



Факултет за Информатички Науки и Компјутерско Инженерство

## ИЗВЕШТАЈ ОД ИЗРАБОТЕН ПРОЕКТ

Предмет: Дигитално Процесирање на Слика

### ТЕМА:

Детекција на лица и замена на лице



Ментор:

Вон. Проф. Д-р Ивица Димитровски

Изработиле:

Андреј Скендерски 181117

Иван Марковски 185051

## Contents

Абстракт .....	3
Метода .....	3
Модел на Замена на Лица .....	3
Детектирање на лице.....	4
Деланееви Триаголници (Delaunay Triangulation) .....	5
Трансформирање и искривување на триаголници .....	6
Детали за алгоритмот .....	8
Усогласување лице.....	8
Face warping (искривување на лицето) .....	10
Примери од демо-апликацијата .....	12
Користена литература.....	14

## Абстракт

Целта на овој проект е да претставиме алгоритам за детекција на лице и замена на лица. Проблемите со замена на лица е различни форми, големини, бои и ротација на лицата. Потребно е да се најде алгоритам кој ќе може правилно да го детектира лицето на една слика и соодветно да го замени со лице на втора слика. Притоа финалниот продукт да изгледа реално. Ова е овозможено преку библиотеката за машинско учење во Python, OpenCV.

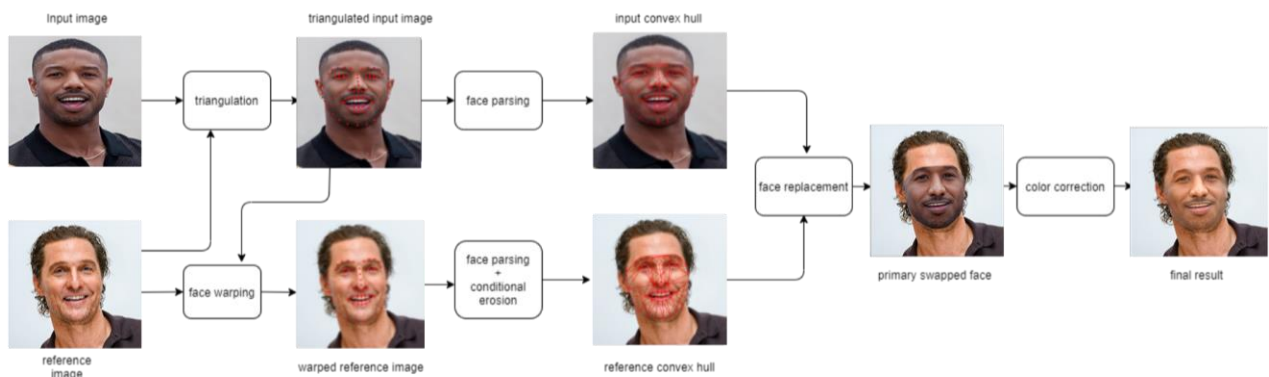
## Метода

Алгоритмот е составен од 5 чекори:

1. Детекција на лицето
2. Правење маска на лицето
3. Користење на Деленееви триаголници
4. Искривување (warp) на триаголниците
5. Замена на триаголниците

## Модел на Замена на Лица

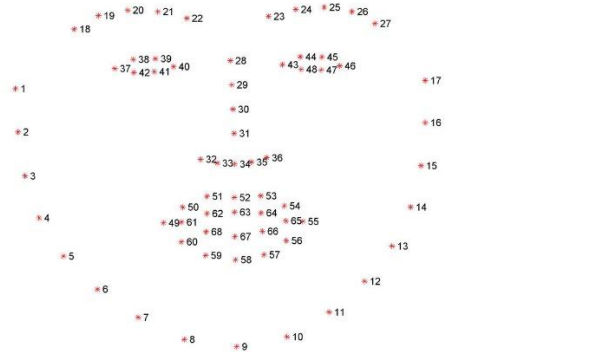
Првично изгледа лесно и предвидливо замена на лице, но алгоритмот има огромно влијание врз финалниот продукт. Моделот на алгоритмот претставува два канали кои на крајот настануваат еден, прикажано на Слика 1. Првично се прави маска од сликата од алгоритам за детекција на точки на лицето (landmarks). Потоа маската се дели на повеќе триаголници, користејќи Деленеев (Delaunay) алгоритам. Секој поединечен триаголник од првата слика се трансформира во соодветниот триаголник од втората слика. На крајот сите трансформирани триаголници од првата слика се поврзуваат во една целина и таа се вметнува во втората слика.



