ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ №4(31), 2016, с. 287–303

УДК 004.032.24, 004.932.72'1

Д. Н. Степанов

Параллельная программная система для проведения полунатурных экспериментов по визуальной навигации беспилотных летательных аппаратов

Аннотация. Статья посвящена разработке программной системы, предназначенной для решения навигационной задачи для беспилотных летательных аппаратов с использованием методов и алгоритмов компьютерного зрения, обработки и анализа изображений. Система функционирует на кластерной вычислительной установке (КВУ). Источниками данных для решения задачи визуальной навигации являются полунатурные данные — результаты генерации видеорядов с виртуального БПЛА, для чего служит подсистема моделирования полета и генерации видео. Разработаны алгоритмы решения задачи визуальной навигации БПЛА как при полете над плоской местностью, так и над местностью с рельефом. Представлены результаты экспериментов, демонстрирующие эффективность применения КВУ в задаче предварительной подготовки эталонных изображений местности для последующего решения визуальной навигации, а также в задаче параллельной обработки нескольких независимых видеопоследовательностей, поступающих от разных БПЛА.

Ключевые слова и фразы: визуальная навигация, программная система, кластерная вычислительная установка, БПЛА, моделирование, GPU, параллельные вычисления.

Введение

Методы и алгоритмы обработки и анализа изображений и видео применяются для решения задач в большом количестве прикладных областей, в том числе для визуальной навигации беспилотных летательных аппаратов (БПЛА). Актуальность развития методов визуальной навигации объясняется тем, что точность спутниковых навигационных систем (GPS/ГЛОНАСС) зависит от качества радиосигнала, а для инерциальных навигационных систем характерно

Работа поддержана Министерством образования и науки РФ, соглашение №14.607.21.0012 (уникальный идентификатор RFMEFI60714X0012).

© Д. Н. Степанов, 2016
© Институт программных систем имени А. К. Айламазяна РАН, 2016
© Программные системы: теория и приложения, 2016

накопление ошибки в вычислениях. Существуют различные программные системы, предназначенные для решения навигационных задач для БПЛА с помощью компьютерного зрения и содержащие средства для имитационного моделирования полетов БПЛА. Приведем примеры таких систем.

- В работах [**1**, **2**] описана программная система синтезированного видения для помощи пилоту летательного аппарата в задаче посадки на взлетно-посадочную полосу. Предложен алгоритм определения положения и ориентации летательного аппарата относительно взлетно-посадочной полосы на основе структурного подхода: выделения линий разметки на изображениях с помощью модификации преобразования Хафа. Далее, положение и ориентация аппарата определяется в ходе решения оптимизационной задачи, в качестве начального приближения для неизвестных параметров используются показания спутниковой и инерциальной навигационных систем. На синтезированных тестах авторами получена приемлемая точность позиционирования летательного аппарата.
- В работе [**3**] описана система взаимной привязки изображений для решения нескольких задач, в том числе для обеспечения безопасной посадки. Для проведения экспериментов, движение камеры моделируется с помощью *MATLAB/Simulink* и *VRM*L. Взаимная привязка осуществляется на основе сопоставления кадров видеоряда и эталонных изображений с помощью алгоритма *SIFT*. Но при этом, накладывается ограничение на возможное движение камеры: между смежными кадрами, ориентация камеры и расстояние до наблюдаемой сцены не меняются.
- В статье [**4**] представлены результаты работ ЗАО «Транзас» в области компьютерного зрения применительно к задачам навигации обработки информации, получаемой с БПЛА: решение задач распознавания найденных объектов, восстановления 3D-рельефа, сшивки карт, точного целеуказания, счисления пути и навигации. Задача обучения классификатора найденных объектов является достаточно ресурсоемкой, для ускорения обучения разработана система параллельных вычислений на нескольких вычислительных узлах, объединенных сетью Ethernet.
- В диссертации [**5**] и цикле связанных с ней статей описывается программный комплекс навигации БПЛА по видеоряду и взаимной привязки изображений земной поверхности на основе сопоставления особых точек на эталонных снимках и кадрах видеоряда.

Особые точки выделяются алгоритмом *SIFT*. На эталонных снимках выделяются опорные участки — характерные объекты или регионы, в незначительной степени подверженные изменениям в зависимости от условий наблюдения (объекты искусственного происхождения или характерные особенности ландшафта). Для имитации (моделирования) полета и проведения полунатурных экспериментов используется авиасимулятор *FlightGear*. Перед началом виртуального полета пользователь выбирает нужный маршрут, устанавливает параметры моделирования, указывает используемые алгоритмы анализа видеоданных и навигации.

