

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Камбиев Заур Мухамедович

Выпускная квалификационная работа бакалавра

Восстановление модели движения

камеры по видеопотоку

Направление 010400

Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Коровкин М. В.

Санкт-Петербург
2017

Содержание

Введение	3
Постановка задачи	5
Обзор литературы	6
Глава 1. Поиск и отслеживание плоскостей	7
1.1. Особые точки изображений	7
1.2. Построение матрицы гомографии	8
1.3. Поиск плоскостей	9
Глава 2. Изменение положения камеры	12
2.1. Связь гомографии со сдвигом и поворотом	12
2.2. Декомпозиция матрицы гомографии	13
2.3. Отсеивание лишних решений	16
2.4. Физическое расстояние	18
Глава 3. Построение траектории	19
3.1. Выбор кадров	19
3.2. Принцип ведущей плоскости	20
3.4. Общая структура алгоритма	21
Глава 4. Программная реализация	23
4.1. Выбор средств и тестирование	23
4.2. Анализ эффективности алгоритма	23
4.3. Способы улучшения алгоритма	24
Заключение	25
Список литературы	26
Приложение	27

Введение

Одной из важнейших задач теории управления является построение модели объекта управления. От этой модели зависит структура системы управления, ее параметры, выбор закона управления. Модель играет важную роль не только при создании системы управления, но и при исследовании закономерностей функционирования естественных и искусственных объектов и процессов.

Одним из самых эффективных методов построения модели объекта является идентификация. Идентификация, как построение математической модели на основе наблюдений, берет свое начало в работах Карла Фридриха Гаусса, который разработал метод наименьших квадратов, решая задачу предсказания траектории движения планет. Как раздел теории управления, идентификация появилась в начале 60-х годов XX века. Этот период характеризовался интенсивным развитием кибернетики и широким применением ее результатов при создании реальных систем управления. Новые подходы требовали повышенной точности описания объектов, остро встал вопрос о построении модели в случаях, когда априорная информация об объекте отсутствует или очень скудна. Отсутствие адекватных моделей сильно тормозило широкое использование новых теоретических результатов и новых технических средств. Идентификация помогла в значительной степени ликвидировать этот пробел.

В настоящее время эта область теории управления хорошо изучена и находит широкое применение на практике. Существует множество подходов к идентификации модели объекта. Часть из них существенным образом использует априорную информацию, другие нуждаются в значительно меньшем ее объеме, но в любом случае необходимо располагать набором данных, полученных в ходе нормальной работы изучаемого объекта либо при целенаправленном эксперименте.

Если объект управления можно отождествить с закрепленной на нем камерой, т. е. нас интересует только его расположение и ориентация в пространстве, то в роли источника наблюдений для задачи идентификации может выступать ее видеопоток. Поэтому для применения существующих методов идентификации модели движения необходимо уметь извлекать из видеопотока данные об изменении положения камеры в пространстве.

Задачу анализа видеопотока относят, как правило, к проблематике компьютерного зрения - междисциплинарной области, в рамках которой разрабатываются методы анализа цифровых изображений и видео. Одним из примеров применения теории компьютерного зрения является автоматизация решения задач, которые может решать зрительная система человека. В настоящее время методы компьютерного зрения активно используются в таких областях, как робототехника, топография, медицина, при создании систем видеонаблюдения и систем дополненной реальности.

Очевидно, что применимость различных подходов при анализе видеопотока сильно коррелирует с содержанием снимаемой сцены. В данном случае поворот и перемещение камеры между двумя кадрами можно вычислить с помощью декомпозиции матрицы гомографии - проективного отображения между кадрами, построенного для общей неподвижной плоскости.

Постановка задачи

Целью настоящей работы является разработка алгоритма извлечения из видеопотока камеры данных, необходимых для применения существующих методов идентификации модели движения. Под данными здесь понимается последовательность поворотов и перемещений камеры между кадрами, т. е. траектория ее движения.

Идея разрабатываемого алгоритма основана на декомпозиции матрицы гомографии между кадрами, поэтому достижение поставленной цели требует от алгоритма решения следующих задач:

1. Поиск и отслеживание плоскостей на видео.
2. Построение проективных отображений между кадрами.
3. Декомпозиция матриц гомографии.
4. Формирование траектории движения камеры.

В рамках данной работы предполагается, что нам известны внутренние параметры камеры, дисторсия на видео отсутствует или пренебрежимо мала.

Обзор литературы

В данной работе используются методы компьютерного зрения, реализованные в открытой библиотеке OpenCV. В книгах [5], [1] и [8] подробно рассмотрены возможности данной библиотеки, описаны основные методы, приведены примеры их использования. Наиболее полезными оказались разделы, посвященные поиску особых точек на изображении и нахождению соответствий между особыми точками двух изображений. Библиотека OpenCV постоянно развивается, подробное описание самых новых версий библиотеки можно найти в документации на официальном сайте [7].

В статье [4] приводятся результаты сравнительного анализа эффективности различных методов поиска соответствий между точками двух изображений. Описаны преимущества и недостатки SIFT, SURF и ORB подходов, выявленные при решении одинаковых задач.

В работе [3] изложены правила построения гомографии, приведен основанный на сингулярном разложении метод декомпозиции матрицы гомографии, используемый в настоящей работе. Получить более глубокое понимание принципов проективного отображения позволяет работа [6], в которой был разработан аналитический метод декомпозиции, связывающий компоненты матрицы гомографии с внутренними параметрами камеры.

