

**И.В. Гужавина**

## **ТЕХНОЛОГИИ ОБРАБОТКИ ИЗОБРАЖЕНИЯ СО СПУТНИКА**

### **ВВЕДЕНИЕ**

Снимки земной поверхности, полученные путём космической съёмки, отличаются тем, что при целостном характере изображения местности они охватывают огромные площади (на одном снимке от десятков тысяч  $км^2$  до всего земного шара). Это позволяет изучать по космическим снимкам основные структурные, региональные, зональные и глобальные особенности атмосферы, литосферы, гидросферы, биосферы и ландшафты нашей планеты в целом. Космические съёмки Земли применяются во многих областях науки; таким образом, накопление базы данных изображений Земли является важной задачей.

Существует несколько видов различных спутников, с которых ведется наблюдение поверхности Земли: геостационарные (ГСО) и высокоэллиптические (ВЭО). Эти спутники различаются орбитами, по которым они совершают движение, углом съёмки, стоимостью и прочим. С геостационарного спутника можно получить прямые изображения Земли низкого разрешения, но данный спутник достаточно дорого стоит. В отличие от ГСО, ВЭО позволяет при наблюдении с различных участков орбиты получать изображения фонов Земли в разных масштабах. Последовательные 9 витков орбиты спутников ВЭО имеют устойчивую конфигурацию, охватывающую всю Землю при различных условиях освещенности. Это позволяет в широких пределах выбирать условия наблюдения фонов Земли в интересующих регионах, максимально приближенные к условиям ГСО. Но при съёмке с высокоэллиптической орбиты камерой с малым углом зрения ( $0,12^\circ \times 0,24^\circ$ ) и фотоприемной матрицей небольшого размера ( $256 \times 500$ ) получаемые изображения покрывают небольшие участки земной поверхности. Для получения изображений высокого разрешения, покрывающих большие площади, необходимо построить панораму. Изображения, получаемые при съёмке с высокоэллиптической орбиты, содержат проективные искажения. Поэтому при создании из этих изображений панорам необходимо компенсировать проективные искажения.

Таким образом, с высокоэллиптического спутника можно получить изображения более высокого разрешения, но с проективными искажениями. Стоимость таких спутников относительно мала, поэтому для получения изображений с большим разрешением и из-за более низкой стоимости лучше использовать высокоэллиптические спутники.

## 1. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Изображения, получаемые при съемке с высокоэллиптической орбиты, содержат проективные искажения, обусловленные изменением координат спутника и точки прицеливания. После съемки получается большое количество разрозненных изображений, которые необходимо объединить для получения более общей картины.

Для компенсации проективных искажений используется известный алгоритм, описанный в [1]. Он представляет собой ряд геометрических преобразований, которые переводят одну координатную сетку в другую. Но так как изображения имеют разный масштаб (на один пиксель изображения ГСО приходится приблизительно 1,5–6 пикселей изображения ВЭО), преобразование «пиксель в пиксель» при расчете яркости будет некорректным. Необходим алгоритм для пересчета яркости в каждом пикселе координатной сетки ГСО, который будет усреднять значение яркости всех пикселей ВЭО, попавших в данный пиксель.

Таким образом, для пересчета изображений необходимо пройти 2 этапа:

- пересчет координатной сетки ВЭО в координатную сетку ГСО, то есть сопоставление каждой точке изображения ВЭО точки на изображении ГСО;
- пересчет яркости для каждой точки изображения.

После компенсации проективных искажений и пересчета яркости осуществляется создание панорамного изображения.

На сегодняшний день существует достаточно литературы, описывающей проблему создания панорам из отдельных изображений, существует также несколько коммерческих приложений, решающих данную задачу, например программное обеспечение PhotoStitch для цифровых фотокамер Canon, REALVIZ Stitcher, Arcsoft PanoramaMaker, iSee PhotoVista Panorama и другие [2]. Практически все приложения требуют предварительной инициализации, т.е. набора входных параметров, по которым будет происходить сопоставление. Так PhotoStitch работает только с набором изображений, полученных вертикальным или горизонтальным сдвигом, либо квад-

ратной матрицей изображений. У REALVIZ Stitcher обеспечивает подход «drag, drop and stitch», то есть пользовательский интерфейс для приблизительного расположения изображений, который ограничивает зону поиска особых точек, а также позволяет приблизительно задать поворот одного изображения относительно другого.

Практически все приложения обладают как автоматическим, так и ручным режимом создания панорам, но автоматические режимы зачастую работают некорректно. Так, при создании панорамы с помощью iSee PhotoVista Panorama точки совпадения определились неправильно при автоматическом режиме. Данные программы ориентированы на ручную подстройку изображений; кроме того, ни одна из представленных программ не позволяет компенсировать проективные искажения, не работает с изображениями, имеющими различный масштаб, плохо сглаживает яркость и зачастую оставляет видимые границы на склеенном изображении. Все программные продукты не предусматривают использования дополнительной информации об изображениях, такой как параметры и положение камеры.

Алгоритм создания панорам состоит из следующих этапов:

1. Поиск особых точек (либо других свойств) изображений для сопоставления.
2. Поиск «ближайших соседей» для сопоставления особых точек.
3. Вычисление гомографии для пар изображений.
4. Вычисления параметров камер.
5. Построение панорамы.

В данном случае геометрия съемки известна: матрица внутренних параметров камеры, матрица поворота камеры и её положение. Таким образом, построить панораму можно сразу. Но если данные имеют ошибку, например, в координатах положения камеры, то панорама может получиться разрывной. Для поправки данных можно использовать алгоритм создания панорам, но в урезанном виде. Также необходим алгоритм, который позволит склеивать изображения так, что в панораме не будет видно границ стыков между изображениями и яркость изображения будет равномерной.

