

1 ПРЕПОЗНАВАЊЕ ЛИЦА

Лице представља врло карактеристичан биометријски елемент човека. Лије је прва ствар по којој памтимо и потом препознајемо људе. Ослањајући се на то, креирани су и биометријски системи помоћу кога алгоритам може да препозна људе читавајући само њихова лица. Убрзо по почетку развоја ових система показало се да лице садржи јединствене карактеристике које се могу користити за препознавање особа са високим степеном прецизности.



Слика 1.1 – Пример слике лица.

Како би особа могла да се препозна помоћу лица, потребно је извући јединствене карактеристике лица које ће се користити за поређење у фази аутентификације. Техника која се користи да би се ово извело се базира на теорији информација и редукцији димензија. Та техника се назива “eigenfaces”, која се може сагледати као PCA (Principal Component Analysis) који садржи неколико иницијалних лица као тренинг податке.

1.1 EIGENFACE ПРИСТУП

Пре овог приступа се већина алгоритама заснивало на геометрији лица. Главне карактеристичне тачке су се тражиле и потом се лице препознавало рачунањем удаљености између сваке.

Eigenface приступ представља приступ са стране теорије информација. Изучавањем кључних црта лица и међусобних корелација дошло се до методе која може да издвоји битно од небитног. Тачније, може да представи лице са најбитнијим карактеристикама, док у исто време занемарује остале које нису од толике важности за сам процес аутентификације особе.

Са становништва теорије информација желимо да извучемо најбитније информације са лица, кодујемо их што је ефикасније могуће и потом поредимо то лице са осталим кодованим лицима која су претходно сачувана у бази података.

Из математичког угла, желимо да нађемо главне компоненте лица, тзв. „Eigenvectors” у матрици коваријанце коју смо добили из сета слика где је свака слика тумачена као засебан вектор у простору веома великих димензија.



Слика 1.2 – Пример eigen лица.

Eigenvector се може гледати као сет особина које заједнички представљају варијације између слика лица. Свака тачка на лицу доприноси крајњем изгледу eigenvector-a, те тиме eigenvector изгледа као сабласно лице које се назива “eigenface”. Ова метода је инспирисана радом L. Sirovich и M. Kirby, “Low-dimensional procedure for the characterization of human faces” који приказује технику за репрезентовање лица коришћењем РСА-а. У раду су расправљали о томе да се свако лице може представити у компресованом облику који редукује неважне делове лица. Уз то су успели да реконструишу оригиналну слику компресованог лица тако што су чували тежински вектор малог броја елемената који би се пројектовали на компресовано лице и тиме извукли оригиналну слику лица.

1.2 РАЧУНАЊЕ EIGENFACE-A

Како би се добила eigen лица пре свега је потребно да припремимо тренинг слике лица. Лица би требала да буду истих димензија како се не би правили проблеми током рачунања. Уколико су различитих димензија потребно их је претходно скалирати на исту величину. Пре почетка извлачења eigen лица је такође потребно пребацити слике лица у сиву скалу која се представља као матрица са вредностима 0 – 255.

Када су слике припремљене даље се може наставити са самим процесирањем кроз остале етапе. Циљ даљих калкулација јесте да се пронађе вектор који најбоље одговара дистрибуцији осталих слика лица на област простирања свих лица. Ове векторе дефинишемо као подскуп слика лица, које називамо „област лица“. У суштини, ови вектори представљају eigen векторе који су добијени из матрице корелације која одговара оригиналним сликама лица. Пошто ова матрица делује као лице када се прикаже у облику слике често се назива и “eigenfaces”.