Глава 1. Поиск и отслеживание плоскостей

1.1. Особые точки изображений

Особой точкой или точечной особенностью называют такую точку на изображении, окрестность которой отличается от окрестностей остальных точек изображения.

Анализ особых точек изображения обычно состоит из двух этапов: поиск точек и их описание. За эти этапы отвечают соответственно метод-детектор и метод-дескриптор. Поскольку особые точки важны при сравнении изображений, детектор должен обеспечивать инвариантность нахождения одних и тех же точек относительно преобразований изображения, а дескриптор должен обеспечивать инвариантность описания точек.

Относительно некоторых видов преобразования изображений инвариантны не все существующие подходы. К инвариантным относительно изменения масштаба относятся: Scale-invariant feature transform (SIFT), Speeded up robust features (SURF) и Oriented FAST and rotated BRIEF (ORB). SIFT является самым точным, но самым медленным из методов. SURF идеологически близок к SIFT, но серьезно сокращает временные затраты. ORB разрабатывался в качестве альтернативы этим методам и является самым быстрым. Его недостатком является неравномерное распределение найденных особых точек - обычно большинство их них оказывается в центре изображения. Более подробное сравнение этих подходов можно найти в [4].

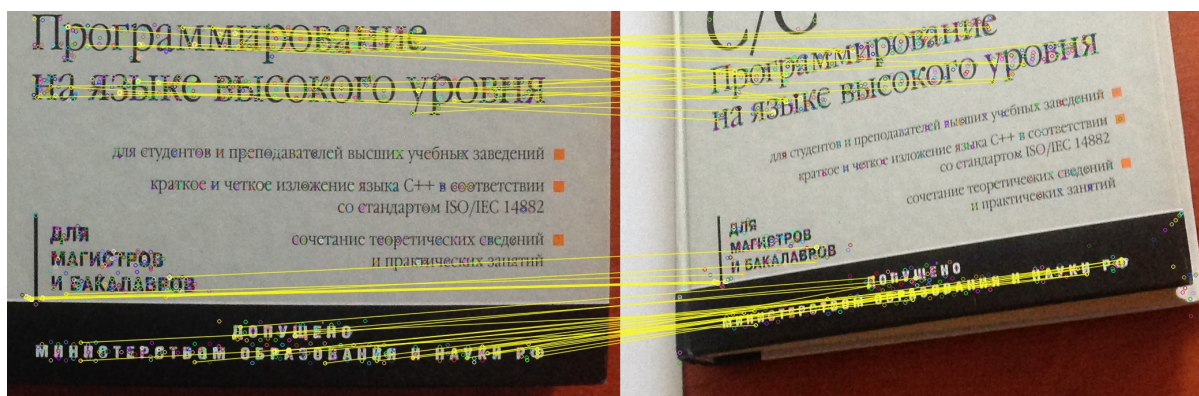


Рис. 1: Использование методов SURF и Brute-Force

Для поиска соответствий между особыми точками, найденными на разных изображениях, можно использовать полный перебор (или метод

«грубой силы», англ. *brute force*), а можно применить менее точный, но более быстрый метод ближайших соседей. Рисунок 1 иллюстрирует результат последовательной работы SURF метода для поиска особых точек и Brute-Force метода для их последующего сопоставления. Желтыми линиями соединены пары точек, окрестности которых оказались наиболее похожими.

1.2. Построение матрицы гомографии

Проекция $M(X, Y, Z) \Rightarrow m(x, y)$ точек трехмерного пространства в системе координат, связанной с камерой (рис. 2), на плоскость изображения определяется следующими соотношениями:

$$X/x = Y/y = Z \quad (1)$$

В перспективной геометрии координаты точек пространства задаются с точностью до общего множителя, потому что все точки, лежащие на прямой, соединяющей точки m и M , проецируются в одну и ту же точку изображения.