Существует два подхода для поиска ключевых особенностей (особых точек) изображений – прямые методы [3] и методы, основанные на выявлении ключевых свойств изображения [3]. Основным преимуществом прямых методов является использование всех сведений об изображении, но для их применения нужна точная инициализация, т.е. прямое указание того как нужно склеивать изображения. Методы, основанные на выявлении ключевых свойств изображения, более стойки к движению, работают гораздо быстрее прямых методов. Однако основным преимуществом этих методов

является автоматическое нахождение перекрывающихся участков изображений в несортированном наборе изображений.

Среди методов, основанных на выявлении ключевых свойств изображения, наиболее известным и широко применяемым является детектор Харриса [4]. Сопоставление особенностей происходит путём сравнения их окрестностей кросскорреляционным или другим методом. Однако на больших по размеру изображениях (более  $256 \times 256$  пикселей) сразу несколько особенностей на первом изображении могут оказаться близкими к одной и той же особенности на втором изображении. Отличить ложные соответствия от верных не всегда возможно. Данный способ плохо работает, и в случае, если изображения имеют разную яркость.

Более эффективным является алгоритм SIFT [5], который лишен недостатков вышеописанного алгоритма, но обладает высокой сложностью обработки.

SIFT (Scale Invariant Feature Transform) преобразует данные изображения в независимые от масштаба координаты, связанные с локальными свойствами изображения. Данный алгоритм находит достаточно большое количество особых точек, которые отвечают различным масштабам и участкам одного и того же изображения. Для каждой точки строится некоторое описание, которое позволяет с большой вероятностью найти точное соответствие для данной точки на других изображениях.

Таким образом, данный алгоритм позволяет выявлять характерные свойства изображения (особые точки), которые не зависят от колебаний яркости, и находить для них точное соответствие.

После того, как были найдены особые точки для всех изображений, необходимо найти соответствия между изображениями, т.е. набор особых точек, совпадающих у двух или более изображений. Эта задача соответствует поиску  $N$  ближайших соседей. Для решения этой задачи существует много подходов. Один из прямых методов решения – метод ячеек. Всё пространство делится на равные небольшие  $k$ -мерные кубики ( $k$  – размерность пространства) – «ячейки» – по определённому ключу, затем поиском по спирали находится лучшее соответствие для заданной записи. Несмотря на то, что эта процедура минимизирует количество рассматриваемых записей, она требует больших временных затрат, а также много памяти.

Другой подход описан в [6]. Этот метод, основан на формировании проекций записей на один или несколько ключей, хранении линейного списка этих ключей и поиска только среди тех записей, которые достаточно близки по значениям к этим ключам. В [6] показано, что данный метод требует

$km^{1/k} N^{1-1/k}$  вычислений, где  $k$  – размерность пространства,  $N$  – число рассматриваемых записей,  $m$  – количество ключей.

Еще один подход, описанный в [7], основан на модели  $k$ - $d$  деревьев, которые являются обобщением бинарных деревьев.  $K$ - $d$  дерево представляет собой двоичное дерево, каждая вершина которого содержит подмножество записей и разделение данного подмножества. Такая структура данных позволяет эффективно разделить записи и осуществлять поиск в пределах небольших множеств. Как показано в [7], этот метод требует  $kN \log N$  вычислений, а время поиска пропорционально  $\log N$ .

Когда вычислены ближайшие соседи, необходимо склеить изображения. Основной проблемой здесь является разница в яркости изображений и появление вследствие этого границ на склеенном изображении. Для решения данной проблемы существует несколько подходов. Первый – линейно уменьшать влияние одного изображения и увеличивать влияние второго изображения. Данный метод достаточно прост и быстр, но после его применения граница может остаться, хотя и будет менее заметна.

Более гладкий переход может быть получен при использовании техники, описанной в [8]. С использованием данной техники строится «самая гладкая» из возможных функция коррекции для каждого изображения, входящего в панораму. Однако данная техника не применима к большим изображениям, так как для вычисления функции коррекции используется итеративный алгоритм релаксации.

Подход, предложенный в [9] – метод сплайнов с переменной разрешающей способностью. В данном подходе склеиваемые изображения фильтруются полосовым фильтром, после чего получается последовательность изображений с различной пространственной частотой. Затем для каждого изображения, участвующего в панораме, строится взвешивающую функцию, которая также фильтруется. Наконец, для каждой частоты изображения складываются с соответствующей взвешивающей функцией и суммируются по всем частотам. Данный подход обеспечивает склейку панорамного изображения; при этом на изображении не появляются такие искажения, как двойные объекты или видимая граница перехода.

## 2. ТРЕБОВАНИЯ К РАЗРАБАТЫВАЕМЫМ АЛГОРИТМАМ И ПРОГРАММНЫМ СРЕДСТВАМ

Алгоритмы и программные средства преобразования изображений разрабатываются для пересчета изображений фонов, наблюдаемых с эллипти-

ческих орбит, в вид, характерный для геостационарных условий наблюдения, с целью формирования базы экспериментальных данных наблюдений фона Земли.

В процессе пересчета полученных данных на условия ГСО необходимо решить следующие основные задачи:

- провести пересчет на условия ГСО с учетом различия условий и аппаратуры наблюдения;
- построить панораму;
- пересчет изображений происходит по заранее подготовленному заданию, в котором содержатся необходимые параметры.

Для пересчета изображения необходимо пересчитать координатную сетку, а затем яркость для каждого пикселя. Для пересчета координатной сетки используется известный алгоритм, компенсирующий проективные искажения.

При пересчете яркости необходимо учесть, что пиксели изображений с ВЭО и с ГСО имеют различные размеры. Пиксель ГСО произвольно отображается на координатную сетку ВЭО; таким образом, возникают пиксели ВЭО, которые частично попали в пиксель ГСО. Для наиболее точного пересчета яркости необходимо учитывать все вхождения пикселей ВЭО в пиксель ГСО. Таким образом, необходим алгоритм, учитывающий все особенности данных изображений.