Следећи кораци представљају кораке по којима се долази до eigen лица. Прво што се ради јесте рачунање просечног лица. Како би израчунали просечно лице потребно је да сва лица која користимо за тренинг учитамо у једну матрицу. Ту матрицу представљамо као $M \times N$ матрицу, где M представља вредности вектора лица које смо из матрице претворили у вектор, док N представља цео вектор лица. После учитавања свих лица у матрицу, средње лице се рачуна по следећој формули:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N x_i \quad (1.1)$$

пример је представљен сликом *Слика 1.3*. Следећи корак је нормализација лица. Лица се нормализују тако што се од сваког лица одузима средње лице које смо добили из (1.1). На тај начин добијамо матрицу која представља одступање сваког лица од просечног лица $\bar{X}$. Ову матрицу разлике добијамо на следећи начин:

$$D = X - \bar{X} \quad (1.2)$$

где X представља $M \times N$ матрицу са свим учитаним тренинг сликама. Пример једне нормализоване слике се може видети на *Слика 1.4*. Даље је потребно да израчунамо матрицу корелације. Она представља однос свих вектора лица и у односу на просечан вектор лица. Матрица корелације се рачуна:

$$C = D^T * D \quad (1.3)$$

Добијена матрица C представља матрицу $N \times N$ димензија. Када смо добили матрицу корелације рачунамо eigen векторе и вредности. Они се рачунају:

$$CV = \lambda V \quad (1.4)$$

где V представља скуп eigen вектора матрице C која је повезана са eigen вредностима λ . Како би при избору квалитетних eigen вектора изабрали најрепрезентативније, потребно је да их прво сортирамо по њиховим вредностима. Када их сортирамо при избору квалитетних вектора, биће довољно да узмемо само одређени број првих вектора који су сортирани од најбољих до оних лошијих. Следећи корак је добијање eigen лица. Она се добијају пресликавањем матрице разлике и eigen вектора:

$$E = D * V \quad (1.5)$$

Последњи корак јесте нормализација eigen вектора. Процес нормализације се врши дељењем вектора са векторском нормализацијом:

$$\|v\|_p = \left[\sum_{i=1}^n |v_k|^p \right]^{1/p} \quad (1.6)$$

где је у нашем случају $p = 2$, те се ова нормализација назива и Еуклидовска нормализација. Следећи корак нормализације је анулирање eigen вектора који имају мале eigen вредности како би се отклонили шумови. По завршетку ова два корака добијамо матрицу која представља нормализована eigen лица. Пример неколико eigen лице се може видети на *Слика 1.5*.

На овај начин смо добили eigen лица од којих можемо да бирамо број вектора које желимо да искористимо за даљи процес класификације који ће бити представљен у наставку рада.

1.3 ПРИМЕРИ ТОКОМ ПРОЦЕСА РАЧУНАЊА EIGEN ЛИЦА



Слика 1.3 – Изглед просечног лица свих тренинг лица.



Слика 1.4 – Нормализовано лице које се добија одузимање просечног лица од лица особе.



Слика 1.5 – Eigen лица након процеса нормализације.

1.3 КЛАСИФИКАЦИЈА ПОМОЋУ EIGENFACE-A

При добијању eigen лица следећи корак јесте класификација кандидата помоћу лица која нису прошла кроз систем у процесу тренирања, као и сви нови кандидати.

Пошто eigen лица представљају однос лица редукованих димензија која садрже најбитније карактеристике лица, идеална су за аутентификацију корисника помоћу њих. То значи да процес аутентификовања особе представља задатак препознавања образаца на слици.

Аутентификација особе се обавља тако што се рачунају тежински вектори за свако лице који ће се после користити за поређење са новим лицем. Тежински вектори се прво рачунају помоћу лица преко којих смо добили eigen лица јер ћемо те тежинске векторе користити за поређење са векторима новог лица. Ови вектори се могу израчунати док систем мирује, тј. кад нема особе која покушава да се аутентификује. Ови тежински вектори се рачунају по формули:

$$W_{train} = D^T * E \quad (1.7)$$

где D представља нормализована тренинг лица, тј. матрицу одступања сваког лица од просечног лица, док E представља сада већ нормализована eigen лица која садрже најквалитетније репрезентације лица.

