

# Дигитално процесирање на слика

Тема:

Алгоритми за креирање на панорамски слики од повеќе  
слики

Марко Каџбанов 233151

Теона Амбаркова 233231

# Содржина

Апстракт.....	3
Вовед.....	3
Основа.....	3
Панорамски слики.....	4
Објаснување.....	5
Детекција на карактеристики со помош на SIFT.....	5
За рафинирање на локациите на клучните точки.....	7
Доделување на ориентација.....	7
Спојување на клучни точки.....	8
Метрика на растојание.....	9
Пресметување на матрицата на хомографија.....	9
RANSAC алгоритам.....	10
Проблемот на видливи рабови при спојување и примена на тежинско Спојување.....	11
Рекурзија за спојување на повеќе слики.....	13
Отстранување на вишок простор добиен при геометриски Трансформации.....	14
Имплементирани алгоритми и проектна структура.....	16
Резултати.....	17
Анализа.....	19
Заклучок.....	19
Користена литература.....	20

# Апстракт

Панорамските слики претставуваат важен дел од дигиталното процесирање на слики, особено во области како фотографија, географски информациски системи, виртуелна реалност и автоматско мапирање на околина. Во оваа семинарска работа е развиен систем за автоматско спојување на повеќе слики во една панорамска слика користејќи алгоритми за детекција на карактеристики (SIFT), изградба на хомографија со RANSAC, и Gaussian blending за неприметливо спојување. Проектот е имплементиран во Python со користење на OpenCV, и дава висококвалитетни резултати дури и кога сликите се делумно изместени или ротирани.

## Вовед

Со зголемената достапност на дигитални фотоапарати и камери, потребата за автоматизирани системи кои можат да комбинираат повеќе слики во една целосна панорама станува сè поголема. Наместо сликите рачно да се усогласуваат, израмнуваат и лепат, денешните решенија се потпираат на напредни алгоритми од областа на компјутерскиот вид кои автоматски ги идентификуваат преклопувачките региони и применуваат геометриски трансформации за нивно прецизно спојување. Во рамките на овој проект се користат современи методи кои овозможуваат стабилно и квалитетно спојување дури и кога сликите се изместени, ротирани или со различно осветлување.

Најпрво, се користи алгоритмот за откривање на карактеристики наречен SIFT (Scale-Invariant Feature Transform), кој идентификува значајни точки во сликата што се отпорни на промена на размер, ротација и светлина. Овие точки претставуваат основа за идентификација на сличности помеѓу повеќе слики. Потоа, со примена на Brute-Force метод за совпаѓање, се поврзуваат соодветните карактеристики од различни слики. За да се обезбеди стабилна геометриска врска, се користи RANSAC (Random Sample Consensus), кој пресметува хомографија со елиминирање на погрешните совпаѓања и го наоѓа најдобриот модел за трансформација помеѓу сликите.

Финалната фаза вклучува спојување на сликите на начин кој обезбедува мазни и природни транзиции. Наместо грубо прикачување, системот користи Gaussian маски за постепено прелевање на пикселите од една слика во друга, што резултира со панорамска слика без видливи рабови или нагли промени во боја и осветлување. Овој пристап не само што овозможува визуелно привлечен резултат, туку и покажува висока робусност и прецизност при работа со реални податоци.

## Основа

Во основа на секој систем за автоматско создавање панорамски слики лежат неколку важни алгоритми и концепти од областа на компјутерски вид и дигитална обработка на слики. Секој чекор од процесот на спојување слики се базира на конкретен математички модел или алгоритам кој овозможува автоматско препознавање, трансформација и спојување на содржината од повеќе слики во една единствена, кохерентна слика со проширено видно поле.

## Панорамски слики

Панорамската слика претставува слика со многу пошироко видно поле од вообичаената перспектива што ја добиваме со еден кадар од камера. Фотографиите мора да имаат делумно преклопување, што овозможува идентификација на заеднички карактеристики. Целта на панорамската обработка е да се добие поголема и непрекината слика, да се задржи визуелната континуираност, да се елиминира деформација или преклопување, како и да се зачуваат боите и осветлувањето. Панорамските слики се користат во различни области како географски информации (GIS), виртуелни тури во недвижности или музеи, видео игри и симулатори, роботика и автономни возила, како и во фотографијата, особено за пејзажи.

За да се изврши спојување на повеќе слики, потребно е прво да се идентификуваат преклопувачките делови, да се трансформираат сликите така што ќе се израмнат една со друга и потоа да се спојат на визуелно прифатлив начин. Овој процес вклучува неколку клучни алгоритми.

SIFT (Scale-Invariant Feature Transform) е еден од најважните алгоритми во компјутерскиот вид за детектирање и опишување на локални карактеристики во сликите. Главните предности на овој алгоритам се инваријантност на размер, што значи дека работи дури и ако сликата е зголемена или намалена, инваријантност на ротација, што овозможува препознавање на клучните точки и кога сликата е ротирана, како и робусност на промена на осветлување, што значи дека алгоритмот ги идентификува клучните точки и кога сликите се направени во различни светлосни услови. Работата на SIFT започнува со претворање на оригиналната слика во сива, потоа се применува Difference of Gaussians за да се најдат точките со екстрими во осветленоста, при што за секоја точка се пресметува ориентација и дескриптор. Овие дескриптори, кои се 128-димензионални вектори, се користат за споредба со други слики. Со помош на SIFT можеме да најдеме точки како агли, рабови или специфични детали што се јавуваат во двете слики, дури и ако се сликани од малку различни агли или позиции.

Хомографијата претставува математички модел кој ја претставува проекцијата на една рамнина врз друга. Во контекст на спојување слики, таа служи за трансформација на една слика така што ќе се совпаѓа со друга, што значи промена на големина, ротација, перспектива и позиција. Технички, хомографијата е  $3 \times 3$  матрица што трансформира координати од едната слика во друга. Формулата  $x' = Hx$  покажува како точка од оригиналната слика ( $x$ ) се проектира во нова точка ( $x'$ ) со помош на матрицата  $H$ . Ова овозможува корекција на перспектива, усогласување на слики со различен агол и прецизно совпаѓање на содржината. За хомографијата да биде точна, претходно мора да се детектираат точни совпаѓања меѓу сликите, што се прави со помош на алгоритмот RANSAC.