Общими недостатками предыдущих исследований и существующих систем является следующее:

- слабо изучена проблема использования эталонных изображений значительных размеров: если эталоны имеют большое разрешение и/или высокое пространственное разрешение, то взаимная привязка изображений становится ресурсоемкой задачей;
- алгоритмы реализованы для исполнения на обычных персональных компьютерах, которые чаще всего не могут обеспечить быструю обработку больших объемов данных. Особенно это касается тестирования алгоритмов визуальной навигации, сопоставления и взаимной привязки изображений, когда необходимо выбрать оптимальные алгоритмов и их настройки в зависимости от характера исходных данных, например, технических характеристик видеокамеры, условий движения БПЛА и др. — то есть провести обработку и анализ больших данных за приемлемое время.

Таким образом, актуальным является создание программной системы, которая бы использовала средства высокопроизводительных вычислений (в частности, кластерную вычислительную установку — КВУ) и позволяла бы проводить разработку, исследование и тестирование алгоритмов позиционирования по фото и видео. Требуется также разработать методы работы с большими эталонными изображениями земной поверхности, которые бы позволили использовать существующие наработки в области взаимной привязки и сопоставления изображений при значительных размерах последних.

1. Краткое изложение разработанных подходов для решения задачи визуальной навигации БПЛА

Подробное описание разработанных методов и алгоритмов позиционирования по видео изложено в предыдущих публикациях [6–8],

поэтому ограничимся кратким изложением сути предложенных решений.

(*1*) На этапе предварительной подготовки данных, с помощью утилиты *gdal2tiles* [**9**] из эталонного изображения того участка земной поверхности, над которой БПЛА будет совершать свои полеты, создается пирамида изображений, разделенных на небольшие фрагменты (тайлы). Подобное представление позволяет выполнять взаимную привязку видеокадров местности с эталонными изображениями значительных размеров. На каждом тайле каждого слоя пирамиды с помощью алгоритма *SURF* [**10**] детектируются особые точки и, соответственно, вычисляются *SURF*–дескрипторы этих точек. Эталонное изображение местности при этом должно иметь геопривязку.

(*2*) Если на рассматриваемой области полетов БПЛА рельеф не является плоским, и на нем присутствуют перепады высот, то на каждом тайле среди особых точек, найденных на тайле, выбираются несколько и принимаются в качестве опорных точек. Для каждой опорной точки вычисляются ее 3D–координаты (с помощью карты высот) и так называемый джет Габора, который является результатом свертки области вокруг точки с каждым из 40 фильтров Габора [**11**] и является описателем (дескриптором) опорной точки.

(*3*) На этапе решения задачи позиционирования по видео, для каждого кадра видеоряда с камеры на БПЛА, на кадре выделяются особые точки и сопоставляются с особыми точками, предварительно найденными на фрагментарном многомасштабном представлении эталонного изображения местности (пирамида изображений, разделенных на фрагменты). При этом используется результаты решения задачи позиционирования для предыдущего видеокадра и разработанные алгоритмы выбора слоя пирамиды, подмножества тайлов со слоя и сопоставления особых точек.

(*4*) Если на рассматриваемой области полетов БПЛА перепадами высот можно пренебречь (местность является плоской), то после получения множества пар общих особых точек на видеокадре и на тайлах, вычисляется матрица гомографии H [**12**], которая связывает два снимка плоской сцены: обрабатываемое и эталонное изображение. Ориентация и положение камеры на БПЛА в глобальной системе координат вычисляется на основе декомпозиции матрицы H на матрицу поворота и вектор параллельного переноса.

(5) Если же перепадами высот пренебречь нельзя, то после получения множества пар общих особых точек вычисляется фундаментальная матрица F [12], связывающая видеокадр и эталонное изображение. С помощью матрицы F, а также информации о положении и ориентации камеры на БПЛА, вычисленных для предыдущего кадра видеоряда, выполняется геометрическая коррекция текущего кадра с целью минимизации различий в масштабе и ориентации между видеокадром и эталонным изображением местности.

(6) После выполнения геометрической коррекции видеокадра, на нем выполняется поиск образов опорных точек с использованием разработанного метода на основе вычисления джетов Габора для пикселей видеокадра. После нахождения координат достаточного количества опорных точек на кадре, задача определения положения и ориентации камеры сводится к задаче внешней калибровки камеры [13]. Исходными данными в этой задаче является множество пар вида «2D–координаты опорной точки на видеокадре — 3D–координаты опорной точки в глобальной системе координат».