У следећем кораку снима се лице особе и оно пролази кроз фазу нормализације (1.2) како би добили нормализован вектор D_{test} . Потом се рачуна тежински вектор новог лица (1.7) и добија се W_{test} . Након тога се пролази кроз сва лица и рачуна се разлика између свих лица из фазе тестирања и новог лица коришћењем Еуклидовске нормализације вектора (1.6) разлике тежинских вектора тренинг и новог лица:

$$W_{diff\ i} = \sqrt{\sum_{i=1}^n |W_{train} - W_{test}|^2} \quad (1.8)$$

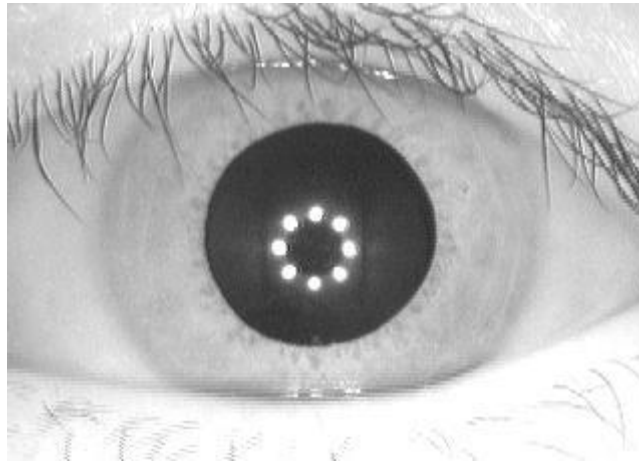
где $W_{diff\ i}$ представља поређење са једним тренинг лицем. Поређењем свих тренинг лица добија се W_{diff} тежински вектор где сваки елемент представља одступање од тренинг лица. Ово се може решити и применом Еуклидовске раздаљине која представља варијацију (1.6) између два вектора:

$$W_{diff\ i}d = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (3.9)$$

где се на крају такође добија W_d вектор који такође садржи разлике тежина између новог лица и свих осталих тренинг лица. На крају се особа проналази тако што се из ових вектора W_{diff} или у другом случају W_d тражи најмања раздаљина. Као сигурносни фактор може се гледати и одређени праг толеранције који ако се пређе лице не припада ни једном од особа у систему, и може се класификовати као уљез.

2 ПРЕПОЗНАВАЊЕ ИРИСА

Ирис представља кружну мембрану која лежи између рожњаче и сочива људског ока. Ирис, *Слика 2.1*, опкружује зеницу ока и његов центар кружнице је приближан центру зенице. Функција ириса јесте да контролише количину светла које пролази кро зеницу, тако што подешава величину зенице скупљањем и ширењем сфинктер и дилатрон мишића око зенице. Просечан пречник ириса је 12mm, док величина зенице може да варира од 10% до 80% величине пречника ириса.



Слика 2.1 – Једна од слика ока које је коришћено за тренинг. Површина између беоњаче и зенице се назива ирис.

Да би се особа успешно препознала помоћу ириса потребно је неколико корака за то. Пре свега је потребно пронаћи ирис на слици и одрадiti потпуну сегментацију ириса чиме се чист ирис, без шума издваја од остатка слике. Након тога следи нормализација ириса чиме се ирис трансформише у формат који је прикладан за даље процесирање и извлачење јединственог кода. На крају се ради кодовање извученог ириса помоћу филтера, како би се главне карактеристике ириса особе извукле и потом користиле за поређење при следећим аутентификацијама. Сваки од ових корака ће бити описан и објашњен у даљем раду.

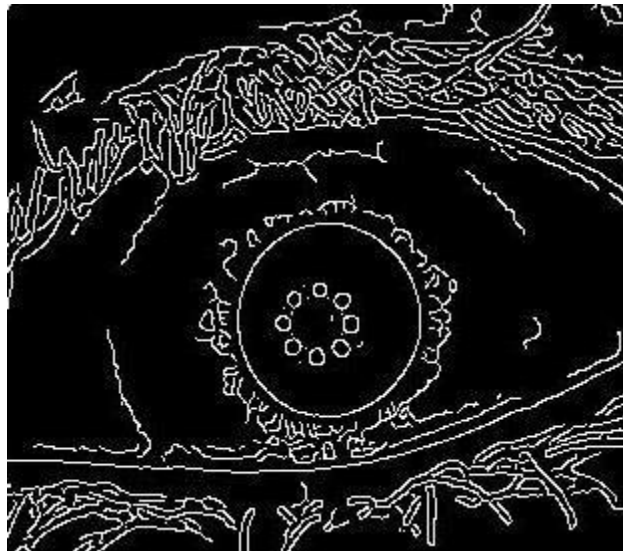
2.1 СЕГМЕНТАЦИЈА ИРИСА

Сегментација ириса представља процес изолације ириса из слике људског ока. Ирис може бити представљен као простор између две кружнице, једне око зенице и друге око спољне ивице самог ириса. Сметње које се често јављају у пределу ириса су очни капци и трепавице. Они се појављују са горње и доње стране ириса и требају бити уклоњени јер представљају шуме унутар самог ириса. Најбитнију ставку представља сам патерн ириса. То су шаре које су јединствене за сваког човека. Чак ни иста особа нема исте шаре на оба ока, те се у процесу аутентификације треба обратити пажњу и на те детаље.

