

## МЕТОДИКИ СОПОСТАВЛЕНИЯ ОСОБЫХ ТОЧЕК В ЗАДАЧЕ ВИЗУАЛЬНОЙ НАВИГАЦИИ БПЛА

*Д.Н. Степанов*

Статья посвящена разработке и экспериментальному сравнению методик сопоставления особых точек на изображениях — снимках земной поверхности с камер, установленных на беспилотном летательном аппарате (БПЛА) и искусственном спутнике Земли. Главная особенность задачи состоит в том, что одно из изображений (спутниковый снимок) разбит на фрагменты. Разработанные методики являются частью комплекса алгоритмов, предназначенных для определения положения и ориентации БПЛА с использованием средств, методов и алгоритмов технического зрения. Приведено описание технологии моделирования полета и решения задачи позиционирования. Особые точки на изображениях выделяются с помощью алгоритма SURF. Также исследован подход к сопоставлению, основанный на разбиении множества особых точек на два подмножества в зависимости от знака лапласиана. Предложены способы увеличения производительности сопоставления точек.

*Ключевые слова:* БПЛА, особые точки, SURF, компьютерное зрение, спутниковые снимки, полный перебор, поисковый индекс, сопоставление изображений.

### Введение

В настоящее время сфера применения беспилотных летательных аппаратов (БПЛА) достаточно широка и включает в себя как военные, так и гражданские области. Создание надежных систем навигации и управления БПЛА является актуальным и важным направлением разработок. Широко применяемые радионавигационной системы GPS/ГЛОНАСС имеют свой набор недостатков (отсутствие автономности, потеря и затухание сигнала, естественные и искусственные помехи, вызванные погодой или рельефом местности и т.д.). Инерциальные навигационные микромеханические системы способны работать автономно, но их использование связано с проблемой накопления ошибки в вычислениях положения и ориентации.

Одним из альтернативных способов навигации является использование средств, методов и алгоритмов технического зрения. В 2011–2012 годах в ИПС им. А. К. Айламазяна РАН проводилась научно-исследовательская работа, в рамках которой был разработан алгоритм решения задачи позиционирования БПЛА по видеоряду для полетов над плоской местностью, основанный на сопоставлении так называемых особых точек, найденных на кадре и на спутниковом снимке того участка местности, над которой БПЛА совершает свои полеты. После сопоставления, вычисляются параметры проективного преобразования между двумя изображениями, что позволяет вычислить положение и ориентацию камеры на БПЛА в глобальной системе координат. Проведение экспериментов с БПЛА возможно далеко не всегда из-за ограничений российского законодательства (намекы на некоторое улучшение ситуации появились совсем недавно [1]). Поэтому разработанные алгоритмы тестировались на виртуальных видеорядах — результатах моделирования полета БПЛА с помощью методов компьютерной графики.

Задача выделения и сопоставления особенностей на изображениях достаточно хорошо рассмотрена в литературе. Один из наиболее важных критериев, по которому определяется эффективность алгоритмов выделения и сопоставления особых точек —

устойчивость к изменениям яркости, аффинным преобразованиям (поворот, масштабирование), проективным преобразованиям. Для сопоставления особенностей с разных изображений каждой особой точке приписывается некоторый дескриптор (описатель). Простейший дескриптор – сама окрестность точки, но такой подход будет работать, если окрестность точки предварительно нормализована по масштабу, ориентации и яркости (или же для обеспечения инвариантности требуется много шаблонов в разных ориентациях). Для сравнения окрестностей двух точек используются подходы, основанные на корреляции [2].

Одними из простейших алгоритмов выделения особых точек являются алгоритмы выделения углов на изображениях: примером здесь является детектор углов Харриса [3]. Но детекторы углов не инвариантны к масштабированию: при разных масштабах одного и того же изображения на них могут быть найдены разные угловые точки. Более устойчивы алгоритмы, в которых используется так называемое scale-space представление изображения: размытие изображения при помощи свертки с функцией Гаусса с разными значениями среднеквадратического отклонения [4]: SIFT [5], SURF [6], GLOH [7] и т.д. Во всех этих алгоритмах для каждой особой точки вычисляется ее дескриптор, который в определенной мере инвариантен к изменениям яркости, аффинным и перспективным преобразованиям.

Существует набор алгоритмов, связанных с вычислением так называемого оптического потока, классическим подходом здесь является алгоритм Лукаса—Канаде и его расширенная пирамидальная версия, в которой используется представление изображения в нескольких масштабах и которая характеризуется большей устойчивостью к изменению масштаба [8]. Но алгоритмы оптического потока изначально были разработаны для сопоставления точек с двух изображений, которые незначительно различаются между собой (чего нельзя сказать о снимках с ИСЗ и БПЛА).

Для сопоставления дескрипторов особых точек с двух изображений используются следующие методы:

- полный перебор: для каждого дескриптора из первого набора вычисляется расстояние до всех дескрипторов со второго изображения и берется лучшая;
- ускоренные приближенные метод с использованием иерархических структур данных. В этих методах строится так называемый поисковый индекс для поиска ближайших соседей в многомерном пространстве. Примером такой структуры данных является kd-дерево [9].

Если же рассмотреть задачу выделения и сопоставления особых точек с изображений применительно к фотоснимкам земной поверхности (например, с искусственного спутника Земли — ИСЗ), то мы приходим к понимаю следующей проблемы: эталонные снимки земной поверхности могут иметь очень большие размеры, и непосредственное сопоставление особых точек со всего эталонного снимка может быть крайне ресурсозатратной операцией (особенно с учетом вычислительных возможностей бортовых компьютеров). Если строить для всего набора эталонных данных поисковый индекс, то его представление в оперативной памяти так же будет весьма значительным по размеру. Для решения проблемы в рамках НИР было предложено разбиение спутникового снимка на независимые фрагменты. Также был разработан алгоритм сужения области поиска [10], который основан на определении координат и размеров той области спутникового снимка, которая запечатлена на текущем кадре с борта БПЛА.

