

И. П. Тищенко, Д. Н. Степанов, В. П. Фраленко

## Разработка системы моделирования автономного полета беспилотного летательного аппарата

Аннотация. В работе приводится описание системы моделирования полета беспилотного летательного аппарата (БПЛА) в автономном режиме. В своей работе система ориентируется на генерируемые программно снимки, являющиеся аналогом снимков, сделанных с бортовых камер БПЛА. Производится распознавание объектов.

*Ключевые слова и фразы:* система моделирования, БПЛА, распознавание.

### Введение

Разработанная коллективом исследователей лаборатории интеллектуального управления исследовательского центра мультипроцессорных систем ИПС им. А.К. Айламазяна РАН программная система предназначена для моделирования полета беспилотного летательного аппарата в автономном режиме. Ее важность и актуальность обусловлены тем, что современные БПЛА, широко применяемые в военном деле, мониторинге лесных пожаров, составлении топографических карт и других областях, для ориентации используют данные таких систем как GPS или ГЛОНАСС, которые в определенных ситуациях могут оказаться недоступными. Консультации с известными компаниями Эникс (Казань), Транзас (Санкт-Петербург), ZALA AERO (Ижевск), МКБ «Компас» (Москва) и ЦАГИ (Москва) показали необходимость создания полноценных систем навигации, альтернативных GPS/ГЛОНАСС.

---

Исследования проводятся в рамках работ по Государственному контракту № 07.514.11.4033 по теме «Разработка алгоритмического, математического и программного обеспечения системы моделирования автономного позиционирования БПЛА (аппаратно-программного комплекса) на основе технологий машинного зрения, искусственного интеллекта и параллельных вычислений» (шифр заявки “2011-1.4-514-002-031”).

В настоящей работе описываются разработанные схемы, позволяющие моделировать полет БПЛА без внешних источников навигационных данных. В качестве платформы для разработки была использована ПС НСКиД [1].

## 1. Состав системы моделирования

В состав разработанной системы моделирования входят следующие программные модули:

- (1) модуль создания набора опорных изображений объектов местности (базы объектов, БО) из серии аэрофотоснимков участков местности;
- (2) модуль создания видеороликов различных участков местности, аналогичных видео с БПЛА, при различных условиях и различного качества (базы видеороликов, БВ);
- (3) модуль вычисления дескриптора объекта на аэрофотоснимке;
- (4) модуль обучения оптимального классификатора объектов из базы объектов по базе видеороликов;
- (5) модуль локализации и прослеживания распознанного объекта в видеокдрах;
- (6) модуль отбраковки плохих видеороликов и улучшения качества видео;
- (7) модуль определения положения видеокамер относительно распознанного объекта;
- (8) модуль задания интегрированной базы фото/видео данных базы объектов и базы видеороликов;
- (9) модуль моделирования полета БПЛА с возможностью доступа через графический пользовательский интерфейс.

Все программные модули предназначены для эксплуатации на многопроцессорных персональных компьютерах и кластерных вычислительных устройствах, находящихся под управлением ОС Linux и Windows. Далее предлагается рассмотреть перечисленные выше программные модули более детально.

## 2. Программный модуль создания набора опорных изображений объектов местности

В состав программного модуля входит набор модулей, объединенных в логическую схему, показанную на рис. 1. Модуль *“navigator-M Create reference object images”* получает на вход аэрофотоснимок и

читает из текстового файла координаты и размеры опорных объектов на снимке с искусственного спутника Земли (ИСЗ). С помощью библиотеки GDAL [2] из исходного спутникового снимка вырезаются фрагменты, соответствующие опорным объектам. Результат работы модуля — множество изображений, передаваемых модулю сохранения изображений “*Save ImageCV*”.

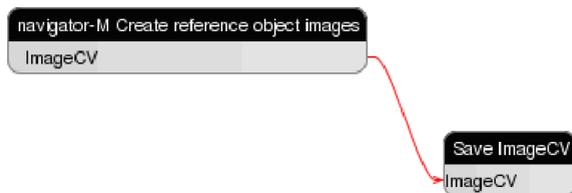


Рис. 1. Схема программного модуля создания набора опорных изображений объектов местности

### 3. Программный модуль создания видеороликов различных участков местности, аналогичных видео с БПЛА

В состав программного модуля входит набор модулей, объединенных в логическую схему создания видеофайлов. Схема изображена на рис. 2 и предназначена для генерации видеороликов с помощью камеры, установленной на виртуальном БПЛА.

