

BABEȘ BOLYAI UNIVERSITY, CLUJ NAPOCA, ROMÂNIA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Reconstruction 3D

– MIRPR report –

Team members

Salsigan Razvan-Dan, *rsalsigan@gmail.com*

Pop Leonard-Andrei, *leonard.pop@stud.ubbcluj.ro*

Paslariu Alexandru Ilie, *alex_pislariu2000@yahoo.com*

Specializarea informatica romana, grupa 236

2021-2022

Abstract

Pentru a proteja pacientii supusi unor anumite terapii radiologice, este utila purtarea unor masti de protectie. Cu cat aceste masti respecta mai mult fizionomia pacientului, cu atat gradul de securitate al organelor care nu trebuie iradiate este mai mare.

Cuprins

1	Introducere	1
2	Problema stiintifica abordata	2
3	Metode existente de rezolvare a problemei	3
3.1	Mediapipe Face Mesh	3
3.2	Pixel2Mesh	3
4	Metode efectiv folosite pentru rezolvarea problemei	4
4.1	Main flow	4
4.2	Detalii de implementare	5
5	Rezultate experimentale obtinute	8
5.1	Incerari cu Pixel2Mesh++	8
6	Concluzii si posibile imbunatatiri	12
6.1	Posibile imbunatatiri	12
7	Latex examples	13

List of Tables

7.1	The parameters of the PSO algorithm (the micro level algorithm) used to compute the fitness of a GA chromosome.	14
-----	---	----

Pomelnic grafic

4.1	Mindmap	5
4.2	Rezultate din unghiuri diferite	6
4.3	Rezultatul interpolarii	7
5.1	Rezultat doua imagini cu un scaun	8
5.2	Rezultat John Cena cu background	9
5.3	Rezultat John Cena fara background	9
5.4	Rezultat trei unghiuri cu background	10
5.5	Rezultat frontal fara background	10
5.6	Rezultat bust alb inchis	11
7.1	The evolution of the swarm size during the GA generations. This results were obtained for the f_2 test function with 5 dimensions.	13

List of Algorithms

1	SGA - Spin based Genetic AAlgorithm	14
---	---	----

Capitol 1

Introducere

În terapia radiologică a diverselor boli este nevoie de construirea unor masti de protecție a anumitor părți ale corpului (de ex. protejarea fetei). Dacă aceste masti respectă cât mai mult fizionomia unui pacient, ele se transformă în aliați siguri ai pacientului, dar și ai medicului în combaterea bolilor și creșterea gradului de securitate a organelor care nu trebuie iradiate.

Construcția unei masti se poate adapta specificului unui pacient, respectând trasaturile fetei. În acest context se dorește dezvoltarea unei aplicații care pe baza imaginii fetei unei persoane să poată extrage caracteristicile faciale și să frunizeze un model 3D a acestora. Acest model va putea fi folosit mai departe pentru a printa masca la o imprimantă 3D.

Scanarea 3D a obiectelor implică în mod obișnuit echipament specializat și o putere mare de calcul. Pentru a putea face acest proces mai accesibil, este necesară folosirea unor algoritmi inteligenți care pot extrapola modele 3D din doar câteva imagini.

Capitol 2

Problema stiintifica abordata

Algoritmul va primi ca date de intrare mai multe imagini cu fata unei persoane, din mai multe unghiuri. Aceste imagini vor contribui impreuna la generarea unei aproximari 3D cat mai apropiate a fetei. Dupa acest pas, modelul va fi prelucrat astfel incat sa ia forma unei masti de protectie personalizate persoanei din imagini. Modelul final va fi convertit intr-un format usor de folosit pentru a genera un fisier de instructiuni intelese de o imprimanta 3D. Masca finala va fi rezultatul imprimarii 3D.