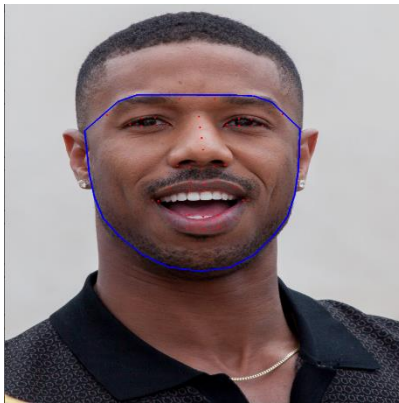
Слика 1 Модел на алгоритмот за замена на лица

## Детектирање на лице

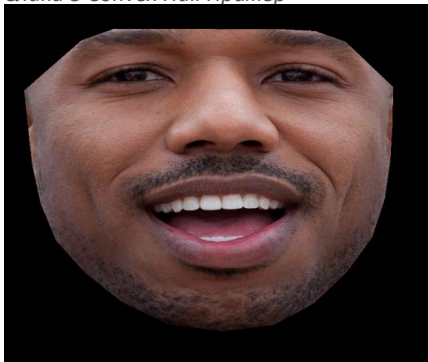
Детектирање на лице се прави со вградените команди на dlib библиотеката. Првично за тоа е потребно оригиналните слики да бидат во сива боја, за ова ја користиме библиотеката OpenCV. Dlib користи детектор за формата на лицето, кое се добива од шема од 68 точки, прикажано на Слика2.



Слика 2 68 Точки на лицето за детекција на форма



Слика 3 Convex Hull Пример



Слика 4 Маска добиена со Convex Hull

Откако детектор ќе ги лоцира 68-те точки на сликата, се генерираат соодветните координати на точките. Со помош на шемата од 68 точки се прави Convex Hull, односно се поврзуваат надворешните точки од шемата за да се добие форма околу лицето.

Convex Hull всушност претставува група од надворешните отсечки на лицето каде ниту еден агол не е поголем од  $180^\circ$ , прикажано на Слика3. Заедно со Convex Hull и со точните координати може да се направи маска од лицето, како на Слика4.

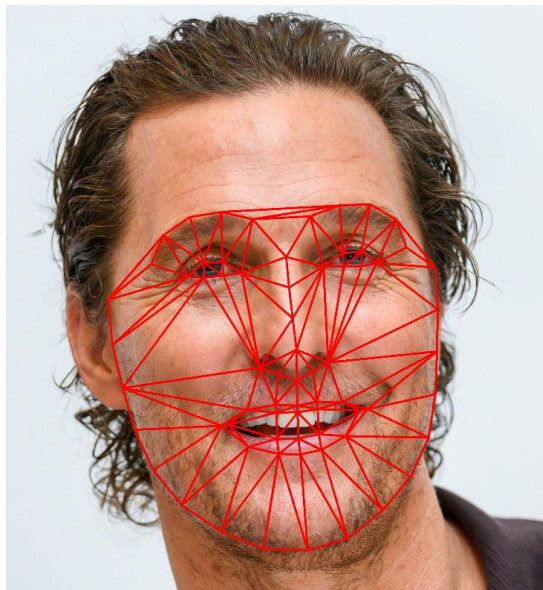
## Деланееви Триаголници (Delaunay Triangulation)

