

УДК 004.023, 004.921, 004.932.72'1

## **ПРОГРАММНАЯ СИСТЕМА ДЛЯ МОДЕЛИРОВАНИЯ ПОЛЕТА БПЛА И ЕГО АВТОНОМНОГО ПОЗИЦИОНИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ И АЛГОРИТМОВ ТЕХНИЧЕСКОГО ЗРЕНИЯ**

Д.Н. Степанов (*mitek1989@mail.ru*)

И.П. Тищенко (*billy@billy.botik.ru*)

Институт программных систем

им. А.К. Айламазяна РАН, Переславль-Залесский

В работе описывается программная система, предназначенная для моделирования полета беспилотного летательного аппарата с помощью средств, методов и алгоритмов технического зрения. Представлена общая схема генерации и обработки данных, а также краткое описание модулей программной системы, предназначенные для решения отдельных подзадач.

### **Введение**

Задачи, которые ставятся перед БПЛА, обусловлены их многочисленными приложениями в военной и в гражданской сферах (картографирование, сельское хозяйство, охранные мероприятия, мониторинг и т. д.). Создание надежных систем навигации и управления БПЛА - важное направление разработок. Стабильность движения большинства БПЛА существенно зависит от качества сигналов спутниковой связи GPS/ГЛОНАСС. Общий недостаток любой радионавигационной системы - при определенных условиях сигнал может не доходить до приёмника, или приходиться со значительными искажениями или задержками (городские постройки, рельеф местности, облачность). Кроме того, особенность GPS - зависимость условий получения сигнала от Минобороны США. Инерциальные навигационные микромеханические системы (ИНМС) способны работать автономно, но их использование неизбежно связано с проблемой накопления ошибки в вычислениях, что с течением времени приводит к большим погрешностям.

Поэтому создание системы технического зрения, которая позволила бы меньше зависеть от спутниковой навигации и ИНМС, является

актуальной, перспективной и важной научной задачей, и востребована разработчиками и производителями БПЛА. Имеется ряд эффективных методов компьютерной графики, искусственного интеллекта и параллельных вычислений, которые могут быть использованы в качестве интеллектуальной основы при разработке методов и алгоритмов решения задачи автономного позиционирования с применением фото- или видеокамер.

Коллектив исследователей ИПС РАН участвовал в научно-исследовательской работе «Разработка алгоритмического, математического и программного обеспечения системы моделирования автономного позиционирования БПЛА (аппаратно-программного комплекса) на основе технологий машинного зрения, искусственного интеллекта и параллельных вычислений». В рамках проекта был создан прототип программной системы, предназначенной для моделирования полета БПЛА и его автоматического позиционирования при полете над плоской местностью. В настоящий момент разрабатывается и реализуется система моделирования полета аппарата и позиционирования при полете аппарата над местностью с рельефом.

## **1. Постановка задачи и общее описание программной системы**

Постановка задачи заключается в следующем: задан спутниковый снимок местности, над которой БПЛА, оснащенный фото- или видеокамерой, совершает свои полеты. Для кадров с камеры на БПЛА (она направлена вниз) необходимо определить, где в эти моменты находился аппарат, и возможно, определить ориентацию аппарата относительно Земли.

В основе решения задачи лежит использование искусственного спутника Земли (ИСЗ) для тех участков местности, над которой БПЛА совершает свои полеты. Снимок имеет геопривязку. На любом снимке с БПЛА будет запечатлен некоторый фрагмент снимка со спутника. Вычисляются параметры проективного преобразования, которое связывает координаты ряда точек, наблюдаемых на обоих снимках, что позволяет рассчитать положение и ориентацию камеры на БПЛА сначала относительно положения камеры на спутнике, а затем в глобальной системе координат.

Эти же спутниковые снимки используются для моделирования полета. Если необходимо учитывать рельеф местности и кривизну поверхности Земли, в дополнение к спутниковому снимку используется карта высот. Это псевдоизображение, в котором яркость каждого пикселя – высота соответствующей точки над уровнем моря.

Программная система функционирует на кластерной вычислительной (КВУ), которая состоит из управляющей машины (фронтэнда) и

вычислительных узлов, КВУ находится под управлением одного из вариантов ОС Linux. Главные особенности системы: обработка данных в так называемом конвейерно-параллельном режиме и модульная архитектура [Тищенко и др., 2012]. Вычислительное ядро отвечает за решение задач планирования и распределения нагрузки, основная же функциональность заложена в модулях, каждый из которых может иметь произвольное назначение. Модульность позволяет легко расширять функциональность системы для решения самых разных задач, главное требование: модули, образующие логическую схему обработки, должны подходить друг другу по типу входных и выходных данных.

