

Reconstruction 3D

Team FaceSMesh

Salsigan Razvan-Dan
Pop Leonard-Andrei
Pislariu Alexandru Ilie

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.

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.

Interfata

- In partea de sus a ecranului apar instructiunile de folosire (cum trebuie facute fotografiile).
- La inceput este afisat “Take a left profile picture (-50°)”, prin care se sugereaza ca imaginea ce urmeaza a fi capturata sa fie una din profil stanga (la un unghi de -50° fata de planul frontal).
- Dupa ce prima poza este capturata, utilizatorul va reveni in aplicatie si va vedea fotografia si mesajul “Take a frontal picture” prin care se sugereaza ca imaginea ce urmeaza a fi capturata sa fie una frontala.
- Dupa ce a doua poza este capturata, utilizatorul va reveni in aplicatie si va vedea fotografia si mesajul “Take a right profile picture (50°)”, prin care se sugereaza ca imaginea ce urmeaza a fi capturata sa fie una din profil dreapta (la un unghi de +50° fata de planul frontal).

Interfata

- Dupa ce fiecare imagine este capturata, aplicatia afiseaza un toast message prin care utilizatorul este informat cu privire la numarul de imagini pe care mai trebuie sa le captureze.
- Dupa ce si cea de-a treia poza a fost capturata, acestea se trimit catre server si apare un loading indicator, alaturi de mesajul “The mesh is being generated...”.
- Cand meshul se termina de generat, apare un toast message care confirma acest fapt, si sugereaza vizualizarea meshului in Scene Viewer.

Interfata

- Interfata cu utilizatorul dispune de 2 butoane:
 - 1) Take photo - deschide camera si salveaza temporar imaginea capturata
 - 2) Show 3D – afiseaza meshul generat in Scene Viewer

Arhitectura si functionalitate

- Aplicatia Android 3DReconstruction reprezinta clientul in cadrul acestui proiect.
- Aplicatia este compusa dintr-o activitate cu un fragment principal. Butoanele, mesajele informative si fotografiile sunt parte a acestui fragment.

Arhitectura si functionalitate

- La actionarea butonului Take photo, se lanseaza un `ActivityResultCaller`, avand ca si parametru call contractul `TakePicture`, care deschide camera default a dispozitivului, iar dupa ce se captureaza o imagine, asteapta confirmarea ca este in regula, dupa care trimite inapoi in aplicatie un URI corespunzator locatiei temporare unde este salvata.
- In continuare, imaginea este convertita intr-un byte array encoded in Base64, sub forma de String, care este salvat intr-un container.

Arhitectura si functionalitate

- Odata ce in container sunt stocate 3 imagini, acestea sunt trimise catre server, iar utilizatorul vede un loading indicator.
- Clientul HTTP este de tipul OkHttpClient, iar requestul se face printr-un post, avand ca body un form cu cele 3 imagini.

Arhitectura si functionalitate

- Deoarece Google AR Core are nevoie de obiecte in format GLTF stocate pe un server, raspunsul serverului la requestul de mai sus este doar o confirmare ca meshul a fost generat cu succes, la primirea caruia loading indicatorul dispare, iar utilizatorului i se sugereaza sa vizualizeze obiectul generat prin apasarea butonului Show 3D.

Arhitectura si functionalitate

- La apasarea butonului Show 3D, aplicatia face un apel la Scene Viewer, caruia ii furnizeaza URL-ul unde se gaseste modelul generat pe server, dupa care deschide aplicatia AR Viewer, unde incarca meshul ce poate fi vizualizat.

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.

