

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Системное программирование

Жуков Роман Сергеевич

Методы реконструкции статических  
трехмерных сцен с использованием  
алгоритмов автоматического определения  
положения и фокусного расстояния камер

Бакалаврская работа

Научный руководитель:  
к. ф.-м. н. Вахитов А. Т.

Рецензент:  
Кривоконь Д. С.

Санкт-Петербург  
2016

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems

Software Engineering

Roman Zhukov

Methods of reconstruction of static 3D scenes  
using algorithms for automatic determining  
position and focal length of cameras

Bachelor's Thesis

Scientific supervisor:  
Ph. D. Alexander Vakhitov

Reviewer:  
Dmitry Krivokon

Saint-Petersburg  
2016

# Оглавление

<b>Введение</b>	<b>4</b>
<b>1. Постановка задачи</b>	<b>6</b>
<b>2. Обзор</b>	<b>7</b>
2.1. Метод определения положения камер EPnP . . . . .	7
2.2. Метод определения положения и фокусного расстояния камер EPnP <sub>f</sub> R . . . . .	8
<b>3. Реализация метода EPnP<sub>f</sub>R</b>	<b>10</b>
<b>4. Алгоритмические улучшения EPnP<sub>f</sub>R</b>	<b>13</b>
4.1. Изменение структуры алгоритма . . . . .	13
4.2. Изменение метода нахождения внешних параметров камеры . . . . .	15
<b>5. Интеграция в проект Bundler</b>	<b>16</b>
5.1. Структура Bundler . . . . .	16
5.2. Замена метода DLT на EPnP <sub>f</sub> R . . . . .	18
<b>Заключение</b>	<b>20</b>
<b>Список литературы</b>	<b>21</b>

# Введение

На сегодняшний день технологии стали очень доступными. Люди могут фотографировать все вокруг себя и делиться этим в интернете, потому что почти у каждого в телефоне или планшете есть камера. Таким образом, в сети мы можем найти большие, постоянно растущие коллекции фотографий городов и достопримечательностей со всего мира, сделанных со всевозможных ракурсов. Например, при запросе "Собор Парижской Богоматери" поиск изображений Google выдает более 15 000 фотографий, которые могут быть использованы в качестве входных наборов данных для 3D моделирования.

Проблема создания точных трехмерных моделей представляет большой интерес и имеет широкую область применения. Например, модели городов необходимы для городского планирования и визуализации. Помимо этого, реконструированные 3D сцены важны для таких научных дисциплин, как история, археология, география и компьютерная графика, и могут использоваться в дополненной реальности. Например, навигационные приложения с 3D моделями и дополненной реальностью помогут людям гораздо лучше ориентироваться в незнакомых местах.

Существуют решения задачи реконструкции статических трехмерных сцен по наборам изображений. В большинстве из них считается, что фокусные расстояния камер заранее известны, в противном случае производится их грубая оценка. Так как камеры на многих современных устройствах имеют функцию автофокуса, возможности узнать фокусное расстояние нет. В связи с этим было принято решение попытаться улучшить существующие решения с помощью алгоритмов с более точным определением положения и фокусного расстояния камер.

Проблема оценивания положения камеры по соответствиям 3D координат точек и их проекций на плоскость камеры является очень известной в компьютерном зрении и называется PnP или PnPf, если неизвестно фокусное расстояние. Существует множество решений PnP проблемы. Главное требование к этим решениям — быстрое время выполнения для нескольких сотен точек при сохранении приемлемой точности.

Методом, удовлетворяющим данному требованию и имеющим одно из лучших соотношений скорости и точности, является EPnP [5].

Совсем недавно был представлен метод EPnPfR [4], который является расширением EPnP для задачи с неизвестным фокусом. Этот метод не уступает в точности своим конкурентам, но зато выигрывает в скорости. В EPnPfR было значительно уменьшено пространство поиска решений путем введения регуляризации. Также с помощью предобработки данных удалось сократить время выполнения методов, основанных на EPnP, для большого количества точек, что в [9] было указано в качестве недостатка данных методов.

В качестве системы для 3D реконструкции был выбран проект Bundler [7]. Он является одним из крупнейших открытых проектов в своем роде и лежит в основе многих масштабных работ. Одной из таких, например, является Building rome in a day [1].

# 1. Постановка задачи

Целью работы является развитие и улучшение алгоритма EPnPfR, и исследование его применимости в методах реконструкции статических трехмерных сцен.

В рамках работы были поставлены следующие задачи:

- реализация алгоритма определения положения и фокусного расстояния камер EPnPfR на языке Си++;
- улучшение и оптимизация данного алгоритма;
- интеграция полученного решения в Bundler;
- апробация и анализ полученных результатов.

## 2. Обзор

### 2.1. Метод определения положения камер EPnP

Метод EPnP оценивает положение камеры по соответствиям мировых 3D координат точек  $\mathbf{p}_i^w$  и их проекций  $\mathbf{u}_i$  на плоскость камеры.

