 piksela, rezultatna će biti dimenzija 998x998).

Ulazne slike za potrebe ovog projekta treba da se nalaze u datotekama BMP formata (.bmp) sa RGB prostorom boja. To znači da se opis svake tačke u slici sastoji od vrednosti crvene, zelene i plave komponente te tačke. Zarad uprošćenja, filtriranje će se u ovom projektu raditi nad crno-belom slikom (eng. *grayscale*). Ovaj deo je za vas već odrađen tako da ne morate da vodite računa o njemu, samo je napomena da će izlazna slika biti crna sa belim ivicama.

Zadaci

Implementirati sekvencijalni program za detekciju ivica ulazne slike pomoću Prewitt operatora (slika 1.). Ulaznu sliku je potrebno filtrirati operatorima za horizontalne, odnosno vertikalne ivice, a rezultujuću vrednost (tačku izlazne slike) izračunati zbirom apsolutnih vrednosti horizontalne i vertikalne komponente. Iz izlazne slike otkloniti sivu komponentu postavljanjem svih vrednosti ispod globalnog praga $T = 128$ na belu boju, a vrednosti iznad praga na crnu. Time ćemo dobiti crnu sliku sa izraženim belim ivicama (koje će biti više ili manje jasne u zavisnosti od same slike kao i ugla gde se nalaze ivice). Izmeriti vreme potrebno za obradu slike (ali iz merenja isključiti vreme potrebno za učitavanje i pretvaranje slike u crno-beli format: samo vreme obrade nas interesuje).

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \quad Gy = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A$$

$$|G| = |Gx| + |Gy|$$

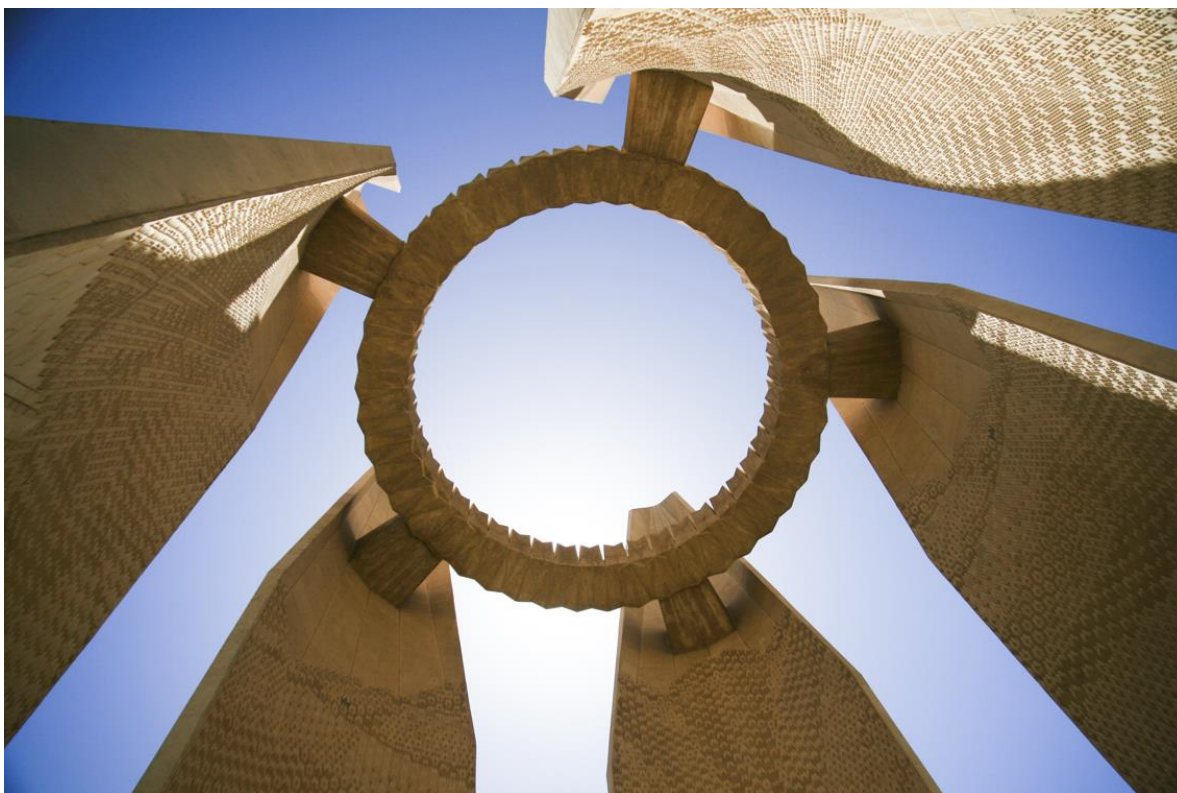
Slika 1 – Prewitt operator (Gx i Gy predstavljaju horizontalne odnosno vertikalne promene u ulaznoj slici A)

