



Универзитет „Св. Кирил и Методиј“ во Скопје  
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И  
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

# ДЕТЕКЦИЈА НА ИРИС

Обработка на слика

## Abstract

Ирисот е обоениот дел од окото, кој има форма на крофна и се наоѓа позади рожницата, заобиколувајќи ја зеницата. Во денешно време важи за една од најстабилните и најкористените биометрички карактеристики. Во оваа семинарска работа е опишана токму примената на ирисот како алатка за споредување на две личности користејќи првично алгоритам за локализација на ирисот, по што следува споредување на двата локализирани региони од две личности соодветно со користење на SIFT, ORB, FAST и BRISK дескриптори.

Ана Ангелевска  
Андреј Наумовски  
Скопје, 2020

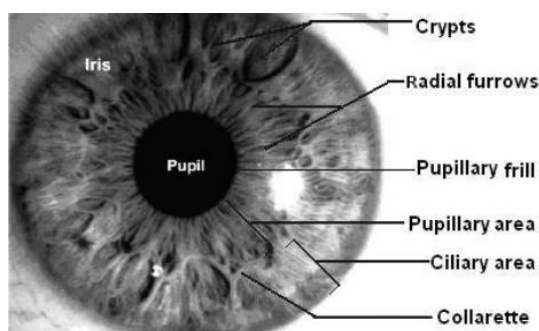
## Содржина

Вовед .....	2
Сродни истражувања .....	2
Аквизиција на фотографии .....	3
Сегментирање на ирис .....	4
CLANE .....	4
Локализација на зеница .....	5
Нормализација на сегментиранiot регион .....	7
Методи за генерирање на дескриптори .....	8
SIFT .....	8
FAST .....	10
ORB .....	11
BRISK .....	12
Алгоритми за совпаѓање .....	13
Brute Force .....	13
FLANN .....	14
Резултати .....	16
Заклучок .....	16
Користена литература .....	17

## Вовед

Во денешно време, потребата за доверлива идентификација на корисникот расте како резултат на зголеменото складирање на се повредни и повредни податоци. Биометриката, како наука која користејќи некоја од соодветните карактеристики на човекот идентификува една и единствена личност, зема се поголем замав во индустријата. Меѓу биометричките карактеристики на човекот како што се отпечатоци од прсти, отпечаток од дланка, ретина, лице, потпис како и начинот на движење, ирисот дава најголема прецизност при идентификација на личности од погоренаведените методи. Уште повеќе, бидејќи текстурата на ирисот има висок степен на слобода, веројатноста да се најдат два исти ириси е многу блиску до нулата, вклучувајќи ириси на близнаци како и ириси од лево и десно око на ист човек. Текстурата на еден ирис е потполно формирана до старост од 10 месеци и останува иста до крајот на животот, за разлика од другите карактеристики за идентификација кои можат да трпат промени во текот на животот. Како резултат на тоа, системите базирани на препознавање на ирис се многу доверливи и се користат во некои од најобезбедените места.

Слика 1. Структура на ирис



Сепак, како и секој метод, и препознавањето на ирис си има свои недостатоци. Еден недостаток е цената. Препознавањето на ирис е еден од поскапите методи, поради цените на специјалните скенери кои се користат. Друг недостаток е тоа што се користат скенери со инфрацрвени зраци, кои на долгорочен план може да бидат штетни за окото. Исто така друг проблем е процесот на скенирање, односно, треба да постои конзистентна средина во која ирисите ќе бидат скенирани, што значи дека секоја дополнителна рефлексија или попречување на ирисот од трепки или очни капаци може да влијае врз резултатот.

Еден карактеристичен систем за препознавање на ирис се состои од 4 фази: Аквизиција на фотографии, Сегментација на ирисот, Нормализација на сегментиратион регион, Наоѓање и поврзување на карактеристики или дескриптори од соодветните нормализирани фотографии.

## Сродни истражувања

Поради обемот на индустријата, постојат разновидни алгоритми за комплетирање на некој процес од препознавањето на ириси, па различни трудови користејќи различни методи базирани на оваа тематика целат кон слични резултати. Тука се опишани неколку трудови кои имаа влијание врз нашето решение.

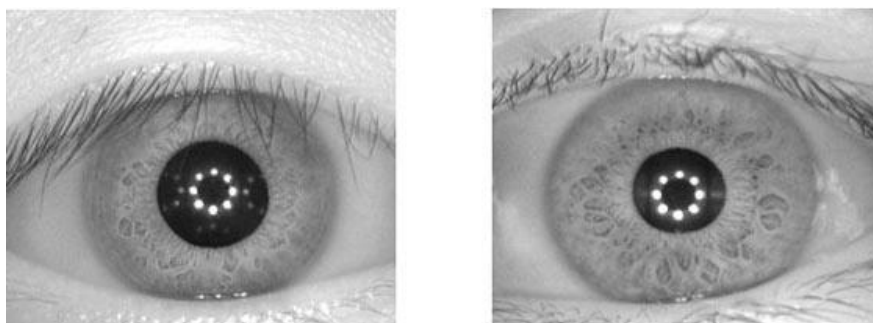
***New Methods in Iris Recognition, John Daugman*** - Овој труд, напишан од човекот кој има голема улога во препознавањето на ириси, John Daugman, предложува неколку напредни алгоритми за локализација на ирисот. Тука се предложени методи за локализација кои би работеле и на нецентрализиран ирис, односно локализација на ирисот кога погледот е кон некоја точка настрана од камерата, за разлика од нашиот метод кој би работел само на централизиран ирис. Исто така, за разлика од нас, користи методи за детекција и отстранување на трепките.