2.1.1 Canny edge детектор

Први корак у процесу сегментације ириса представља извлачење ивица и контура са слике ока. Овај корак је потребан јер даљи процес извлачења ириса представља процес препознавања патерна на слици. Како бисмо препознали патерн нама су битне ивице објеката, а не и сваки детаљ истих. Како бисмо издвојили ивице на самој слици користићемо тзв. „Canny edge“ детектор.

Canny edge детектор нам омогућава да из слике извучемо најбитније контуре и да тиме драстично смањимо број података потребан за обраду. Ова техника се највише користи у системима рачунарског вида. Пример излаза из Canny edge-а се може видети на слици Слика 2.2.



Слика 2.2 – Излазна слика при примени Canny edge детектора. На слици остају само битне контуре (ивице) помоћу којих вршимо детекцију и сегментацију ириса.

За извлачење ивица помоћу canny edge детектора прво је потребно да отклонимо шум из слике како нам алгоритам не би пријавио лажне контуре и ивице. Шум са слике можемо отклонити применом Гаусових филтера који ће нормализовати и исполирати слику како не би постојали превише јаки прелази:

$$g(x, y) = \frac{1}{2\pi\sigma^2} * e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.1)$$

где x и y представљају хоризонтално и вертикално одстојање од центра респективно. Након примене Гаусовог филтра потребно је да пронађемо интензитет и смер градијената. Пошто ивице на слици могу бити у разним смеровима, Canny edge детектор

детектује вертикалне, хоризонталне и дијагоналне ивице како би успешно пронашао све ивице. Ивице можемо добити рачунањем:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.2)$$

$$\Theta = \text{atan2}(G_y, G_x) \quad (2.3)$$

где G_x и G_y представљају први извод у хоризонталном и вертикалном смеру респективно, где се угао рачуна применом (2.2) и (2.3). Како не би било претерано углова који морају да се памте, сви углови се на крају заокружују на 0° , 45° , 90° и 135° . Након одређивања смера добијамо замагљену слику на којој се ивице препознају, али су неправилне и широке.

Како бисмо извукли танке и чисте ивице, на слику примењујемо технику не-максималног сузбијања ивица где се све ивице које су и даље замагљене шире и сужавамо до јасног и танког облика.

На овако обрађеној слици се сада могу приметити јасније ивице, али се у слици налази доста шума које су проузроковали детаљи на слици који нису изоловани применом Гаусовог филтра. Како бисмо отклонили ове ивице примењују се два прага прихватљивости која представљају горњи и доњи праг пропустљивости. На овај начин отклањамо све ивице које се налазе ван ових опсега.

Последњи корак је финално отклањање шума. Њих отклањамо тако што пратимо повезаности ивица. Ако су неке ивице лошије повезане са осталима њих уклањамо, док оне које су добро повезане остављамо као најочигледније ивице. Начин на који се ово решава је тај да се за сваки пиксел ивице гледају њене комшије, те ако постоји јака ивица око пиксела онда се и тај пиксел убраја у јаку ивицу, у супротном се одбацује.