Nakon uspešne implementacije Prewitt operatora, implementirati sekvencijalni program za detekciju ivica ulazne slike koji radi na sledeći način. Postaviti vrednosti komponenta ulazne slike odsecanjem za prag $T = 128$ na 0 ili 1. Ovo će rezultovati crno-belom slikom ali bez izdvojenih ivica. Potom pretražiti okolinu svake tačke na sledeći način:

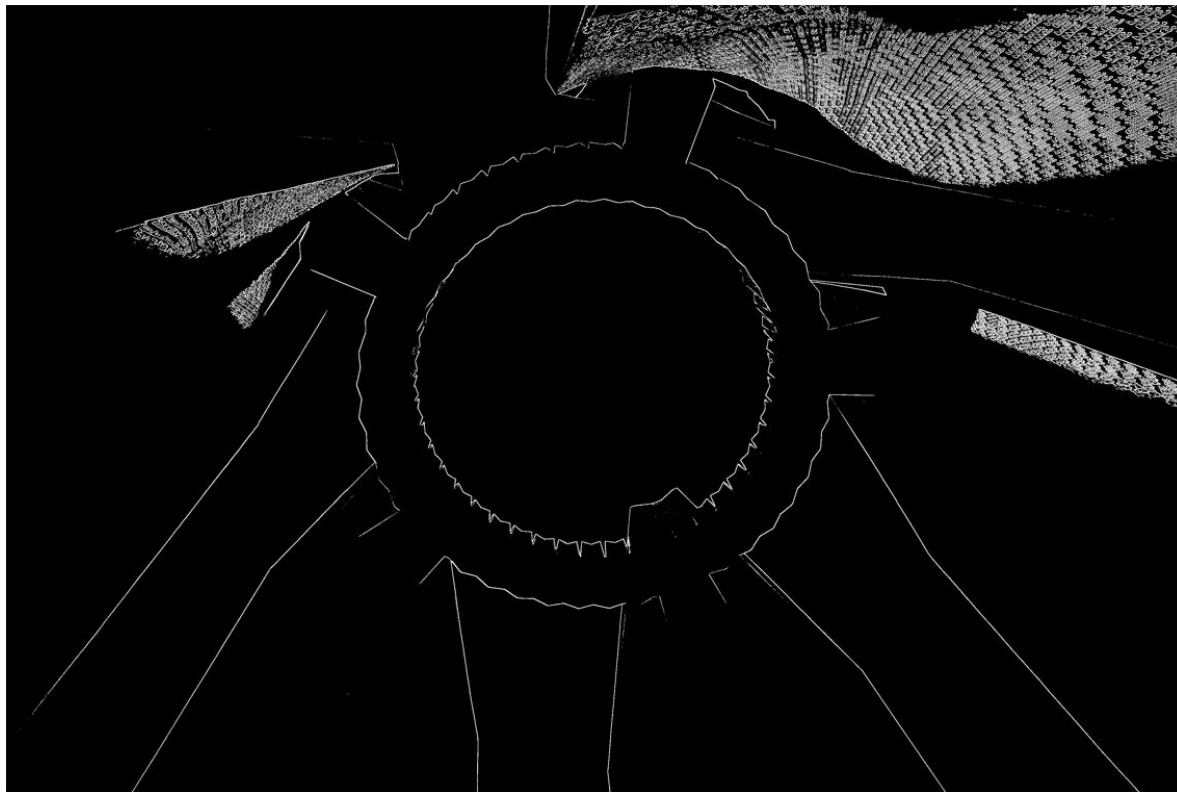
- $P(i,j) = 1$, ako u okolini tačke postoji tačka sa vrednošću 1
- $P(i,j) = 0$, ako u okolini tačke ne postoji tačka sa vrednošću 1
- $O(i,j) = 0$, ako u okolini tačke postoji tačka sa vrednošću 0
- $O(i,j) = 1$, ako u okolini tačke ne postoji tačka sa vrednošću 0