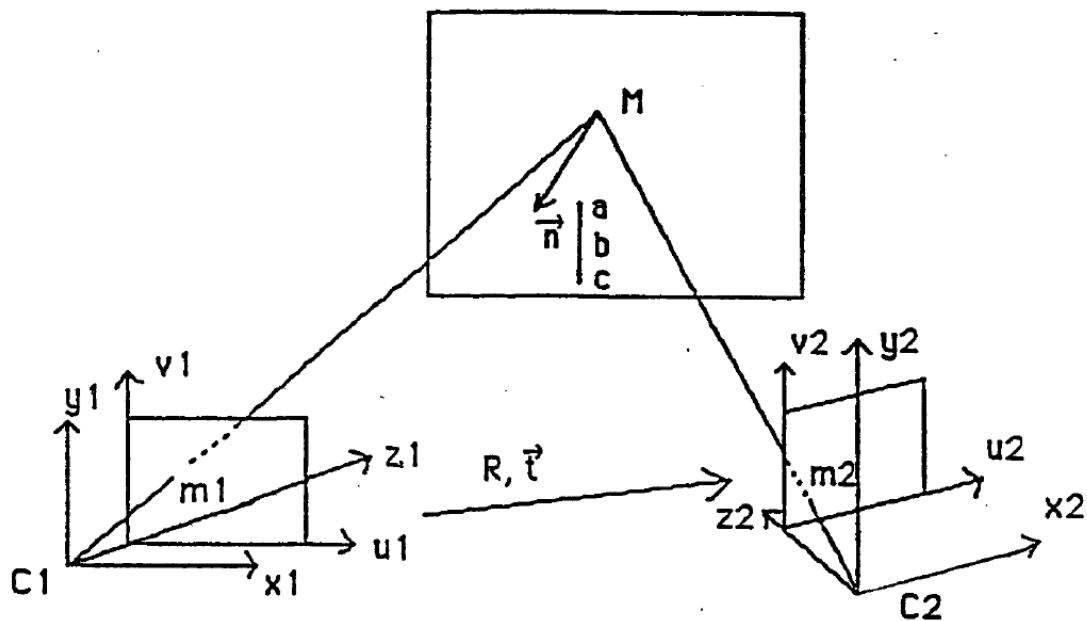


Рис. 2: Геометрическая иллюстрация

Знание внутренних параметров камеры позволяет произвести перевод 3D координат в 2D координаты изображения в пикселях. Процесс вычисления внутренних параметров камеры называется калибровкой. Этот процесс подробно описан в [2]. В библиотеке OpenCV реализовано несколько методов калибровки камеры, применение одного из них можно найти

в [5]. В настоящей работе внутренние параметры камеры предполагаются известными. Работа с внутренними параметрами камеры представлена в приложении.

Гомография - это проективное отображение между двумя изображениями плоскости, переводящее точки плоскости на одном изображении в соответствующие им точки на другом изображении.

$$m_2 = Am_1 = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} m_1 \quad (2)$$

В соответствии с принципами перспективной геометрии матрица гомографии вычисляется с точностью до масштабного множителя, поэтому для удобства мы можем ввести дополнительное ограничение, например, $a_{33} = 1$.

Поскольку гомография является отображением между двумя изображениями плоскости, одно точечное соответствие позволяет, используя соотношения (1) и (2), составить два уравнения относительно компонент матрицы:

$$\begin{cases} a_{11}x_1 + a_{12}y_1 + a_{13} - a_{31}x_2x_1 - a_{32}x_2y_1 - a_{33}x_2 = 0 \\ a_{21}x_1 + a_{22}y_1 + a_{23} - a_{31}x_2x_1 - a_{32}x_2y_1 - a_{33}x_2 = 0 \end{cases}$$

Здесь $m_1(x_1, y_1)$ и $m_2(x_2, y_2)$ - это две проекции точки $M_1(X_1, Y_1, Z_1)$ на плоскости изображений, соответствующих разным положениям камеры.

Таким образом, четыре точечных соответствия между изображениями плоскости позволяют составить систему из восьми линейных уравнений с восьмью неизвестными. Для разрешимости этой системы необходимо, чтобы три из четырех точек были не коллинеарны. Если найденных общих точек окажется больше, то более точный результат можно получить методом наименьших квадратов.

1.3. Поиск плоскостей

При анализе кадров видеопотока задачу поиска плоскостей удобно решать одновременно с задачей построения гомографии. Критерием того, что некоторое множество особых точек изображения принадлежит одной

плоскости, может выступать качество построения гомографии между двумя кадрами по нескольким точкам из этого множества. Если построенная гомография правильно отображает точки проверяемого множества, то гипотеза о существовании плоскости может быть признана правдоподобной по крайней мере на этапе анализа рассматриваемой пары кадров. При этом построение гомографии между другими кадрами по тому же набору особых точек может как опровергнуть эту гипотезу, так и снова подтвердить ее правдоподобность.

Общий подход к поиску плоскостей подразумевает отсутствие начальных гипотез. Поэтому имеет смысл в качестве гипотез брать случайные наборы из четырех особых точек, строить по ним гомографию и пытаться применить ее ко всем остальным особым точкам. Точки, которые корректно отображаются с помощью построенной гомографии, будем условно считать принадлежащими той же плоскости, что и испытываемая четверка точек. Чем больше точек совпало, тем более вероятно существование общей плоскости. Рассмотрев достаточно большое количество таких случайных гипотез о существовании плоскости, следует выбрать самый правдоподобный вариант.

Такой подход к оценке параметров модели на основе случайных выборок называется RANdom SAmple Consensus (RANSAC). Применив этот метод несколько раз, каждый раз выделяя полученное подмножество точек в отдельный кластер, мы получим несколько подмножеств, каждое из которых по сути определяет пока не опровергнутую гипотезу о существовании плоскости. Рисунок 3 иллюстрирует выделение кластеров по двум кадрам. Точки первого кластера имеют белый цвет, точки второго кластера - синий.



Рис. 3: Пример выделения двух кластеров

Поскольку мы имеем дело с последовательностью кадров, то переход к каждому следующему кадру может повлиять на правдоподобность старых гипотез или стать источником для новых. Чтобы проверить старую гипотезу, достаточно построить по произвольному набору из кластера гомографию от прошлого кадра к новому и применить ее ко всем точкам кластера, имеющим соответствия на новом кадре. Каждую гипотезу следует рассматривать до опровержения, до конца видео или до выхода соответствующего кластера особых точек из поля зрения камеры.

Предложенное решение задачи поиска и отслеживания плоскостей подразумевает решение следующих подзадач:

1. Поиск и сопоставление особых точек.
2. Выбор правдоподобных гипотез о существовании плоскости.
3. Выделение кластеров и наблюдение за ними.

