

СИСТЕМА ПОЗИЦИОНИРОВАНИЯ БПЛА НА ОСНОВЕ РАСПОЗНОВАНИЯ ГРАФИЧЕСКИХ ОБРАЗОВ

Аношин И.С.¹, Жолбин А.П.², Проскоков А.В.³

¹ *ЮТИ ТПУ, зр. 17В11, e-mail: legenda.drovasek@gmail.com*

² *ЮТИ ТПУ, зр. 17В11, e-mail: pszholbin@gmail.com*

³ *ЮТИ ТПУ, к.т.н., доцент, e-mail: proskokov@tpu.ru*

Аннотация

Развитие технологий, основанных на алгоритмах ИИ позволяет определять местоположение БПЛА на основе графических образов, полученных в реальном времени. Система, оснащенная таким алгоритмом в перспективе, может значительно улучшить достоверность получаемых координат в условиях потери радиосигнала оператора или сигнала спутниковой навигации. Данная функция ориентирована на совершенствование алгоритмов не только летательных, но и наземных роботизированных мобильных платформ.

Ключевые слова: искусственный интеллект, БПЛА, картография, позиционирование, распознавание графических образов.

Введение

На рынке систем позиционирования беспилотных летательных аппаратов (БПЛА) выделяют: радиосистемы, оптические, геолокационные, а также различные их комбинации. Интерес в системах позиционирования резко возрос, и продолжает расти. Но распространённые на данный момент системы, сильно зависят от связи с GPS или ГЛОНАСС, или же, управления в дистанционном режиме. Одним из важных недостатков распространенных систем ориентирования является их слабая автономия и помехозащищённость как в условиях радиопомех, так и при больших удалениях от оператора. В связи с этим можно предположить, что бюджетное решение может востребовано для транспортных логистических компаний, занимающихся доставкой небольших грузов. Так же в системах позиционирования могут быть заинтересованы МЧС для быстрого нахождения пострадавших в стихийных бедствиях или потерянных в лесу людей. Возможные заинтересованные организации: МЧС, Банки (Сбербанк, ВТБ, Тинькофф и тд...), Авиакомпании (Аэрофлот, Победа, Citrus), МОРФ, Различные бизнесы, Почтовые организации компании, МЧС. В общем, все организации, которые используют дроны могут так или иначе модернизировать свои БПЛА для сокращения штата операторов БПЛА и повышения надежности работы оборудования.

На данный момент на Российском рынке нет открытых систем, использующих нейросеть для точного позиционирования БПЛА в воздушном пространстве, но в последнее время больше число компаний занимаются разработкой и изготовлением БПЛА. Системы управления традиционно предусматривают управление человеком через телеметрию. Основным методом позиционирования является глобальная система навигации. Приемник устанавливается на борт БПЛА и получает данные со спутников. Для увеличения точности показаний данных с глобальной системы сейчас активно стали использоваться сети наземных стационарных вышек. Такие вышки являются реперными для системы навигации БПЛА, они определяют погрешности показаний глобальной системы навигации и отправляют поправки по радиоканалу на приемники БПЛА. Одной из распространенных, на данный момент, систем такого типа является DGPS (differential global positioning system). Существуют работы, в которых предлагается использовать стационарные вышки не только как корректоры, но и как основной источник для получения данных о местоположении. В работе [2] описана система и алгоритм уточнения местонахождения легкого БПЛА на основе калмановской фильтрации измерений пеленгационного типа. Предложена система позиционирования БПЛА по данным расстояний до единиц сети вышек. Также для получения информации о положении БПЛА в пространстве на борт БПЛА, помимо приемников GPS, устанавливают комплекс инерциальных систем. В нее входит комплекс датчиков, по данным которых автопилот получает информацию о воздушной скорости, кренах, ускорениях барометрической высоте и др. В комплекс инерциальной системы могут входить:

- Барометр, с помощью которого возможно определять высоту относительно от задаваемого нулевого уровня. Расчет высоты основан на использовании закономерного измерения атмосферного давления в зависимости от высоты движения, относительно некоторого уровня. Минус такого датчика – относительно большая погрешность в измерениях (около 10-15 м).