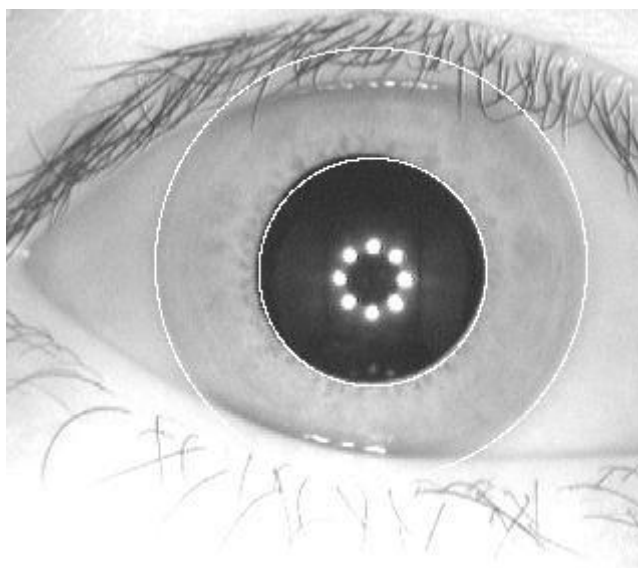
2.1.2 Hough трансформација

Hough трансформација представља један од стандардних алгоритама рачунарског вида, помоћу ког можемо да претражујемо једноставне геометријске патерне на слици, као што су линије и кругови. Ову трансформацију дефинишемо као:

$$[-(x - h_j) \sin \theta_j + (y - k_j) \cos \theta_j]^2 = a_j [(x - h_j) \cos \theta_j + (y - k_j) \sin \theta_j] \quad (2.4)$$

где a_j контролише закривљеност, (k_j, h_j) су врхови парабола и θ_j представља угао релативан на x-осу.

Када је ирис у питању, ова трансформација представља одлично решење за тражење ивица зенице, ириса и капака. Кружна трансформација се примењује како би се нашле спољне ивице зенице и ириса. Ова трансформација се примењује на слици где је примењен canny edge са вертикалним ивицама јер се показало да оне најбоље дефинишу спољне ивице кругова. Потом, како би се детектовали и уклонили капци примењује се линијска трансформација која детектује линије на слици. Линијска трансформација се користи на слици где су ивице извучене применом canny edge-а са хоризонталним ивицама. Пример извлачења кружница зенице и ириса се може видети на *Слика 2.3*.



Слика 4.3- Пронађене спољне ивице зенице и ириса. Површина између кружница представља ирис који се користи за аутентификацију особа.

Резултати коришћења Хаugh трансформације су координате центра и пречник кружнице који се представља као:

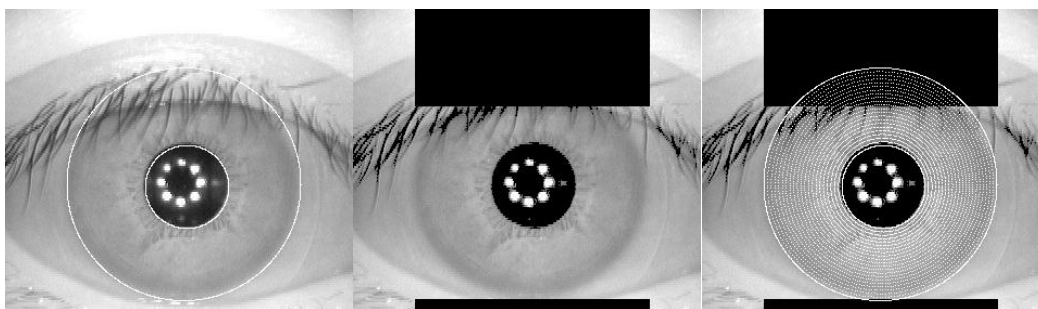
$$x_c^2 + y_c^2 - r_c^2 = 0 \quad (2.5)$$

и координате x_l и y_l које представљају линију очног капка. Очни капци се траже у две итерације, како би се нашла горња и доња ивица очног капка.

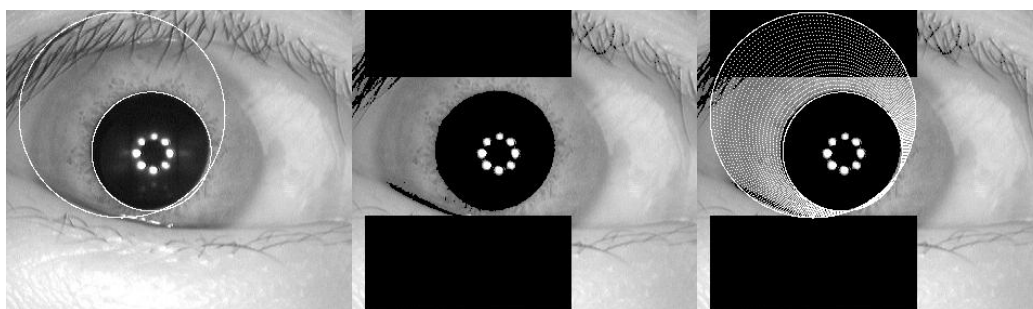
Проблеми овог алгоритма су што се ручно морају дефинисати прагови пропустљивости, што може довести до грешака и одбацивања битних ивица, или прихватања превише шума. Такође, ова трансформација је доста спора јер испитује пиксел по пиксел сваке ивице како би нашла најодговарајуће облике.

