

Санкт-Петербургский государственный университет  
**Факультет прикладной математики–процессов управления**  
**Кафедра компьютерного моделирования и многопроцессорных**  
**систем**

**Сурков Никандр Максимович**

**Выпускная квалификационная работа бакалавра**

**Создание карты рельефов местности по спутниковым  
изображениям на платформе Android**

Направление 01.03.02

Прикладная математика и информатика

Заведующий кафедрой,  
доктор физ.-мат. наук, профессор

Андианов С.Н.

Научный руководитель,  
кандидат технических наук, доцент

Гришкин В. М.

Рецензент,  
доктор физ.-мат. наук, профессор

Матросов А.В.

Санкт-Петербург,  
2018 г.

# Содержание

<b>Введение</b> . . . . .	<b>3</b>
<b>Постановка задачи</b> . . . . .	<b>5</b>
<b>Обзор литературы</b> . . . . .	<b>6</b>
<b>Глава 1. Особенности работы со спутниковыми стереоснимками</b> . . . . .	<b>7</b>
1.1. Диспаратность и цифровая модель высот . . . . .	7
1.2. Стереосопоставление . . . . .	7
1.3. Начальные данные . . . . .	9
1.4. Искажение изображений . . . . .	11
<b>Глава 2. Распределение вычислительных задач</b> . . . . .	<b>13</b>
<b>Глава 3. Выбор спутниковых данных</b> . . . . .	<b>15</b>
<b>Глава 4. Алгоритм построения карты рельефов</b> . . . . .	<b>17</b>
4.1. Аффинное эпиполярное выравнивание . . . . .	17
4.2. Blob detection с помощью использования фильтра LoG . .	22
4.3. Построение карты диспаратностей с помощью алгоритма Semi-Global Matching . . . . .	23
4.4. Взвешенный медианный фильтр . . . . .	26
4.5. Триангуляция . . . . .	27
4.6. Вывод информации пользователю Android приложения . .	28
<b>Глава 5. Экспериментальные результаты и сравнения</b> . . . . .	<b>31</b>
5.1. Сравнение аффинного эпиполярного выравнивания и преобразования с использованием гомографии . . . . .	31
5.2. Использование LoG фильтра . . . . .	36
5.3. Сравнение локального алгоритма сопоставления и алгоритма сопоставления Semi-Global Matching . . . . .	38

5.4. Сравнение 3D форматов, используемых в Android приложе- ниях . . . . .	40
5.5. Выводы из экспериментальных данных . . . . .	41
<b>Заключение . . . . .</b>	<b>43</b>
<b>Список литературы . . . . .</b>	<b>45</b>

# Введение

Цифровая модель рельефа описывается набором точек в трехмерном пространстве. Она может быть представлена в растровой (например, с помощью форматов GeoTiff, ASCIIGRID) или другой произвольной форме. Вместо выражения “цифровая модель рельефа” также может использоваться термин “цифровая модель высот”. Цифровая модель высот дает трехмерное представление области, для которой она была сделана.

Цифровая модель высот играет фундаментальную роль в картографии. Цифровое описание трехмерной поверхности важно для многих применений. Сегодня наиболее часто используемым фотограмметрическим продуктом является ортоизображение, созданное с помощью одного изображения и цифровой модели высот. Космические датчики с очень высоким разрешением в основном работают в режиме одного изображения. Правильная геопривязка возможна только на основе корректной цифровой модели высот. Но такая корректная цифровая модель высот должна быть создана. Существующие цифровые модели высот обычно не имеют достаточного уровня качества и надежности для прикладных задач, требующих высокой точности, либо их использование дорого в обслуживании.

На сегодняшний день существует ряд программных средств, позволяющих осуществлять создание карт рельефов местности. Среди основных программных продуктов можно выделить следующие:

- 1) ERDAS Imagine;
- 2) ArcGIS;
- 3) PCI Geomatica;
- 4) ENVI.

Основным минусом большинства существующих программных реализаций задачи создания карты рельефов местности является низкий уровень автоматизации процесса. Вмешательство человеческого фактора требуется практически на всех уровнях в процессе работы. Это делает задачу построения карты рельефов местности для большого количества спутниковых данных затратной по временным и человеческим ресурсам.

Также следует подчеркнуть, что многие существующие программные продукты являются коммерческими и распространяются на платной основе.

Большинство существующих программных продуктов, решающих задачу построения карты рельефов местности, не имеют кроссплатформенной реализации и требуют от пользователя использования специфического программного обеспечения. Такой подход значительно сокращает количество потенциальных пользователей.

## **Постановка задачи**

В данной работе основной задачей является написание алгоритма для создания карты рельефов местности по спутниковым изображениям и поиск возможных подходов для вывода конечной карты рельефов пользователю Android приложения. Такая постановка задачи требует решения следующих основных проблем:

- 1) выбор спутниковых данных для работы;
- 2) поиск подходящих решений для предварительной обработки спутниковых снимков;
- 3) выбор алгоритма для построения карты диспаратностей;
- 4) создание карты рельефов на основе карты диспаратностей;
- 5) сохранение карты рельефов в наиболее удобной для дальнейшего использования форме;
- 6) вывод результирующих данных пользователю.

Также целью данной работы является изучение и сравнение существующих подходов в задаче получения карты рельефов местности по спутниковым стереоснимкам.

# Обзор литературы

Алгоритм построения карты рельефов местности включает в себя несколько основополагающих этапов. Первым шагом обработки стереоизображений чаще всего выступает ректификация снимков. Наиболее полную информацию о геометрии стереоизображений и подходах к их ректификации и приведению к общей плоскости можно найти в книге [7]. В рамках рассматриваемой задачи наиболее интересны главы, касающиеся аффинной геометрии. Более детальное описание возможностей ректификации стереоизображений посредством аффинного эпиполярного выравнивания можно найти в статье [5].

Следующим шагом была рассмотрена работа со световым балансом на снимках. Этот шаг необходим для сглаживания различий в освещенности одинаковых участков Земли на левом и правом изображениях стереопары. Для решения данной задачи была использована статья [10], описывающая работу фильтра LoG<sup>1</sup>.

Третьим шагом является стереосопоставления снимков. Общие положения о том, как происходит стереосопоставление, приведены в статье [1]. В данной работе были рассмотрены и реализованы два алгоритма стереосопоставления: локальный и алгоритм SGM<sup>2</sup>. В статьях [11], [12] приведены основные принципы работы локального алгоритма. Подробную информацию относительно работы алгоритма SGM можно найти в статьях [13], [14], [15], [16]. Результатом работы алгоритмов стереосопоставления является карта диспаратностей.

Следующим шагом в данной работе был рассмотрен алгоритм, помогающий уточнить карту диспаратностей с помощью взвешенного медианного фильтра. Его подробная реализация описана в статье [17].

Последним шагом в построении карты рельефов местности является триангуляция - получение данных об относительных высотах из карты диспаратностей. Подробная информация о методе приведена в статье [19].

---

<sup>1</sup>Laplacian of Gaussian filter - (от англ.) фильтр лапласиана от гауссиана.

<sup>2</sup>Semi-Global Matching - (от англ.) полу-глобальный алгоритм.

# Глава 1. Особенности работы со спутниковыми стереоснимками

## 1.1. Диспаратность и цифровая модель высот

Стереоизображения позволяют генерировать два связанных типа данных, описывающих информацию о высоте в рассматриваемой местности: карту диспаратностей и цифровую модель высот. Оба этих продукта представляют из себя реконструкцию третьего измерения (высот) для двухмерной проекции участка Земли.

Карта диспаратностей и цифровая модель высот имеют схожее представление и близкую связь относительно описания спроектированной поверхности. Цифровая модель высот является результатом применения фотограмметрической триангуляции на множестве точек, полученных путем сопоставления изображений. С другой стороны, карты диспаратностей получаются путем сопоставления измерений на двух перекрывающихся изображениях в эпиполярной геометрии. Диспаратности представляют расстояния сдвигов (в пикселях) пикселя одного изображения относительно подобного пикселя на сопряженном стереоизображении. Эти сдвиги обусловлены изменениями углов обзора для изображений.

## 1.2. Стереосопоставление

Стереосопоставление используется для создания карты диспаратностей для рассматриваемой области, спроектированной на снимках из разных точек. С ее помощью можно произвести оценку глубины для данной области. На рисунке 1 проиллюстрирована базовая стереогеометрия, где  $O$  и  $O'$  центры левого и правого изображений соответственно.  $p_L$  и  $p_R$  - точки, принадлежащие левому и правому изображениям соответственно, они представляют из себя  $2D$  проекции точки из  $3D$  пространства.

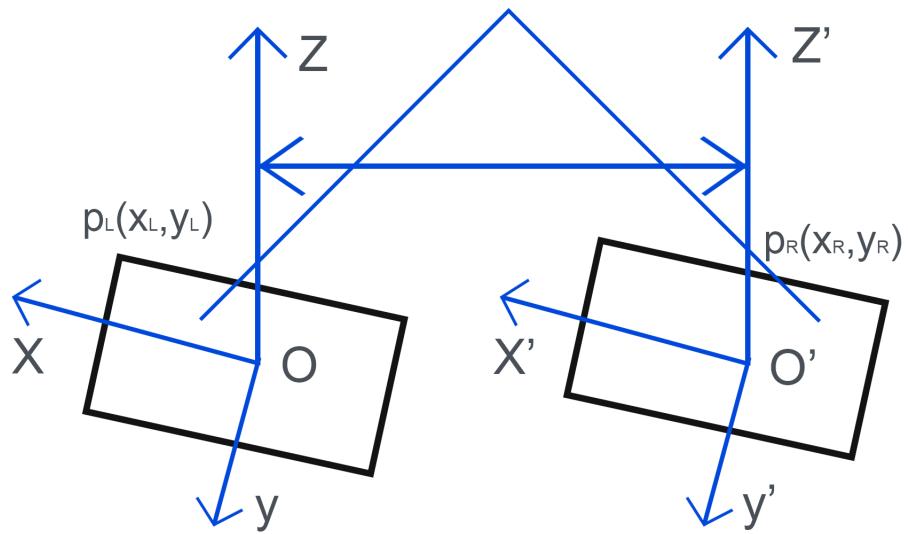


Рис. 1. Базовая стереогеометрия

Можно записать отношения, отраженные на рисунке 1, с помощью формулы (3). Решая уравнения (1) и (2) можно вычислить значение глубины точки в пространстве и представить как  $z_0$ .  $\lambda$  - фокусное расстояние. Значение  $z_0$  рассчитывается на основе коэффициента знаменателя, который называется диспаратностью, как указано в уравнении (3).

$$\frac{x_0}{x_L} = \frac{y_0}{y_L} = \frac{\lambda - z_0}{\lambda} \quad (1)$$

$$\frac{x_0 + \Delta x}{x_R + \Delta x} = \frac{y_0}{y_R} = \frac{\lambda - z_0}{\lambda} \quad (2)$$

$$z_0 = \frac{\lambda + \lambda \Delta x}{x_L - (x_R + \Delta x)} \quad (3)$$

Пиксели с меньшей диспаратностью находятся дальше от камеры, а пиксели с большим значением диспаратности находятся ближе к камере. Другими словами, глубина является обратной величиной по отношению к диспаратности, как показано в равенстве (3). Чтобы найти точную глубину объектов, необходимо иметь точную карту диспаратностей.