- Радиотехнический датчик основан на измерении отрезка времени между посылкой и приемом электромагнитных волн, отраженных от поверхности, до которой измеряется высота. Подходящий для легких мобильных роботов датчик имеет большое энергопотребление и при этом работает на расстояниях до 20–30 м, что недостаточно для летающих роботов.

- Сонар — средство звукового обнаружения подводных объектов с помощью акустического излучения. Подходящий для легких мобильных роботов сонар работает на расстояниях до 10 м, что недостаточно для летающих роботов.

- ЛИДАР (LIDAR — световое обнаружение и определение дальности) — датчики построенные по технологии получения и обработки информации об удаленных объектах с помощью активных оптических систем, использующих явления отражения света и его рассеяния в прозрачных и полупрозрачных средах. Датчики имеют большое энергопотребление и большой вес, что исключает использование таких датчиков в сверхлегких мобильных роботах.

- Комплекс из трехосного гироскопа и акселерометра, который позволяет определить углы наклона БПЛА относительно горизонта и ускорения вращения.

- Термодатчики в шести направлениях, по данным которых микрокомпьютер БПЛА оценивает разность температур в разных направлениях и делает выводы о углах наклона БПЛА относительно горизонта.

Помимо развития средств инерциальных систем для позиционирования БПЛА, в последнее время развивается направления визуального позиционирования БПЛА в пространстве. В таких методах используется бортовые сенсоры, фото- видео регистраторы или комплекс таких регистраторов и сенсоров, а также программные средства для обработки полученных данных, что и будет по большей части использоваться в нашей работе. Позиционирование БПЛА по данным встроенных сенсоров впервые было применено для автоматического ориентирования крылатых ракет в XX веке. В процессе полета на микрокомпьютер ракеты поступает информация с бортового высотомера в виде последовательности разностей расстояний до поверхности в данный момент времени и в предыдущие. Эта последовательность сопоставляется с записанной на земле картой рельефа, при этом сравниваются именно последовательности относительных высот, а не абсолютные значения высот. Как только микрокомпьютер обнаруживает совпадения, система управления получает координаты маршрута, производится расчет величины накопившейся ошибки и вносит поправки в курс движения. Такая система требует высокой точности измерения высоты, имеет большой вес (более 20 кг) и большое энергопотребление, что не позволяет использовать такой метод в сверхлегких БПЛА. В работе [2] описан алгоритм определения относительных (от станции запуска) трехмерных координат и углов ориентации БПЛА без использования сигналов спутниковой навигации. Данный подход заключается в использовании системы компьютерного зрения для генерации и обработки потока фотографий подстилающего рельефа. Из-за требования большого ресурса вычислительной мощности такая система предполагает обработку полученных данных на базовой станции и, следовательно, постоянную связь с БПЛА, однако ввиду возможных различных радиопомех или перехватов сигнала, предлагается:

- а) построение карт, нахождение объектов и выполнение заранее записанных действий удаленно на БПЛА без постоянной радиосвязи со станцией.

- б) установка маршрута картографирования и поиска объектов по определённому курсу или площади до запуска БПЛА, фотографирование местности без передачи данных на станцию, возврат на станцию, которая является основным сервером взаимодействия с остальным роём БПЛА для загрузки свежих записей местности. В этот момент все сложные вычисления по сшиванию записей в одну единую карту, поиск путей и объектов на этой карте происходит на станции-сервере. Данные карт и доступных маршрутов будут обновляться на БПЛА только в момент прямого контакта со станцией, что исключит вероятность передачи неправильных данных ввиду помех и/или перехвата важных данных третьей стороной.

Для увеличения точности следования заданному маршруту в условиях редкого обновления местоположения (например, полета в гористой местности) возникает необходимость внедрения в программный код микрокомпьютера БПЛА алгоритмов оптимизации полета за счет оценивания действующих на БПЛА неизвестных внешних процессов.