Okolinu tačke predstavljaju njoj susedne tačke: levo, desno, iznad, ispod i dijagonalno na sve strane. Izračunati vrednost rezultujuće tačke razlikom apsolutnih vrednosti $P(i,j)$ i $O(i,j)$ a rezultat vretiti u opseg 0-255. Izmeriti vreme potrebno za obradu slike za serijsku obradu slike (ali iz merenja isključiti vreme potrebno za učitavanje i pretvaranje slike u crno-beli format: samo filtriranje nas interesuje).

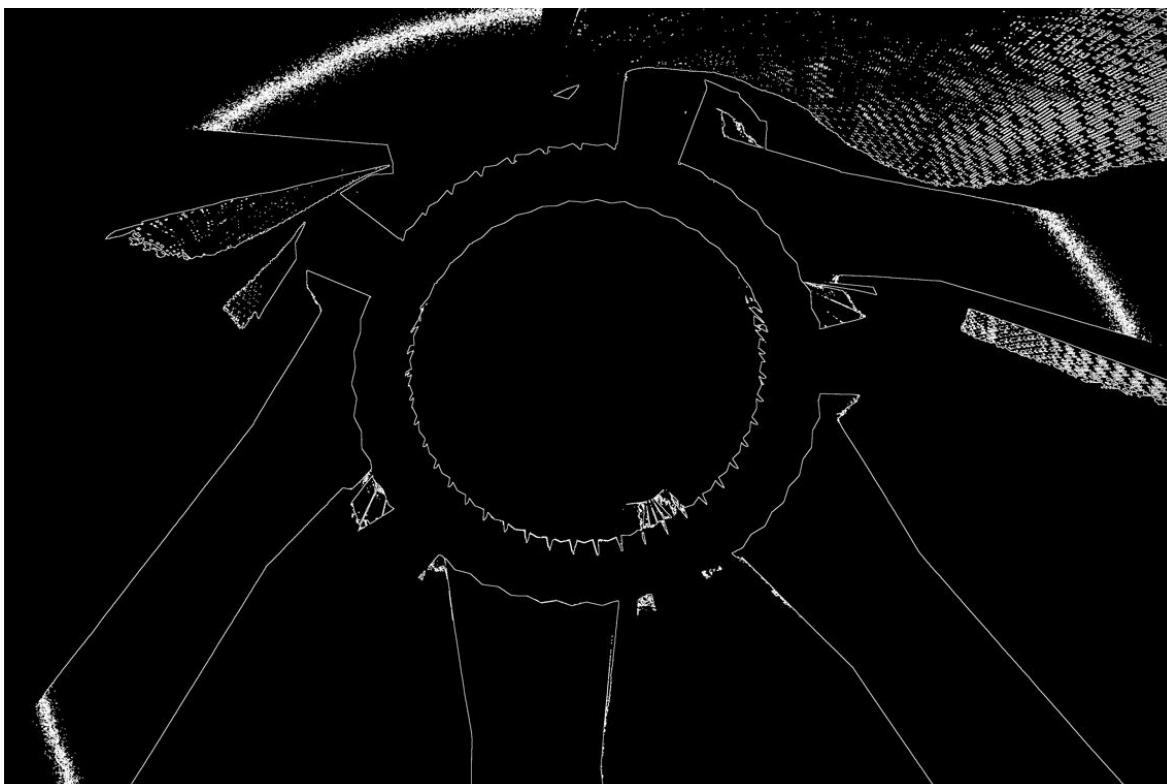
Nakon uspešno urađene serijske verzije programa, rezultat bi trebao da izgleda kao na slikama 2, 3 i 4.