Ирис се сегментира тако што је прво детектован највећи круг на слици који је представљао спољну ивицу ириса. Потом је ирис извучен из слике и унутар ириса је нађен круг који је представљао спољну ивицу зенице. Овако смо добили површину ириса која се налази између ове две кружнице без сметње саме зенице. Како би се нашао горњи капак, извучен је горњи део ириса и нађена је линија која одговара капку. На исти начин је нађена и доња ивица капка. Последњи корак је отклањање шума. Шумови су отклоњени тако што је извучена слика ириса између две кружнице око зенице и спољне ивице ириса, и са те слике су отклоњени делови до горњег и од доњег капка. Уз то су отклоњене и трепавице једноставним отклањањем по прагу пропустљивости јер су трепавице по нијанси доста одступале од целокупне слике, те их је на тај начин било доста лако уклонити.

2.1.3 Резултати сегментације ириса



Слика 2.4- Пример успешне сегментације ириса. Прво се проналазе спољне ивице зенице и ириса. Потом се траже локације капака и врши се њихово анулирање како ти региони ириса не би сметали при даљем поређењу. На последњој слици се могу видети регије ириса које неће улазити у обзир приликом аутентификације.



Слика 2.5 – Пример неуспешног процеса сегментације ириса. Пошто је процес сегментације неуспешан особа ће бити класификована као уљез иако је валидна у процесу аутентификације.

2.2 НОРМАЛИЗАЦИЈА ИРИСА

Како бисмо успешно поредили ирисе особа потребно је да ириси буду конзистентних димензија. То значи да је потребно нормализовати све ирисе на исте димензије и у правилан формат како би поређење било могуће. Још један проблем који се јавља јесте тај да се зеница не налази нужно у центру ириса. Последице тога могу бити да се појаве нежељени шумови при лошој нормализацији. Све ове проблеме решава Даугманов „Rubber Sheet Model“.

2.2.1 Даугманов Rubber Sheet модел

Даугманов модел служи за нормализацију ириса из кружног облика у правоугаони облик. Ова метода функционише тако што ремапира сваку тачку ириса (x, y) на поларне координате (r, θ) где r представља пречник између ивица зенице и ириса $[0, 1]$, док θ представља угао пречника ириса $[0, 2\pi]$.

Мапирање ириса из Декартовог у поларни координатни систем се одвија по принципу:

$$I(x(r, \theta), y(r, \theta)) = I(r, \theta) \quad (2.6)$$

где су $x(r, \theta)$ и $y(r, \theta)$ једнаки:

$$\begin{aligned} x(r, \theta) &= (1 - r) x_p(\theta) + r x_i(\theta) \\ y(r, \theta) &= (1 - r) y_p(\theta) + r y_i(\theta) \end{aligned} \quad (2.7)$$

док $I(x, y)$ представља оригиналну слику ириса, а (x_p, y_p) и (x_i, y_i) представљају координате кружница, тј. спољних ивица, зенице и ириса респективно. Даугманов модел такође узима у обзир наведене проблеме са различитим величинама зенице, као и могућим померајима зенице у односу на центар ириса. По примени ове методе добија се нормализован ирис увек исте величине у облику правоугаоника. Илустровани пример се може видети на *Слика 2.6*.