2. Подсистема моделирования полета БПЛА и генерации видеоряда

Для проведения экспериментальных исследований и тестирования алгоритмов визуальной навигации необходима тестовая база данных — видеоролики, записанные с камер на БПЛА. Была разработана подсистема моделирования полета БПЛА и генерации видеоряда с виртуальной камеры, в силу следующих причин:

- проведение активных экспериментов с реальными данными возможно далеко не всегда (плохие погодные условия для полетов, сложности с управлением БПЛА на больших высотах и др.);
- для оценки погрешности разработанных алгоритмов позиционирования необходима информация об истинных значениях положения и ориентации БПЛА в те моменты, когда был выполнен каждый кадр видео. Любая существующая система позиционирования (GPS, ГЛОНАСС, инерциальные навигационные системы) вычисляет положение и ориентацию камеры с некоторой погрешностью, что затрудняет оценку точности алгоритмов визуальной навигации;
- полунатурные эксперименты позволяют ускорить тестирование и отладку алгоритмов, поскольку возможна генерация большого количества видеороликов при различных параметрах движения камеры и при ее различных технических характеристиках.

Моделирование осуществляется по следующей схеме [14]:

(1) задание контрольных точек траектории движения виртуальной камеры на уменьшенной копии эталонного изображения местности;
(2) генерация траектории полета: вычисление положения и ориентации камеры в промежуточных точках полета между каждой парой соседних контрольных точек P_i и P_{i+1}. Каждая промежуточная точка соответствует моменту времени, когда камера выполнила очередной снимок;
(3) генерация кадров видеоряда для каждой промежуточной точки с помощью системы трехмерного моделирования. Для генерации используется эталонное изображение местности (спутниковый снимок). В том случае, если необходимо промоделировать полет БПЛА над местностью с рельефом, для генерации видео используется также карта высот данного участка земной поверхности. Генерация выполняется на клиентском компьютере, созданные кадры передаются на КВУ;
(4) сохранение всех кадров в единый видеофайл — результат моделирования. Также в текстовый файл сохраняется информация об эталонных значениях положения и ориентации камеры в каждой промежуточной точке траектории, для возможности оценки погрешностей алгоритмов визуальной навигации.

3. Система управления распараллеливанием вычислений

В качестве платформы для разработки и реализации программной системы был использован разрабатываемый в ИПС им. А. К. Айламазяна РАН программный инструмент *DPS* (*Distributed Processing System*) [15], предназначенный для потоковой обработки данных в конвейерно–параллельном режиме. Задача разбивается на несколько независимых подзадач, каждая подзадача решается отдельным модулем, такие создаются прикладными программистами. Модули разрабатываются на языке C++, компилируются в динамически подгружаемые библиотеки и объединяются в логические схемы решения задач. Схемы в общем случае представляют собой направленные ориентированные графы. За планирование вычислений и распределение данных в процессе работы схемы отвечает вычислительное ядро и интеллектуальный планировщик. При этом в качестве отдельных вычислителей могут выступать как узлы кластерной вычислительной установки (КВУ), так и обычные персональные компьютеры, и малогабаритные микрокомпьютеры.

Инструмент *DPS* позволяет создавать параллельные программы на языке C++ для КВУ без необходимости прямого использования программных библиотек, предназначенных для распределения данных и операций между процессами (например, библиотека *MPI*) или между потоками (*pthread*, *OpenMP*, *boost::thread*, *TBB*). Для параллельного решения поставленных задач с использованием *DPS* были разработаны следующие программные компоненты:

- программная библиотека *UAV_utils*, которая содержит реализации разработанных алгоритмов и состоит из набора объектно-ориентированных модулей (классов);
- логическая схема для *DPS* и высокоуровневый программный модуль *UAV_prepare* для подготовки в параллельном режиме эталонных данных, необходимых для решения задачи визуальной навигации БПЛА;
- логическая схема для *DPS* и высокоуровневый программный модуль *UAV_navigate* для решения задачи взаимной привязки видеокадров из нескольких видеопоследовательностей (навигации БПЛА по видео) в параллельном режиме.

В разработанной библиотеке *UAV_utils*, каждый класс предназначен для решения определенной подзадачи. У всех модулей определен метод *create()*, который вызывается при инициализации модуля, в этом методе происходит считывание настроек из единого конфигурационного XML-файла под названием *UAV_params.xml*, а также инициализация всех приватных переменных, определяющих внутреннее состояние модуля и настройки алгоритмов. Модули *UAV_prepare* и *UAV_navigate* написаны с использованием библиотеки *UAV_utils*.

Управление распараллеливанием в задаче предобработки тайлов пирамиды эталонных изображений показано на рис. 1. Каждому экземпляру модуля *UAV_prepare* присваивается свой уникальный номер. На основании этих номеров, все тайлы пирамиды эталонов равномерно распределяются между всеми модулями для дальнейшей обработки. Таким образом, достигается равномерная загруженность каждого модуля, и отсутствует необходимость в передаче данных между модулями.