RANSAC (Random Sample Consensus) е итеративен алгоритам кој овозможува робусна пресметка на модели како хомографијата и одбивање на „лоши“ податоци или отстапувања. При спојување на слики, RANSAC ги отфрла клучните точки кои не се совпаѓаат добро со друга слика. Ако, на пример, имаме 100 совпаѓања помеѓу слики, но само 60 се точни, RANSAC ги наоѓа најдобрите неколку кои создаваат стабилен модел. Процесот вклучува случајно избирање на неколку совпаѓања, пресметување на хомографијата, проверка колку други точки се вклопуваат, повторување на ова стотици пати и избор на најдобрата трансформација со најмногу согласни точки. Благодарение

на RANSAC, процесот на добивање на хомографија е стабилен дури и ако постои шум, изместување или објекти кои се појавуваат само на една од сликите.

По израмнувањето на сликите со помош на хомографијата, се јавува проблем со видливи рабови, разлики во боја, осветлување или преклопување. За да се добие природен изглед, се применува техниката на мазно спојување позната како Gaussian blending. Ова значи создавање на маски кои постепено ја намалуваат видливоста на едната слика додека ја зголемуваат видливоста на другата, избегнувајќи тврд раб и овозможувајќи природен премин помеѓу сликите. Во пракса, ако две слики имаат преклопување, едната полека исчезнува додека другата станува видлива, слично на видео транзиција. За тоа се користат линиски градиенти и Gaussian функции кои делуваат како филтри.

Сите овие алгоритми заедно овозможуваат прецизна анализа на содржината на сликите, усогласување на различни перспективи и природно спојување во една визуелно пријатна целина. Комбинацијата на SIFT, RANSAC, Homography и Blending се користи во реални апликации, од Google Street View до мобилни апликации за камери, а нашата имплементација го следи токму овој пристап.

## Објаснување

### Детекција на карактеристики со помош на SIFT

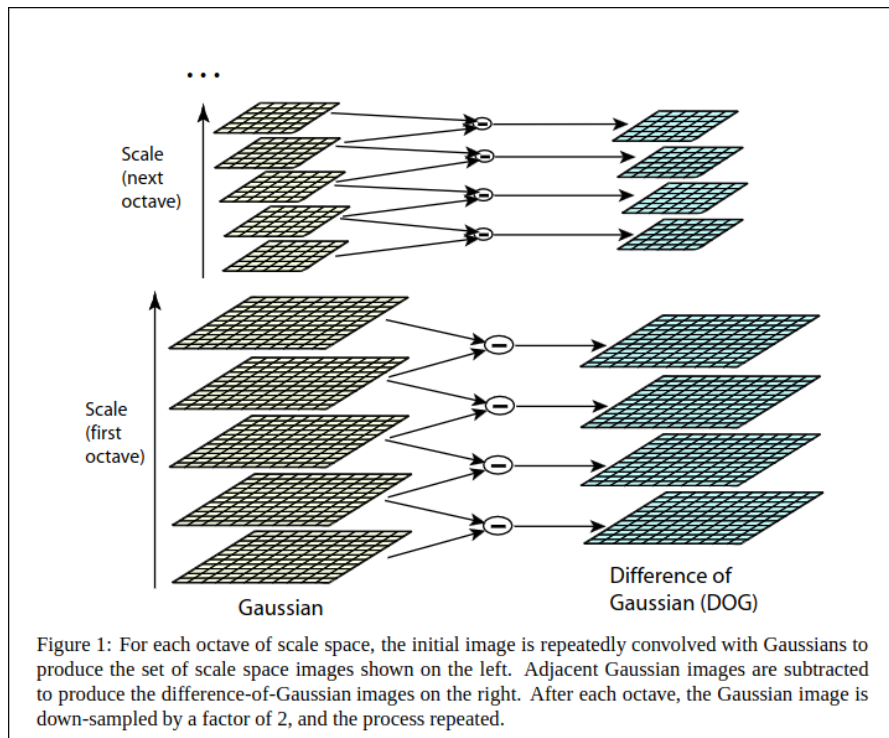
Трансформацијата на карактеристики што не е скалабилна е алгоритам на компјутерска визија за откривање на точки на интерес, опишување и совпаѓање на локалните карактеристики во сликите. [Дејвид Лоу 1999]

Сликата е конвертирана со серија Гаусови филтри на различни размери за да се создаде репрезентација на scale space. Локалните екстрими во овој scale space се идентификуваат како потенцијални клучни точки. Затоа, просторот на скалата на сликата е дефиниран како функција,  $L(x,y,\sigma)$ , која е добиена од конволуцијата на гаусова шема со променлива скала,  $G(x,y,\sigma)$ , со влезна слика,  $I(x,y)$ :

каде  $L(x,y,\sigma)=G(x,y,\sigma)*I(x,y)$

За ефикасно откривање на стабилни локации на клучни точки во просторот на скалата, се користат екстрими на просторот на скалата во функцијата на разликата на Гаусовата шема конволвена со сликата,  $D(x,y,\sigma)$ , која може да се пресмета од разликата на две блиски скали одделени со константен мултипликативен фактор  $k$ :