Для построения панорам существует достаточное количество различного программного обеспечения, но оно в большинстве своем рассчитано на интерактивную работу, т.е. ручную настройку пользователем параметров и приблизительное совмещение изображений, и не учитывает информацию о съемке. Так как известны параметры съемки, необходимо разработать алгоритм, который позволит автоматически создавать панорамные изображения, уточнять параметры съемки, сглаживать границы склеенных изображений, усреднять яркость в пределах одного панорамного изображения. Алгоритм для создания панорамы будет состоять из следующих шагов:

1. Поиск особых точек (либо других свойств) изображений для сопоставления.
2. Поиск «ближайших соседей» для сопоставления особых точек.
3. Уточнение параметров съемки.
4. Построение панорамы.

При поиске особых точек необходимо учесть, что изображения могут иметь различную яркость.

Для более точного совмещения необходимо достаточное количество особых точек, поэтому для поиска ближайших соседей необходим доста-

точно простой алгоритм, который будет за небольшое время находить возможные варианты для сопоставления изображений с большой вероятностью.

Также необходимо выбрать алгоритм для «склейки» изображений. Данный алгоритм должен достаточно гладко совмещать изображения, т.е. после склейки граница между фрагментами изображения должна отсутствовать.

Таким образом, необходим алгоритм, который будет удовлетворять следующим требованиям:

- учитывать такие особенности изображений, как различная яркость и различный масштаб;
- достаточно быстро работать с большим количеством изображений;
- автоматически собирать панорамные изображения по уточнённым параметрам съемки;
- обеспечивать отсутствие видимых границ склейки на панорамном изображении.

Для удовлетворения данных требований для каждого этапа были выбраны соответствующие алгоритмы:

- SIFT для поиска особых точек.
- Метод Best Bin First (BBF) на основе  $k$ - $d$  деревьев для поиска ближайших соседей.
- Метод слайнов с переменной разрешающей способностью для склейки изображений.

### 3. АЛГОРИТМЫ

#### 3.1. Пересчет координатной сетки

Пересчёт координат изображений из ВЭО в ГСО производится в предположении, что участок поверхности, наблюдаемый в текущий момент, аппроксимируется фрагментом плоскости касательной к поверхности Земли в точке наведения  $P$ .

Для пересчёта изображения используются следующие входные параметры:

- $t$  – время получения исходного изображения.

- Параметры спутников: векторы положения спутника  $\overline{t_{ВЭО}}$  и  $\overline{t_{ГСО}}$ , форматы фотоприёмных матриц  $(Nx_{ВЭО}, Ny_{ВЭО})$  и  $(Nx_{ГСО}, Ny_{ГСО})$ , угловые разрешения  $v_{ВЭО}$  и  $v_{ГСО}$ , координаты главных точек аппаратур наблюдения в элементах фотоприёмной матрицы  $(u_{ВЭО}^0, v_{ВЭО}^0)$  и  $(u_{ГСО}^0, v_{ГСО}^0)$ .
- Координаты точки  $P$  на поверхности Земли, на которую наведена визирная ось аппаратуры спутника, находящегося на высокой эллиптической орбите.

Алгоритм пересчёта координатной сетки состоит из следующих этапов: по входным значениям вычисляется матрица ориентации аппаратуры наблюдения ГСО, затем координаты изображения с ВЭО пересчитываются в ГСО.

Положение и ориентация камеры спутника для текущего кадра задается вектором трансляции  $\overline{t_{ВЭО}}$  и матрицей поворота  $R_{ВЭО}$ , которая в геоцентрической неинерциальной системе представляется в виде:

$$R_{ВЭО}(t) = \begin{pmatrix} i_1(t) & i_2(t) & i_3(t) \\ j_1(t) & j_2(t) & j_3(t) \\ k_1(t) & k_2(t) & k_3(t) \end{pmatrix} = \begin{pmatrix} i^T(t) \\ j^T(t) \\ k^T(t) \end{pmatrix}.$$

Здесь  $k^T(t)$  – вектор, задающий направление оси визирования камеры, а  $i^T(t)$  и  $j^T(t)$  – векторы, определяющие систему координат плоскости изображения. Если координаты спутника задаются вектором  $\overline{t_{ВЭО}}$ , а координаты точки наведения вектором  $\overline{P}$ , то

$$k^T(t) = \frac{\overline{P} - \overline{t_{ВЭО}}}{|\overline{P} - \overline{t_{ВЭО}}|}.$$

Наведение на точку визирования осуществляется сначала поворотом камеры в плоскости орбиты, затем выносом оси визирования в требуемую точку. Следовательно, одна из осей камеры (пусть это будет ось  $j(t)$ ) находится в плоскости орбиты. Это означает, что она ортогональна вектору нормали к плоскости орбиты  $n_0(t)$ . По определению же она ортогональна вектору  $k(t)$ . Из этих условий следует, что



$$j(t) = \frac{n_0(t) \times k(t)}{|n_0(t) \times k(t)|}.$$

Ось  $i(t)$  дополняет систему до правой тройки, следовательно  $i(t) = j(t) \times k(t)$ .

Аналогичные вычисления проводятся для матрицы ориентации  $R_{ГСО}$ .

Преобразование координат изображений из ГСО в ВЭО производится по формулам:

$$[u_{ГСО}, v_{ГСО}] = \frac{1}{\gamma_{ГСО}} \cdot \frac{R_{ГСО} \cdot \left[ (R^2 - \bar{p}^T \bar{t}_{ВЭО}) \cdot \bar{I} + (\bar{t}_{ВЭО} - \bar{t}_{ГСО}) \cdot \bar{p}^T \right] \cdot R_{ВЭО}^T A_{ВЭО}^{-1} \bar{v}_{ВЭО}}{\bar{e}_z^T R_{ГСО} \cdot \left[ (R^2 - \bar{p}^T \bar{t}_{ВЭО}) \cdot \bar{I} + (\bar{t}_{ВЭО} - \bar{t}_{ГСО}) \cdot \bar{p}^T \right] \cdot R_{ВЭО}^T A_{ВЭО}^{-1} \bar{v}_{ВЭО}} \cdot [\bar{e}_x^T, \bar{e}_y^T],$$

где

$$A_{ВЭО} = \begin{bmatrix} v_{ВЭО}^{-1} & 0 & u_{ВЭО}^0 \\ 0 & v_{ВЭО}^{-1} & v_{ВЭО}^0 \\ 0 & 0 & 1 \end{bmatrix} -$$

матрица внутренних параметров камеры ВЭО,  $\bar{s} = (X, Y, Z)^T$  – координаты изображения произвольной точки  $S$ , заданной в неинерциальной геоцентрической системе координат,  $\bar{v}_{ВЭО} = (u_{ВЭО}, v_{ВЭО}, 1)^T$  – координаты в плоскостях изображения камер в пикселях (аналогичные параметры для ГСО),  $\bar{p} \cdot \bar{s} = R^2$ .

