Система трехмерной реконструкции на основе многоракурсных изображений

Введение

Данный проект реализует систему трехмерной реконструкции на основе многоракурсных изображений. Путем обработки нескольких изображений, снятых с разных ракурсов, система извлекает особые точки, устанавливает их соответствия и в итоге реконструирует трехмерное облако точек объекта.

Этапы реализации

Проект разделен на следующие основные этапы:

1. Извлечение и описание особых точек
2. Сопоставление особых точек
3. Оценка положения камеры
4. Реконструкция трехмерного облака точек
5. Визуализация облака точек

1. Извлечение и описание особых точек

Этапы реализации:

Предварительная обработка изображения (преобразование в оттенки серого)

Исходным входным изображением является цветное изображение, которое преобразуется в оттенки серого по формуле яркости:

$$I = 0.298 * r + 0.588 * g + 0.114 * b$$

Вычисление градиента изображения

Здесь мы используем полученное изображение в оттенках серого и операторы Собеля (G_x) и (G_y) для выполнения операции свертки. Поскольку после свертки изображение уменьшается в размерах, перед операцией свертки необходимо выполнить добавление границ (padding).

Гауссов фильтр

Перед следующим шагом применяется гауссов фильтр для подавления шума и сглаживания изображения. Сначала строится гауссов ядро, а затем выполнить свертку изображения с гауссовским ядром.

Детектор углов Харриса

Здесь мы используем метод Харриса для обнаружения значений отклика R на углах. Обычно коэффициент k принимается равным 0.04:

$$R = det(M) - k * (trace(M))^2$$

$$det(M) = I_x^2 * I_y^2 - (I_x I_y)^2$$

$$trace(M) = I_x^2 + I_y^2$$

Подавление не-максимумов

Нам необходимо пройти по матрице, проверить 3x3 область вокруг каждого пикселя и сохранить точку с наибольшим значением отклика в локальной области как ключевую точку. В итоге мы получаем словарь, содержащий положение угловой точки (i, j) и соответствующее значение отклика R.

Использование алгоритма SIFT (scale-invariant feature transform) для создания дескрипторов

После обнаружения ключевых точек с помощью метода Харриса у нас есть набор точек с их позициями. Однако одной информации о позициях недостаточно для установления соответствий между разными изображениями. Для этого каждому ключевому пункту генерируется дескриптор признака, описывающий окружающие характеристики изображения.

Здесь мы реализовали упрощенную версию алгоритма SIFT: вместо построения гауссовой пирамиды и поиска экстремумов в масштабе, мы используем ключевые точки, обнаруженные методом Харриса.

Вычисление градиентов изображения

Используя градиенты изображения по x и y, вычисляются величина и направление градиента для каждого пикселя.

Согласно литературе, ядро Гаусса является единственным, способным создавать пространство масштабов. Пространство масштабов L(x,y,σ) определяется как свёртка исходного изображения I(x,y) с двумерной гауссовой функцией G(x,y,σ) переменного масштаба.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$theta(x,y) = \arctan\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right)$$

Поскольку мы не вычисляем пространство масштабов, используются упрощенные формулы:

$$m(x,y) = \sqrt{I_x(x,y)^2 + I_y(x,y)^2}$$

$$theta(x,y) = \arctan\left(\frac{I_y(x,y)}{I_x(x,y)}\right)$$

где m - величина градиента, theta - его направление.

Определение основного направления ключевой точки

Для каждой ключевой точки вычисляется основное направление, что обеспечивает инвариантность дескриптора к повороту:

- Для каждой ключевой точки строится гистограмма направлений градиентов в её окрестности. Гистограмма разбивается на 36 бинов по 10 градусов каждый, где высота бина представляет суммарную величину градиентов данного направления
- Применяется гауссово взвешивание: чем дальше пиксель от ключевой точки, тем меньше его вклад в характеристику точки
- Пик гистограммы принимается за основное направление ключевой точки

Создание SIFT-дескриптора:

- Вокруг каждой ключевой точки выделяется область 16×16 пикселей
- Область разделяется на 4×4 подобласти
- В каждой подобласти вычисляется гистограмма по 8 направлениям градиентов
- Формируется 128-мерный дескриптор (согласно литературе, 128-мерный дескриптор даёт оптимальный результат), учитывающий основное направление ключевой точки. Все углы нормализуются для повышения устойчивости к изменениям освещения