Задача предсказания значений случайного процесса, порождаемого белым шумом, пропущенным через линейный фильтр, является наиболее типичной для калмановской фильтрации. Для оценки случайных процессов в системах навигации мобильных объектов достаточно часто и эффективно применяется фильтр Калмана, который удовлетворительно работает на больших

мобильных объектах с разнообразным навигационным оборудованием. Для сверхлегких БПЛА задача снижения уровня отклонения от заданной траектории движения остается открытой, так как влияние на них сопутствующих неопределенностей более значительно. При оптимизации движения одиночного БПЛА и группы в режиме реального времени возникают потребности в решении за ограниченное время многомерных оптимизационных задач при зашумленных наблюдениях. В таких условиях хорошо зарекомендовали себя рандомизированные рекуррентные алгоритмы стохастической оптимизации. Детальному анализу возможностей рандомизированных алгоритмов в задачах оценивания и оптимизации при произвольных помехах посвящена монография О.Н. Граничина и др [1].

Основная часть

Основываясь на вышеупомянутых работах с учетом уровня технологий искусственного интеллекта необходимо подготовить программно-аппаратный модуль, способный управлять траекторией движения мобильного робота, основываясь на результаты сопоставления текущего местоположения с общей картой. Поставленная задача в наше время осуществима с применением уже готовых программных библиотек, основанных на алгоритмах нейронных сетей. Известны следующие библиотеки, свободно распространяемые в глобальной сети Интернет: Tourchvision, CUDA, TensorFlow, Opencv. Наиболее простая в применении данных алгоритмов на языке Python является библиотека OpenCv. Библиотека с открытым исходным кодом, разработанная и выпущенная Intel в 2000 году для реализации компьютерного зрения, решающая такие задачи, как обнаружение лиц, обнаружение объектов, распознавание лиц, сегментация изображений и многое другое. Вот некоторые из основных особенностей OpenCV: алгоритмическая эффективность, широкий доступ к алгоритмам, несколько интерфейсов для реализации.

Решение задачи позиционирования должно происходить в два этапа. Первый этап – это создание карты местности на основе произведенной серии фотографий и последующей графической обработки. Второй этап – это идентификация текущего местоположения другим БПЛА с учетом разной высоты полета и возможной модификации местности.

Описание алгоритма

Программный модуль был разработан на языке программирования Python с использованием библиотек и программных модулей OpenCV, numpy, tqdm, os, time.

По сравнению с такими языками, как C/C++, Python несколько медленнее. Тем не менее, Python может быть легко расширен с помощью C / C++, что позволяет нам писать интенсивный с точки зрения вычислений код на C / C++ и создавать оболочки Python, которые можно использовать в качестве модулей Python. Это дает нам два преимущества: во-первых, код выполняется так же быстро, как исходный код на C/C++ (поскольку это фактический код на C++, работающий в фоновом режиме), и, во-вторых, его проще кодировать на Python, чем на C/C++. OpenCV-Python — это оболочка Python для оригинальной реализации OpenCV на C++. Данный язык легко интегрируется с одноплатными мобильными микрокомпьютерами, такими как Jetson Nano, Asus tinkerbord, Raspberry Pi.

OpenCV-Python использует Numpy, которая является высокооптимизированной библиотекой для числовых операций с синтаксисом в стиле MATLAB. Все структуры массивов OpenCV преобразуются в массивы библиотеки Numpy, что также упрощает интеграцию с другими библиотеками, использующими Numpy, такими как SciPy и Matplotlib.

Tqdm библиотека — это модуль Python, предназначенный для отображения интеллектуальных индикаторов выполнения, которые показывают ход выполнения вашего кода на Python. Эту библиотеку также можно использовать для просмотра прогресса модели машинного обучения при обучении модели на очень большом наборе данных.

Модуль OS — это модуль Python, который предоставляет интерфейс для взаимодействия с базовой операционной системой, на которой запущен Python. Этот модуль предоставляет переносимый способ использования функциональности, зависящей от операционной системы. Модуль OS поставляется по умолчанию в стандартных служебных модулях Python.

Библиотека Time предоставляет множество способов представления времени в коде Python, таких как объекты, числа и строки. Он также предоставляет функции, отличные от представления времени, такие как ожидание во время выполнения кода и измерение эффективности вашего кода.