Цель настоящего исследования — развитие и совершенствование технологии позиционирования БПЛА с помощью методов и алгоритмов технического зрения. Задачи исследования — разработать методики эффективного сопоставления особых точек со снимков с ИСЗ и БПЛА, а также исследовать эффективность их программных реализаций. В разделе 1 дано общее описание технологии моделирования полета БПЛА и решения задачи визуальной навигации. В разделе 2 описаны предложенные методики сопоставления особых точек. В разделе 3 предложены способы сравнения методик на предмет эффективности. В разделе 4 приведены результаты экспериментальных исследований. В заключении приведены результаты сравнения методик, а также возможные способы увеличения их производительности.

## 1. Описание технологии моделирования полета и решения задачи позиционирования

Опишем технологию моделирования полета виртуального БПЛА и его позиционирования с помощью методов и алгоритмов технического зрения.

Этап подготовки данных:

1. Для исходного спутникового снимка строится так называемая пирамида изображений, при этом каждый слой пирамиды разбивается на фрагменты (тайлы, англ. tiles).
2. На каждом тайле выполняется поиск особых точек с помощью алгоритма SURF, а также вычисляются их дескрипторы (каждый дескриптор — вектор из 64 вещественных чисел).
3. Среди особых точек, найденных на тайле, выбирается несколько точек, для каждой из которых вычисляется еще один дескриптор, состоящий из откликов различных вейвлетов Габора [11], а также вычисляются трехмерные координаты этой точки. Эти несколько точек назовем опорными.
4. Все вычисленные данные сохраняются на жесткий диск для дальнейшего использования в рамках решения задачи визуальной навигации.

Моделирование полета посредством порождения виртуального видеоряда:

1. Пользователь задает желаемую траекторию полета БПЛА, отмечая на уменьшенной копии спутникового снимка ряд опорных точек, которые затем соединяются прямыми отрезками.
2. На каждом отрезке выбирается ряд промежуточных точек, которые будут соответствовать тем моментам времени, когда камера на БПЛА сняла очередной кадр видеоряда.
3. Для каждой промежуточной точки вычисляется ориентация камеры на БПЛА (при условии, что в случае горизонтального полета камера направлена перпендикулярно поверхности Земли) и генерируется кадр видеоряда, с использованием спутниковых снимков местности и (возможно) цифровых карт высот.

Собственно, этап решения задачи навигации состоит в выполнении следующих действий для каждого кадра видеоряда:

1. С помощью алгоритма SURF на кадре выделяются особые точки, вычисляются их дескрипторы.
2. Исходя из положения БПЛА в предыдущий момент времени, с помощью алгоритма сужения области поиска [10, 12] выбираются подходящий слой пирамиды спут-

никовых снимков, а также набор тайлов из этого слоя, особые точки с которых будут использоваться для сопоставления с особыми точками на кадре с БПЛА.

3. Выполняется сопоставление особых точек со снимка с БПЛА и тайлов спутникового снимка. На выходе имеем множество пар координат особых точек, при этом часть пар могут быть ложными (в силу того, что какие-то особые точки на снимках с БПЛА или ИСЗ могут иметь схожие дескрипторы).
4. С помощью 5-точечного алгоритма [13], объединенного с методикой RANSAC [14] вычисляется фундаментальная матрица  $F$ , связывающая координаты особых точек на снимках с БПЛА и ИСЗ. Методика RANSAC позволяет устранить часть ложных соответствий. Производится декомпозиция матрицы  $F$  на матрицу поворота  $R$  (определяет взаимную ориентацию камер на БПЛА и ИСЗ) и вектор  $T$  (определяет взаимное положение камер, но с точностью до некоторого неизвестного положительного коэффициента).
5. С помощью матрицы  $R$  и вектора  $T$  производится геометрическая коррекция снимка с БПЛА для проведения процедуры поиска опорных точек со спутникового снимка на снимке с БПЛА с применением фильтров (вейвлетов) Габора. Детальное описание процедуры будет приведено в следующих публикациях. Результатом поиска является набор пар вида «двумерные координаты опорной точки на снимке с БПЛА — трехмерные координаты этой точки в глобальной системе координат». Это позволяет свести задачу позиционирования к задаче внешней калибровки камеры на БПЛА (вычисления ее положения и ориентации в глобальной системе координат) [15]. Поскольку на этапе генерации видеоряда мы изначально знали положение и ориентацию камеры на БПЛА для каждого кадра, эти данные можно использовать для проверки точности навигации путем вычисления расстояния между истинным положением БПЛА и результатом работы алгоритма позиционирования.

## 2. Методики сопоставления особых точек

В реализации алгоритма сопоставления точек, который использовался в рамках выполненного проекта, было разработано два подхода к сопоставлению, оба подхода основаны на том, что исходный спутниковый снимок разбивается на тайлы и представлен в нескольких масштабах. Позже было разработано еще четыре методики сопоставления. Опишем все шесть разработанных методик.

