

Оригинальная статья / Original article

УДК 519.85

<https://doi.org/10.21869/2223-1560-2025-29-2-186-200>



## Программное обеспечение для преобразования двумерных изображений в трехмерные модели

А. А. Зоткина<sup>1</sup>

<sup>1</sup> Пензенский государственный технологический университет  
пр. Байдукова/ул. Гагарина, д. 1а/11, г. Пенза 440039, Российская Федерация

e-mail: alena.zotkina.97@mail.ru

### Резюме

**Цель исследования.** Разработка программного комплекса для автоматического создания фотогенералистичных трехмерных моделей лиц на основе одного входного изображения, который обеспечит высокую степень детализации и реалистичности моделей, а также простоту использования,

**Методы.** В исследовании используется комплексный подход для создания фотогенералистичных трехмерных моделей лиц из двумерных изображений, основанный на методах обратного рендеринга и каскадных сверточных нейронных сетей (CNN). Основным элементом является трехмерная трансформируемая модель (3DMM), которая описывает геометрию и альбиноса лица через линейные комбинации базисов главных компонент (PCA). Для соответствия 3D-геометрии и 2D-изображению применяется слабая перспективная проекция, учитывающая углы Эйлера и условия освещения. Оптимизация целевой функции с использованием метода Гаусса-Ньютона минимизирует различия между входным и визуализированным изображениями, а коррекция глубины и деталей лица достигается через адаптацию 3D-графики. Линейная интерполяция альбиноса уточняет детали модели в ключевых областях, что способствует созданию высококачественных и реалистичных 3D-моделей лиц.

**Результаты.** В данной статье успешно реализован программный комплекс, способный генерировать фотогенералистичные трехмерные модели лиц из одномерных изображений с использованием обратного рендеринга и каскадных сверточных нейронных сетей. Проведенные эксперименты подтвердили способность алгоритма к восприятию важных характеристик лиц и создают возможности для дальнейших приложений в сферах компьютерной графики, анимации и виртуальных интерфейсов.

**Заключение.** Полученные результаты свидетельствуют о высокой эффективности разработанного алгоритма для генерации фотогенералистичных трехмерных моделей лиц из двумерных изображений. Кроме того, результаты подтверждают, что применение методов обратного рендеринга совместно с каскадными сверточными нейронными сетями позволяет добиться значительных улучшений в качестве визуализации.

**Ключевые слова:** программный комплекс; 3D-объект/модель; сверточные нейронные сети; рендеринг; 3DMM.

**Конфликт интересов:** Автор декларирует отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

**Для цитирования:** Зоткина А. А. Программное обеспечение для преобразования двумерных изображений в трехмерные модели // Известия Юго-Западного государственного университета. 2025; 29(2): 186-200. <https://doi.org/10.21869/2223-1560-2025-29-2-186-200>.

Поступила в редакцию 08.05.2025

Подписана в печать 20.06.2025

Опубликована 23.07.2025

© Зоткина А. А., 2025

## Software for converting two-dimensional images into three-dimensional models

Alena A. Zotkina<sup>1</sup> 

<sup>1</sup> Penza State Technological University  
1a/11, Bairukova ave. / Gagarina str., Penza 440039, Russian Federation

 e-mail: alena.zotkina.97@mail.ru

### Abstract

**Purpose of research.** Development of a software package for the automatic creation of photorealistic three-dimensional models of faces based on a single input image, which will provide a high degree of detail and realism of models, as well as ease of use,

**Methods.** The study uses an integrated approach to create photorealistic three-dimensional models of faces from two-dimensional images based on reverse rendering methods and cascading convolutional neural networks (CNN). The main element is a three-dimensional transformable model (3DMM), which describes the geometry and albedo of a face through linear combinations of principal component bases (PCA). To match the 3D geometry and the 2D image, a weak perspective projection is used, taking into account Euler angles and lighting conditions. Optimization of the objective function using the Gauss-Newton method minimizes the differences between the input and rendered images, and correction of depth and facial details is achieved through the adaptation of 3D graphics. Linear albedo interpolation clarifies the details of the model in key areas, which contributes to the creation of high-quality and realistic 3D models of faces.

**Results.** This article successfully implements a software package capable of generating photorealistic three-dimensional models of faces from one-dimensional images using reverse rendering and cascading convolutional neural networks. The experiments have confirmed the algorithm's ability to perceive important facial characteristics and create opportunities for further applications in the fields of computer graphics, animation and virtual interfaces.

**Conclusion.** The results obtained indicate the high efficiency of the developed algorithm for generating photorealistic three-dimensional models of faces from two-dimensional images. In addition, the results confirm that the use of reverse rendering methods in conjunction with cascading convolutional neural networks allows for significant improvements in visualization quality.

**Keywords:** software package; 3D object/model; convolutional neural networks; rendering; 3DMM.

**Conflict of interest.** The Author declare the absence of obvious and potential conflicts of interest related to the publication of this article.