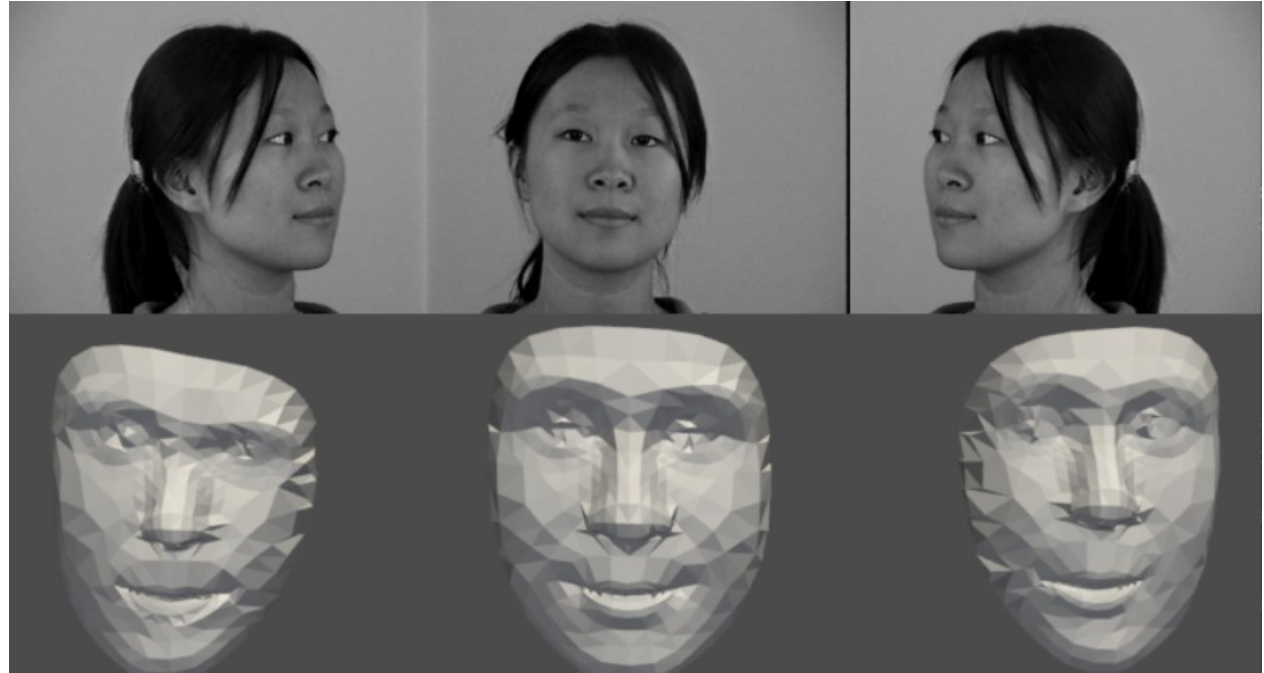
Server

Pe partea de server sunt disponibile 3 endpoints

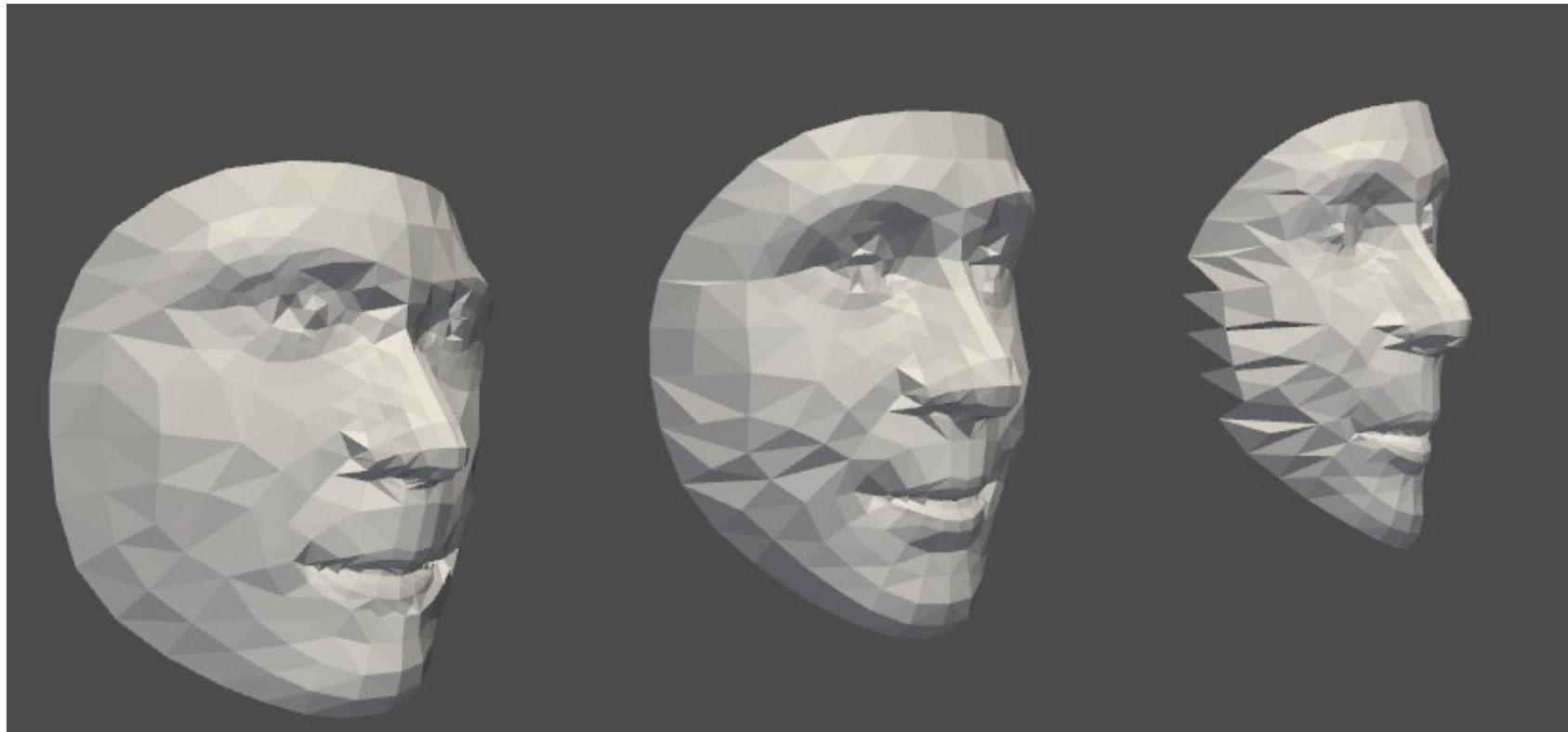
- /process
- Primeste prin POST cele 3 imagini sub forma de Base64 encoded byte array String, prin parametri image1, image2 si image3.
- Imaginile sunt trimise spre prelucrare si se raspunde cu OK daca modelul 3D a fost generat cu success.