Сначала находятся контрольные точки  $c_j^w$  такие, что

$$\mathbf{p}_i^w = \sum_{j=1}^4 \alpha_{ij} c_j^w, \quad i = 1 \dots n, \quad \sum_{j=1}^4 \alpha_{ij} = 1. \quad (1)$$

Индексы  $^w$  и  $^c$  означают какие координаты имеются в виду: мировые или в системе координат камеры соответственно.

Затем рассматривается матрица внутренних параметров камеры  $A$ , которая участвует в следующих соотношениях:

$$\forall i, \quad \omega_i \begin{bmatrix} \mathbf{u}_i \\ 1 \end{bmatrix} = A \mathbf{p}_i^c = A \sum_{j=1}^4 \alpha_{ij} c_j^c. \quad (2)$$

Эти соотношения формируют линейную систему с матрицей  $M$  размера  $2n \times 12$ :

$$Mx = 0, \quad (3)$$

где  $x = [c_1^c, c_2^c, c_3^c, c_4^c]$  — двенадцатимерный вектор неизвестных. Найдя координаты контрольных точек в системе координат камеры, можно вычислить внешние параметры камеры: матрицу поворота  $R$  и вектор перемещения  $t$ .

Решение системы (3) лежит в ядре матрицы  $M$  и может быть представлено в виде следующей линейной комбинации:

$$x = \sum_{i=1}^N \beta_i v_i, \quad N = 1 \dots 4, \quad (4)$$

где  $v_i$  — правые сингулярные векторы матрицы  $M$ , соответствующие  $N$  нулевым сингулярным числам.

На следующем шаге используется тот факт, что при евклидовом

преобразовании одной системы координат в другую сохраняются расстояния между точками:

$$\|c_i^c - c_j^c\|^2 = \|c_i^w - c_j^w\|^2 = r_{ij}^2, \quad (5)$$

где  $r_{ij}$  — известное расстояние между точками с номерами  $i$  и  $j$ . Так как контрольных точек 4, то из равенств расстояний получаются 6 уравнений, которые являются квадратными относительно  $\beta_i$ , но в то же время линейными относительно  $b$ :

$$b = ( \beta_1^2 \cdots \beta_N^2 \beta_1\beta_2 \cdots \beta_{N-1}\beta_N ). \quad (6)$$

Получающаяся система, линейная относительно  $b$ , записывается следующим образом:

$$Lb = r, \quad (7)$$

где  $r$  — вектор расстояний между контрольными точками.

На последнем шаге система (7) решается для  $N = 1, \dots, 4$ . При  $N \geq 3$  уравнений становится недостаточно для решения, и тогда используется техника под названием релинеаризация, подробное описание которой можно найти в статье, посвященной EPnP [5].

## 2.2. Метод определения положения и фокусного расстояния камер EPnPFr

Метод EPnPFr решает ту же задачу, что и EPnP, но при условии, что фокусное расстояние камеры является неизвестным. Следовательно основные шаги алгоритма тоже сохраняются.

При наличии шума в проекциях точек на плоскость камеры система (3) не остается точной, поэтому она решается в смысле наименьших квадратов:

$$F_1(x) = \|Mx\|^2 \rightarrow \min. \quad (8)$$



При подстановке в  $F_1$  линейную комбинацию (4) получается:

$$F_1(x) = F_1 \left( \sum_{i=1}^N \beta_i v_i \right) = \sum_{i=1}^N \beta_i^2 \sigma_i^2, \quad (9)$$

где  $\sigma_i$  - сингулярные числа матрицы  $M$ , соответствующие векторам  $v_i$ .

Система уравнений (7) также рассматривается в смысле наименьших квадратов:

$$F_2(x) = \|Lb - r\|^2 \rightarrow \min. \quad (10)$$

Для нахождения коэффициентов  $\beta_i$ , предлагается регуляризовать  $F_1$  с помощью  $F_2$ , то есть рассмотреть следующую функцию:

$$F_R(x) = F_2(x) + \gamma F_1(x), \quad (11)$$

где  $\gamma$  — коэффициент регуляризации.

К оригинальной  $6 \times D$  матрице  $L$  добавляется диагональная матрица  $L_R^1$  размера  $N \times D$  с соответствующими сингулярными числами  $M$ :

$$L_R^1 = (\text{diag}(\sigma_1, \dots, \sigma_N) \mathbf{0}). \quad (12)$$

Таким образом, задача минимизации  $F_R$  становится эквивалентной решению в смысле наименьших квадратов следующей расширенной системы:

$$\hat{L}b = \hat{r}, \quad \hat{L} = \begin{pmatrix} L \\ L_R^1 \end{pmatrix}, \quad \hat{r} = \begin{pmatrix} r \\ \mathbf{0} \end{pmatrix}. \quad (13)$$

Для решения задачи РнРф определяются функции  $F_1^f, F_2^f$  и  $F_R^f$  аналогичные  $F_1, F_2$  и  $F_R$ , но зависящие не только от  $\beta_i$  и  $b$ , но и от  $f^2 b, f \beta_i$  для  $i = 1, \dots, N$ .