### 1.3. Начальные данные

Для генерации цифровой модели высот с помощью оптических изображений нам необходимо иметь 2 и более изображений, показывающих одну и ту же область со всех сторон (рисунок 2). Центр проекции должен быть известен в заданной системе координат объекта в дополнение к направлению просмотра для правильного определения координат точек Земли. Если необходимо создать ортоизображение, в дополнение к цифровой модели высот требуется только одно изображение (рисунок 3). Если доступна только средняя высота, расхождения в горизонтальном положении  $\Delta l$  вызываются в зависимости от несоответствия высоты (рисунок 4, формула (4)). Такое же соотношение имеет место для расхождений  $\Delta h$  цифровой модели высот.

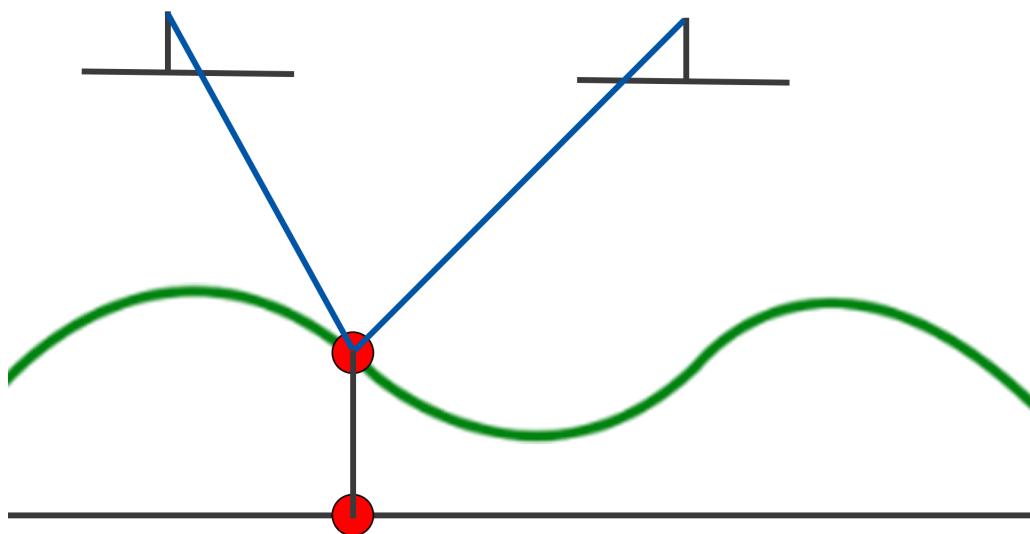


Рис. 2.

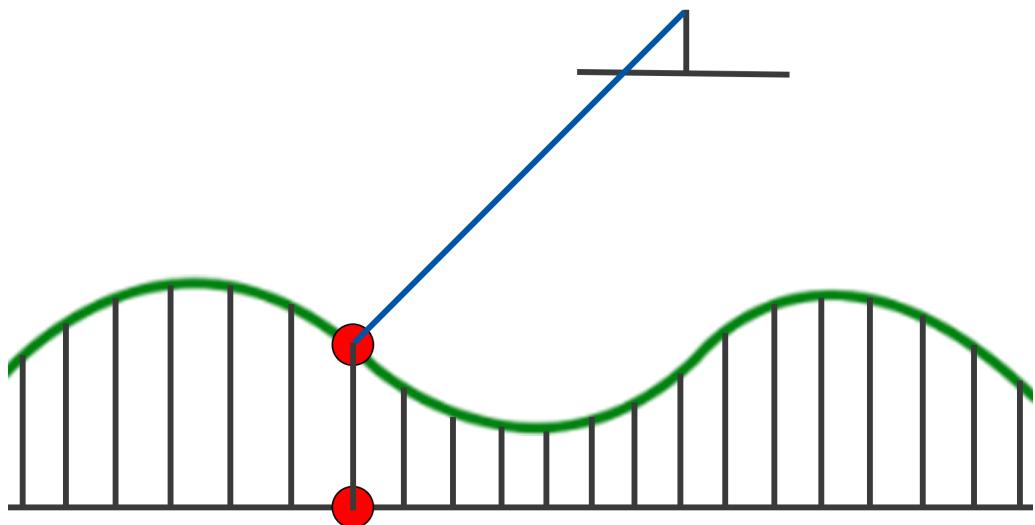


Рис. 3.

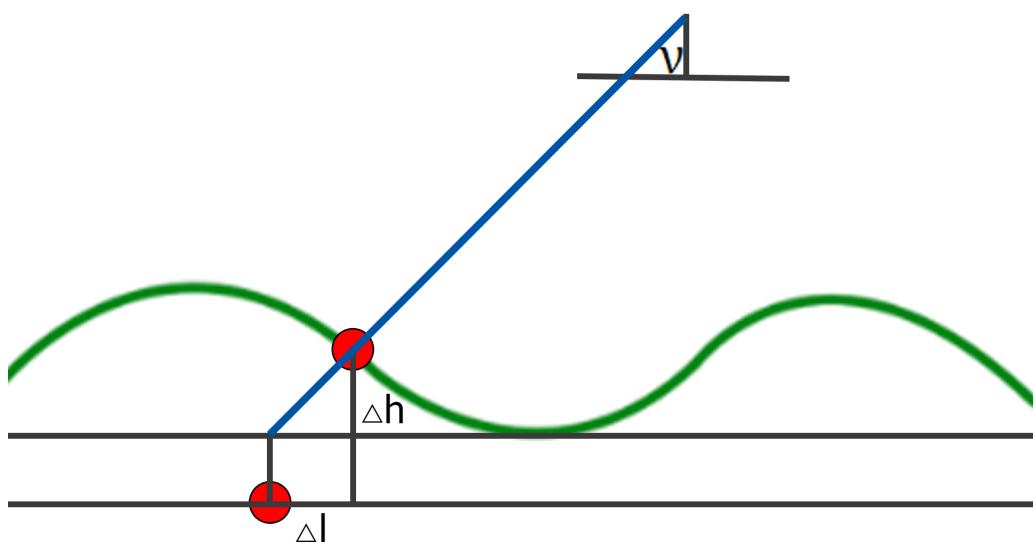


Рис. 4.

$$\Delta l = \Delta h \times \tan(\nu) \quad (4)$$

Если цифровая модель высот используется также для других целей, таких как генерация ортоизображений, требуются ортометрические высоты, которые связаны с геоидом. Космические данные часто доступны в

эллипсоидальных высотах WGS84, которые необходимо скорректировать относительно волнистости геоида. Не во всех районах мира волнообразные колебания геоида известны с достаточной точностью, поэтому в этом случае требуется самостоятельно определить вертикальные контрольные точки для данной области, что можно сделать с помощью ортометрической высоты.

## 1.4. Исажение изображений

Источник искажений может быть связан с двумя общими категориями: наблюдателем или системой сбора данных (платформа и датчик наблюдения) и наблюдаемым (атмосфера и Земля). В дополнение к этим искажениям необходимо учитывать деформации, связанные с проекцией карты, поскольку рельеф местности и большинство приложений конечных пользователей ГИС обычно представлены и выполняются на упомянутых топографических картах.

Искажения, вызванные платформой, в основном связаны с изменением эллиптического движения вокруг Земли, которое вызвано в первую очередь изменением земной гравитации. В зависимости от времени захвата и размера изображения изменение эллиптического движения оказывает различное влияние на искажение изображения. Можно перечислить следующие негативные эффекты, связанные с этим:

- 1) перепады высот в сочетании с фокусным расстоянием, плоскостью и рельефом Земли изменяют расстояние между пикселями;
- 2) изменения ориентации осей меняют ориентацию и форму изображений;
- 3) изменения скорости создают разрывы / перекрытия линий в изображениях.

Искажения, вызванные датчиком захвата изображения:

- 1) отклонения в параметрах калибровки, таких как фокусное расстояние и мгновенное поле зрения (IFOV), могут вызывать искажения;
- 2) панорамное искажение в сочетании с системой наклонного просмот-

ра, кривизна Земли и топографический рельеф изменяют положение пикселя вдоль столбца.

Искажения, вызванные Землей:

- 1) вращение создает боковые смещения в направлении столбца между линиями изображения в зависимости от широты;
- 2) кривизна создает изменения в интервалах пикселей изображения.

## Глава 2. Распределение вычислительных задач

Алгоритмы, используемые для вычисления карты рельефов с помощью спутниковых стереоснимков, требуют большой вычислительной мощности для своей полноценной работы. Сами изображения, которые представляются спутниковыми датчиками, не могут быть обработаны с использованием оперативной памяти стандартного Android смартфона. На момент написания этой работы платформа Android позволяет использовать на большей части устройств максимально лишь 32 мегабайта оперативной памяти для одного приложения. Поэтому было принято решение использовать клиент-серверную модель Android приложения.

Такой подход позволяет переложить основные вычислительные задачи на мощную серверную машину. Пользователь же получает результат в виде полноценных 3D моделей, описывающих рельеф необходимой местности.

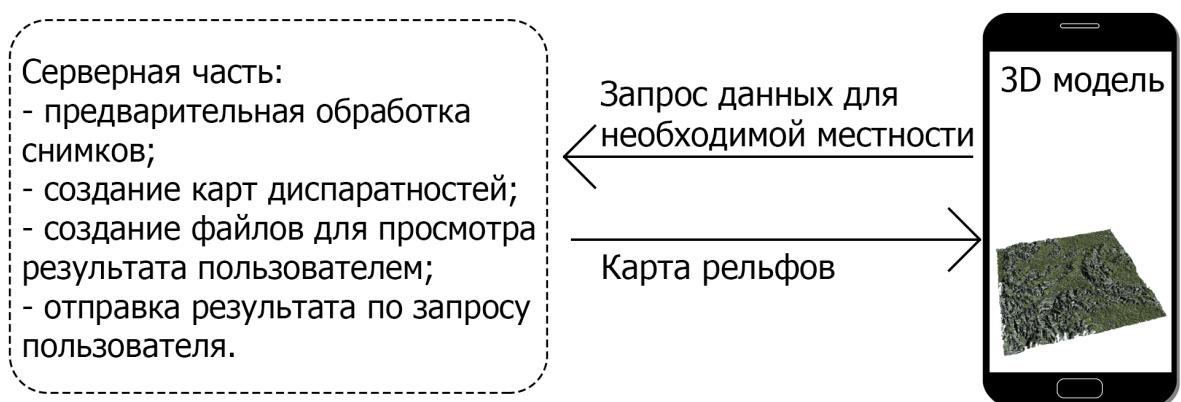


Рис. 5. Клиент-серверный подход для текущей задачи

Основные алгоритмы, которые были реализованы в рамках данной работы, написаны на языке программирования C++. Это дает возмож-

ность встраивать их внутрь приложений для мобильных платформ Android и iOS с незначительными изменениями. Поэтому при должном уровне развития оперативной памяти мобильных устройств в будущем можно будет рассматривать другие модели приложений, используя прежний код.

