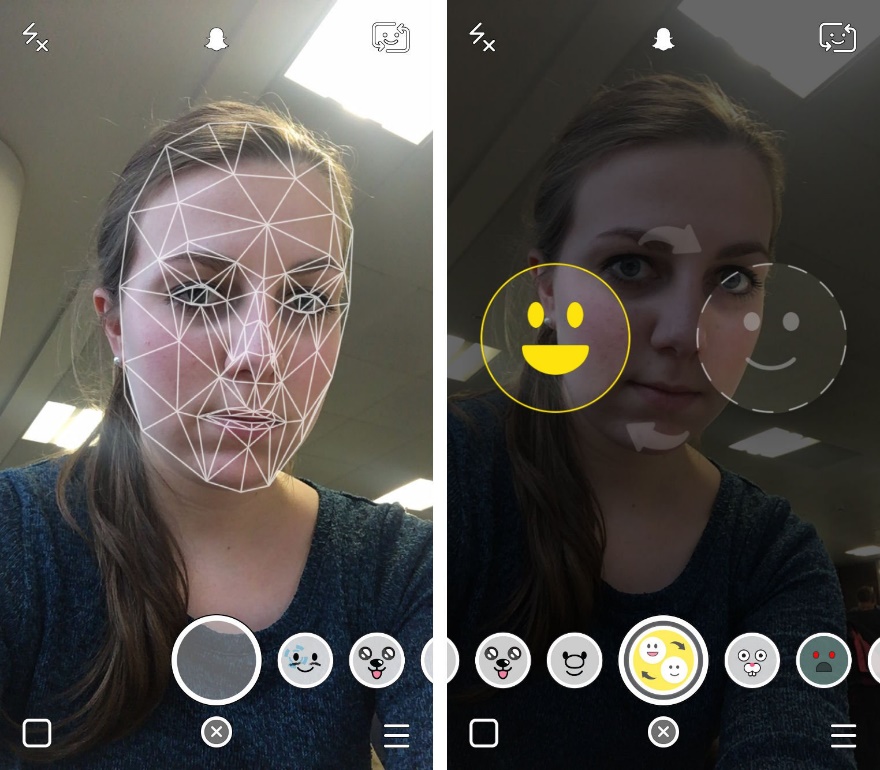
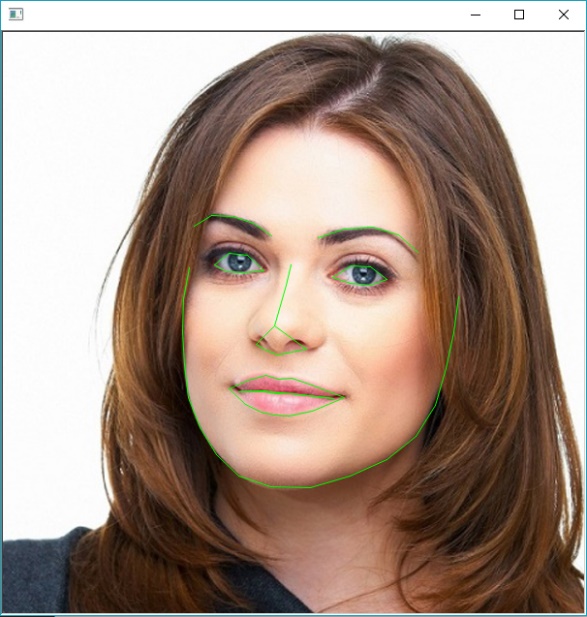
**3D реконструкција на лице**

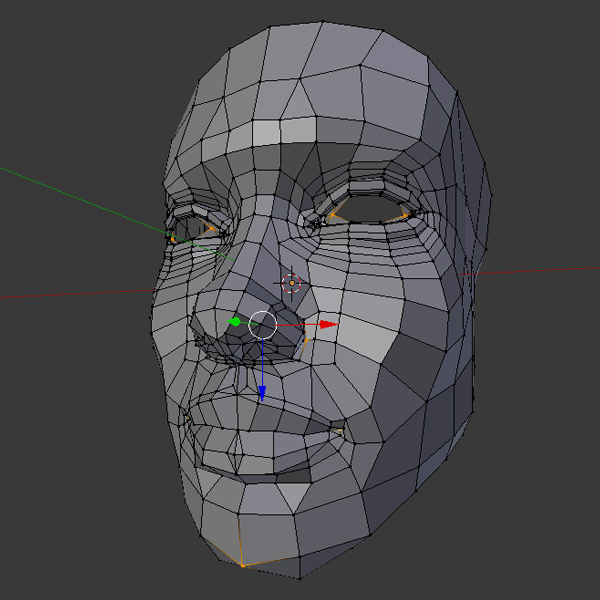
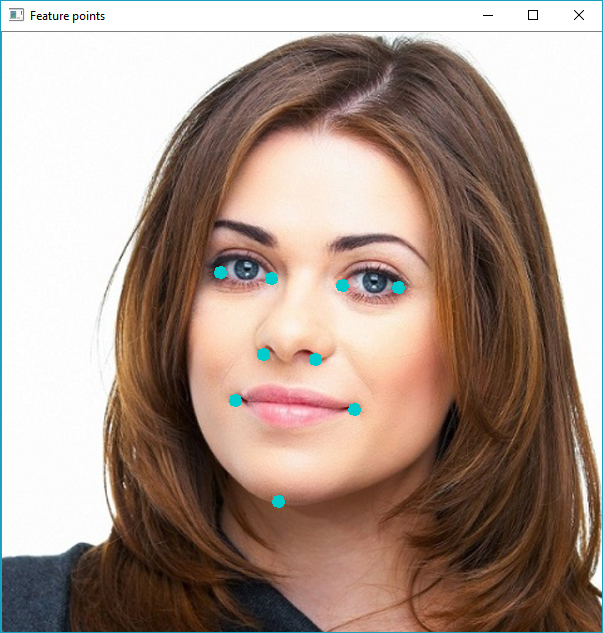
Голем дел од модерните апликации поддржуваат детектори на лица. Уште повеќе, дел од нив се способни за препознавање на позицијата и ротацијата на лицето релативно во однос на камерата, неговата топологија и слично. Користејќи ја оваа информација истите се способни да креираат ефекти на AR кои го подобруваат корисничкото искуство.