1. Для каждого тайла  $B_i$  (пусть их общее количество равно  $N$ ) выполняется полный перебор всех пар особых точек, найденных на видеокадре и на тайле  $B_i$ . Для каждой точки  $A[j]$  выбирается такая точка  $B_i[number]$ , что  $number = \arg \min_k (eucl\_dist(dA[j], dB_i[k]))$ . Здесь  $dA$  — массив дескрипторов особых точек с видеокадра,  $dB_i$  — аналогичный массив для тайла  $B_i$ ,  $eucl\_dist$  — функция вычисления евклидова расстояния между двумя дескрипторами. Если найденная точка лежит вне рассматриваемой области поиска, которая представляет собой круг [1, 4], то она отбрасывается. Недостатком подхода является то, что если среди тайлов  $B_1 \dots B_N$  есть такой тайл  $B_j$ , что между ним и видеокадром заведомо нет общих особых точек (на них запечатлены непокрываемые фрагмен-

ты поверхности Земли), то сопоставление может быть проведено неверно. Псевдокод алгоритма представлен на рис. 1.

```

для каждого тайла  $B_i$ 
{
  для каждой особой точки  $A[j]$ 
  {
    min_dist = +infinity
    index = 0
    для каждой особой точки  $B_i[k] \leftarrow B_i$ 
    {
      d = eucl_dist( $dA[j]$ ,  $dB_i[k]$ )
      если  $d < \text{min\_dist}$  то
      {
        min_dist = d
        index = k
      }
    }
    если point_in_area( $B_i[\text{index}]$ ) то
      add_to_result( $A[j]$ ,  $B_i[\text{index}]$ )
  }
}

```

Рис. 1. Псевдокод методики 1

Функция point\_in\_area проверяет, попадает ли точка  $B_i[\text{index}]$  в область поиска, функция add\_to\_result добавляет точки  $A[j]$  и  $B_i[\text{index}]$  в массив-результат работы алгоритма.

2. Каждая особая точка из видеокadra сопоставляется со всеми особыми точками со всех тайлов. Такой подход свободен от недостатка, когда в ходе сопоставления используется тайл, с которым у видеокadra с БПЛА заведомо нет пересечения по общим особым точкам. Псевдокод представлен на рис. 2.

```

для каждой особой точки  $A[j]$ 
{
  min_dist = +infinity
  index1 = 0
  index2 = 0
  для каждого тайла  $B_i$ 
  {
    для каждой особой точки  $B_i[k] \leftarrow B_i$ 
    {
      если point_in_area( $B_i[k]$ ) то
      {
        d = eucl_dist( $dA[j]$ ,  $dB_i[k]$ )
        если  $d < \text{min\_dist}$  то
        {
          min_dist = d
          index1 = i
          index2 = k
        }
      }
    }
  }
  add_to_result( $A[j]$ ,  $B_{\text{index1}}[\text{index2}]$ )
}

```

Рис. 2. Псевдокод методики 2

3. Для каждого тайла  $B_i$  выполняется следующее: для всех точек из  $B_i$  выполняется полный перебор всех точек из видеокadra. При этом в ходе сопоставления пропускаются те точки с тайлов, которые лежат вне области поиска. Псевдокод представлен на рис. 3.

```

для каждого тайла  $B_i$ 
{
  для каждой особой точки  $B_i[k] \leftarrow B_i$ 
  {
    если point_in_area( $B_i[k]$ ) то
    {
      min_dist = +infinity
      index = 0
      для каждой особой точки  $A[j]$ 
      {
        d = eucl_dist( $dA[j]$ ,  $dB_i[k]$ )
        если  $d < \text{min\_dist}$  то
        {
          min_dist = d
          index = j
        }
      }
      add_to_result( $A[\text{index}]$ ,  $B_i[k]$ )
    }
  }
}

```

Рис. 3. Псевдокод методики 3

4. Для каждого тайла  $B_i$  выполняется следующее: для множества дескрипторов особых точек, найденных на нем, строится поисковый индекс [16]. Далее, для дескриптора каждой точки из видеокadra находится ближайший к ней дескриптор из  $B_i$  с помощью поискового индекса. После проведения сопоставления с поисковым индексом, отбрасываются те точки, которые лежат вне рассматриваемой области поиска. Псевдокод представлен на рис. 4.

```

для каждого тайла  $B_i$ 
{
  S = make_FLANN_index( $dB_i$ )
  для каждой особой точки  $A[j]$ 
  {
    index = search(S,  $dA[j]$ )
    если point_in_area( $B_i[\text{index}]$ ) то
      add_to_result( $A[j]$ ,  $B_i[\text{index}]$ )
  }
}

```

Рис. 4. Псевдокод методики 4

Функция make\_FLANN\_index строит поисковый индекс, функция search находит для дескриптора  $dA[j]$  наиболее близкий к нему дескриптор с тайла  $B_i$  с помощью поискового индекса. Все поисковые индексы можно построить и сохранить заранее, в процессе выделения особых точек на тайлах пирамиды спутниковых снимков.

5. Эта методика также основана на использовании поискового индекса, но здесь он строится для множества дескрипторов, которое является объединением дескрип-

торов всех особых точек со всех тайлов  $B_1 \dots B_N$ . После проведения сопоставления с поисковым индексом, отбрасываются те точки, которые лежат вне рассматриваемой области поиска в виде круга. Псевдокод представлен на рис. 5.

```

S = make_FLANN_index(dB1, dB2, ... dBN)

для каждой особой точки A[j]
{
    (index1, index2) = search_2(S, dA[j])
    если point_in_area(Bindex1[index2]) то
        add_to_result(A[j], Bindex1[index2])
}

```

Рис. 5. Псевдокод методики 5

Конечно, здесь дополнительные ресурсы тратятся на создание индекса, но данная методика имеет то преимущество, что если для позиционирования БПЛА по текущему кадру требуется использовать те же самые тайлы пирамиды спутниковых снимков, что и для предыдущего кадра, то можно применить тот же самый поисковый индекс для предыдущего кадра.