Откако ќе се добие маската на лицето, следи делење на повеќе триаголници. Првично 68-те точки се делат на групи од 3 точки и се формираат триаголници. Најважното кај

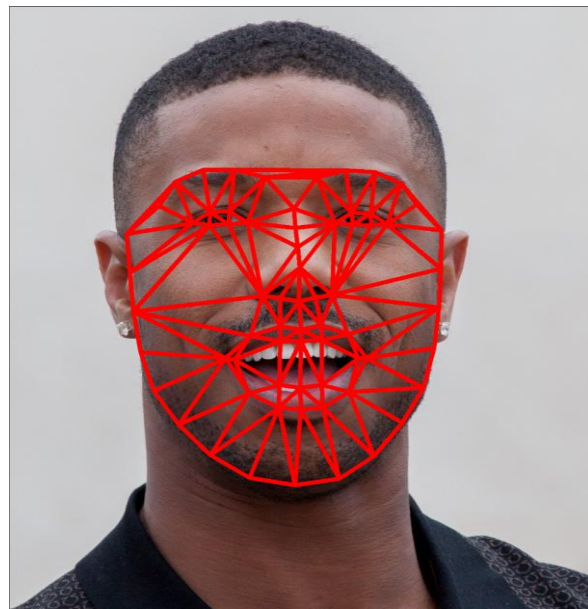
Деланееви триаголници е што се зимаат најголемите триаголници, додека најмалите се изоставуваат бидејќи не влијаат врз крајниот резултат.

Лицето е поделено на триаголници, бидејќи е комплицирано да се земе целото лице од првата слика и да се вметне врз втората слика. Компликацијата е во тоа што може да се доведе до губење на оригиналните пропорции. Со поделба на триаголници, па нивната соодветна замена се задржуваат пропорциите и изразите на лицето.

Триаголниците на втората слика мораат да бидат со иста шема како првата слика. За ова да биде овозможено секоја од координатите на првата слика е со соодветен индекс од 68-те точки на лицето. Со помош на овој индекс на втората слика може да се креираат триаголници со иста шема како и првата слика.



Слика 5 Пример за Деланееви Триаголници



Слика 6 Пример 2 за Деланееви Триаголници

## Трансформирање и искривување на триаголници

Откако ќе ја добиеме триангулацијата на двете лица, ги земаме триаголниците на изворното лице и ги трансформираме. Исто така, треба да ги земеме координатите на триаголниците на резултантното лице, за да можеме да ги искривиме триаголниците на изворниот лик за да имаат иста големина и форма на резултатниот триаголник за да има совпаѓање на лицата. Откако ќе ги исечеме и искривиме сите триаголници, треба да ги поврземе. Ние едноставно го обновуваме лицето користејќи ја шемата за триангулација, со единствена разлика што овој пат го ставивме искривениот триаголник. На крајот на оваа операција, лицето е подготвено за замена.



Слика 7 Лицето од првата слика



Слика 8 Избришано лицето на втората слика

Лицето сега е подготвено за замена, види Слика 7. Го отсекуваме лицето на резултантната слика за да направиме простор за новото лице, види Слика 8. Така, го земаме новото лице, и сликата на дестинацијата без лице и ги поврзуваме заедно.



Конечно, лицата се правилно заменети и време е да се прилагодат боите така што изворната слика ќе одговара на дестинациската слика. На OpenCV имаме вградена функција наречена „seamlessClone“ што ја извршува оваа операција.