***DeepIris: Iris Recognition Using A Deep Learning Approach, Shervin Minaee and Amirali Abdolrashidi*** и ***Iris Recognition with Off-the-Shelf CNN Features: A Deep Learning Perspective*** од ***Kien Nguyen, Arun Ross*** - Во овие трудови, покрај локализирањето и нормализирањето на ирисот, наместо детектирање на дескриптори и соодветно поврзување на тие дескриптори, користат конволуциски невронски мрежи кои како влезови ги примаат тие нормализирани ириси а на излез го даваат идентитетот на личноста. Ова, секако, има свои недостатоци и предности спрема методот со дескриптори кој ние го користевме. Уште повеќе, начините на кои овие два методи би се искористиле во некој систем би бил различен.

## Аквизиција на фотографии

[CASIA V4](#) е најновата верзија на CASIA датабазите. Се состои од вкупно 54,607 фотографии од ириси од повеќе од 1800 луѓе. Базата е поделена на 6 делови: CASIA-Iris-Interval, CASIA-Iris-Twins, CASIA-Iris-Thousand, CASIA-Iris-Lamp, CASIA-Iris-Distance, CASIA-Iris-Syn. Сите овие подмножества се разликуваат помеѓу себе.

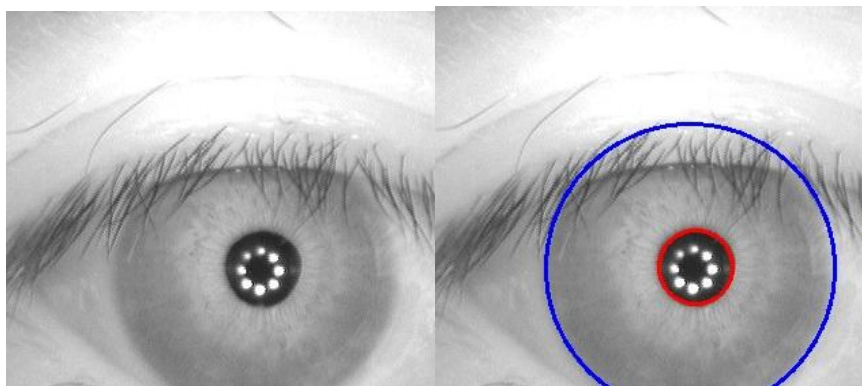
За целта на овој проект се користат фотографиите кои се дел од подмножеството CASIA-Iris-Interval. Множеството се содржи од леви и десни ириси за 250 личности. Сите фотографии се 8 bit gray-level фотографии и имаат резолуција од 320 x 280. Фотографиите се фотографирани со специјална камера за фотографирање на ирис изградена од центарот за истражување во биометрика и обезбедување(CBSE), која во средината содржи круг од ЛЕД диоди со цел појасни текстури на ирисот на излезните фотографии.



Слика 2: Пример на фотографии од ирис во CASIA-Iris-Interval

## Сегментирање на ирис

Сегментирањето на ирисот е битен чекор во овој систем бидејќи сите наредни процесирачки чекори над фотографијата зависат од него. Тоа е процес во кој се лоцираат и детектираат кружните граници како за зеницата така и за ирисот со цел да се земе оној дел од окото во кој се наоѓа помеѓу двете граници



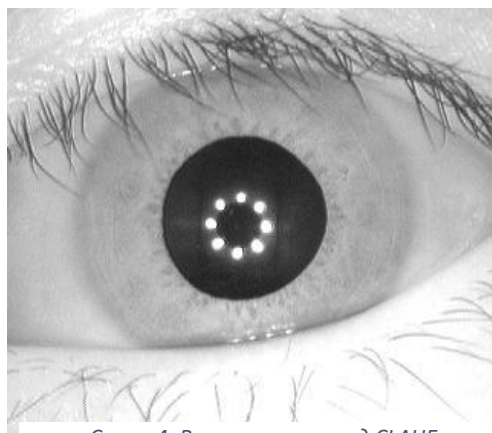
Слика 3. Влез и излез од овој чекор

## CLAHE

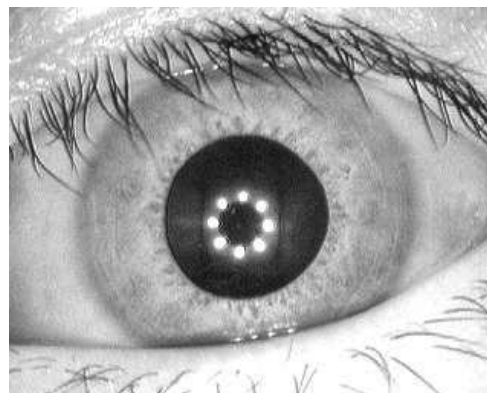
Пред да се почне со чекорите за сегметирање на ирисот, се извршива мало претпроцесирање на влезните фотографии. Тоа се прави со цел резултатите кои ќе се добиваат во понатамошните чекори (локализацијата) да бидат поконзистенти.

За оваа цел се користи CLAHE, кој е варијанта на адаптивно изедначување на хистограмот (AHE) која се грижи за прекумерно засилување на контрастот. CLAHE наместо на целата слика, работи на мали региони на сликата, наречени плочки. Соседните плочки потоа се комбинираат со помош на билинеарна интерполација за отстранување на вештачките граници.

Овој алгоритам го користевме за да го подобриме контрастот на сликите.