6. Поисковый индекс строится для множества дескрипторов особых точек, найденных на видеокadre. Затем, для каждого тайла  $B_i$  выполняется следующее: для всех точек из  $B_i$  с помощью индекса находится ближайшая точка из видеокadre (ближайшая — в смысле расстояния между их дескрипторами). Затем отбрасываются те точки с тайла  $B_i$ , которые лежат вне области поиска. Псевдокод представлен на рис. 6.

```

S = make_FLANN_index(dA)

для каждого тайла Bi
{
    для каждой особой точки Bi[k] ← Bi
    {
        index = search(S, dBi[k])
        если point_in_area(Bi[k]) то
            add_to_result(A[index], Bi[k])
    }
}

```

Рис. 6. Псевдокод методики 6

Все шесть методик комбинируются с подходом, взятым из статьи [5] и позволяющим частично устранить пары неправильно найденных точек. Пусть есть два набора дескрипторов особых точек. Для каждого дескриптора  $p_i$  из первого набора находятся две наиболее подходящие ей точки  $q_{j_1}$  и  $q_{j_2}$  из второго набора (исходя из критерия минимизации евклидова расстояния между дескрипторами). Пусть  $d_1 = \text{eucl\_dist}(p_i, q_{j_1})$ ,  $d_2 = \text{eucl\_dist}(p_i, q_{j_2})$ ,  $d_2 < d_1$ . Если  $d_1/d_2 < \text{thresh}$  (где  $\text{thresh}$  — некоторое пороговое значение от 0 до 1), то точка  $p_i$  сопоставляется точке  $q_{j_1}$ , иначе считается, что для точки  $p_i$  найти пару не удалось.

### 3. Способы сравнения разработанных методик

Сравнение методик сопоставления особых точек только по критерию быстродействия является недостаточным, поскольку важнейшим показателем в конечном счете является точность позиционирования БПЛА (погрешность в вычислении положения и ориентации). Точность зависит от качества исходных данных для алгоритма решения задачи внешней калибровки камеры (использовалась реализация из библиотеки OpenCV [17]), которая, в свою очередь, зависит от результатов работы алгоритма поиска опорных точек на видеокadre с помощью вейвлетов Габора (будет описан в следующих публикациях). А результаты его работы зависят от того, насколько точно с помощью 5-точечного алгоритма и методики RANSAC вычислена фундаментальная матрица, связывающая изображения с БПЛА и ИСЗ.

Используемая реализация алгоритма вычисления фундаментальной матрицы позволяет не только найти матрицу  $F$ , но и пометить пары соответствующих точек, которые не удовлетворяют фундаментальной матрице (ложные срабатывания алгоритма сопоставления точек). Эксперименты показали, что если после вычисления матрицы  $F$  остается слишком мало пар точек (как правило, менее 20-25), то матрица  $F$  в подавляющем большинстве случаев оказывается неправильной.

Таким образом, были сформулированы следующие критерии сравнения методик сопоставления особых точек:

1. Время сопоставления особых точек.
2. Количество найденных пар точек.
3. Время вычисления фундаментальной матрицы.
4. Количество пар особых точек, которые после вычисления матрицы  $F$  не были помечены как ложные (так называемые «инлайеры», англ. inliers).
5. Итоговая погрешность в определении координат БПЛА.

В алгоритме SURF, как и во многих других алгоритмах обработки и анализа изображений, исходное изображение рассматривается как дискретная двумерная функция яркости. Одним из описателей особой точки является знак лапласиана  $sign(\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2})$  (положительный или отрицательный). Как утверждают авторы алгоритма, если две особые точки с двух изображений имеют разный знак лапласиана, то они гарантированно не могут являться парой соответствующих точек. Таким образом, множество особых точек (и их дескрипторов) можно разбить на две группы, в зависимости от знака лапласиана, и сопоставлять между собой наборы точек, имеющих один и тот же знак. Было решено реализовать и исследовать этот подход на примере шести предложенных методик сопоставления точек.

