

# МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ТРЁХМЕРНОГО СКАНИРОВАНИЯ ГОЛОВЫ ЧЕЛОВЕКА С ПОМОЩЬЮ КАМЕРЫ СМАРТФОНА

Разработано мобильное приложение для Android, позволяющее получить текстурированную трёхмерную модель головы человека с помощью стандартной камеры смартфона. Интерфейс на Kotlin направляет пользователя при съёмке 24 кадров с разных ракурсов, а ядро реконструкции, написанное на Rust, выполняет предварительную обработку прямо на устройстве. Для высококачественной реконструкции данные передаются на локальный сервер с видеокартой NVIDIA RTX 2060 (6 ГБ). Средняя точность серверной модели составила 3,8 мм при времени обработки 4–5 минут, что сопоставимо с результатами работы более дорогих сканеров.

## Введение

Трёхмерные модели головы человека востребованы в нескольких быстро развивающихся областях: персонализированная 3D-печать (фигурки, ортопедические изделия), создание цифровых аватаров для видеоигр и метавселенных, пластическая и челюстно-лицевая хирургия, а также криминалистическая идентификация. Профессиональные структурированные сканеры, обеспечивающие суб-миллиметровую точность, стоят от 5 000 долларов, что делает их недоступными для массового пользователя.

Разработанное приложение решает текущие проблемы: оно работает с обычной камерой любого Android-смартфона, предоставляет интуитивно понятный пользовательский интерфейс и позволяет выполнять реконструкцию либо непосредственно на устройстве, либо на локальном компьютере в той же сети — без передачи данных третьим лицам.

## I. Архитектура системы

Система состоит из двух основных компонентов: мобильного приложения и серверной части. Мобильное приложение написано на Kotlin с использованием Android Camera2 API для точного управления камерой. Вычислительное ядро — библиотека на языке Rust, скомпилированная через Android NDK. Серверный конвейер реализован на Python с использованием библиотек COLMAP, Open3D и OpenCV [1].

Сервер принимает набор фотографий по REST API в локальной сети и возвращает готовую модель. Связь между телефоном и сервером осуществляется через автоматическое обнаружение сервера по протоколу mDNS [2].

## II. Процесс сканирования

Пользователь проходит три последовательных шага, каждый из которых сопровождается визуальными подсказками.

При запуске приложение анализирует видеопоток в реальном времени. Нативный модуль на вычисляет среднюю яркость кадра и оценку расстояния до объекта по размеру обнаруженного лица. Зелёный индикатор «Sharp» подтверждает достаточную резкость, цветная полоса прогресса информирует о готовности. Если освещение недостаточно или пользователь расположен слишком близко, выводятся соответствующие предупреждения.

После этого на экране отображается круговая направляющая с 24 точками захвата (рис. 1). Пользователь (или помощник) плавно перемещает телефон вокруг головы объекта. Текущая позиция выделяется жёлтой точкой, а успешно снятые ракурсы отмечаются зелёным. Камера фиксирует кадры автоматически при достижении каждой контрольной точки, одновременно сохраняя внутренние параметры камеры (фокусное расстояние, главная точка) из метаданных Camera2 API. Размытые кадры отбрасываются на лету с помощью оценки оператора Лапласа, реализованной в Rust-модуле. Весь процесс съёмки занимает 1,5–2,5 минуты.



Рис. 1 – Интерфейс приложения во время процесса сканирования

По окончании съёмки, приложение автоматически проверяет наличие локального сервера в сети. При его обнаружении 24 кадра с параметрами камеры передаются на сервер, и обработка занимает 4–5 минут. В противном случае на устройстве запускается генерация менее детализированной модели (15–25 минут). Пользователь получает уведомление с прогнозируемым временем завершения и может свернуть приложение.

## III. Технические решения

В основе реконструкции лежит метод *фотограмметрии*: алгоритм находит одни и те же харак-

терные точки поверхности на разных снимках и по их взаимному смещению восстанавливает трёхмерные координаты.