### 3.2. Вычисление яркости в пикселе ГСО

Пересчет яркости для каждого пикселя ГСО заключается в интерполяции значений яркости элементов изображения ВЭО, накрываемых им целиком или частично.

Так как размер и ориентация пикселей изображений, получаемых с ВЭО и с ГСО, различны, то необходимо найти проекцию пикселя ГСО на координатную сетку ВЭО и рассчитать яркость, учитывая все вхождения пикселей ВЭО в пиксель ГСО.

Для уменьшения вычислений, используя проективные преобразования, сначала вычисляем координаты углов изображения ВЭО ( $\text{Im}_{ВЭО}$ ) на координатной сетке изображения ГСО и выделяем на этой сетке прямоугольник минимального размера, накрывающий изображение ВЭО.

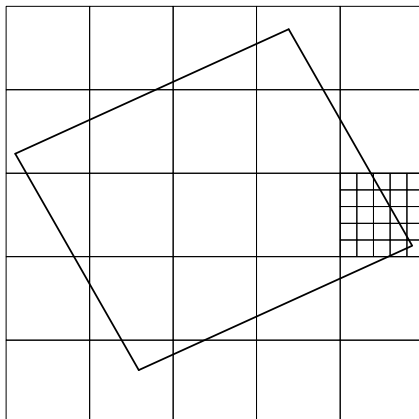


Рис. 1. Пример соотношения пикселей ВЭО и ГСО

Затем для вычисления яркости в пикселе, отображаем каждый пиксель изображения ГСО ( $P_{GCO}$ ) на координатную сетку изображения ВЭО. Если  $P_{GCO} \in \text{Im}_{VZO}$ , то вычисляем яркость, иначе значение яркости не вычисляется.

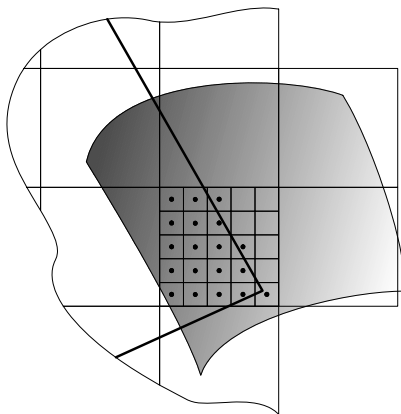


Рис. 2. Разбиение пикселя ВЭО на более мелкую решётку с вычислением значений через сплайн

Рассмотрим рис. 1, те пиксели  $P_{VZO}$ , которые попадают целиком в пиксель ГСО, просто складываем в общую сумму. Остальные рассматриваем

на более подробных решётках  $\varepsilon \times \varepsilon$ . Для этого натягиваем на соседние точки сплайн рис. 2), вычисляем значения ячеек решётки и считаем среднюю сумму для тех, которые хотя бы частично попали в пиксель ГСО (на рисунке обозначены чёрными точками).

Таким образом, яркость в пикселе ГСО будет иметь следующее значение:

$$\frac{1}{m_{\text{inside}}} \sum_{i=1}^m P^i_{B \in \text{Ofull}} + \frac{1}{n_{\text{inside}}} \sum_{j=1}^n P^j_{\varepsilon \text{Spline}}.$$

### 3.3. Алгоритмы для создания панорамы

Для получения панорамы из набора изображений при известной геометрии съемки, необходима только компенсация разницы яркостей на изображениях и краевых эффектов. Но зачастую данные содержат ошибки, поэтому необходимо уточнять параметры съемки. Для этого был использован алгоритм, основанный на выявлении особых точек на изображениях. Между особыми точками на различных изображениях находятся соответствия, затем по этим соответствиям уточняются матрицы гомографии для пар изображений. По уточненным данным изображения склеиваются в панораму, компенсируется разница яркостей разных изображений, и сглаживаются границы склейки.

#### 3.3.1. Алгоритм для вычисления особых точек

Для вычисления особых точек были использованы идеи алгоритма SIFT [5]. Этот подход состоит из 4 шагов:

**1. Определение масштабно-пространственных экстремумов:** на этом этапе для изображения строится пирамида с её различными масштабами, на каждом уровне которой и находятся особые точки

Для отыскания точечных особенностей была использована функция, получившая название Difference-of-Gaussian (DOG). Необходимо построить целую пирамиду различных масштабов изображения с её применением:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma),$$

где  $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$  – распределение Гаусса,  $I(x, y)$  – исходное изображение,  $*$  – операция свертки, а  $\sigma$  – масштаб (степень размытия изображения).

Для уменьшения вычислений использован следующий приём: можно заметить, что уменьшение размеров изображения в два раза приводит к двукратному увеличению её масштаба (то есть  $\sigma$ , с которой она свёрнута). Каждый шаг удвоения  $\sigma$  разбивается на  $S$  частей, таким образом, выбираем  $k = 2^{1/S}$ .

Итак, получен набор разностей изображений с различной степенью размытости (рис. 3).