**For citation:** Zotkina A. A. Software for converting two-dimensional images into three-dimensional models // Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University. 2025; 29(2): 186-200 (In Russ.). <https://doi.org/10.21869/2223-1560-2025-29-2-186-200>.

Received 08.05.2025

Accepted 20.06.2025

Published 23.07.2025

\*\*\*

### Введение

В современном мире существует значительная потребность в высококачественных 3D-моделях лиц, которые

могут быть использованы в различных областях, таких как игровые технологии, киноиндустрия, виртуальная реальность и безопасность. Традиционные методы создания 3D-моделей зачастую

требуют множества изображений или сложных процессов ручного моделирования, что ограничивает их применение и увеличивает временные и финансовые затраты [1, 2, 3, 4, 5, 6]. Кроме того, существует необходимость в автоматизации процессов генерации моделей лиц на базе ограниченного количества исходных данных. Это вызывает потребность в разработке интеллектуального программного комплекса, который мог бы создать фотoreалистичные 3D-модели с минимальными затратами времени и ресурсов.

Многие из традиционных методов создания 3D-моделей лиц полагаются на сложные процессы ручного моделирования или требуют большого объема изображений для обучения моделей. Эти методы могут включать комбинирование 2D изображений, сложные алгоритмы фотограмметрии или использование специализированного оборудования для 3D-сканирования. В предложенной работе акцент делается на автоматизации и использовании лишь одного изображения для генерации фотoreалистичной 3D-модели лица с помощью алгоритмов глубокого обучения. Использование каскадных CNN-сетей и 3D-моделей позволяет добиться высокой скорости обработки и относительно высокой точности преобразования.

Таким образом, разработка программного комплекса для построения фотoreалистичных 3D-моделей лиц на основе одного изображения является крайне актуальной задачей в условиях стремитель-

ного развития технологий и увеличения числа приложений, где требуется высокая степень реалистичности и детализации. В связи с этим, исследования и разработки в этой области способствуют удовлетворению растущих потребностей рынка и внедрению новейших технологий в повседневную практику.

## **Материалы и методы**

В процессе реализации будет разработан программный комплекс преобразования изображений в 3D объекты, используя основанный на обучении обратный рендеринг лица, новый метод генерации данных путем рендеринга большого количества изображений лиц, которые являются фотoreалистичными и обладают различными свойствами. На основе 3D-реконструкции лица с мелкомасштабной текстурой в реальном времени, включающей с помощью хорошо сконструированных наборов данных, обучаем две каскадные CNN-сети от грубой до точной обработки [7, 8, 9]. Сети обучены для реальной детальной 3D-реконструкции лица по одному изображению.

Основные этапы преобразования изображений в 3D объекты:

1. В специальную форму на главной странице пользователь загружает необходимую фотографию лица человека.
2. Далее происходит обработка фотографии и предварительная подготовка для преобразования.
3. После происходит извлечение низкочастотной и высокочастотной карты альбиноса.

4. После извлечения карты происходит извлечение текстуры.

5. Далее программный комплекс с помощью деформируемой модели создает 3D объект.

Трехмерная трансформируемая модель (3DMM) представляет собой низкоразмерную статистическую модель геометрии лица и альбено [10]. Она также помечает трехмерную геометрию лица по альбено для PCA:

$$S - \bar{S} = \sum_{\eta} A_{\eta} \alpha_{\eta}, \eta \in \{id, exp\}, \quad (1)$$

$$T - \bar{T} = \prod_{\sigma} \sigma_{alb}, \sigma \in \{A, \alpha\}. \quad (2)$$

Здесь  $\bar{S}$  и  $\bar{T}$  обозначают альбено и форму обычной трехмерной поверхности соответственно. Оси  $A_{id}$  и  $A_{alb}$  ориентируют при обучении на расположение текстурированных трехмерных сеток с непредвзятой оценкой. Более того, для отдельных людей  $A_{exp}$  является главной осью, построенной на смещениях между нейтральными сетками и сетками выражений.  $\alpha_{id}$ ,  $\alpha_{exp}$  и  $\alpha_{alb}$  – это векторы коэффициентов сравнения, которые описывают конкретного человека. В этой работе  $A_{id}$  и  $A_{alb}$  начинаются с BFM, а  $A_{exp}$  – с FaceWarehouse. То первые 154 параметра – это коэффициенты PCA, которые подчиняются нормальному распределению. Базис формы/альбено – это базис PCA; таким образом, мы знаем диапазон значений, который имеет смысл для предполагаемого нормального распределения совокупности

лиц в наборе данных. Базис выражения создает базис смешанной формы; таким образом, допустимые значения находятся в диапазоне  $[0, 1]$ , но мы не применяем это, поскольку в конечном итоге это ограничивает выразительность. Следующие три угла Эйлера предназначены для градуса. Вектор перемещения и параметр масштаба определяются положением и размером лица на плоскости изображения.