### 3. Реализация метода EPnPfR

Причины, по которым понадобилась реализация EPnPfR на Си++:

1. Единственная реализация представлена только на языке Matlab;
2. Повышение скорости выполнения;
3. Интеграция в существующие проекты, так как Bundler и большинство других написано на Си++.

Изучив различные библиотеки, для реализации были выбраны следующие:

- Eigen — библиотека линейной алгебры, одна из наиболее быстрых в своем роде;
- OpenCV — для решения задач, связанных с компьютерным зрением;
- Levmar — Си++ реализация алгоритма Левенберга-Марквардта для решения задач о наименьших квадратах;
- Clp — библиотека для решения задач линейного программирования.

Основными критериями при выборе библиотек были их скорость и открытый исходный код.

В качестве основы для Си++ реализации использовались статья [4], описывающая метод, и Matlab реализация.

После того, как метод EPnPfR был реализован, было проведено тестирование. Точность Си++ реализации получилась точно такой же, как у Matlab реализации. На графиках, изображенных на рисунке 1, можно увидеть сравнение точности с одним из последних методов в этой области GPnPf, который представлен в двух вариантах: с нелинейным уточнением и без. Также можно сравнить с одним из лучших

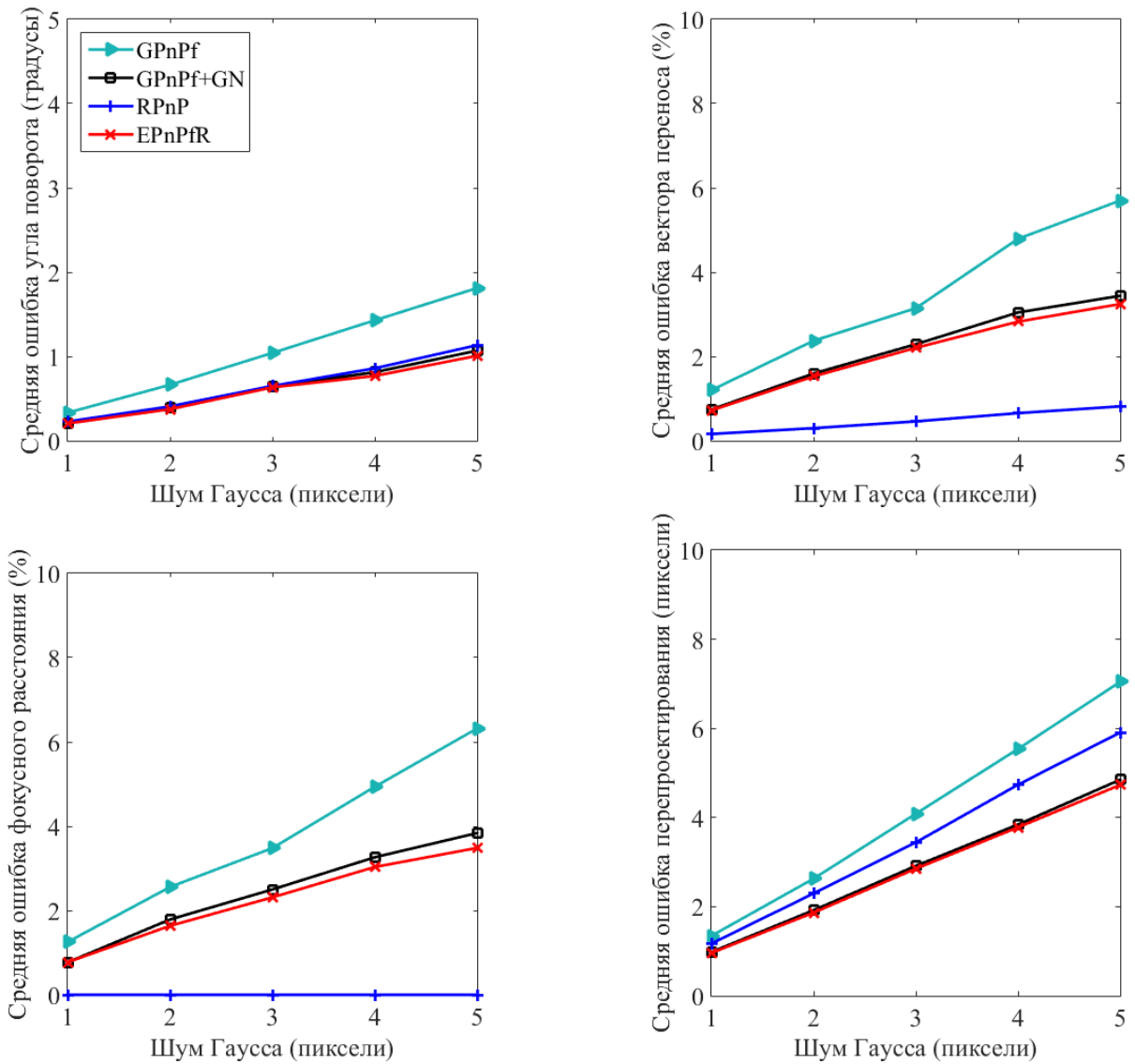


Рис. 1: Сравнение методов определения положения и фокусного расстояния камер

РnP методов RPnP для того, чтобы посмотреть, как неизвестное фокусное расстояние влияет на оценку параметров (этот метод запускался с известным фокусным расстоянием).