Слика 10 Целосната слика пред 'seamlessClone'



Слика 9 Крајна слика

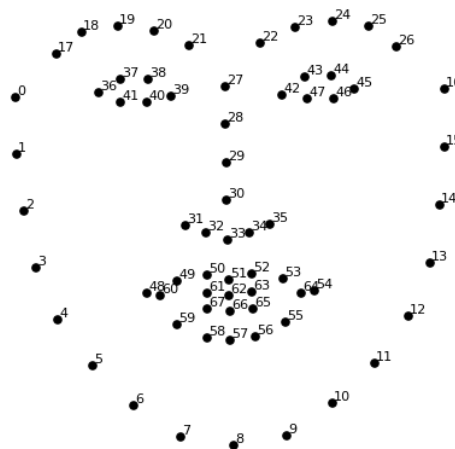
## Детали за алгоритмот

### Усогласување лице

Како прв чекор на замена на лицето, усогласување се однесува на усогласување на влезната слика на лицето  $I_{in}$  и референтната слика на лицето  $I_{ref}$  во големина и насока. За цел на откривање лица во слики, ги применуваме релевантните методи во хартија која предлага роман повеќекратно рамка за застапување за визуелно следење за откривање на лица на слики. Освен зголемување на брзината на алгоритам, примена на ретко кодирање и речник учењето исто така им овозможува на овие методи да научат повеќе знаење од релативно помалку примерочни податоци. Десет извлекуваме неколку стабилни клучни точки од сликите за означување на лицата, наведени да се открие како обележје на лицето (накратко FLD). Во овој документ, ние користиме популарен FLD алгоритам базиран на ансамбл на дрвја за регресија за откривање обележја на лицето  $\Omega_s = \{S_i \mid i = 1, 2, \dots, 68\}$ , како што е прикажано на слика 11. Ние го користиме ова метод кој се потпира на нашата имплементација. Секоја обележје точката  $s_i$  е симетрична со друга точка  $s_i'$  во однос на централната оска на лицето како што се  $s_{22}$  и  $s_{23}$ ,  $s_{49}$  и  $s_{55}$ . Те точките лоцирани на централната оска се симетрични сами по себе како што се  $s_{28}$  и  $s_{31}$ . Да се оцени ротацијата помеѓу влезот и референтниот лик, централната оска на влезната страна треба да се извлече претходно. Според основната дефиниција, централната оска може да се оцени врз основа на постапка за оптимизација како што е

$$l_s = \min_l \sum_{i \in \Omega_s} \|s_i^{(l)} - s_i'\|^2,$$

Равенка 1



Слика 11 68 Точки за Лице



каде  $s_i^{(1)}$  е пресликување на  $s_i$  во однос на линијата  $l$ . Равенката (1) покажува дека централната оска  $l_s$  треба да биде линијата што ја оптимизира симетријата на сите 68 откриени точки на лицето. Следствено, центарот  $c_s$  на лицето е дефиниран како средната точка на проекциите на сите 68 откриени точки на лицето од централната оска  $l_s$ . Просечно растојание од сите откриените знаменитости до централната точка  $c_s$  се означува со  $d_s$ , што може да се користи како метрика на големината на влезната површина. Затоа, просечното растојание  $d_s$  може да се запише како

$$d_s = \frac{1}{|\Omega_s|} \sum_{i \in \Omega_s} \|s_i - c_s\| ,$$

Равенка 2

каде  $|\Omega_s| = 68$ . Слично на тоа, централната оска и големината на референтното лице, соодветно, се означува со  $l_r$  и  $d_r$ . Тогаш усогласувањето на лицето може да се спроведе со ротирање на влезот слика  $I_{in}$  „со агол“  $\Theta$  и скалирање со фактор  $k$ , каде што  $\Theta$  е аголот на ротација од  $l_s$  кон  $l_r$ , а  $k$  е односот помеѓу  $d_r$  и  $d_s$ , како што е прикажано на

$$\theta = \langle l_s, l_r \rangle ,$$

$$k = \frac{d_r}{d_s} .$$

Равенка 3

По усогласување на лицето, оригиналните точки на лицето  $s_i$  се пренесуваат на новите локации  $S_i^a$ , и усогласената влезна слика се означува со  $I_{in}^a$ .