Функция photocar проверяет наличие папок с фотографиями для обработки, их создания и вывода готового результата. Функция videocar занимается тем же что и photocar, только для обработки видео и перевод его в формат серии снимков.

Функция findphoto является нашим компьютерным зрением для поиска объектов на карте. Однако ввиду недоработки программного кода пока что мы его реализовали как поиск фото внутри фото, что все ещё отражает полный функционал нашей программы.

Алгоритм работы программного модуля представлен на рис. 1. В данном модуле осуществляется следующие вычислительные действия. Сначала готовятся папки для работы с файлами, затем из видеофрагмента с выбранным интервалом собираются произвольные кадры. В случае успешной подготовки необходимых кадров производится последовательное склеивание выбранных фрагментов. Таким образом, может сформирована единая карта местности.

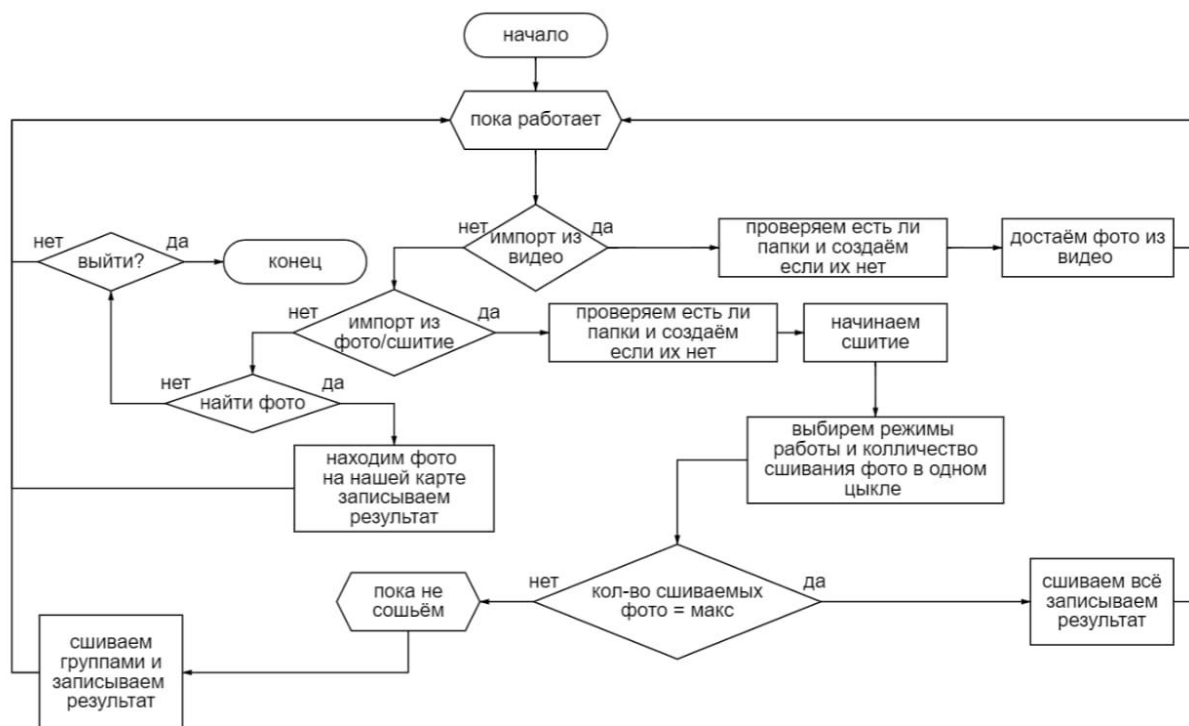


Рис. 1. Алгоритм программного модуля

После запуска программы в консольном режиме выбираем нужный режим обработки данных. Либо работа с видео, или сразу с готовыми фотографиями. Здесь программа считывает параметры съёмки: разрешение, общее количество кадров, и количество кадров в секунду.

С использованием вышеуказанных библиотек формируется набор пронумерованных фотографий. Далее программа преобразует видео в серию фотографий с интервалом в 1 секунду. Имеется возможность изменения интервала в зависимости от скорости полета самолета картографа.