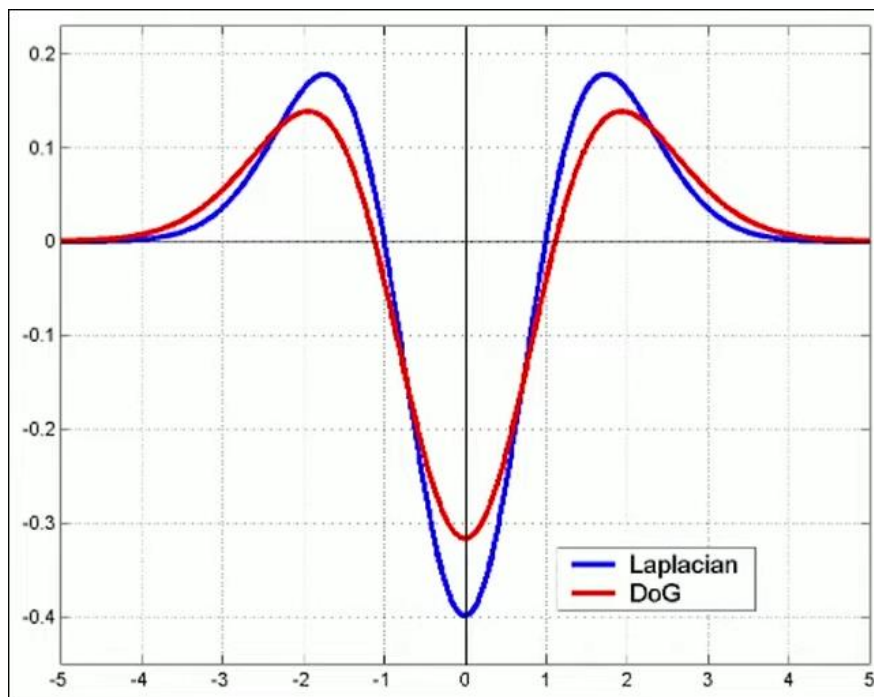
$D(x,y,\sigma)=(G(x,y,k\sigma)-G(x,y,\sigma))*I(x,y)=L(x,y,k\sigma)-L(x,y,\sigma)$



Сл. 1: Scale space со повеќе резолуции, прва фаза на SIFT

Покрај тоа, Difference of Gaussian функцијата обезбедува блиска апроксимација на скално-нормализираната Laplacian of Gaussian,  $\sigma^2 \nabla^2 G$ , како што е проучено од Линдеберг (1994). и затоа,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1) \sigma^2 \nabla^2 G$$

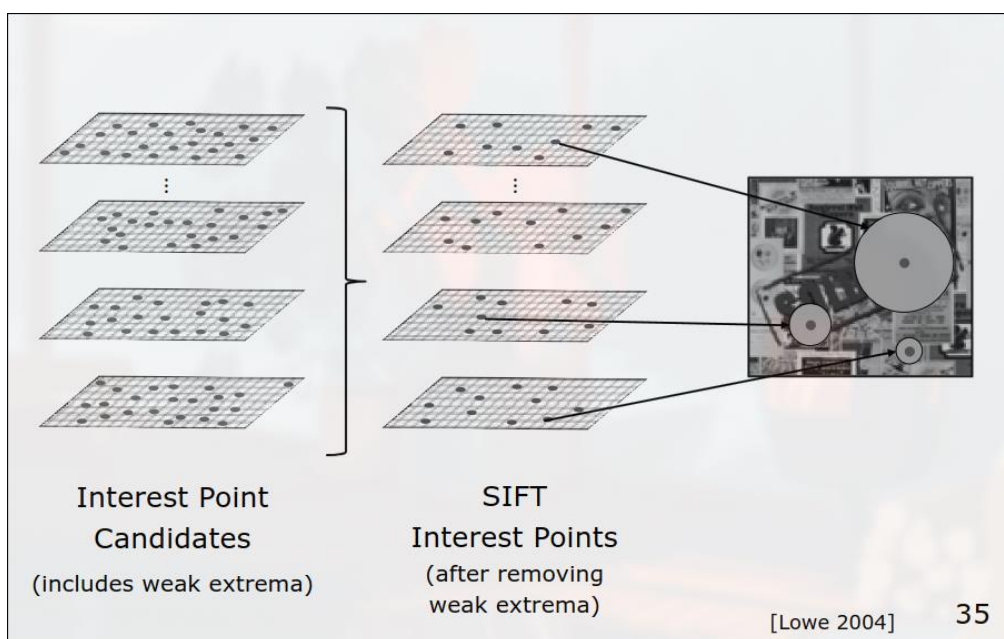


Сл. 2: Споредба на DoG и LoG

Laplacian of Gaussian методот се користи за откривање на карактеристики со истакнување на региони со брза промена на интензитетот на сликата, а често се применува за идентификување на клучни точки или точки од интерес на сликата.

### За рафинирање на локациите на клучните точки:

Се прилагодува тродимензионална квадратна функција (парабола) околу секоја екстремна точка во DoG (Difference of Gaussian) сликите, со цел да се добие поточно (субпикселско) позиционирање на клучната точка. Потоа, се елиминираат клучните точки со низок контраст, бидејќи тие не носат доволно информација и може да предизвикаат грешки при споредување. Исто така, се отфрлаат клучните точки кои се наоѓаат долж рабови (на пример, на остар прекин помеѓу две зони), бидејќи на тие места клучната точка е слабо дефинирана во една насока и лесно може да се помести при мали промени на сликата.