Глава 2. Изменение положения камеры

2.1. Связь гомографии со сдвигом и поворотом

Изменение положения камеры между кадрами (см. рисунок 2) можно задать вектором смещения \mathbf{t}^* и матрицей поворота \mathbf{R} . Тогда между трехмерными координатами точек в разных системах координат, связанных с камерой, устанавливаются следующие соотношения:

$$\mathbf{X}_2 = \mathbf{R}\mathbf{X}_1 + \mathbf{t}^* \quad (3)$$

Запишем уравнение плоскости, заданной нормалью $\mathbf{n} = (a, b, c)^T$, в системе координат, связанной с первым положением камеры:

$$\mathbf{n} \cdot \mathbf{X}_1 = aX_1 + bY_1 + cZ_1 = d^* \quad (4)$$

Здесь величина d^* является расстоянием от оптического центра камеры до плоскости. Из (4) и (3) следует справедливость выражения:

$$\mathbf{X}_2 = \left(\mathbf{R} + \frac{\mathbf{t}^* \mathbf{n}^T}{d^*} \right) \mathbf{X}_1 \quad (5)$$

Поскольку \mathbf{X}_1 и \mathbf{X}_2 это координаты проекций на рассматриваемую плоскость точек изображений камеры m_1 и m_2 , для матрицы гомографии \mathbf{A} справедливо следующее соотношение:

$$\mathbf{A} = d\mathbf{R} + \mathbf{t}\mathbf{n}^T \quad (6)$$

Так как матрица гомографии \mathbf{A} строится по проекционным координатам, здесь величина d уже не является физическим расстоянием, а вектор \mathbf{t} лишь коллинеарен реальному вектору сдвига

$$\mathbf{t} = \frac{d}{d^*} \mathbf{t}^*$$

Соотношение (6) связывает матрицу гомографии \mathbf{A} с перемещением и поворотом камеры. Процесс декомпозиции матрицы гомографии заключается в поиске элементов правой части этого выражения при известной левой.

2.2. Декомпозиция матрицы гомографии

Для декомпозиции матрицы гомографии применим подход, изложенный в [3]. Этот подход основан на сингулярном разложении матрицы гомографии.

Сингулярное разложение матрицы \mathbf{A} - это представление ее в виде:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$$

Здесь \mathbf{U} и \mathbf{V} являются ортогональными матрицами ($\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}$), а $\mathbf{\Lambda}$ - диагональная матрица, на диагонали которой расположены собственные числа матрицы $\mathbf{A}\mathbf{A}^T$. Эти собственные числа d_i положительны и для определенности могут быть расположены в порядке: $d_1 \geq d_2 \geq d_3$

Используя это разложение, из (6) можно получить новое выражение:

$$\mathbf{\Lambda} = d'\mathbf{R}' + \mathbf{t}'\mathbf{n}'^T \quad (7)$$

\mathbf{R} , \mathbf{t} и \mathbf{n} связаны с \mathbf{R}' , \mathbf{t}' и \mathbf{n}' следующими соотношениями:

$$\begin{cases} \mathbf{R} = s\mathbf{U}\mathbf{R}'\mathbf{V}^T \\ \mathbf{t} = \mathbf{U}\mathbf{t}' \\ \mathbf{n} = \mathbf{V}\mathbf{n}' \\ d = sd' \\ s = \det\mathbf{U}\det\mathbf{V} \end{cases} \quad (8)$$

Заметим, что \mathbf{R}' также является матрицей поворота:

$$\mathbf{R}'\mathbf{R}'^T = \mathbf{I}, \quad \det\mathbf{R}' = 1$$

Используя канонический базис ($\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$) и выписав разложение вектора нормали $\mathbf{n}' = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + x_3\mathbf{e}_3$, можно получить из выражения (7) три векторных уравнения:

$$d_i\mathbf{e}_i = d'\mathbf{R}'\mathbf{e}_i + \mathbf{t}'x_i, \quad i = 1, 2, 3 \quad (9)$$

Поскольку \mathbf{n} единичный вектор, а \mathbf{V} ортогональная матрица, \mathbf{n}' также будет иметь единичную норму: $\sum_{i=1}^3 x_i^2 = 1$. Исключая \mathbf{t}' из (9), полу-

чаем:

$$d' \mathbf{R}'(x_j \mathbf{e}_i - x_i \mathbf{e}_j) = d_i x_j \mathbf{e}_i - d_j x_i \mathbf{e}_j, \quad i \neq j \quad (10)$$

Так как умножение на матрицу \mathbf{R}' сохраняет векторную норму, вычисляя нормы правых и левых частей, можем записать:

$$\begin{cases} (d'^2 - d_2^2)x_1^2 + (d'^2 - d_1^2)x_2^2 = 0 \\ (d'^2 - d_3^2)x_2^2 + (d'^2 - d_2^2)x_3^2 = 0 \\ (d'^2 - d_1^2)x_3^2 + (d'^2 - d_3^2)x_1^2 = 0 \end{cases} \quad (11)$$

Рассматривая (11) как линейную систему относительно x_1^2, x_2^2 и x_3^2 , получаем, что она может иметь нетривиальное решение только при равенстве определителя нулю:

$$(d'^2 - d_1^2)(d'^2 - d_2^2)(d'^2 - d_3^2) = 0$$

Возможны три случая:

1. $d_1 \neq d_2 \neq d_3$ и тогда $d' = \pm d_2$.
2. $d_1 = d_2 \neq d_3$ или $d_1 \neq d_2 = d_3$, и тогда $d' = \pm d_2$.
3. $d_1 = d_2 = d_3$ и $d' = \pm d_2$.

