1. Uvod

Cilj ovog dokumenta je predstaviti razvoj i implementaciju algoritma za rekonstrukciju slike kroz puzle. Pristup koji ćemo istražiti obuhvata analizu karakteristika svake sličice, razmatranje topologije slike, i primenu algoritamskih tehnika koje omogućavaju efikasno sastavljanje originalne slike.

Očekivani rezultati uključuju preciznu i brzu rekonstrukciju slike, sa značajnim naglaskom na očuvanju detalja i strukture. Ovaj algoritam može poslužiti kao osnova za dalje istraživanje i primenu u širem spektru oblasti gde je rekonstrukcija slika kroz puzle od suštinskog značaja.

U nastavku dokumentacije, detaljno ćemo razmatrati metodologiju, implementaciju, rezultate i mogućnosti daljeg unapređenja ovog inovativnog algoritma.

2. Opis algoritma

Pristup rešavanju problema sastavljanja slike kroz puzle zasniva se na algoritmu koji efikasno kombinuje fragmente slika kako bi formirao konačnu sliku. Ovaj algoritam, koji smo razvili i implementirali, prolazi kroz nekoliko ključnih koraka kako bi ostvario precizno sastavljanje slike iz niza manjih komponenti.

1. **Učitavnje finalne slike:** Prvi korak algoritma podrazumeva učitavanje ciljane slike, koja će poslužiti kao referentna tačka za rekonstrukciju. Ova slika služi kao osnova za formiranje krajnjeg rezultata.
2. **Učitanje ostalih delova (puzli):** Nakon ucitavanja finalne slike, algoritam treba da ucita ostale slike u jedan vektor, što predstavlja savšeno mesto za paralelizaciju o kojoj će biti više reči posle.
3. **Inicijalizacija prazne slike:** Na osnovu dimenzija finalne slike, algoritam kreira praznu sliku koja će na kraju predstavljati sastavljeni rezultat. Ova prazna slika inicijalno nema informacija, i cilj je postepeno je popuniti pomoću manjih delova.
4. **Pronalaženje najbolje puzle za svaku poziciju:** Algoritam se zatim kreće kroz originalnu sliku i analizira svaku poziciju kako bi pronašao najbolji odgovarajući deo slike. Ovaj proces se zasniva na izračunavanju euklidske distance.
5. **Postavljanje puzli na praznu sliku:** Kada se pronađe odgovarajuća puzla, algoritam je postavlja na odgovarajuću poziciju na praznoj slici. Nakon postavljanja, puzla se izbacuje iz vektora svih puzli kako bi se izbegla ponovna upotreba.
6. **Iterativan proces:** Ovaj proces se ponavlja dok se sve puzle ne postave na praznu sliku, pridržavajući se procesa selektivnog izbacivanja puzli iz vektora kako bi se osiguralo da svaka puzla bude korišćena samo jednom.

3. Paralelizacija

Prilikom učitavanja puzli, algoritam prolazi kroz vektor putanja do puzli, a zatim ih učitava u drugi vektor, `images`. Svaka puzla se učitava pojedinačno, što čini ovaj proces idealnim kandidatom za paralelizaciju. Rezultati performansi, uključujući vreme izvođenja za određene primere, jasno ukazuju na poboljšanja:

- IMAGE 1: Sequential: ~ 13.177541ms → Parellel load: 3.6993ms (12 images)
- IMAGE 1-1: Sequential: ~ 35.234812ms → Parellel load: 8.720259ms (348 images)
- IMAGE 2: Sequential: ~ 16.889508ms → Parellel load: 4.134215ms (15 images)
- IMAGE 2-1: Sequential: ~ 50.643325ms → Parellel load: 6.579948ms (224 images)
- IMAGE 3: Sequential: ~ 24.019431ms → Parellel load: 6.199604ms (192 images)
- IMAGE 4: Sequential: ~ 20.516626ms → Parellel load: 5.317043ms (15 images)
- IMAGE 5: Sequential: ~ 54.943702ms → Parellel load: 17.755043ms (96 images)

Rezultati ukazuju na značajno smanjenje vremena izvođenja u paralelnom režimu, posebno kod scenarija sa većim brojem sličica. Ova optimizacija omogućava brže učitavanje puzli, što dalje doprinosi ubrzanju celokupnog algoritma za sastavljanje slike.

Algoritam se kreće kroz originalnu sliku, a za svaki deo slike vrši pretragu svih puzli iz vektora `images` koristeći euklidsko odstojanje. Najmanja euklidska udaljenost ukazuje na najbolju puzlu koja se zatim lepi na finalnu sliku. Ovaj proces pretrage može se efikasno paralelizovati, kao što potvrđuju rezultati performansi:

- IMAGE 1: Sequential: ~ 75.464212ms → Parallel: ~ 36.41129ms
- IMAGE 1-1: Sequential: ~ 1.050258763s → Parallel: ~ 393.177699ms
- IMAGE 2: Sequential: 81.254223ms → Parallel: ~ 40.175297ms
- IMAGE 3: Sequential: ~ 712.71049ms → Parallel: ~ 255.992121ms
- IMAGE 4: Sequential: ~ 87.407405ms → Parallel: ~ 42.342929ms
- IMAGE 5: Sequential: ~ 1.400382475s → Parallel: ~ 446.090356ms

Rezultati ukazuju na značajno smanjenje vremena izvođenja tokom paralelnog procesa pretrage. Ova optimizacija doprinosi ukupnoj efikasnosti algoritma za sastavljanje slike, pružajući bržu rekonstrukciju uz očuvanje kvaliteta.

Paralelizacija procesa pretrage za najbliže puzle na osnovu euklidskog odstojanja značajno unapređuje performanse algoritma sastavljanja slike kroz puzle. Ovi rezultati potvrđuju da paralelizacija igra ključnu ulogu u optimizaciji celokupnog algoritma, čime se postiže bolje vreme izvođenja i povećava efikasnost.