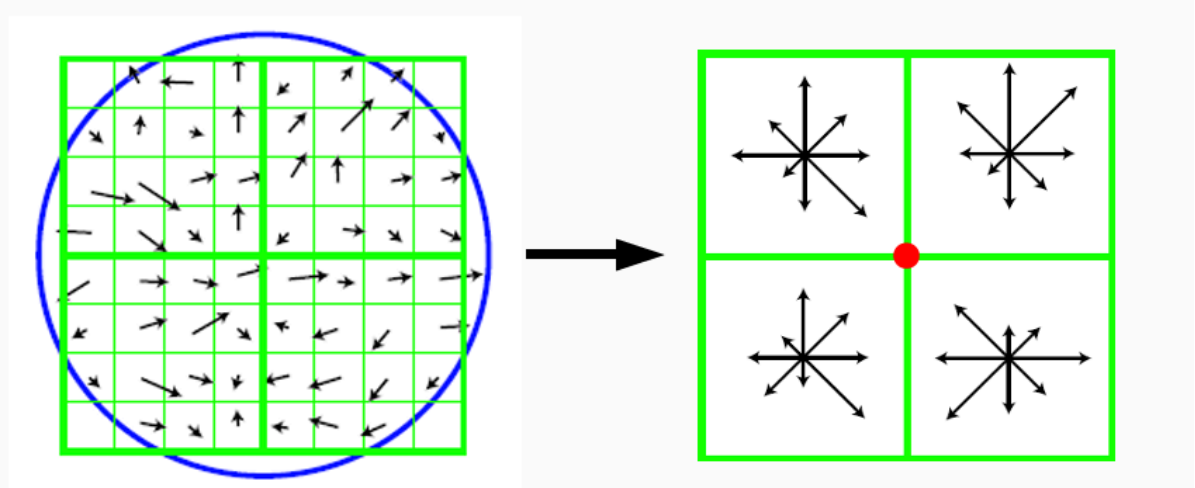

Рисунок 1: Градиенты изображения вокруг ключевой точки и сформированный дескриптор

- Нормализованный дескриптор сохраняется в словаре ключевых точек и добавляется в список дескрипторов.

Недостатки и возможные улучшения

Детектор углов Харриса имеет вычислительную сложность O(height×width×W^2), W - размер ядра Гаусса. Общая временная сложность SIFT составляет O(height × width + N), где N - количество ключевых точек. В данной реализации мы не построили пространство масштабов. Оригинальный алгоритм SIFT включает построение пирамиды Гаусса для обнаружения ключевых точек на разных масштабах. Вместо этого мы использовали ключевые точки, обнаруженные методом Харриса. Неполная реализация SIFT может повлиять на последующие этапы.

2. Сопоставление особых точек

Методы реализации:

Вычисление евклидова расстояния между дескрипторами

Тест отношения ближайших соседей (Nearest Neighbor Distance Ratio)

Тест отношения (Ratio Test) - это эффективный метод фильтрации соответствий, предложенный Lowe в статье о SIFT.

1. Для особой точки A находятся ближайшее соответствие B1 (ближайший сосед) и второе ближайшее соответствие B2 (второй ближайший сосед)
2. Вычисляется отношение: расстояние до ближайшего соседа / расстояние до второго ближайшего соседа
3. Если это отношение меньше порогового значения, умноженного на расстояние до второго ближайшего соседа, соответствие считается хорошим

Значение этого теста:

- Если соответствие хорошее, оно должно быть значительно лучше других возможных соответствий
- Пороговое значение подбирается эмпирически и зависит от конкретных изображений:
  - Уменьшение порога (например, до 0.7) даёт более надёжные, но менее многочисленные соответствия
  - Увеличение порога (например, до 0.9) даёт больше соответствий, но менее надёжных
- Этот метод фильтрации значительно уменьшает количество ложных соответствий

SVD разложение

В дальнейшем мы будем использовать 8-точечный метод для оценки фундаментальной матрицы. В этом методе мы строим систему однородных линейных уравнений $Af = 0$, которую нам нужно решить.

