

## 1 Uvod

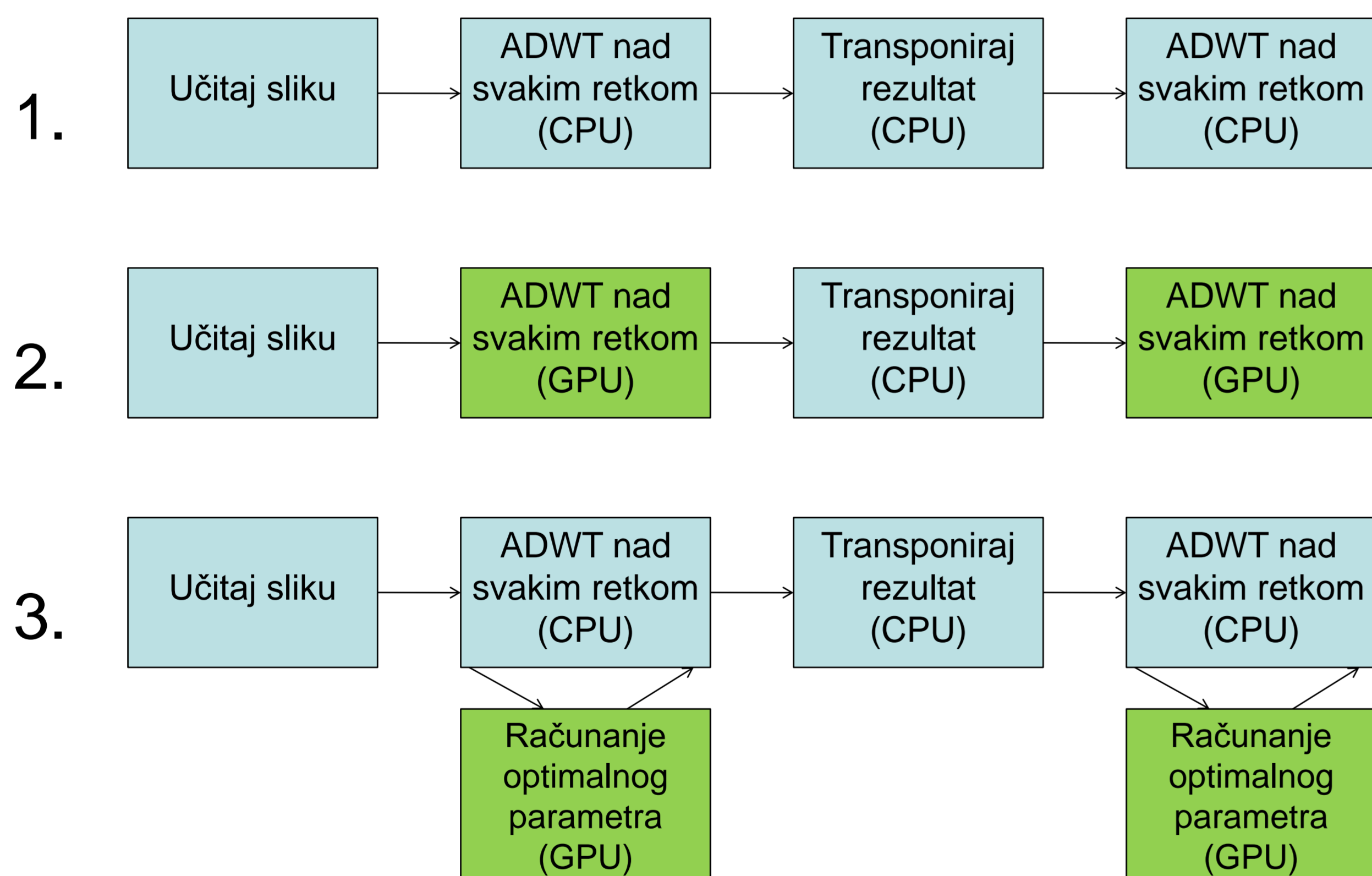
Diskretna valična (*wavelet*) transformacija koristi se u različitim primjenama obrade signala: kompresiji slike, videa, internetskih komunikacija, pri prepoznavanju objekata i u numeričkoj analizi. Ona omogućuje vrlo učinkovitu reprezentaciju signala koji sadrže lokalizirane promjene. Adaptivna valična transformacija posebna je vrsta valične transformacije. Koristi se kako bi se smanjilo rasipanje valičnih koeficijenata, odnosno, kako bi se postigao što kvalitetniji zapis. Vremenski učinkovitom realizacijom adaptivne diskretne valične transformacije omogućuje se njezina šira primjena u situacijama u kojima su točnost i kvaliteta razlaganja iznimno bitne.