Capitol 3

Metode existente de rezolvare a problemei

3.1 Mediapipe Face Mesh

Mediapipe Face Mesh [\[1\]](#)

3.2 Pixel2Mesh

Pixel2Mesh [\[3\]](#)

Capitol 4

Metode efectiv folosite pentru rezolvarea problemei

4.1 Main flow

Proiectul consta intr-o aplicatie Android, in care utilizatorul va putea incarca poze, fie facandu-le direct cu camera telefonului, fie din memorie interna sau din alte forme de stocare.

Dupa ce pozele sunt incarcate, acestea vor fi procesate in urmasorii pasi:

- Extragere caracteristici (folosind MediaPipe [1] pentru fata, TBD pentru partea de gat si partile laterale ale capului)
- Generare poligoane pe baza caracteristicilor extrase (care vor fi sub forma de puncte intr-un spatiu 3D)
- Generare mesh bazat pe modelul obtinut

Rezultatul va fi un fisier in format *.stl*, care va putea fi vizualizat local (folosind android AR Scene Viewer, daca dispozitivul permite), dar si exportat.

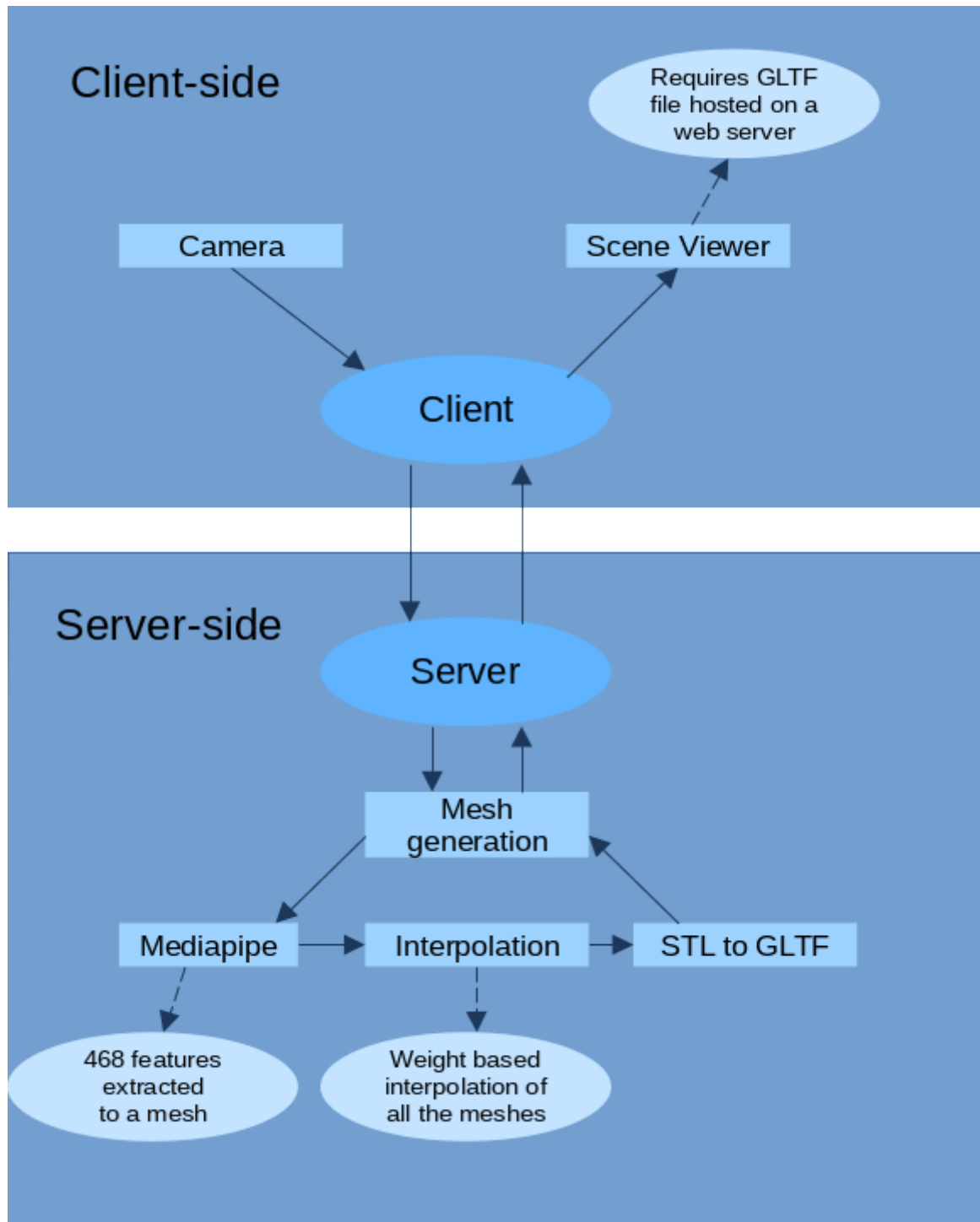


Figure 4.1: Mindmap

4.2 Detalii de implementare

Folosim MediaPipe [1] pentru extragerea celor 468 de landmarks din fiecare imagine. Ca output avem coordonatele fiecarui punct, iar pentru generarea meshului le conectam folosind pyvista [2], aplicand

filtrul delaunay pe point cloud.

Din fiecare imagine se poate genera cate o aproximare a structurii 3D a fetei, iar calitatea modelului va fi dependenta de unghiul din care imaginea a fost facuta (partile vizibile vor putea fi approximate mai bine, iar cele obstructionate vor fi mai greu de generat).