Где:

- A - матрица размером $N \times 9$ ($N \geq 8$), содержащая информацию о соответствующих точках
- $\mathbf{f}$ - 9-мерный вектор, полученный из фундаментальной матрицы F
- Наша цель - найти ненулевой вектор $\mathbf{f}$, минимизирующий $A\mathbf{f}$

Необходимо выполнить SVD разложение матрицы A:

$A = U\Sigma V^T$

- U - ортогональная матрица размером m×m, каждый столбец которой является собственным вектором AA^T
- Σ - диагональная матрица размером m×n, содержащая сингулярные значения
- V^T - транспонированная ортогональная матрица n×n, каждый столбец которой является собственным вектором A^TA

Нам нужно вычислить f, который является правым сингулярным вектором A, соответствующим наименьшему сингулярному значению:

- Собственные векторы A^TA являются правыми сингулярными векторами $\mathbf{v}_i$
- Собственные векторы AA^T являются левыми сингулярными векторами $\mathbf{u}_i$

Последний столбец v_9 матрицы V, соответствующий наименьшему сингулярному значению σ_9, и есть искомый вектор f.

Оптимизация результатов сопоставления с помощью алгоритма RANSAC

1. Случайно выбираем 8 пар точек
2. Вычисляем фундаментальную матрицу F
3. Проверяем соответствие всех точек матрице F (используя расстояние от точки до эпиполярной линии для определения инлайеров; идеальное соответствие имеет нулевое расстояние, но из-за различных факторов возникают ошибки, поэтому установит порог, и точки с расстоянием меньше порога считаются инлайерами
4. Сохраняем хорошие соответствия (инлайеры)
5. Повторяем процесс несколько раз, выбирая лучший результат

В RANSAC инлайеры (inliers) - это точки данных, соответствующие текущей модели, а выбросы (outliers) - точки, не соответствующие модели.

Сопоставление особенностей для всех пар изображений

- Использование алгоритма RANSAC для удаления неверных соответствий
- Возврат результатов сопоставления для всех пар изображений

Построение траекторий особых точек

Присваивание одинаковых ID траекторий для соответствующих ключевых точек.

Визуализация траекторий особых точек

Для тестирования предыдущих алгоритмов реализована визуализация траекторий соответствующих особых точек на исходном изображении с использованием библиотеки matplotlib.

Недостатки и возможные улучшения

В этой части есть несколько проблем. Во-первых, проблема эффективности: сложность метода сопоставления составляет O(N×M). При использовании улучшенных структур данных, например, kd-дерева, можно снизить сложность сопоставления . В RANSAC используется фиксированный порог, но для разных изображений разные пороги дают разные результаты. Лучше было бы адаптировать порог RANSAC на основе распределения данных.

Кроме того, мы реализовали алгоритмы SVD и RANSAC без использования внешних библиотек, что включает сложные матричные операции и может содержать незамеченные ошибки. Эти ошибки могут вызвать серьезную цепную реакцию. Если SVD разложение неверно, это приведет к ошибкам в оценке фундаментальной матрицы, затем к ошибкам в оценке существенной матрицы, что сделает оценку положения камеры неточной и приведет к плохим результатам реконструкции.

3. Оценка положения камеры

Методы реализации:

Первый этап: оценка существенной матрицы E

Цель: Существенная матрица E содержит информацию о движении камеры:

E = [t]× * R

где:

- R - матрица поворота камеры (3x3)
- t - вектор переноса камеры (3x1)
- [t]× - кососимметричная матрица t:
  если t = [tx, ty, tz],
- форму

$$[t]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

1. Сначала проверяем случай чистого вращения
   - Вычисляем среднее смещение особых точек между изображениями

- Если смещение очень мало, считаем это чистым вращением и возвращаем нулевую матрицу, так как при чистом вращении t=0, а поскольку существенная матрица $E = [t]_\times R$, то при t=0 матрица E будет нулевой

2. Оценка фундаментальной матрицы F
    - Используем 8-точечный алгоритм для построения системы уравнений
    - Решаем с помощью SVD для получения F
    - Принудительно устанавливаем ранг F равным 2 (обнуляем наименьшее сингулярное значение)
3. Вычисление существенной матрицы E
    - Используем формулу $E = K^T F K$
    - K - матрица внутренних параметров камеры
    - Принудительно устанавливаем сингулярные значения E как (1,1,0) (этот шаг обеспечивает алгебраические свойства существенной матрицы: ранг 2 и равенство двух ненулевых сингулярных значений)