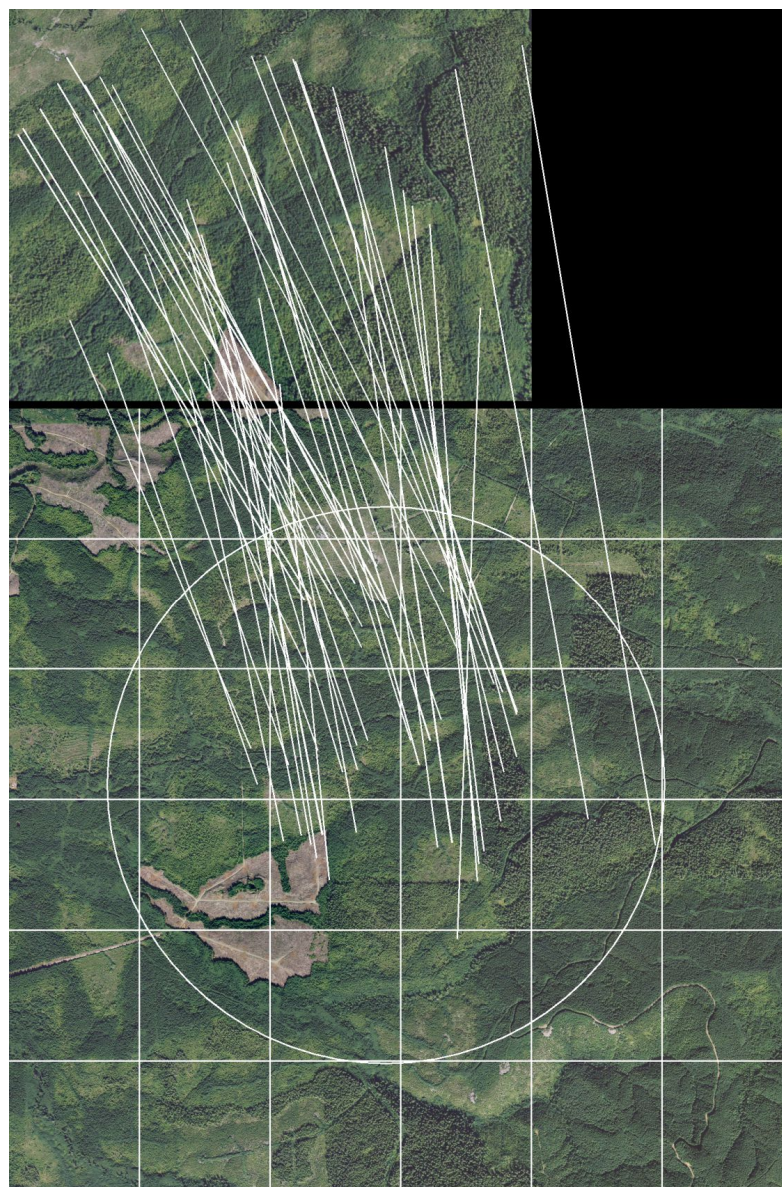
### 4. Экспериментальное сравнение разработанных методик

Для детектирования особых точек и вычисления дескрипторов с помощью алгоритма SURF использовалась библиотека OpenCV, применялась однопоточная версия алгоритма, который выполнялся на одном из ядер процессора Intel Xeon E5410. Разрешение сгенерированных видеокadre с виртуального БПЛА составляло 1024\*768 пикселей. Реализация алгоритма SURF из библиотеки OpenCV находила на снимках с виртуального БПЛА примерно от 2700 до 3600 особых точек (время счета — 1,5–1,7 с). Количество



используемых тайлов одного из слоев пирамиды спутниковых снимков составляло  $6 \cdot 6 = 36$  тайлов, каждый размером  $256 \times 256$  пикселей. Количество особых точек на каждом тайле варьировалось примерно от 50 до 700, зависело от текстуры земной поверхности и от количества деталей на изображениях. Пространственное разрешение спутникового снимка — 1 метр на пиксель.

Рис. 7 иллюстрирует задачу сопоставления особых точек: сверху представлен кадр видеоряда с БПЛА, снизу — часть тайлов одного из слоев спутникового снимка, для наглядности тайлы разделены белыми вертикальными и горизонтальными линиями. Пары соответствующих особых точек на снимке с БПЛА и ИСЗ соединены отрезками.



**Рис. 7.** Сопоставление особых точек с видеокadra и тайлов спутникового снимка

Для тестирования использовались три траектории полета виртуального БПЛА с генерацией видеоряда с учетом рельефа Земли (карты высот), а также три аналогичных траектории, но с генерацией видеоряда при полете над плоской местностью:

1. Горизонтальный полет в северном направлении на постоянной высоте над плоской местностью.

2. Горизонтальный полет в северном направлении на постоянной высоте над местностью с рельефом.
3. Горизонтальный полет в северо-восточном направлении над плоской местностью. Здесь можно тестировать инвариантность алгоритма SURF к поворотам изображений.
4. Горизонтальный полет в северо-восточном направлении над местностью с рельефом.
5. Полет в северо-восточном направлении с набором высоты над плоской местностью. Здесь можно тестировать инвариантность алгоритма SURF к поворотам, масштабированию и проективным преобразованиям.
6. Полет в северо-восточном направлении с набором высоты над местностью с рельефом. На рис. 7 как раз и приведен результат сопоставления одного из кадров такого видеоряда с тайлами спутникового снимка.

Сразу отметим, что методика под номером 4 оказалась неспособной корректно сопоставлять особые точки: время сопоставления было слишком велико (3 секунды и более для одного кадра видеоряда), она находила слишком много ложных соответствий, что негативно сказывалось на времени вычисления фундаментальной матрицы (несколько секунд и даже несколько десятков секунд для одного кадра). Результаты вычисления матрицы  $F$  во всех случаях оказывались весьма далеки от эталонных значений, что делало невозможным решение задачи визуальной навигации.

Во всех остальных случаях задача позиционирования решалась довольно успешно, погрешность в вычислении положения БПЛА составляла от 1,5 до 3,3 метра (за исключением ряда случаев).

В табл. 1 приведено сравнение пяти методик, для каждой из шести тестовых траекторий полета. По времени сопоставления наиболее эффективны методики 5 и 6 (основанные на использовании поисковых индексов), затем идут методики 2 и 3, наименее эффективна методика 1. По времени вычисления матрицы  $F$  наиболее эффективны методики 2 и 3, затем идут методики 1 и 5, наименее эффективна методика 6. Но в целом, для всех подходов характерен недостаток, что в случае наличия значительных проективных искажений (тест №6) между эталонными данными (спутниковый снимок) и тестовыми (видеокадр с БПЛА) удается найти недостаточно пар «инлайеров», что не позволяет решить задачу позиционирования.