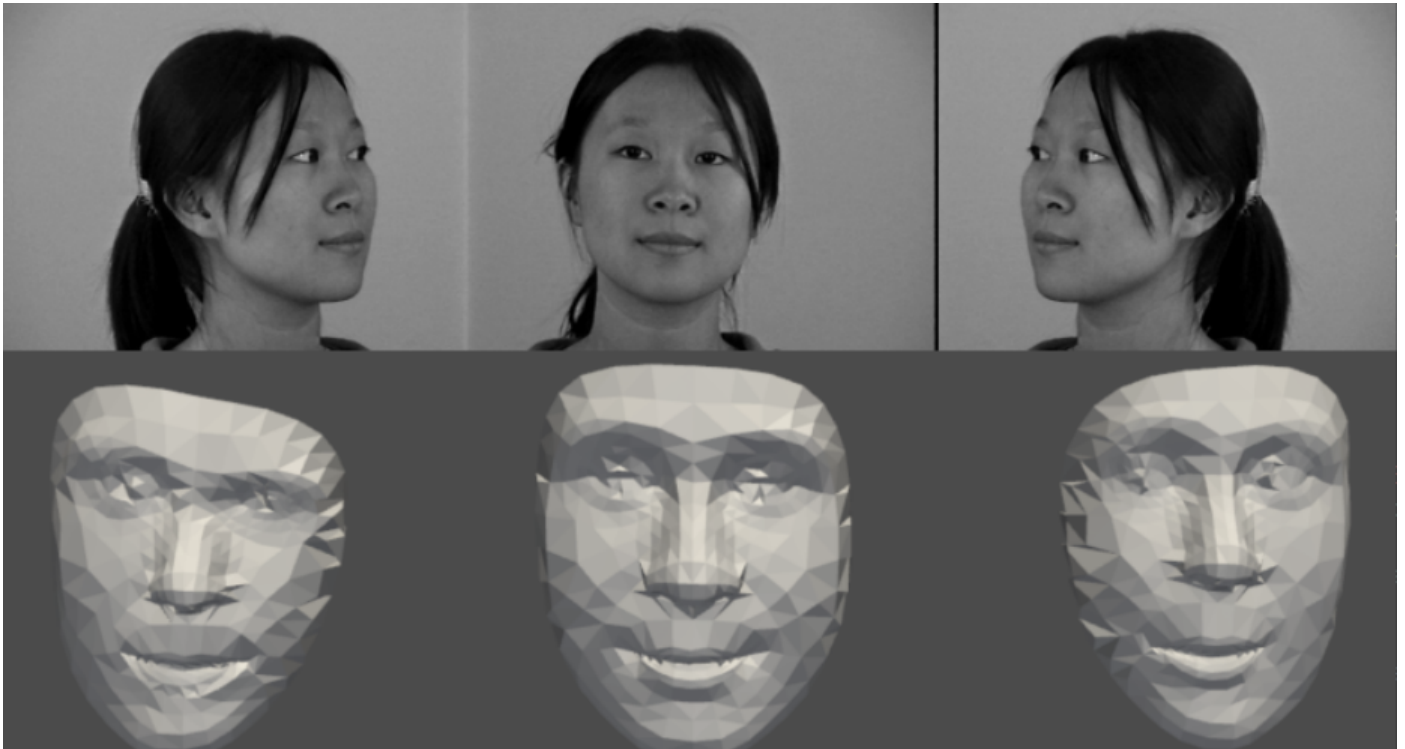
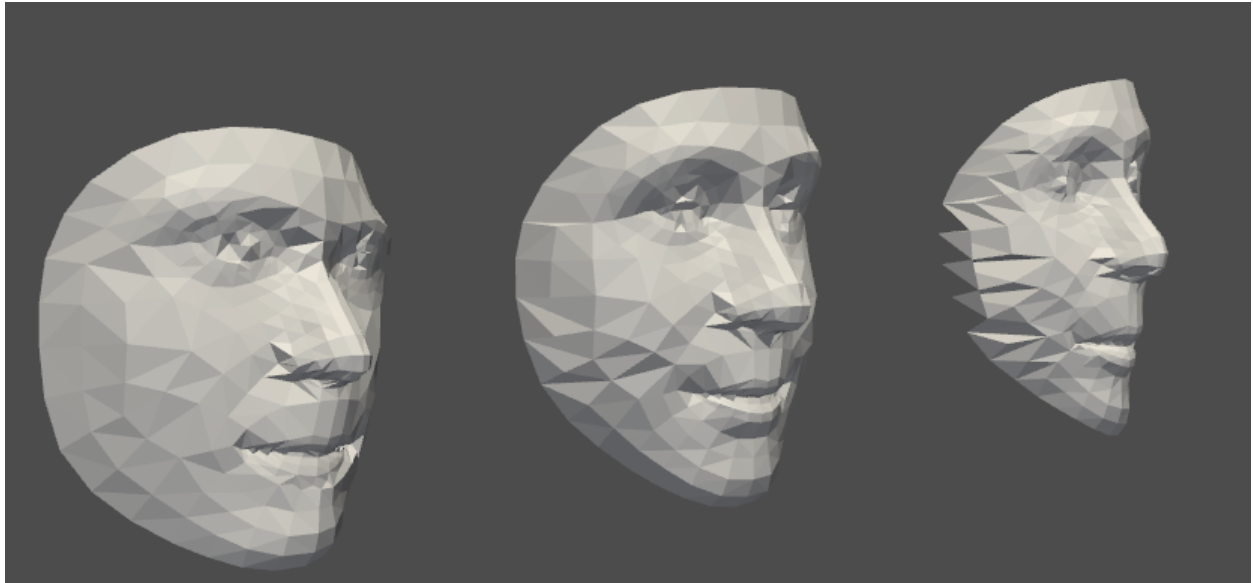


Figure 4.2: Rezultate din unghiuri diferite

Se poate vedea ca din poza frontala este generat un mesh care are cea mai mare fidelitate pentru ochi, nas si gura, dar cu mai mult "zgomot" la obraji. In mod nesurprinzator, din pozele din dreapta si din stanga sunt generate rezultate mai bune pentru partea vizibila in ele, dar partea care nu se vede este distorsionata.

Diferentele intre calitatea rezultatelor se pot vedea mai bine din lateral:



Pentru a combina cele mai bune parti din toate rezultatele, folosim un algoritm de interpolare in care atribuim liniar weighturi fiecarui rezultat. Punctele din jurul centrului (pe orizontala) vor fi cel mai mult afectate de meshul generat din poza frontala, iar meshurile din profil vor avea o influenta mai mare asupra partii care a fost vizibila in poza din care au fost generate. Generarea noului mesh consta in centrarea celor trei rezultate in acelasi punct, iterarea prin cele 468 de landmarks si calcularea unei medii ponderate a coordonatelor pe care le au meshurile pentru acel punct.

