



Факултет техничких наука

Универзитет у Новом Саду

Паралелне и дистрибуиране архитектуре и језици

Паралелна реконструкција слика у Руст програмском језику

Аутор:
Андрија Станишић

Индекс:
Е2 105/2023

30. јануар 2024.

Сажетак

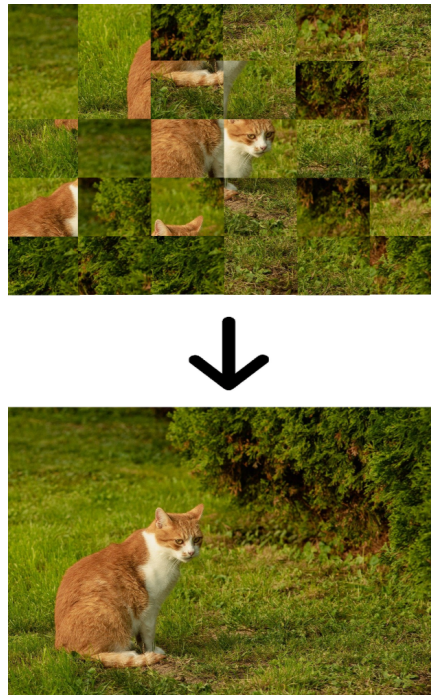
Ефикасна обрада великих или високо-резулацијских слика може бити ресурсно захтеван задатак. У овом контексту, често се приступа методи дељења слика на n мањих делова, који се касније засебно обрађују. Овакав приступ је посебно користан за слике које су тешке за обраду у целости због ограничења меморије или процесорског времена. Иако дељење почетне слике значајно побољшава процес обраде, из њега произилази проблем поновног састављања већ подељене слике. Стога, ово истраживање има за циљ да пружи конкретан начин за решавање поменутог проблема. Решење је развијено у програмском језику *Rust*, који се истиче не само својом брзином извршавања и строгим контролом меморије, већ и способношћу да обезбеди конкурентно и паралелно програмирање без угрожавања безбедности.

Садржај

1	Дефиниција проблема	1
2	Опис решења	2
2.1	Прозор претраге	2
2.2	Еуклидска удаљеност	2
3	Паралелни приступ	3

1 Дефиниција проблема

Генерисање слике на основу њених делова је неизоставни део бројних приступа заступљених у модерној анализи и обради слика. Овај процес захтева прецизну анализу и усклађивање како би се реконструисала оригинална слика у целости. Партиционисање слика на мање сегменте налази своју примену у разним областима, почевши од дигиталне форензике, преко медицинске дијагностике, па све до уметности. Ефикасно и прецизно састављање ових сегмената може значајно унапредити квалитет и вредност коначног производа у свакој од поменутих области. Иако веома ефикасан, овакав приступ генерисања слика често представља изузетно комплексан задатак. Комплексност овог процеса произилази из потребе да се сваки сегмент слике прецизно уклопи, како би се очувао интегритет крајње слике. Управо зато, овај рад ће се фокусирати на развијање алгоритма за генерисање слике базиран на њеним деловима, као и на њој самој. Овакав приступ значајно олакшава процес реконструкције и омогућава прецизније усклађивање, што може довести до бољег квалитета и веће веродостојности реконструисане слике. На слици 1, приказан је улазни скуп фрагмената, као и очекивани излаз, што илуструје проблем којим се овај истраживачки рад бави.



Слика 1: Пример проблема реконструкције слике.

2 Опис решења

У наредној секцији пружен је детаљан увид у имплементацију решења за претходно дефинисан проблем. Решење је у потпуности реализовано помоћу *Rust* програмског језика, са фокусом на ефикасност и прецизност. Иницијално је развијен секвенцијални приступ. Анализом овог приступа идентификовани су сегменти алгоритма погодни за паралелизацију. Имплементација паралелног приступа значајно је унапредила перформансе алгоритма, доказујући предности паралелне обраде у овом контексту.

2.1 Прозор претраге

Централна идеја алгоритма је употреба прозора претраге који одговара димензијама тренутног сегмента слике. Померањем прозора претраге по иницијалној слици омогућава се идентификација оптималне позиције за сваки сегмент појединачно. Ради лакше манипулације, иницијална слика, као и сви њени сегменти, трансформисани су у матрице. Матрица је организована као вектор вектора, где сваки елемент представља боју пиксела, изражену помоћу уређене тројке која означава интензитет црвене, зелене и плаве боје.

2.2 Еуклидска удаљеност

Као што је већ дефинисано, сваки пиксел слике представљен је као уређена тројка интензитета црвене, зелене и плаве боје, стога се сви они графички могу представити помоћу тродимензионалног Декартовог координатног система. Како би се нашла оптимална, позиција за сваки сегмент појединачно, коришћена је еуклидска удаљеност (1). Оптимална позиција за сваки сегмент је она где је еуклидска удаљеност између пиксела сегмента и одговарајућих пиксела на оригиналној слици минимална, што значи да су на тој позицији боје најсличније. Кординате првог пиксела оптимале позиције се затим бележе како би се омогућило ефикасно спајање фрагмената. Битно је напоменути да се пореде само они пиксели који се налазе на ивицама прозора претраге. Поређење сваког пиксела прозора претраге би значајно успорило алгоритам без приметне разлике у квалитету генерисане слике.

$$d(p_1, p_2) = \sqrt{(r_2 - r_1)^2 + (g_2 - g_1)^2 + (b_2 - b_1)^2} \quad (1)$$

3 Паралелни приступ

Приликом анализе секвенцијалног приступа у решавању проблема реконструкције слика, уочена су два аспекта који су погодни за паралелизацију. Први аспект је учитавање фрагмената слике. Када је у питању велики број фрагмената, паралелизација овог процеса може значајно убрзати укупно време извршавања алгоритма. Други аспект је могућност паралелног одређивања оптималних позиција фрагмената, јер различите итерације алгоритма нису међусобно зависне. Ова чињеница омогућава истовремено процесирање више фрагмената, доприносећи тако додатном повећању ефикасности алгоритма.

За оптимизацију процеса реконструкције одабран је *Rayon*, алат специјализован за паралелно програмирање у Rust програмском језику. Употребом овог алата омогућава се ефикасна расподела послова на више нити, што је кључно у контексту брзог учитавања и обраде великог броја фрагмената улазне слике. Овај приступ посебно је користан у ситуацијама када се врши обрада слика високих резолуција, које су подељене на велики број фрагмената. Додатно, употреба Rayon-а доприноси бољој скалабилности алгоритма, омогућавајући му да се прилагоди и ефикасно рукује великим скуповима слика. Другим речима, овај приступ минимизира време чекања и оптимизује употребу процесорске снаге.

Како би се нагласила ефикасност коју уводи паралелизација, извршена је серија тестова. Алгоритам за реконструкцију слика тестиран је на сликама различитих резолуција, које су подељене у групе са 20 до 1000 фрагмената. Основни циљ ових тестова био је да се упореде перформансе паралелног са перформансама секвенцијалног приступа. Тестирања су изведена на рачунару опремљеном са Intel Core i7-8700 процесором, који има 6 језгара са радним тактом од 3.20GHz, и који управља са 32GB RAM меморије. Анализом добијених резултата, уочено је изузетно велико побољшање у ефикасности алгоритма, посебно на тестовима који су обухватили обраду великог броја фрагмената.