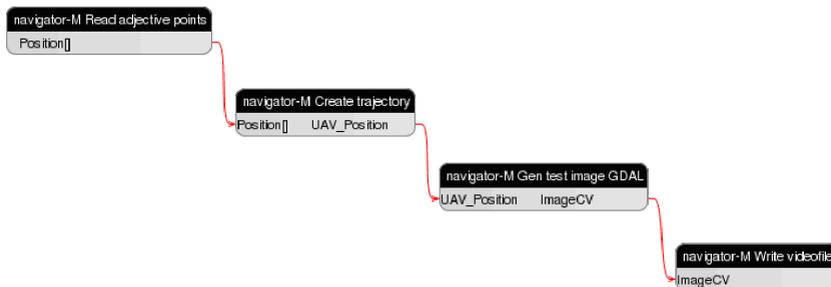


Рис. 2. Схема для создания видеороликов с камеры, установленной на БПЛА

Для задания входных данных был разработан модуль интерфейса. С его помощью пользователь может загрузить уменьшенное изображение карты местности и указать траекторию предполагаемого полета БПЛА. Далее эта информация сохраняется в файл, используемый модулем *“navigator-M Read adjective points”*. Внешний вид интерфейса в момент выбора траектории изображен на рис. 3. Все графические интерфейсы выполнены на базе интерфейса [3] системы ПС НСКид.

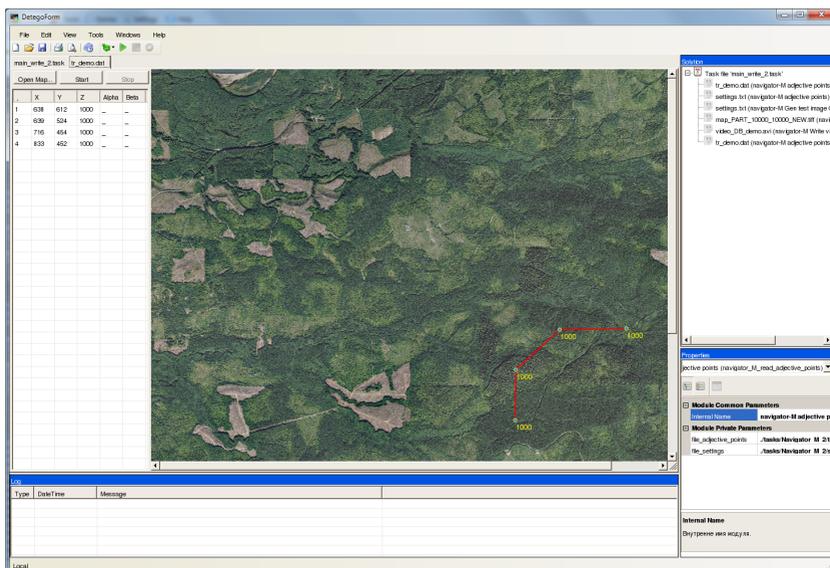


Рис. 3. Внешний вид интерфейса задания траектории в ПС НСКид

Модуль *“navigator-M Read adjective points”* читает из текстового файла координаты опорных точек в трехмерном пространстве. Координаты опорных точек задаются в системе координат, которые привязаны к камере на ИСЗ.

Модуль *“navigator-M Create trajectory”* генерирует траекторию полета БПЛА, используя информацию о координатах опорных точек, а также желаемую ориентацию аппарата в каждой опорной точке. Траектория полета представляется последовательностью точек

$(X_i, Y_i, Z_i)$  в трехмерном пространстве, лежащих на траектории. Также каждой точке сопоставляется три угла  $(\varphi_i, \psi_i, \theta_i)$  ориентации аппарата в этой точке. Другими словами, траектория — массив шестерок вида  $(X_i, Y_i, Z_i, \varphi_i, \psi_i, \theta_i)$ , каждая из которых соответствует некоторому положению и ориентации БПЛА.

Модуль “*navigator-M Gen test image GDAL*” получает на вход спутниковый снимок местности, а также одну из вышеупомянутых шестерок. Используя этот снимок, модуль создает изображение, соответствующее снимку с камеры на БПЛА, находящейся в данном положении и в данной ориентации. Каждое такое изображение является кадром видеоряда, полученного с помощью камеры. Модуль “*navigator-M Write videofile*” получает последовательность снимков с камеры на виртуальном БПЛА, объединяет их в видеофайл и сохраняет на жесткий диск.

#### 4. Программный модуль вычисления дескриптора объекта на аэрофотоснимке

Программный модуль вычисления дескриптора объекта на аэрофотоснимке — цепочка из следующих модулей: “*EMarking*”, “*Auto Rotate*” и “*Resize canvas*” (рис. 4).



Рис. 4. Схема вычисления дескриптора объекта на аэрофотоснимке

Модуль “*EMarking*” получает на вход изображение с объектом и делает его разметку с использованием канала прозрачности с помощью классификатора на основе расстояния Евклида-Махаланобиса [4], используя метод выделения спектрографических текстур [5].

Модуль “*Auto Rotate*” получает от модуля “*EMarking*” полноцветное изображение, альфа-составляющая пикселей объекта на котором установлена в значение “0”, а альфа-составляющая фона — в значение “255”. Модулем производится поворот изображения по линии положения выделенного на предыдущем модуле объекта [6].