Проблем помераја зенице се решава тако што се центар зенице и вектор помераја између центра зенице и центра ириса узимају као референтне тачке за добијање нормализованог ириса помоћу Даугмановог модела:

$$r' = \sqrt{\alpha\beta} \pm \sqrt{\alpha\beta^2 - \alpha - r_i^2} \quad (2.8)$$

где су α и β добијени:

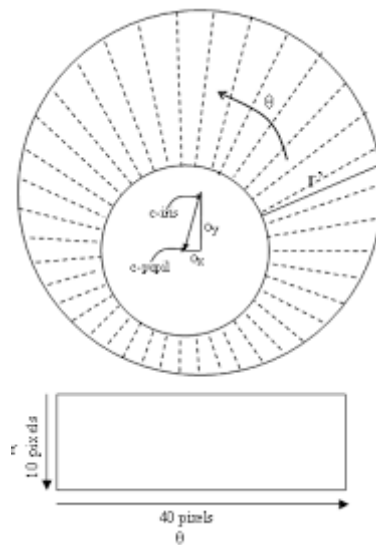
$$\alpha = o_x^2 + o_y^2$$

$$\beta = \cos(\pi - \arctan\left(\frac{o_y}{o_x}\right) - \theta) \quad (2.9)$$

где o_x и o_y дефинишу вектор помераја зенице у односу на ирис, r' представља размак између спољних ивица зенице и ириса у односу на угао θ и r_i представља пречник ириса.

Уз креирање нормализованог ириса креира се и маска нормализације. Она служи за искључивање шума ириса током фазе поређења. Та маска садржи локације капака ириса који су преклопили делове ириса, трепавице и остале елементе за које је утврђено да представљају шуме. Пример нормализованог ириса се може видети на *Слика 2.7*.

2.2.2 Резултати нормализације ириса



Слика 2.6 – Визуални приказ процеса нормализације ириса применом Даугманове методе *rubber sheet-a*.



Слика 2.7 – Пример нормализованог ириса.

2.3 КОДОВАЊЕ ИРИСА

Како би поређење ириса било успешно, потребно је да патерни ириса истих особа буду доста слични, и поред тога што се делимично деформишу са променом светлости уз последицу ширења или скупљања зенице. Решење овог проблеме је да се из ириса извуку само кључне тачке које представљају јединственост ириса, док би се остале анулирале. На овај начин, иако дође до ситнијих деформација у самом ирису укупна разлика између истих ириса ће опет бити доста мања него у случају различитих, што омогућава успешно аутентификовање особе.

Извлачење главних карактеристика ириса кодовањем врши се применом Габорових филтера. За проблем кодовања ириса ћемо користити 1D логаритамски Габоров филтер. Пример кодованог ириса се може видети на *Слика 2.8*.

2.3.1 1D логаритамски Габоров филтер

Габорови филтери обједињују приказивање сигнала у просторном домену и домену просторне фреквенције. Они раде тако што модулирају синусне и косинусне таласе применом Гаусове методе:

$$f(x) = a e^{-\frac{(x-b)^2}{2c^2}} \quad (2.10)$$

где a , b и c представљају производне реалне константе. Даљом модулацијом Гаусова метода успешно локализује сигнал у простору, али уз то губи локализацију у простору фреквенција. Након локализације, декомпозиција сигнала на реални и имагинарни део обавља се коришћењем пара Габорових филтера, где је реални специфициран косинусном модулацијом, а имагинарни део синусном. Комбинација ова два филтера, реалног и имагинарног се такође назива и симетрични филтри.

Иако су доста корисни, Габорови филтери имају своје недостатке. Максимални пропуст филтера је лимитиран на једну октаву сигнала, и поред тога, Габорови филтери нису оптимални у случају тражења ширих спектралних информација сигнала са максималном просторном локализацијом. Решење дела проблема који су нама потребни јесте коришћење Габорових филтера који користе логаритамски скалирану Гаусову методу која се назива логаритамски Габоров филтер:

$$G(f) = \exp\left(\frac{-\left(\log\left(f/f_0\right)\right)^2}{2\left(\log\left(\sigma/f_0\right)\right)^2}\right) \quad (2.11)$$