Решения $d' = \pm d_1$ и $d' = \pm d_3$ недопустимы. Для иллюстрации рассмотрим один из случаев. Пусть $d' = d_1$, тогда из (11) следует:

$$x_1 = 0 \text{ и } (d_1^2 - d_3^2)x_2^2 + (d_1^2 - d_2^2)x_3^2 = 0$$

Так как $d_1 > d_2 > d_3$, единственным решением является $x_2 = x_3 = 0$. Оно недопустимо, так как \mathbf{n}' имеет единичную норму.

Если $d_1 \neq d_3$, мы можем вычислить x_1, x_2 и x_3 , используя (11) и требование $\sum_{i=1}^3 x_i^2 = 1$:

$$\begin{cases} x_1 &= \varepsilon_1 \sqrt{\frac{d_1^2 - d_2^2}{d_1^2 - d_3^2}} \\ x_2 &= 0 \\ x_3 &= \varepsilon_3 \sqrt{\frac{d_2^2 - d_3^2}{d_1^2 - d_3^2}} \end{cases} \quad (12)$$

Здесь $\varepsilon_1, \varepsilon_3 = \pm 1$, а поскольку $d' = \pm d_2$, существует восемь различных решений.

Рассмотрим подробно три вышеописанных случая при $d' > 0$:

1. Рассматривая (9) при $i = 2$, получаем $\mathbf{R}'\mathbf{e}_2 = \mathbf{e}_2$. Значит, \mathbf{R}' является поворотом вида:

$$\begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix}$$

Используя выражения (10) и (12), находим:

$$\begin{cases} \sin \theta = (d_1 - d_3) \frac{x_1 x_3}{d_2} = \varepsilon_1 \varepsilon_3 \frac{\sqrt{(d_1^2 - d_2^2)(d_2^2 - d_3^2)}}{(d_1 + d_3)d_2} \\ \cos \theta = \frac{d_1 x_3^2 + d_3 x_1^2}{d_2} = \frac{d_2^2 + d_1 d_3}{(d_1 + d_3)d_2} \end{cases}$$

Подставляя эти значения в (9), получаем:

$$\mathbf{t}' = (d_1 - d_3) \begin{pmatrix} x_1 \\ 0 \\ -x_3 \end{pmatrix}$$

2. В этом случае $x_1 = x_2 = 0$, а $x_3 = \pm 1$. Тогда получаем:

$$\begin{cases} \mathbf{R}' = \mathbf{I} \\ \mathbf{t}' = (d_3 - d_1)\mathbf{n}' \end{cases}$$

3. Значения x_1, x_2 и x_3 определить невозможно, а из выражения (9) и (10) следует:

$$\begin{cases} \mathbf{R}' = \mathbf{I} \\ \mathbf{t}' = 0 \end{cases}$$

Аналогичные рассуждения в случае $d' < 0$ приводят к следующим результатам:

1. $\mathbf{R}'\mathbf{e}_2 = -\mathbf{e}_2$. Значит, \mathbf{R}' является поворотом вида:

$$\begin{pmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & -1 & 0 \\ \sin \varphi & 0 & -\cos \varphi \end{pmatrix}$$

при этом

$$\begin{cases} \sin \varphi = (d_1 + d_3) \frac{x_1 x_3}{d_2} = \varepsilon_1 \varepsilon_3 \frac{\sqrt{(d_1^2 - d_2^2)(d_2^2 - d_3^2)}}{(d_1 - d_3)d_2} \\ \cos \varphi = \frac{d_3 x_1^2 - d_1 x_3^2}{d_2} = \frac{d_1 d_3 - d_2^2}{(d_1 - d_3)d_2} \end{cases}$$

Подставляя эти значения в (9), получаем:

$$\mathbf{t}' = (d_1 + d_3) \begin{pmatrix} x_1 \\ 0 \\ -x_3 \end{pmatrix}$$

2. $x_1 = x_2 = 0$, а $x_3 = \pm 1$. Тогда получаем:

$$\begin{cases} \mathbf{R}' = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \mathbf{t}' = (d_3 + d_1)\mathbf{n}' \end{cases}$$

3. Из (7) следует, что $\mathbf{R}'\mathbf{x} = -\mathbf{x}$ для любого вектора \mathbf{x} , лежащего в плоскости, ортогональной \mathbf{n}' . Это значит, что \mathbf{R}' - симметрия, относительно оси \mathbf{n}' , поэтому $\mathbf{R}' = -\mathbf{I} + 2\mathbf{n}'\mathbf{n}'^T$. Тогда, в соответствии с (7), $\mathbf{t}' = -2d'\mathbf{n}'$.

2.3. Отсеивание лишних решений

Так как мы изначально не знаем знака величины d' , в общем случае мы имеем восемь возможных решений; в случае совпадения двух собственных чисел - четыре решения; в случае совпадения трех собственных чисел мы имеем неопределенность.