Управление распараллеливанием в задаче одновременного решения задачи визуальной навигации для нескольких видео показано на рис. 2. Каждому экземпляру модуля *UAV_navigate* также присваивается свой уникальный номер. Экземпляр каждого модуля читает

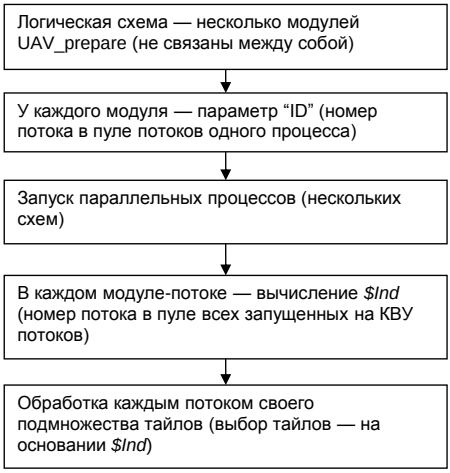

Рис. 1. Управление распараллеливанием предобработки эталонных изображений (тайлов пирамиды)

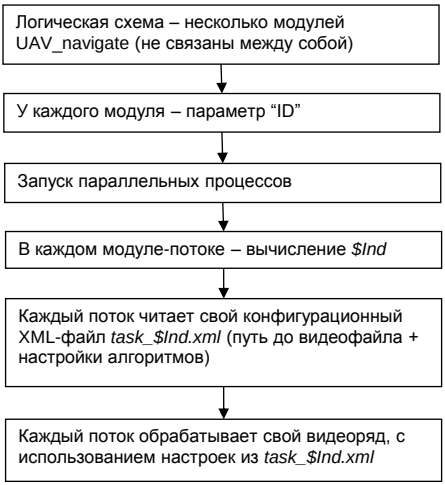

Рис. 2. Управление распараллеливанием решения задачи визуальной навигации при обработке нескольких независимых видеорядов

свой дополнительный конфигурационный XML-файл, название файлов имеет структуру *task_ $Ind.xml*, где *$Ind* — уникальный номер. Каждый экземпляр обрабатывает свой видеоряд, с использованием конкретных значений настроек алгоритмов. Результаты работы каждого потока заносятся в выходные файлы с уникальными именами, зависящими от номера потока. Таким образом, достигается параллельная обработка нескольких видеорядов с использованием *DPS*.

4. Результаты экспериментов по параллельной предобработке эталонных изображений с помощью многоядерных процессоров

Суть эксперимента состояла в поиске особых точек и вычислении их дескрипторов с помощью алгоритма SURF на всех тайлах пирамиды эталонных изображений местности, а также вычислении дескрипторов опорных точек с помощью вейвлетов Габора. Для экспериментов использовался спутниковый снимок размером 10000×10000 пикселей (пространственное разрешение — 1 метр на пиксель) и карта высот размером 1000×1000 пикселей (10 метров на пиксель). Построенная пирамида изображений содержит в общей сложности 2139 тайлов.

Также использовалась КВУ, состоящая из фронтэнда и трех вычислительных узлов. На фронтэнде, первом и втором узлах — по два 4-ядерных процессора *Intel Xeon E5410* (2.33 Ггц, кэш 6 Мб), на третьем — два 4-ядерных процессора *Intel Xeon X5570* (2.93 Ггц, кэш 8 Мб). Дисковое хранилище расположено отдельно. Непосредственно вычисления выполнялись только на узлах, на фронтэнде выполнялась управляющая программа, которая отвечает за распределение данных по узлам.

На рис. 3 показана зависимость времени обработки всех 2139 тайлов от количества запущенных в рамках *DPS* параллельных процессов, для разного числа вычислительных потоков внутри каждого процесса. На рис. 4 — зависимость ускорения от числа процессов и потоков (по сравнению со временем обработки одним процессом в один поток).

Наибольшее ускорение (около 16 раз) достигается на следующих параметрах запуска: 8 процессов по 3 потока, 6 процессов по 4 потока, 3 процесса по 7 потоков. Таким образом, предложенная структура системы распараллеливания вычислений позволила ускорить подготовку эталонных данных для решения задачи визуальной навигации.

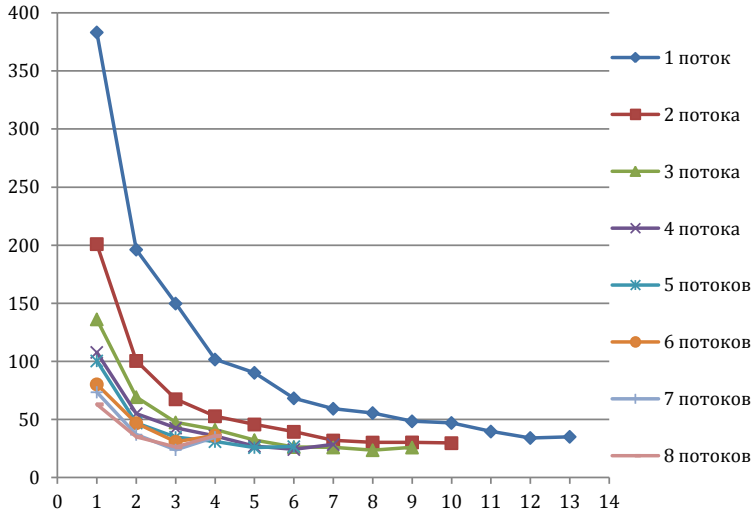

Рис. 3. Зависимость времени обработки эталонных изображений (в сек.) от количества запущенных процессов для разного числа потоков внутри одного процесса