Помоћу логаритамског Габоровог филтера можемо вршити кодовање нормализованог ириса. Пошто је нормализован ирис конструисан у дводимензионалном облику применићемо технику једнодимензионалних логаритамских Габор филтера разбијањем нормализованог ириса у више једнодимензионалних сигнала. Потом сваки појединачни сигнал кодујемо коришћењем Даутманове методе кодовања помоћу 2D Габор филтера:

$$h_{\{Re,Im\}} = sgn_{\{Re,Im\}} \int_{\rho} \int_{\phi} I(\rho, \phi) e^{-i\omega(\theta_0 - \phi)} e^{\frac{-(r_0 - \rho)^2}{\alpha^2}} e^{\frac{-(\theta_0 - \phi)^2}{\beta^2}} \rho d\rho d\phi \quad (2.12)$$

где $h_{\{Re,Im\}}$ представља комплексан бит чије су реалне и имагинарне вредности 0 или 1, sgn је знаг који зависи од знака 2D интеграла; $I(\rho, \phi)$ је слика ириса представљена у поларним координатама; α и β представљају величине 2D таласа; ω представља таласну фреквенцију; (r_0, θ_0) репрезентују поларне координате сваког региона ириса за који су фазне координате $h_{\{Re,Im\}}$ израчунате. Овај филтер ће произвести два бита за сваки фазор.

Процес кодовања ће на крају произвести битски шаблон који ће садржати битне карактеристике ириса кодоване у низ бита. Поред тога ће се добити и маска шума која одговара регионима ириса који су под шумом. Након тога ћемо кодован шаблон ириса, заједно са маском шума искористити за поређење ради аутентификације особе.

2.3.2 Резултати кодовања ириса



Слика 2.8 – Пример кодованог ириса. Овај образац се заједно са маском шума користи за аутентификацију особе.



Слика 2.9 – Маска шума ириса. Ова маска показује регије ириса које нису употребљиве за сам процес аутентификације особе.

2.4 КЛАСИФИКАЦИЈА ПОМОЋУ ИРИСА

После комплетне обраде ириса и коначног добијања кодованог шаблона, спремни смо да користимо тај ирис за аутентификацију осталих кандидата.

Како бисмо аутентификовали кандидате потребно је да користимо одређену метрику којом бисмо утврђивали разлике између појединих ириса. У случају ириса је одабрано Хамингово растојање.

Опште Хамингово растојање представља удаљеност два елемента која се пореде. То могу бити удаљености два броја, стринга, низа битова и сл. У нашем случају то ће бити удаљености два кодована шаблона ириса. Општа формула Хаминга се дефинише као:

$$H(x, y) = \sum_{i=1}^n x_i \oplus y_i \quad (2.13)$$

Резултат Хаминговог растојања ће бити сума различитих елемената у низу елемената над којим се траже дистанце. Обичан Хаминг у нашем случају неће радити најбоље, јер су присутне маске шума које нам назначују који делови ириса нису употребљиви. Из тог разлога, у овом раду се користи модификована верзија Хаминговог растојања:

$$H = \frac{1}{N - \sum_{k=1}^N Xn_k \vee Yn_k} \sum_{j=1}^N X_j \oplus Y_j \wedge Xn_j \wedge Yn_j \quad (2.14)$$

где X_j и Y_j представљају два кодована шаблона ириса, Xn и Yn представљају маске шума које одговарају X_j и Y_j , а N представља суму битова која одговара сваком шаблону ириса.

Иако би у теорији два ириса од исте особе требало да буду исти, у пракси није тако. Због несавршености свих слика и несавршености самог процеса нормализација, између сваког ириса ће се јавити одређена раздаљина.

Пошто ириси умеју често да варирају на сликама, поготово због саме позиције ако се глава за нијансу заротира у страну и тиме се заротира и ирис, може доћи до великих растојања чак и код истих ириса. Како би се решио овај проблем примењује се Даугманов трик са померајем шаблона ириса у једну и другу страну за одређени број битова. Ови помераји ириса представљају ротирање ириса када није нормализован, те се тиме отклања проблем ако је ирис за нијансу ротиран током сликања. У току поређења нови ирис се ротира пар пута и пореди са старим, рачуна се растојање између њих, и на крају се узима онај резултат где је растојање најмање, тачније, где нови ирис најближе одговара старом. На самом крају се гледа да ли раздаљина за конкретног корисника не прелази праг прихватљивости. Ако је то случај, корисник се аутентификује као валидан и пролази проверу система.