Чтобы построить соответствие между трехмерной формой лица, описанной с помощью 3DMM, и 2D-изображением, следуя, мы используем слабую перспективную проекцию:

$$a = f * M * R * V + t, \quad (3)$$

$$\text{где } a = \begin{pmatrix} a_x \\ a_y \\ a_{\eta} \end{pmatrix}, M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, V = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}, t = \begin{pmatrix} t_x \\ t_{\eta} \end{pmatrix}.$$

Здесь  $a$  и  $V$  – площади вершины в плоскости изображения и в глобальной системе координат соответственно [11, 12].  $f$  – масштабный коэффициент, а  $R$  – матрица поворота, полученная из углов Эйлера тангажа, рыскания, крена и,  $t$  – вектор интерпретации. Для условий освещения мы принимаем коэффициент отражения по Ламберту и аппроксимируем освещенность базисными функциями сферических гармоник второго порядка. После этого процедура рендеринга зависит от параметров:

$$\zeta = \{\alpha_{id}, \alpha_{exp}, \alpha_{alb}, f, y, b, k, t_x, t_{\eta}, L\}, \quad (4)$$

где  $y$  – высота тона;  $b$  – рыскание;  $k$  – крен.

$L$  обозначает эти сферические гармоники коэффициенты. С помощью этих параметров мы можем определить модель формирования изображения  $\zeta$ , которая позволяет нам создавать синтетические изображения виртуальных лиц. Учитывая параметры  $\zeta$ , которые определяют структуру всей сцены,  $\zeta = 245$  – общее количество параметров. Изображение лица визуализируется с параметром  $\zeta$ . Сначала создается текстуированная трехмерная сетка с использованием уравнений (1) и (2). Затем мы выполняем растеризацию с использованием проекции Инжир. Конвейер алгоритма 3D-реконструкции лица: учитывая входное изображение лица, предлагаются обратный рендеринг лица процесс состоит из трех этапов, а именно: грубый слой, 3DMM, параметры положения и уточнение геометрии, которые отображаются последовательно слева направо сверху вниз. Сначала мы реконструируем грубый слой одного изображения, а затем используем 3DMM, параметры положения и глубину грубого слоя для реконструкции сети тонких слоев. Обновление компонентов контролируется индикатором со стрелками функция. Растеризация поддерживает систему для получения основной список трех точек на трехмерной сетке и барицентрические координаты для каждого пикселя в области грани. Для каждого пикселя мы получаем качество альбено, просматривая барицентрический интерполированные вершины базового

треугольника и, как правило, используя обычные характеристики базового треугольника. Альбено и освещение используются для визуализации цвета пикселя на заключительном этапе.

$$\frac{C_s(p)}{\rho_p} = L^T \phi(n_p), \quad (5)$$

где  $C_s(p)$  указывает отображаемый цвет пикселя  $p$ ;

$\phi(n_p)$  – базовая сферическая гармоника второго порядка для нормального  $n_p$ ;

$\rho_p$  – альбено пикселя.

Наша система подбора зависит от методологии анализа с помощью синтеза. То есть мы ищем ответ на запрос  $\zeta$ , минимизируя различия между входным изображением лица и визуализируемым изображением для  $\zeta$ . Мы используем эмулирующую целевую функцию:

$$\varepsilon(\zeta) = \sum_{\sigma} w_{\sigma} \varepsilon_{\sigma} + \varepsilon_{col}, \sigma = \{lan, reg\}, \quad (6)$$

где  $\varepsilon_{reg}$  – термин для упорядочения;

$\varepsilon_{lan}$  – знаковый термин;

$\varepsilon_{col}$  – термин для согласования фотографий, в то время как  $W_{reg}$  и  $W_{lan}$  являются компромиссными параметрами;

$\varepsilon_{reg}$  – термин регуляризации, который гарантирует достоверность параметров 3DMM.

$$\begin{aligned} \varepsilon_{reg}(\zeta) = & \sum_{i=1}^{85} \left[ \frac{\alpha_{id,i}^2 \sigma_{alb,i}^2 + \alpha_{alb,i}^2 \sigma_{id,i}^2}{\sigma_{id,i}^2 \sigma_{alb,i}^3} \right] + \\ & + \sum_{i=1}^{69} \left( \frac{\alpha_{exp,i}}{\sigma_{exp,i}} \right)^2. \end{aligned} \quad (7)$$

Стандартное отклонение соответствующих основных трендов представлено в виде  $\sigma$ . В предлагаемом алгоритме используются 69 основных компонентов для выражения и 85 основных компонентов для определения идентичности и альбедо. Мы выбрали это число таким образом, чтобы оно отражало 99% энергии. Мощность представления достаточна, и они хорошо обобщаются на тестовом наборе. Мы оптимизируем целевую функцию 6, используя подход Гаусса-Ньютона.  $\varepsilon_{lan}$  – термин ROI (точка интереса), который приближает ожидаемую вершину к соответствующим ориентирам на плоскости изображения [13, 14]:

$$\varepsilon_{lan}(\zeta) = \frac{1}{|F|} \sum_{f_i \in F} \varepsilon_{f_i}^2, \quad (8)$$

где  $\varepsilon_x$  – евклидова норма для  $x$  И  $f_i = f_i - (t + \Pi RV_i)$ .