На первом этапе *COLMAP* выполняет извлечение признаков (*SIFT*), попарное сопоставление и разреженную реконструкцию методом *Structure-from-Motion* с *CUDA*-ускорением на видеокарте *NVIDIA RTX 2060* (6 ГБ *VRAM*). Затем алгоритм плотной реконструкции *PatchMatch Stereo* формирует облако из 8–12 млн точек. Библиотека *Open3D* преобразует облако в герметичный меш (150–200 тыс. полигонов) с помощью алгоритма экранированной реконструкции Пуассона. Текстура проецируется на готовый меш из исходных фотографий с коррекцией баланса белого, формируя атлас 4096×4096 пикселей. При обработке учитывается ограничение по видеопамяти: плотная реконструкция выполняется пакетами, чтобы не превысить порог в 6 ГБ *VRAM* [3].

На мобильном устройстве применяется нейросетевая модель *HMR 2.0*, адаптированная для области головы. Модель восстанавливает параметрическую форму по серии кадров, формируя меш из 40–60 тыс. полигонов с текстурой 1024×1024 пикселей. Инференс оптимизирован для мобильного *GPU* с использованием квантизации *INT8* через *TensorFlow Lite* [4].

После реконструкции система автоматически проверяет размеры полученной модели: ширина головы должна составлять 15–22 см, глубина — 18–24 см, высота — 19–25 см. Отклонения за пределы допустимого диапазона сигнализируют о возможных ошибках калибровки. Экспорт доступен в формате для редактирования в приложениях трёхмерного моделирования. Файлы сохраняются в памяти устройства.

#### IV. Результаты и оценка

Система протестирована на 34 участниках в возрасте 22–61 года с различными типами лиц, причёсок и наличием аксессуаров (очки, борода). Точность измерялась путём сравнения с эталонным сканером

*FARO Freestyle3D X* (заявленная точность 0,3 мм). Для каждого участника вычислялось среднеквадратичное отклонение точек поверхности между полученной моделью и эталоном.

Серверный вариант продемонстрировал среднюю ошибку 3,8 мм при 24 входных кадрах, что ниже порога визуальной заметности для большинства прикладных задач. Мобильный вариант показал среднюю ошибку 8,1 мм — приемлемо для предварительного просмотра, но недостаточно для 3D-печати деталей. Принципиальное отличие от существующих решений — оба варианта работают полностью *офлайн* и не передают биометрические данные третьим лицам.

#### Выводы

Разработанное приложение сочетает в одном продукте нативную *Android*-разработку с высокопроизводительным ядром на *Rust*, серверную фотограмметрию на базе *COLMAP* и полную автономность от облачных сервисов. Точность серверной реконструкции (3,8 мм) сопоставима с профессиональными сканерами, при этом пользователю не требуются специальные знания: интерфейс проводит через весь процесс за две минуты.

Дальнейшая работа будет направлена на повышение качества мобильного конвейера за счёт использования модели *PIFuHD* с ускорением на встроенном *GPU*, улучшение реконструкции волос и бороды, а также добавление поддержки *iOS* через кросс-компиляцию *Rust*-ядра.

#### Список литературы

1. Schönberger J. L., Frahm J. M. Structure-from-Motion Revisited // CVPR. — 2016.
2. Zhou Q.-Y. et al. Open3D: A Modern Library for 3D Data Processing // arXiv:1801.09847. — 2018.
3. Goel S. et al. Humans in 4D: Reconstructing and Tracking Humans with Transformers // ICCV. — 2023.
4. Saito S. et al. PIFuHD: Multi-Level Pixel-Aligned Implicit Function // CVPR. — 2020.

*Иосько Михаил Алексеевич*, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, ioskomihailaa@gmail.com.

*Савин Тимофей Александрович*, студент 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, gudiniboy@gmail.com.

*Научный руководитель: Шилин Леонид Юрьевич*, профессор кафедры информационных технологий автоматизированных систем БГУИР, доктор технических наук, shilin@bsuir.by.