

Puzla projekat

Algoritam koji je korišten nema specifičan naziv, ali funkcioniše na sledeći način. Na osnovu najčešćih dimenzija delova slike određuje se veličina prozora i njegov korak kojim će on “klizati” po originalnoj slici. Na svakom koraku, pikseli slike koji se nalaze unutar prozora porede se sa svim učitanim delovima puzle. Poređenje delova slike sa pikselima unutar prozora radi se računanjem njihove razlike to jest sume razlika pojedinačnih piksela. Pokušano je poređenje piksela uz pomoć RGB, HSV i LAB modela, korišćenjem gausove srednje vrednosti, apsolutne razlike i delta E LAB metrike i kao najbolja metoda ispostavila se apsolutna razlika RGB kanala. Nakon završetka poređenja svih delova sa trenutnim prozorom pronalazi se najbolji kandidat, kome se upisuju koordinate početka prozora i na taj način dodeljuje njegova pozicija na slici. Budući da u prvom prolazu neke puzle ne mogu da se reše, još jedan deo algoritma se stara o tome da neraspodeljene delove upari sa koordinatama delova originalne slike koje još uvek nikome nisu dodeljene. Granični slučajevi javljaju se kod delova koji se nalaze na obodima slike zbog toga što oni nisu istih dimenzija kao ostali delovi, pa su za poslednju kolonu i red primenjivana malo drugačija pravila i ti delovi puzle su rešavani u zasebnim petljama, čija je ideja rešavanja generalno ista, samo su neki parametri izmenjeni. Kada su svi delovi indeksirani algoritam ih sortira i grupiše prema koordinatama, prvo sastavlja horizontalne delove, a zatim horizontalne delove sklapa u konačnu sliku. Svaka petlja koja rešava puzlu paralelizovana je tako da se kreira onoliko niti koliko ima procesorskih jezgara u računaru i zatim svakoj niti dodeli jednak deo puzle koji treba da proverava, a u slučaju da se puzla ne može tačno podeliti, ostatak se ravnomerno raspoređuje po svim nitima. Sistemske niti su izabrane uprkos postojanju implementacija zelenih niti unutar određenih biblioteka iz razloga što je procesiranje u ovom algoritmu isključivo CPU-bound te se kreiranje zelenih niti i vršenje time slicing-a na nivou sistemske niti ne isplati (algoritam ne sadrži I/O operacije) i može samo pogoršati performanse. Nakon poređenja sa sekvencijalnim rešenjem nad primerima koji su dati zajedno sa specifikacijom projekta, paralelno rešenje se ispostavilo 38% brže.