Положение интересующей точки на изображении представлено плоскостью  $f_i$ , а  $V_i$  показывает положение соответствующей вершины на красивой трехмерной грани.  $\varepsilon_{col}$  – термин для определения согласованности фотографий, который минимизирует расстояние между входным изображением лица и визуализируемым изображением с помощью пространства оттенков RGB по отношению к уровню пикселя:

$$\varepsilon_{col}(\zeta) = \frac{1}{|P|} \sum_{p \in P} \varepsilon_p^2, (p = C_s(p) - C_l(p)), \quad (9)$$

где  $C_s$  – визуализированное изображение;

$F$  – набор большей части пикселей в области лица;

$C_l$  – входное изображение.

Поскольку 3DMM – малоразмерная модель, отдельные детали лица, например, морщины и складки, не кодируются должным образом [15, 16]. Мы кодируем эти детали со смещением в направлении глубины для каждого пикселя. В частности, проецируя адаптированную трехмерную грань по параметру  $\zeta$ , можем получить значение глубины для каждого пикселя в области грани. Пусть  $x$  – все значения глубины в пикселях;  $F$  – все смещения, а  $\tilde{x}$  – все значения глубины, добавляя  $F$  к  $x$ . Затем нормаль в пикселе  $(i, j)$  может быть вычислена с помощью нормали треугольника  $P_{i,j}, P_{i+1,j}, P_{i,j+1}$ , где  $P_{i,j} = (i, j, \tilde{x}(i, j))$  – координата пикселя  $(i, j)$  рамки камеры. Для этого обновленного нормального значения  $C_s$  обновляется с помощью уравнения 8. Мы оцениваем  $F$  с помощью целевой функции:

$$\varepsilon(F) - \varepsilon_{col} = v_1 \|F\|_2^2 + v_2 \|LF\|_1, \quad (10)$$

где  $\varepsilon_{col}$  изображен по площади в уравнении 6, а  $LF$  показывает лапласианское смещение  $F$ , а  $v_1$  и  $v_2$  являются параметрами. Функция 10 оптимизирована с использованием итеративного подхода к повторному взвешиванию.

Как и в случае с геометрией, малая размерность делает  $\rho_c$  альбено гладким, что выражается в 3DMM [17, 18]. Фотореалистичный рендеринг, полученный на основе мелкомасштабного альбено, используется в следующем уравнении:

$$L^T \phi(n_p)^* \rho_f = C_I(p), \quad (11)$$

где  $C_I$  – цвет входного изображения в пикселе  $p$ ;  $n_p$  – нормаль в точке  $p$ , вычисленная исходя из этой улучшенной геометрии.

Однако из-за несовершенных геометрических приращений мелкомасштабное альбено  $\rho_f$  может содержать некоторые геометрические детали. Чтобы предотвратить это, мы линейно смешиваем  $\rho_c$  и  $\rho_f$  – оттенки, придавая различный вес различным участкам. Мы выполняем это на участках, где могут проявиться детали, например, в уголках глаз и на лбу. Кроме того, мы устанавливаем альбено, близкое к  $\rho_c$ .

## Результаты и их обсуждение

Данный проект реализован в формате SPA-приложения. SPA (Single Page Application) – это современный способ реализации веб-проектов. Главная идея состоит в том, что приложение или веб-сайт использует один HTML – документ для всех страниц приложения. Благодаря этому скорость загрузки страниц возрастает и пользователю приятнее и удобнее работать с системой. Frontend будет реализован на react.js. А вот Backend данного проекта реализован с помощью языка программирования Python, известного своей простотой и универсальностью, стал предпочтительным языком для машинного обучения. Простота синтаксиса и активная поддержка сообщества Python ML делают его популярным выбором

среди начинающих и опытных разработчиков [19].

Для построения трехмерной трансформируемой модели нам понадобится ряд библиотек, а именно:

- pytorch (оптимизированная тензорная библиотека для глубокого обучения с использованием графических и центральных процессоров) [20];
- torchvision (библиотека предварительной подготовки изображений оптимизированных для компьютерного зрения) [20];
- numpy (библиотека для произведения вычислений на python) [21];
- dlib (библиотека для обнаружения лиц на фотографии) [22];
- opencv (популярная библиотека компьютерного зрения на python) [23].