В тестах моделировалась камера с центром в начале координат и размером изображения 800 x 600. Фокусное расстояние выбиралось случайно в промежутке от 200 до 2200 с равномерным распределением. Также случайно генерировались 7 точек и делалось 500 попыток для каждого значения дисперсии шума. Стандартное отклонение шума с нормальным распределением изменялось от 1 до 5 с шагом 1.

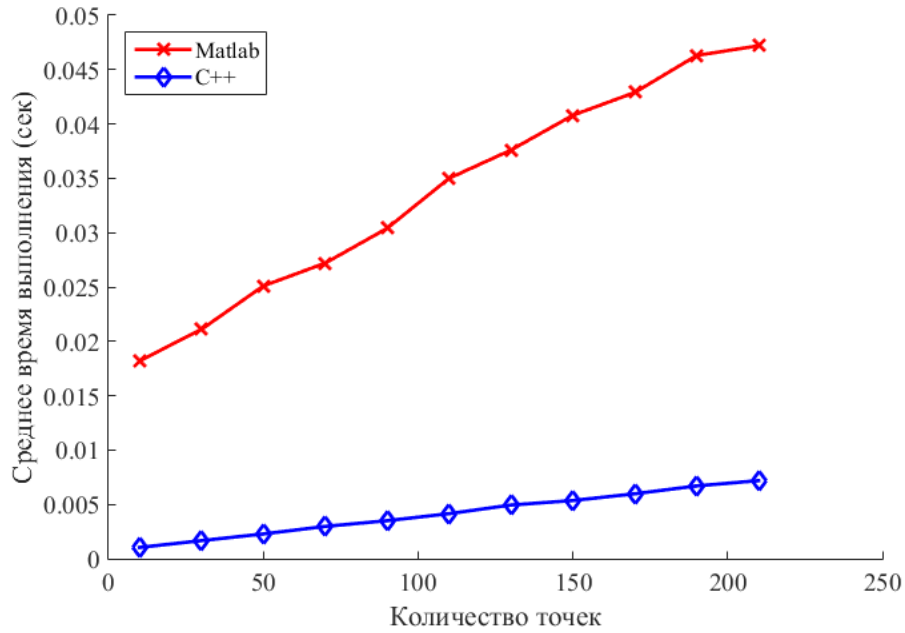


Рис. 2: Сравнение времени выполнения Си++ и Matlab реализаций

Также сравнивалось время работы Си++ и Matlab реализаций. Для этого были проведены тесты с количеством точек от 10 до 210 с шагом 20 и фиксированной дисперсией шума  $\delta = 2$ . На рисунке 2 представлены результаты сравнения, из которых видно, что Си++ реализация получилась гораздо быстрее.

Затем проводился анализ времени работы отдельных компонент метода EPnPR, в ходе которого были выявлены недостатки некоторых библиотек. Например, оказалось, что в Eigen очень много времени тратится на SVD разложение матриц, в то время как метод из OpenCV справляется с этой задачей в несколько раз быстрее.

## 4. Алгоритмические улучшения EPnPfR

### 4.1. Изменение структуры алгоритма

Следующим этапом данной работы являются алгоритмические улучшения, которые были сделаны для того, чтобы ускорить метод EPnPfR, но при этом несильно потерять в точности.

Чтобы описать первое улучшение необходимо более подробно рассмотреть некоторые шаги алгоритма.

После того, как вычисляются значения  $f$  и  $\beta_i$  для  $i = 1, \dots, N$ , находится решение  $x$  (4), которое в свою очередь является координатами контрольных точек в системе координат относительно камеры. Зная координаты точек в двух системах координат, с помощью специального метода вычисляются внешние параметры камеры: матрица поворота  $R$  и вектор перемещения  $t$ . Затем происходит уточнение этих параметров путем минимизации ошибки перепроектирования:

$$res = \sum_i dist^2 \left( A[R|t] \begin{bmatrix} \mathbf{p}_i^w \\ 1 \end{bmatrix}, \mathbf{u}_i \right), \quad (14)$$

где  $dist(m, n)$  — расстояние на плоскости между точкой  $m$ , выраженной в однородных координатах, и точкой  $n$ .

Оказалось, что больше всего времени тратится на алгоритм Левенберга-Марквардта, который решает задачу о наименьших квадратах для уменьшения ошибки перепроектирования. Проблема в том, что происходит уточнение семи неизвестных:  $\mathcal{Z}$  — матрица поворота  $R$  в форме Родрига,  $\mathcal{Z}$  — вектор перемещения  $t$ ,  $1$  — фокусное расстояние  $f$ , причем на каждом шаге уточнения вычисляется ошибка (14), которая зависит от количества точек, а оно может быть очень большим.