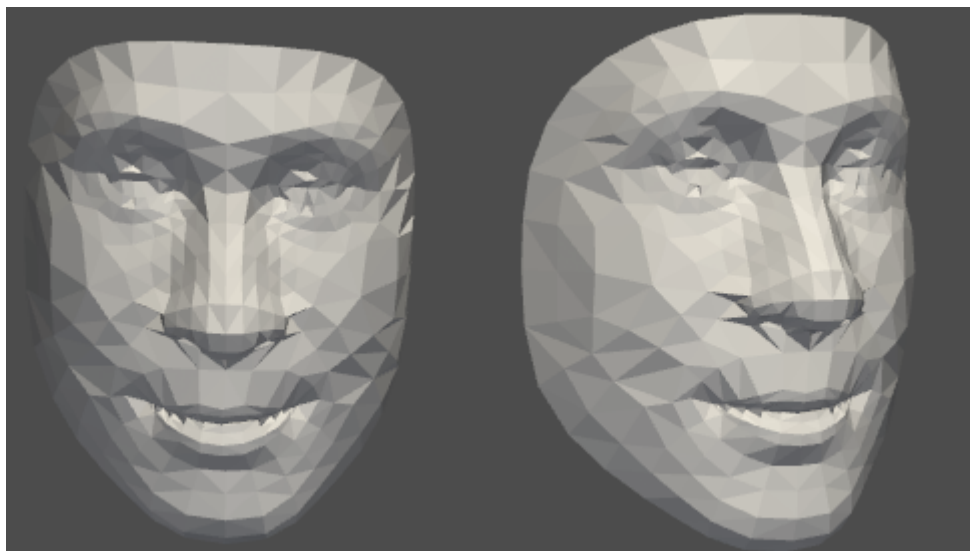


Figure 4.3: Rezultatul interpolarii

Capitol 5

Rezultate experimentale obtinute

5.1 Incercari cu Pixel2Mesh++

Folosind Pixel2Mesh++ [?], dorim sa obtinem meshul pentru partea de gat si eventual umeri, peste care sa suprapunem meshul pentru fata obtinut cu MediaPipe [1] (findca este foarte fidel)

Am incercat generarea cu modelele pretrained pentru coarse si refined¹, si am obtinut urmatoarele rezultate:

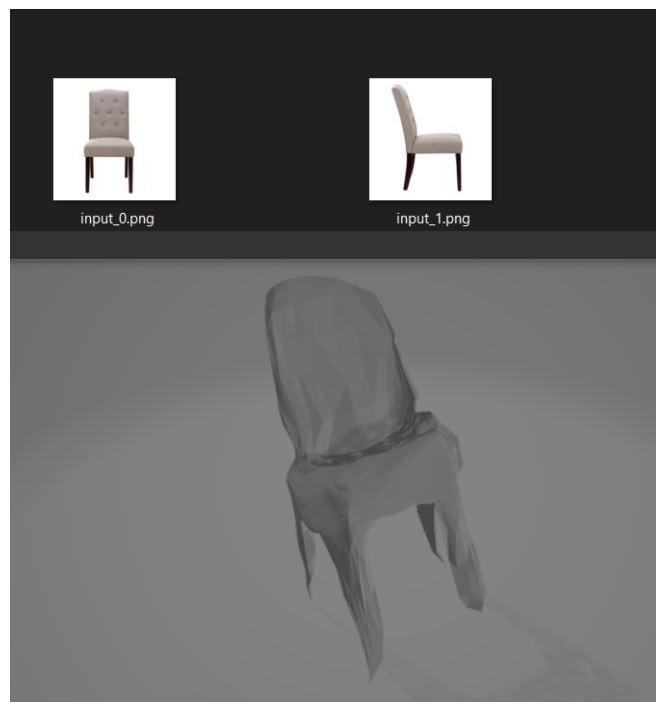


Figure 5.1: Rezultat doua imagini cu un scaun

¹Modele luate de [aici](#)