Модуль “*Resize canvas*” получает на вход развернутое относительно линии положения находящегося на нем объекта изображение

от модуля *“Auto Rotate”*. Результат работы модуля — фиксированного размера двумерный дескриптор развернутого относительно своей линии положения объекта, представляющий собой полноцветное изображение, альфа-составляющая пикселей объекта на котором установлена в значение "0", а альфа-составляющая фона — в значение "255".

## 5. Программный модуль обучения оптимального классификатора

Программный модуль предназначен для обучения оптимального классификатора объектов из базы объектов по базе видеороликов. В состав программного модуля входит набор вспомогательных модулей, объединенных в логическую схему: *“Read ImageCV”* и *“Hemming”* (рис. 5).



Рис. 5. Схема обучения оптимального классификатора объектов из базы объектов

Модуль *“Read ImageCV”* загружает в память заранее сохраненные двумерные дескрипторы.

Модуль *“Hemming”*, основывающийся на искусственной нейронной сети Хемминга [7] (ИНС), получает на вход двумерные дескрипторы от модуля *“Read ImageCV”*, используемые для обучения оптимального классификатора, применяемого в дальнейшем для распознавания объектов по базе видеороликов. Для подтверждения характеристик классификатора используется схема поиска объектов из БО в базе видеороликов (см. рис. 6).

Модуль *“navigator-M Read videofile”* последовательно читает кадры из видеофайла, индексирует и отправляет их на вход модулям *“EMarking”* (для выделения всех объектов в каждом кадре), *“Get Data Entities”* (для связывания промаркированных, то есть разделенных объектов, с оригинальным кадром) и модулю *“Scan Windows Mark”* (для получения видеокадров с отмеченными найденными объектами).

Модуль *“EMarking”* получает на вход изображение и делает его разметку альфаканалом с помощью классификатора на основе расстояния Евклида-Махаланобиса.

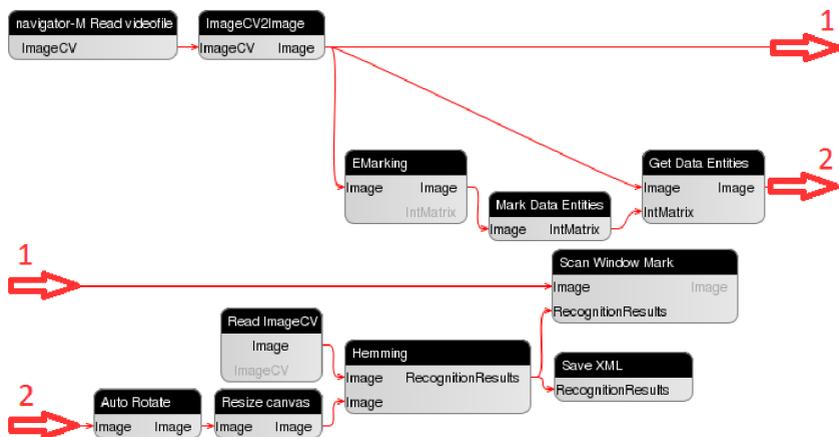


Рис. 6. Схема поиска объектов из БО в базе видеороликов

Модуль “*Mark Data Entities*” осуществляет маркировку, то есть отделение одних найденных модулем “*EMarking*” объектов от других. Для этого применяется метод связанных компонент [8].

Модуль “*Get Data Entities*” осуществляет связывание промаркированных модулем “*Mark Data Entities*” объектов с оригинальным необработанным кадром. Результат работы модуля — независимые полноцветные изображения отдельных объектов, альфа-составляющая пикселей объекта на каждом из которых установлена в значение “0”, а альфа-составляющая фона — в значение “255”.

Модуль “*Auto Rotate*” получает от модуля “*Get Data Entities*” соответствующие изображения и производит их поворот по линии положения выделенного объекта.

Модуль “*Resize Canvas*” получает на вход развернутое относительно линии положения находящегося на нем объекта изображение от модуля “*Auto Rotate*”. Результат работы модуля — фиксированного размера двумерный дескриптор развернутого относительно своей линии положения объекта, представляющий собой полноцветное изображение, альфа-составляющая пикселей объекта на котором установлена в значение “0”, а альфа-составляющая фона — в значение “255”.

Далее происходит распознавание полученных дескрипторов ранее уже описанным модулем “*Hemming*”, прошедшим предварительное обучение. Результаты с координатами найденных объектов передаются в модули “*Scan Windows Mark*” и “*Save XML*”.

Модуль “*Save XML*” сохраняет результаты распознавания объектов (в том числе и их идентификаторы класса) в XML-файл.

Модуль “*Scan Windows Mark*” предназначен для получения кадров с отмеченными найденными объектами. Модуль рисует на исходных кадрах рамки, отмечая все объекты, информация о которых получена от модуля “*Hemming*”. Рамки рисуются фиолетовым цветом.

## 6. Программный модуль локализации и прослеживания распознанного объекта в видеокадрах

Программный модуль предназначен для локализации и прослеживания распознанного объекта в видеокадрах. В состав программного модуля входит набор модулей, объединенных в логическую схему (рис. 7).