Сл. 3: Филтрирање на клучни точки во SIFT

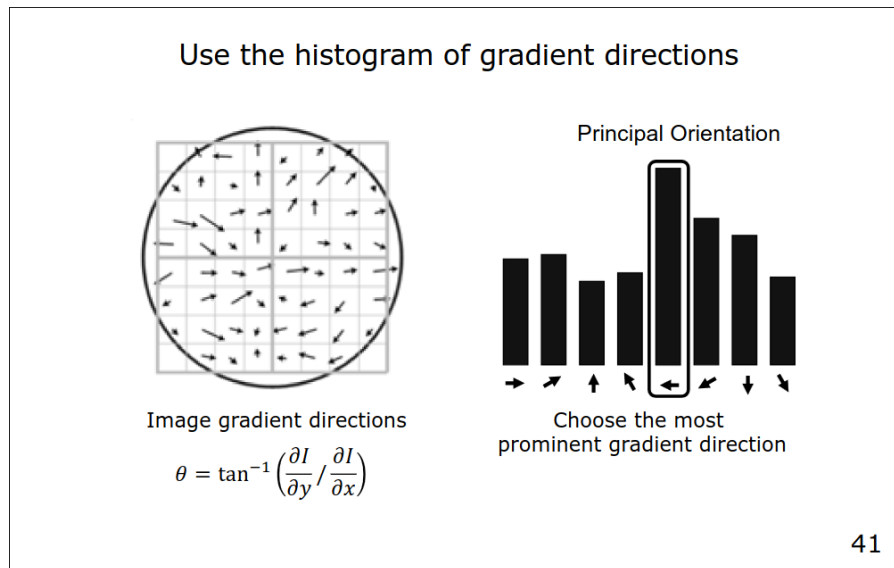
Резултат: стабилни точки од интерес

### Доделување на ориентација:

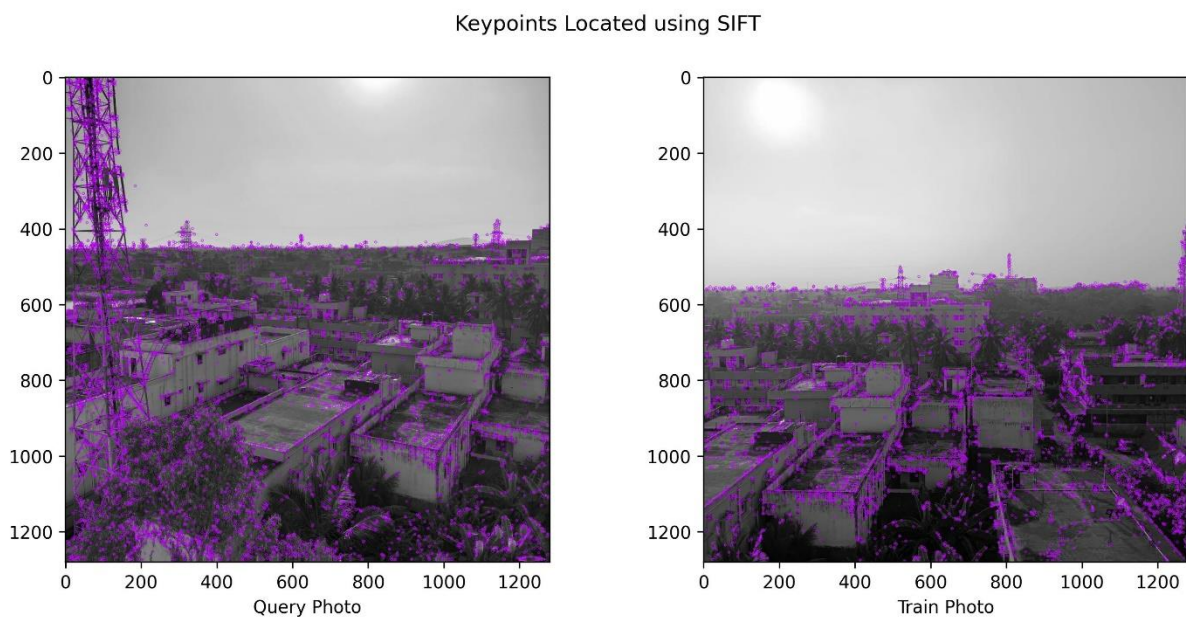
За секоја клучна точка, се пресметува јачината и насоката на градиентот во нејзината околина. Потоа се креира хистограм на ориентации за да се одреди доминантната насока. На клучната точка ѝ се доделува ориентација врз основа на врвовите на хистограмот.

Со тоа што ја анализира ориентацијата на градиентот околу клучната точка, SIFT обезбедува дека дескрипторот ќе биде инваријантен на ротација.

Информацијата за градиентите потоа се користи за да се изгради дескриптор кој ја опфаќа локалната структура околу клучната точка.



Сл. 4: Ориентација на градиенти



Сл. 5: Клучни точки кои се пронајдени во влезните слики на нашиот алгоритам, користејќи SIFT

## Спојување на клучни точки

При споредувањето на карактеристиките, примарната цел е да се воспостават кореспонденции помеѓу клучните точки откриени на различни слики. Овој процес е фундаментален во задачи како што се спојување на слики, препознавање на објекти и 3D реконструкција.



## Метрика на растојание

Клучните точки се карактеризираат со дескриптори, кои се вектори на карактеристики што ги претставуваат локалните информации за сликата околу секоја клучна точка. Вообичаен пристап е да се користи метрика на растојание, како што е Евклидовото растојание, за да се измери сличноста или различноста помеѓу дескрипторите на две клучни точки. Помалите растојанија означуваат поголема сличност.

Нормата L2 се пресметува со помош на Евклидовата формула за растојание, која е квадратен корен од збирот на квадратните разлики помеѓу соодветните елементи на два вектори.

Да ги означиме векторите на дескрипторите на две клучни точки како  $v_1$  и  $v_2$ , а нивното соодветно L2 (Евклидово) растојание како  $d_{L2}$ :

$$d_{L2}(v_1, v_2) = \sqrt{\sum_{i=1}^n (v_{1i} - v_{2i})^2}$$



Сл. 6: Совпаѓање на клучни точки во две од влезните слики во нашиот алгоритам

## Пресметување на матрицата на хомографија

Матрицата на хомографија, често означена како  $H$ , е трансформациона матрица што се користи во компјутерската визија за да претставува проективна трансформација помеѓу две слики од иста рамнинска површина. Матрицата на хомографија го опишува геометрискиот однос помеѓу соодветните точки во двете слики.