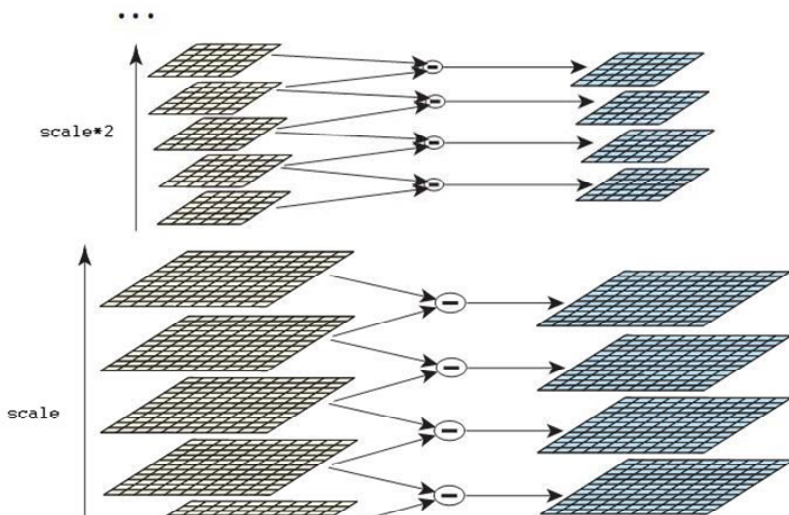


Рис. 3. Изображение последовательно сворачивается с функцией Гаусса, а затем соседние размытые изображения вычитаются друг из друга

**2. Локализация особых точек:** определяются примерные точки-кандидаты, проверяются на устойчивость, неустойчивые отсеиваются, и для каждой такой точки уточняются ее координаты и масштаб  $\sigma$ .

Для увеличения быстродействия в полученной пирамиде DOG ищутся локальные экстремумы следующим образом. На каждом изображении каждого уровня пирамиды рассматриваются все точки, каждая точка сравнивается с 8 соседними точками на своём уровне и с 9 соседними точками на предыдущем и следующем уровнях (рис. 4). Точка принимается как кандидат в особые точки, если она или больше, или меньше всех своих 26 соседей.

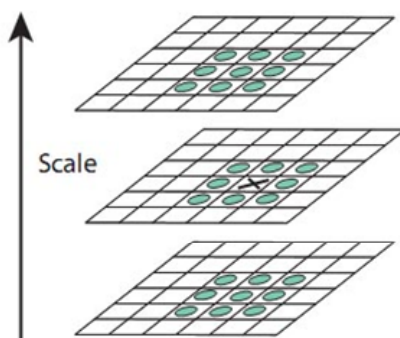


Рис. 4. Поиск локальных экстремумов

Затем найденные кандидаты проверяются дифференциальным путём: если соседняя точка оказывается ближе к локальному экстремуму, то новым кандидатом становится она. Затем отбрасываем точки с низкой контрастностью и большим отношением главной кривизны по  $X$  к главной кривизне по  $Y$ .

**3. Определение направления:** определяется градиентное поле вокруг каждой особой точки, выделяется один или несколько векторов основных его направлений, если их оказывается несколько, то для каждого определяется новая особая точка. Вектор основного направления поможет отыскивать совпадения особых точек на различных изображениях, повернутых на какой-то угол относительно друг друга.

Все последующие манипуляции производятся с особыми точками, преобразованными в соответствии с вычисленным направлением, масштабом и расположением для каждой особой точки. Это обеспечивает инвариантность относительно подобных преобразований.

**4. Вычисление дескрипторов особых точек:** для каждой особой точки вычисляем гистограмму направлений градиентного поля вокруг данной точки и поворачиваем её на угол вектора основного направления, вычисленного в предыдущем пункте (рис. 5).

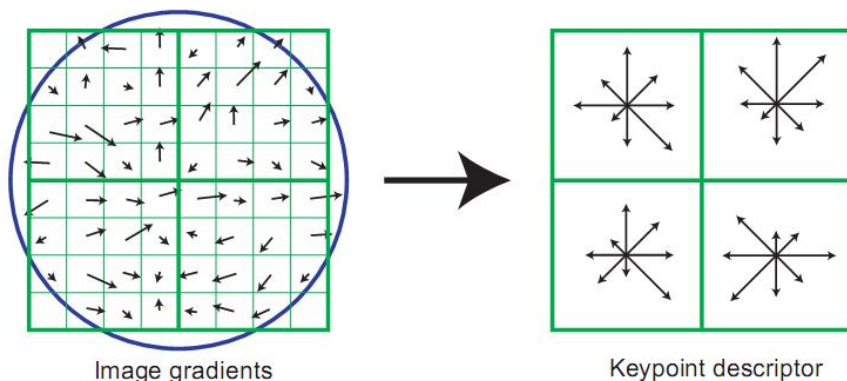


Рис. 5. Дескриптор особой точки

Локальные градиенты вокруг каждой особой точки и масштаб  $\sigma$  обеспечивают устойчивость такого подхода к локальным изменениям формы и яркости изображений.

### 3.3.2. Алгоритм для поиска ближайших соседей.

Задача поиска ближайших соседей в множестве из  $N$ -элементов в пространстве  $R^k$  решена при помощи алгоритма Best Bin First (BBF).

В данном алгоритме используется модель k-d деревьев. K-d дерево представляет собой бинарное дерево, которое строится по следующему правилу:

- начальное множество разбивается по значениям векторов в одной из координат, например, по  $i$ , где  $i = 1, \dots, k$ , на два подмножества;
- $i$  выбирается таким образом, чтобы разброс значений по данной координате был максимальным;
- разбиение проводится по медиане  $m$ , так что одинаковое количество точек оказывается с одной и с другой стороны;
- в вершине дерева хранятся значения  $i$ ,  $m$ , разброс значений векторов по каждой координате;
- для полученных вершин процесс повторяется.

В итоге получим бинарное дерево глубины  $d = \log_2 N$ .

Для поиска  $n$  ближайших соседей к вектору  $q$  в построенном дереве:

- сначала дерево обходится вниз до листа содержащего «ближайшую» к  $q$  точку. Эта точка не обязана быть ближайшей, это только первое приближение;
- во время спуска по дереву заполняется список поддеревьев, которые еще не обходили. Также запоминаются расстояния до них, которое определяется как минимальное расстояние от точки  $q$  до любой точки, находящейся в границах значений поддерева;
- из списка выбираем ближайшее к  $q$  поддерево и продолжаем поиск в нём;
- расстояние до каждого нового найденного претендента сравнивается с радиусом сферы найденных точек с центром в точке  $q$ . Если данное расстояние меньше, то точку на сфере заменяем этим претендентом.