Перейдем к сравнению предложенных методик в случае, когда особые точки с видеокadra разбивались на две группы, в зависимости от знака лапласиана (табл. 2). Процессорное время, необходимое на разбиение, оказалось невелико (не более 1 миллисекунды), поэтому это не повлияло на быстродействие всех шести методик. По времени сопоставления наиболее эффективны методики 5 и 6 (основанные на использовании поисковых индексов), затем идут методики 2 и 3, наименее эффективна методика 1. По времени вычисления матрицы  $F$  наиболее эффективны методики 2 и 3, затем идут методики 1 и 5, наименее эффективна методика 6. Как видим, разбиение множества особых точек на два подмножества по знаку лапласиана позволило повысить количество найденных пар примерно в 1,5–2 раза, увеличилось и количество найденных пар «инлайеров», что повысило точность вычисления фундаментальной матрицы (но даже в этом случае методика 6 дала сбой).

Таблица 1

Сравнение методик сопоставления особых точек

Номер методики	Время сопоставления, с	Кол-во найденных пар точек	Время вычисления матрицы $F$ , с	Кол-во пар точек-инлайеров	Примечание
1	1,0–1,25	632–654 403–468 200–225 171–191 132–142 120–140	0,12–0,27 0,28–0,55 0,21–0,51 0,70–0,95 0,05–0,33 0,12–0,28	400–430 200–240 60–83 49–61 25–46 14–29	В тесте №6 инлайеров оказалось мало, ошибка в позиционировании составила 19 м.
2	0,7–1	486–510 247–292 80–102 54–69 38–47 24–33	0,06–0,21 0,08–0,17 0,06–0,10 0,03–0,07 0,02–0,04 0,02–0,05	345–384 127–164 45–54 30–35 20–26 15–21	
3	0,50–0,75	515–538 273–309 100–134 70–80 34–46 25–38	0,08–0,15 0,09–0,19 0,06–0,18 0,10–0,14 0,04–0,07 0,03–0,05	326–403 162–188 42–66 30–35 18–26 12–22	В тесте №6 инлайеров оказалось мало, ошибка в позиционировании составила 8 м.
5	0,15–0,18 (без учета построения индекса поиска — 0,09–0,11)	640–659 420–484 238–240 185–199 129–160 126–148	0,21–0,31 0,26–0,46 0,41–0,49 0,17–1,10 0,02–0,28 0,13–0,31	393–476 195–242 19–102 48–68 33–50 14–34	В тестах №3 и №6 инлайеров оказалось мало, вычислить положение БПЛА не удалось.
6	0,14–0,18	714–738 503–532 279–321 241–276 150–182 159	0,29–0,36 0,28–0,48 0,62–1,59 0,20–1,58 0,14–1,75 0,04	387–425 235–265 96–109 53–81 40–56 12	В тесте №6 инлайеров оказалось мало, вычислить положение БПЛА не удалось.

Таблица 2

Сравнение методик сопоставления особых точек с учетом знака лапласиана

Номер методики	Время сопоставления, с	Кол-во найденных пар точек	Время вычисления матрицы $F$ , с	Кол-во пар точек-инлайеров	Примечание
1	1,0–1,50	1287–1313 844–912 386–419 312–341 214–227 188–203	0,23–0,39 0,39–0,82 0,24–0,92 0,70–1,00 0,25–0,65 0,30–1,10	737–942 391–513 141–174 114–130 72–88 66–78	
2	0,75–1,00	1050–1083 580–638 190–211 133–159 83–105 64–87	0,15–0,39 0,22–0,28 0,13–0,19 0,08–0,14 0,06–0,10 0,03–0,14	732–816 308–372 93–110 65–90 39–53 29–43	
3	0,50–0,75	1103–1132 600–670 218–254 157–175 78–96 54–77	0,11–0,50 0,24–0,50 0,10–0,33 0,11–0,22 0,08–0,15 0,06–0,10	637–894 304–400 99–127 77–85 36–46 29–36	
5	0,2–0,23 (без учета построения индекса поиска — 0,09–0,11)	1316–1352 900–976 423–489 356–357 248–264 227–243	0,28–0,51 0,37–0,62 0,57–1,06 0,15–1,33 0,40–1,09 0,05–1,29	785–1006 424–527 155–211 126–143 82–102 71–83	
6	0,26–0,31	1523–1566 1099–1149 616–674 505–547 349–366 296–338	0,40–0,81 0,64–1,03 0,80–2,63 2,40–3,20 0,40–5,68 1,86–3,88	828–973 515–562 186–229 163–175 73–109 22–93	В тесте №6 инлайеров оказалось мало, вычислить положение БПЛА не удалось.

## Заключение

Экспериментальное сравнение разработанных методик сопоставления особых точек с двух изображений, одно из которых разделено на фрагменты, показало, что эффективность и производительность их реализаций зависит от исходных данных: количества особых точек, вида проективного преобразования, которое связывает два изображения (параллельный перенос, поворот, масштабирование), а также наблюдаемой сцены (является ли она плоской или имеет перепады высот). Также оправдал себя подход, основан-

ный на разбиении множества особых точек, найденных алгоритмом SURF, на две группы (в зависимости от знака лапласиана).

Методики под номерами 1, 2, 3 и 5 показали свою эффективность, хотя на данном этапе производительность реализаций методик 1 и 5 не позволяет использовать их в режиме времени, близкого к реальному. В дальнейших экспериментах предполагается опробовать следующие способы увеличения производительности всех методик:

1. Использовать различных значений порогового значения *thresh* (в текущих экспериментах использовалось значение 0,65).
2. Провести эксперименты с параметрами алгоритма SURF для уменьшения времени поиска особых точек.
3. Искусственно уменьшить количества точек, которые подаются на вход алгоритму вычисления фундаментальной матрицы. Слишком большое количество избыточно, оно увеличивает время счета, точность позиционирования при этом почти не меняется. Возможно, стоит из всех пар соответствующих точек в количестве  $P$  выбрать случайные  $Q$  пар, где  $Q$  может быть фиксированным (например, 100–150), или же зависеть от числа особых точек на видеокadre с БПЛА или от числа  $P$ .
4. Использовать возможности векторизации кода и распараллеливания по вычислительным ядрам. Конкретные способы увеличения быстродействия зависят от используемого вычислителя, который будет использоваться на борту БПЛА.