5. Результаты экспериментов по параллельной предобработке эталонных изображений с помощью графических процессорных устройств

Два из трех вычислительных узлов кластера, используемого в экспериментах, были оснащены графическими ускорителями (*GPU*): NVIDIA Tesla K20c и NVIDIA Tesla C1060. Вычислительный эксперимент был схож с тем, что было описано в предыдущем разделе, с той лишь разницей, что поиск особых точек *SURF* на тайлах и вычисление их дескрипторов осуществлялся на *GPU*. На рис. 5 показана зависимость времени обработки всех 2139 тайлов от количества процессов, запущенных в рамках *DPS*, для разного числа вычислительных потоков внутри каждого процесса. Как видим, использование даже всего двух графических ускорителей позволило достичь времени обработки, сопоставимого со временем, которое было получено в экспериментах с применением всех 24 доступных ядер с трех вычислительных узлов.

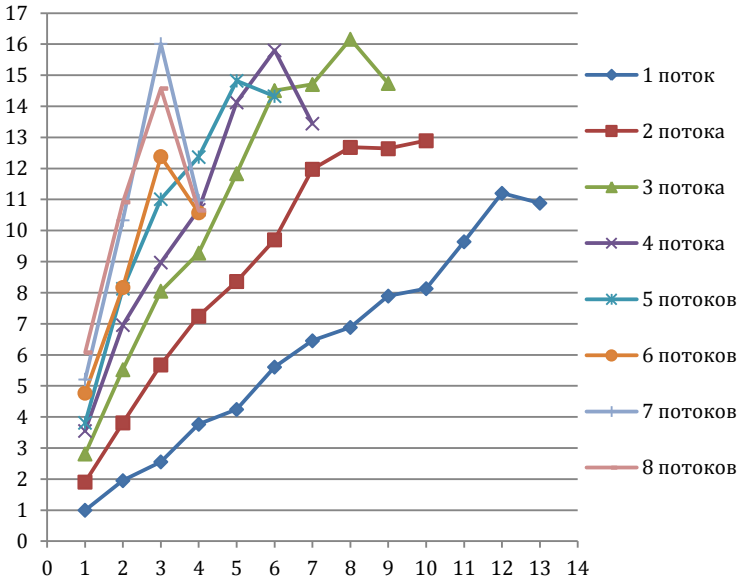

Рис. 4. Зависимость ускорения обработки эталонных изображений (в разах) от количества запущенных процессов для разного числа потоков внутри каждого процесса

6. Результаты экспериментов по параллельному решению задачи визуальной навигации для нескольких видео

Суть эксперимента состояла в следующем: вначале выбирался некоторый видеоряд и набор настроек всех алгоритмов, запускалась программа для решения задачи позиционирования по видеоряду (1 процесс, 1 вычислительный поток), замерялось общее время работы программы. По результатам серии запусков вычислялось среднее время обработки всех кадров видеоряда при конкретных настройках алгоритмов, обозначим это время как t_0. Разрешение сгенерированных видеокадров составляло 1024×768 пикселей.

Далее, выполнялась параллельная обработка нескольких наборов исходных данных: запускалось несколько параллельных процессов, в каждом процессе — несколько параллельных потоков, каждый поток обрабатывает свой видеоряд. Для чистоты эксперимента и адекватной оценки эффективности параллельной обработки, все потоки работали

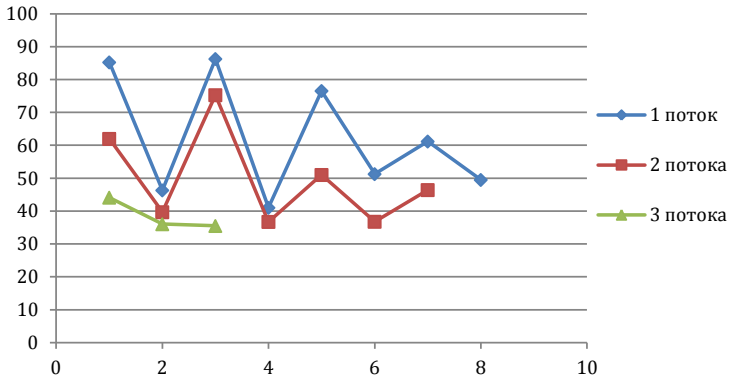

Рис. 5. Зависимость времени обработки эталонных изображений (в сек.) от количества запущенных процессов для разного числа потоков внутри одного процесса (вычисления на GPU)

