BABEŞ 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 fizionommia pacientului, cu atat gradul de securitate al organelor care nu trebuie iradiate este mai mare.

Contents

List of Tables

List of Figures

List of Algorithms

Capitol 1

Introducere

In terapia radiologica a diverselor boli este nevoie de construirea unor masti de protectie a anumitor parti ale corpului (de ex. protejarea fetei). Daca aceste masti respecta cat mai mult fizionomia unui pacient, ele se transforma in aliati siguri ai pacientului, dar si ai medicului in combaterea bolilor si cresterea gradului de securitate a organelor care nu trebuie iradiate.

Constructia unei masti se poate adapta specificului unui pacient, respectand trasaturile fetei. In acest context se doreste dezvoltarea unei aplicatii care pe baza imaginii fetei unei persoane sa poata extrage caracteristicile faciale si sa frunizeze un model 3D a acestora. Acest model va putea fi folosit mai departe pentru a printa masca la o imprimanta 3D.

Scanarea 3D a obiectelor implica in mod obisnuit echipament specializat si o putere mare de calcul. Pentru a putea face acest proces mai accesibil, este necesara folosirea unor algoritmi inteligenti care pot extrapola modele 3D din doar cateva 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 [2]

3.2 Pixel2Mesh

Pixel2Mesh [7]

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 urmatorii pasi:

- Extragere caracteristici (folosind MediaPipe 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.

4.2 Detalii de implementare

Folosim MediaPipe pentru extragerea celor 468 de landmarks. Ca output avem coordonatele fiecarui punct, iar pentru generarea meshului le conectam folosind pyvista, 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 aproximate mai bine, iar cele obstructionate vor fi mai greu de generat).

Capitol 5

Application (Study case)

5.1 App's description and the main functionalities

Application description

Main functionalities and their specification

Add a flow diagram (very useful for the client)

5.2 App's design

- use cases
- diagrams (class diagram, sequence diagram, database structure)

5.3 Implementation

Give details about app's implementations (language, technologies, frameworks, special settings, etc.)

5.4 Testing of the codebase

Asserts, unitTestings, coverage, etc.

5.5 Numerical validation

Explain the experimental methodology and the numerical results obtained with your approach and the state of art approache(s).

Try to perform a comparison of several approaches.

Statistical validation of the results.

5.5.1 Methodology

- What are criteria you are using to evaluate your method?
- What specific hypotheses does your experiment test? Describe the experimental methodology that you used.
- What are the dependent and independent variables?
- What is the training/test data that was used, and why is it realistic or interesting? Exactly what performance data did you collect and how are you presenting and analyzing it? Comparisons to competing methods that address the same problem are particularly useful.

5.5.2 Data

Describe the used data.

5.5.3 Results

Present the quantitative results of your experiments. Graphical data presentation such as graphs and histograms are frequently better than tables. What are the basic differences revealed in the data. Are they statistically significant?

5.5.4 Discussion

- Is your hypothesis supported?
- What conclusions do the results support about the strengths and weaknesses of your method compared to other methods?
- How can the results be explained in terms of the underlying properties of the algorithm and/or the data.

Capitol 6

Conclusion and future work

Try to emphasise the strengths and the weaknesses of your approach. What are the major shortcomings of your current method? For each shortcoming, propose additions or enhancements that would help overcome it.

Briefly summarize the important results and conclusions presented in the paper.

- What are the most important points illustrated by your work?
- How will your results improve future research and applications in the area?

Capitol 7

Latex examples

Item example:

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

Figure example

. . . (see Figure 7.1)

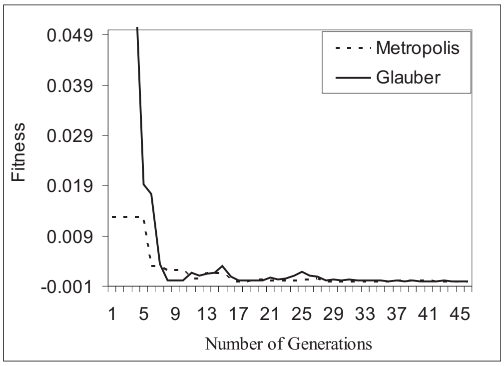

Figure 7.1: The evolution of the swarm size during the GA generations. This results were obtained for the f_2 test function with 5 dimensions.

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 AQlgorithm

```
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 ← RandomlySelectParticleFromGrid();
    n ← 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] E. Berlekamp, J. Conway, and J. Guy. *Winning Ways*. Academic Press Inc., London, 1982.

[2] Google. Mediapipe face mesh. link.

[3] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *International Conference on Neural Networks*, pages 1942–1948. IEEE, 1995.

[4] B. Koh, A. George, R. Haftka, and B. Fregly. Parallel asynchronous particle swarm optimization. *International Journal for Numerical Methods in Engineering*, 67(4):578–595, 2006.

[5] Mozilla. Firefox iOS application. link, 2018.

[6] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, March 1995.

[7] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++. link.