Чтобы найти лица на изображении, мы можем использовать библиотеку dlib. Процесс распознавания лица делится на несколько этапов:

- создание экземпляра детектора лиц на основе HOG в dlib и загрузка предсказателя ориентиров лица;
- загрузка и предварительная обработка изображений;
- обнаружение лиц на нашем изображении.

На рис. 1 показан фрагмент программного кода обнаружения лица.

Теперь, когда лицо обнаружено на фотографии, можно перебирать каждую из областей интереса лица индивидуально. Чтобы фактически извлечь каждую из областей лица, нам просто нужно вычислить ограничивающую рамку

координат (x, y), связанных с конкретной областью, и использовать срез масива NumPy для ее извлечения. В результате проведения подобных манипуляций над изображениями получаем трехмерную трансформируемую модель лица человека.

Сначала реконструируем грубый слой одного изображения, а затем используем 3DMM, параметры положения и глубину грубого слоя для реконструкции сети тонких слоев.

В предлагаемом алгоритме используются 69 основных компонентов для выражения и 85 основных компонентов

для определения идентичности и альбето. Число было выбрано таким образом, чтобы оно отражало 99% энергии. Мощность представления достаточна, и они хорошо обобщаются на тестовом наборе. В результате выполнения данной операции получаем сходства или различия.

Поскольку модель не передает полной достоверности модели на основе исходного изображения, продолжаем кодировать детали со смещением в направлении глубины. В результате получаем более точную картинку с фотoreалистичным результатом.

```

ap = argparse.ArgumentParser()
ap.add_argument("-p", "--shape-predictor", required=True,
    help="path to facial landmark predictor")
ap.add_argument("-i", "--image", required=True,
    help="path to input image")
args = vars(ap.parse_args())

detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(args["shape_predictor"])

image = cv2.imread(args["image"])
image = imutils.resize(image, width=500)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

rects = detector(gray, 1)

```

**Рис. 1.** Фрагмент программного кода обнаружения лица с помощью библиотеки dlib

**Fig. 1.** A fragment of the face detection program code using the dlib library

Интерфейс программного комплекса, представленного на рис. 2, будет состоять из главной веб-страницы и подключаемых к ней модулей.

Для загрузки и дальнейшего преобразования изображения в 3D-объект пользователю доступна кнопка «Загрузить» (рис. 3).

Далее после загрузки изображения программа начинает процедуру преобразования изображения в 3D-объект с помощью алгоритмов, которые были описаны ранее, показав при этом уведомление о процессе преобразования. Данное уведомление представлено на рис. 4.

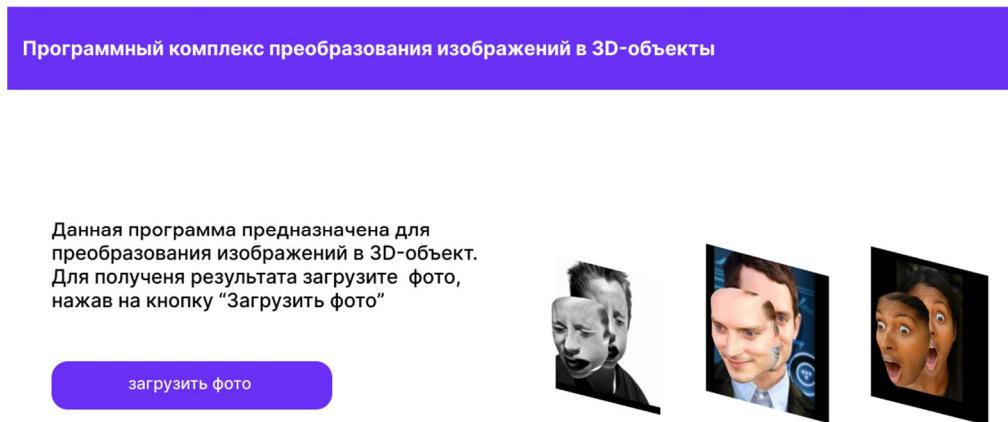


Рис. 2. Графический интерфейс пользователя

Fig. 2. Graphical User Interface

Данная программа предназначена для преобразования изображений в 3D-объект. Для получения результата загрузите фото, нажав на кнопку "Загрузить фото"

загрузить фото

Рис. 3. Приветственное сообщение и кнопка загрузки фотографии

Fig. 3. Welcome message and photo upload button

Программный комплекс преобразования изображений в 3D-объекты

Идет процесс преобразования...

Рис. 4. Экран с уведомлением о процессе преобразования

Fig. 4. Screen with notification of the conversion process

После обработки на главный экран выводится окно с 3D-моделью. Резуль-

тат выполнения данной процедуры будет показан на рис. 5.