Пусть M - некоторая точка рассматриваемой плоскости, $\mathbf{X}_1(X_1, Y_1, Z_1)$

и $\mathbf{X}_2(X_2, Y_2, Z_2)$ - ее координаты в канонических системах координат, связанных с камерами (см. рисунок 2). Точка M находится в поле зрения обеих камер, поэтому $Z_1 > 0$ и $Z_2 > 0$. Используя выражения (1) и (2), получаем

$$\alpha \mathbf{X}_2 = \mathbf{A} \mathbf{X}_1, \quad (13)$$

где α - некоторое вещественное число. Из (6) и (3) следует, что $\alpha = d$, поэтому справедливо неравенство

$$\frac{Z_2}{Z_1} = \frac{a_{31}x_1 + a_{32}y_1 + a_{33}}{d} > 0$$

Пока точка M принадлежит плоскости, Z_1 и Z_2 сохраняют свой знак; d - постоянная величина, поэтому знак $a_{31}x_1 + a_{32}y_1 + a_{33}$ тоже не должен меняться пока M принадлежит плоскости. Значит, знание одной точки плоскости позволяет однозначно определить знак d . Так как $d' = sd$, а s зависит только от матрицы A , у нас остается только четыре решения, определенных знаком d' .

Оставшиеся решения определяют четыре возможных варианта вектора нормали $\mathbf{n}_1, -\mathbf{n}_1, \mathbf{n}_2$ и $-\mathbf{n}_2$ (см. (12)). \mathbf{X}_1 принадлежит плоскости, поэтому $\mathbf{n}^T \mathbf{X}_1 = d$. Используя (1) и тот факт, что $Z_1 > 0$, можно сформулировать дополнительное условие:

$$\frac{\mathbf{n}^T \mathbf{m}_1}{d} > 0$$

Это условие оставляет только два варианта решения из четырех.

В случае трех различных собственных чисел у нас остается только два решение из восьми, в случае совпадения двух собственных чисел корректным является единственное решение.

При анализе видеопотока избавиться от неопределенности позволяет анализ третьего кадра. В этом случае необходимо найти две пары решений и выбрать согласующиеся варианты, т.е. выбрать уравнение плоскости, соответствующее сразу трем кадрам.

2.4. Физическое расстояние

Поскольку $\alpha = d$, (13) можно переписать в виде:

$$\mathbf{X}_2 = \frac{1}{d} \mathbf{A} \mathbf{X}_1$$

Рассматривая это выражение вместе с (5) и (6), получаем:

$$\frac{1}{d} (d\mathbf{R} + \mathbf{t}\mathbf{n}^T) \mathbf{X}_1 = \left(\mathbf{R} + \frac{\mathbf{t}^* \mathbf{n}^T}{d^*} \right) \mathbf{X}_1$$

Раскрывая скобки, приходим к выражению для \mathbf{t}^* :

$$\mathbf{t}^* = \frac{\mathbf{t}\mathbf{n}^T \mathbf{X}_1}{d} = \frac{d^*}{d} \mathbf{t} \quad (14)$$

Таким образом, если нам известно физическое расстояние до плоскости или реальные координаты одной из ее точек, мы можем найти настоящий вектор перемещения камеры.

Глава 3. Построение траектории

3.1. Выбор кадров

В процессе анализа видеопотока естественным образом возникает вопрос о выборе последовательности кадров, по которой будет строиться траектория.

Непригодными для использования являются размытые кадры. Использование фильтров для восстановления четкости изображений может серьезно влиять на описания особых точек, поэтому в рамках данной работы данные методы не рассматриваются. Не получится использовать и кадры, на которых не удалось найти достаточное число особых точек. Отдельно стоит упомянуть движущиеся объекты в сцене - присутствие подвижных объектов может сильно влиять на корректность работы применяемых методов. Поэтому мы будем предполагать, что сцена неподвижна.

Если камера медленно меняет свое положение с течением времени или частота кадров видеопотока очень высокая, то рассматривать всю последовательность кадров крайне неэффективно. Во-первых, число кадров напрямую влияет на временные затраты; если мы работаем с видео высокого разрешения, то обработка каждого кадра будет занимать достаточно большое количество времени. Во-вторых, построение гомографии подразумевает, что мы имеем пару кадров, сделанных камерой в разных точках пространства. Если изображения очень похожи, то гомография, построенная по случайному множеству особых точек почти всегда будет признаваться правдоподобной.

Вторую проблему можно решить не прибегая к уменьшению количества кадров - для этого достаточно просто строить гомографию к следующему кадру не от предыдущего, а от того, который был за несколько кадров до него.

Возможные подходы к построению последовательности гомографий изображены на рисунке 4. В первом случае гомография строится между соседними кадрами последовательности. Во втором случае часть кадров пропускается, гомография строится по кадрам, равномерно распределенным по последовательности. В третьем случае кадры не пропускаются, но гомография строится не между соседними кадрами, а между кадрами, обо-