## Face warping (искривување на лицето)

За да ја зачуваме формата на заменетото лице, ја искривуваме референтната слика до fit израмнетиот влезен лик пред замена на лицето. Искривувањето се спроведува врз основа на усогласување на обележјата на лицето. Избираме 18 од 68 обележја на лицето и означете ги со  $\Phi_r = \{1, 2, \dots, 17, 34\}$  (види слика 3 (a)), за кои се смета дека имаат значајно значење влијание врз обликот на лицето. Знаменитости на референтниот лик се означуваат со  $r_i$ ,  $i \in \Phi_r$ . Новите локации  $r_i^w$  од повеќето од точките (освен  $r_1$ ,  $r_{17}$  и  $r_{34}$ ) по искривувањето на сликата треба да бидат совршено усогласени со input сликата, така што имаме:

$$r_i^w = \begin{cases} r_i & \text{if } i = 1, 17, 34 \\ c_r + (s_i^a - c_s) & \text{otherwise.} \end{cases}$$

Равенка 4

За да се реализира искривување на сликата, референтната слика  $I_{ref}$  првично се разложува на многу парчиња триаголник врз основа на обележјата. Триангулацијата е потребно за да се минимизира промената на позадината на сликата затоа што генерално се надеваме дека ќе зачуваме некои делови во позадина како што се косата, телото и облекувањето (затоа го правиме тоа не се движат  $r_1$ ,  $r_{17}$  и  $r_{34}$ ). Десет сликата се искривува може да се реализира со примена на специфичната трансформација на соодветните парчиња триаголник.

Да претпоставиме дека постои триаголник чии три темиња се  $r_i$ ,  $r_j$  и  $r_k$ , а соодветните локации потоа од искривувањето се означуваат со  $r_i^w$ ,  $r_j^w$  и  $r_k^w$ , соодветно. Тогаш трансформацијата на триаголникот може да се опише како

$$R^w = M\hat{R},$$

Равенка 5

каде:

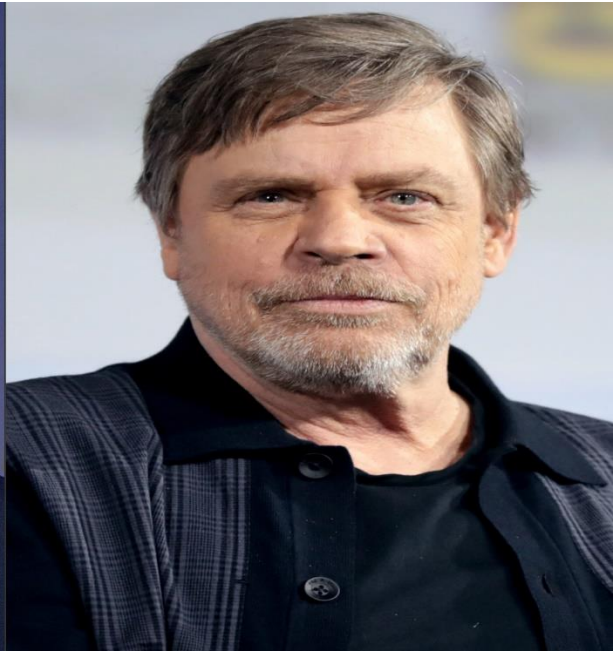
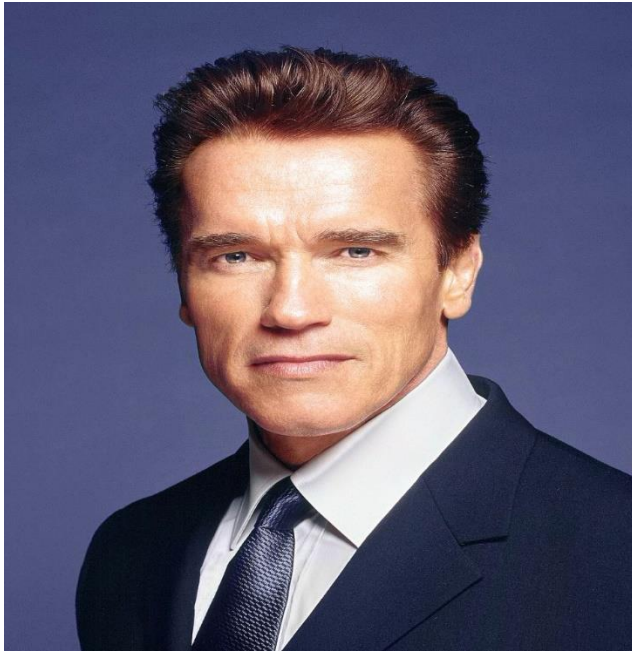
$$R^w = [r_i^w, r_j^w, r_k^w] = \begin{bmatrix} x_i^w & x_j^w & x_k^w \\ y_i^w & y_j^w & y_k^w \end{bmatrix},$$