Было решено изменить структуру метода EPnPfR.

После того, как находится вектор  $b$  (6), происходит его уточнение с помощью алгоритма Левенберга-Марквардта путем минимизации функции

$$\hat{F}_2^f(x) = \left\| \hat{L}b - \hat{r} \right\|^2 \rightarrow min, \quad (15)$$

где  $\hat{L}$  и  $\hat{r}$  из (13). Улучшение заключается в том, что необходимо уточнить от 2 до 5 неизвестных в зависимости от  $N$ , для которого вычисляются коэффициенты (4). Причем функция  $\hat{F}_2^f$  не зависит от количества входных точек, так как размеры матрицы  $\hat{L}$  и вектора  $\hat{r}$  фиксированы.

Далее находится решение  $x$ , из которого, как уже было описано выше, вычисляются внешние параметры камеры  $R$  и  $t$ . На этот раз уточнения этих параметров не происходит.

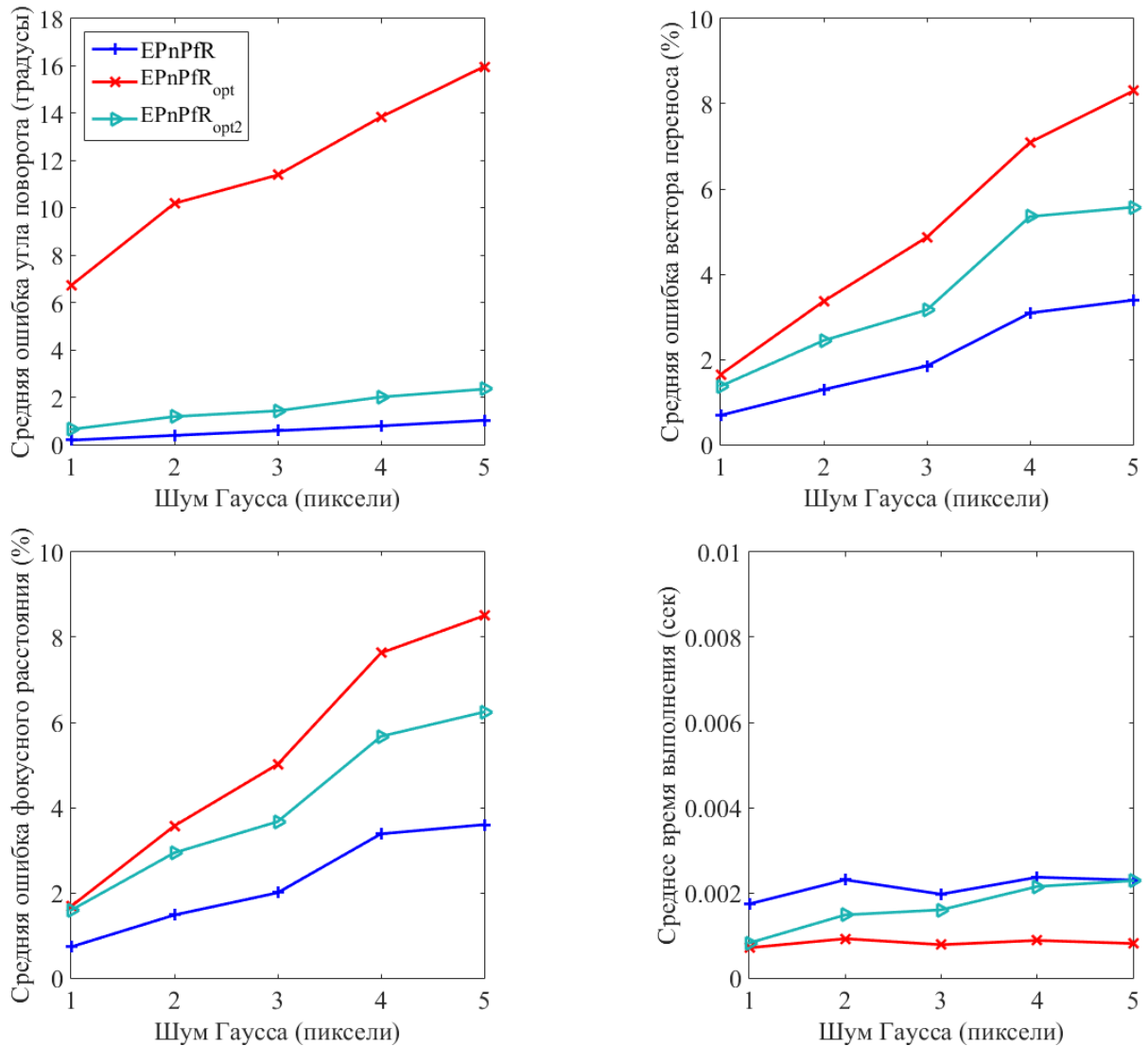


Рис. 3: Результаты алгоритмических изменений

Точность и время работы улучшенного метода в сравнении с оригинальным можно увидеть на графиках, изображенных на рисунке 3. EPnPfR<sub>opt</sub> обозначает улучшенный метод. Получилось, что точность

оценки параметров ухудшилась, особенно это заметно в случае с оценкой угла поворота камеры, но в то же время удалось добиться значительного выигрыша в скорости.