Алгоритм работает до тех пор, пока в списке есть поддеревья с расстоянием, меньшим радиуса сферы найденных точек.

### 3.3.3. Компенсация отличий в яркости

Изображения, входящие в панораму, могут иметь различную яркость. Вследствие этого на конечной панораме могут появиться области с различной яркостью. Необходимо минимизировать функцию ошибки для разности яркостей в пересекающихся областях изображений [11]. Для этого находим коэффициенты для каждого изображения, на которые необходимо их домножить. Пусть  $g_i$  – коэффициент  $i$ -го изображения, входящего в панораму. Тогда функцию ошибки представим как сумму:

$$e = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \sum_{\substack{u_i \in R(i,j) \\ u_i = H_j u_j}} (g_i I_i(u_i) - g_j I_j(u_j))^2,$$

где  $R(i, j)$  – область пересечения  $i$ -го и  $j$ -го изображений. На практике значение  $I(u_i)$  аппроксимируется средним значением в пересекающейся области  $\bar{I}_{ij}$ :

$$\bar{I}_{ij} = \frac{\sum_{u_i \in R(i,j)} I_i(u_i)}{\sum_{u_i \in R(i,j)} 1}.$$

Это упрощает вычисления и дает устойчивость к ложным сопоставлениям особых точек, которые могут возникать ввиду небольших ошибок сопоставления изображений. Чтобы избежать нулевого решения, которое,

очевидно, является оптимальным в данной схеме, добавим дополнительный член, чтобы коэффициенты  $g_i$  стремились к единице при совпадении яркости изображений:

$$e = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n N_{ij} \left( \frac{(g_i \bar{I}_i(u_i) - g_j \bar{I}_j(u_j))^2}{\sigma_N^2} + \frac{(1 - g_i)^2}{\sigma_g^2} \right),$$

где  $N_{ij} = |R(i, j)|$  – число пикселей в пересечении  $i$ -го и  $j$ -го изображений. Параметры  $\sigma_N$  и  $\sigma_g$  – стандартное отклонение от нормализованной ошибки интенсивности и ошибки коэффициентов, соответственно. Эта квадратичная функция на вектор параметров  $\bar{g}$  может быть разрешена методом приравнивания производной к нулю.

### 3.3.4. Метод сплайнов с переменной разрешающей способностью

При составлении панорамы необходимо учесть, что входящие в неё изображения могут иметь разную яркость. Из-за этого на конечном изображении будут видны границы склейки.

Для того, чтобы уменьшить этот эффект можно применить метод сплайнов с переменной разрешающей способностью.

Для склейки изображений:

1. Для каждого изображения построим маску

$$W(x) = \begin{cases} 1, r(c_0^i, x) < r(c_0^j, x), \forall j, \\ 0, \text{ иначе} \end{cases},$$

где  $r(c_0^i, x)$  – расстояние от центра  $i$  изображения до точки  $x$ ,  $j = 1, \dots, n-1$ ;

2. Получим набор изображений и масок для разных диапазонов частот;
3. Высокочастотная часть изображения может быть получена следующим образом:

$$\begin{aligned} B_\sigma^i(x, y) &= I^i(x, y) - I_\sigma^i(x, y), \\ I_\sigma^i(x, y) &= I^i(x, y) * G(x, y, \sigma), \\ W_\sigma^i(x, y) &= W^i(x, y) * G(x, z, \sigma), \end{aligned}$$

где  $I^i(x, y)$  –  $i = 1, \dots, n$  изображение,  $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$ .



$B_\sigma(x, y)$  представляет пространственную частоту в диапазоне длин волн  $\lambda \in [0, \sigma]$

4. Для следующего диапазона частот  $k$  :

$$B_{(k+1)\sigma}^i(x, y) = I_k^i(x, y) - I_{(k+1)\sigma}^i(x, y),$$

$$I_{(k+1)\sigma}^i(x, y) = I_{k\sigma}^i(x, y) * G(x, y, \sigma'),$$

$$W_{(k+1)\sigma}^i(x, y) = W_{k\sigma}^i(x, y) * G(x, z, \sigma'),$$

где  $\sigma' = \sqrt{(2k+1)\sigma}$ .

5. Затем для диапазона пересекающиеся изображения объединяются:

$$I_{k\sigma}^i(x, y) = \frac{\sum_{i=1}^n B_{k\sigma}^i(x, y) W_{k\sigma}^i(x, y)}{\sum_{i=1}^n W_{k\sigma}^i(x, y)}.$$

Благодаря такому подходу высокочастотные объекты размываются слабее, а низкочастотные сильнее. Таким образом, не теряются мелкие объекты, попавшие в полосу склейки двух изображений.

#### 4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И РЕЗУЛЬТАТЫ РАБОТЫ

В результате работы было написано приложение, которое создает панорамы из изображений, содержащих проективные искажения. Для пересчёта яркости после компенсации проективных искажений было реализовано два алгоритма: простой (яркость пикселя ГСО берется из пикселя ВЭО, попавшего в середину пикселя ГСО) и с использованием сплайна.