## Глава 3. Выбор спутниковых данных

Можно выделить несколько основных компаний, занимающихся поставками спутниковых изображений высокого разрешения: GeoEye, DigitalGlobe, Astrium, RapidEye, Indian Remote Sensing satellites и т.д.. После сравнения предоставляемых ими продуктов было решено использовать для целей данной задачи продукты компании DigitalGlobe. Эта компания является ведущей в области спутниковых снимков высокого разрешения. Ее ключевыми преимуществами перед другими компаниями являются:

- 1) высокое разрешение снимков;
- 2) большое количество предоставляемой информации о характеристиках камеры;
- 3) широкий диапазон продуктов.

DigitalGlobe предоставляет изображения со спутников QuickBird и трех спутников WorldView. В базовую серию продуктов DigitalGlobe входят следующие спутниковые изображения:

- 1) Basic (1B) Imagery;
- 2) Standard (2A) Imagery;
- 3) OrthoReady (2A) Imagery;
- 4) Basic (1B) Stereo Imagery;
- 5) OrthoReady (2A) Stereo Imagery.

Так как наиболее выгодным решением для моего проекта является использование готовых стереоизображений, для дальнейшего сравнения и рассмотрения были выделены последние два вида снимков: Basic Stereo Imagery и OrthoReady Stereo Imagery. К первому виду снимков была применена грубая цифровая модель рельефа со стороны поставщика, чтобы нормализовать рельеф относительно эталонного эллипсоида. Но степень нормализации этих снимков относительно невелика, поэтому, хотя этот продукт имеет поправки к местности, он не считается орторектифицированным. OrthoReady Stereo Imagery проецируются с учетом средней базовой высоты снимаемой территории, поэтому абсолютная горизонтальная точ-

ность может смещаться от ее истинного местоположения, особенно в областях с высоким уровнем рельефа. Такое смещение может вызвать большое количество ошибок во время предварительной обработки данных. Поэтому было решено использовать Basic Stereo Imagery. Пример спутникового снимка, используемого в данной работе, приведен на рисунке 6.



Рис. 6. Рио-де-Жанейро, Бразилия

# Глава 4. Алгоритм построения карты рельефов

## 4.1. Аффинное эпиполярное выравнивание

Пара стереоизображения – это два изображения, полученные из двух разных точек зрения для одной и той же сцены. Проекции точки, принадлежащей сцене, на правое и левое изображения используются для расчета диспаритетности и в дальнейшем определения глубины. Когда положение камеры изменяется только горизонтально, и оптические оси располагаются параллельно, корреспондирующая эпиполярная линия для двух изображений будет лежать на одной горизонтальной линии сканирования. Соответствующие друг другу точки нужно будет искать вдоль одной прямой, что существенно облегчает задачу.

Алгоритмы для 3D реконструкции основаны на последовательности снимков, которая может быть использована без дополнительной обработки только в идеальном случае. Для того, чтобы эпиполярные линии лежали на одних и тех же горизонтальных строках, необходимо трансформировать исходные изображения. Такая трансформация называется ректификацией.

В данной работе были протестированы два алгоритма ректификации стереоизображений:

- 1) использование гомографии для проецирования стереоизображений на общую плоскость;
- 2) аффинное эпиполярное выравнивание.

После тестирования обоих алгоритмов выбор был сделан в сторону аффинного эпиполярного выравнивания.

Аффинное эпиполярное выравнивание выполняется в 3 этапа:

- 1) расчет наклона эпиполярных линий и поворот их на такой угол, чтобы они были горизонтальными. Для этого используется сопоставление изображений для решения аффинной фундаментальной матрицы. С помощью решения будут получены эпиполярные линии и мат-

- рицы вращения;
- 2) выравнивание и масштабирование по оси  $Y$  правого изображения в соответствии с левым изображением;
  - 3) выравнивание и масштабирование по оси  $X$  обоих изображений.

Аффинная эпиполярная матрица (5) отличается от регулярной фундаментальной матрицы тем, что ее эпиполы находятся на бесконечности. Это делает верхнюю левую  $2 \times 2$  область матрицы нулевой. Только остальные 5 элементов содержат какую-либо информацию. Для нахождения этих элементов рассматриваем ограничения (6), (7).

$$F_a = \begin{pmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{pmatrix} \quad (5)$$

$$x_r^T \times F \times x_l = 0 \quad (6)$$

$$ax_r + by_r + cx_l + dy_l + e = 0 \quad (7)$$

Последняя строка и столбец аффинной фундаментальной матрицы представляют собой эпиполы левого и правого изображений соответственно:

$$F_a \times S_l = 0, S_l = (-d, c, 0)^T \quad (8)$$

$$F_a^T \times S_r = 0, S_r = (-b, a, 0)^T \quad (9)$$

Можно использовать их для нахождения матриц вращения (11), (12).

$$h_l = ||S_l||, h_r = ||S_r|| \quad (10)$$

$$Rot_l = \begin{pmatrix} \frac{-d}{h_l} & \frac{c}{h_l} \\ \frac{-c}{h_l} & \frac{-d}{h_l} \end{pmatrix} \quad (11)$$

$$Rot_r = \begin{pmatrix} \frac{-b}{h_r} & \frac{a}{h_r} \\ \frac{-a}{h_r} & \frac{-b}{h_r} \end{pmatrix} \quad (12)$$

Теперь переходим ко второму шагу – нахождение масштабирования и смещения по оси  $Y$  для правого изображения. Рассмотрим матрицы Left и Right.

$$Left = \begin{pmatrix} \frac{-d}{h_l} & \frac{c}{h_l} & 0 \\ \frac{-c}{h_l} & \frac{-d}{h_l} & 0 \\ 0 & 0 & 1 \end{pmatrix}, Right = \begin{pmatrix} \frac{-b}{h_r} & \frac{a}{h_r} & 0 \\ \frac{-a}{h_r} & \frac{-b}{h_r} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (13)$$

Далее введем в рассмотрение матрицы  $A$  и  $B$  размерности  $n \times 3$  и  $n \times 1$  соответственно.

$$T(i) = (Right \times \begin{pmatrix} x_r(i) \\ y_r(i) \\ 0 \end{pmatrix})^T, i = 1..n \quad (14)$$

$$A(i) = \begin{pmatrix} T(i, 2) & T(i, 3) \end{pmatrix}, i = 1..n \quad (15)$$

$$P(i) = (Left \times \begin{pmatrix} x_l(i) \\ y_l(i) \\ 0 \end{pmatrix})^T, i = 1..n \quad (16)$$

$$B(i) = \begin{pmatrix} P(i, 2) \end{pmatrix}, i = 1..n \quad (17)$$

Масштабирование и смещение можно получить, решив переопределенную систему вида (18). Решение производится методом наименьших квадратов.

$$A \times \begin{pmatrix} y_s & y_0 \end{pmatrix} = B \quad (18)$$

$$Right = \begin{pmatrix} y_s \frac{-b}{h_r} & y_s \frac{a}{h_r} & 0 \\ y_s \frac{-a}{h_r} & y_s \frac{-b}{h_r} & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (19)$$

Полученное значение  $y_s$  отражает масштабирование по оси  $Y$ , а  $y_0$  - смещение по оси  $Y$ .

На третьем шаге вычисляется масштабирование, наклон и смещение по оси  $X$  для обоих изображений.

$$B'(i) = \left( P(i, 1) \right), i = 1..n \quad (20)$$

$$T \times \begin{pmatrix} k_1 & k_2 & k_3 \end{pmatrix} = B', \quad (21)$$

где  $k_1, k_2, k_3$  - коэффициенты растяжения, наклона и смещения по оси  $X$  соответственно.

$$Left = Left \times \begin{pmatrix} 1 & \frac{-k_2}{2} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (22)$$

$$Right = Right \times \begin{pmatrix} k_1 & \frac{k_2}{2} & k_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (23)$$

В итоге были получены матрицы (22) и (23), описывающие преобразования для левого и правого изображений соответственно.

На входе у данного алгоритма было два снимка, расположение которых относительно Земли проиллюстрировано на рисунке 7.



Рис. 7. Исходное расположение входных данных стереоснимков относительно Земли

На выходе были получены результаты, изображенные на рисунке 8. Красными линиями обозначены эпиполярные линии. Как можно видеть, эпиполярные линии располагаются на одних и тех же горизонтальных стро-  
ках для левого и правого изображений.

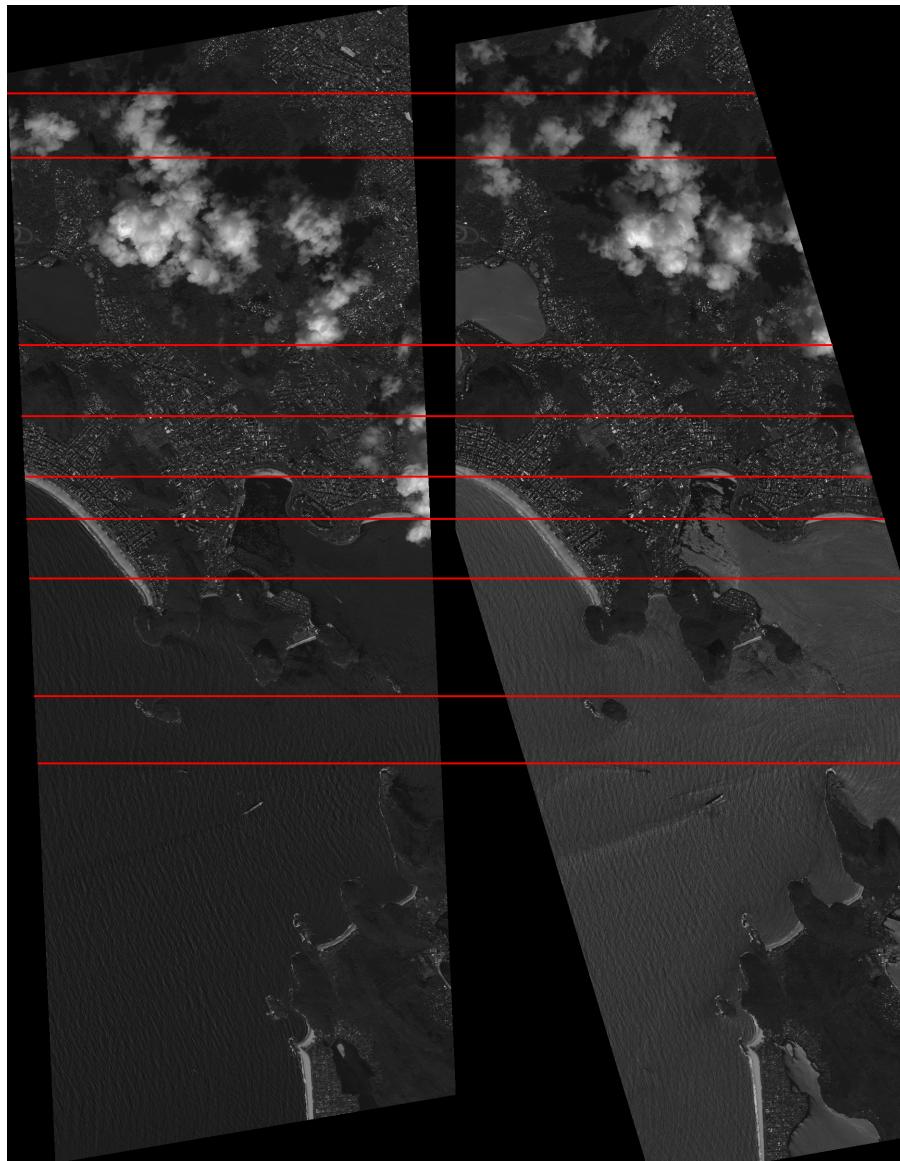


Рис. 8. Результаты, полученные после использования аффинного эпиполярного выравнивания