Рис. 7. Схема локализации и прослеживания распознанного объекта в видеокадрах

Модуль “*Localization and tracking*”, обеспечивающий локализацию и прослеживание объекта в видеокадрах, взаимодействует со схемой поиска объектов из базы объектов в базе видеороликов, а именно получает от модуля классификации “*Hemming*”, входящего в состав схемы поиска объектов из БО в базе видеороликов, необходимые данные о найденных в кадрах видеообъектах, в частности, такие как координаты левого верхнего угла, физические размеры и уверенность в распознавании. Перед получением данные упорядочиваются модулем “*RecognitionResults sorter*” (сортируются в порядке поступления кадров). Уверенность в распознавании одного и того же объекта мало меняется от кадра к кадру, что существенно упрощает решение задачи прослеживания.

Поступление информации о новых видеокадрах добавляет дополнительные строки в файл с результатами работы программного модуля. Например, при отслеживании в файле "video.mpg" одновременно трех объектов с идентификаторами  $a$ ,  $b$  и  $c$ , результат может быть следующим:

```
video.mpg [0] : a{150, 70}; b{30, 140}; c{1110, 280}  
video.mpg [1] : a{660, 90}; b{60, 135}; c{1000, 180}  
video.mpg [2] : a{770, 99}; b{90, 140}; c{1110, 380}.
```

Модуль "*RecognitionResults sorter*" запаковывает данные о найденных на видеоролике объектах и отправляет дальше модулю "*Localization and tracking*". Модуль "*Hemming*", основывающийся на нейронной сети Хемминга, получает на вход заранее сохраненные фиксированного размера двумерные дескрипторы развернутых относительно своей линии положения объектов, представляющие собой полноцветные изображения, альфа-составляющая пикселей объектов на которых установлена в значение "0", а альфа-составляющая фона — в значение "255".

## 7. Программный модуль отбраковки плохих видеороликов и улучшения качества видео

Программный модуль отбраковки плохих видеороликов и улучшения качества видео не является выделенным программным модулем, задача решается цепочкой из трех модулей: "*Rejection*", "*Median*" и "*navigator-M Write videofile*". Для внесения шума в видеоролики дополнительно предлагается модуль "*White noise*". Для тестирования программного модуля предлагается две схемы. В первой происходит запись зашумленного видеоролика, а во второй производится его оценка и улучшение качества. В обеих схемах задействованы модуль чтения ("*navigator-M Read videofile*") и модуль оценки и записи видеороликов ("*navigator-M Write videofile*"), а также вспомогательные модули конвертации данных из формата *ImageCV* в формат *Image* и обратно ("*ImageCV2Image*" и "*Image2ImageCV*" соответственно).

Модуль "*navigator-M Read videofile*" последовательно читает кадры из видеофайла, индексирует их и отправляет дальше.

Модуль "*navigator-M Write videofile*" получает ряд индексированных видеокадров в формате *ImageCV*, объединяет их в видеофайл и

сохраняет на жесткий диск. Если в режиме отбраковки зашумленных видеороликов общее количество кадров со статусом «зашумлен» в видеоряде становится недопустимым, то модуль завершает работу — видеоряд признается дефектным и отбраковывается, при этом формат имени сохраняемого файла изменяется с *[имя файла без расширения]/[расширение]* на *[имя файла без расширения].rejected/[расширение]*.

### 7.1. Схема зашумления и записи видеоролика

Схема, представленная на рис. 8, предполагает использование модуля “White noise”, вносящего «белый» (гауссовский) шум в поступающие видеокadres. Модуль “navigator-M Write videofile” не производит проверку состояния сохраняемого видеоролика (параметр модуля *check* установлен в значение "0").

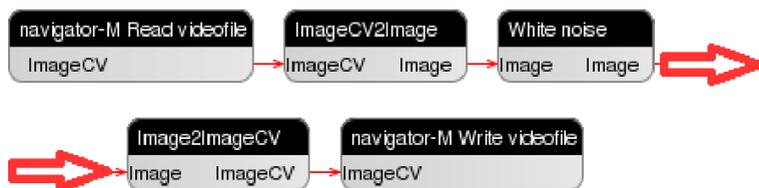


Рис. 8. Схема зашумления и записи видеоролика

### 7.2. Схема оценки и улучшения качества видеоролика

В схеме задействованы в качестве основных модули “Rejection”, “Median” и “navigator-M Write videofile”. Модуль “navigator-M Write videofile” в данной схеме проверяет состояние сохраняемого видеоролика (параметр модуля *check* установлен в значение "1").

Модуль “Rejection” получает на вход очередной кадр из видеоряда и производит его проверку на зашумленность. Если отношение сигнала к шуму не превосходит некоторого заданного порога, указанного в настройках модуля, то видеокadру присваивается статус «зашумлен». Далее видеокadр может быть передан модулю медианной фильтрации (“Median”), удаляющему шум только с видеокadров с соответствующим статусом (в соответствии с рис. 9), либо напрямую на модуль оценки и записи видеороликов “navigator-M Write videofile”.