Второй этап: извлечение положения камеры из E

1. SVD-разложение матрицы E
    - E = U S V^T
2. Построение четырех возможных решений
    - Использование специальной матрицы W для вычисления двух возможных матриц вращения R1 и R2
    - Вектор переноса t является последним столбцом матрицы U
    - Комбинирование даёт четыре возможных варианта: (R1,t), (R1,-t), (R2,t), (R2,-t), из которых только один является правильным

При выполнении SVD-разложения матрицы E:

$$E = U * S * V^T$$

Существует два способа восстановления матрицы вращения R:

$R1 = U * W * V^T$ (Первый вариант матрицы вращения)

$R2 = U * W^T * V^T$ (Второй вариант матрицы вращения)

Матрица W является вспомогательной матрицей для извлечения матрицы вращения R из существенной матрицы E.

4. Реконструкция облака точек

Методы реализации:

Первый этап: триангуляция особых точек

Цель триангуляции:

Восстановление координат 3D-точки X по её проекциям p1 и p2 на двух ракурсах камеры с известным относительным положением (R,t)

1. Построение матриц проекции

- Первая камера: P1 = [I|0] (положение первой камеры принимается за начало мировой системы координат)
- Вторая камера: P2 = [R|t] (P2 описывает преобразование относительно P1, где [R|t] представляет движение камеры из первого положения во второе)

где K - матрица внутренних параметров камеры, [R|t] - матрица внешних параметров

2. Составление системы линейных уравнений
    - Использование проективных соотношений
    - Каждая пара точек даёт два уравнения
3. Определение координат 3D-точки
    - Решение системы уравнений методом SVD
    - Нормализация результата

Второй этап: выбор правильного положения камеры

Критерии выбора правильного положения:

- Триангулированные 3D-точки должны находиться перед обеими камерами (положительная глубина)
- Ошибка репроекции должна быть минимальной

1. Для каждого возможного положения:
    - Триангуляция всех соответствующих точек для получения 3D-координат
    - Проверка положительности глубины 3D-точек относительно обеих камер
    - Вычисление ошибки репроекции (здесь можно было бы использовать оптимизацию методом связок (Bundle Adjustment) для уточнения положений камер и 3D-точек, минимизируя ошибку репроекции, но реализация без внешних библиотек слишком сложна, поэтому здесь его не использовали)
2. Выбор оптимального положения:
    - Подсчёт количества точек с положительной глубиной
    - Выбор положения с максимальным числом точек с положительной глубиной и минимальной ошибкой репроекции

5. Визуализация облака точек

Для визуализации полученного облака точек использована библиотека matplotlib.