#### 4.2. Blob detection с помощью использования фильтра LoG

Пара спутниковых снимков была сделана с небольшим временным интервалом (в данном конкретном случае он составляет около 50 секунд) и под разными углами. Поэтому, помимо аффинных выравниваний, нужно произвести работу с освещением на спутниковых изображениях, чтобы скорректировать световые перепады. Это в свою очередь повысит точность

функции стереосопоставления при построении карты диспаратностей. Для данной задачи было решено использовать выделение и сопоставление blob features на изображениях.

Одним из наиболее удобных blob детекторов является оператор (или сверточное ядро) лапласиана от гауссиана.

$$L(x, y, \sigma) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (24)$$

В формуле (24)  $\sigma$  - масштаб фильтра,  $x, y$  - координаты пикселя изображения.

Маска фильтра LoG соответствует круговой центральной пространственной структуре с положительными весами в центральной области и отрицательными весами в окружающей кольцевой структуре. Таким образом, функция дает максимальную реакцию, если применить к окрестности изображения, которая содержит аналогичную (грубо круговую) blobe структуру в соответствующем масштабе. Таким образом, при поиске экстремумов масштабного пространства LoG мы можем, обнаружить blobe структуры.

### 4.3. Построение карты диспаратностей с помощью алгоритма Semi-Global Matching

Semi-Global Matching или SGM – это метод, разработанный Heiko Hirschmüller. Данный метод был впервые опубликован в 2005 году в статье [14]. Для того, чтобы понять особенность этого метода, нужно более ярко осветить тему алгоритмов стереосопоставления.

Существующие стереоалгоритмы могут быть разделены на две группы: локальные и глобальные методы. Локальные методы пытаются найти оптимальные диспаратности для небольших регионов изображений, например, одной строки, что может привести к разрыву между различными регионами (например, известные полосовые эффекты в динамическом программировании). Глобальные подходы оптимизируют все диспаратности сразу, обеспечивая лучшую общую производительность по сравнению с

локальными алгоритмами. Однако требуемое время вычисления для таких алгоритмов значительно больше.

SGM, как полу-глобальный метод, включает в себя преимущества обеих групп, обеспечивая относительно низкую вычислительную сложность и относительно высокое качество результатов. Его можно разбить на несколько отдельных фаз, которые будут кратко объяснены ниже.

Возьмем максимальное значение смещения  $d_{max}$  и пару ректифицированных изображений. Для каждого пикселя на позиции  $(x, y)$  основного изображения согласованный пиксель сопоставляющего изображения находится на позиции между  $(x, y)$  и  $(x - d_{max}, y)$  (предполагается, что левое изображение выбрано в качестве базового изображения).

Выбранные смещения  $d$  оцениваются функцией стоимости, например, с помощью формулы (25), которая просто вычисляет абсолютную разницу уровней серого между пикселям базового изображения и пикселям сопоставляемого изображения.

$$C(d) = |I_b(x, y) - I_m(x - d, y)| \quad (25)$$

Целевая функция  $E(D)$  из уравнения (26), которая назначает затраты для карты диспаритетностей  $D$ , должна быть сведена к минимуму для достижения оптимальных диспаритетностей.

$$E(D) = \sum_{d \in D} (C(d) + \sum_{d' \in N(d)} P_1 T[|d - d'| = 1] + \sum_{d'' \in N(d)} P_2 T[|d - d''| > 1]), \quad (26)$$

где  $T[arg]$  равняется единице, если его аргумент правда, и нулю, если его аргумент ложь. Через  $N(d)$  обозначены соседи  $d$ . Функция суммирует начальные затраты  $C(d)$  для выбранных диспаритетностей и два дополнительных штрафа, которые зависят от разницы с соседними диспаритетностями. Если они отличаются на 1, применяется небольшой штраф  $P_1$ . Для различий больших 1 применяется большой штраф  $P_2$ . Выбор меньшего  $P_1$  позволяет адаптироваться к наклонным поверхностям. Большие скачки в диспаритетностях возможны только в случае, если  $C(d'') - C(d) > P_2$ .

Минимизация  $E(D)$  двумерным образом была бы очень дорогостоящей. Поэтому SGM упрощает оптимизацию путем перемещения одномер-

ных путей и накладывает ограничения в связи с явным видом направлений. Этот подход требует второго этапа, называемого агрегации затрат. Уравнение (27) описывает эту процедуру для горизонтального пути движения слева направо в произвольной строке изображения  $y$ .

$$\begin{aligned} E(x, y, d) = & C(x, y, d) + \min [ \\ & E(x - 1, y, d), \\ & E(x - 1, y, d - 1) + P_1, \\ & E(x - 1, y, d + 1) + P_1, \\ & \min_i E(x - 1, y, d + i) + P_2 ] \end{aligned} \quad (27)$$

Рекурсивная функция  $E$  определяет вторую меру затрат, назначенную каждому пикслю в определенном месте  $(x, y)$  со значением диспаратности  $d$ . Поэтому стоимость одного пикселя требует всех значений стоимости предыдущих пикселей. Это позволяет вычислить затраты через обход пути слева направо.

Этот обход эффективно эмулирует ограничения, введенные исходной целевой функцией  $E(D)$ . Пикслю может быть присвоена более низкая стоимость, если он адаптирует его несоответствие к соседям. Однако это справедливо только для одного направления, и поэтому необходимо рассмотреть несколько различных ориентированных путей. Согласно [14] минимальное количество путей должно быть не менее 8 (т.е. два пути для горизонтальных, вертикальных и обоих диагональных направлений).

Конечные (сглаженные) затраты для каждого пикселя и каждая диспаратность  $S(x, y, d)$  получается суммированием затрат  $E_r(x, y, d)$  путей во всех направлениях  $r$ .

$$S(x, y, d) = \sum_r E_r(x, y, d) \quad (28)$$

На последнем этапе вычисляется диспаратность, которая минимизирует стоимость для каждого пикселя. С помощью описанного метода диспаратность присваивается каждому пикслю базового изображения незави-

сими от окклюзий. Однако существуют методы для уточнения (увеличения точности) значений диспаратностей.

Обнаружение окклюзий достигается за счет дополнительного вычисления диспаратностей для согласуемого изображения, за которым следует попиксельная проверка согласованности. Если диспаратности различаются, соответствующее значение в исходном изображении считается недействительным.

Достаточно простое уточнение карты диспаратностей может быть получено путем использования небольшого медианного фильтра. Таким образом, удаляются одиночные выбросы, а края изображения могут быть улучшены.

#### 4.4. Взвешенный медианный фильтр

Медианный фильтр широко используется при уточнении карт диспаратностей. Но в большинстве случаев его использование может спровоцировать нечеткие края объектов. В целях сокращения эффекта размытия было решено использовать его модификацию - взвешенный медианный фильтр.

Взвешенный медианный фильтр - это оператор, который заменяет текущий пиксель взвешенной медианой соседних пикселей в локальном окне. При обработке пикселя  $p$  в изображении  $I$  рассматриваются только пиксели в локальном окне  $R(p)$  радиуса  $r$ , центрированном по  $p$ . В отличие от обычного медианного фильтра, взвешенный медианный фильтр каждый пиксель  $q \in R(p)$  связывает с его весом  $w(p, q)$ , отражающим близость пикселей. Вес  $w(p, q)$  между двумя пикселями  $p$  и  $q$  находится по формуле (29).

$$w(p, q) = \exp\left(-\frac{|I(p) - I(q)|^2}{2\sigma^2}\right) \quad (29)$$

В формуле (29)  $\sigma$  - параметр фильтра,  $I(p)$ ,  $I(q)$  - значения пикселей  $p$  и  $q$  соответственно.

Обозначив  $I(q)$  как значение в пикселе  $q$  на изображении  $I$  и  $n$  как

число пикселей в  $R(p)$ , можно описать каждый пиксель из  $R(p)$  следующим образом:  $\{I(p), w(p, q)\}$ . После сортировки значений в порядке возрастания взвешенный медианный оператор возвращает для пикселя  $p$  новый пиксель  $p*$ , а  $I(p)$  заменяется на  $I(p*)$  по принципу, описанному в формуле (30).

$$p^* = \min(k), \text{ где } \sum_{q=1}^k w(p, q) \geq \sum_{q=1}^n \frac{1}{2} w(p, q) \quad (30)$$

## 4.5. Триангуляция

Рассмотрим геометрию проекции местности на плоскости снимков (рисунок 9).

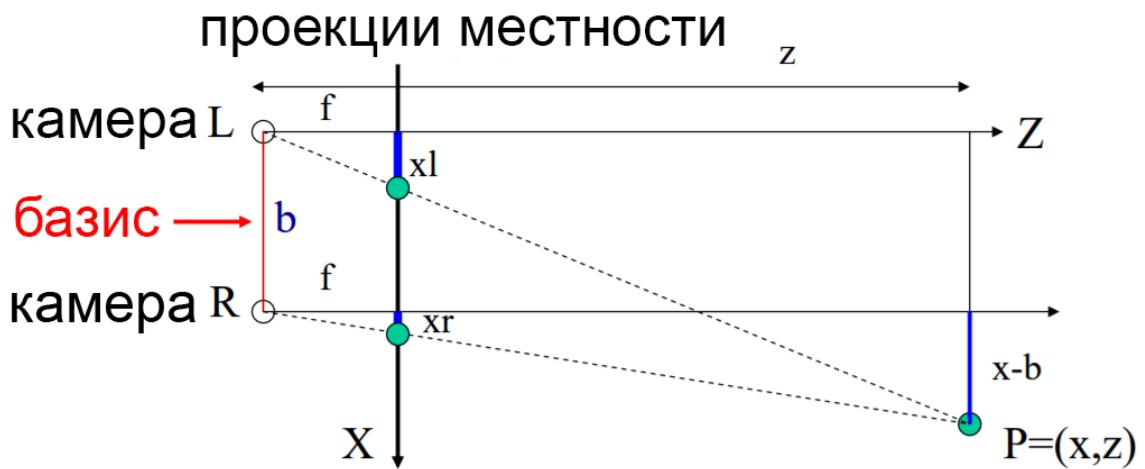


Рис. 9.