Модуль медианной фильтрации “*Median*” обеспечивает удаление «белого» шума с видеок кадров со статусом «зашумлен» (после обработки статус снимается), кадры без установленного статуса пропускаются без обработки.



Рис. 9. Схема оценки и улучшения качества видеоролика

## 8. Программный модуль определения положения видеок камер относительно распознанного объекта

Программный модуль “*navigator-M Calc 3D coords of objects*” разработан для вычисления трехмерных координат камеры на БПЛА относительно некоторого опорного объекта, который задан на спутниковом снимке местности и для которого определены координаты его проекции на снимке с БПЛА. Это эквивалентно вычислению положения объектов в системе координат, связанных с положением и ориентацией камеры на БПЛА, которые в свою очередь связаны с данным кадром. Необходимо вычислить двумерные координаты проекций опорных объектов на спутниковом снимке, а также положение и ориентацию БПЛА (для этого применяются другие модули, входящие в состав комплекса).

## 9. Программный модуль задания интегрированной базы фото/видео данных базы объектов и базы видеороликов

Программный модуль представляет собой набор графических интерфейсов, служащих для добавления видеороликов и изображений объектов. Внешний вид модулей представлен на рис. 10. Пользователю представляется возможность создания базы объектов с указанием класса принадлежности каждого изображения объекта и возможность создания базы видеороликов.

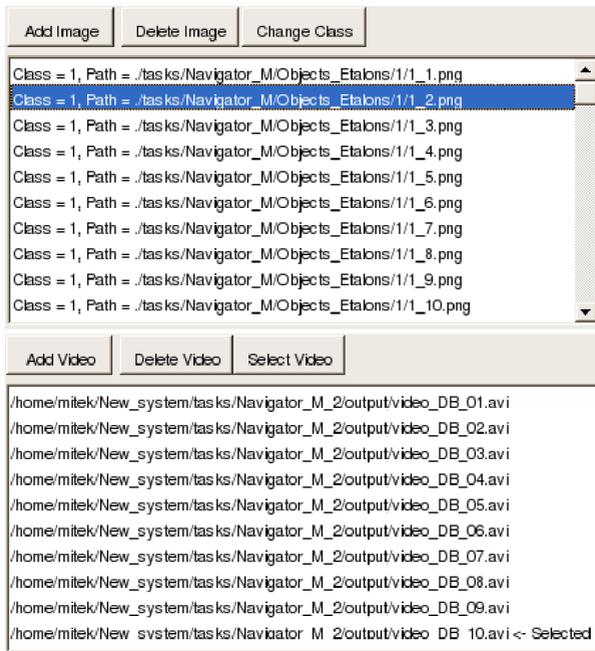


Рис. 10. Внешний вид модулей задания базы объектов и базы видеороликов

## 10. Программный модуль моделирования полета БПЛА

Данный программный модуль предназначен для моделирования полета БПЛА средствами компьютерной графики (генерация кадров видеоряда с помощью камеры, установленной на аппарате). Решается задача автоматического определения положения и ориентации аппарата во время полета, с использованием методов и алгоритмов технического зрения [9, 10]. Программный модуль предназначен для работы с теми участками поверхности Земли, перепады высот на которых незначительны (равнинные поверхности).

Пользователь системы моделирования имеет возможность задать спутниковый снимок местности, в пределах которой виртуальный аппарат совершает свои полеты, а также задать технические характеристики как самого аппарата, так и установленной на нем камеры. Также он имеет возможность выбрать, какие программные средства

(библиотеки и алгоритмы) будут использоваться в ходе решения задачи. Пользователь может запустить процесс моделирования полета БПЛА и вычисления положения и ориентации аппарата, связанные с каждым кадром сгенерированного видеоряда. После того, как программный модуль завершил свою работу, пользователь может оценить погрешности в определении положения и ориентации аппарата. Эти выходные данные представляют собой набор чисел. Также есть возможность оценить погрешности путем визуального анализа исходного спутникового снимка с нанесенными действительной и вычисленной траекториями полета БПЛА.

### 10.1. Схема нахождения особых точек

Схема, представленная на рис. 11, предназначена для нахождения координат особых точек на спутниковой карте местности, над которой виртуальный БПЛА совершает полеты, а также для вычисления дескрипторов каждой особой точки. Все это является исходными данными для основной схемы, которая будет описана далее. Поскольку спутниковый снимок остается неизменным, то данная схема запускается однократно.

