

# Reconstruction 3D

Team FaceSMesh

Salsigan Razvan-Dan

Pop Leonard-Andrei

Pislariu Alexandru Ilie

Prof. Coordonator: Diosan Laura

# Introducere

- 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.

## Solutia propusa

- Solutia propusa se bazeaza pe utilizarea unui algoritm inteligent pentru extragerea de puncte cheieale configuratiei fetei, sub forma unui nor de puncte.
- Odata ce aceste puncte sunt extrase, este necesara unirea lor, pentru a obtine o retea (mesh) corespunzatoare, ceea ce presupune generarea de poligoane in interiorul norului de puncte.
- Ultimul pas de efectuat dupa ce reteaua este generata este cel de asigurare a unei structuri printabile, prin ajustarea muchiilor retelei.

# Interfata si functionalitate

- In partea de sus a ecranului, apar instructiuni pentru utilizator, prin care i se comunica acestuia din ce unghi ar trebui facute pozele ca urmare a actionarii butonului Take Photo: prima din profil stanga, a doua frontal, iar ultima din profil dreapta.

# Interfata si functionalitate

- Dupa ce cea de-a treia poza a fost capturata, acestea se trimit serverului, iar in aplicatie apare un loading indicator si un mesaj.
- Odata ce imaginile au fost procesate si meshul generat, loading indicatorul dispare, iar utilizatorul este instiintat ca poate vizualiza wireframeul actionand butonul Show 3D.

# Server

Pe partea de server sunt disponibile 3 endpoints:

- /process

Primește prin POST cele 3 imagini sub forma de Base64 encoded byte array String, prin parametri image1, image2 și image3. Imaginile sunt trimise spre prelucrare și se răspunde cu OK dacă modelul 3D a fost generat cu succes.