Для стереокамер с параллельными оптическими осями с фокусным расстоянием  $f$  и длинной базиса  $b$  можно найти расположение точки поверхности Земли в трехмерном пространстве, имея карту диспаратностей с помощью формулы (31).

$$z = f \times \frac{b}{(x_l - x_r)} = f \times \frac{b}{d} \quad (31)$$

Этот метод определения глубины с помощью диспаратности  $d$  называется

триангуляцией. После выполнения этого шага в ходе работы была получена карта глубины рассматриваемой местности (рисунок 10).

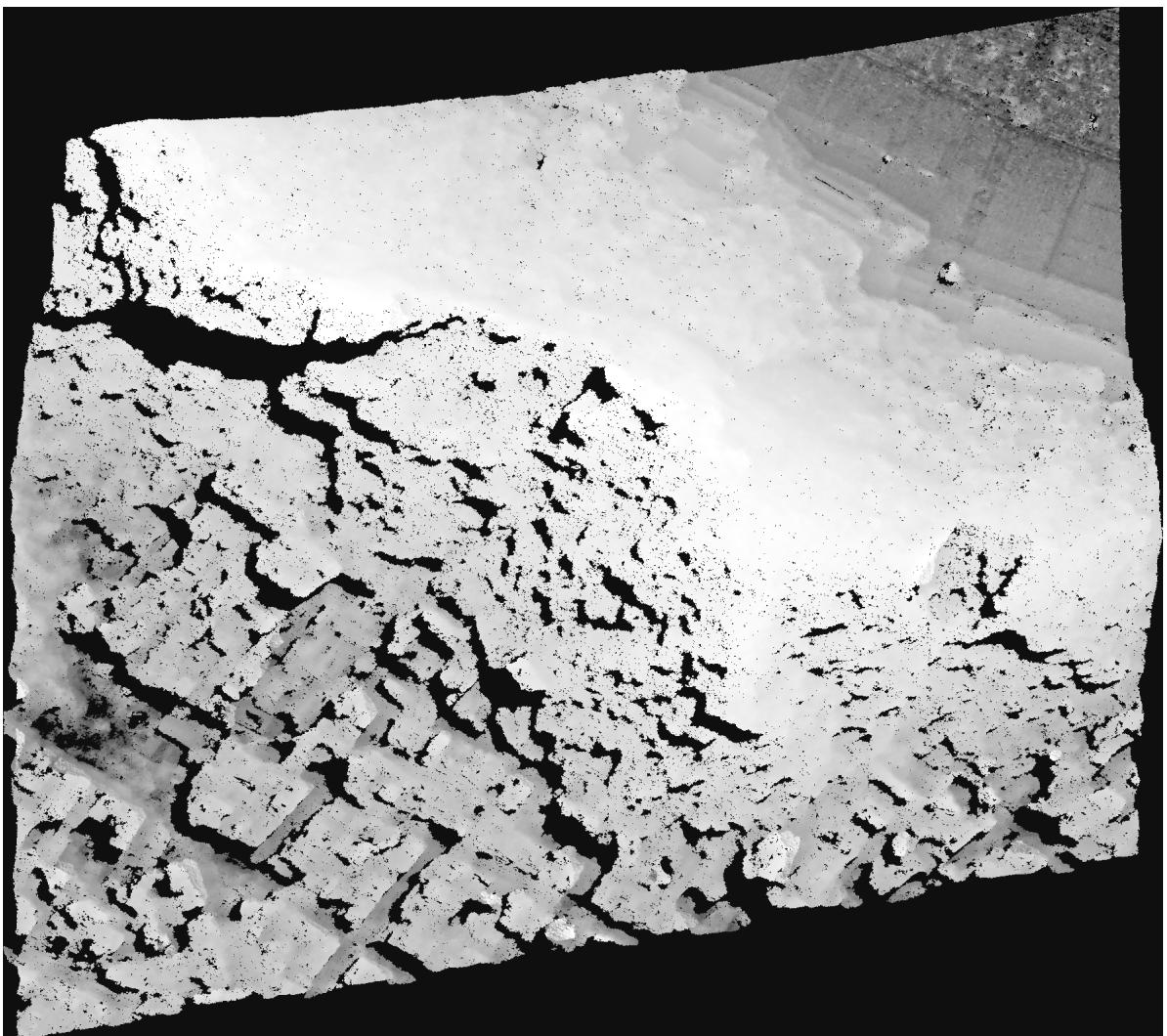


Рис. 10. Результат триангуляции

#### **4.6. Вывод информации пользователю Android приложения**

Последним шагом алгоритма является вывод данных карты рельефов местности пользователю Android приложения. Полученная на предыдущем шаге карта глубины помогает получить представление о данной местности. Поэтому можно использовать ее в качестве данных для вывода пользователю (рисунок 11).

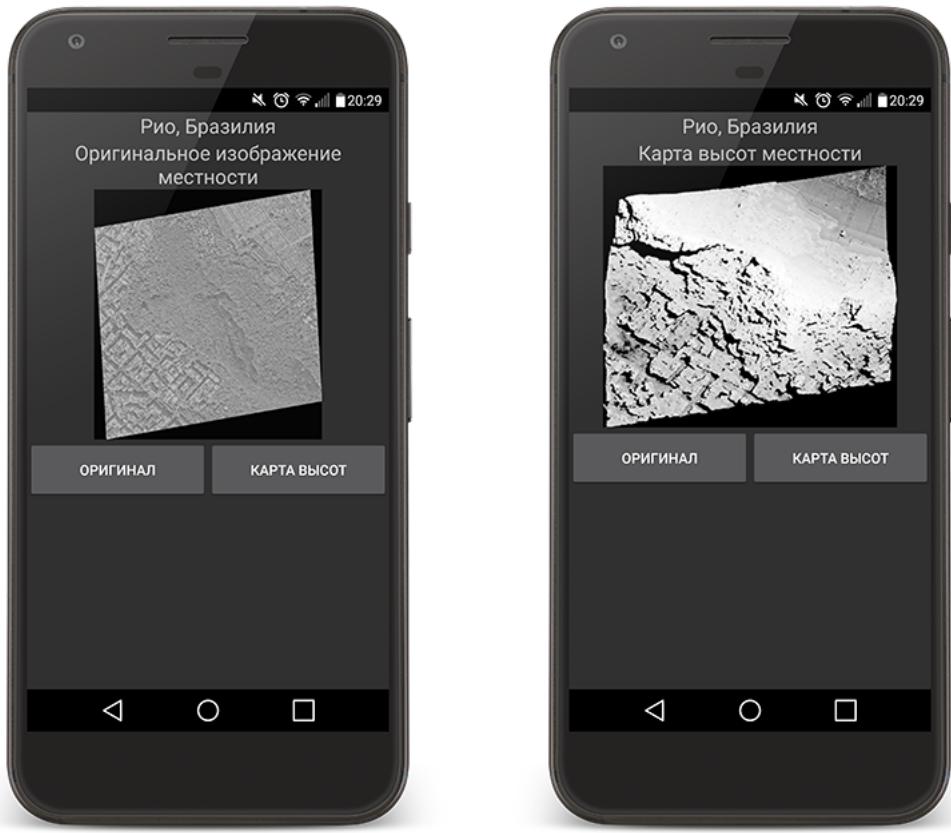


Рис. 11. Пример вывода результатов пользователю Android приложения

Но такое представление имеет ряд минусов, среди которых:

- 1) измерение, отвечающее за высоты, представлено черно-белым цветом, что является не очень наглядным способом конечного отображения данных о высоте;
- 2) интерактивность между пользователем и полученными результатами заключается только в возможности приближать и отдалять интересующие области изображения.

Поэтому было решено использовать 3D модели для представления конечного результата пользователю. Среди наиболее популярных 3D форматов, использующихся в Android приложениях, можно выделить: *stl*, *obj* и *3ds*. После сравнения результатов, получаемых каждым из этих форматов, было решено использовать *3ds*. Результаты представления карты рељефов местности с использованием формата *3ds* на Android приложении

представлены на рисунке 12.



Рис. 12. Использование формата *3ds* для визуализации

# Глава 5. Экспериментальные результаты и сравнения

## 5.1. Сравнение аффинного эпиполярного выравнивания и преобразования с использованием гомографии

На рисунках 13, 14 приведены результаты тестирования двух алгоритмов ректификации стереоизображений совместно с локальным алгоритмом стереосопоставления снимков для данных, которые были получены в безоблачных условиях.

На картинке 13.a был использован метод проецирования изображений на общую плоскость с использованием гомографии. На картинке 13.b - алгоритм аффинного эпиполярного выравнивания.

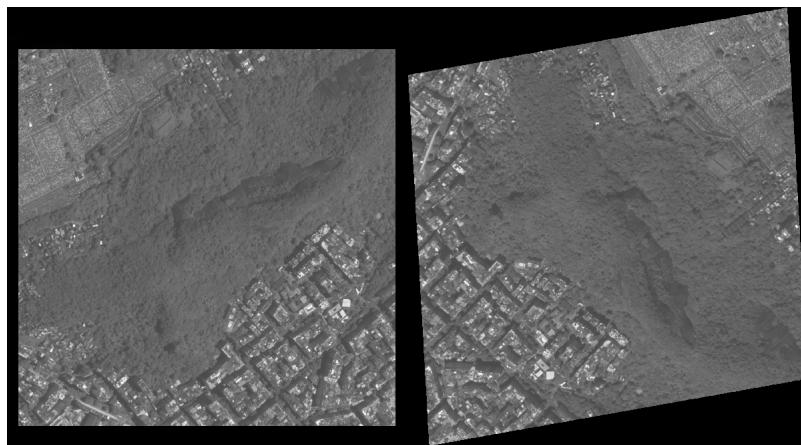


Рис. 13. а) ректификация с помощью гомографии б) ректификация с помощью аффинного эпиполярного выравнивания

На рисунке 14 белым цветом выделены пиксели, которые удалось распознать с помощью локального алгоритма сопоставления стереоснимков. На левом изображении количество распознанных пикселей 61%, на правом 68%.