Матрицата на хомографија е матрица 3x3 и може да се претстави како:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

Равенка за трансформација на хомографија:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

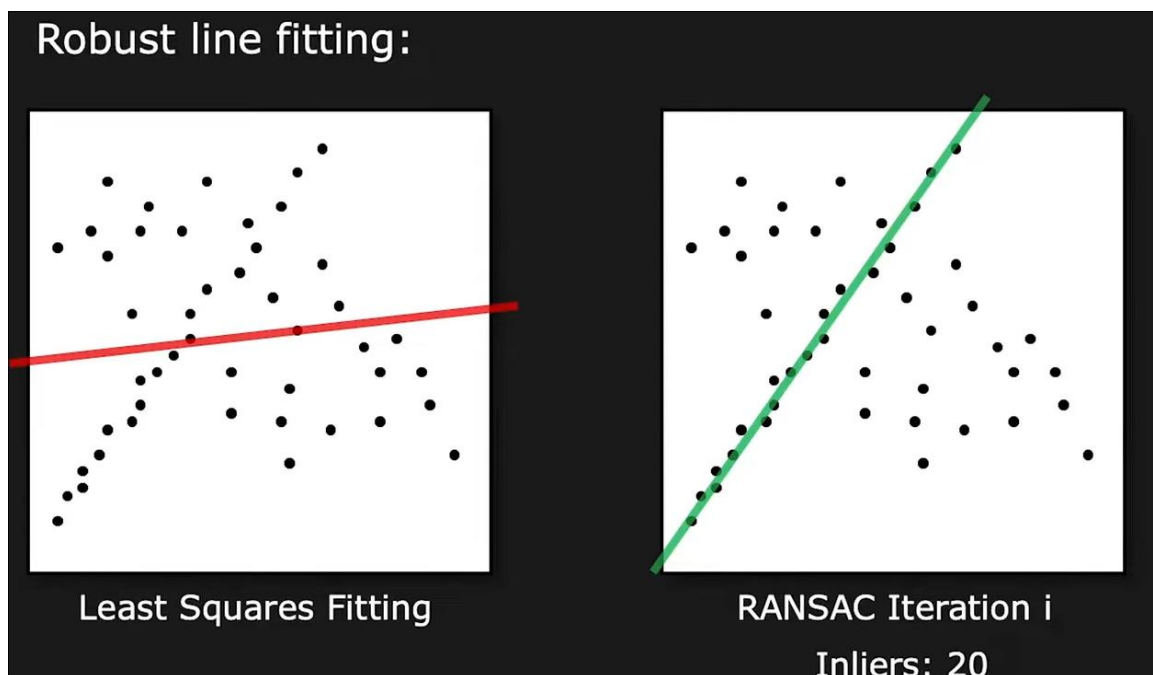
Помага во правилно усогласување и трансформирање на сликите, осигурувајќи дека соодветните точки на различни слики се мапирани на истите координати во конечната панорама.

## RANSAC алгоритам

RANSAC (Random Sample Consensus) е робустен алгоритам кој се користи за проценка на хомографска матрица помеѓу две слики врз основа на соодветни карактеристики (keypoints). Алгоритмот работи во неколку итеративни чекори:

1. Случајно се избираат 4 кореспондирачки точки (најмал број потребен за хомографија).
2. Се пресметува хомографската матрица врз основа на избраните точки.
3. Се пресметува бројот на inliers – точки што добро се вклопуваат во моделот со грешка помала од зададен праг.
4. Процесот се повторува N пати за да се зголеми веројатноста за проналаоѓање на оптимален модел.
5. Како финален модел се избира оној што има најголем број inliers.

Овој пристап овозможува елиминација на аномалии и неточни совпаѓања (outliers), обезбедувајќи стабилна и точна проценка на трансформацијата помеѓу сликите.



Сл. 7: RANSAC робустност

Забелешка: Бројот на отстапувања треба да биде  $< 50\%$  за да функционира RANSAC.

## Проблемот на видливи рабови при спојување слики и примената на тежинско спојување

Во процесот на автоматско генерирање на панорамски слики, едно од најчестите предизвици е појавата на видливи рабови (seams) во областите каде што се преклопуваат повеќе слики. Овие артефакти значително го нарушуваат визуелниот квалитет и го намалуваат степенот на реалистичност на финалната панорама. Појавата на овие рабови најчесто се должи на:

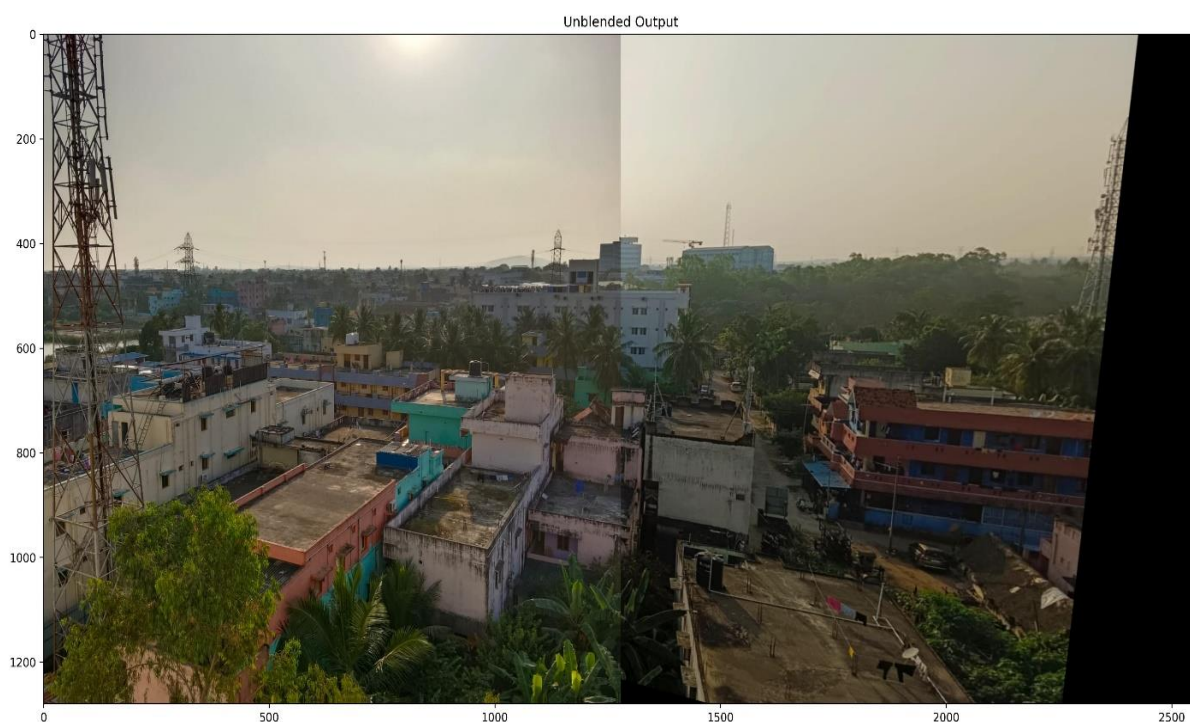
- Разлики во експозиција – варијации во осветлувањето како резултат на различни поставки на камерата (време на експонирање, ISO чувствителност).
- Хетерогено осветлување на сцената – природни или вештачки извори на светлина што предизвикуваат различна осветленост на различни делови од сликите.