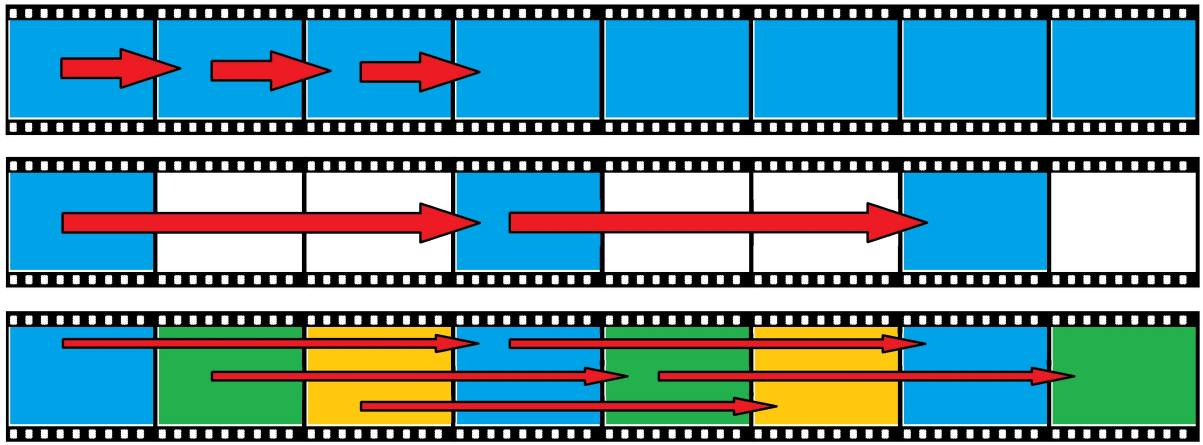


Рис. 4: Подходы к построению последовательности гомографий

значенными на рисунке одинаковым цветом.

У каждого из подходов есть свои преимущества и недостатки. Преимуществом первого варианта является высокая вероятность нахождения общих особых точек для построения гомографии, но, как было описано ранее, на похожих кадрах плохо работает RANSAC. Второй вариант является самым эффективным с точки зрения временных затрат; он может улучшить работу RANSAC, но при этом увеличивает шаг дискретизации при построении траектории. Третий подход не влияет на шаг дискретизации, но при этом улучшает работу RANSAC; по временным затратам он сопоставим с первым подходом, но требует хранения гораздо большего объема информации о предыдущих кадрах. Заметим, что в третьем случае процесс обработки кадров одного цвета может происходить независимо от обработки кадров всех других цветов.

3.2. Принцип ведущей плоскости

Если в процессе анализа видеопотока удалось обнаружить несколько плоскостей, но расстояние известно лишь до одной из них, то с помощью гомографии можно рассчитать расстояния и до других плоскостей.

Будем называть плоскость, расстояние до которой известно, ведущей плоскостью. Осуществив декомпозицию матрицы гомографии для ведущей плоскости, можно вычислить реальное перемещений \mathbf{t}^* камеры по формуле (14).

Если после этого для любой другой плоскости произвести декомпозицию матрицы гомографии и подставить в (14) найденное ранее значение реального сдвига, то, взяв норму от обеих частей равенства, можно полу-

чить выражение для модуля d^* :

$$|d^*| = |d| \frac{\|\mathbf{t}^*\|}{\|\mathbf{t}\|}$$

Здесь \mathbf{t} и d вычислены с помощью декомпозиции матрицы гомографии для не ведущей плоскости. Знак d^* зависит от знака d и от сонаправленности векторов \mathbf{t} и \mathbf{t}^* .

Этот метод вычисления расстояний незаменим в ситуациях, когда ведущая плоскость выходит из поля зрения камеры. Рассчитав расстояния до других плоскостей, мы можем на этапе обработки следующих кадров ориентироваться уже по ним.

Если у нас совсем нет информации о расстояниях до обнаруженных плоскостей, то можно выбрать в качестве ведущей любую из них, а расстояние до нее принять равным единице. В этом случае использование принципа ведущей плоскости позволяет при построении траектории сохранить единый масштаб.

3.4. Общая структура алгоритма

Общая структура алгоритма выглядит следующим образом:

1. Рассматриваем первые два кадра:
 - 1.1 Ищем особые точки и соответствия между ними.
 - 1.2 На множестве общих особых точек несколько раз используем RANSAC подход для поиска гомографии и на основе его результатов выделяем кластеры особых точек, определяющих плоскости.
2. Рассматриваем каждый следующий кадр:
 - 2.1 Ищем особые точки.
 - 2.2 Выделяем соответствия для старых кластеров.
 - 2.3 Если по какому-то из старых кластеров удастся построить гомографию, то, рассматривая последние три кадра, находим однозначное решение задачи декомпозиции матрицы гомографии. Если ведущая плоскость еще не назначена, назначаем плоскость, соответствующую кластеру, ведущей; иначе, если расстояние до плоскости еще не известно, используем принцип ведущей плоскости.

2.4 Старые кластеры, по которым не удалось построить гомографию, удаляются.

2.5 Из точек, не попавших в старые кластеры, но имеющих соответствия на предыдущем кадре, и из точек, попавших в удаленные старые кластеры, выделяем новые кластеры на основе RANSAC.

Алгоритм описан для последовательного подхода к выбору кадров; для других вариантов (рис. 4) нужно под следующим кадром понимать следующий кадр рассматриваемой подпоследовательности.

Ранее было упомянуто несколько методов поиска и описания особых точек. Принципы их использования примерно одинаковы, поэтому не влияют на общую структуру алгоритма.

Глава 4. Программная реализация

4.1. Выбор средств и тестирование

Для реализации алгоритма был выбран язык программирования C++ и библиотека методов компьютерного зрения OpenCV.

Для более подробного анализа эффективности различных подходов код программы предусматривает возможность выбора методов поиска и описания особых точек и настройки параметров на этапе компиляции. Также была реализована возможность выбора длины шага между кадрами.