## 4.2. Изменение метода нахождения внешних параметров камеры

Потеря точности оказалась довольно большой для того, чтобы полученный вариант метода EPnPfR мог конкурировать с остальными методами. Поэтому были рассмотрены различные варианты нахождения внешних параметров камеры из решения  $x$ .

Метод, который используется в EPnPfR, основывается на однородных координатах. Для преобразования мировых координат контрольных точек в координаты относительно камеры необходимо умножить эти координаты на матрицу, содержащую  $R$  и  $t$ . Так как решение  $x$ , то есть координаты контрольных точек в системе координат относительно камеры, и мировые координаты известны, можно найти матрицу преобразования и вычленив из нее необходимые параметры.

Вместо этого было решено применить более сложный метод, использующийся в EPnP. Он также основывается на преобразовании систем координат, но уже не однородных. Более того, он использует координаты всех точек, а не только контрольных, что повышает точность нахождения  $R$  и  $t$ , но замедляет скорость.

На рисунке 3 показаны результаты применения нового метода нахождения  $R$  и  $t$ , где EPnPfR<sub>opt2</sub> — рассматриваемый метод после применения двух изменений. Точность оценки параметров повысилась и стала сравнимой с оригинальным методом. Время выполнения улучшенного метода также повысилось, но все равно осталось меньше, чем у оригинального. В некоторых случаях удалось достичь двукратного увеличения скорости.

В итоге, поставленная цель была достигнута. Несмотря на небольшую потерю точности, удалось добиться ускорения метода.

## 5. Интеграция в проект Bundler

### 5.1. Структура Bundler

Bundler представляет собой крупную систему, на вход которой подается неупорядоченная коллекция фотографий, а в результате получается облако особых точек в пространстве и восстановленные положения камер. Достоинствами этого проекта является открытость исходного кода и то, что он лежит в основе приложения для пространственного просмотра фотографий [7] и используется во многих крупных работах [8], [1]. В связи со всем этим, было принято решение попытаться интегрировать улучшенную реализацию метода EPnPR и посмотреть, как это повлияет на результаты работы Bundler.

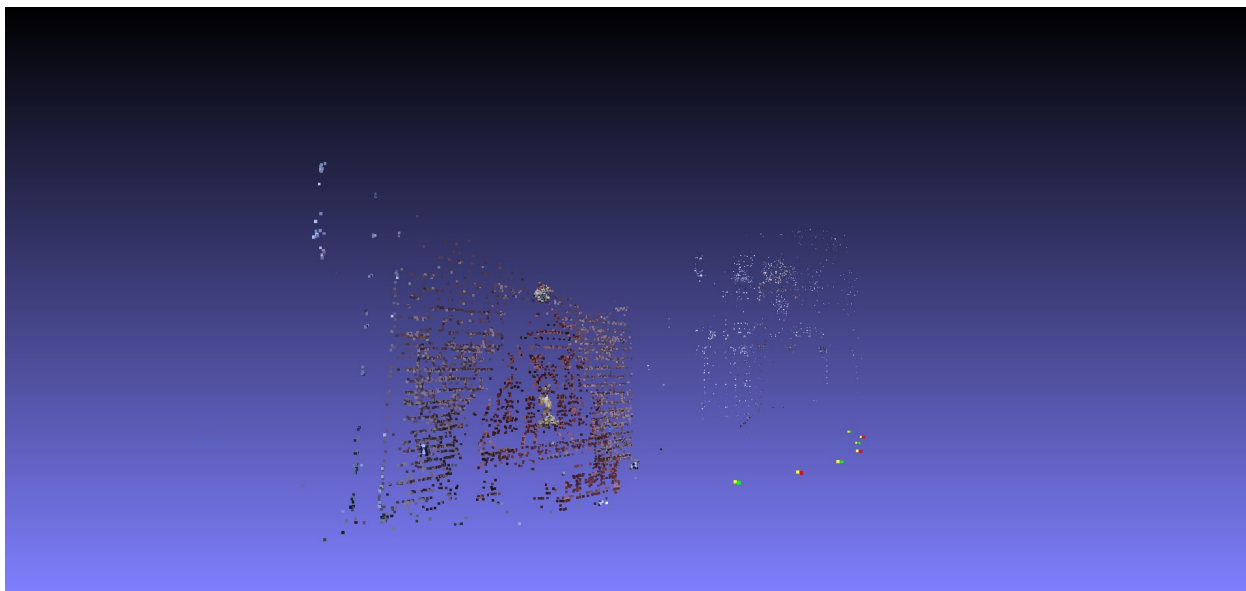


Рис. 4: Пример 3D реконструкции с помощью Bundler на наборе изображений fountain-P11

Для того, чтобы понять куда и как интегрировать метод, была изучена структура системы. Вначале на всех изображениях детектируются особые точки методом SIFT и затем сопоставляются на разных изображениях. Для каждой точки по соответствиям строится трек, с помощью которого можно проследить положение точки.