# Server

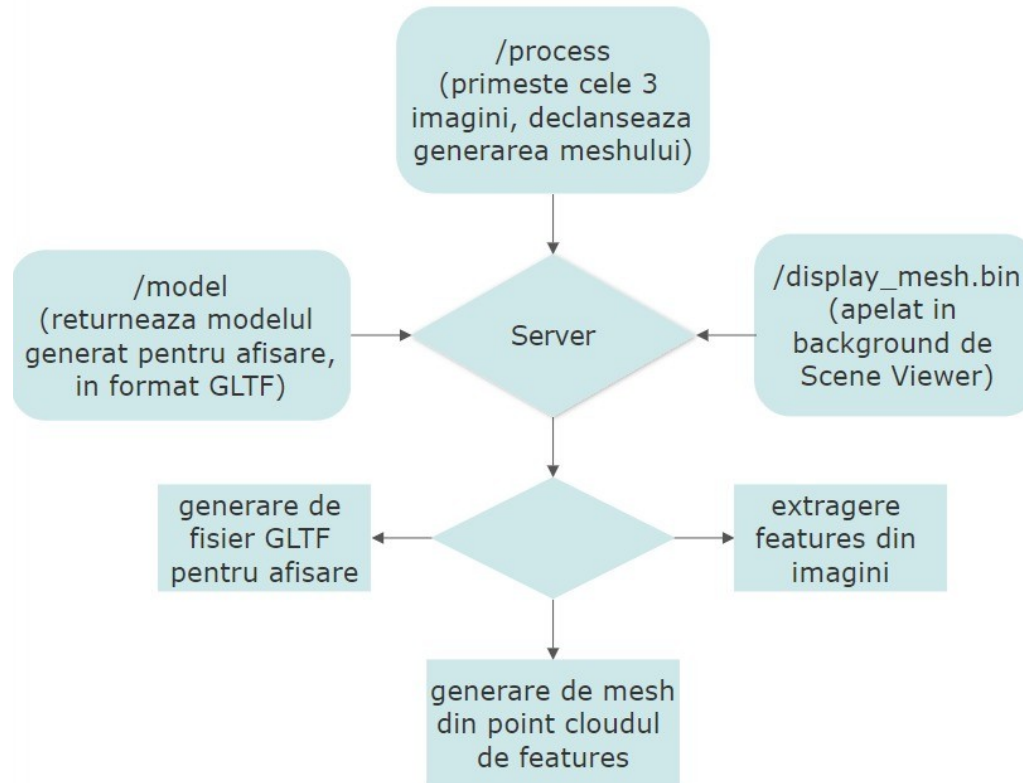
- `/model`

Acest endpoint este apelat de catre Scene Viewer si returneaza un fisier GLTF generat pe baza meshului rezultat din procesarea imaginilor.

- `/display_mesh.bin`

Deoarece fisierul GLTF generat de noi produce separat un bin file, parte din GLTF, Scene Viewer va face un request spre server pentru obtinerea acestuia.