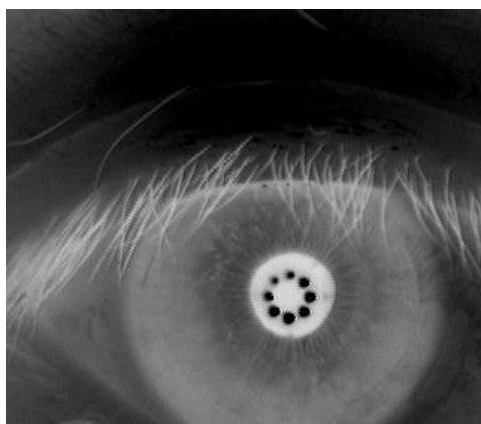
Слика 4. Влезна слика пред CLAHE



Слика 5. Влезната слика по воведување на CLAHE

## Локализација на зеница

Како што може да се забележи на фотографиите погоре, зеницата е црниот круг кој се наоѓа на средината од окото, па нашето решение е донекаде интуитивно. Она што прво се прави е поставување на threshold на инверзната фотографија, односно, прво влезната фотографија се заменува со инверзната, а потоа се поставува threshold со праг 190. Ова ќе направи сите пиксели со вредност над 190 да добијат вредност 255(потполно бел пиксел), а оние под 190 да добијат вредност 0(потполно црн пиксел). Инверзијата се извршува бидејќи контурите кои подоцна се бараат работат на принцип кој побарува објектот врз кој се бараат тие контури да е бел и да се наоѓа на црна површина.



Слика 6. Инверзна фотографија



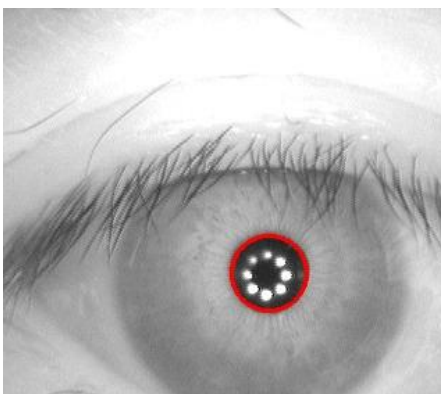
Слика 7. Применет thresholding

Како што може да се види на бинарната фотографија, примената на thresholding може да остави шум, кој во зависност од првичната фотографија може да биде значително голем и да предизвика лоша локализација на зеницата. Па за таа цел ја искористивме морфолошката трансформација наречена **ерозија**. Она што ерозијата го овозможува е со користење на кернел со променлива големина(во нашиот случај 4x4) да ги намали белите површини на сликата, а во овој случај да го исчисти шумот.

Кога се доби фотографија кај која е донекаде јасно изразена зеницата, со примена на методи за пронаоѓање на контури лесно се добива кругот кој ја претставува зеницата.



Слика 9. Применета морфолошка



Слика 8. Успешно детектирана зеница

За разлика од локализацијата на зеницата, поради неконзистентоста на бојата на ирисот, не можевме конзистентно да го локализираме само со употреба на thresholding, па го употребивме методот Hough Circle Transform за детектирање на кружници.

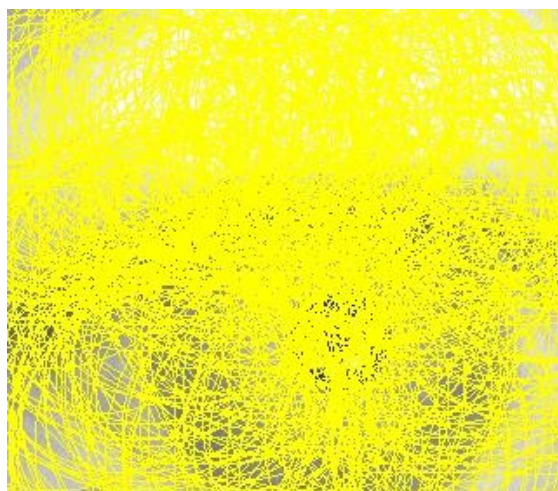
Во дводимензионален простор, кругот е претставен со следната равенка.

$$(x - a)^2 + (y - b)^2 = r^2,$$

каде точката (a,b) претставува центар на кругот, а r е радиусот.

Целта на овој метод е со употреба на Canny алгоритмот за детекција на рабови да пронајде здружени рабови кои формираат круг со некој предефиниран радиус. Работи на начин што за секоја точка од рабовите генерира кругови со соодветен предефиниран радиус и пресечната точка на сите овие кругови го зема за центар на вистинскиот круг.

Во овој конкретен проблем, со низа од експерименти одредивме некоја граница за максимален и минимален радиус на ирисот, па со примена на Hough Circles со овие граници се добива следниот резултат:



Слика 10. Сите детектирани кругови на фотографијата

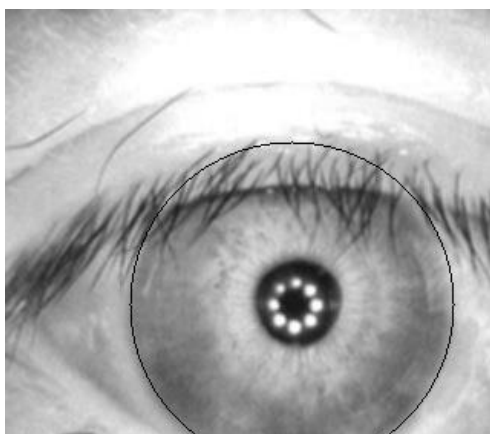


Како што може да се забележи, детектирани се преголем број на кругови, а нам ни треба кругот кој го опишува само ирисот. За таа цел, од сите овие кругови, како краен се зема кругот кој има најмалото растојание до центарот од претходно детектираната зеница. Растојанието помеѓу двата центри се пресметува со Евклидовото растојание.

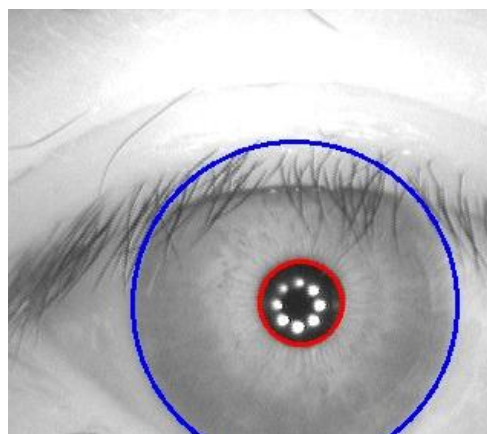
$$D = \sqrt{((Ix - Px)^2 + (Iy - Py)^2)},$$

каде зеницата има центар  $(Px, Py)$ , а ирисот  $(Ix, Iy)$ .