На следующем шаге выбираются 2 изображения с наибольшим количеством соответствий, и по ним строится первоначальная трехмерная



модель и находятся пространственные координаты особых точек. Потом одно за другим добавляются остальные изображения, причем на каждой итерации происходит оценка положения камеры и добавление новых треков.

Основная задача заключается в минимизации ошибки перепроектирования путем подбора внешних и внутренних параметров камеры, и она решается с помощью метода bundle adjustment [6]. Этот метод является очень точным, но для его работы необходимы начальные приближения неизвестных параметров. Оценка матрицы поворота  $R$  и вектора перемещения  $t$  производится методом DLT (Direct Linear Transformation [3]). Фокусное расстояние оценивается этим же методом в случае, если не удалось считать его из дополнительной информации к файлу с изображением.

Можно заметить, что метод DLT решает ту же самую задачу, что и EPnPfR, а значит его можно заменить, причем это единственное возможное место для применения нашего метода. Но для начала было проведено сравнение методов EPnPfR и DLT, чтобы понять, имеет ли смысл проводить замену.

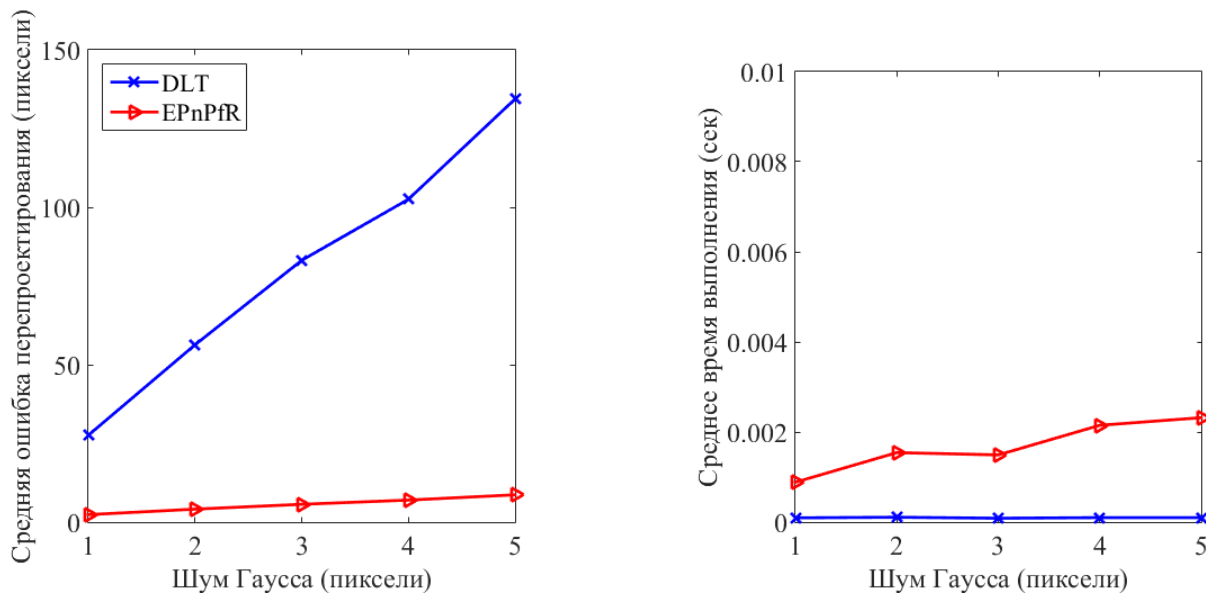


Рис. 5: Сравнение улучшенного метода EPnPfR и DLT

На графиках, изображенных на рисунке 5, видно, что измененный нами EPnPfR гораздо точнее, а главным достоинством DLT является

его скорость. Тем не менее, наш метод достаточно быстрый для того, чтобы его можно было применить в Bundler и посмотреть, как повлияет более точная первоначальная оценка параметров на результат.

## 5.2. Замена метода DLT на EPnPfR

С помощью инструмента CMake была создана статическая библиотека, содержащая оптимизированную реализацию EPnPfR. А после изучения того, как с использованием make-файлов собирается Bundler, удалось понять куда и как нужно интегрировать полученную библиотеку, чтобы можно было заменить использование DLT метода.

При первой попытке интеграции результаты оказались плохими. Точность первоначальной оценки параметров камер сильно ухудшилась по сравнению с тем, что было до этого. Соответственно ухудшился результат работы bundle adjustment, который уточнял эти параметры, и итоговая трехмерная модель стала очень неточной.

Необходимо было выяснить в чем заключается проблема. После изучения реализации DLT метода, которая используется в Bundler, оказалось, что помимо этого метода применяется фильтрация точек, являющихся входными данными для оценки. Фильтрация происходит с помощью метода RANSAC [2].