## 2 Opis problema

Adaptivna transformacija računski je vrlo zahtjevna. Ubrzanje je potrebno izvršiti izvođenjem algoritma na grafičkom procesoru opće namjene korištenjem arhitekture CUDA. Adaptaciju je potrebno realizirati na dva načina: korištenjem minimuma sume kvadrata pogreške (L2) i korištenjem minimuma sume apsolutnih vrijednosti pogreške (L1), optimizirajući pritom jedan varijabilni parametar.

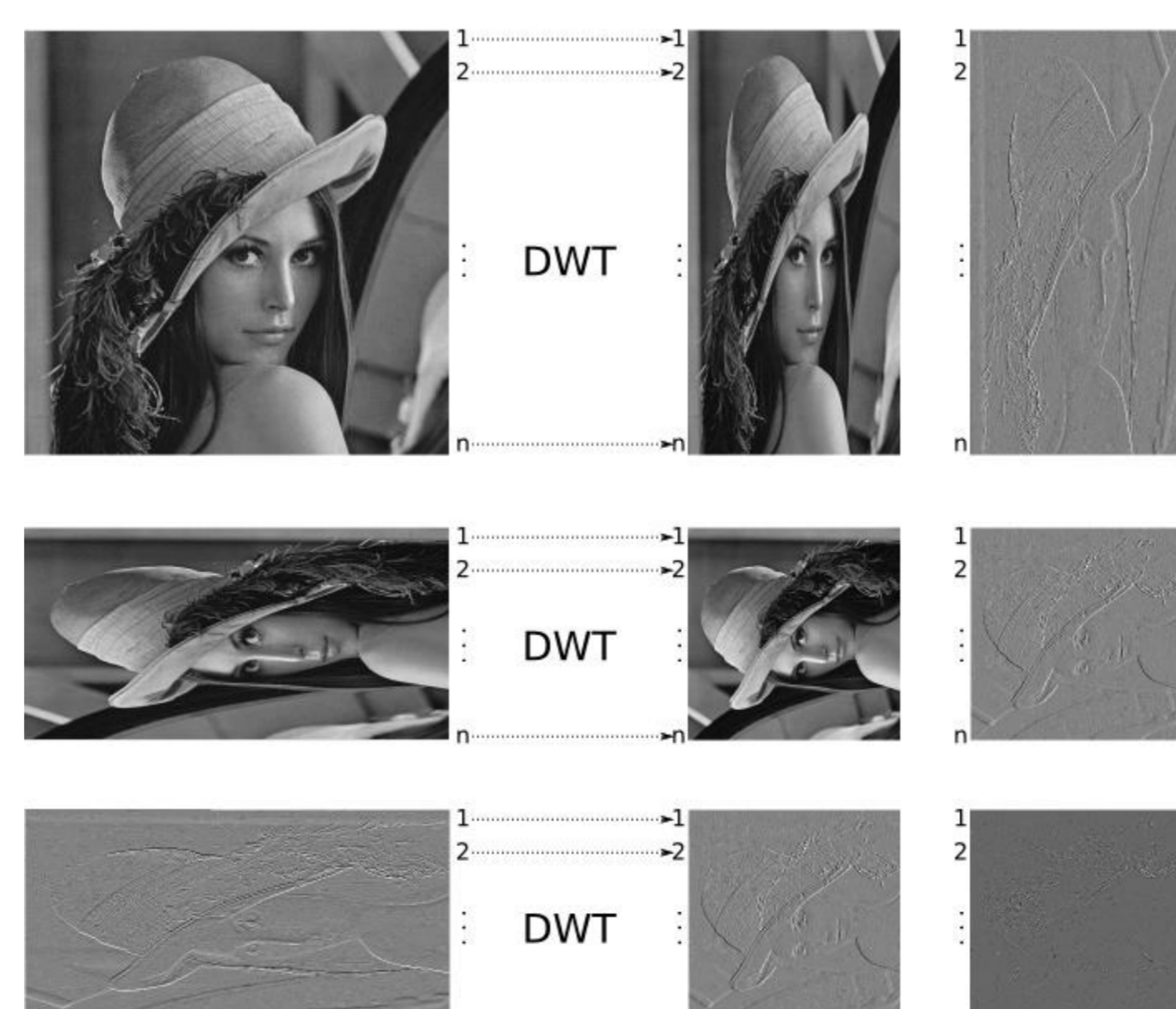
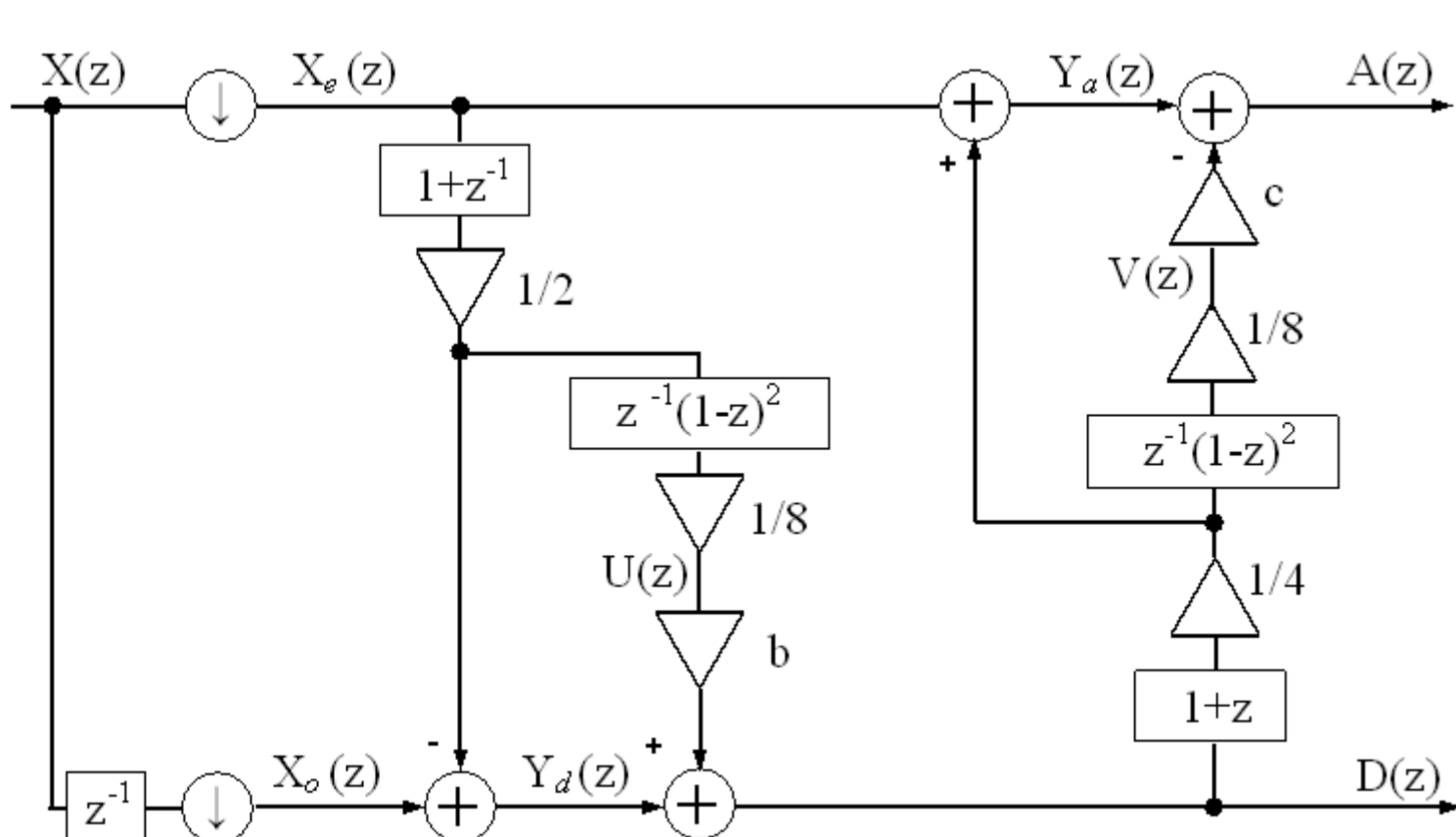
## 3 Metoda