# Server



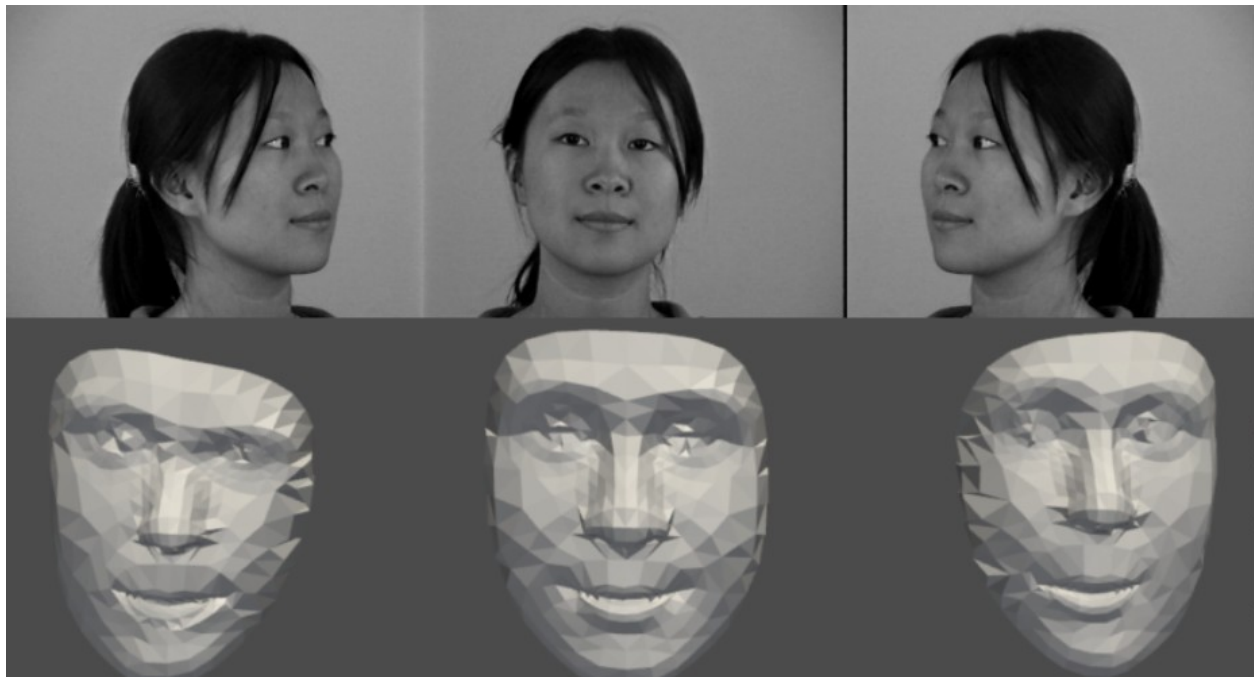


## Generarea modelului

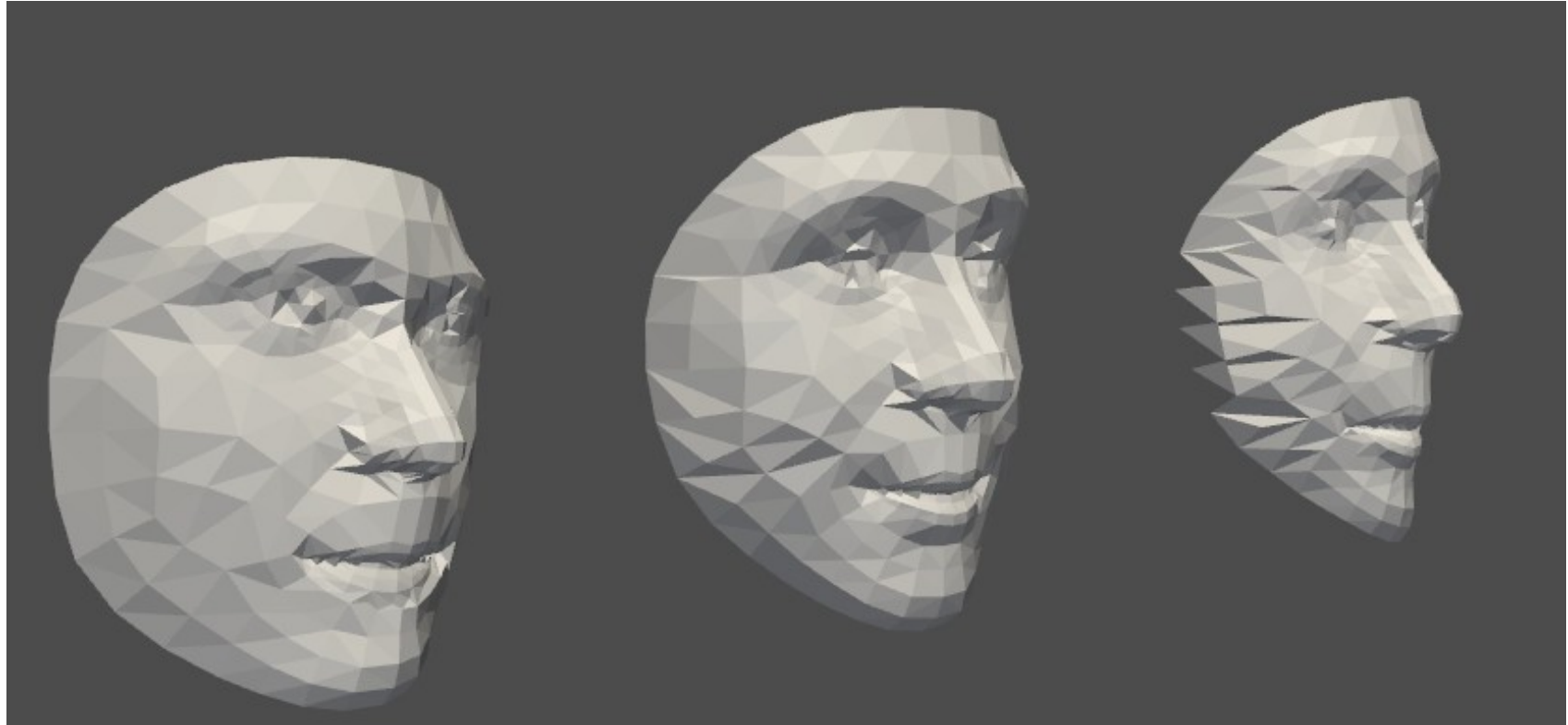
- Folosim MediaPipe pentru extragerea celor 468 de landmarks. Ca output avem coordonatele fiecarui punct, iar pentru generarea meshului le conectam folosind pyvista, aplicand filtrul delaunay pe point cloud.

# Generarea modelului

- In functie de unghiul din care a fost facuta fotografia, anumite zone din rezultat au o fidelitate mai mare.