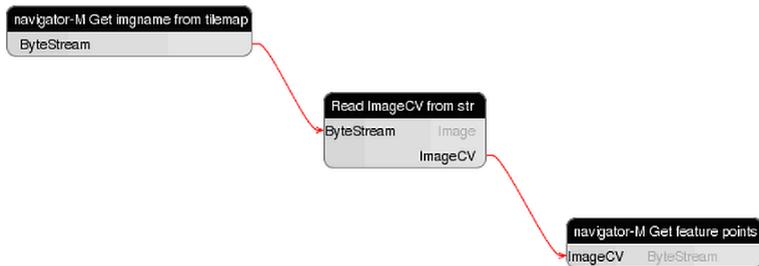


Рис. 11. Схема для подготовки исходных данных

Поскольку исходный спутниковый снимок представлен пирамидой изображений, модуль “*navigator-M Get imgname from tilemap*” генерирует множество строк-путей до каждого изображения и последовательно (по одной) отправляет все строки следующему модулю. Модуль “*Read ImageCV from str*” получает на вход путь до изображения, и загружает его (изображение) в память. Далее, модуль

“*navigator-M Get feature points*” находит на изображении координаты особых точек, вычисляет дескрипторы для каждой особой точки и сохраняет эту информацию на жесткий диск в виде набора файлов.

## 10.2. Схема моделирования полета БПЛА с использованием карты местности

Схема для моделирования полета БПЛА представлена в двух вариантах. Первый вариант изображен на рис. 12, состоит из десяти модулей и предназначен для моделирования полета БПЛА, а также определения его положения и ориентации. Для большей удобочитаемости схема представлена в виде двух частей.

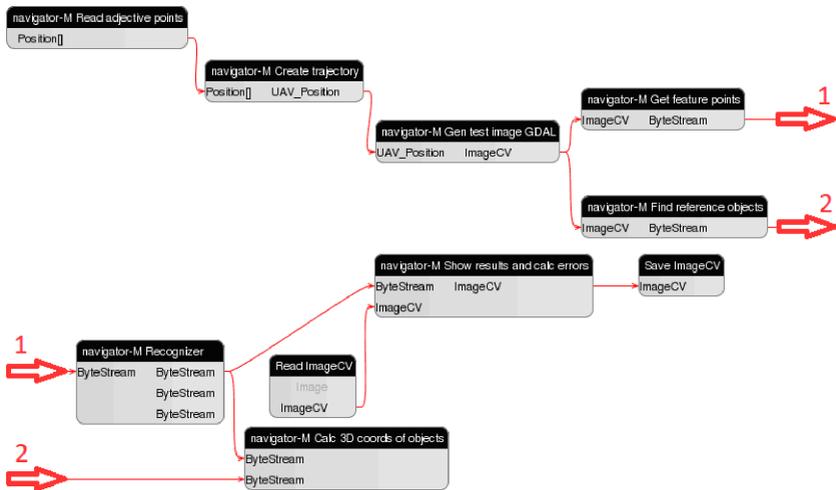


Рис. 12. Основная схема (вариант без использования сгенерированных видеороликов)

Модули “*navigator-M Read adjective points*”, “*navigator-M Create trajectory*” и “*navigator-M Gen test image GDAL*” были описаны ранее в разделе 2.

Модуль “*navigator-M Get feature points*” принимает на вход сгенерированное изображение с камеры виртуального БПЛА и находит на снимке координаты особых точек, а также вычисляет дескрипторы для каждой особой точки. По своему функциональному назначению модуль идентичен одноименному модулю из схемы нахождения

особых точек. Отличие состоит в том, что в данной схеме модуль вызывается с другими параметрами, и данные, которые связаны с особыми точками на изображении и их дескрипторами, не сохраняются на жесткий диск в виде файлов, а отправляются следующему модулю схемы (в двоичном виде).

Модуль *“navigator-M Recognizer”* получает на вход информацию об особых точках, найденных на сгенерированном снимке с БПЛА, а также дескрипторы особых точек. Модулем осуществляется вычисление положения и ориентации БПЛА. На выходе получается пара шестерок, соответствующих действительному и вычисленному положению и ориентации БПЛА, которая подается далее модулю *“navigator-M Calc 3D coords of objects”*.

Модуль *“navigator-M Find reference objects”* получает изображение от модуля *“navigator-M Gen test image GDAL”*, вычисляет координаты опорных объектов на нем и отправляет список их координат следующему модулю.

Модуль *“navigator-M Calc 3D coords of objects”* вычисляет положение видеокамеры относительно каждого из распознанных опорных объектов, координаты которых были найдены на некотором кадре из видеоряда. Это эквивалентно вычислению положения объектов в системе координат, связанных с положением и ориентацией камеры на БПЛА, которые в свою очередь связаны с данным кадром.

Модуль *“Read ImageCV”* загружает в память уменьшенную копию спутникового снимка местности. Копия предназначена для визуализации результатов.