 Пример за таква апликација е Snapchat. Во продолжение е дадена слика од апликацијата и детекцијата на лице за понатамошно аплицирање на ефект (пр. замена на лица):

Инспирирајќи се од постоечките технологии, целта на овој проект е да направи 3D реконструкција на детектирано лице во виртуелен простор, апроксимирајќи ја неговата топологија и позиционирајќи го онака како што е позиционирано во реалниот свет.

 За таа цел се употребени библиотеките OpenCV, dlib и OpenGL API.

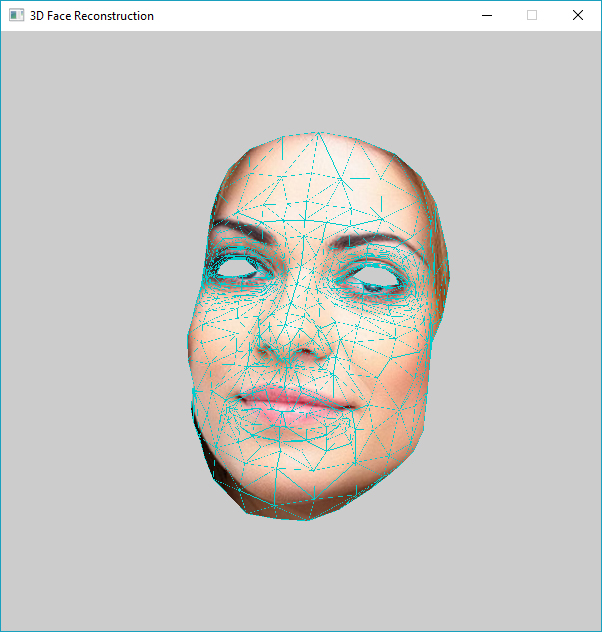
Вчитувањето на слика е прилично едноставен процес. Следен чекор е детекција на лицето. За таа цел се користи детектор на фронтално лице (frontal face detector) и предвидувач на форма (shape detector) на dlib. Предвидувачот на форма е иницијализиран со shape\_predictor\_68\_face\_landmarks, кој претставува однапред научен предвидувач. Детекторот ќе ги детектира сите лица кои се видливи, додека предвидувачот ќе ги извлече карактеристичните точки од лицето.

За да се извлечат потребните параметри на трансформацијата, неопходно е да се креираат два вектори, еден со 2D точки кои се детектирани и еден со 3D точки на моделот кој ќе се користи за апроксимација на формата на лицето. Точките на овие два вектори треба да бидат тополошки соодветни еден на друг. За таа цел се користат следните карактеристични точки:

Карактеристичните точки од 3D моделот се зачувани во низа која е непроменлива независно од лицето кое е детектирано.

Откако двата вектори се успешно креирани, следува извлекување на параметрите на трансформацијата. За таа цел се решава PnP (Perspective-n-Point). По решавањето на PnP добиваме матрица на ротација и вектор на транслација. За успешна трансформација потребно е да се транспонира матрицата на ротација, бидејќи OpenGL е column-major, каде пак OpenCV е row-major. Од векторот на транслација е креирана матрица на транслација користејќи ја glm библиотеката. Множењето на овие две матрици, ја креира копозитната матрица на трансформација.

За исцртување на лицето на соодветна позиција во 3D просторот се користи програмабилниот дел на OpenGL, односно shaders. Креирани се два, и тоа vertex и fragment shader. Целта на првиот е да ја изврши трансформацијата на сите точки од 3D моделот. Добиената матрица на трансформација со решавањето на PnP се предава како model матрица во овој shader. Покрај позиционирањето, потребно е и да се проектира сликата врз 3D моделот со што истиот ќе се обезбеди со информација за бојата. За таа цел се користи NDC координатите на точките од моделот мапирани во интервал [0, 1] за успешно семплирање на текстурата.

Како резултат е добиено следново: