

Санкт-Петербургский государственный университет
Математико-механический факультет
Кафедра системного программирования

**Задача верификации лица на основе 3D
модели**

Курсовая работа студента 445 группы

Петрова Николая Сергеевича

Научный руководитель

К. ф.-м. н. Вахитов А.Т.

Санкт-Петербург
2012 г.

Оглавление

Введение.....	3
1. Постановка задачи.....	5
2. Обзор существующих решений.....	6
3. Предложенный метод.....	9
3.1. POSIT.....	9
3.2. Применение его к задаче.....	16
4. Примеры экспериментов.....	18
Заключение.....	19
Список литературы.....	20

Введение.

Биометрические системы для распознавания человека пользуются постоянным спросом. Среди всех биометрических технологий, которые используются сейчас, распознавание лиц является одним из наиболее широко распространённых видов биометрии. Его ежедневно используют практически все, в качестве одного из основных способов распознавания людей, и его естественность превратила распознавание лиц в общепринятый метод.

Тем не менее, несмотря на то, что различные системы распознавания лиц уже существуют, многие из них могут быть улучшены. Системы распознавания лиц по двумерным изображениям имеют ограничения по применению, связанные с физическими изменениями внешнего вида, старением, разнообразием возможных позиций головы на снимке и изменениями в интенсивности освещения. В последнее время для решения некоторых из этих проблем были введены системы 3D-распознавания лиц, новые биометрические техники, показавшие высокий уровень точности и надежности, так как они более устойчивы к изменениям лица в связи с перечисленными факторами.

Биометрические системы, основанные на распознавании лиц, состоят из устройств получения данных, предварительной обработки, извлечения неких особенностей, хранения данных и их сравнения. Устройством получения данных может являться 2D или 3D камера. Предварительная обработка может заключаться в обнаружении особых точек лица (landmarks), выравнивании изображения и вырезании из него лицевой области. При обработке можно фильтровать лишнюю информацию, такую как волосы, фон и снижать изменения в лице в связи с изменением положения головы. В 2D-изображениях, особые точки лица, такие как глаза, брови, рот и другие, могут быть обнаружены с высокой надёжностью (например, если выделять их вручную). В 3D-распознавании лиц нос является наиболее важной особой точкой.

3D-информация (карта глубины и текстурная информация), соответствующая поверхности лица может быть получена с помощью различных средств: стереоскопической камерой, просто набором снимков объекта с различных точек пространства или 3D сканером. Задача распознавания людей по трехмерным моделям может ставиться и решаться по-разному. Первая группа собирает в себе все 3D-подходы, которые требуют один и тот же формат данных, как на стадии обучения, так и на стадии тестирования. Вторая группа объединяет все подходы, которые используют 3D-данные во время обучения системы, но на

стадии распознавания используют только 2D-данные. Подходы из первой категории дают лучшие результаты, чем из второй, однако главным их недостатком являются высокие требования, предъявляемые к входным данным. Они не подходят для приложений наружного наблюдения или точек доступа на какое-либо предприятие, где возможно получение только одного обычного 2D-изображения, к тому же зачастую - еще и с произвольным ракурсом съемки. Вторая категория охватывает подходы на основе сопоставления трехмерной модели и двумерного изображения. Подходы к распознаванию лиц, основанные на моделях, имеют основным недостатком высокую вычислительную нагрузку, необходимую для нахождения соответствия 2D-изображений и 3D-моделей. Такие подходы называются асимметричным или гетерогенным анализом лиц, находящие соответствия лиц между различными типами данных, которые могут быть связаны путем применения некоторых методов (канонический корреляционный анализ или парная спектральная регрессия [2]). Таким образом, можно получить соответствие между 2D и 3D данными лица.

Несколько задач в литературе рассматривают так называемые асимметричные 3D-2D задачи распознавания. Например, Рама и соавт. предложил частичный анализ главных Компонент (Partial Principle Component Analysis - P2CA) для извлечения особенностей и снижение размерности в обоих случаях: цилиндрическое представление текстуры (3D) в галерее и 2D изображения в тестируемом наборе. Совсем недавно Ян и соавт. предложил канонический анализ корреляции ядра, основанного на патчах (patch based Kernel Canonical Correlation Analysis) для обучения отображения между изображениями 3D и текстурой лица. Все задачи частично полагаются на 2D данные, однако, не предоставляют надежное выполнение при изменении освещения или положения.

Задачей моей курсовой является описать алгоритм, с помощью которого возможно верифицировать лицо человека на фотографии с 3D моделью некоего человека. Если это его модель, то надо сказать, что вероятно один человек на фотографии и на 3D модели, иначе сказать, что это не верно.

1. Постановка задачи

Пусть дан набор двумерных точек и набор соответствующих им трехмерных точек. Задача, которая решается в курсовой работе – разработка метода автоматизированного сопоставления двумерных изображений и трехмерных моделей путём сопоставления вышеназванных наборов точек. Надо найти наилучшие значения параметров проектирования, таких как матрицы поворота и вектора сдвига, для максимального совпадения проектированных трехмерных точек и двумерных точек.

Рассмотрим классическую модель камеры-обскуры с центром проектирования O . Её плоскость изображения G находится на расстоянии f от O (фокусное расстояние). Считается, что фокусное расстояние известно. Оси Ox и Oy направлены вдоль строк и столбцов сенсора камеры, а третья ось Oz направлена вдоль оптической оси камеры. Единичные вектора этих осей назовём i, j, k . Объект с отмеченными точками M_0, \dots, M_n расположен в поле зрения камеры. Относительное расположение точек предполагается известным. Назовём точку M_0 относительной точкой объекта. В относительных координатах (относительно M_0) мы знаем координаты точек M_i ($i > 0$). Также мы знаем координаты точек на изображении m_i (x_i, y_i).

Необходимо найти координаты точек M_i в координатах камеры - (X_i, Y_i, Z_i) . В процессе их нахождения мы получим матрицу поворота камеры и вектор сдвига, что даст нам положение камеры.

2. Обзор существующих решений

2.1. UR3D

Для верификации человека по 3D модели существует автоматический метод, разработанный Kakadiaris I. A. и соавторами, названный UR3D [3, 4], проверенный на базе данных FRGC 2.0, которой пользовался и я. Этот метод основан на описании лицевых данных с помощью деформированной лицевой модели. Эта модель захватывает индивидуальные детали лица и представляет эту 3D информацию в некой форме 2D, так называемой UV-параметризации. С помощью этой параметризации возможно дальнейшее сравнение данных (3D-3D, 3D-2D) с использованием вейвлетного анализа и спектральных коэффициентов. Процедура верификации в рассматриваемом подходе состоит из 2 шагов: регистрация и идентификация.

Как происходит регистрация? Сначала грубые данные, которые приходят с сенсора, преобразуются в 3D полигональное представление. То есть сначала данные фильтруются с целью убирания «вылетевших» 3D точек. Затем происходит заполнение дыр (в 3D модели со сканера существуют пропуски в районе глаз и бровей). Полученная модель сглаживается, чтобы убрать белый шум.

Затем данные выравниваются, то есть вычисляется положение модели и производится сдвиг/поворот к известному, общему положению. Это является ключевым шагом в любом геометрическом подходе. После чего предложенная модель лица сопоставляется входным 3D данным. Этот шаг основан на подходе, основанном на деформируемой модели, предложенном Metaxas и Kakadiaris. Затем происходит вычисления карты нормалей; вейвлетный анализ применяется для получения суженного набора коэффициентов как метаданных.

Во время идентификации происходит вычисление неких метрик для параметров из метаданных. В случае, если входными данными является 2D изображение, то сначала производится поворот проверяемой модели по 7 ключевым точкам (4 точки в уголках глаз, кончик носа и 2 точки в углах носа). Поворот производится с помощью SDK PittPatt (Pittsburgh Pattern Recognition), к которой нет открытого доступа. Затем 3D модель проектируется на входное изображение по найденному повороту и извлекается соответствующая текстура с изображения. Затем к полученной текстуре применяется некий алгоритм пересчета освещённости (Bidirectional Relighting), чтобы новое освещение совпадало с тем, которое присутствует на 3D модели, и далее считаются метрики (CWSSIM

или CWSPT). Если полученные значения больше, чем некий порог, то говорится, что это один человек.

2.2. LBP

Существует метод описания 3D модели с помощью Локальных Бинарных Паттернов (Zhou и соавт. [6]). Этот метод описывает локальный текстурный паттерн с помощью бинарного кода. Он построен на пороговой классификации некоторой окрестности точки некоторого радиуса (обычно рассматривают 8 соседних пикселей). То есть если значения яркости (в ч/б изображении) соседнего пикселя больше, чем в рассматриваемом пикселе, то он кодируется единицей, иначе нулём. Так каждому пикселю текстуры соответствует набор из 8 бит (количество соседей). Потом для каждой области строится гистограмма подобных наборов и таким образом описывается лицо. В случае 3D этот алгоритм немного изменяется, чтобы описывать внутреннюю геометрическую информацию, снижая эффект от изменения выражения лица эффективно. На стадии обучения спектральная регрессия применяется для обучения главных компонент (principle components) для каждого 3D-изображения. Изначально этот метод применялся для 3D-3D распознавания, но существует применения ЛБП и для 2D-3D распознавания.

2.3. Automatic Asymmetric 3D-2D Face Recognition

Этот метод представляет собой асимметричный метод 3D-2D распознавания лиц, имеющий целью ограничить использование 3D-данных только там, где это действительно помогает повысить производительность. Этот подход использует текстурированные 3D модели лица для регистрации, тогда как только 2D изображение лица для идентификации, что делает его уникальным по сравнению с современными методами [2].

Поскольку каждой 3D-модели лица соответствует облако точек и 2D-изображение, этот подход состоит из двух отдельных действий: 2D-2D на основе разреженного классификатора представления (Sparse Representation Classifier); 3D-2D на каноническом корреляционном анализе (Canonical Correlation Analysis - CCA). Обе соответствующие оценки объединяются для окончательного решения. Надёжность значительно улучшена за счет нового конвейера предварительной обработки, использующее логарифмическое общее изменение (Logarithmic Total variation) для уменьшения влияния освещенности и модель активного представления (Active Appearance Model – AAM [5]) для нормализации положения. Экспериментальные результаты на FRGC v2.0 базе данных являются перспективными, доказывая, что по сравнению с традиционными методами 2D

распознавания, предложенный асимметричный метод распознавания лица обеспечивает лучшую производительность, и по сравнению с основанными на 3D моделях методами, имеет более низкую стоимость сбора данных и вычислений.

На этапе обучения необходимы текстурированные 3D модели лица, каждая из которых содержит одно плотно зарегистрированное 2D-изображение и 3D облако точек. Для каждой модели лица, есть 64 вручную выделенных особых точек. На 2D фазе, ААМ построена с помощью 2D-изображения лица с выровненным освещением, а на 3D этапе все облака точек сначала зарегистрированы, а затем получены изображения лица.

Все текстуры и 3D лица преобразуются к средней форме лица ААМ. Локальные бинарные паттерны (Local Binary Patterns) применяются для повышения локальной структуры текстуры и 3D изображения, а затем два вида результирующих LBP лиц используются для подготовки подпространства ССА для обучения отображения между 2D и 3D данными лица. Кроме того, четыре типа PCA (Principle Component Analysis) подпространств получаются из набора изображений 2D лиц, 3D изображения лица, 2D изображений LBP лица, и 3D LBP изображений лица соответственно.

На этапе тестирования, текстурированные 3D модели лиц используются как галерея образцов. После предварительной обработки, такой же, как и на этапе обучения, процесс наложения ААМ обрабатывает 2D-изображения, чтобы найти лицо. На основе позиций локализованных ключевых точек, оба соответствующие 2D и 3D изображения лица преобразуются к средней форме лица. Все операции происходят в автономном режиме. Тестовые образцы являются 2D-изображениями лица, следовательно, он-лайн часть включает в себя нормализацию освещения, установку ААМ, 2D-2D и 3D-2D соответствия, а также оценка синтеза.

3. Предложенный метод

3.1. POSIT

Здесь я опишу метод нахождения положения объекта относительно одиночной камеры. Предполагаем, что мы можем отметить и найти соответствие для хотя бы 4 точек объекта, не лежащих в одной плоскости. Рассматриваемый метод комбинирует 2 метода: POS (Pose from Orthography and Scaling), аппроксимирующий перспективную проекцию с помощью масштабированной ортогональной проекции и находящий матрицу поворота и вектор сдвига объекта путём решения линейной системы и POSIT (POS with Iterations), который использует в цикле аппроксимацию нахождения положения POS для нахождения более хорошей масштабированной ортогональной проекции особых точек, а затем применяет POS к этим точкам, а не к исходным. POSIT сходится к точному решению за несколько итераций.

POSIT может быть применён ко многим точкам, а не только к 4, для устойчивости к ошибкам измерения или шуму изображения. Данный метод зарекомендовал себя высокой производительностью, и он является применимым для рассматриваемой задачи.

Задача состоит в вычислении матрицы поворота и вектора сдвига объекта. Матрица поворота R для объекта является матрицей, чьи строки являются координатами единичных векторов i, j, k координатной системы камеры, выраженных в объектной координатной системе (M_0u, M_0v, M_0w) . Цель матрицы поворота состоит в переводе координат из объектных в координаты камеры. Скалярное произведение M_0M_i на i даёт проекцию этого вектора на вектор i , то есть разницу между X -координатами точек M_i и M_0 в координатах камеры.

Для вычисления поворота достаточно найти координаты векторов i и j в координатах объекта. Вектор k далее может быть получен как векторное произведение $i \times j$. Вектор сдвига, T , это вектор OM_0 между центром проектирования (центр камеры) и относительной точкой объекта, точкой, являющейся началом координат в системе объекта. Получаем, что $T = \{X_0, Y_0, Z_0\}$. Если эта относительная точка M_0 была выбрана так, что ей соответствует точка на картинке m_0 , то длина вектора сдвига выражается через длину вектора Om_0 и равна $Z_0/f * Om_0$. Таким образом, чтобы найти вектор сдвига достаточно найти лишь Z -координату точки M_0 . Получилось, что положение объекта полностью задаётся через i, j и Z_0 .

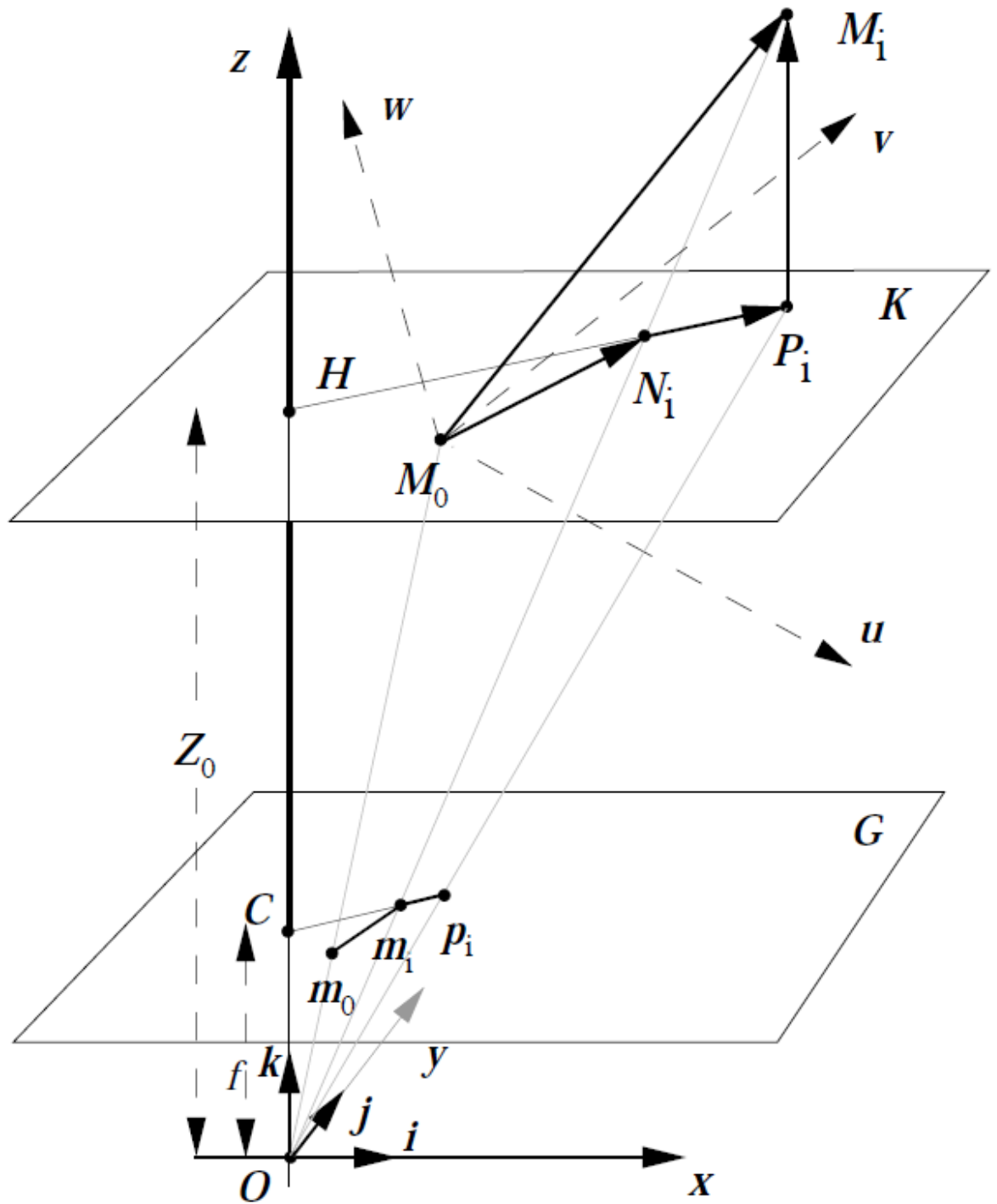


Рис. 1. Геометрия проектирования.

Масштабированная Ортографическая Проекция (МОП) является аппроксимацией к настоящей перспективной проекции. В этой аппроксимации для данного объекта, находящегося в зоне видимости камеры, предполагается, что Z -координаты всех точек объекта (M_i) в координатах камеры не сильно отличаются друг от друга. Таким образом, все

они могут считаться равными Z_0 (координате относительной точки объекта). В МОП проекция точки объекта M_i имеет координаты:

$$x'_i = fX_i/Z_0, \quad y'_i = fY_i/Z_0,$$

В то время как перспективная проекция той же точки имеет координаты

$$x_i = fX_i/Z_i, \quad y_i = fY_i/Z_i$$

Это отношение f/Z_0 называется масштабным множителем МОП. Для относительной точки объекта перспективная проекция совпадает с МОП. Также для точки M_i проекцию можно записать в виде

$$\begin{aligned} x'_i &= fX_0/Z_0 + f(X_i - X_0)/Z_0 = x_0 + s(X_i - X_0) \\ y'_i &= y_0 + s(Y_i - Y_0) \end{aligned} \tag{1}$$

Геометрическая конструкция МОП.

На рис. 1 можно увидеть построение МОП для точек объекта. Перспективная проекция m_i - это пересечение луча из центра проектирования, проходящего через 3D-точку M_i , и плоскости изображения. В МОП же мы рисуем плоскость K через точку M_0 , параллельно плоскости изображения G . Эта плоскость находится на расстоянии Z_0 от центра проектирования O . Точка M_i проектируется на K ортогонально, а затем полученная точка проектируется на G перспективно. Вектор m_0p_i параллелен M_0P_i и отличается в f/Z_0 раз. Это записано уравнением 1.

Фундаментальные уравнения.

Рассмотрим уравнения, которые описывают настоящую перспективную проекцию и отношение неизвестных векторов i и j матрицы поворота и неизвестной Z_0 координаты вектора сдвига к известным координатам векторов M_0M_i (в объектной системе координат) и известным координатам проекций x_i, y_i . Решения этих уравнений даст информацию, достаточную для восстановления положения объекта.

$$(2) \quad \mathbf{M}_0\mathbf{M}_i \cdot \frac{f}{Z_0} \mathbf{i} = x_i(1 + \varepsilon_i) - x_0,$$

$$(3) \quad \mathbf{M}_0\mathbf{M}_i \cdot \frac{f}{Z_0} \mathbf{j} = y_i(1 + \varepsilon_i) - y_0$$

Через ε_i обозначено

$$\varepsilon_i = \frac{1}{Z_0} \mathbf{M}_0\mathbf{M}_i \cdot \mathbf{k}$$

и \mathbf{k} находится как $\mathbf{i} \times \mathbf{j}$.

Доказательство: Посмотрим на рисунок 1. Рассмотрим точки M_0 , M_i объекта и плоскость K , параллельную G , проходящую через M_0 . Пусть луч из центра проектирования O , проходящий через точку M_i , пересекает плоскость K в точке N_i . А ортогональную проекцию точки M_i на плоскость K обозначим через P_i .

Тогда вектор M_0M_i выражается через сумму:

$$\mathbf{M}_0\mathbf{M}_i = \mathbf{M}_0\mathbf{N}_i + \mathbf{N}_i\mathbf{P}_i + \mathbf{P}_i\mathbf{M}_i$$

Вектор M_0N_i пропорционален вектору m_0m_i в отношении $Z_0:f$. Пара векторов N_iP_i и Cm_i также пропорциональны из подобия двух треугольников Cm_iO и $N_iP_iM_i$ в отношении равном отношению z -координат других соответствующих векторов в треугольниках, P_iM_i и OS . Это отношение равно $M_0M_i \cdot k/f$. Сумма трёх векторов может быть выражена как

$$\mathbf{M}_0\mathbf{M}_i = \frac{Z_0}{f} \mathbf{m}_0\mathbf{m}_i + \frac{\mathbf{M}_0\mathbf{M}_i \cdot \mathbf{k}}{f} \mathbf{C}\mathbf{m}_i + \mathbf{P}_i\mathbf{M}_i$$

Домножим скалярно это равенство на единичный вектор \mathbf{i} из координатной системы камеры. Заметим, что $\mathbf{P}_i\mathbf{M}_i \cdot \mathbf{i} = 0$; $\mathbf{m}_0\mathbf{m}_i \cdot \mathbf{i}$ - это x -координата вектора, то есть $x_i - x_0$; $\mathbf{C}\mathbf{m}_i \cdot \mathbf{i}$ это координата x_i вектора $\mathbf{C}\mathbf{m}_i$. С обозначением $\varepsilon_i = \mathbf{M}_0\mathbf{M}_i \cdot \mathbf{k}/Z_0$ получаем уравнение 2. Уравнение 3 получается аналогично домножением на \mathbf{j} . \square

На самом деле, в правых частях уравнений 2 и 3 термы $x_i(1 + \varepsilon_i)$ и $y_i(1 + \varepsilon_i)$ являются координатами x_i' и y_i' точки p_i , которая является МОП точки M_i .

Рассмотрим точки M_0 , M_i , проекции P_i на плоскость K , и её образ p_i . Назовём координаты точки p_i на изображении x_i' и y_i' . Вектор $M_0M_i = M_0P_i + P_iM_i$. Первый вектор, M_0P_i , пропорционален вектору m_0p_i в отношении $Z_0:f$. Следовательно, помножая это равенство скалярно на i или на j получим равенства

$$M_0M_i \cdot \frac{f}{Z_0} \mathbf{i} = x_i' - x_0$$

$$M_0M_i \cdot \frac{f}{Z_0} \mathbf{j} = y_i' - y_0,$$

Сравнивая полученные уравнения с равенствами 2 и 3 получаем, что

$$x_i' = x_i * (1 + \varepsilon_i), y_i' = y_i * (1 + \varepsilon_i)$$

POS и POSIT

Равенства 2 и 3 могут быть переписаны в виде

$$M_0M_i \cdot \mathbf{I} = x_i(1 + \varepsilon_i) - x_0,$$

$$M_0M_i \cdot \mathbf{J} = y_i(1 + \varepsilon_i) - y_0,$$

Уравнения 3, 4.

где $\mathbf{I} = i * f / Z_0$, а $\mathbf{J} = j * f / Z_0$.

Основная идея рассматриваемого метода заключается в том, что если известны значения ε_i , то уравнения выше позволяют найти координаты \mathbf{I} и \mathbf{J} , из которых затем могут быть получены вектора i и j (нормализацией), а T_z может быть получена из нормы \mathbf{I} или \mathbf{J} . Этот алгоритм называется POS (Pose from Orthography and Scaling). Конечно же, нахождение положения зависит от параметров ε_i . Решение алгоритмом POS является лишь приближением, если ε_i не являются точными значениями. Но как только мы посчитаем i и j для каких-то ε_i , более точные значения могут быть получены для ε_i , чтобы использовать их ещё раз. И затем можно повторить этот шаг уточнения несколько раз. Такой алгоритм называется POSIT (POS with Iterations). За несколько итераций этот алгоритм находит i , j , Z_0 .

Изначально возьмём $\varepsilon_i = 0$. Предполагая, что $\varepsilon_i = 0$, мы предполагаем, что $x_i = x_i'$, $y_i = y_i'$, то есть p_i совпадает с m_i . Если z-координаты точек объекта почти не отличаются, то приемлемый результат найдётся на одной из первых итераций.

На каждой итерации POSIT надо решать систему, составленную из уравнений вида 3, 4 для точек M_i ($i = 1..N$).

Перепишем их в более компактной форме:

$$M_0 M_i \cdot \mathbf{I} = \xi_i,$$

$$M_0 M_i \cdot \mathbf{J} = \eta_i,$$

где

$$\mathbf{I} = \frac{f}{Z_0} \mathbf{i}, \quad \mathbf{J} = \frac{f}{Z_0} \mathbf{j}, \quad \xi_i = x_i(1 + \varepsilon_i) - x_0, \quad \eta_i = y_i(1 + \varepsilon_i) - y_0,$$

и где ε_i известные параметры из предыдущего шага.

Получили линейные уравнения, где неизвестными являются вектора \mathbf{I} и \mathbf{J} , а известными параметрами являются x_i , y_i , x_0 , y_0 , как координаты точек m_i , m_0 на изображении. U_i , V_i , W_i - координаты точек в относительной системе координат (объектной). Они тоже известны.

Записав уравнение 3 для n точек объекта M_1, \dots, M_n и их образов на изображении, составим систему уравнений

$$A\mathbf{I} = \mathbf{x}', \quad A\mathbf{J} = \mathbf{y}'$$

где A - матрица координат точек объекта M_i в объектной системе координат, \mathbf{x}' - вектор, составленный из ξ_i , а \mathbf{y}' - вектор, составленный из η_i . В общем случае, если имеется хотя бы 3 видимые точки кроме M_0 и эти 4 точки не лежат в одной плоскости, то матрица A имеет ранг 3 и решение системы в смысле наименьших квадратов будет иметь вид

$$\mathbf{I} = B\mathbf{x}', \quad \mathbf{J} = B\mathbf{y}'$$

где B - псевдообратная матрица матрицы A . Назовём эту матрицу объектной. Зная взаимное расположение точек объекта можно предрассчитать объектную матрицу. Можно посчитать её разными способами, например, как $[A^T A]^{-1} A^T$ или с помощью декомпозиции матрицы A по сингулярным числам (Singular Value Decomposition). Такое решение

действительно дает наименьшее значение ошибки $|AI-x'|$ и $|AI-y'|$. Преимуществом второго метода является то, что при анализе этих сингулярных чисел можно выяснить являются ли выбранные точки лежущими в одной плоскости перед запуском самого алгоритма.

Как только мы нашли решения для I и J методом наименьших квадратов, единичные векторы i и j получаются из них простой нормализацией. Как говорилось ранее - первая строка матрицы поворота состоит из 3 координат вектора i , вторая строка состоит из 3 координат вектора j , а последняя строка состоит из 3 координат вектора k , полученного векторным произведением первых двух.

Теперь может быть получен вектор сдвига: это вектор OM_0 от центра проектирования до точки M_0 - относительной точки объекта. Этот вектор равен $Z_0 * Om_0/f$, то есть Om_0/s . Масштабный множитель s получен как норма вектора I или вектора J (или среднее между ними). Если параметр ϵ_i является точным, то матрица поворота и вектор сдвига задают точное представление положения объекта относительно камеры. Иначе можно уточнить значения ϵ_i и повторить итерацию.

Интуитивное представление POSIT алгоритма.

Изначально известны расположения 3D-точек на объекте и их перспективные проекции. Если мы можем построить МОП образы точек объекта из перспективной проекции, то мы можем применить POS алгоритм для этих МОП точек, чтобы получить точное расположение объекта. Вычисление точных МОП образов требует знание точного положения объекта. Однако, если применить POS к изначальным точкам на изображении, то получим приблизительную глубину для каждой точки объекта и сможем вычислить МОП образы. На следующем шаге применим POS к МОП образам, чтобы уточнить их. Повторять, пока не сойдемся к точному положению.

3.2. Применение POSIT к рассматриваемой задаче.

Пусть нам даны некая 3D модель лица человека. Она может быть получена, например, с помощью 3D сканера, как было сделано в базе данных FRGC 2.0. Для работы алгоритма необходимо вручную выделить особые, габитоскопические точки, такие как уголки глаз, уголки носа, уголки рта и края ушей. Эту операцию можно произвести заранее, до работы алгоритма и хранить эти точки вместе с моделями в базе данных.

На вход программы подаётся 2D изображение с лицом человека. На этом лице отмечаются те же габитоскопические точки, что и на моделях. Затем запускается RANSAC (RANdom SAmple Consensus), чтобы уменьшить влияние шума на выбранные точки. То есть случайным образом выбирается набор пар 2D-3D точек и запускается POSIT. Затем считается ошибка проектирования для остальных точек при полученной матрице поворота и векторе сдвига. Логично, выбирается тот результат, который подходит для максимального количества точек. Для дальнейшего уточнения решения применяется нелинейная оптимизация из библиотеки `nlopt` методом Ньютона для минимизации функции ошибки проектирования.

Применяя полученную матрицу поворота и вектор сдвига к 3D модели, мы можем наложить её на изображение лица. Качество этого наложения зависит от того с какими ошибками перепроектировались точки. Меньшим весом обладают точки, связанные со ртом и бровями, потому что при мимических изменениях эти точки не стоят на месте. Более надёжными являются точки на глазах, носу и ушах.

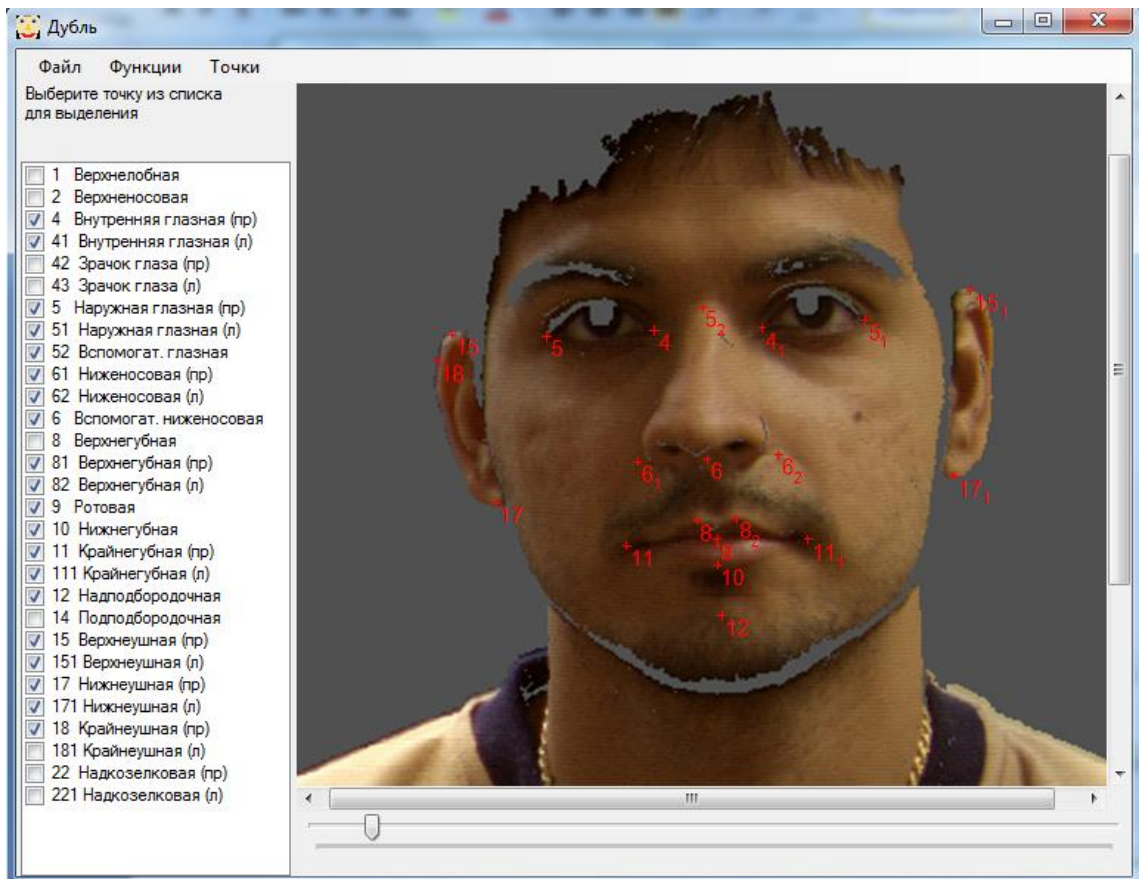


Рис. 2. Выделение точек на 3D.

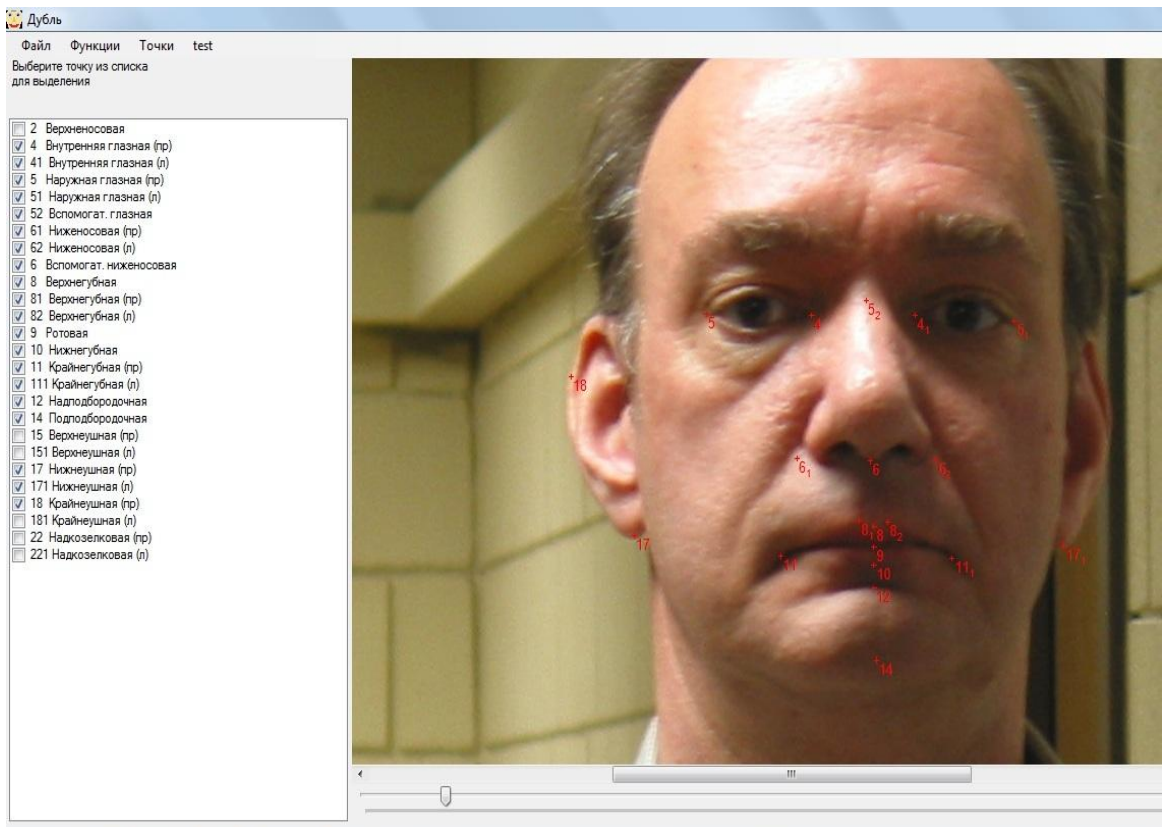


Рис. 3. Выделение точек на изображении.

4. Примеры экспериментов

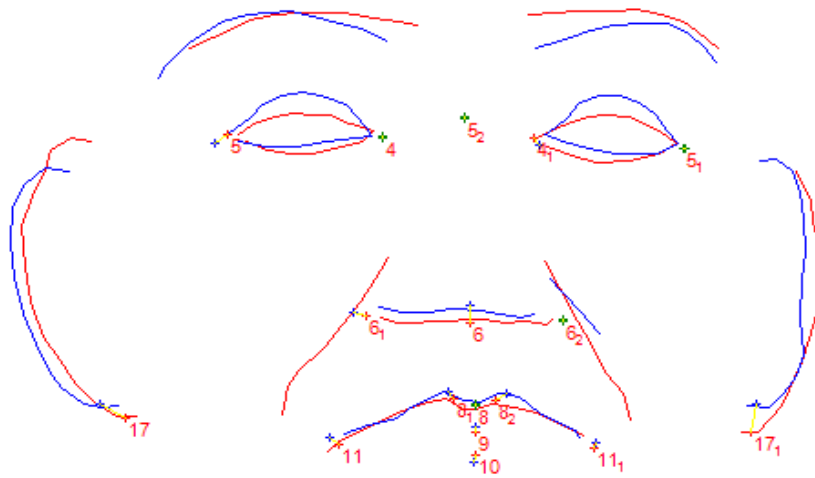


Рис. 4. 3D модель не соответствует человеку на изображении.

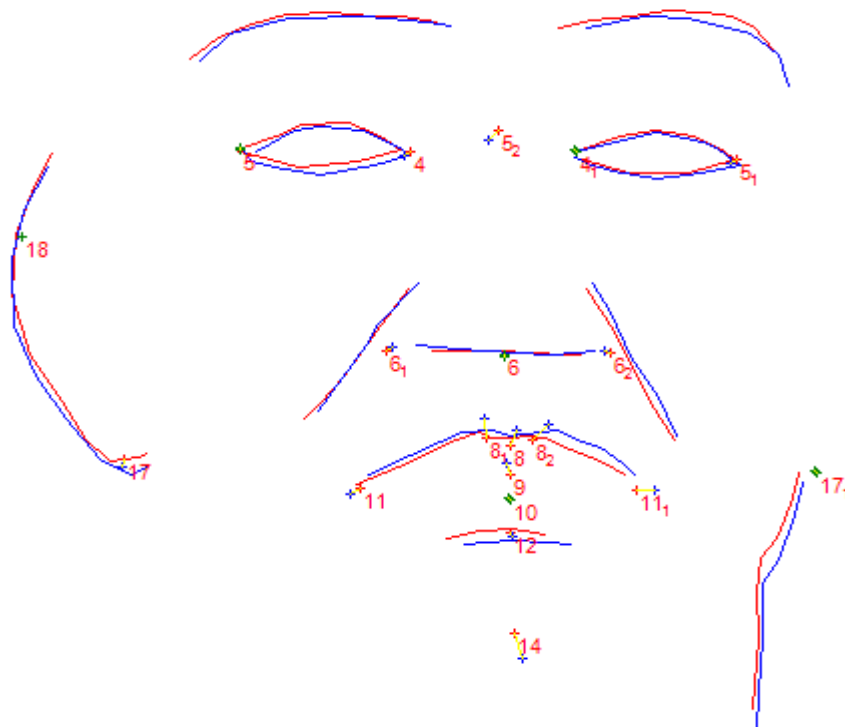


Рис. 5. 3D модель соответствует человеку на изображении.

Заключение

В ходе курсовой работы изучено множество подходов для верификации лица на изображении и 3D модели лица. У многих из этих подходов недостатком является зависимость от освещения и вычислительная стоимость он-лайн обработки.

В результате работы получена программа, на вход которой подаётся размеченная 3D модель лица и размеченная фотография. Затем эти точки сопоставляются максимально возможным способом, высчитывая положение объекта (лица) относительно камеры с помощью алгоритма POSIT, описанного выше. Такой подход не зависит от освещения, потому что точки выделяются вручную.

Список литературы:

- [1] Daniel F. DeMenthon and Larry S. Davis, “Model-Based Object Pose in 25 Lines of Code”, International Journal of Computer Vision, 1995
- [2] Di Huang, Mohsen Ardabilian, Yunhong Wang, Liming Chen, “Automatic Asymmetric 3D-2D Face Recognition”, pp. 1225 – 1228, 2010 20th International Conference on Pattern Recognition (ICPR), 2010
- [3] I.A. Kakadiaris, G. Passalis, G. Toderici, E. Efraty, P. Perakis, D. Chu, S. Shah, and T. Theoharis, “Face Recognition Using 3D Images” in Handbook of Face Recognition, S. Z. Li and A. K. Jain, Eds. Cambridge University Press, 2011, pp. 429-459.
- [4] G. Toderici, G. Passalis, S. Zafeiriou, G. Tzimiropoulos, M. Petrou, T. Theoharis, I.A. Kakadiaris, “Bidirectional relighting for 3D-aided 2D Face Recognition”, pp. 2721 – 2728, 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010
- [5] Daniel Pizarro, Julien Peyras, Adrien Bartoli, "Light-invariant fitting of active appearance models", pp.1-6, 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008
- [6] Zhou, X.; S`anchez, S. A. & Kuijper, A. , “3D Face Recognition with Local Binary Patterns”, Proceedings of Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2010