с одним и тем же видеорядом и с одинаковыми настройками. Среднее время обработки $p \cdot q$ наборов исходных данных с помощью p параллельных процессов и q потоков в каждом процессе обозначим как t_{pq} (среднее — по результатам нескольких запусков). Ускорение вычислялось по следующей формуле:

$$Acc\,(p, q) = \frac{t_0 \cdot p \cdot q}{t_{pq}}. \tag{1}$$

Графики, демонстрирующие ускорение в зависимости от количества процессов, для разного числа вычислительных потоков внутри каждого процесса, показаны на рис. 6.

Наибольшее ускорение (21–23 раза) достигается на следующих параметрах запуска: 24 процесса по 1 потоку, 12 процессов по 2 потока, 8 процессов по 3 потоков, 6 процессов по 4 потока — во всех этих случаях общее число потоков равно общему числу доступных вычислительных ядер на трех узлах, на каждом из которых — по два 4-ядерных процессора ($3 \times 2 \times 4 = 24$). Таким образом, предложенная структура системы распараллеливания вычислений позволила ускорить обработку нескольких видеорядов в задаче навигации по видео.

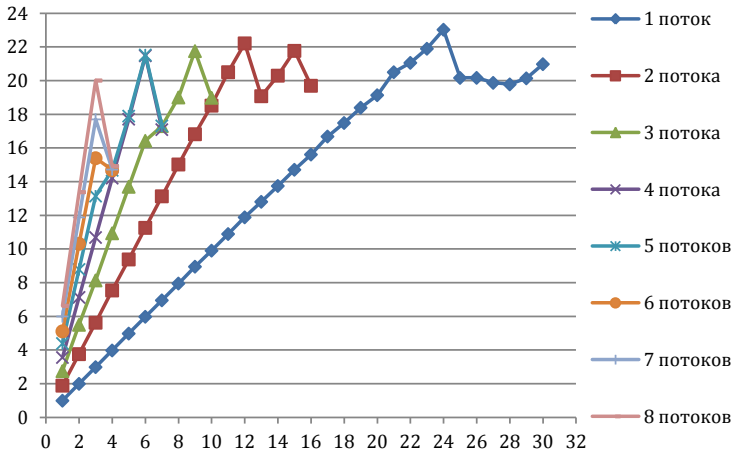

Рис. 6. Зависимость ускорения обработки видеорядов (в разах) от количества процессов при разном числе потоков внутри каждого процесса

Заключение

Разработанная программная система успешно применяется для тестирования алгоритмов визуальной навигации БПЛА. Предложенная система управления распараллеливанием вычислений обеспечивает ускорение, близкое к линейному (при увеличении числа вычислительных ядер), в задаче параллельной предподготовки эталонных данных и в задаче параллельной обработки нескольких видео. В качестве исходных данных используются результаты полунатурного моделирования, в дальнейшем планируется использовать программную систему для решения задач навигации по видеоданным с реальных БПЛА.

Список литературы

[1] Д. В. Комаров, Ю. В. Визильтер, О. В. Выголов. «Разработка алгоритма автоматического обнаружения взлетно-посадочной полосы на видеоизображениях», Научно-техническая конференция «Техническое зрение в системах управления-2011» (Институт космических исследований РАН, Москва, 15–17 марта 2011 г.), с. 81 (дата обращения: 12.11.2016), URL: http://tvcs2011.technicalvision.ru/docs/Abstracts2011.pdf ↑ 288

[2] M. A. Lebedev, D. G. Stepaniants, D. V. Komarov, O. V. Vygolov, Yu. V. Vizilter, S. Yu. Zheltov. “A Real-Time Photogrammetric Algorithm for Sensor and Synthetic Image Fusion with Application to Aviation Combined Vision”, ISPRS Technical Commission III Symposium (Zurich, Switzerland, 5–7 September, 2014), *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, **XL-3** (2014), pp. 171–175, URL: http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-3/171/2014/isprsarchives-XL-3-171-2014.pdf ↑ 288

[3] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, S. Longhi. “A Vision-Based Guidance System for UAV Navigation and Safe Landing using Natural Landmarks”, *Journal of Intelligent and Robotic Systems*, **57**:1–4 (2009), pp. 233–257. ↑ 288

[4] А. Е. Семенов, Е. В. Крюков, Д. П. Рыкованов, Д. А. Семенов. «Практическое применение технологий компьютерного зрения в решении задач распознавания, восстановления 3D, сшивки карт, точного целеуказания, счисления пути и навигации», *Известия ЮФУ. Технические науки*, 2010, №3(104), с. 92–102. ↑ 288