Како резултат на ова, се добива следниот локализиран ирис, како и потребната целосно сегментирана фотографија.



Слика 11. Локализиран ирис



Слика 12. Целосно сегментиран ирис

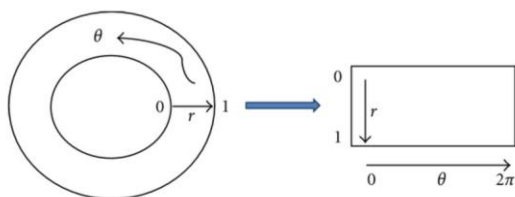
Овој пристап на детектирање на најблискиот круг до центарот на зеницата, покажува доста добри резултати.

## Нормализација на сегментираниот регион

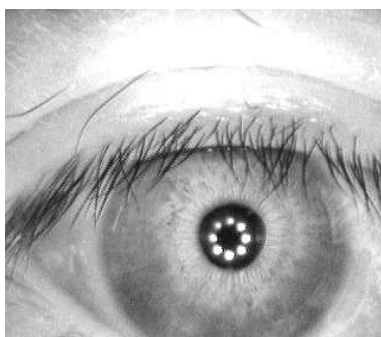
Откако се детектирани границите на ирисот и зеницата, она што се прави со делот помеѓу се нарекува **нормализација**. Тоа подразбира конвертирање на ирисот во стандардна, предефинирана големина и потоа енкодирање во еден стандарден темплејт (шема) за поедноставно поврзување на карактеристиките подоцна. Всушност, ова конвертирање подразбира пренос на пикселите од еден Декартов координатен систем во не-концентрична поларна репрезентација.

Алгоритмот кој ние го искористивме е следниот. Прво се креира празна слика со големина **радиус на ирис – радиус на зеница x 360**, а потоа со итерирање низ оваа празна слика, се пресметува кој е соодветниот пиксел во оригиналната фотографија со помош на тригонометриски функции над аголот и радиусот.





Слика 13. Одвиткување на ирис“



Слика 15. Влезната слика пред нормализација



Слика 14. Влезната слика по нормализација

Еден проблем што тука се појави, како што може да се забележи на сликата погоре, е кругот кој го опишува ирисот (надворешниот круг) може да излегува од оригиналната фотографија. Бидејќи информацијата која се наоѓа надвор од фотографијата нема никаква улога, пикселите од кругот кои би се наоѓале надвор, се заменуваат со црни пиксели, со едноставна проверка во нормализацијата.

Еден од другите начини на нормализација или одвиткување на ирисот кои се предложени од слични студии е Daugman-овиот Rubber Sheet модел кој работи прилично слично со алгоритмот кој ние го употребивме.

## Методи за генерирање на дескриптори

### SIFT

SIFT или Scale Invariant Feature Transform е воведен од страна на David Lowe во 1999 година. Овој пристап има за цел да ги трансформира податоците кои се содржат во одредена фотографија во scaleinvariant (непроменливи) координати кои соодветствуваат на локални карактеристики. Тоа значи дека дури и ако сликата е ротирана, можеме да ги најдеме истите агли. Поради тоа пристапот ги покрива проблемите со ротација на сликата, скалирање, промена на гледаната точка и афин деформации, промена во 3D гледна точка и присуство на шум. Како резултат на сето ова, дури и за многу мали објекти може да се генерираат голем број на карактеристики. Иако SIFT е стар повеќе

од една деценија, сепак се уште наоѓа широка примена во различни области како на пример, панорамска фотографија, детекција на предмети, следење итн.

Четири главни фази на пресметка со SIFT се: scale space extrema, локализација на клучни точки, определување на ориентација и опишување на клучните точки (креирање дескриптори).

Во првата фаза на овој пристап се прави идентификација на кандидатните клучни точки. Каскадно филтрирање се користи за да се идентификуваат кандидатските локации кои во следните фази се испитуваат подетално. Со цел да се откријат локациите кои остануваат непроменети во однод на промените на обемот, се применува континуирана функција позната како scale space (простор на обем). За подобрување на пресметковната брзина се користат The Difference of Gaussian (DoG) функции (Слика 16). Просторот на обемот е функцијата која е дефинирана како конволуција на Гаусов кернел со влезната слика.

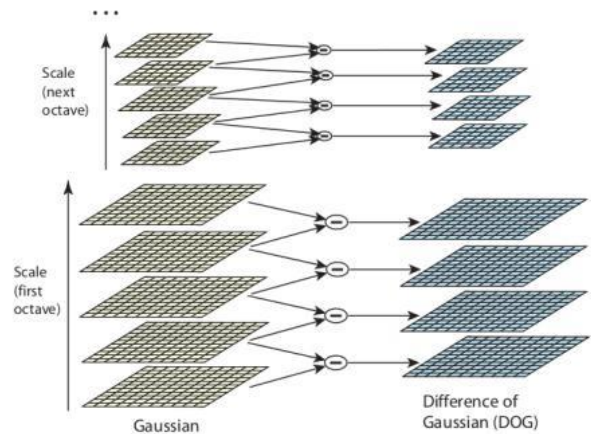
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Со овој процес се субсемплира Гаусовата слика со фактор 2 во секоја октава (процесот се повторува повеќепати). DoG сликите се генерираат како резултат на два обеми, блиски еден до друг, одделени со константен мултипликативен фактор  $k$ .