Программный комплекс преобразования изображений в 3D-объекты

Результат

**Рис. 5.** Экран результата выполнения преобразования**Fig. 5.** Screen of the conversion result

Сравнительный анализ с нейросетевыми подходами NeRF и GAN-based реконструкцией выявил ключевые преимущества предложенного метода. В отличие от NeRF, требующего десятков входных изображений и минутного времени обработки, предложенный алгоритм демонстрирует сопоставимое качество при работе с единственным изображением и времени выполнения менее 1 секунды. GAN-методы, хотя и быстрее NeRF, уступают в точности воспроизведения геометрии лица (IoU 0.83 против 0.78 у StyleGAN3D) и склонны к артефактам на нестандартных чертах.

Несмотря на высокую эффективность предложенного решения, алгоритм имеет ряд ограничений, связанных с качеством входных данных и особенностями обработки, в частности алгоритм зависит от качества входного изображения. Низкое разрешение или размытость изображения могут привести к потере деталей в 3D-модели, а сильное сжатие (например, JPEG) ухудшает точность реконструкции текстуры. Неравномерное освещение (на-

пример, резкие тени или засвеченные области) затрудняет извлечение альбено и нормалей. Слишком темные или переэкспонированные изображения могут вызвать ошибки в оценке глубины.

Алгоритм демонстрирует наилучшие результаты для фронтальных изображений. При отклонении угла поворота головы более чем на 30° точность снижается. Сильно наклоненные головы или закрытые лицо объекты (например, руки, волосы) приводят к некорректной реконструкции.

Лица с аномалиями (например, шрамы, врожденные дефекты) или нестандартными пропорциями могут обрабатываться с ошибками из-за ограничений обучающей выборки 3DMM.

Разработанный алгоритм демонстрирует высокую эффективность для стандартных условий, но требует доработки для обработки неидеальных входных данных. Дальнейшие исследования будут направлены на улучшение устойчивости к освещению и расширение диапазона поддерживаемых ракурсов.

## Оценка эффективности предложенного алгоритма

Для оценки эффективности предложенного алгоритма были проведены тесты на различных конфигурациях оборудования, включая высокопроизводительные GPU: NVIDIA RTX 3090, RTX 3060. Основными метриками стали время обработки одного изображения, потребление видеопамяти и точность реконструкции, измеряемая через Intersection over Union (IoU) относительно эталонных данных из набора FaceScape. Наиболее быстрая обработка (0.8 секунды) достигнута на RTX 3090 при потреблении 6.2 ГБ VRAM. В то же время, CPU-реализация

алгоритма продемонстрировала значительно более низкую эффективность - среднее время обработки 12.3 секунды, что указывает на необходимость дальнейшей оптимизации для процессорных архитектур. Полученные результаты свидетельствуют о том, что предложенный алгоритм демонстрирует удовлетворительную производительность на современных графических ускорителях, сохраняя при этом разумные требования к объему видеопамяти.

## Сравнение с аналогами

В табл. 1 представлен сравнительный анализ с известными моделями.

**Таблица 1.** Сравнительная таблица методов 3D-реконструкции

**Table 1.** Comparative table of 3D reconstruction methods

Модель / Технология / Model / Technology	Параметры ввода / Input parameters	Применяемые методы / Applied methods	Примечания / Notes
Предлагаемое решение	Одно изображение	Обучаемый обратный рендеринг, каскадные CNN	Минимальные затраты времени и ресурсов
Multi-view 3D Reconstruction	Множественные изображения	Стереовидение, LBM, многопроекционные методы	Зависит от количества съемок и условий освещения
3D Face Reconstruction from 4 Views	4 изображения	Обработка изображений и геометрия	Необходимость в точном позиционировании кадров
FaceNet	Несколько изображений	Глубокое обучение, CNN	Высокая точность, но требует много различных изображений
Photometric Stereo Techniques	Несколько изображений	Статистические методы, рендеринг	Ограниченные возможности при изменении освещения

Методы Multi-view 3D Reconstruction и 3D Face Reconstruction from 4 Views хотя и способны обеспечить более высокое качество результатов, но требуют значительных вычислительных ресурсов и тщательной настройки процессов захвата изображения. Они зависят от точного позиционирования камер и условий освещения, что может затруднить их применение в реальных сценариях.

FaceNet подходит для распознавания лиц, однако для генерации 3D-моделей она требует множества изображений под различными углами. Этот аспект значительно усложняет процесс и увеличивает его затраты, что может быть непрактично для некоторых приложений, особенно в условиях ограниченного времени и ресурсов.

Photometric Stereo Techniques подход обеспечивает высокое качество генерации 3D-моделей, но зависит от наличия нескольких источников света и удобных углов съемки. Эти требования могут ограничить его использование в динамичных или сложных ситуациях, где условия освещения могут варьироваться.

Таким образом, предлагаемое решение представляет собой быстрый и эффективный инструмент для создания высокодетализированных 3D-моделей лиц на основе всего лишь одного входного изображения. Это обеспечивает значительную мобильность и легкость использования, что делает решение доступным для широкого круга пользователей, включая тех, кто не имеет специализированных знаний в области 3D-моделирования.