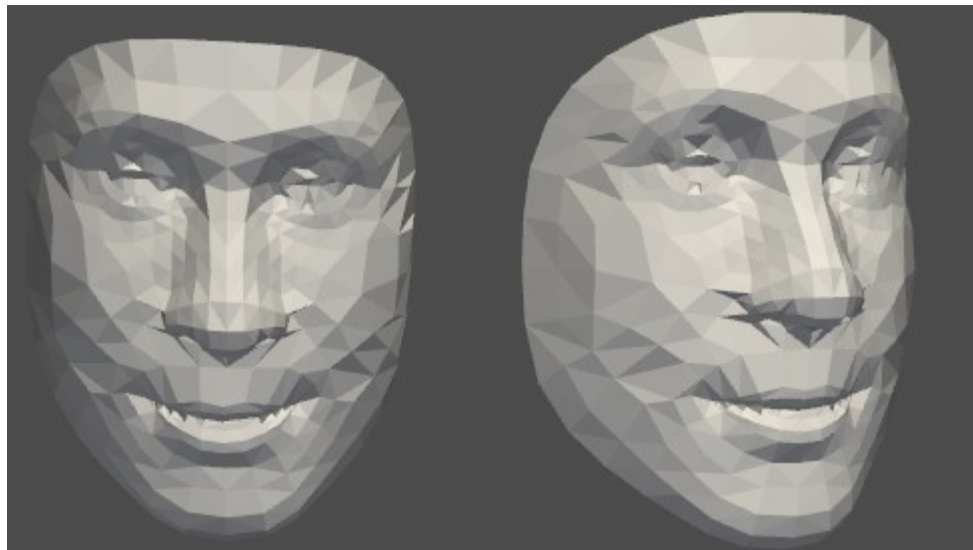


# Generarea modelului



# Generarea modelului

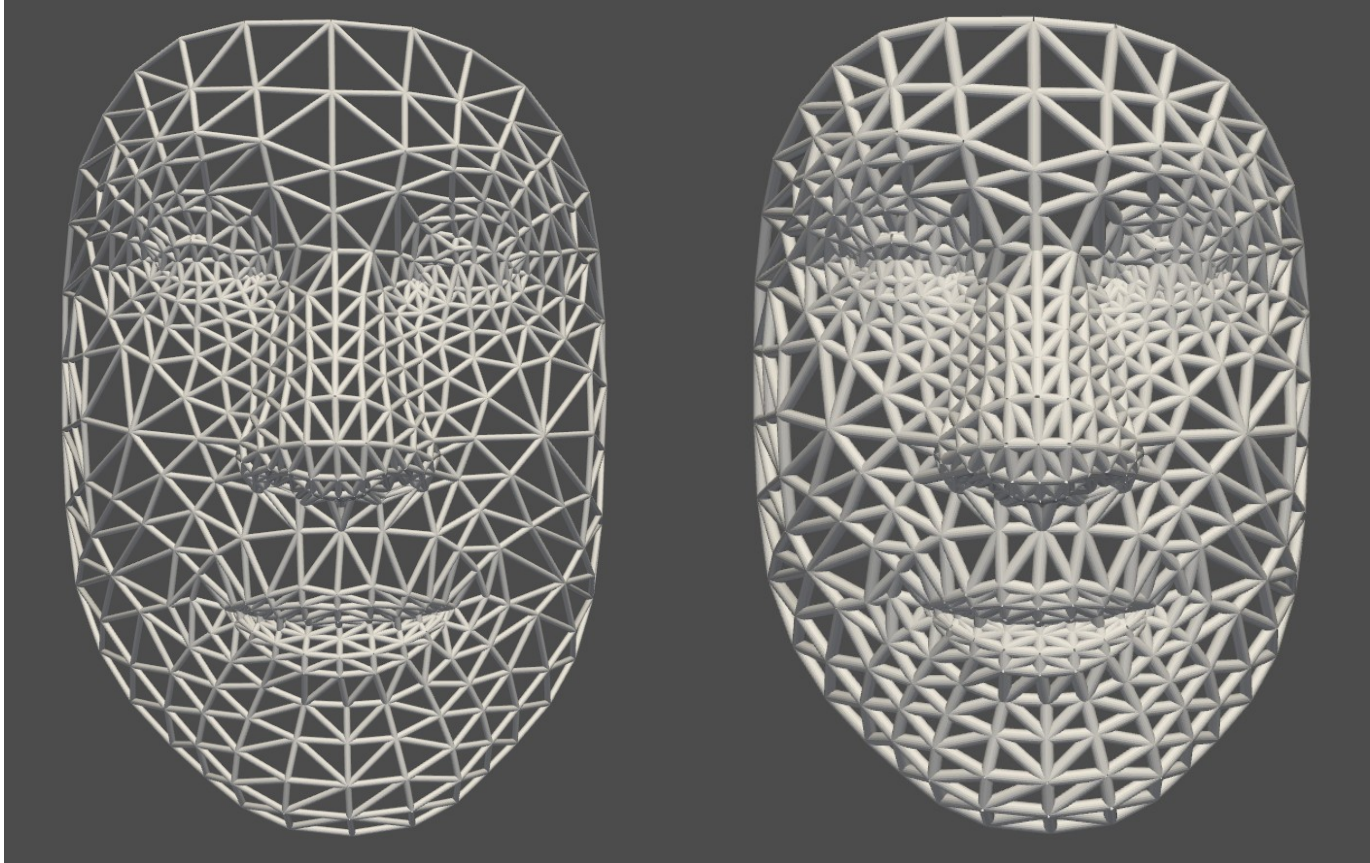
- Algoritmul nostru primește ca input cele 3 imagini, iar pentru a combina cele mai bune parti din toate rezultatele, folosim un algoritm de interpolare in care atribuim liniar weighturi fiecarui rezultat.