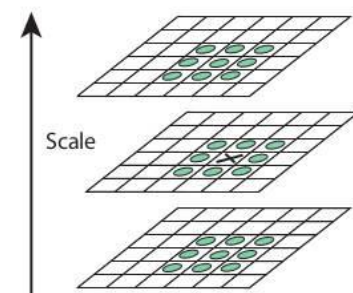
$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Секој пиксел во DoG сликите се споредува со неговите 26 соседи 3x3 региони на обем над и под (Слика 12). Клучните точки со слаб контраст се бришат. Исто така и оние клучни точки кои се детектирани по рабовите на сликата се елиминираат. Кандидатната клучна точка (обележана со X на Слика 17) се избира ако пикселот е локален максимум или локален минимум. Карактеристиките на клучната точка се мерат со нејзината ориентација, со што се обезбедува непроменлива (invariance) ротација.

Следна цел е да се најдат локацијата, обемот и соодносот на главните закривувања. Според Brown and Lowe се користи 3D квадратна функција врз локалните примероци (точки). Тоа се прави за да се утврди интерполарната локација на максимумот. Хасиева матрица се користи за пресметување на главните закривувања со сооднос на прагот на локацијата и обемот (scale) на клучната точка. Квадратните коефициенти се добиваат со пресметување на изводите користејќи ги разликите добиени помеѓу соседните примероци (точки). Кандидатните клучни точки се избираат од локацијата на екстремниот дел со отфрлање нестабилната екстрема со мал контраст.



Слика 16. Гаусова разлика добиена како разлика на Гаусово заматување на дадена слика со две различни сигми. Овој процес се прави за различни октави на сликата во Гаусовата пирамида.

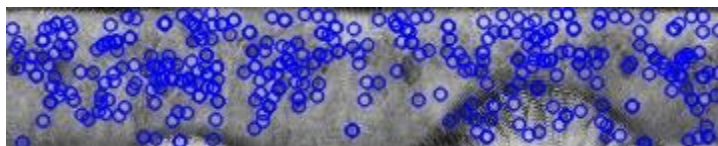


Слика 17. Детекција на клучни точки во обем-просторот (scale-space)

Ориентацијата на клучната точка се пресметува врз база на локалниот градиент на слика. Ориентациониот хистограм е формиран со 36 bins за да се покрие опсегот на ориентации од 360 степени. На секој примерок му се придружува тежина на градиент магнитуда и Гаусов тежински кружен прозорец. Врвовите на хистограмот за ориентација ја одредуваат насоката на локалните градиенти. Секој врв кој е 80% од највисокиот врв исто така се користи за создавање на клучната точка со таа ориентација. Така, постојат повеќе врвови создадени на иста локација и обем но, во различна насока. Со ова се зголемува стабилноста во процесот на совпаѓање на дескрипторите на клучните точки.

Откако ќе се додели ориентацијата, дескрипторот се пресметува како збир на хистограми на ориентација во соседството на даден пиксел. Се зема блок со големина 16x16 околу клучната точка креирана од соседните пиксели. Тој е поделен на 16 подблокови со големина 4x4. За секоја подблок се креира хистограм на ориентација со 8 bins, што значи дека има вкупно 128 достапни вредности. Од нив се формира вектор со карактеристики кој на крај може да се нормализира така што промените на големината на градиентот немаат никаков ефект врз промената на осветлувањето. На тој начин, дескрипторот на клучната точка ќе биде непроменлив на афиналните промени во осветлувањето.

На Слика 18 може да се видат клучните точки кои се генерираат со користење на SIFT методот врз една влезна слика.

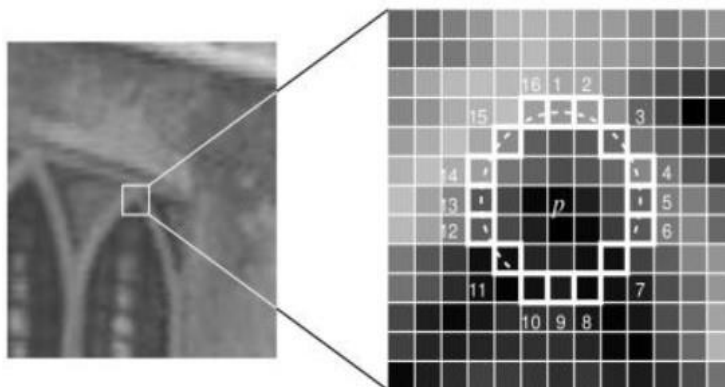


Слика 18. Генерирани клучни точки со користење на SIFT

## FAST

FAST е алгоритам кој се служи со метод на детектирање на агли со цел да детектира карактеристики од фотографијата. Познат е по својата употреба во апликации за видео процесирање во реално време баш поради причината која беше спомената погоре, односно, својата брзина и супериорен перформанс, кој исто така, со употреба на техники за машинско учење може да биде уште помокен. Накратко, работи на следниот начин.

За секој пиксел  $p$  со интензитет  $I_p$  засебно, зема во предвид круг од 16 пиксели кои се наоѓаат околу моменталниот пиксел  $p$ .

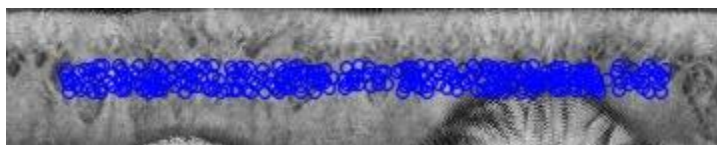


Слика 19. Приказ на работата на FAST врз еден пиксел

Земајќи го во предвид предефинираниот threshold ( $p_{par}$ )  $t$ , се обидува одлучи дали овој пиксел  $p$  е точка на интерес или не. Односно, пикселот  $p$  го класифицира како агол ако постои множество од  $n$  последователни пиксели од оние 16 во кругот, за кои важи дека се сите посветли од  $I_p + t$ , или потемни од  $I_p - t$ .