[5] А. Б. Фельдман. *Алгоритмы оценивания параметров объектов на изображениях для бортовых систем обработки информации и управления летательных аппаратов*, Диссертация на соискание ученой степени канд. техн. наук, Рязанский государственный радиотехнический университет, 2013, 177 с. ↑ 288

[6] Д. Н. Степанов. «Методы и алгоритмы определения положения и ориентации беспилотного летательного аппарата с применением бортовых видеокамер», *Программные продукты и системы*, **1**:1 (2014), с. 150–157 (дата обращения: 12.11.2016), URL: http://swsys.ru/index.php?page=article&id=3776 ↑ 289

[7] Д. Н. Степанов. «Методики сопоставления особых точек в задаче визуальной навигации БПЛА», *Вестник Южно-Уральского государственного университета. Серия «Вычислительная Математика и Информатика»*, **4**:4 (2015), с. 32–47. ↑ 289

[8] Д. Н. Степанов. «Применение вейвлетов Габора в задаче навигации БПЛА с использованием видеокамеры», *Фундаментальные исследования*, 2015, №12-1, с. 85–92. ↑ 289

[9] Утилита gdal2tiles (дата обращения: 12.11.2016), URL: http://www.klokan.cz/projects/gdal2tiles ↑ 290

[10] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. “SURF: Speeded Up Robust Features”, *Computer Vision and Image Understanding*, **110**:3 (2008), pp. 346–359. ↑ 290

[11] J.-K. Kamarainen, V. Kyrki, H. Kalviainen. “Invariance Properties of Gabor Filter Based Features Overview and Applications”, *IEEE Transactions on Image Processing*, **15**:5 (2006), pp. 1088–1099. ↑ 290

[12] R. Hartley, A. Zisserman. *Multiple View Geometry In Computer Vision*, 2nd edition, Cambridge University Press, 2003, 670 p. ↑ 290,291

[13] А. Кравцов, В. Вежневец. «Общая формулировка задачи внешней калибровки камеры», *Компьютерная графика и мультимедиа*, 2003, №1(2) (дата обращения: 12.11.2016), 5 с., URL: http://ict.edu.ru/ft/002404/num1krav.pdf ↑ 291

[14] Д. Н. Степанов, И. П. Тищенко. «Программная система для моделирования полета БПЛА и его автономного позиционирования с использованием методов и алгоритмов технического зрения», *XIV национальная конференция по искусственному интеллекту с международным участием КИИ-2014*. Т. 3 (Казань, Россия, 24–27 сентября 2014 г.), с. 333–341. ↑ 292

[15] A. Kondratyev, I. Tishchenko. “Concept of Distributed Processing System of Images Flow in Terms of π-Calculus”, 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT 2016) (ITMO University, Saint Petersburg, Russia, 18–22 April, 2016), pp. 131–137. ↑ 292

Рекомендовал к публикации *д.т.н. В. М. Хачумов*

Пример ссылки на эту публикацию:

Д. Н. Степанов. «Параллельная программная система для проведения полунатурных экспериментов по визуальной навигации беспилотных летательных аппаратов», *Программные системы: теория и приложения*, 2016, **7**:4(31), с. 287–303.

URL: http://psta.psiras.ru/read/psta2016_4_287-303.pdf

Об авторе:

Дмитрий Николаевич Степанов

Инженер-исследователь ИПС им. А. К. Айламазяна РАН, автор 20 публикаций. Область научных интересов: компьютерное зрение, анализ изображений, распознавание образов, параллельное программирование, методы оптимизации.

e-mail: mitek1989@mail.ru

Dmitry Stepanov. *Parallel program system for HIL experiments on visual navigation of unmanned aerial vehicles.*

Abstract. The article is devoted to the development of a program system designed to solve the navigation problem for unmanned aerial vehicles using the methods and algorithms of computer vision, image processing and analysis. The system operates on a cluster computer. The sources of data for solving the problem are the visual navigation are the HIL data — the results of generation of video sequences from virtual UAV. A subsystem of flight simulation and video generation is developed. Also the algorithms for solving the problem of visual navigation of UAV in flight over flat terrain and terrain with relief are developed. The results of experiments demonstrating the effectiveness of cluster computer in the problem of preliminary processing of reference images of terrain for subsequent visual navigation solutions, as well as in the problem of parallel processing of multiple independent video sequences coming from different UAVs are presented. (*In Russian*).

Key words and phrases: visual navigation, program system, cluster computer, UAV, modeling, GPU, parallel computing.