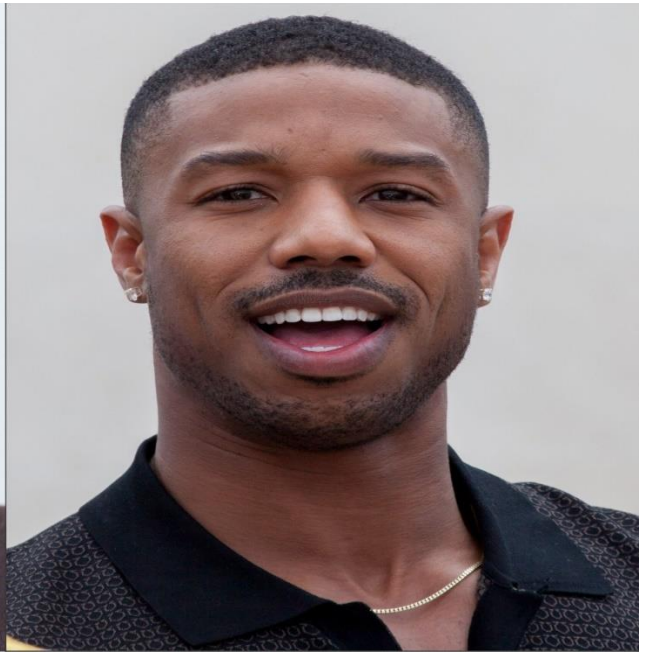
$$\hat{R} = \begin{bmatrix} r_i & r_j & r_k \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ 1 & 1 & 1 \end{bmatrix}.$$

Равенка 6

$M \in \mathbb{R}^{2 \times 3}$  е матрица за трансформација чие решение во затворена форма е  $R' R^{-1}$ . Со матрицата  $M$ , секој пиксел во внатрешноста на триаголникот може да се трансформира од првобитната локација  $r$  на нова локација  $r^w$ . Бидејќи координатите на новите локации се воопшто не е цел број, билинеарната интерполација се користи за генерирање на конечната искривена референтна слика  $I_{\text{ref}}^w$ .

## Примери од демо-апликацијата





## Користена литература

<https://www.hindawi.com/journals/mpe/2019/8902701/>

<https://pysource.com/2019/05/28/face-swapping-explained-in-8-steps-opencv-with-python/>

[https://www.youtube.com/watch?v=dK-KxuPi768&ab\\_channel=Pysource](https://www.youtube.com/watch?v=dK-KxuPi768&ab_channel=Pysource)

[https://github.com/italojs/facial-landmarks-recognition/blob/master/shape\\_predictor\\_68\\_face\\_landmarks.dat](https://github.com/italojs/facial-landmarks-recognition/blob/master/shape_predictor_68_face_landmarks.dat)