Во вакви услови, едноставно спојување преку аритметичка средина на пикселите во преклопените региони не е доволно. И покрај тоа што просечното блендирање го ублажува контрастот меѓу сликите, рабовите остануваат визуелно перцептивни, особено при поголеми разлики во боја, осветлување или текстура.

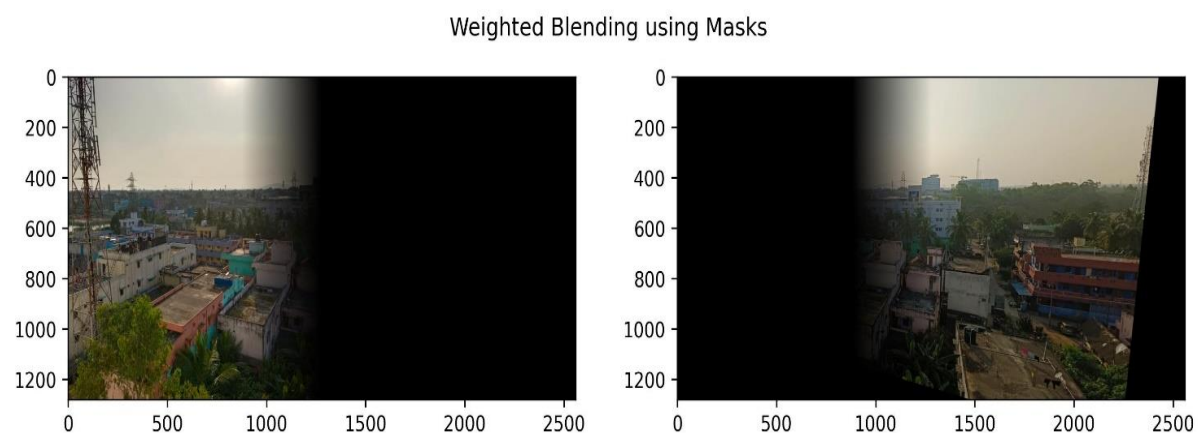
За надминување на овие ограничувања, во практиката се применува тежинско блендирање (weighted blending) – техника што овозможува постепено и визуелно неприметно преминување од една слика во друга. Овој метод користи тежински маски, со кои се контролира придонесот на секоја слика во преклопените области. Маските дефинираат нумерички вредности (тежини) што го одредуваат влијанието на соодветните пиксели при финалното комбинирање.

Тежинските маски обично се претставени како грејскејл (или бинарни) слики, каде што секој пиксел има вредност помеѓу 0 и 1, што ја рефлектира неговата важност во процесот на мешање. Пикселите во централните делови на сликата најчесто имаат повисока тежина, додека оние блиску до рабовите, особено во зоните на преклоп, имаат помала тежина. Врз основа на овие вредности, се врши пондерирано сумирање на интензитетите на пикселите од различни слики.

Предноста на овој пристап е во тоа што создава континуиран и непрекинат премин помеѓу споените слики, со што се минимизираат остри граници и се зголемува визуелната конзистентност.



Сл. 8: Излезна слика добиена без тежинско спојување



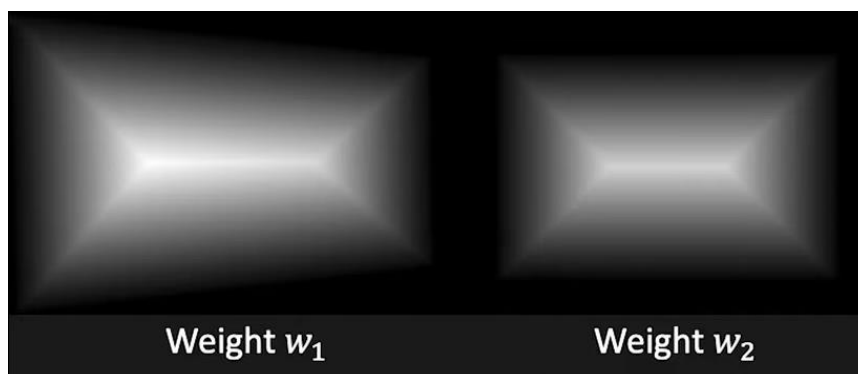
Сл. 9 : Тежинско спојување со користење на маски на двете влезни слики

Забелешка: маската на лева слика има вредност 1-0, додека маската на десната слика има вредност 0-1



Сл. 10 : Формула за тежинско спојување

Маската треба да има исти димензии како и сликата, а на секој пиксел во маската му е доделена вредност на тежина помеѓу 0 и 1. Потоа маската се нормализира за да се одржи целокупниот интензитет на бојата и осветленоста за време на мешањето.



Сл. 11: Споредба на двете тежини

Целта на тежинското мешање е да се создаде беспрекорна транзиција помеѓу преклопувачките региони на сликите, земајќи ја предвид релативната важност или придонесот на секој пиксел, овозможувајќи мазен преод и контрола на транзиција во преклопувачките региони.

## Рекурзија за спојување на повеќе слики

Во системите за создавање на панорами од повеќе слики, клучно е да се има алгоритам кој може да работи со произволен број на влезни слики, без потреба од рачна интервенција или тешко одржлив код. Примената на рекурзија во вакви ситуации претставува елегантен и скалабилен пристап кој ја поедноставува целата логика на обработка.