Бидејќи FAST не обезбедува методи за генерирање на дескриптори од добиените точки на интерес, корисиме BRIEF. Генерираните клучни точки од FAST алгоритмот се праќаат на BRIEF кој ќе генерира дескриптори кои потоа можат да се споредуваат помеѓу различни фотографии.

На Слика 20 може да се видат клучните точки кои се генерираат со користење на FAST методот врз една влезна слика.



Слика 20. Генерирани клучни точки со користење на FAST

## ORB

ORB дескрипторот покажува добри перформанси исто како и SIFT во многу случаи за детекција на карактеристики и може да се каже дека е подобар од SURF. Неговата работа се темели врз добро познатиот FAST детектор за клучни точки и BRIEF дескрипторот, па од таму го добива и името Oriented FAST and Rotated BRIEF. Двете главни разлики помеѓу ORB и BRIEF се:

1. ORB користи ориентациски механизам за компензација, со што се постигнува непроменета ротација (инваријантност).
2. ORB ги учи оптималните семплирани парови, додека пак BRIEF ги селектира по случаен избор.

ORB користи едноставна пресметка за ориентација на рабовите – центроид на интензитет. Прво, се дефинираат моменти (moments of patch) на следниот начин:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y)$$

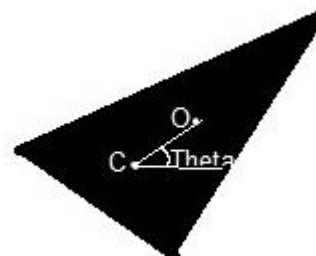
Со овие моменти може да се најде центроид, „центар на маса“ за пачот соодветно, преку следната формула:

$$C = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

Потоа се конструира вектор од работ на центарот О до центроидот ОС. Ориентацијата на пачот е зададена со:

$$\theta = \text{atan2}(m_{01}, m_{10})$$

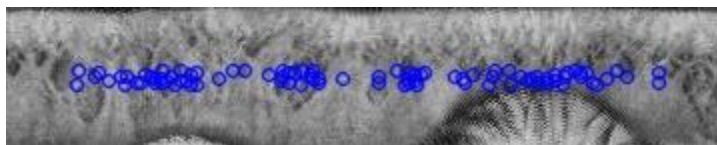
Ова може да се види на Слика 16. Откако ќе се пресмета ориентацијата на пачот, може да се ротира во канонска ротација и да се пресмета дескрипторот, задржувајќи одредена непроменливост на ротацијата.



Слика 21. ORB дескриптор – пресметување на аголот на ориентација

Сакаме нашите парови примероци да ги имаат следните две својства: некорелираност и висока варијанса. Односно, сакаме да нема корелација помеѓу паровите примероци со цел секој нов пар да дава нови информации на дескрипторот, а со тоа ќе се зголеми количината на информациони што ќе ја содржи одреден дескриптор. А од друга страна, сакаме да имаме висока варијанса што ја прави секоја карактеристика подискриминирачка. За да ги добиеме овие две својства ORB нуди учење на паровите примероци.

На Слика 22 може да се видат клучните точки кои се генерираат со користење на ORB методот врз една влезна слика.



Слика 22. Генерирани клучни точки со ORB

## BRISK

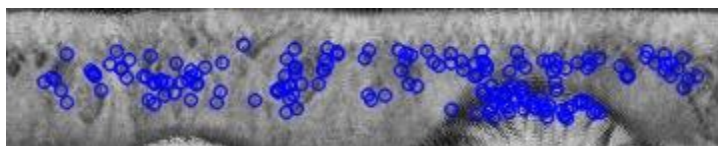
BRISK дескрипторот има предефинирана шема на семплирање како BRIEF и ORB. Пикселите се семплираат преку концентрични кругови. За секоја семплирана точка, се зема мал patch околу неа. Пред да се започне со алгоритмот тој patch се измазнува со користење на Гаусово измазнување.

Два типа на парови се користат за семплирање, кратки и долги парови. Кратките парови се оние каде што растојанието е под одреден праг (distmax), додека долгите парови имаат расојание над (distmin). Долгите парови се користат за ориентација а кратките парови се користат за пресметување на дескрипторите. BRISK постигнува инваријантност на ротација така што пресметува локален градиент на семплираните парови.

$$g(p_i, p_j) = \underbrace{\frac{(p_j - p_i)}{\|p_j - p_i\|}}_{\text{unit vector}} \cdot \underbrace{\frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|}}_{\text{gradient magnitude}}$$

Сите локални градиенти меѓу долгите паровите сумирани и  $\arctan2(g_x, g_y)$  помеѓу  $x$  и  $y$  компонентите на сумата се земаат како агол на клучната точка. Следно, само треба да се ротираат кратките парови под тој агол за да се овозможи дескрипторот да биде со поголема инваријантност на ротацијата. Дескрипторот се гради со споредба на интензитет. За секој краток пар ако првата точка има поголем интензитет од втората, тогаш се запишува е 1, во другиот случај 0 на соодветниот бит на дескрипторот.

На Слика 23 може да се видат клучните точки кои се генерираат со користење на BRISK методот врз една влезна слика.



Слика 23. Генерирани клучни точки со користење на BRISK

## Алгоритми за совпаѓање

Дескрипторите се градат со цел да се премине на следниот чекор а тоа е процесот на наоѓање точки на совпаѓање, или со други зборови *matching* помеѓу две слики (два сета на дескриптори). Во овој проект, користени два типа на алгоритми за совпаѓање од кои едниот е Brute Force, а другиот FLANN.