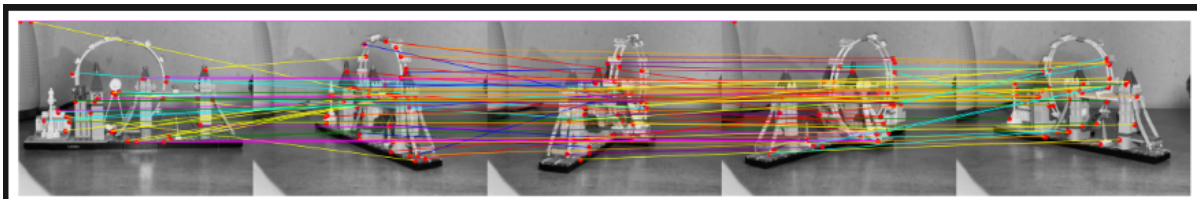

Рисунок 2: Визуализация соответствующих точек и их траекторий

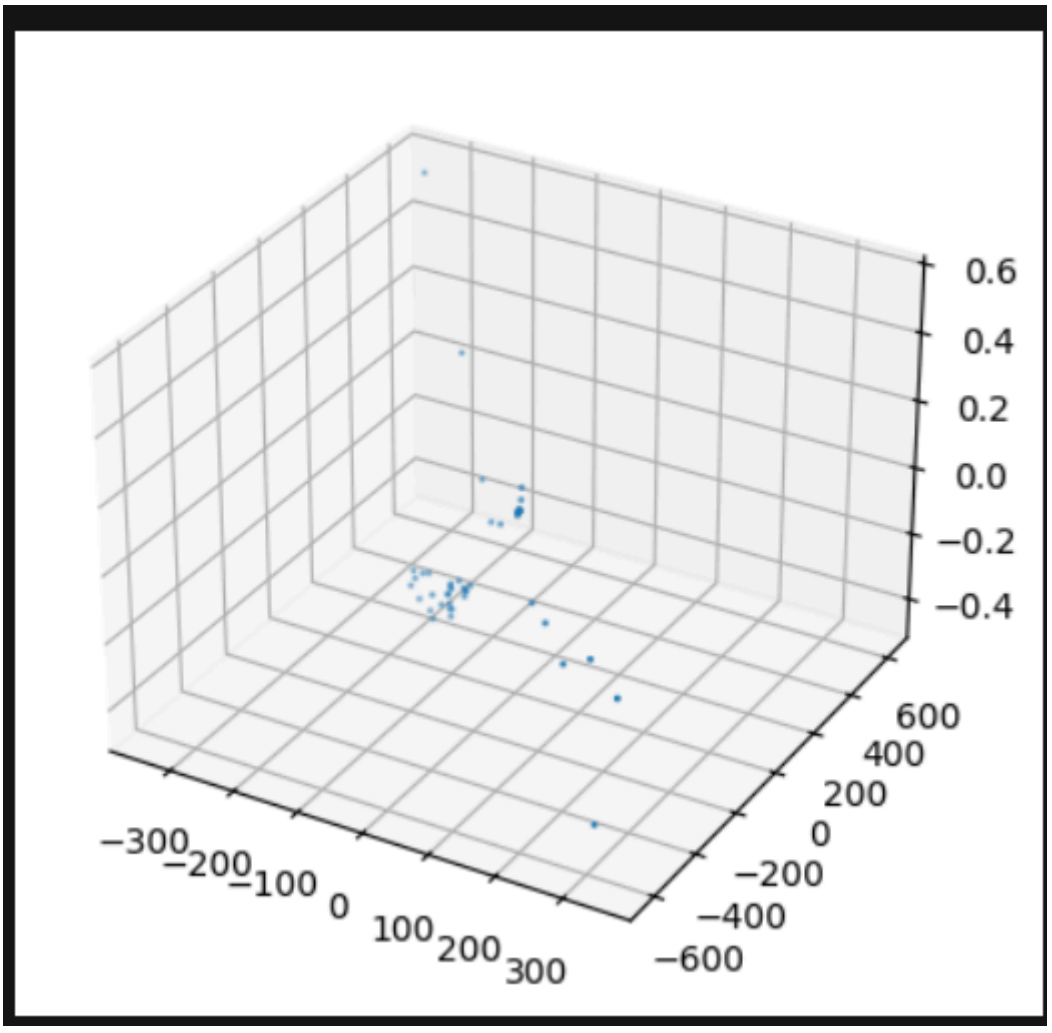

Рисунок 3: Визуализация облака точек

Заключение

В данном проекте реализована система трехмерной реконструкции на основе многоракурсных изображений, однако результаты оказались не удовлетворительными по нескольким причинам. Во-первых, не была полностью реализована система SIFT, во-вторых, реализация SVD и RANSAC без использования внешних библиотек могла содержать ошибки в матричных вычислениях, что привело к неточной оценке фундаментальной матрицы. SVD является ключевым этапом в оценке положения камеры и триангуляции, и любые ошибки существенно влияют на результат. Точность этих матриц напрямую влияет на восстановление положения камеры и реконструкцию облака точек. Кроме того, на конечный результат влияет точность вычисления матрицы внутренних параметров камеры, и ошибка на любом этапе может привести к неудачной реконструкции.

Литература

1. Image_processing_algorithms_*SPbU* (5)
2. January 5, 2004, David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints
3. D Kalman, 2002, A Singularly Valuable Decomposition: The SVD of a Matrix
4. Tony Lindeberg 1994, Scale-space theory: a basic tool for analyzing structures at different scales
5. Ashwin Nanjappa, 2015, How to compute intrinsic camera matrix for a camera