## Вывод

На сегодняшний день проблема построения фотoreалистичной 3D модели лица до конца не решена. Востребованность исследований в данной области обусловливается высокими темпами развития и повсеместного внедрения цифровых технологий.

В данной работе были рассмотрены основные методы построения фотoreалистичных 3D моделей по входному изображению, выведены основные этапы работы интеллектуального программного комплекса преобразования изображений в 3D объект. Были проведены исследования каждого из этапов и проанализированы различные методы, которые на текущий момент времени применяются.

Для проведения практических исследований был спроектирован и разработан интеллектуальный программный комплекс преобразования изображений в 3D объект. Данное приложение реализовано на языке программирования python с помощью современного фреймворка DRF (Django Rest Framework). Разработанная программа позволяет достичь точность более 80%, результаты были получены на основе тестовых бланков.

Предложенная структура программы может быть внедрена в работу, не нарушая работу общей системы, т. к. оно представляет собой отдельную веб-платформу, пригодную для дальнейшего расширения.

### Список литературы

1. Lee H., Ranganath R., Ng A.Y. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. Grosse // Proceedings of the 26th Annual International Conference on Machine Learning. 2009. P. 34–45.
2. Bengio Y. Learning deep architectures for AI // Foundations and Trends in Machine Learning, 2009. P. 245–257.
3. Ignatenko A., Konushin A. A Framework for Depth Image-Based Modeling and Rendering // Graphicon-2003 Proceedings. Moscow, 2003. 246 p.
4. Василенко В. А. Сплайн-функции: теория, алгоритмы, программы. Новосибирск: Наука, 1983. 215 с.
5. Соловьева А. Алгоритм модификации типового трехмерного портрета по заданным фотоизображениям // Труды XX международной конференции по компьютерной графике и машинному зрению. М.: изд-во «Графикон», 2010. С. 346–347.
6. Zhao H., Osher S. Visualization, analysis and shape reconstruction of unorganized data sets // Geometric Level Set Methods in Imaging and Vision and Graphics. Springer-Verlag, 2002. 256 p.
7. Зоткина А.А. Анализ алгоритмов машинного обучения, используемых в классификации изображений, публикуемых пользователями социальных сетей // Современные информационные технологии. 2023. № 38 (38). С. 38-40.
8. Зоткина А.А. Распознавание изображений с помощью сверточных нейронных сетей // Современные информационные технологии. 2023. № 38 (38). С. 60-63.
9. Зоткина А.А., Мартышкин А.И., Ткаченко А.В. Особенности работы сверточных нейронных сетей: архитектура и применение // Современные методы и средства обработки пространственно-временных сигналов: сборник статей XX Всероссийской научно-технической конференции. Пенза, 2023. С. 32-35.
10. Blanz V., Vetter T. Face recognition based on fitting a 3-D morphable model // IEEE Trans Pattern Anal Mach Intell. 2003. 25(9). P. 1063-1074.
11. Learning detailed face reconstruction from a single image // IEEE conference on computer vision and pattern recognition (CVPR) / E. Richardson, M. Sela, R. OR-EL, R. Kimmel. Honolulu, HI, 2017. P. 5553-5562.
12. Blanz V., Vetter T.A. Morphable model for the synthesis of 3-D faces // 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99). ACM Press/AddisonWesley Publishing Co., USA, 1999. P. 5553-5562.
13. Heo J. Three-dimensional generic elastic models for two-dimensional pose synthesis and face recognition. Proquest, Umi Dissertation Publishing, 2011. 154 c.
14. Korikov A.M., Tungusova A.V. Neural network technologies for image classification // 21st Int. Symp. Atmos. Ocean Opt. Atmos. Phys. 2015. Vol. 9680. P. 426–429.
15. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks // Adv. Neural Inf. Process. Syst. 2012. Vol. 25.

16. Face recognition: A literature survey / W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld // ACM Comput. Surv. 2003. Vol. 35, № 4. P. 399– 458.
17. Is faster R-CNN doing well for pedestrian detection? / L. Zhang, L. Lin, X. Liang, K. He // Eur. Conf. Comput. vision. Springer, Cham, 2016. P. 443–457.
18. Коэльо Л. П., Ричарт В. Построение систем машинного обучения на языке Python / пер. с англ. А. А. Слинкин. 2-е изд. М.: ДМК Пресс, 2016. 302 с. URL: <https://e.lanbook.com/book/82818>
19. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. М.: ДМК Пресс, 2015. 400 с. URL: <https://e.lanbook.com/book/69955>
20. PyTorch. URL: <https://pytorch.org/>
21. Numpy. URL: <https://numpy.org/>
22. How to install dlib library for Python in Windows 10. URL: <https://medium.com/analytics-vidhya/how-to-install-dlib-library-for-python-in-windows-10-57348ba1117f>
23. Opencv. URL: <https://opencv.org/>