### Brute Force

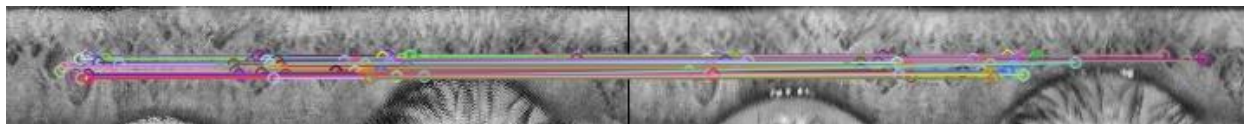
Brute Force алгоритмот работи така што зема дескриптор на една карактеристика од првиот сет и врши совпаѓање или споредба со сите карактеристики од вториот сет со користење на одредена метрика за растојание. Како резултат се враќа најблиското растојание. Метриката за растојание варира во зависност од методот кој се користи за наоѓање на дескрипторите. Поради тоа, кога ги користевме бинарно базираните дескриптори ORB и BRISK, ги комбиниравме со Хамингово растојание (`cv2.NORM_HAMMING2`), додека пак кога го користивме SIFT користевме Менхетен растојание. Во секој од случаите за совпаѓање користевме `knnMatcher`, каде бројот на соседи е поставен на 2, што означува дека 2 најдобри совпаѓања ќе се земаат во предвид што ќе резултира со две повлечени линии од еден дескриптор. Дополнителни за да се најдат само оние најдобрите совпаѓања се воведува Aspect Ratio тест. Бидејќи за секоја клучна точка имаме две совпаѓања ( $k=2$ ) со Aspect Ratio тестот се проверува дали двете растојанија се доволно различни. Ако не се, тогаш клучната точка се елиминира и нема да се користи за понатамошни пресметки. Ова се прави со познатиот тест на Lowe претставен на следниот начин:



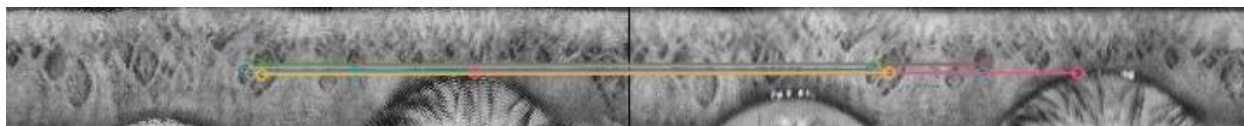
$$\text{If } distance_1 < distance_2 * C \text{ then}$$

Каде што  $distance_1$  е растојанието помеѓу клучната точка и нејзиното најдобро совпаѓање, а  $distance_2$  е растојанието помеѓу клучната точка и нејзиното второ најдобро совпаѓање. Константата  $C$  кажува колкава ќе биде границата на различност помеѓу двете совпаѓања. Во нашиот случај таа е поставена на 0.8. Клучните точки кои го задоволуваат овој услов се скалдираат како добри совпаѓања.

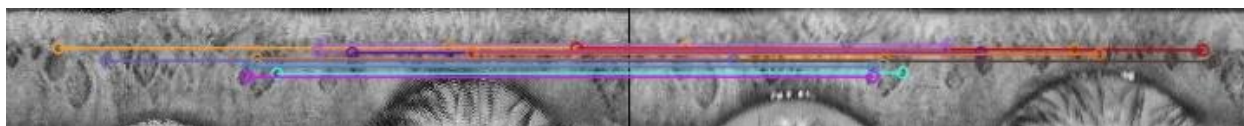
Генерираните дескриптори со секој од алгоритмите споменати погоре се споредуваат со Brute Force. Резултатите кои се добиваат со споредбата (врз две слики кои се однесуваат на ирис на ист човек) може да се видат визуелно на сликите подолу.



Слика 24. Поврзани дескриптори од FAST во комбинација со BRIEF на различни фотографии од ист човек



Слика 25. Поврзани дескриптори од ORB на различни фотографии од ист човек



Слика 26. Поврзани дескриптори од BRISK на различни фотографии од ист човек



Слика 27. Поврзани дескриптори од SIFT на различни фотографии од ист човек

## FLANN

FLANN или Брза библиотека за наоѓање на најблиски соседи содржи колекција алгоритми оптимизирани за брзо проаѓање на најблиски соседи во големи податочни множежства и за високо димензионални карактеристики. Работи многу побрзо од BFMatcher за големи групи на податоци. За да се овозможи работа на овој совпаѓач треба да се проследат два речника и тоа `index_params` и `search_params`.

Во комбинација со SIFT алгоритмот кој се користи во речникот `index_params` е `FLANN_INDEX_KDTREE` поставен на 0, а бројот на дрва `trees = 10`. Во речникот `search_params`

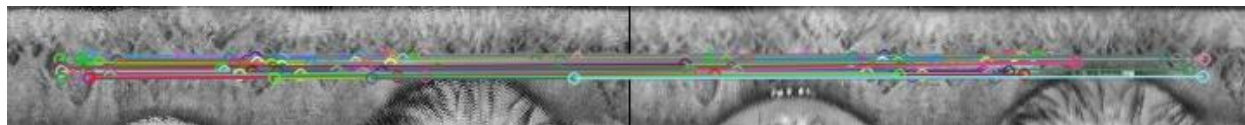


параметарот `cheek` се поставува на 100, што означува колку пати дрвата во индексот треба да се изминат рекурзивно. Исто и тука, се користи `knnMatcher` кој работи на истиот начин како и кај `BFMatcher`-от со `k=2` и истиот `Aspect Ratio` тест за одредување на најдобрите совпаѓања.

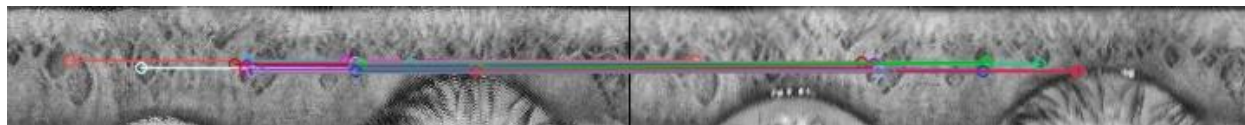
Во комбинација со ORB, BRISK и FAST се користи `FLANN_INDEX_LSH` алгоритам со вредност 6 и дополнително се поставуваат уште три параметри `table_number = 6`, `key_size = 12`, `multi_probe_level = 1`. `Search_params` речникот останува ист, а исти остануваат и наредните чекори за одредување на најдобри совпаѓања.