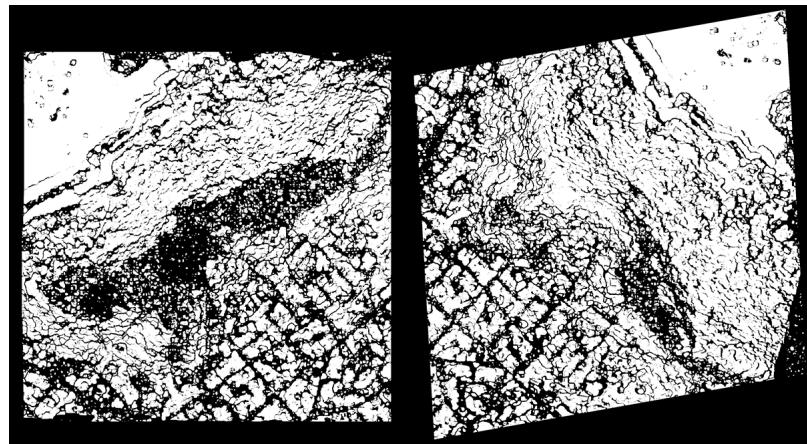


Рис. 14. а) ректификация с помощью гомографии б) ректификация с помощью аффинного эпиполярного выравнивания

На рисунках 15, 16 приведены результаты тестирования двух алгоритмов ректификации стереоизображений совместно с локальным алгоритмом стереосопоставления снимков для данных, которые были получены в условиях небольшой облачности.

На картинке 15.а был использован метод проецирования изображений на общую плоскость с использованием гомографии. На картинке 15.б - алгоритм аффинного эпиполярного выравнивания.

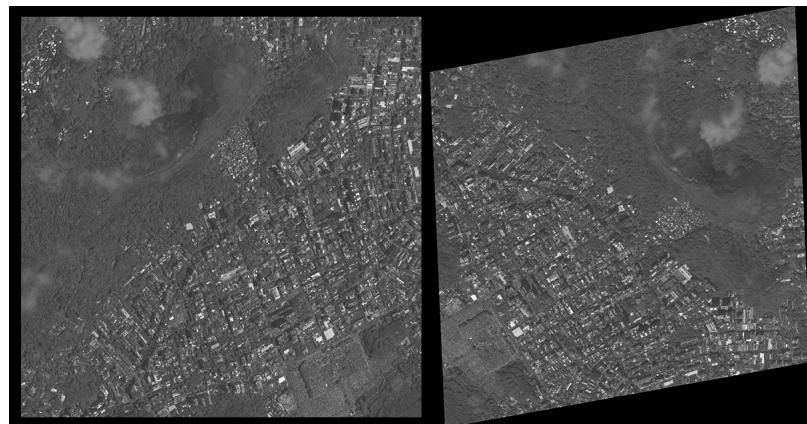


Рис. 15. а) ректификация с помощью гомографии б) ректификация с помощью аффинного эпиполярного выравнивания

На рисунке 16 белым цветом выделены пиксели, которые удалось распознать с помощью локального алгоритма сопоставления стереосним-

ков. На левом изображении количество распознанных пикселей 48%, на правом 53%.

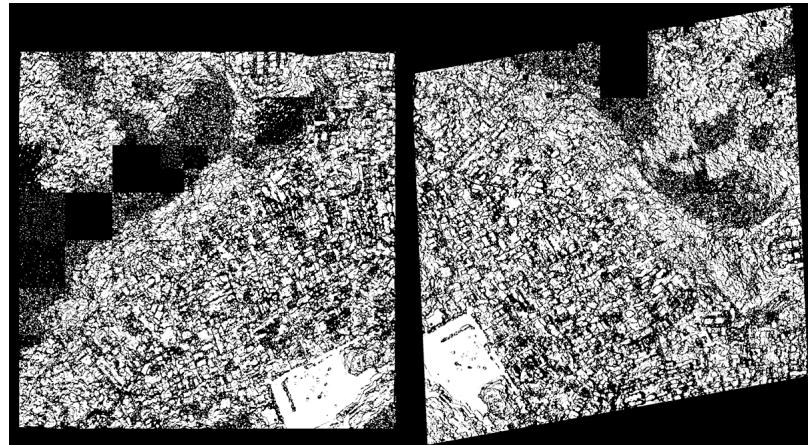


Рис. 16. а) ректификация с помощью гомографии б) ректификация с помощью аффинного эпиполярного выравнивания

На рисунках 17, 18 приведены результаты тестирования двух алгоритмов ректификации стереоизображений совместно с алгоритмом сопоставления стереоснимков Semi-Global для данных, которые были получены в безоблачных условиях.

На картинке 17.a был использован метод проецирования изображений на общую плоскость с использованием гомографии. На картинке 17.b - алгоритм аффинного эпиполярного выравнивания.



Рис. 17. а) ректификация с помощью гомографии б) ректификация с помощью аффинного эпиполярного выравнивания

На рисунке 18 белым цветом выделены пиксели, которые удалось распознать с помощью алгоритма сопоставления стереоснимков SGM. На левом изображении количество распознанных пикселей 87%, на правом 91%.

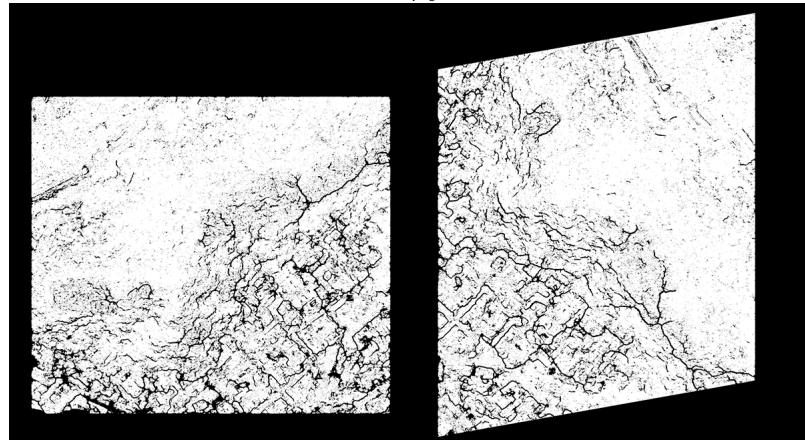


Рис. 18. а) ректификация с помощью гомографии б) ректификация с помощью аффинного эпиполярного выравнивания

На рисунках 19, 20 приведены результаты тестирования двух алгоритмов ректификации стереоизображений совместно с алгоритмом сопоставления стереоснимков Semi-Global для данных, которые были получены в условиях небольшой облачности.

На картинке 19.а был использован метод проецирования изображений на общую плоскость с использованием гомографии. На картинке 19.б - алгоритм аффинного эпиполярного выравнивания.

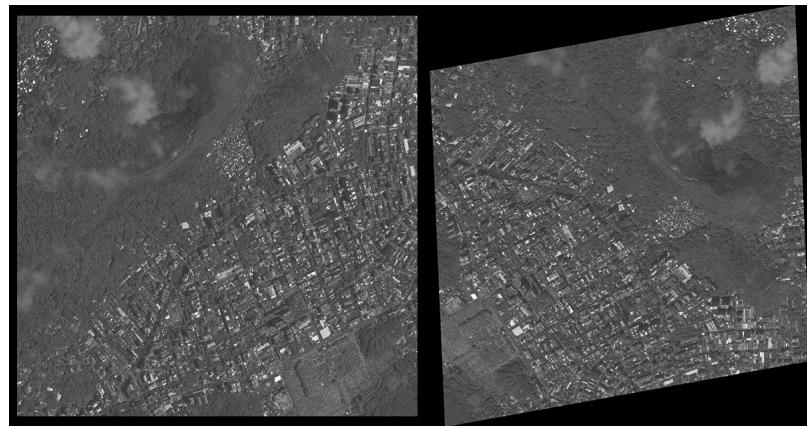


Рис. 19. а) ректификация с помощью гомографии б) ректификация с помощью аффинного эпиполярного выравнивания

На рисунке 20 белым цветом выделены пиксели, которые удалось распознать с помощью алгоритма сопоставления стереоснимков SGM. На левом изображении количество распознанных пикселей 84%, на правом 87%.

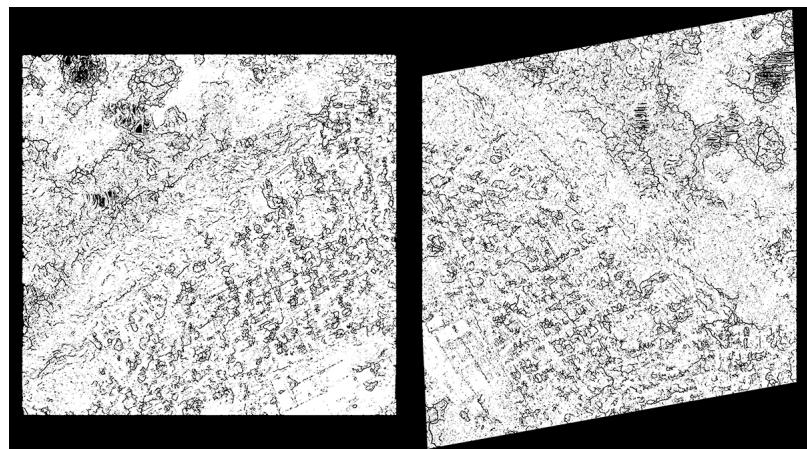


Рис. 20. а) ректификация с помощью гомографии б) ректификация с помощью аффинного эпиполярного выравнивания

Как видно из рисунков, использование аффинного эпиполярного выравнивания дает лучшие результаты совместно с локальным алгоритмом стереосопоставления по сравнению с использованием гомографии с тем же локальным алгоритмом. В случаях, когда на стереоизображениях есть области с зашумлением, это особенно хорошо видно.

Результаты использования алгоритма SGM на порядок лучше по сравнению с результатами использования локального алгоритма. Качество работы алгоритма SGM в среднем на 3-5% лучше при использовании его совместно с аффинным эпиполярным алгоритмом ректификации по сравнению с использованием SGM совместно с алгоритмом ректификации с использованием гомографии.

## 5.2. Использование LoG фильтра

В данной работе были произведены сравнения результатов для одного набора данных, полученных без использования фильтра LoG и результатов, полученных с использованием фильтра.

На рисунке 21 приведены результаты, полученные при использовании LoG фильтра совместно с локальным алгоритмом сопоставления стереоизображений и результаты, полученные без использования LoG фильтра в случае снимков, полученных в безоблачную погоду.

На картинке 21.*a* изображено рассматриваемое ректифицированное изображение. На картинке 21.*b* - результаты без использования фильтра LoG. На картинке 21.*c* - с использованием фильтра LoG. На картинках 21.*b* и 21.*c* белым цветом выделены пиксели, для которых удалось определить высоту.