## References

1. Lee H., Ranganath R., Ng A.Y. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. Grosse. *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009. P. 34-45.
2. Bengio Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*. 2009. P. 245-257.
3. Ignatenko A., Konushin A. Framework for Depth Image-Based Modeling and Rendering. *Graphicon-2003 Proceedings*. Moscow; 2003. 246 p.
4. Vasilenko B. A. Spline functions: theory, algorithms, programs. Novosibirsk: Nauka; 1983. 215 p. (In Russ.).
5. Solovyova A. Algorithm for modification of a typical three-dimensional portrait based on specified photographic images. In: *Trudy XX mezhdunarodnoi konferentsii po kompyuternoi grafike i mashinnomu zreniyu = Proceedings of the XX International Conference on Computer Graphics and Machine Vision*. Moscow; 2010. P. 346-347. (In Russ.).
6. Zhao H., Osher S. Visualization, analysis and shape reconstruction of unorganized data sets. In: *Geometric Level Set Methods in Imaging and Vision and Graphics*. Springer-Verlag, 2002. 256 p.
7. Zotkina A.A. Analysis of machine learning algorithms used in the classification of images published by users of social networks. *Sovremennye informatsionnye tekhnologii = Modern information technologies*. 2023; (38): 38-40. (In Russ.).
8. Zotkina A.A. Image recognition using convolutional neural networks. *Sovremennye informatsionnye tekhnologii= Modern information technologies*. 2023; (38): 60-63. (In Russ.).
9. Zotkina A.A., Martyshkin A.I., Tkachenko A.V. Features of convolutional neural networks: architecture and application. In: *Sovremennye metody i sredstva obrabotki prostranstvenno-*

*vremennoykh signalov. Sbornik statei XX Vserossiiskoi nauchno-tehnicheskoi konferentsii = Modern methods and means of processing spatiotemporal signals. Collection of articles of the XX All-Russian Scientific and Technical Conference.* Penza; 2023. P. 32-35. (In Russ.).

10. Blanz V., Vetter T. Face recognition is based on fitting a 3-D morphable model. *IEEE Trans Pattern Anal Mach Intell.* 2003; 25(9):1063-1074.
11. Richardson E., Sela M., OR-EL R., Kimmel R. Learning detailed face reconstruction from a single image. In: *IEEE conference on computer vision and pattern recognition (CVPR)*. Honolulu, HI; 2017. P. 5553-5562.
12. Blanz V., Vetter T.A. Morphable model for the synthesis of 3-D faces. In: *26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99)*. ACM Press/AddisonWesley Publishing Co., USA; 1999. P. 5553-5562.
13. Heo J. Three-dimensional generic elastic models for two-dimensional pose synthesis and face recognition. Proquest, Umi Dissertation Publishing; 2011. 154 p.
14. Korikov A.M., Tungusova A.V. Neural network technologies for image classification. *21st Int. Symp. Atmos. Ocean Opt. Atmos. Phys.* 2015; 9680: 426–429.
15. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 2012; 25.
16. Zhao W., Chellappa R., Phillips P.J., Rosenfeld A. Face recognition: A literature survey. *ACM Comput. Surv.* 2003; 35(4): 399 – 458.
17. Zhang L., Lin L., Liang X., He K. Is faster R-CNN doing well for pedestrian detection? *Eur. Conf. Comput. vision.* Springer, Cham; 2016. P. 443–457.
18. Coelho L. P., Richart V. Building machine learning systems in Python. Moscow: DMK Press; 2016. 302 p. (In Russ.). Available at: <https://e.lanbook.com/book/82818>
19. Flach P. Machine learning. The science and art of building algorithms that extract knowledge from data. Moscow: DMK Press; 2015. 400 p. (In Russ.). Available at: <https://e.lanbook.com/book/69955>
20. PyTorch. Available at: <https://pytorch.org/>
21. Numpy. Available at: <https://numpy.org/>
22. How to install dlib library for Python in Windows 10. Available at: <https://medium.com/analytics-vidhya/how-to-install-dlib-library-for-python-in-windows-10-57348ba1117f>
23. Opencv. Available at: <https://opencv.org/>

---

### Информация об авторе / Information about the Author

**Зоткина Алена Александровна**, старший преподаватель кафедры «Программирование», Пензенский государственный технологический университет, г. Пенза, Российская Федерация, e-mail: alena.zotkina.97@mail.ru, ORCID: <http://orcid.org/0000-0002-2497-6433>

**Alena A. Zotkina**, Senior Lecturer of the Programming Department, Penza State Technological University, Penza, Russian Federation, e-mail: alena.zotkina.97@mail.ru, ORCID: <http://orcid.org/0000-0002-2497-6433>