# Generarea modelului

- Din rezultatul interpolării extragem muchiile și generăm “tuburi” în forma lor, pentru a obține masca finală (grosimea e ajustabilă).

# Generarea modelului



# Concluzii

- Am reuist sa obtinem rezultate destul de bune folosind MediaPipe si interpolarea intre modelele generate din mai multe unghiuri, iar o posibila imbunatatire ar fi sa generam si partea de gat si umeri a persoanei, apoi sa combinam cele doua rezultate.

# System requirements

- Pentru vizualizarea modelului in Scene Viewer, dispozitivul cu android trebuie sa aiba suport pentru ARCore si Android 7.0 Nougat (API Level 24) sau mai recent.



# Referinte

- [https://google.github.io/mediapipe/solutions/face\\_mesh.html](https://google.github.io/mediapipe/solutions/face_mesh.html)
- <https://github.com/microsoft/Deep3DFaceReconstruction>
- <https://developers.google.com/ar/develop/java/scene-viewer#3d-or-ar>
- <https://stackoverflow.com/questions/54898657/i-want-to-generate-a-mesh-from-a-point-cloud-in-python>
- <https://docs.pyvista.org/examples/00-load/create-tri-surface.html>
- [https://www.researchgate.net/publication/220427980\\_3D\\_Reconstruction\\_from\\_Multiple\\_Images\\_Part\\_1\\_-\\_Principles](https://www.researchgate.net/publication/220427980_3D_Reconstruction_from_Multiple_Images_Part_1_-_Principles)
- <https://arxiv.org/pdf/1607.00662.pdf>
- <https://arxiv.org/pdf/1906.06543.pdf>
- [https://docs.opencv.org/2.4/modules/calib3d/doc/camera\\_calibration\\_and\\_3d\\_reconstruction.html](https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html)
- <http://3d-r2n2.stanford.edu/viewer/>
- Dataset: [http://robotics.csie.ncku.edu.tw/Databases/FaceDetect\\_PoseEstimate.htm#Our\\_Database\\_](http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm#Our_Database_)