Рассмотрим разработанные программные модули, которые функционируют в рамках программной системы и предназначены для решения отдельных подзадач.

## **2. Подготовка исходных данных**

Эта задача решается один раз, если у нас зафиксирован спутникового снимка, который используется для моделирования полета и позиционирования. Для спутникового снимка строится так называемая пирамида изображений: в основании лежит исходный снимок, а каждый следующий слой по ширине и высоте в 2 раза меньше предыдущего. Все изображения нарезаются на квадратные фрагменты одинакового размера. На каждом фрагменте с помощью алгоритма SURF [Bay et al., 2008] находятся так называемые особые точки, для каждой точки вычисляется ее дескриптор (набор вещественных чисел). Причина разбиения снимка на фрагменты и построение пирамиды изображений будет обоснована далее.

## **3. Генерация траектории**

Создание реалистичных траекторий полета не является первоочередной задачей исследований, поэтому было решено поступить следующим образом: пользователь отмечает на уменьшенной копии спутникового снимка ряд опорных точек, которые соединяются прямыми отрезками. На полученной цепной линии вычисляются координаты ряда точек  $A_i$  (соседние точки равноудалены друг от друга). Этим точкам будут соответствовать те моменты времени, в которые камера на виртуальном БПЛА выполняет съемку. Также для каждой точки вычисляется ориентация «беспилотника» в глобальной системе координат (навигационные углы: тангаж, крен и рысканье). Если учитывать кривизну поверхности Земли, то эти отрезки будут прямыми только в криволинейной системе координатах, которая определяется поверхностью земного эллипсоида. В геоцентрических (прямоугольных) координатах это будут кривые линии с достаточно сложными уравнениями.

Решение задачи можно разделить на два этапа. Первый – вычисление расстояния между двумя соседними опорными точками (вычисление длины кривой линии). Математически задача сводится к вычислению определенного интеграла от функции, первообразная которой не выражается в элементарных функциях, и поэтому расстояние проще найти приближенно, разделив кривую линию на множество прямых отрезков (т.е. используя аналог одного из численных методов вычисления определенного интеграла). На втором этапе кривая линия делится на некоторое количество частей, зависящее от ее длины и расстояния, которое пролетает БПЛА между моментами съемки двух соседних кадров видеоряда. Затем вычисляются координаты всех промежуточных точек.

#### **4. Порождение видеоряда**

Каждой точке  $A_i$  сопоставляется свой кадр видеоряда. Задача сводится к перспективному преобразованию исходного спутникового снимка с помощью так называемой матрицы гомографии, которая вычисляется, исходя из положения и ориентации камеры на БПЛА, а также параметров камер на ИСЗ и БПЛА. При этом виртуальная камера располагается так, что при нулевом угле тангажа БПЛА камера направлена перпендикулярно поверхности земного эллипсоида.

Учет рельефа местности и кривизны Земли обуславливает применение более продвинутых средств: систем трехмерного моделирования (например, `osgEarth`), предназначенные для визуализации 3D-моделей. Пользователь с помощью мыши или клавиатуры позиционирует виртуальную камеру, которая «парит» над поверхностью Земли (пример показан на рис. 1). В нашем случае, нет необходимости отображать сгенерированные снимки, более предпочтительно проводить так называемый “`offscreen rendering`” - процесс, в результате которого сгенерированное изображение не выводится на экран компьютера, а остается в оперативной памяти.

Чаще всего пользователи работают с КВУ в удаленном режиме посредством клиентских ПК и сетевых протоколов взаимодействия. Компьютер клиента служит только для ввода/вывода информации, а все вычисления проводятся на сервере. Серия экспериментов с `osgEarth` показали, что обеспечить корректный рендеринг трехмерных сцен в удаленном режиме удастся не всегда. Это связано с самим принципом работы графических приложений и недостаточной развитостью программных средств, которые предназначены для отображения интерфейсов программ, работающих в удаленном режиме. Вторая причина состоит в том, что узлы кластеров если и оснащаются видеокартами, то специализирующимися именно на вычислениях, а не на отображении графики (например, серия видеокарт `NVIDIA Tesla`).

Поэтому предложен следующий подход: рендеринг трехмерных сцен, происходит на пользовательской машине, а КВУ производит все остальные вычисления. КВУ посылает на ПК набор чисел, которые задают положение и ориентацию камеры в пространстве. Компьютер пользователя производит рендеринг сцены и отправляет сгенерированное изображение на КВУ. Связь КВУ и ПК можно осуществить с применением стандартных TCP-сокетов. При таком подходе КВУ выступает в качестве клиента, а ПК – в качестве сервера. Копии одного и того же спутникового снимка хранятся и там, и там. Пример сгенерированного кадра видеоряда представлен на рис. 1.