Figure 5.2: Rezultat John Cena cu background



Figure 5.3: Rezultat John Cena fara background

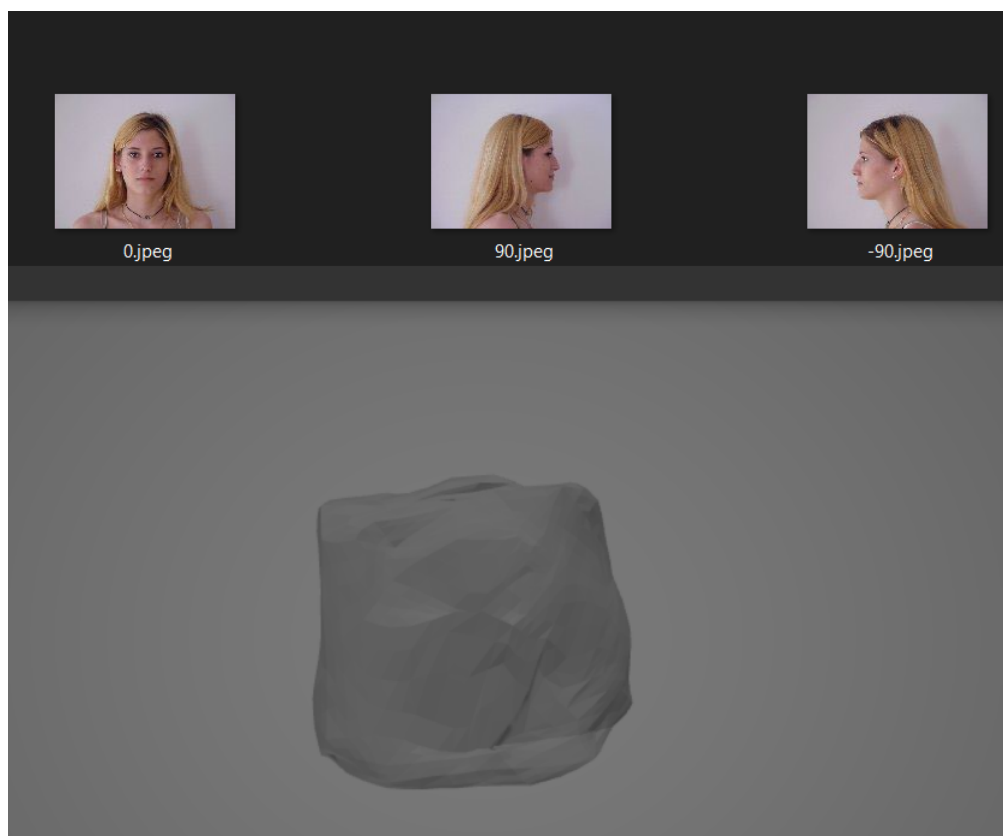


Figure 5.4: Rezultat trei unghiuri cu background

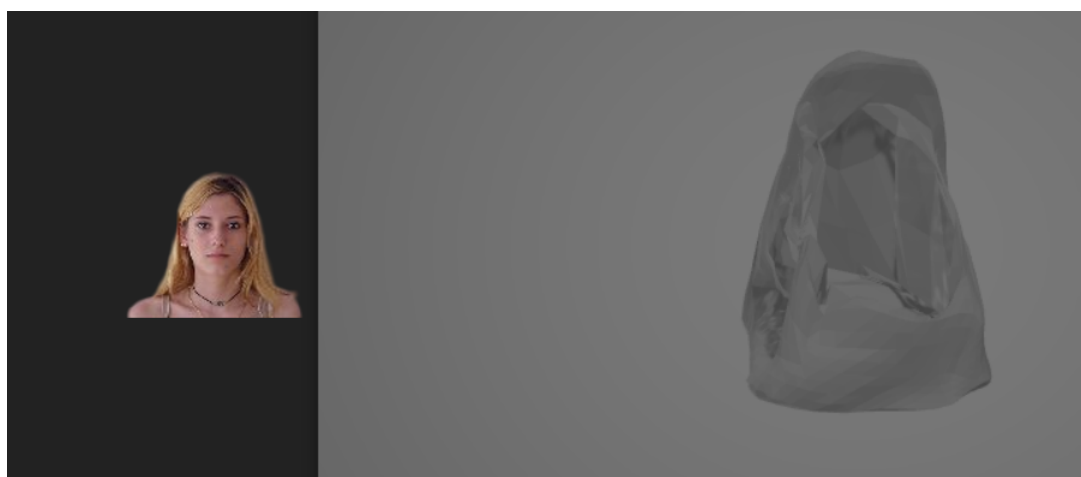


Figure 5.5: Rezultat frontal fara background

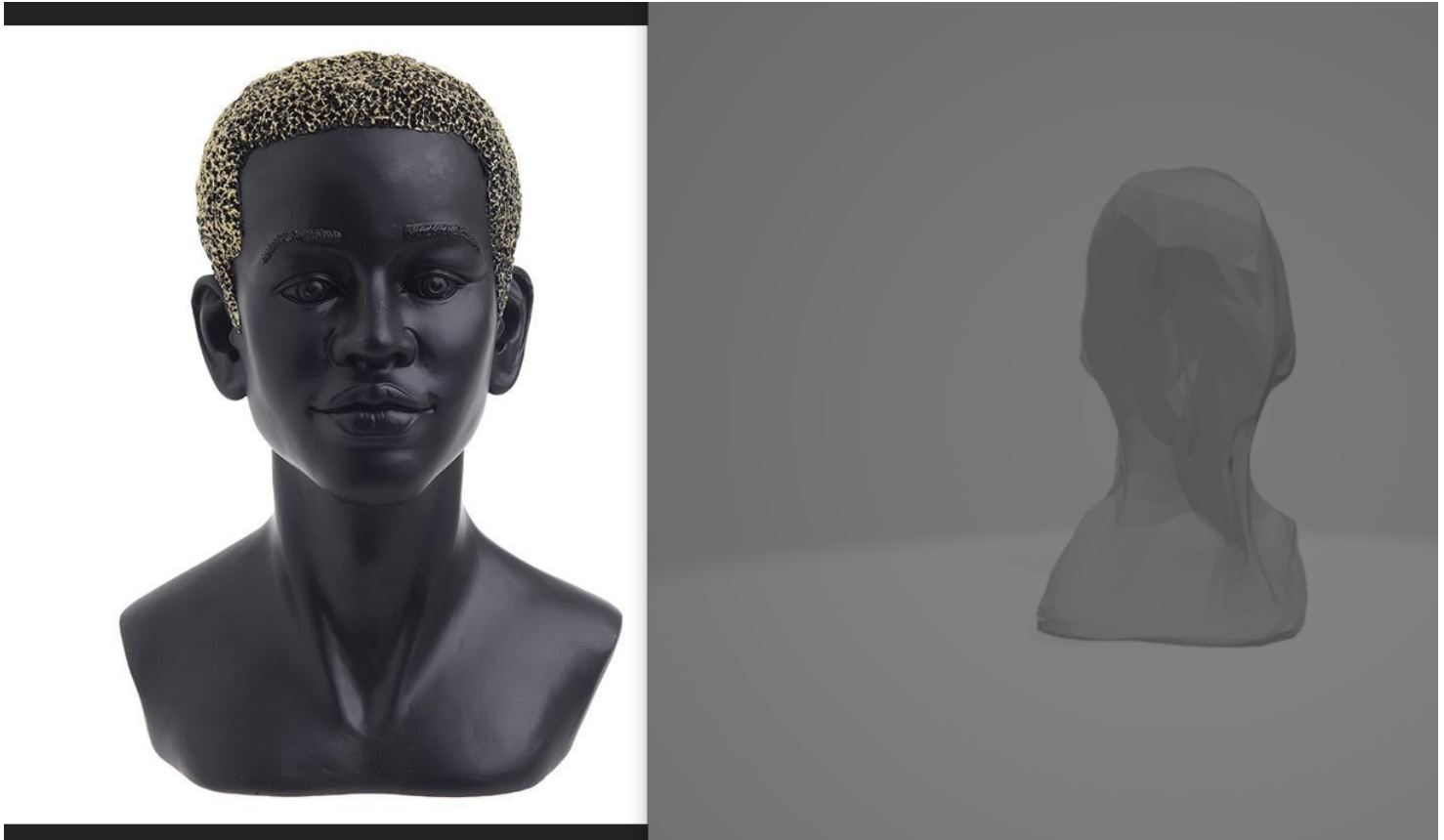


Figure 5.6: Rezultat bust alb inchis

In urma acestor teste am facut urmatoarele observatii:

- Imaginile nu trebuie sa aiba background, altfel outputul e un kinda cubic blob
- Daca e un contrast mare intre piele si alte elemente din poza (de ex. parul), partea mai deschisa la culoare pare sa fie considerata mai departe, uneori fiind eliminata din obiectul generat
- Daca bustul se extinde pana la marginea imaginii, generarea obiectului are de suferit
- Algoritmul este rasist

Capitol 6

Concluzii si posibile imbunatatiri

6.1 Posibile imbunatatiri