## Литература

1. Кабмин РФ предложил изменить законодательство для использования БПЛА. URL: <http://ria.ru/politics/20150217/1048089808.html> (дата обращения: 14.07.2015).
2. Correlation based similarity measures — Summary. URL: <https://siddhantahuja.wordpress.com/tag/sum-of-absolute-differences-sad> (дата обращения: 14.07.2015).
3. Harris, C. A combined corner and edge detector / C. Harris, M. Stephens // Proceedings of the 4th Alvey Vision Conference — 1988 — P. 147—151. DOI: 10.5244/c.2.23.
4. Гаганов, В. Инвариантные алгоритмы сопоставления точечных особенностей на изображениях / В. Гаганов // Компьютерная графика и мультимедиа — 2009. Выпуск №7(1) — URL: [http://cgm.computergraphics.ru/issues/issue17/invariant\\_features](http://cgm.computergraphics.ru/issues/issue17/invariant_features) (дата обращения: 06.07.2015)
5. Lowe, D.G. Distinctive image features from scale-invariant keypoints / D.G. Lowe // International Journal of Computer Vision — 2004 — Vol. 60, Issue 2 — P. 91–110. DOI: 10.1023/b:visi.0000029664.99615.94.
6. Bay, H. SURF: Speeded Up Robust Features / H. Bay, A. Ess, T. Tuytelaars, L Van Gool // Computer Vision and Image Understanding (CVIU) — 2008. — Vol. 110, №. 3 — 14 p. DOI: 10.1016/j.cviu.2007.09.014.
7. Mikolajczyk, K. A performance evaluation of local descriptors / K. Mikolajczyk, C. Schmid // IEEE Transactions on Pattern Analysis and Machine Intelligence — 2005 — Vol. 10, №. 27 — P. 1615-1630. DOI: 10.1109/tpami.2005.188.
8. Bouguet, J. Y. Pyramidal implementation of the Lucas Kanade feature tracker / J. Y. Bouguet // Intel Corporation, Microprocessor Research Labs — 2000 — 9 p.

9. Muja, M. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration / M. Muja, D.G. Lowe // International Conference on Computer Vision Theory and Application VISSAPP'09 — 2009 — P. 331-340
10. Степанов, Д.Н., Тищенко, И.П., Поляков, А.В., Ватутин, В.М., Соболев, Д.Б. Подсистема определения положения и ориентации беспилотного летательного аппарата / Д.Н. Степанов, И.П. Тищенко, А.В. Поляков, В.М. Ватутин, Д. Б.Соболев, // Ракетно-космическое приборостроение и информационные технологии. 2012: Труды V Всероссийской научно-технической конференции «Актуальные проблемы ракетно-космического приборостроения и информационных технологий» (Москва, 5-7 июня 2012 г.) / под ред. Ю.М. Урличича, А.А. Романова. — М.: Радиотехника, 2013. - С. 9–27.
11. Kamarainen, J.-K. Invariance Properties of Gabor Filter Based Features — Overview and Applications / J.-K. Kamarainen, V. Kyrki, H. Kälviäinen // IEEE Transactions on Image Processing — 2006. — Vol. 15, No. 5 — 12 p. DOI: 10.1109/tip.2005.864174.
12. Степанов, Д.Н. Методы и алгоритмы определения положения и ориентации беспилотного летательного аппарата с применением бортовых видеокамер / Д.Н. Степанов // Программные продукты и системы (международный журнал) — 2014. — Т. 1, № 1. URL: <http://www.swsys.ru/index.php?page=article&id=3776> (дата обращения: 06.07.2015).
13. Stewenius. H. Recent developments on direct relative orientation / H. Stewenius, C. Engels, D. Nister. // ISPRS Journal of Photogrammetry and Remote Sensing — 2006. — Vol. 60, Issue 4 — P. 284–294. DOI: 10.1016/j.isprsjprs.2006.03.005.
14. Fischler, M.A. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography / M.A. Fischler, R.C. Bolles // Comm. Of the ACM 24 — 1981. - P. 381–395. DOI: 10.1145/358669.358692.
15. Кравцов, А. Общая формулировка задачи внешней калибровки камеры / А. Кравцов, В. Вежневек // Компьютерная графика и мультимедиа — 2003. — Выпуск №1(2) URL: <http://cgm.computergraphics.ru/content/view/34> (дата обращения: 06.07.2015).
16. FLANN — Fast Library for Approximate Nearest Neighbors. URL: <http://www.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN> (дата обращения: 06.07.2015).
17. Bradski, G. A. Learning OpenCV / G. Bradski., A. Kaehler // O'Reilly Media — September 2008 — 576 p.

Степанов Дмитрий Николаевич, инженер-исследователь, Исследовательский центр мультипроцессорных систем Института программных систем им. А.К. Айламазяна РАН (Переславль-Залесский, Российская Федерация), [mitek1989@mail.com](mailto:mitek1989@mail.com).