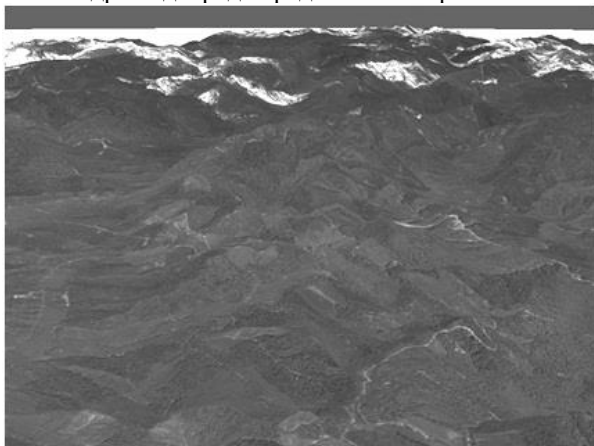


Рис.1. Пример сгенерированного кадра видеоряда

## **5. Поиск пар общих точек на снимках с БПЛА и ИСЗ**

Задан кадр видеоряда, на нем с помощью алгоритма SURF находятся особые точки и вычисляются их дескрипторы. Далее они сопоставляются с особыми точками со спутникового снимка (в качестве метрики используется евклидово расстояние между векторами-дескрипторами). Поскольку для предыдущего кадра видеоряда уже вычислено положение и ориентация в глобальной системе координат, это позволяет выполнять сопоставление с особыми точками не со всего спутникового снимка (размер которого – от нескольких сотен до нескольких десятков тысяч мегапикселей), а только с некоторого подмножества его фрагментов с определенного слоя пирамиды, который соответствует некоторой ориентировочной высоте полета БПЛА. Для выбора нужных фрагментов используется разработанный алгоритм сужения области поиска [Степанов, 2014].

## 6. Поиск пар общих точек на снимках с БПЛА и ИСЗ

Имеется множество пар координат общих особых точек со снимков с БПЛА и ИСЗ. В случае моделирования полета над плоской местностью и без учета кривизны Земли, оба снимка являются проективными преобразованиями плоскости, и поэтому координаты пар особых точек связаны матрицей гомографии  $H$ . Для вычисления матрицы  $H$  необходимо как минимум 4 пары точек, использование большего количества точек позволяет повысить устойчивость вычислений (с применением метода наименьших квадратов). Но поскольку алгоритм сопоставления точек также основан на эвристиках, это может стать причиной зашумленности данных. Существуют стабильные методы оценки параметров некоторой модели на основе случайных выборок, позволяющие уменьшить влияние шума, один из них носит название RANSAC [Zuliani, 2008]. Далее производится декомпозиция матрицы  $H$  на матрицу поворота (она определяет ориентацию камеры на БПЛА в системе координат, привязанных к ИСЗ) и вектор параллельного переноса (он определяет положение). Зная положение и ориентацию камеры на ИСЗ в глобальной системе координат, легко вычислить, где находится и как ориентирована камера на БПЛА.

В общем случае, когда на двух снимках запечатлена одна и та же сцена, не являющаяся плоской, координаты точек на снимках одной и той же сцены с двух разных камер в разных ракурсах связаны так называемой фундаментальной матрицей  $F$ . Существуют различные алгоритмы вычисления матрицы  $F$ , обзор показал, что наиболее устойчивым является пятиточечный алгоритм. Кроме того, он корректно работает в той ситуации, когда все пять выбранных пар точек располагаются на одной плоскости, другие алгоритмы в такой ситуации выдают некорректные результаты. Как и в случае с матрицей гомографии, для уменьшения влияния шума целесообразно использовать данный алгоритм в паре с одним из стабильных методов оценки параметров некоторой модели на основе случайных выборок (например, RANSAC).

Фундаментальную матрицу также можно разложить на матрицу поворота и вектор параллельного переноса (они определяют взаимное положение и ориентацию камер). Но вектор будет определен только с точностью до некоторого положительного коэффициента  $k$ . Для вычисления  $k$  необходимо знать трехмерные координаты хотя бы одной точки, образ которой есть на снимке с БПЛА. Координаты этой точки на спутниковом снимке известны. Для повышения устойчивости можно использовать сразу несколько таких точек, применяя связку «метод наименьших квадратов + RANSAC» для вычисления  $k$ .

В общем случае, проекция каждой такой точки может располагаться на всем снимке с БПЛА, но наличие фундаментальной

матрицы позволяет использовать так называемые эпиполярные ограничения: поиск образов точек на снимке с БПЛА следует производить вдоль эпиполярных линий, уравнение которых зависит от матрицы  $F$ . Поскольку матрица  $F$  также может быть вычислена с некоторой погрешностью, предлагается рассматривать те пиксели снимка с БПЛА, которые располагаются на эпиполярной линии и на расстоянии в несколько пикселей от нее.