Slika 2 – Polazna slika u BMP formatu



Slika 3 – Izlazna slika nakon detekcija ivica Prewitt operatorom dimenzije 3



Slika 4 – Izlazna slika nakon detekcija ivica drugom metodom sa najužom okolinom

Nakon uspešnih serijskih implementacija, paralelizovati obe obrade korišćenjem OneTBB zadataka, po uzoru na serijsko rešenje. Program implementirati tako da funkcioniše ispravno za slike proizvoljnih dimenzija. Varirati veličinu ulaznih slika (koje moraju biti u .bmp formatu; bilo bi dobro da postoje i slike sa jasnim oštrim ivicama i one kod kojih ivice nisu tako lako uočljive). Varirati i veličinu Prewitt operatora (obratiti pažnju da dimenzija kvadratne matrice mora biti neparna i proširiti sa vrednostima -1, 0 i 1) kao i okolinu za tačku u drugoj metodi (tako da obuhvata sve tačke na „udaljenosti“ 2, 3 ili više). Uporediti rezultate i komentarisati ubrzanje u zavisnosti od broja jezgara.

Dodatne napomene

Uz projekat je data i uvezana biblioteka EasyBMP koja se koristi za rad sa .bmp datotekama. Nemojte menjati izvorne kodove ove biblioteke jer u suprotnom program možda neće raditi. Dokumentaciju za ovu biblioteku možete pronaći na linku: <http://easybmp.sourceforge.net/documentation.html>

Potrebno je dostaviti i ispitnu dokumentaciju (obavezno odštampati jedan primerak i poneti na odbranu – može crno-bela štampa, samo povedite računa o slikama da se vide jasno). Ispitna dokumentacija treba da sadrži sledeća poglavlja:

- Analiza problema (ukratko svojim rečima opisati problematiku)
- Koncept rešenja (opisati pristup prilikom rešavanja problema, koje tehnike su korištene itd.)
- Programsko rešenje (ukratko opisati najbitnije funkcije i spomenuti koje funkcije iz biblioteke su korišćene i za koju svrhu)
- Ispitivanje (u ovom poglavlju je neophodno spomenuti karakteristike računara na kom je rađeno, prikazati sve testne slučajeve, dimenzije slika i filtera prikazati tabelarno, nacrtati grafik da možemo da vidimo ubrzanje spram povećanja dimenzija problema)
- Analiza rezultata (ukratko prodiskutovati celokupan projekat, da li je postignuto očekivano ubrzanje, ako nije – zašto nije, kako promena veličine filtra i slike utiče na performanse itd.)

Dodatno obratiti pažnju na sledeće stvari:

- Dokumentacija treba da bude dobro formatirana, ažurna i da sadrži sva navedena poglavlja sa adekvatnim sadržajem. Po želji pisati ćirilčnim ili latiničnim pismom, ali nikako „ošišanom“ latinicom.
- Programski kod treba da bude modularan, korektno formatiran i komentaran u stilu koji je već dat u kosturu projekta. Ne treba da postoje zakomentarisani delovi koda u konačnom rešenju.
- Program treba da ispravno radi, da se pokreće i izvršava bez grešaka, eventualno treba uvesti dodatno rukovanje greškama, proveru ispravnosti i oslobađanje resursa.
- Testiranje treba da pokrije više slučajeva, za različite dimenzije kako ulaznih slika, tako i filtera. Po potrebi izmeniti program tako da se svi relevantni parametri zadaju na ulazu prilikom pokretanja.
- Prilikom menjanja dimenzija filtera, slika će se dodatno smanjiti (npr. ako je filter umesto 3x3, dimenzija 5x5 – onda će početna slika dimenzija 1000x1000 na kraju biti 996x996 i tako dalje).