Generarea modelului

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

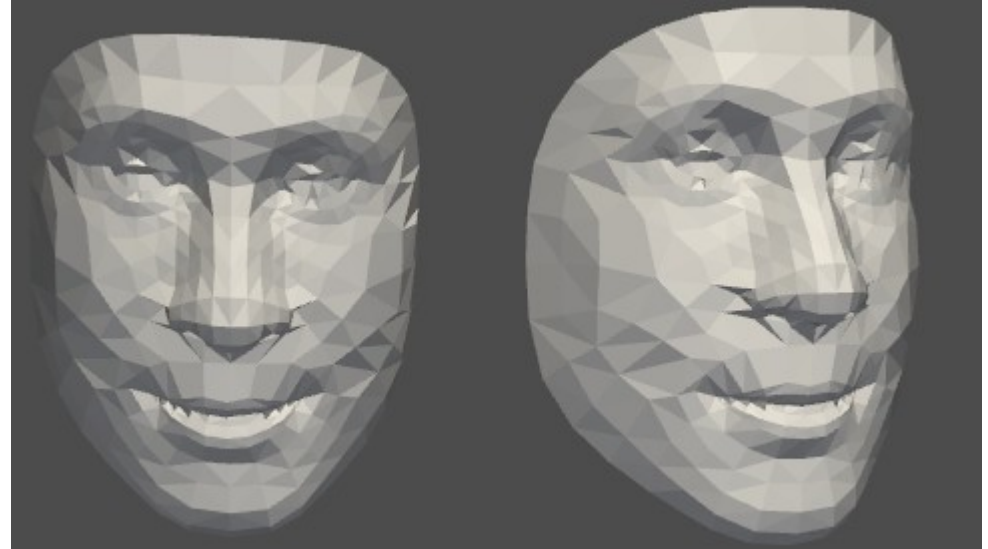


Generarea modelului



Generarea modelului

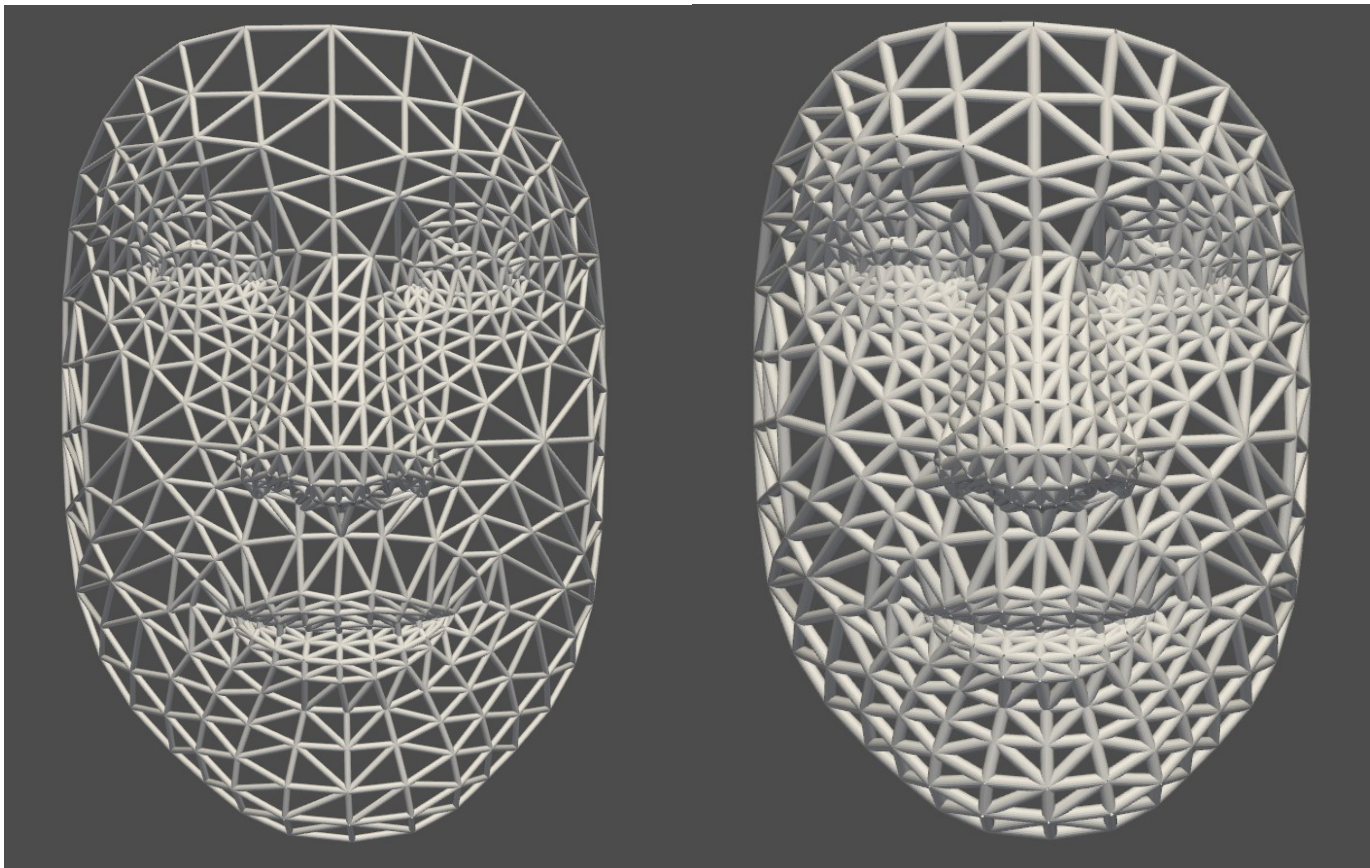
- Algoritmul nostru primește ca input cele 3 imagini, iar pentru a combina cele mai bune parti din toate rezultatele, folosim un algoritm de interpolare in care atribuim liniar weighturi fiecarui rezultat.



Generarea modelului

- Din rezultatul interpolarii extragem muchiile si generam “tuburi” in forma lor, pentru a obtine masca finala (grosimea e ajustabila)

Generarea modelului



Concluzii

- asd

Referinte

- 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://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
- <http://3d-r2n2.stanford.edu/viewer/>
- Dataset: http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm#Our_Database_