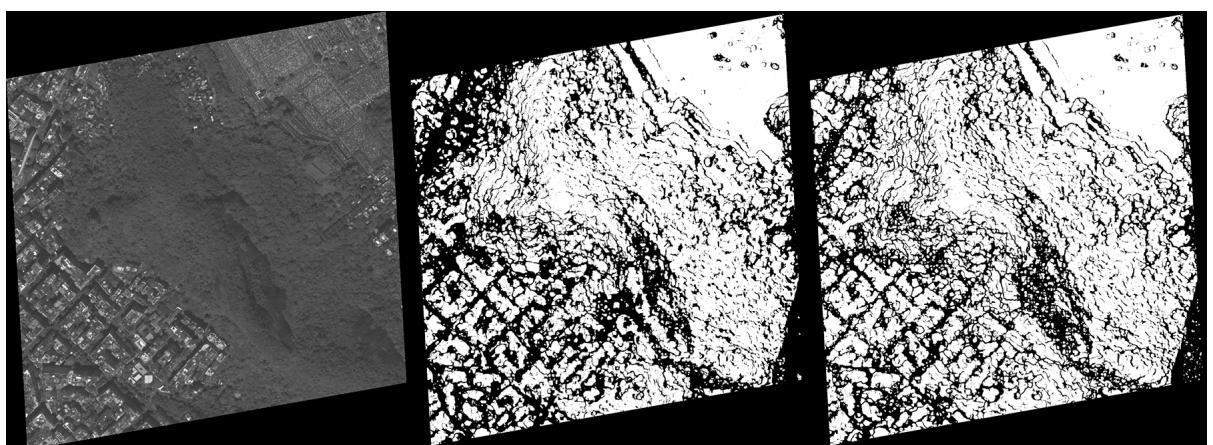


Рис. 21. а) ректифицированное изображение б) не используется фильтр LoG с) используется фильтр LoG

На рисунке 21 на среднем изображении количество распознанных пикселей 63%, на правом 68%.

На рисунке 22 приведены результаты, полученные при использовании LoG фильтра совместно с алгоритмом сопоставления стереоизображений SGM и результаты, полученные без использования LoG фильтра, в случае снимков, полученных в безоблачную погоду.

На картинке 22.*a* изображено рассматриваемое ректифицированное изображение. На картинке 22.*b* - результаты без использования фильтра LoG. На картинке 22.*c* - с использованием фильтра LoG. На средней и правой картинках белым цветом выделены пиксели, для которых удалось определить высоту.

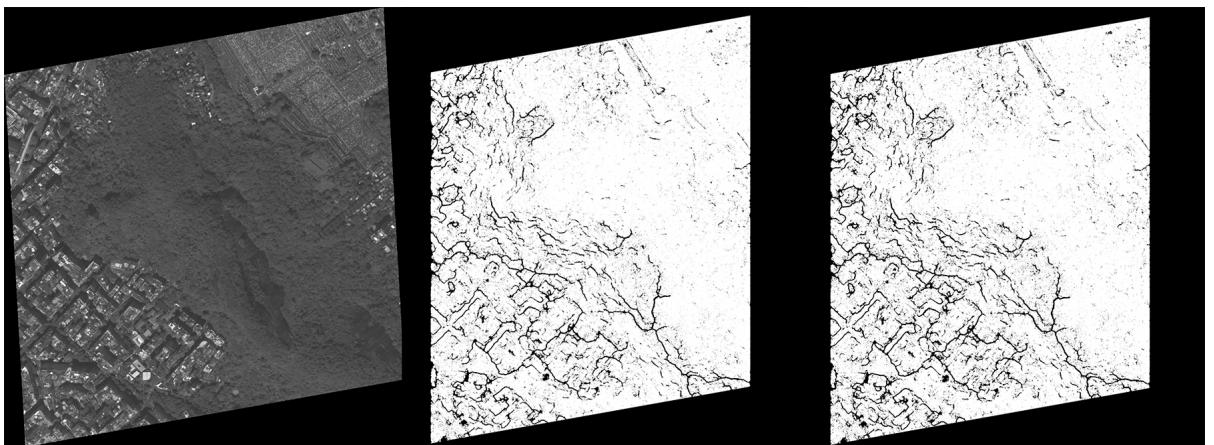


Рис. 22. а) ректифицированное изображение б) не используется фильтр LoG с) используется фильтр LoG

На рисунке 22 на среднем изображении количество распознанных пикселей 90%, на правом 91%.

На рисунке 23 приведены результаты, полученные при использовании LoG фильтра совместно с Semi-Global алгоритмом сопоставления стереоизображений и результаты, полученные без использования LoG фильтра, в случае снимков, полученных в условиях небольшой облачности.

На картинке 23.*a* изображено рассматриваемое ректифицированное изображение. На картинке 23.*b* - результаты без использования фильтра LoG. На картинке 23.*c* - с использованием фильтра LoG. На средней и

правой картинках белым цветом выделены пиксели, для которых удалось определить высоту.

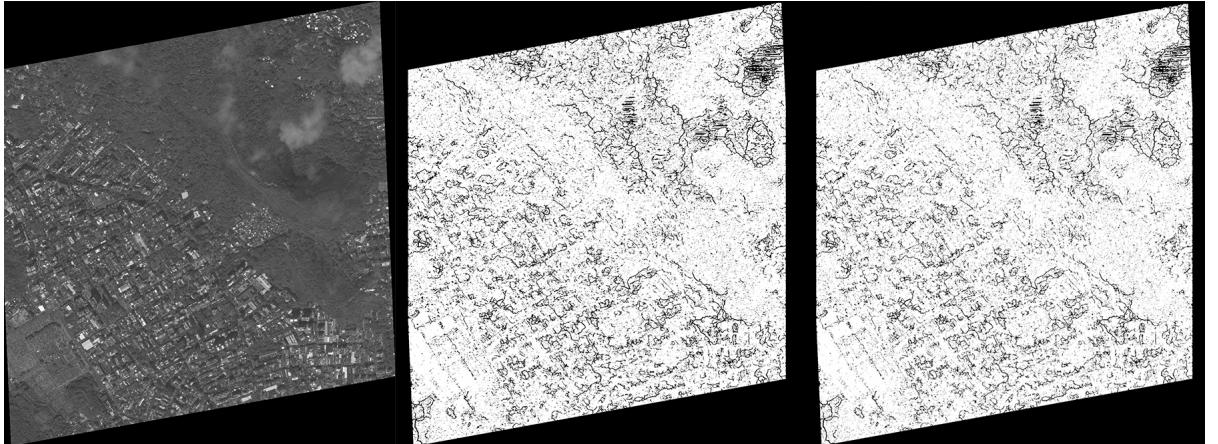


Рис. 23. а) ректифицированное изображение b) не используется фильтр LoG c) используется фильтр LoG

На рисунке 23 на среднем изображении количество распознанных пикселей 86%, на правом 87%.

Фильтр LoG улучшает конечный результат в среднем на 5%, если используется совместно с локальным алгоритмом стереосопоставления. Использование данного фильтра совместно с алгоритмом сопоставления стереоизображений SGM дает незначительные преимущества в пределах 1% конечного результата. Поэтому использование фильтра LoG совместно с алгоритмом SGM можно рассматривать как optionalное решение.

### 5.3. Сравнение локального алгоритма сопоставления и алгоритма сопоставления Semi-Global Matching

В ходе данной работы были произведены сравнения работы локального алгоритма стереосопоставления с работой алгоритма стереосопоставления Semi-Global Matching.

На рисунке 24 приведены результаты сравнения данных алгоритмов для случая снимков, сделанных в безоблачную погоду.

На картинке 24.*a* изображено рассматриваемое ректифицированное изображение. На картинке 24.*b* - результаты, полученные с помощью локального алгоритма. На картинке 24.*c* - результаты, полученные с помощью алгоритма SGM. На средней и правой картинках белым цветом выделены пиксели, для которых удалось определить высоту.

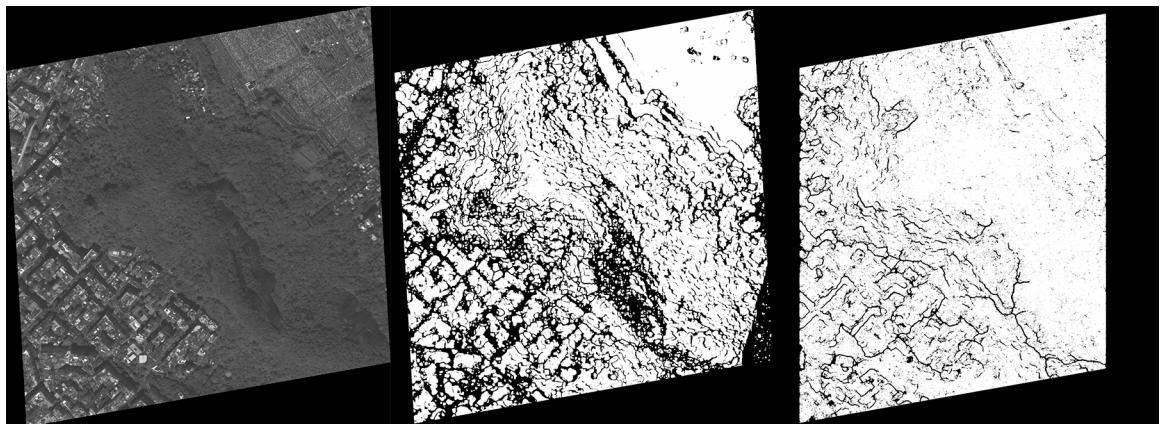


Рис. 24. а) ректифицированное изображение б) результат локального алгоритма с) результат алгоритма SGM

На рисунке 24 на среднем изображении количество распознанных пикселей 68%, на правом 91%.

На рисунке 25 приведены результаты сравнения данных алгоритмов для случая снимков, сделанных в условиях небольшой облачности.

На картинке 25.*a* изображено рассматриваемое ректифицированное изображение. На картинке 25.*b* - результаты, полученные с помощью локального алгоритма. На картинке 25.*c* - результаты, полученные с помощью алгоритма SGM. На средней и правой картинках белым цветом выделены пиксели, для которых удалось определить высоту.

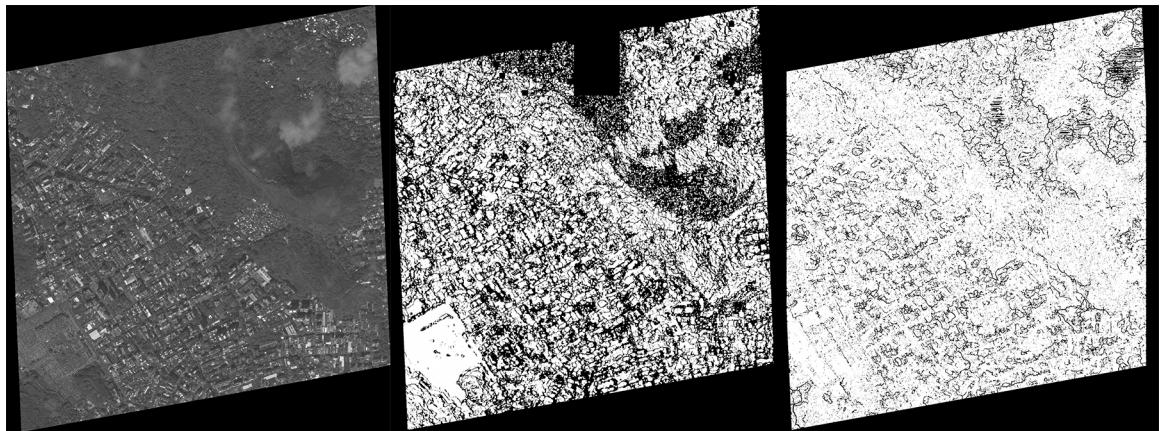


Рис. 25. а) ректифицированное изображение б) результат локального алгоритма с) результат алгоритма SGM

На рисунке 25 на среднем изображении количество распознанных пикселей 53%, на правом 87%.