Далее встает вопрос: если известны координаты образа некоторой точки на снимке со спутника, то как найти образ этой точки на снимке с БПЛА? Необходимо выбрать некоторую метрику, которая позволяет сравнивать точки с двух изображений. Анализ источников показал, что одним из наиболее эффективных методов являются вейвлеты (фильтры) Габора [Kamagainen et al., 2006]. Рассматривается область некоторого радиуса вокруг точки, производится цифровая свертка этой области с 40 различными масками Габора, в итоге имеем 40-мерный вектор вещественных чисел – дескриптор точки. В качестве метрики для сравнения двух дескрипторов предлагается использовать косинус угла между векторами: чем он ближе к единице, тем более точки с двух изображений «похожи» друг на друга. Фильтры Габора обладают свойством инвариантности к аффинным преобразованиям изображения и изменениям яркости.

На этапе подготовки исходных данных, на каждом из фрагментов из каждого слоя пирамиды спутниковых снимков выбирается один случайный пиксель, с помощью семейства фильтров Габора вычисляется 40-мерный дескриптор пикселя, а наличие карты высот позволяет вычислить трехмерные координаты точки, чей образ находится в этом пикселе. Далее вся информация сохраняется для дальнейшего использования.

## **7. Использование GPU-вычислений в рамках программной системы**

В последние несколько лет всё более актуальным становится перенос различных алгоритмов на графические процессорные устройства, GPU позволяют добиться существенного ускорения в задачах с так называемым мелкогранулярным массовым параллелизмом (иногда в десятки и сотни раз), хотя и имеют свою специфику написания программ для них. Существуют различные программно-аппаратные платформы параллельных GPU-вычислений, наиболее популярной является CUDA.

В разрабатываемой программной системе есть несколько алгоритмов, для которых возможны эффективные GPU-аналоги. Например, для алгоритма выделения особых точек SURF существует несколько реализаций на CUDA. Другой алгоритм – поиск образа точки на снимке с

БПЛА с известными ее координатами на снимке с ИСЗ. Поиск затрагивает эпиполярную линию и все близлежащие пиксели. Каждая вычислительная GPU-нить обрабатывает свой пиксель, принадлежащий снимку с БПЛА, и в качестве результата выдает «расстояние» (по описанной выше метрике) между двумя дескрипторами, один из которых описывает пиксель с БПЛА-снимка, а второй – пиксель с ИСЗ-снимка.

### **Заключение**

Разработаны и реализованы модули программной системы, отвечающие за моделирование полета виртуального БПЛА и позиционирования при полете аппарата над местностью с рельефом. На GPU реализован алгоритм поиска образа некоторой точки на снимке с БПЛА при известных ее координатах на снимке с ИСЗ. Технология обработки данных включает в себя удаленный персональный компьютер, который используется для порождения видеоряда с борта виртуального БПЛА.

В ближайшее время планируется провести всестороннее тестирование как отдельных модулей, так и всей схемы обработки данных (моделирование и позиционирование). В случае положительных результатов, полученные наработки могут быть применимы для реализации программно-аппаратных комплексов, предназначенных для решения задач автономного позиционирования БПЛА, а также других подвижных технических средств.

### **Список литературы**

- [**Степанов, 2014**] Степанов Д.Н. Методы и алгоритмы определения положения и ориентации беспилотного летательного аппарата с применением бортовых видеокамер // Программные продукты и системы (международный журнал): 2014, Т. 1, № 1. - <http://www.swsys.ru/index.php?page=article&id=3776>
- [**Тищенко и др., 2012**] Тищенко И.П., Степанов Д.Н., Фраленко В.П. Разработка системы моделирования автономного полета беспилотного летательного аппарата // Программные системы: теория и приложения. – электрон. научн. журн.: 2012, Т. 3, № 3(12). - [http://psta.psisras.ru/read/psta2012\\_3\\_3-21.pdf](http://psta.psisras.ru/read/psta2012_3_3-21.pdf)
- [**Bay et al., 2008**] Bay H., Ess A., Tuytelaars T., Van Gool L. SURF: Speeded Up Robust Features // Computer Vision and Image Understanding (CVIU): 2008, Vol. 110, №. 3
- [**Kamarainen et al., 2006**] Kamarainen J.-K., Kyrki V., Kälviäinen H. Invariance Properties of Gabor Filter Based Features - Overview and Applications // IEEE Transactions on Image Processing: Vol. 15, No. 5, 2006
- [**Zuliani, 2008**] Zuliani M. RANSAC for Dummies // 2008