Како и претходно она што беше направено со кај Brute Force, се прави и тука. Brute Force методот се заменува со FLANN и се прави совпаѓање на дескрипторите генерирани со методите наведени во овој труд. Важно е да се напомене дека, FLANN работи одлично кога се комбинира со небинарни алгоритми за генерирање на дескриптори, како што е во нашиот случај SIFT. Но, кога се користи со бинарни алгоритми, може да се случи овој метод да не работи онака како што се очекува, бидејќи тој првично е наменет за совпаѓање на дескриптори генерирани со SIFT.

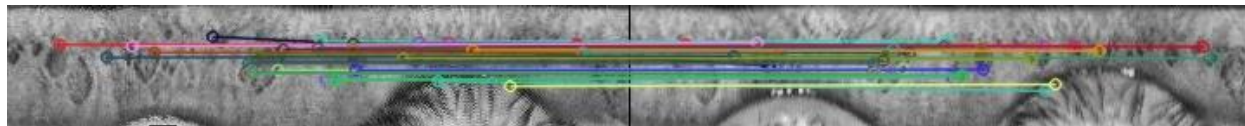
На следните слики е прикажан визуелен приказ на совпаѓањата направени врз дескриптори генерирани со секој од алгоритмите за дескриптори во комбинација на FLANN методот за совпаѓање.



Слика 28. Поврзани дескриптори од FAST во комбинација со BRIEF на различни фотографии од ист човек



Слика 29. Поврзани дескриптори од ORB на различни фотографии од ист човек



Слика 30. Поврзани дескриптори од BRISK на различни фотографии од ист човек



Слика 31. Поврзани дескриптори од SIFT на различни фотографии од ист човек

Како што може да се забележи на сликите погоре, најдобра визуелизација е онаа на SIFT во комбинација со FLANN. Со други зборови, со SIFT алгоритмот за генерирање на дескриптори се генерираат најголем број на клучни точки кои дополнително визуелно претставени и поврзани покажуваат добра конзистентност. Поради тоа, понатаму го користиме SIFT комбиниран со FLANN за да го одредиме прагот кој ќе одлучува дали две влезни слики претставуваат еден ист човек или не.

Прагот се одредува рачно. Се прават повеќе проверки над исти и различни луѓе и се добива вредност 20 за прагот. Тоа значи, ако во процесот на совпаѓање се добијат помалку од 20 добри совпаѓања двете слики кои се испратени на влез се однесуваат на различен човек, а доколку вредноста на добри совпаѓања е поголема од 20 се работи за истиот човек (истиот ирис, само аголот на проследените фотографии е различен).

## Резултати

На крај за да го тестираме она што го правиме во овој проект направиве 100 експерименти со користење на SIFT во комбинација со FLANN. Еден експеримент може да се опише како споредба на две фотографии било да се работи за фотографии на ист или различен човек. Во секој од експериментите ги меревме следните вредности: TP (точно позитивно предвидување – и двете слики се на ист човек), TN (точно негативно предвидување – двете слики се на различни луѓе) FP (лажно позитивно предвидување – и двете слики се на ист човек во случаи кога не се), FN (лажно негативно предвидување – двете слики се на различни луѓе во случаи кога се работи за ист човек). Резултатите може да се видат во Табела 1.

TP	62
TN	31
FP	7
FN	0
Total Experiments	100

Табела 1. Резултати добиени со SIFT во комбинација со FLANN

Според овие резултати може да кажеме дека нашиот пристап дава добри резултати. Она што е значајно да се напомене е тоа дека FN = 0, што значи дека не може да се случи двајца различни луѓе да бидат идентификувани како еден ист човек. Точноста која се добива во нашиот пристап измерена врз направените 100 експерименти е 0.93.

## Заклучок

Во оваа семинарска работа беше опишан процесот на градење на еден систем за споредба на ириси, првично со алгоритми за локализација, сегментација и нормализација на ирисот, а потоа и споредба на карактеристиките детектирани на добиените фотографии со помош на горенаведените алгоритми за генерирање дескриптори. Како најуспешен за понатамошна работа и за генерирање резултати, беше избран алгоритмот SIFT со кој се доби прецизност од околу 93%. Секако, постои простор за подобрување на овој систем. Вклучувањето на машинско учење во овој процес би можело да биде од клучно значење.

## Користена литература

- [1] John Daugman: “New Methods in Iris Recognition”, IEEE Transaction on Cybernetics, 2007
- [2] Shervin Minaee and Amirali Abdolrashidi: “DeepIris: Iris Recognition Using A Deep Learning Approach”, 2019
- [3] Kien Nguyen, Arun Ross: “Iris Recognition with Off-the-Shelf CNN Features: A Deep Learning Perspective”, Article in IEEE Access, 2017
- [4] SukTing Pui and Jacey-Lynn Minoi: “Keypoint Descriptors in SIFT and SURF for Face Feature Extractions” in Lecture Notes in Electrical Engineering, 2018
- [5] David G. Lowe: “Distinctive image features from scale-invariant keypoints” in International Journal of Computer Vision, 2004
- [6] Ethan Rublee, Vincent Rabaud, Kurt Konolige and Gary Bradski: “ORB: an efficient alternative to SIFT or SURF”, Conference Paper in Proceedings / IEEE International Conference on Computer Vision, 2011
- [7] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua: “BRIEF: Binary Robust Independent Elementary Features★”, Conference Paper, 2010