Тестирование проводилось на нескольких видео с пиксельным разрешением 1920×1080 и частотой 30 кадров в секунду на ноутбуке с процессором Intel Core i5-3230 2.6 ГГц. Видеозаписи различались количеством плоскостей в сцене, дальностью до них, характером перемещений камеры.

4.2. Анализ эффективности алгоритма

В ходе тестирования алгоритм показал хорошие результаты при работе со сценой, содержащей единственный плоский объект. В случае нескольких плоскостей не всегда корректно происходит процесс кластеризации особых точек. Как следствие, невозможно оценить эффективность принципа ведущей плоскости.

Процесс построения траектории не удается продолжить при резких поворотах камеры и при большом количестве подряд идущих нечетких кадров.

Увеличение длины шага при выборе следующего кадра в большинстве случаев улучшает работу алгоритма, но иногда не удается найти достаточное количество совпадающих особых точек.

При поиске и описании особых точек ORB подход работает немного быстрее, чем SURF, но особые точки расположены гораздо кучнее, часто очень близко друг к другу. SURF дает более равномерное распределение особых точек. Использование для поиска соответствий метода ближайших соседей вместо полного перебора не дает заметного выигрыша по времени работы, но иногда серьезно ухудшает результаты.

Максимальной эффективности алгоритма получается достигнуть лишь при ручной настройке входных параметров применяемых методов. Универ-

сальной для любого видео комбинации параметров подобрать не удастся.

При пиксельном разрешении видеопотока 1920×1080 на обработку одного кадра уходит в среднем 3 секунды. Серьезного выигрыша во времени не дает ни одна из модификаций реализации алгоритма.

4.3. Способы улучшения алгоритма

Как показало тестирование, самым неэффективным этапом алгоритма является кластеризация на основе RANSAC подхода к поиску гомографии. Результаты такой кластеризации в случае нескольких плоскостей часто некорректны.

Качественного улучшения в работе алгоритма можно добиться с помощью поиска замнутых контуров. Если предварительно разбить изображения на участки, ограниченные контурами, и искать плоские объекты на каждом участке отдельно, то результаты такой двухуровневой кластеризации могут оказаться гораздо лучше, чем у текущего варианта алгоритма.

Поиск и сопоставление особых точек является одной из причин больших временных затрат. Можно попробовать повысить эффективность алгоритма путем использования методов отслеживания особых точек, основанных на вычислении оптического потока.

Заключение

В рамках данной работы был разработан алгоритм нахождения траектории движения камеры по ее видеопотоку, основанный на процессе декомпозиции матрицы гомографии.

В ходе работы над алгоритмом был предложен метод для поиска и отслеживания плоскостей на видео; была решена проблема перехода от решения задачи декомпозиции матрицы гомографии к реальным физическим величинам; сформулирован и применен принцип ведущей плоскости.

На основе тестовых данных был проведен анализ эффективности алгоритма, были выявлены его недостатки и намечены пути улучшения.

Поскольку траектория движения камеры может служить основой для проведения восстановления модели движения, в дальнейших работах планируется применение методов идентификации.

Список литературы

1. Computer Vision: Algorithms and Applications. <http://szeliski.org/Book/>
2. Faugeras O., Toscani G. Camera calibration for 3d machine vision // Proc. International Workshop on Machine Vision and Machine Intelligence, February 2-5 1987, P. 240-247.
3. Faugeras O., Lustman F., Motion and structure from motion in a piecewise planar environment // International Journal of Pattern Recognition and Artificial Intelligence, 1988. Vol. 2, No. 3. P. 485–508.
4. Karami E., Prasad S., Shehata M. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images // Newfoundland Electrical and Computer Engineering Conference, November 2015.
5. Laganiere R. OpenCV 2 Computer Vision Application Programming Cookbook. Birmingham: Packt Publishing Ltd., 2011. 287 p.
6. Malis E., Vargas M. Deeper understanding of the homography decomposition for vision-based control. Sophia Antipolis: INRIA, 2007. 90 p.
7. OpenCV documentation index. <http://docs.opencv.org>
8. Prateek J., Escrivá D. M., Godoy V. OpenCV By Example. Birmingham: Packt Publishing Ltd., 2016. 274 p.

Приложение

Данный раздел посвящен работе с внутренними параметрами камеры. Матрица внутренних параметров камеры \mathbf{K} определяет переход от евклидовых координат изображения $\mathbf{m}(x, y, 1)$ к пиксельным $\mathbf{p}(u, v, 1)$:

$$\mathbf{p} = \mathbf{K}\mathbf{m}$$

Следовательно, найденная по двум пиксельным изображениям матрица гомографии \mathbf{H} , связана с поворотом и сдвигом камеры более сложным соотношением, чем ранее описанная матрица \mathbf{A} :

$$\mathbf{H} = d\mathbf{K}(\mathbf{R} + \mathbf{t}\mathbf{n}^T)\mathbf{K}^{-1}$$

Поэтому, если мы знаем матрицу \mathbf{K} , то матрица гомографии для евклидового пространства определяется следующим образом:

$$\mathbf{A} = \mathbf{K}^{-1}\mathbf{H}\mathbf{K}$$

Это выражение описывает необходимый в нашем алгоритме этап перехода от матрицы \mathbf{H} к матрице \mathbf{A} .