Модуль *“navigator-M Show results and calc errors”* аккумулирует внутри себя пары шестерок в упомянутом выше формате до тех пор, пока не будут обработаны все кадры сгенерированного видеоряда. Также модуль получает на вход исходный спутниковый снимок, на который наносится действительная траектория полета БПЛА (в виде красной линии) и вычисленная траектория полета (в виде синей линии). Изображение с нанесенными на него траекториями подается на вход следующему модулю. В текстовый файл записывается информация о минимальных, средних и максимальных погрешностях в определении каждого из трех углов ориентации ( $\varphi, \psi, \theta$ ), каждой из трех компонент ( $X, Y, Z$ ), задающих положение БПЛА, а также информация о минимальном, среднем и максимальном расстоянии между действительными и вычисленными положениями «беспилотника» — всего  $7 \times 3 = 21$  числовое значение.

Модуль “*Save ImageCV*” предназначен для сохранения спутникового снимка с нанесенными на него действительной и вычисленной траекториями полета БПЛА на жесткий диск.

### 10.3. Схема моделирования полета БПЛА с использованием видеоролика, созданного с помощью камеры на БПЛА

Второй вариант схемы для моделирования полета БПЛА изображен на рис. 13, состоит из девяти модулей и предназначен для определения положения и ориентации БПЛА, которые связаны с каждым кадром уже сгенерированного видеофайла. Для большей удобочитаемости схема представлена в виде двух частей.

Модуль “*navigator-M Read videofile*” последовательно читает кадры из видеофайла и отправляет их на вход следующему модулю.

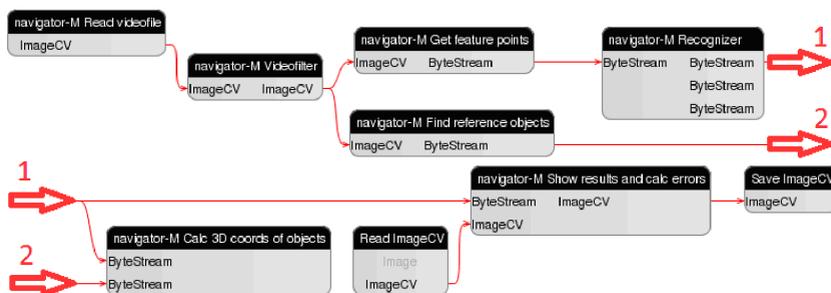


Рис. 13. Основная схема (вариант с использованием сгенерированного видеоролика)

Назначение остальных восьми модулей схемы, а также описание их параметров, совпадает с тем, что было приведено для первого варианта схемы для моделирования полета БПЛА.

Результат работы схем моделирования полета БПЛА может быть показан в специальном модуле-интерфейсе, отображающем уменьшенный вариант изображения карты местности. Там же визуализируется список результатов расчетов координат и положения для каждой точки. Внешний вид такого модуля в режиме работы отображен на рис. 14.

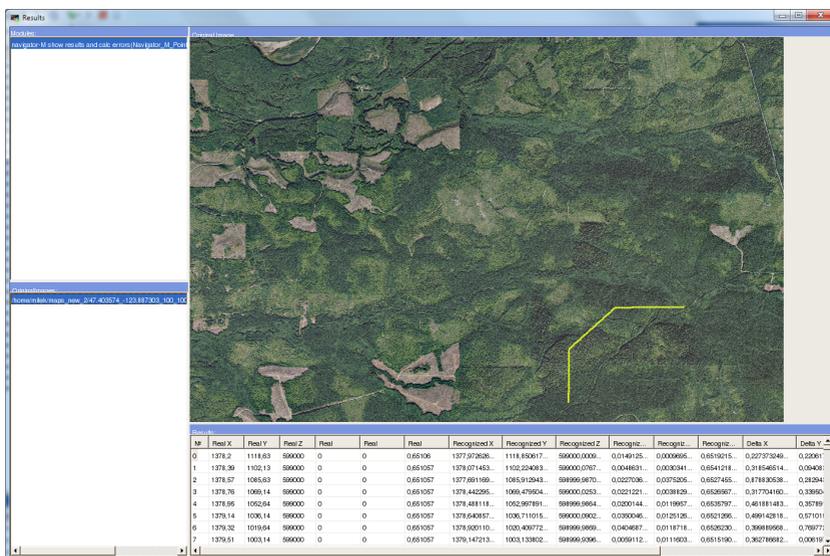


Рис. 14. Внешний вид модуля визуализации результатов моделирования полета БПЛА в ПС НСКиД

## Заключение

Полученные результаты позволят обеспечить новое качество системы навигации БПЛА, повышение надежности и живучести систем навигации и управления, расширение функциональных возможностей применения БПЛА. Это выведет отечественные разработки в данной области на качественно новый уровень и создаст возможность перехода от импорта БПЛА к их экспорту.

Результаты работы могут применяться при разработке систем навигации летательных аппаратов и мобильных роботов, в том числе бортовых навигационных систем БПЛА, с целью получения новых технических решений по аппаратному и программному обеспечению систем управления. Разработанные методы моделирования системы ориентирования БПЛА способны обеспечить эффективное по точности, скорости и надежности решение задачи автоматического ориентирования БПЛА.