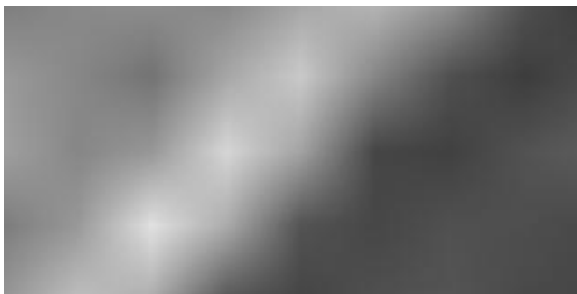


Рис. 6. Изображение с ВЭО, сгенерированное в PovRay

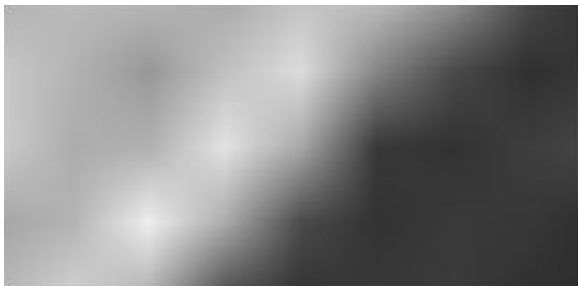


Рис. 7. Результат работы простого алгоритма

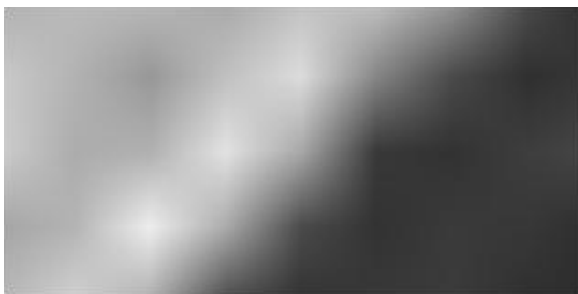


Рис. 8. Результат работы алгоритма с использованием сплайна

Оба алгоритма дают визуально похожие результаты. При подсчете среднеквадратичного отклонения данных двух изображений от исходного получаются следующие величины:

- 4,1501 для простого метода,
- 3,722 для метода с использованием сплайна.

Таким образом, метод с использованием сплайна дает более точный результат.

После компенсации проективных искажений и пересчета яркости из полученных изображений создаются панорамы. Основной проблемой здесь был выбор алгоритмов сглаживающих границы изображений на панораме. Для сглаживания были выбраны два алгоритма: компенсация яркости и метод сплайнов с переменной разрешающей способностью. Далее приведены результаты работы данных алгоритмов.

Далее представлены результаты работы создания панорам. Если изображения имеют одинаковую яркость, то панорама получается гладкой и границы практически не видны (рис. 9). Но если изображения, которые

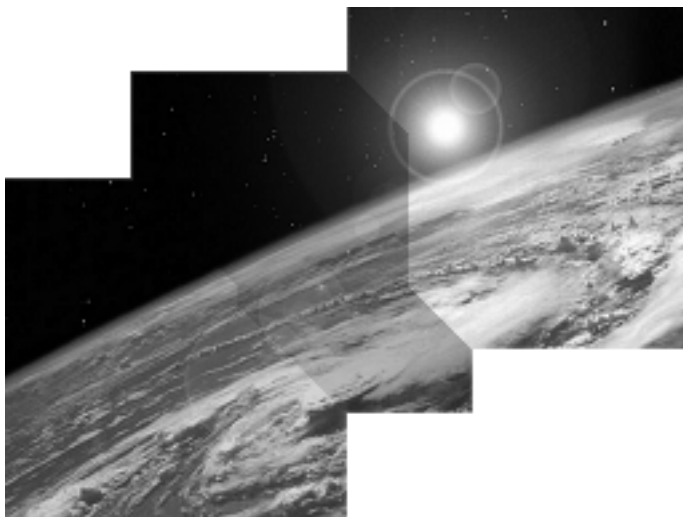
склеиваются в панораму, имеют различную яркость, на результирующем изображении границы видны отчётливо (рис. 10).

Для создания панорамы были выбраны три изображения с различной яркостью. Сначала они были соединены без применения алгоритмов сглаживания границ и компенсации яркости. На рис. 10 отчетливо видны границы между изображениями. После применения алгоритма компенсации яркости изображение стало более гладким (рис. 11). Для большей гладкости границ между изображениями используется метод сплайнов с переменной разрешающей способностью, но при применении только его без компенсации яркости результат получается плохим: яркость разных частей изображений сильно отличается (рис. 12). Если изображения сильно отличаются по яркости и, соответственно, имеют ярко-выраженные границы на панораме, наилучшего результата можно добиться, применяя алгоритм компенсации яркости вместе с методом сплайнов с переменной разрешающей способностью (рис. 13). Но при этом теряется четкость объектов на изображении. В случае, когда яркость отличается не сильно, хороший результат даёт применение только алгоритма компенсации яркости.