Sustav je implementiran na više različitih načina koji su potom ispitani i uspoređeni. Realizirane su tri metode: metoda koja u potpunosti koristi CPU, metoda koja koristi paralelnu obradu svakog retka na grafičkoj kartici i hibridna metoda koja koristi najbolje od obje metode.



Schema adaptivnog valičnog filterarskog sloga

Valična transformacija slike



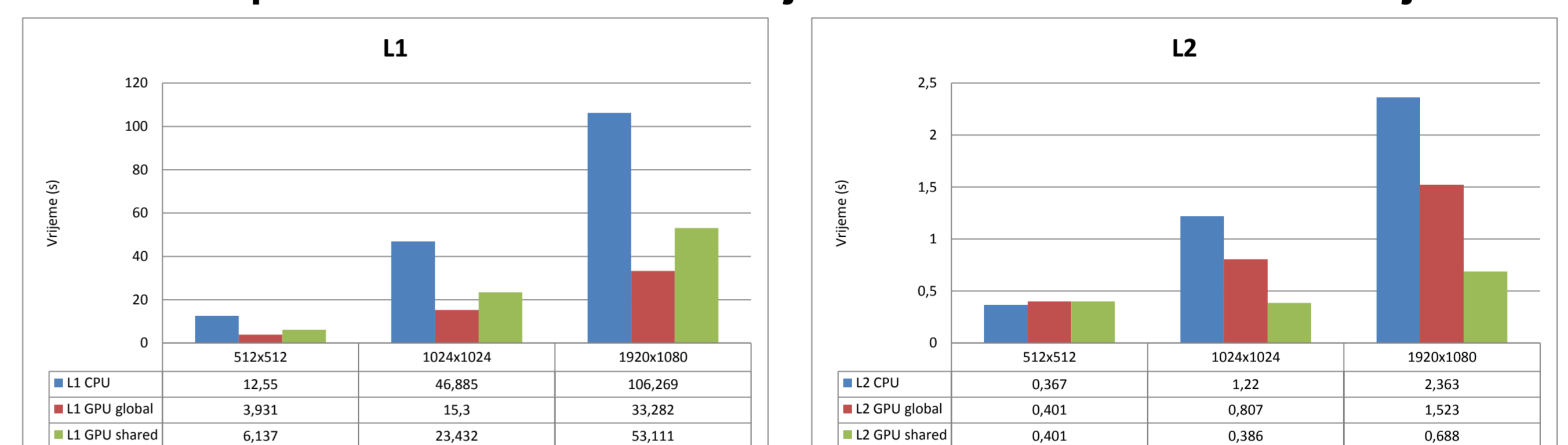
## 4 Rezultati

Metoda direktne paralelne realizacije adaptivne valične transformacije na GPU pokazala se lošom i neučinkovitom. Hibridna metoda koja koristi GPU samo za najintenzivniji dio izračuna, računanje optimalnog varijabilnog parametra, učinkovita je i brza. Kako bi se optimalno iskoristili računski resursi grafičkog procesora, umjesto globalne memorije potrebno je strateški raspodijeliti korištenje dijeljene (*shared*) memorije grafičkog procesora. Povećanjem slike koja se obrađuje povećava se i postignuto ubrzanje hibridne realizacije s obzirom na realizaciju na centralnom procesoru.

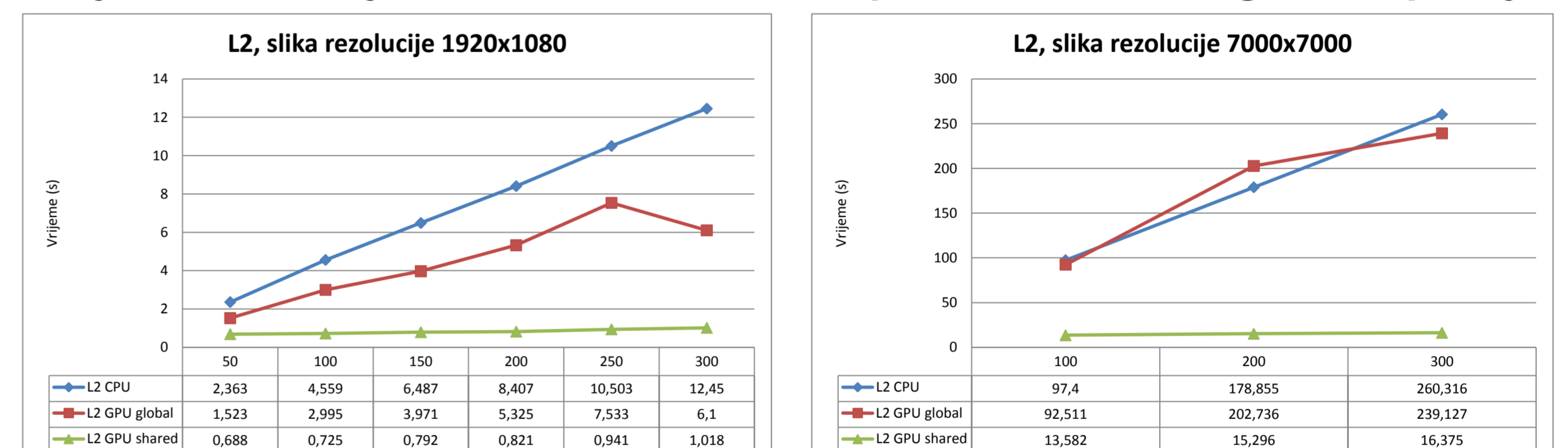
Usporedba ubrzanja izvođenja u odnosu na realizaciju na CPU za manju sliku