*Поступила в редакцию 8 июля 2015 г.*

DOI: 10.14529/cmse150402

## TECHNIQUES OF FEATURE POINTS MATCHING IN THE PROBLEM OF UAV'S VISUAL NAVIGATION

*D.N. Stepanov*, Aylamazyan Program Systems Institute of the Russian Academy of Sciences (Pereslavl-Zalessky, Russia) mitek1989@mail.com

The paper is devoted to development and experimental comparison of techniques feature points matching on the images — images of the earth's surface with cameras mounted on unmanned aerial vehicles (UAVs) and artificial earth satellite. The main feature of the problem is that one of the images (satellite image) is divided into fragments. The developed techniques are part of a complex of algorithms for determining the position and orientation of the UAV using the methods and algorithms of machine vision. A description of the flight simulation technology and solving positioning tasks are described. Feature points on the image are extracted using an SURF algorithm. Also the approach to matching based on the partition of feature points' set into two subsets depending on the sign of the Laplacian are studied. Methods of increasing the performance of matching points are offered.

*Keywords:* UAV, feature points, SURF, computer vision, satellite images, brute-force, search index, image matching.

### References

1. Kabmin RF predlozhl izmenit' zakonodatel'stvo dlja ispol'zovanija BPLA [The Cabinet of Ministers of the Russian Federation proposed to change the law for the use of UAVs]. URL: <http://ria.ru/politics/20150217/1048089808.html> (accessed: 14.07.2015)
2. Correlation based similarity measures — Summary. URL: <https://siddhantahuja.wordpress.com/tag/sum-of-absolute-differences-sad> (accessed: 14.07.2015).
3. Harris C., Stephens M. A combined corner and edge detector // Proceedings of the 4th Alvey Vision Conference. 1988. P. 147—151. DOI: 10.5244/c.2.23.
4. V. Gaganov. Invariantnye algoritmy sopostavlenija tochechnyh osobennostej na izobrazhenijah [Invariant algorithms of feature points' matching on images] // Computer graphics and multimedia. 2009. Vol. 7, No 1. URL: [http://cgm.computergraphics.ru/issues/issue17/invariant\\_features](http://cgm.computergraphics.ru/issues/issue17/invariant_features) (accessed: 14.07.2015).
5. D.G. Lowe, Distinctive image features from scale-invariant keypoints // International Journal of Computer Vision. 2004. Vol. 60, Issue 2. P. 91–110. DOI: 10.1023/b:visi.0000029664.99615.94.
6. Bay H., Ess A., Tuytelaars T., Van Gool L. SURF: Speeded Up Robust Features // Computer Vision and Image Understanding (CVIU). 2008. Vol. 110, №. 3. 14 p. DOI: 10.1016/j.cviu.2007.09.014.
7. Mikolajczyk K., Schmid C. A performance evaluation of local descriptors // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2005. Vol. 10, №. 27. P. 1615-1630. DOI: 10.1109/tpami.2005.188.

8. Bouguet J. Y. Pyramidal implementation of the Lucas Kanade feature tracker // Intel Corporation, Microprocessor Research Labs. 2000. 9 p.
9. Muja M., Lowe D.G. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration // International Conference on Computer Vision Theory and Application VISSAPP'09. 2009. P. 331–340
10. Stepanov D.N., Tishchenko I.P., Poljakov A.V., Vatutin V.M., Sobolev D.B. Podsystema opredelenija polozhenija i orientacii bespilotnogo letatel'nogo apparata [The subsystem determine the position and orientation of the UAV]. 5 Vserossijskaja nauchno-tehnicheskaja konferencija “Aktual'nye problemy raketno-kosmicheskogo priborostroenija i informacionnyh tehnologij”, OAO «Rossijskie kosmicheskie sistemy» (Moskva, 5–7 ijunja, 2012) [Proc. of 5th All-Russian Scientific Conference “Actual problems of missile and space instrumentation and information technology”, Joint Stock Company “Russian Space Systems” (Moscow, 5–7 june, 2012)], Moscow, Radiotekhnika, 2013, P. 9–27
11. Kamarainen J.-K., Kyrki V., Kälviäinen H. Invariance Properties of Gabor Filter Based Features - Overview and Applications // IEEE Transactions on Image Processing. 2006. Vol. 15, No. 5. 12 p. DOI: 10.1109/tip.2005.864174.
12. Stepanov D.N. Metody i algoritmy opredelenija polozhenija i orientacii bespilotnogo letatel'nogo apparata s primeneniem bortovyh videokamer [Methods and algorithms for determining the position and orientation UAV usage of on-board videocameras]. Programmnye produkty i sistemy (International Journal). 2014. Vol. 1, No. 1. URL: <http://www.swsys.ru/index.php?page=article&id=3776> (accessed: 06.07.2015).
13. Stewenius H., Engels C., Nister D. Recent developments on direct relative orientation // ISPRS Journal of Photogrammetry and Remote Sensing. 2006. Vol. 60, Issue 4. P. 284–294. DOI: 10.1016/j.isprsjprs.2006.03.005.
14. Fischler M.A., Bolles R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography // Comm. Of the ACM 24. 1981. P. 381–395. DOI: 10.1145/358669.358692.
15. A. Kravcov, V. Vezhnevec. Obshhaja formulirovka zadachi vneshnej kalibrovki kamery [General formulation of the problem of extrinsic camera calibration] // Computer graphics and multimedia. 2003. Vol. 1, No 2. URL: <http://cgm.computergraphics.ru/content/view/34> (accessed: 06.07.2015).
16. FLANN – Fast Library for Approximate Nearest Neighbors. URL: <http://www.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN> (accessed: 06.07.2015).
17. Bradski G., Kaehler A. Learning OpenCV. O'Reilly Media. September 2008. 576 p.

*Received July 8, 2015.*