Можно сделать вывод, исходя из экспериментальных результатов, что алгоритм сопоставления стереоснимков SGM может распознать на 23% больше пикселей по сравнению с локальным алгоритмом сопоставления стереоснимков для изображений, которые были сделаны в безоблачных условиях. И на 35% больше для изображений, которые были сделаны в условиях небольшой облачности.

#### 5.4. Сравнение 3D форматов, используемых в Android приложениях

Для вывода информации о карте рельефов рассматриваемой местности пользователю Android приложения было решено использовать 3D визуализацию. Наиболее распространенными 3D форматами, используемыми Android приложениями, являются форматы *stl*, *obj* и *3ds*.



Рис. 26. а) stl формат б) obj формат с) 3ds формат

На рисунке 26 показано представление полученной ранее карты рельефов в каждом из трех рассматриваемых форматов. Стоит заметить, что модель на картинке 26.b, соответствующая формату *obj*, имеет наименьшую возможность детализации по сравнению с двумя другими.

Также следует рассмотреть объем памяти, занимаемый каждым из форматов:

- 1) stl файл: 11 Mb;
- 2) obj файл: 4.9 Mb;
- 3) 3ds файл: 1.4 Mb.

Исходя из вышеперечисленного, наиболее выгодным в использовании для данной задачи является формат *3ds*.

## 5.5. Выводы из экспериментальных данных

После проведения экспериментальных тестов, необходимых для анализа задачи построения карты рельефов местности и ее вывода пользователю Android приложения, был сделан ряд заключений:

- 1) если рассматривать работу алгоритма аффинного эпиполярного выравнивания совместно с алгоритмом сопоставления стереоснимков SGM, то использование данного алгоритма ректификации увеличивает количество распознанных пикселей в среднем на 3% по сравнению с использованием алгоритма ректификации проецирования снимков на общую плоскость с использованием гомографии;
- 2) использование фильтра LoG совместно с алгоритмом SGM увеличивает количество распознанных пикселей в пределах 1%;
- 3) использование алгоритма сопоставления стереоснимков SGM повышает количество распознанных пикселей на 23%-35% по сравнению с локальным алгоритмом сопоставления.

## Заключение

В ходе проделанной работы был реализован алгоритм создания карты рельефов местности по спутниковым изображениям с возможностью последующего вывода результатов пользователю Android приложения. Финальный вид алгоритма представлен на рисунке 27.

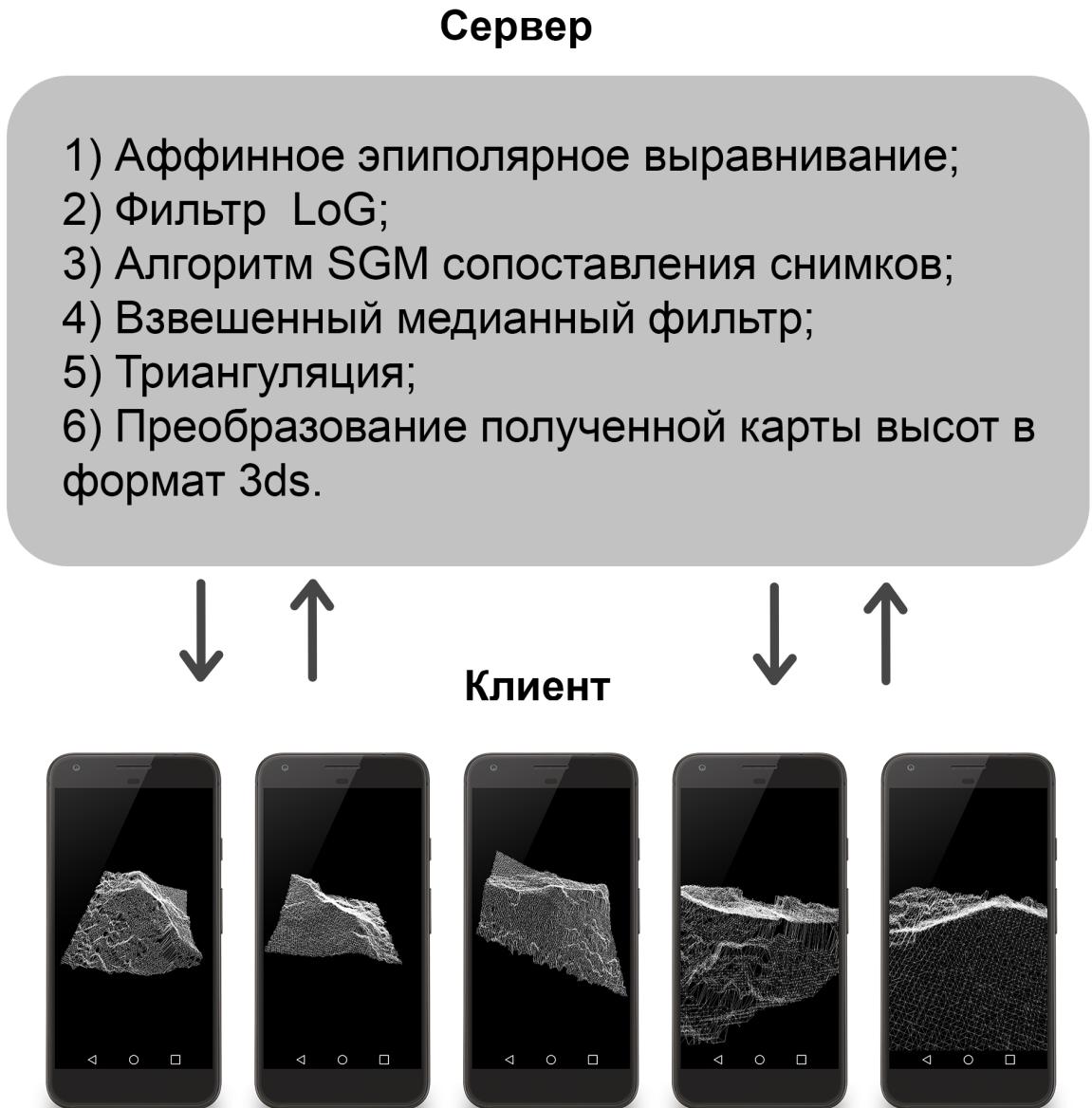


Рис. 27. Схематичное представление реализованного алгоритма

Качество получаемых результатов доходит до 94%. Под качеством здесь понимается количество пикселей, находящихся в перекрываемых об-

ластих стероизображений, для которых удалось установить высоту той области поверхности Земли, проекцией которой они являются.

Алгоритм был получен в результате сравнений возможных подходов к построению карты рельефов местности по спутниковым стереоизображениям.

Были произведены следующие экспериментальные сравнения:

- 1) сравнение ректификации стереоснимков с помощью аффинного эпиполярного выравнивания и с помощью проекции снимков на общую плоскость с использованием гомографии;
- 2) сравнение использования фильтра LoG совместно с локальным алгоритмом стереосопоставления и с помощью алгоритма SGM;
- 3) сравнение локального алгоритма стереосопоставления снимков и алгоритма стереосопоставления SGM;
- 4) сравнение форматов данных *stl*, *obj*, *3ds* в рамках задачи визуализации карты рельефов на платформе Android.

## Список литературы

- [1] K. Jacobsen. DEM generation from satellite data, 2003.
- [2] Th. Toutin, R. Chénier, Y. Carboneau. 3d models for high resolution images: examples with Quickbird, Ikonos and Eros. 2002.
- [3] Jiayin Liu, Xiaolan Qiu, Wen Hong. Automated ortho-rectified SAR image of GF-3 satellite using reverse-range-doppler method. 2016.
- [4] Hong Zhiganga, Chu Meihua, Yan Qin. Rectification of high-resolution satellite images using an approach of image series. 2005.
- [5] Sui Liansheng, Zhang Jiulong, Cui Duwu. Image Rectification Using Affine Epipolar Geometric Constraint. 2009.
- [6] Jaehong Oh. Novel Approach to Epipolar Resampling of HRSI and Satellite Stereo Imagery-based Georeferencing of Aerial Images. 2011.
- [7] Hartley, Richard, and Andrew Zisserman. Multiple view geometry in computer vision. Vol. 2. Cambridge, 2000.
- [8] L. S. Shapiro, A. Zisserman, J. M. Brady. 3D Motion Recovery via affine epipolar geometry. International Journal of Computer Vision, Volume 16, Number 2, страницы 147–182, 1995
- [9] Kristen Grauman, Bastian Leibe. Visual Object Recognition. Synthesis Lectures on Artificial Intelligence and Machine Learning. 2011.
- [10] Yu Shuchun, Yu Xiaoyang, Hu Lijuan, Wang Jue. Preprocessing for stereo vision based on LOG filter. Strategic Technology (IFOST), 6th International Forum on. 2011.
- [11] Gaojian Li. Stereo Matching using Normalized Cross-Correlation in LogRGB Space. Lab of Algorithm for Cognitive Model School of Computer Science & Technology Fudan University, Shanghai, China. 2012.
- [12] Akhil Appu Shetty, V.Itti.George, C.Gurudas Nayak, Raviraj Shetty. Normalized Cross Correlation for Stereo Matching Under Varying Illumination. IJCTA, 9(21), pp. 39-42. 2016.
- [13] Simon Hermann, Reinhard Klette. Iterative Semi-Global Matching for Robust Driver Assistance Systems. The .enpeda.. Project, Department of

Computer Science The University of Auckland, New Zealand. 2012.

- [14] Heiko Hirschmüller. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA. 2005.
- [15] Matthias Michael, Jan Salmen, Johannes Stallkamp, Marc Schlipsing. Real-time Stereo Vision: Optimizing Semi-Global Matching. 2013.
- [16] F. Bethmann, T. Luhmann. Semi-Global Matching in Object Space. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-3/W2. 2015.
- [17] Wenzuan Wu, Li Li, Weiqi Jin. Disparity refinement based on segment-tree and fast weighted median filter. 2016.
- [18] Abdul Qayyum, Aamir Saeed Malik, Mohammad Nuafal, Mahboob Iqbal, Mohad Faris Abdullah. Design of Digital Elevation Model Based on Orthorectified Satellite Stereo Images. Electrical and Electronics Engineering Department, Universiti Teknologi PETRONAS Tronoh, Perak, Malaysia. 2015.
- [19] Rostam Affendi Hamzah, Haidi Ibrahim, Anwar Hasni Abu Hassan. Stereo Matching Algorithm for 3D Surface Reconstruction Based on Triangulation Principle. 1st International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia. 2016.