Metoda	Faktor ubrzanja
L2 GPU	0,4
L1 GPU	0,31
L2 Hibridno	0,92
L1 Hibridno	3,19
L2 Hibridno, optimizirano	0,92
L1 Hibridno, optimizirano	2,05

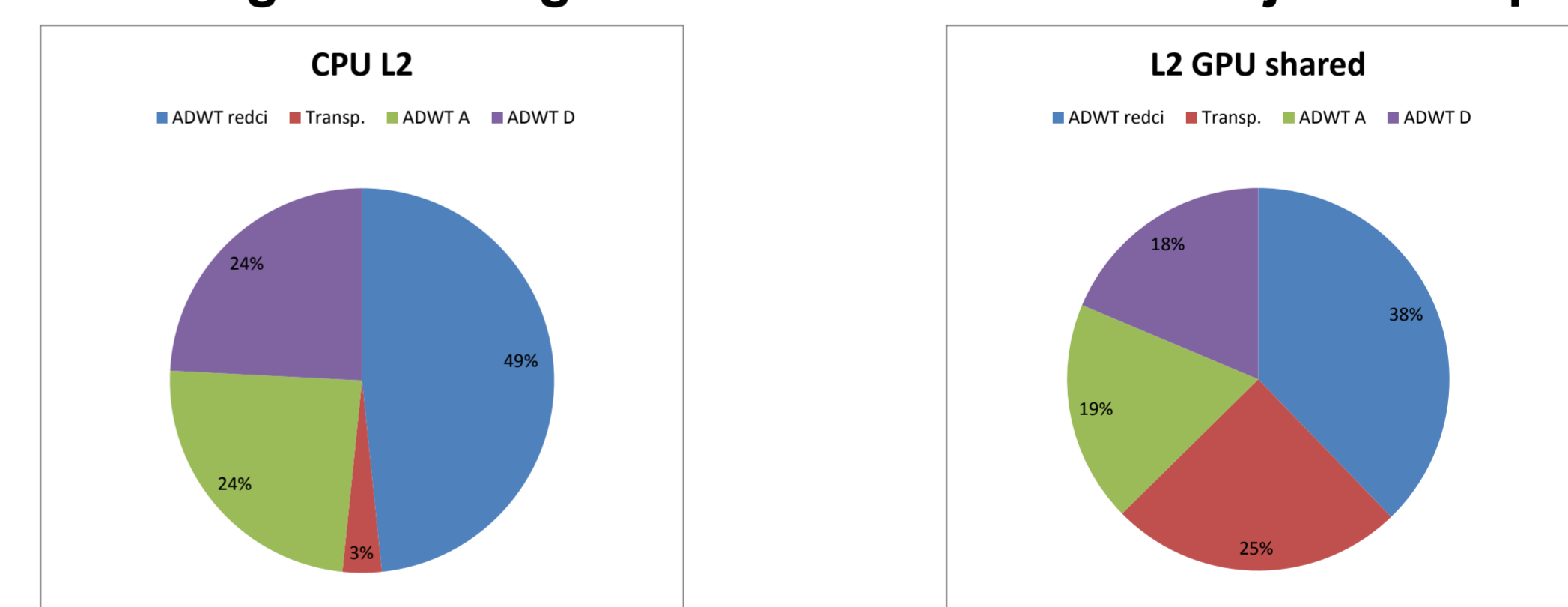
Usporedba vremena izvođenja na slikama različite rezolucije



Vrijeme izvođenja s obzirom na veličinu prozora korištenog za adaptaciju



Udjeli pojedinih segmenata algoritma u vremenu izvođenja dekompozicije slike



## 5 Zaključak

U većini slučajeva hibridna implementacija adaptivne valične dekompozicije slike pokazala se bržom od implementacije koja se izvodi samo na centralnom procesoru. Što je slika veća, potrebno je više računalne snage, a prednosti paralelne implementacije na grafičkom procesoru sve su izraženije. Dobivena su ubrzanja koja u nekim slučajevima iznose i više od deset puta. CUDA se pokazala kao dobar alat za ubrzanje izvođenja ovog algoritma. Ovo je samo jedna od mnogih potencijalnih primjena paralelnog programiranja. Kako se povećava veličina informacija koje obrađujemo i složenost modela kojim ih opisujemo, povećavat će se i potreba za računalnom snagom. Paralelno programiranje i tehnologije poput CUDA-e bit će stoga sve važniji u napretku obradbe informacija.