References

[1] D. V. Komarov, Yu. V. Vizil'ter, O. V. Vygolov. "Development of Algorithm of Runway's Automatic Detection on the Videoframes", Nauchno-tekhnicheskaya konferentsiya "Tekhnicheskoye zreniye v sistemakh upravleniya-2011" (Institut kosmicheskikh issledovaniy RAN, Moskva, 15–17 marta 2011 g.), pp. 81 (data obrashcheniya: 12.11.2016) (in Russian), URL: http://tvcs2011.technicalvision.ru/docs/Abstracts2011.pdf

[2] M. A. Lebedev, D. G. Stepaniants, D. V. Komarov, O. V. Vygolov, Yu. V. Vizilter, S. Yu. Zheltov. "A Real-Time Photogrammetric Algorithm for Sensor and Synthetic Image Fusion with Application to Aviation Combined Vision", ISPRS Technical Commission III Symposium (Zurich, Switzerland, 5–7 September, 2014), *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, **XL-3** (2014), pp. 171–175, URL: http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-3/171/2014/isprsarchives-XL-3-171-2014.pdf

[3] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, S. Longhi. "A Vision-Based Guidance System for UAV Navigation and Safe Landing using Natural Landmarks", *Journal of Intelligent and Robotic Systems*, **57**:1–4 (2009), pp. 233–257.

[4] A. Ye. Semenov, Ye. V. Kryukov, D. P. Rykovanov, D. A. Semenov. "Computer Vision Techniques for Recognition, 3D Extraction, Map Stitching, Targeting and Navigation", *Izvestiya YuFU. Tekhnicheskiye nauki*, 2010, no.3(104), pp. 92–102 (in Russian).

[5] A. B. Fel'dman. *Algorithms for Objects Parameters' Estimation on Images for Aircrafts' On-Board Systems of Data Processing and Control*, Dissertatsiya na soiskaniye uchenoy stepeni kand. tekhn. nauk, Ryazanskiy gosudarstvennyy radiotekhnicheskiy universitet, 2013 (in Russian), 177 p.

© D. N. Stepanov, 2016
© Ailamazyan Program Systems Institute of RAS, 2016
© Program systems: Theory and Applications, 2016

[6] D. N. Stepanov. “Methods and Algorithms for Determining the Position and Orientation UAV Using On-Board Videocameras”, *Programmnyye produkty i sistemy*, **1**:1 (2014), pp. 150–157 (data obrashcheniya: 12.11.2016) (in Russian), URL: http://swsys.ru/index.php?page=article&id=3776

[7] D. N. Stepanov. “Techniques of Feature Points Matching in the Problem of UAV’s Visual Navigation”, *Vestnik Yuzhno-Ural’skogo gosudarstvennogo universiteta. Seriya “Vychislitel’naya Matematika i Informatika”*, **4**:4 (2015), pp. 32–47 (in Russian).

[8] D. N. Stepanov. “Use of Gabor Wavelets in the Problem of UAV’s Navigation Using Video Camera”, *Fundamental’nyye issledovaniya*, 2015, no.12-1, pp. 85–92 (in Russian).

[9] gdal2tiles (data obrashcheniya: 12.11.2016) (in Russian), URL: http://www.klokan.cz/projects/gdal2tiles

[10] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. “SURF: Speeded Up Robust Features”, *Computer Vision and Image Understanding*, **110**:3 (2008), pp. 346–359.

[11] J.-K. Kamarainen, V. Kyrki, H. Kalviainen. “Invariance Properties of Gabor Filter Based Features Overview and Applications”, *IEEE Transactions on Image Processing*, **15**:5 (2006), pp. 1088–1099.

[12] R. Hartley, A. Zisserman. *Multiple View Geometry In Computer Vision*, 2nd edition, Cambridge University Press, 2003, 670 p.

[13] A. Kravtsov, V. Vezhnevets. “The General Formulation of the External Camera Calibration Problem”, *Komp’yuternaya grafika i mul’timedia*, 2003, no.1(2) (data obrashcheniya: 12.11.2016) (in Russian), 5 p., URL: http://ict.edu.ru/ft/002404/num1krav.pdf

[14] D. N. Stepanov, I. P. Tishchenko. “Software Systems for UAV Flight Simulation and its Autonomous Positioning Using Vision Methods and Algorithms”, *XIV natsional’naya konferentsiya po iskusstvennomu intellektu s mezhdunarodnym uchastiyem KII-2014*. V. 3 (Kazan’, Rossiya, 24–27 sentyabrya 2014 g.), pp. 333–341 (in Russian).

[15] A. Kondratyev, I. Tishchenko. “Concept of Distributed Processing System of Images Flow in Terms of π-Calculus”, 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT 2016) (ITMO University, Saint Petersburg, Russia, 18–22 April, 2016), pp. 131–137.

Sample citation of this publication:

Dmitry Stepanov. “Parallel program system for HIL experiments on visual navigation of unmanned aerial vehicles”, *Program systems: Theory and applications*, 2016, **7**:4(31), pp. 287–303. (*In Russian*).

URL: http://psta.psiras.ru/read/psta2016_4_287-303.pdf