S-ar putea obtine rezultate mai bune prin generarea unui obiect 3D folosind pixel2mesh++, acesta avand ca output modelul intregului bust, nu doar cel al fetei. Dupa ce se obtine acest model, prin interpolare sau alte tehnici de combinare ale outputurilor furnizate de cei doi algoritmi, s-ar putea obtine un model complet al bustului pacientului, avand in componenta atat partea de umeri si gat, cat si un model foarte precis al fetei.

Capitol 7

Latex examples

Item example:

- content of item1
- content of item2
- content of item3

Figure example

... (see Figure 7.1)

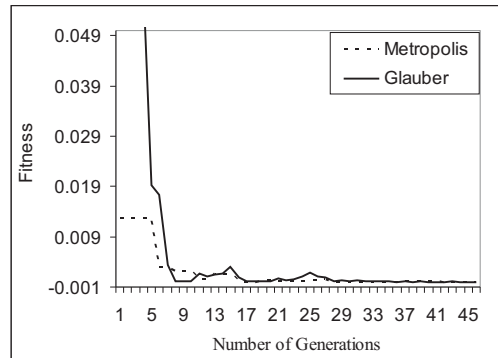


Figure 7.1: The evolution of the swarm size during the GA generations. This results were obtained for the f_2 test function with 5 dimensions.

Table example: (see Table 7.1)

Algorithm example

... (see Algorithm 1).

Table 7.1: The parameters of the PSO algorithm (the micro level algorithm) used to compute the fitness of a GA chromosome.

Parameter	Value
Number of generations	50
Number of function evaluations/generation	10
Number of dimensions of the function to be optimized	5
Learning factor c_1	2
Learning factor c_2	1.8
Inertia weight	$0.5 + \frac{rand()}{2}$

Algorithm 1 SGA - Spin based Genetic AAlgorithm

BEGIN

@ Randomly create the initial GA population.

@ Compute the fitness of each individual.

for i=1 TO NoOfGenerations **do**

for j=1 TO PopulationSize **do**

 p \leftarrow RandomlySelectParticleFromGrid();

 n \leftarrow RandomlySelectParticleFromNeighbors(p);

 @ Crossover(p, n, off);

 @ Compute energy ΔH

if ΔH satisfy the Ising condition **then**

 @ Replace(p,off);

end if

end for

end for

END

Bibliography

- [1] Google. Mediapipe face mesh. [link](#).
- [2] C. Bane Sullivan and Alexander Kaszynski. PyVista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK). *Journal of Open Source Software*, 4(37):1450, may 2019.
- [3] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *ICCV*, 2019.