Главната предност од користењето на рекурзивна функција е тоа што процесот на спојување се врши чекор по чекор, почнувајќи од крајот на листата, со што секогаш се комбинираат само две слики истовремено. Овој начин овозможува постепено градење



на конечната панорама, што го прави алгоритмот полесен за одржување, надградба и тестирање.

Дополнително, ваквиот пристап ја подобрува флексибилноста на апликацијата: независно дали корисникот внесува две, три или десет слики, алгоритмот ќе може автоматски да ги обработи без потреба од промена во кодот. Тоа е особено важно кога се работи со динамички влезови или во ситуации каде бројот на слики не е познат однапред.

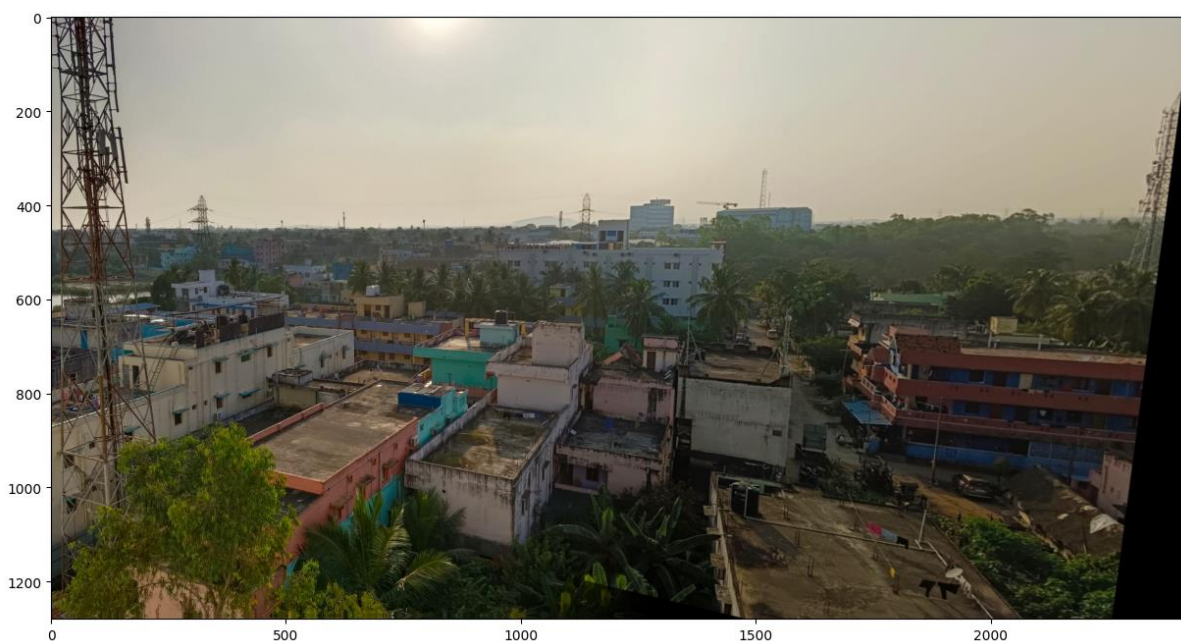
## Отстранување на вишок простор добиен при геометриски трансформации

Потребно да се извршат геометриски трансформации врз дел од сликите — како што се транслација, скалирање или перспективна корекција — за правилно порамнување и усогласување на содржината. Ваквите трансформации овозможуваат прецизно поклопување, но воедно водат до појава на црни празнини околу рабовите на резултантната слика, кои претставуваат пиксели без реална содржина.

За справување со овој проблем, се применува процес за автоматско отстранување на вишок простори, кој е особено корисен кај трансформирани слики. Алгоритмот работи така што ги идентификува сите пиксели што содржат валидни податоци (различни од нула) и врз основа на нив ги пресметува границите на реалната слика. Потоа, го прикажува само на релевантниот регион, при што се отстрануваат сите празни делови околу панорамата.

Благодарение на тоа, финалниот резултат секогаш е чист, компактен и визуелно квалитетен — без црни рабови, без потреба од рачно прилагодување.

Ова го прави алгоритмот робустен и практичен за реална примена, каде условите на снимање и аглите не можат секогаш да се контролираат прецизно.



Сл. 12: Резултат со вишок простор

# Имплементирани алгоритми и проектна структура

Проектот за креирање панорамски слики е структуриран на модуларен начин и изгледа вака:

stitch/

```
|
|
| — image_stitching/      # Главен модул со класи и функции
|   | — image_stitching.py  # Главна класа за Stitch
|   | — init.py
|   | — read_images.py      # Функција за читање на слики
|   | — recursion.py        # Рекурзивна stitch логика
|   | — utils.py            # Pipeline за спојување на 2 слики
|
| — inputs/                # Влезни слики
|   | — back/               # Комплет за panorama - back
|   |   | — back_01.jpeg
|   |   | — back_02.jpeg
|   |   | — back_03.jpeg
|   | — front/              # Комплет за panorama - front
|   |   | — front_01.jpeg
|   |   | — front_02.jpeg
|   |   | — front_03.jpeg
|
| — outputs/                # Излезни резултати
|   | — mapped_image.jpg    # Слика со SIFT совпаѓања
|   | — panorama_image.jpg  # Финална панорамска слика
|
| — panorama.py             # Главна скрипта за стартување на проектот
```

Во овој проект за автоматско спојување панорамски слики е имплементиран систем базиран на класични и проверени алгоритми од областа на компјутерски вид. Главната логика се заснова на идентификација на карактеристики, нивно совпаѓање и изведување на прецизна геометриска трансформација за успешно обединување на повеќе слики во една. Секој дел од кодот е организиран во модули кои комуницираат еден со друг преку прецизно дефинирани функции и класи.