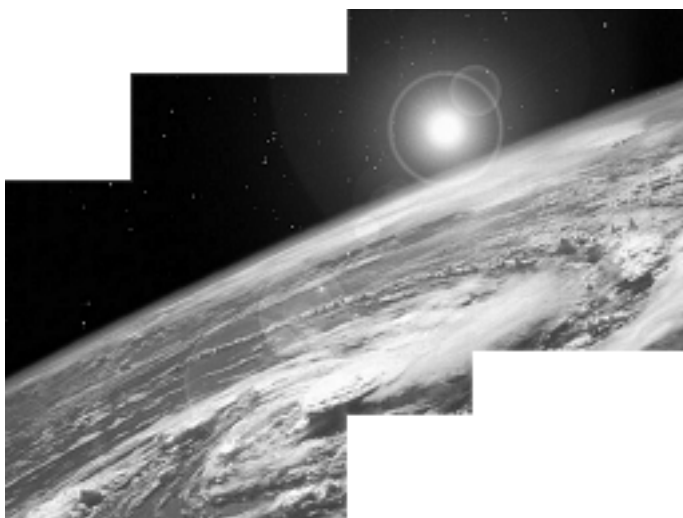


Рис. 9. Панорама из изображений с одинаковой яркостью

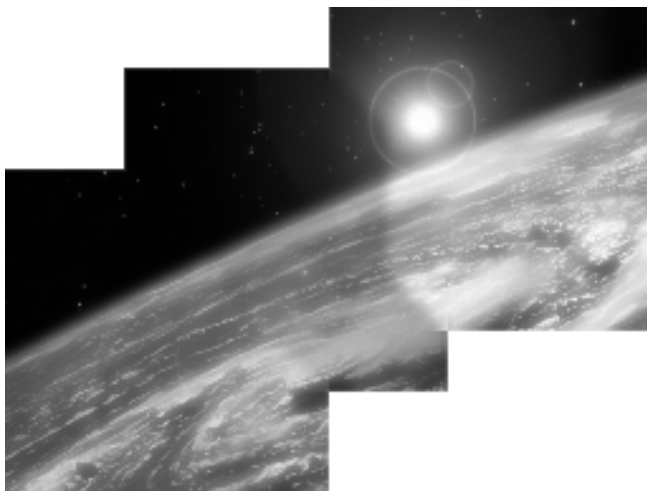
Далее приведены результаты работы программы для изображений, смоделированных в PovRay. В PovRay были смоделированы изображения, получаемые с ВЭО. Затем эти изображения были пересчитаны к условиям наблюдения с ГСО. На заключительном этапе из изображений были созданы панорамы с применением описанных методов сглаживания границ. На рис. 14 представлена панорама из трёх изображений, на рис. 15 – из десяти изображений.



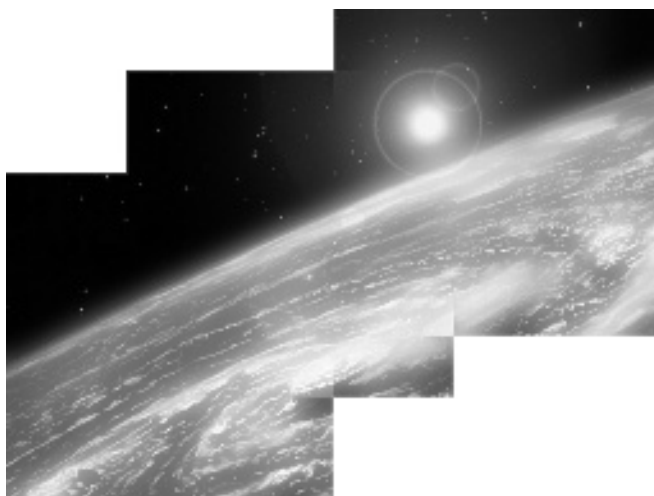
*Рис. 10.* Склейка изображений без применения алгоритмов сглаживания



*Рис. 11.* Склейка изображений с использованием алгоритма компенсации яркости



*Рис. 12.* Склейка изображений с использованием метода сплайнов с переменной разрешающей способностью



*Рис. 13.* Склейка изображений с использованием компенсации яркости и метода сплайнов с переменной разрешающей способностью

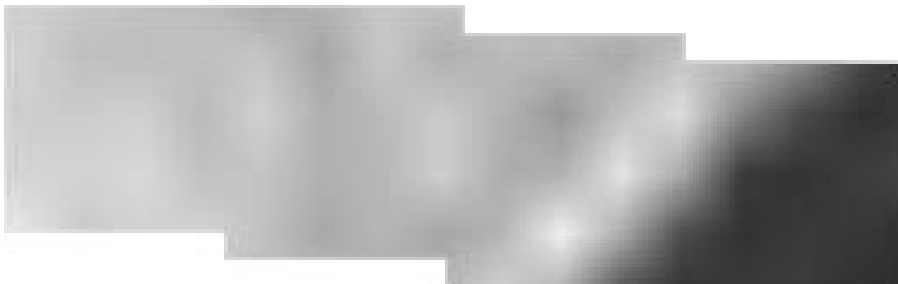


Рис. 14. Панорама из трёх изображений ВЭО, сгенерированных в PovRay



Рис. 15. Панорама из 10 изображений ВЭО, сгенерированных в PovRay

### СПИСОК ЛИТЕРАТУРЫ

1. **Анисимов В.А., Курганов В.Д., Злобин В.К.** Распознавание и цифровая обработка изображений. – М.: Высшая школа, 1983.
2. [Электронный ресурс]: Портал программного обеспечения для работы с фотографиями [текст]. Режим доступа: <http://photosoft.nnm.ru/>.
3. **Szeliski R.** Image Alignment and stitching. – Redmond, WA : Microsoft Research, 2006.
4. **Harris C.** Geometry from visual motion. – MIT Press, 1992.
5. **Lowe D. G.** Distinctive image features from scale-invariant keypoints // International Journal of Computer Vision. – 2004. – Vol. 2.
6. **Friedman J.H., Baskett F., Shustek L.J.** An algorithm for finding nearest neighbors // IEEE Trans. – 1975. – Comput. vol. C-24.
7. **Beis J.S., Lowe D.G.** Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. – Vancouver: University of British Columbia, 1997.
8. **Peleg S.** Elimination of seams from photomosaics // Computer Graphics and Image Processing, 1981 – Vol. 16.

9. **Burt P.J., Adelson E.H.** A multiresolution spline with application to image mosaics // Transaction on Graphics. – 1983. – Vol. 2.
10. **Brown M., Lowe D.G.** Automatic panoramic image stitching using invariant features. – Vancouver: University of British Columbia, 2007.
11. **Richard H., Zisserman A.** Multiple View Geometry in computer vision. – Cambridge: Cambridge University Press, 2003.
12. **Brown M., Lowe D.G.** Invariant features from interest point groups // British Machine Vision Conference. – Cardiff, 2002. – P. 656–665.
13. **Harris C., Stephens M.** A combined corner and edge detector // Fourth Alvey Vision Conference. – Manchester, 1988. – P. 147–151.
14. **Schmid C., Mohr R.** Local grayvalue invariants for image retrieval // IEEE Trans.on Pattern Analysis and Machine Intelligence. – 1997 – Vol. 19.
15. **Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P.** Numerical Recipes in C Second Edition. – Cambridge: Cambridge University Press, 1997.
16. **Киричук В.С., Косых В.П., Перетягин Г.И., Попов С.А.** Использование серии космических снимков с общей точкой наведения для оценивания рельефа местности // Автометрия. – 2002. – Т. 1.