После предварительной подготовки файлов запускается цикл сшивания фотографий. Путем подбора количества форографий в одной группе, промежутка для сшивания, добиваемся наилучшего результата. После нескольких циклов сшивания, где сначала все кадры видео были собраны в группу по 5 фото, повторяется цикл укрупнения фотографий и потом сшитые пятёрки сшиваются вместе в одну большую фотографию.

В результате полученный фрагмент карты имеет высокую степень детализации и вполне пригоден для дальнейшей проработки программного модуля. Для проверки алгоритма поиска расположения участка на основной карте для программы необходимо указать путь, где находится файл с фрагментом, принадлежащим карте и формат документа, также указать погрешность поиска.

Также был разработан удобный интерфейс программы, который в будущем будет реализован на станции-сервере. В интерфейсе программы будет присутствовать готовая сшитая карта с возможностью её масштабирования и передвижения по ней, слева фрагменты с найденными объектами в виде списка с ползунком, справа функциональные кнопки взаимодействия с картой. Кнопка «Выбрать фрагмент» отвечает за поиск фрагмента из левой части окна на общей карте, кнопка «Выстроить маршрут» отвечает за построение маршрута от точки А до точки Б как по прямой, так и во

кривой траектории с возможностью огибания обозначенных в ручную бесполётных зон, в качестве точек А и Б могут быть выбраны найденные объекты, возможно добавление промежуточных пунктов следования (весь функционал по прокладыванию маршрута пока еще не реализован).

Снизу находится консоль, в которой показаны статус и прогресс-бары выполнения задач, такие как сшивание, обработка изображений и следование БПЛА по заданному маршруту.

Сверху находятся функциональные кнопки для загрузки видео/изображений карты для сшивания, сохранение текущей карты с отметками объектов на ней, а также кнопка для ручного пересшивания, если автоматический алгоритм выдал ошибку или что-то сшилось неправильно.

Для получения видеофрагментов использовался беспилотный летательный аппарат собственной сборки, летающий по FPV технологии. Размах крыльев планера 1100 мм, максимальное время полета до 40 мин. Под крылом самолета была установлена видеокамера Thumb Pro, направленная вниз, что позволяет производить видеосъемку местности в формате 4k с минимальным искажением. Кроме того, данная видеокамера оснащена гироскопом и во время съемки осуществляется запись видео совместно отклонения от горизонта. Полученный видеофайл обрабатывался в программном модуле Gyroflow для стабилизации эффекта тряски камеры.

На рис. 2 показана экранная форма программного модуля, реализующего процесс подготовки карты местности на основе видеофрагмента. На нем показан фрагмент готового изображения после процесса обработки графических изображений