Основата на проектот се наоѓа во фајлот `image_stitching.py`, каде што е дефинирана класата `ImageStitching`. Таа е одговорна за претворање на сликите во `grayscale`, детекција на клучни точки со SIFT алгоритмот, пронаоѓање на совпаѓања помеѓу тие карактеристики, пресметка на хомографија користејќи RANSAC, и на крај изведување на мазно спојување преку `Gaussian blending`. По спојувањето на сликите, добиената панорама често содржи црни празнини околу рабовите, како резултат на геометриски трансформации. За да се добие почист и покомпактен резултат, се применува автоматско сечење на сликата. Прво се идентификуваат координатите на сите пиксели што не се нула (т.е. што припаѓаат на вистинската содржина), потоа се пресметуваат минималната и максималната вредност по редови и колони, и се сече само регионот што ја содржи реалната слика. На тој начин се елиминираат непотребните црни области и се подобрува визуелната и просторната ефикасност на излезот. Оваа класа е централниот дел на `pipeline`-от и сите поврзани функции комуницираат со неа за да ја завршат задачата.

Покрај неа, модулот `read_images.py` има улога да ги вчита сликите од кориснички внесени патеки, и да ги конвертира во `RGB` формат. Овие слики потоа се проследуваат до `recursion.py`, каде што се имплементира стратегијата за рекурзивно спојување на повеќе слики. Ако се внесат повеќе од две слики, функцијата `recurse` ги спојува последните две, го заменува резултатот во листата, и продолжува да работи со следната слика, сè додека не се добие единствена целосна панорама.

Целта оваа логика е поврзана преку `utils.py`, кој ја дефинира функцијата `forward`. Таа во себе го инкапсулира целиот процес: од обработка на сликите, до финалната трансформација и изработка на мазна, реалистична слика. `forward` користи класа од `image_stitching.py` и претставува клучна алка што ги поврзува сите чекори од `pipeline`-от во еден интегриран процес.

На крајот, фајлот `panorama.py` претставува главната контролна скрипта што ги повикува сите овие модули. Тој ги собира патеките до сликите преку аргументи од командна линија, ги вчитува сликите, ја активира рекурзивната логика и ги снима добиените резултати. Дополнително, визуелизацијата на клучните точки и совпаѓањата се чува како слика, што е корисно за анализа на квалитетот на спојувањето.

Проектот е организиран така што секој модул има јасна и единствена улога, а комуникацијата помеѓу нив е логично поврзана. Овој пристап овозможува модуларност, лесно дебагирање и проширување на системот. Алгоритмите меѓусебно зависат во точно определен редослед, при што секој нареден чекор зависи од резултатите на претходниот – од читање слики, преку анализа на содржина, до конечно визуелно и технички исправно обединување во една панорамска слика.



# Резултати

Влезни слики:



Сл. 13: Прва влезна слика



Сл. 14: Втора влезна слика



Сл. 15: Трета влезна слика

Генерирана Панорама:

Слика генерирана со спојување на 3 фотографии.

Панорамската слика е со висока прецизност, без видливи прекини, со добро усогласување и мазни рабови.



Сл. 16: Панорама

## Анализа

При анализа на изведбата и перформансите на имплементираниот систем за панорамско спојување слики, беа земени предвид неколку клучни метрики:

Брзината на процесирање беше на високо ниво – за комплетно спојување на три слики, времето на извршување изнесуваше помалку од 5 секунди на стандардна машина. Ова покажува дека системот е ефикасен и погоден за практична употреба.

Робусноста на методот се покажа како силна страна – алгоритмот успешно ги спои сликите и во ситуации кога тие беа изместени, вертикално поместени или снимени од благо различни агли. Ова е постигнато благодарение на употребата на SIFT и RANSAC, кои се отпорни на геометриски трансформации и шумови во податоците.

Квалитетот на финалното спојување беше визуелно многу добар. Благодарение на техниката на Gaussian blending со маски, транзициите меѓу сликите беа мазни, без видливи рабови, промени во осветлување или дуплирање на објекти. Финалната панорама изгледаше природно и кохерентно.

Овие карактеристики го прават системот стабилен, флексибилен и применлив за реални апликации каде што е потребна автоматска реконструкција на широки сцени.

## Заклучок

Во оваа семинарска работа успешно беше реализиран комплетен систем за автоматско создавање панорамски слики со користење на повеќе влезни фотографии со делумно преклопување. Главната цел беше не само да се добие визуелно точна и непрекината панорама, туку и да се имплементираат и искористат некои од најмоќните алгоритми од областа на компјутерски вид и дигитална обработка на слики.

Преку модуларната архитектура на проектот, имплементацијата беше поделена на повеќе логички целини: читање на слики, пресметка на клучни точки и нивно совпаѓање, проекција на слики во заеднички координатен систем, и финално – нивно мазно и реалистично спојување преку Gaussian blending.

Особено значајна беше употребата на SIFT (Scale-Invariant Feature Transform) алгоритмот, кој овозможи детекција на клучни точки кои се отпорни на ротација, промена на размер и светлина. Благодарение на тоа, системот постигна висока стабилност при совпаѓање на детали меѓу сликите. Понатаму, со примена на Brute Force Matcher и RANSAC, беа филтрирани несоодветните совпаѓања и беше пресметана хомографија, која овозможи прецизно трансформирање на сликите во заедничка рамнина.

Дополнително, во фазата на спојување беше користена техника на Gaussian маскирање и blending, што резултираше со мазни преоди меѓу сликите и отстранување на видливи рабови и нагли промени во осветлувањето. Тоа е особено важно кога сликите се сликани под различни услови на експозиција и агли.

Користејќи рекурзивна логика, беше овозможено автоматско спојување на произволен број слики, што го прави системот скалабилен и применлив во реални сценарија – од туристичка фотографија до мапирање на околина за автономни возила.

Тестирањето на системот со реални слики покажа стабилни резултати, дури и кога сликите беа изместени, ротирани или поместени вертикално. Финалната панорама беше визуелно кохерентна, без видливи неправилности и со прецизно совпаѓање на објектите.

## Користена литература

David Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, IJCV, 2004

Richard Szeliski, “Image Alignment and Stitching: A Tutorial”, 2006

OpenCV Python Documentation – <https://docs.opencv.org>

PyImageSearch: Stitching images with OpenCV and Python