Название	Количество изображений	Оригинальный Bundler	Измененный Bundler
fountain-P11	11	2	11
Herz-Jesu-P8	8	6	8
castle-P19	19	5	9
Herz-Jesu-P25	25	16	21
castle-P30	30	11	17

Таблица 1: Количество добавляемых в 3D модель изображений оригинальным и измененным Bundler из различных наборов изображений

В следующей попытке было решено использовать EPnPfR вместе с RANSAC, взяв и исправив его готовую реализацию из Bundler. На этот раз результаты оказались гораздо лучше. В таблице 1 можно сравнить

количество добавляемых в 3D модель изображений оригинальным и измененным Bundler из различных наборов изображений с неизвестными фокусными расстояниями. Они часто используются в компьютерном зрении для проверки работы подобных решений. Получилось, что в реконструированную трехмерную модель стало включаться больше изображений. Не все из входных изображений могут быть добавлены, так как первоначальная оценка их внешних и внутренних параметров является очень большой.

Название	Оригинальный Bundler, %	Измененный Bundler, %
fountain-P11	46.7	41.3
Herz-Jesu-P8	91.6	82.9
castle-P19	43.5	37.5
Herz-Jesu-P25	55.3	49.7
castle-P30	35.1	27.8

Таблица 2: Средние значения ошибок оценки фокусного расстояния камер на различных наборах изображений

В таблице 2 можно увидеть, что также повысилась точность нахождения фокусного расстояния камер.

Таким образом, удалось добиться увеличения количества изображений, используемых для построения трехмерной модели, а также увеличения точности оценки фокусного расстояния. Все это приводит к более лучшей и детальной реконструкции трехмерной сцены, что является очень важным результатом.

Однако, несмотря на общие улучшения, получаемая точность 3D модели оказалась не очень хорошей. Это связано с тем, что метод определения относительного положения камер, используемый в Bundler, не предназначен для применения в ситуации, когда фокусное расстояние камеры неизвестно. Реконструкция начинается с применения этого метода. При этом опубликован код метода [10], устойчиво работающего в этой ситуации. В будущем планируется также встроить этот метод в систему.

## Заключение

В результате работы было выполнено следующее:

- метод EPnPfR реализован на языке Си++;
- сделаны алгоритмические улучшения метода;
- удалось интегрировать полученную реализацию в Bundler;
- проведен анализ полученных результатов;
- удалось увеличить количество добавляемых в реконструируемую модель изображений с неизвестным фокусным расстоянием.

В продолжении данной работы планируется изменение метода построения трехмерной модели по двум изображениям, используемого в Bundler, на тот, который будет более точно работать для изображений с неизвестным фокусным расстоянием камеры.

## Список литературы

- [1] Building rome in a day / Sameer Agarwal, Yasutaka Furukawa, Noah Snavely et al. // Communications of the ACM. — 2011. — Vol. 54, no. 10. — P. 105–112.
- [2] Fischler Martin A, Bolles Robert C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography // Communications of the ACM. — 1981. — Vol. 24, no. 6. — P. 381–395.
- [3] Hartley Richard, Zisserman Andrew. Multiple view geometry in computer vision. — Cambridge university press, 2003.
- [4] Kanaeva Ekaterina, Gurevich Lev, Vakhitov Alexander. Camera Pose and Focal Length Estimation Using Regularized Distance Constraints // Proceedings of the British Machine Vision Conference (BMVC) / Ed. by Mark W. Jones Xianghua Xie, Gary K. L. Tam. — BMVA Press, 2015. — September. — P. 162.1–162.12.
- [5] Lepetit Vincent, Moreno-Noguer Francesc, Fua Pascal. Epnnp: An accurate o (n) solution to the pnp problem // International journal of computer vision. — 2009. — Vol. 81, no. 2. — P. 155–166.
- [6] The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm : Rep. / Technical Report 340, Institute of Computer Science-FORTH, Heraklion, Crete, Greece ; Executor: Manolis Lourakis, Antonis Argyros : 2004.
- [7] Snavely Noah, Seitz Steven M, Szeliski Richard. Photo tourism: exploring photo collections in 3D // ACM transactions on graphics (TOG) / ACM. — Vol. 25. — 2006. — P. 835–846.
- [8] Snavely Noah, Seitz Steven M, Szeliski Richard. Modeling the world from internet photo collections // International Journal of Computer Vision. — 2008. — Vol. 80, no. 2. — P. 189–210.

- [9] A general and simple method for camera pose and focal length determination / Yinqiang Zheng, Satoshi Sugimoto, Imari Sato, Masatoshi Okutomi // Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on / IEEE. — 2014. — P. 430–437.
- [10] A minimal solution for relative pose with unknown focal length / Henrik Stewénus, David Nistér, Fredrik Kahl, Frederik Schaffalitzky // Image and Vision Computing. — 2008. — Vol. 26, no. 7. — P. 871–877.