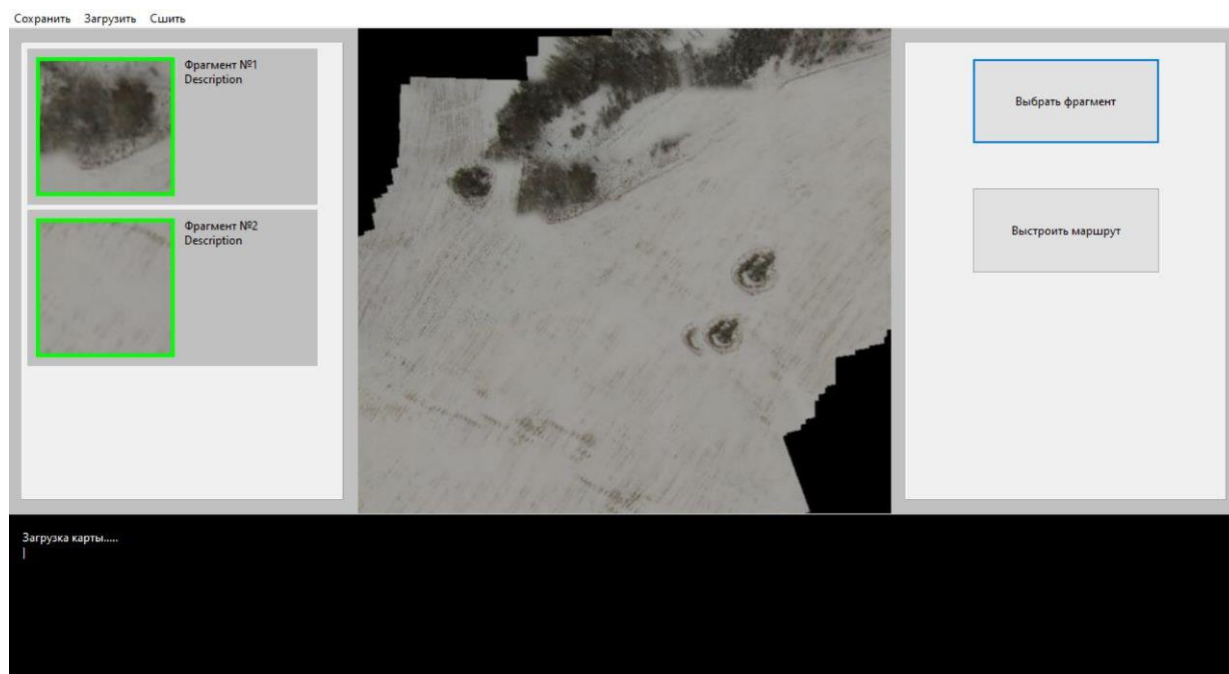


Рис. 2 Интерфейс программного модуля для получения фрагмента карты местности

Заключение

В целом подготовленный программный модуль выполняет поставленные задачи по подготовке карты местности и поиску текущего местоположения. Однако, необходимо повышать достоверность полученных результатов, набирая статистику подготовленных карт. Алгоритм еще выдает ошибки при сшивании нескольких изображений в одно общее, ввиду того что не может найти общие точки-ориентиры. Для улучшения качества работы было предложено разбивать на мелкие группы при возникновении ошибок (например, если не смогла сшить 5 фотографий в одно, будет сшивать вместе попарно), а если и это не поможет, то программа автоматически меняет настройки в маске для того, чтобы это было возможно реализовать. В случае если всё равно не удастся сшить программа выводит информацию о проблемном изображении. Ошибка о недостатке вычислительных ресурсов решена путём оптимизации некоторых загруженных функций, больше не наблюдалась.

Проблема с сохранением готовых изображений ввиду отсутствия доступа к рабочей папке решена использованием дополнительного алгоритма для взаимодействия с рабочей папкой.

Проблема с появлением чёрных пятен на готовой карте связана с тем, что пустые пространства алгоритмом заполняются чёрным цветом (код RGB (256,256,256), всё еще на стадии решения, основная задумка – переход на ARGB формат для заполнения пустых пространств прозрачными пикселями (ARGB (0,0,0,0)).

Проблема с некорректным нахождением объектов, в данный момент не совсем ясно из-за чего она возникает, решим в будущем, предварительно проблема может быть в том, что библиотека OpenCV неспособна выдавать корректный результат, возможно в будущем подключим другие модули для компьютерного зрения и распознавания объектов.

В следующем этапе разработки мы планируем:

- Разработать алгоритм построения маршрута, взаимодействия с картой.
- Продумать модель следования маршруту БПЛА автономно от станции-сервера, протестировать в полевых условиях.
- Воплотить прототип интерфейса в полноценную рабочую среду взаимодействия с БПЛА.
- Дополнить систему обнаружения и поиска объектов, автоматизировать её, откалибровать и протестировать в реальных полевых условиях.

Список использованных источников

1. Граничин О.Н., Сергеев С.Ф., Лень И.А. и др. Самоорганизация и искусственный интеллект в группах автономных роботов (методология, теория, практика): монография Санкт-Петербург: Изд-во ВВМ. – 2020. – 124 с.
2. Амелин К.С., Миллер А.Б. Алгоритм уточнения местонахождения легкого БПЛА на основе калмановской фильтрации измерений пеленгационного типа. // Информационные процессы. – 2013. – Т. 13. – № 4. – С. 338–352.