## Список литературы

- [1] Нейросетевая система контроля телеметрической информации, диагностики подсистем космических аппаратов, обработки космических снимков, Свидетельство о государственной регистрации программы для ЭВМ № 2012613261. ↑[1](#)
- [2] RU: GDAL - Geospatial Data Abstraction Library, URL: [http://www.gdal.org/index\\_ru.html](http://www.gdal.org/index_ru.html). ↑[2](#)
- [3] Тищенко И. П. *Графический интерфейс ПС ИНС* // VIII международная конференция «Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта» (CAD/CAM/PDM-2008. — Институт проблем управления РАН, Москва, 2008, с. 16–19 ↑[3](#)
- [4] Амелькин С. А., Хачумов В. М. *Обобщенное расстояние Евклида-Махаланобиса и его применение в задачах распознавания образов* // 12-я Всероссийская конференция «Математические методы распознавания образов». — М. : МАКС Пресс, 2005, с. 7–9 ↑[4](#)
- [5] Фраленко В. П. *Анализ спектрографических текстур данных дистанционного зондирования Земли* // Искусственный интеллект и принятие решений, 2010, № 2, с. 11–15 ↑[4](#)
- [6] Виноградов А. П., Калугин Ф. В., Недев М. Д., Погодин С. В., Талалаев А. А., Тищенко И. П., Фраленко В. П., Хачумов В. М. *Выделение и распознавание локальных объектов на аэрокосмических снимках* // Авиакосмическое приборостроение, 2007, № 9, с. 39–45 ↑[4](#)
- [7] Уоссермен Ф. *Нейрокомпьютерная техника: Теория и практика*. М. : Мир, 1992 ISBN 5-03-002115-9. — 240 с. ↑[5](#)
- [8] Шапиро Л., Стокман Д. *Компьютерное зрение*. М. : Бинум. Лаборатория знаний, 2006 ISBN 5-94-774384-1. — 752 с. ↑[5](#)
- [9] Степанов Д. Н., Тищенко И. П., Поляков А. В., Вагутин В. М., Соболев Д. Б. *Подсистема определения положения и ориентации беспилотного летательного аппарата* // V Всероссийская научно-техническая конференция «Актуальные проблемы ракетно-космического приборостроения и информационных технологий». — ОАО «Российские космические системы», Москва, 2012 ISBN 978-5-88070-025-7, с. 21–22 ↑[10](#)
- [10] Степанов Д. Н., Тищенко И. П. *Задача моделирования полета беспилотного летательного аппарата на основе системы технического зрения* // Электронный научный журнал «Программные системы: теория и приложения», 2011. Т. 2, № 4(8), с. 33–43, URL: [http://psta.psirras.ru/read/psta2011\\_4\\_33-43.pdf](http://psta.psirras.ru/read/psta2011_4_33-43.pdf) ↑[10](#)

Рекомендовал к публикации

д.т.н. В. М. Хачумов

Об авторах:

### Игорь Петрович Тищенко



Один из главных разработчиков системы параллельно-конвейерных вычислений в области обработки и распознавания изображений, сигналов, космических снимков и др. Автор ряда алгоритмов по перечисленным тематикам. Разработчик системы автономного позиционирования БПЛА. К.т.н.

e-mail: [billy@billy.botik.ru](mailto:billy@billy.botik.ru)

### Дмитрий Николаевич Степанов



Разработчик алгоритмов восстановления трехмерных изображений, идентификации лиц, алгоритмов и методов автономного определения положения и ориентации беспилотных летательных аппаратов. Аспирант.

e-mail: [mittek1989@mail.ru](mailto:mittek1989@mail.ru)

### Виталий Петрович Фраленко



Автор нейросетевого метода сжатия изображений с потерями, превосходящего возможности формата Jpeg-2000, метода выделения регионов на снимках дистанционного зондирования Земли с помощью спектрографических текстур и нейронных сетей, специалист по нейросетевой компьютерной безопасности.

e-mail: [alarmod@pereslavl.ru](mailto:alarmod@pereslavl.ru)

Образец ссылки на эту публикацию:

И. П. Тищенко, Д. Н. Степанов, В. П. Фраленко. *Разработка системы моделирования автономного полета беспилотного летательного аппарата* // Программные системы: теория и приложения : электрон. научн. журн. 2012. Т. 3, № 3(12), с. 3–21.

URL: [http://psta.psisras.ru/read/psta2012\\_3\\_3-21.pdf](http://psta.psisras.ru/read/psta2012_3_3-21.pdf)

I. P. Tishchenko, D. N. Stepanov, V. P. Fralenko. *The development of modeling system of unmanned aerial vehicle autonomous flight.*

ABSTRACT. This paper describes the flight simulation system for unmanned aerial vehicle (UAV) in standalone mode. Its work has been focused on software generated images that are the analogue of real images from the UAV on-board cameras. Object recognition are produced. (In Russian).

*Key Words and Phrases:* system modeling, UAV, pattern recognition.