

С.М. Соколов, А.А. Богуславский, С.А. Романенко

**ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ДЛЯ БОРТОВЫХ СИСТЕМ
ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ ПОДВИЖНЫХ СРЕДСТВ
С ИСПОЛЬЗОВАНИЕМ СТЗ***

Целью исследования является повышение эффективности информационного обеспечения автономных подвижных средств с использованием зрительных данных и технологий его разработки за счёт рационального использования нетрадиционных вычислителей и специальной подготовки алгоритмического обеспечения. Использование нетрадиционных, гетерогенных вычислительных средств позволяет существенно расширить круг задач обработки зрительных данных в масштабе реального времени и, тем самым, увеличить ситуационную осведомлённость автономных РТК и оперативность её получения. Рациональное использование нетрадиционных вычислителей требует существенной переработки алгоритмического обеспечения. Большинство алгоритмов применяемых при обработке изображений, были разработаны в расчёте на реализацию на традиционных, фон-Неймановских архитектурах вычислителей и требуют существенных усилий для реализации на параллельных структурах и разработки специальных инструментальных средств программирования. В работе сделан акцент на исследования в области использования ПЛИС и рассмотрен ряд подходов в специальной подготовке необходимого алгоритмического обеспечения. В качестве модельных, демонстрационных примеров использования нетрадиционных вычислителей рассмотрена реализация таких алгоритмов обработки зрительных данных, которые активно используются в широком круге задач информационного обеспечения целенаправленных перемещений мобильных РТК. Описана подготовка и реализация на ПЛИС таких алгоритмов, как построение гистограмм, вычисление оптического потока, сегментация изображений. Приведены результаты экспериментов на действующих макетах бортовых вычислителей. В качестве исходных данных использованы зрительные данные, собранные при движении мобильных средств в условиях естественной среды. Всё используемое программное обеспечение выполнено на основе унифицированного каркаса программного обеспечения систем технического зрения реального времени отечественной разработки. В заключении обсуждаются дальнейшие шаги в указанном направлении, с учётом стремления к использованию отечественных программно-аппаратных средств.

Мобильные РТК; бортовые вычислители; реальное время; алгоритмическое обеспечение; система технического зрения.

S.M. Sokolov, A.A. Boguslavsky, S.A. Romanenko

**SOFTWARE AND HARDWARE FOR MOBILE MEANS INFORMATION
SUPPORT ONBOARD SYSTEMS WITH USE OF VISION SYSTEM**

Research objective is increase of autonomous mobile means information support efficiency with use of visual data and technologies of its development due to rational use of nonconventional calculators and special preparation of algorithmic providing. Use of nonconventional, heterogeneous computing means allows to expand significantly a circle of visual data processing problems on the real time scale and, thereby, to increase situational awareness of autonomous robots and efficiency of its receiving. Rational use of nonconventional calculators demands essential alteration of algorithmic providing. The majority of visual data algorithms were developed counting on realization on traditional, von Neumann architecture of calculators and demand essential efforts for realization on parallel structures and developments of special programming tools. In work the emphasis on researches in the area use FPGA is placed and a number of approaches in special preparation of necessary algorithmic is considered. As model, demonstration examples of use of nonconventional calculators realization of such algorithms of visual data processing which are

* Работа выполнена при поддержке РФФИ, проект № 19-07-00113.

actively used in a wide range of information support problems of purposeful movements of mobile robot is considered. Preparation and realization on FPGA of such algorithms as creation of histograms, an optical flow calculation, segmentation of images is described. Results of experiments are given in the operating models of airborne computers. As basic data the visual data collected at the movement of mobile means in the conditions of habitat are used. All used software is executed on the basis of the unified software framework of the real time vision systems of domestic development. In the conclusion further steps in the specified direction, taking into account aspiration to use of domestic software and hardware are discussed.

Situational awareness; vision system; mobile robot; nonconventional calculators; algorithmic providing; airborne computers; real time.

Введение. Тенденции мобильной робототехники говорят о всё возрастающей роли ситуационной осведомлённости и оперативности принятия решений в информационно-управляющих системах современных и перспективных робототехнических комплексов (РТК). Информационное обеспечение подвижных средств, особенно системы технического зрения (СТЗ) в его составе, требует всё возрастающих вычислительных ресурсов и интеллектуальных технологий в алгоритмическом обеспечении. Нарастание мощности отдельных процессоров практически исчерпано (плотность ядер внутри одной микросхемы достигло физических пределов). Увеличение производительности возможно в направлении архитектурных решений, реконфигурируемых высокопроизводительных гетерогенных ресурсов, оптимизации коммуникационных взаимодействий и эффективной работе с памятью.

Активно ведутся работы по созданию различных специализированных процессоров [1–3]. Мировой опыт и опыт наших исследований говорят об эффективности сочетаний последовательной и параллельной обработки данных – организации гетерогенных/интеллектуальных вычислительных ресурсов [4–6]. Следует указать, что те же тенденции развития робототехники говорят о том, что для успешного внедрения в практику, необходимо обеспечить технологичность предлагаемых решений и их экономическую целесообразность. С учётом всё возрастающей доли программного обеспечения в стоимости РТК вопросы эффективности и технологичности изготовления алгоритмического обеспечения таких комплексов приобретают большое значение.

В работе рассматривается общая схема использования нетрадиционных вычислителей в процессе сбора и обработки зрительных данных, описывается опыт реализации алгоритмического обеспечения СТЗ с использованием отдельных элементов гетерогенных средств, обсуждаются перспективы развития этого направления. В качестве алгоритмов, исследуемых на первом этапе выбраны алгоритмы, которые по опыту предыдущих исследований актуальны для задач информационного обеспечения целенаправленных перемещений, навигационных задач.

Схема использования вычислительных ресурсов при сборе и обработке зрительных данных. Повышение степени автономности и полная автономность подвижных робототехнических комплексов, действующих в неподготовленных условиях, не возможны без использования систем технического зрения и эффективной реализации интеллектуального алгоритмического обеспечения [7–10]. Одними из ключевых требований, предъявляемых к алгоритмическому обеспечению (АО) подвижных средств является работа в масштабе реального времени, определяемым скоростями перемещений, и использование ограниченных бортовых вычислительных ресурсов. Эффективное/рациональное распределение работ по специализированным вычислителям – сложная задача, требующая разнообразных исследований, в частности, ответа на вопрос: какой исполнитель эффективнее решает какой тип задач. Если для таких традиционных навигационных средств, как БИНС, ГНСС, одометрия, взаимодействие с удалённым оператором, не требуется

специальных усилий в обеспечении работы в масштабе реального времени на базе традиционных вычислителей, то для сбора и обработки зрительных данных в масштабе реального времени такие усилия необходимы.

Практически с первых исследований систем компьютерного видения были выделены такие этапы в сборе и анализе зрительных данных, как предобработка, выделение первичных признаков объектов на изображениях и формирование/использование моделей объектов для распознавания объектов и сцен. В зависимости от этапа требуется обработка различного количества зрительных данных и, главное, различных операций над этими данными.

Первый этап – сбор и предобработка зрительных данных. При первоначальном включении системы, при смене условий наблюдения, потере объекта интереса в процессе прослеживания или необходимости смены/изменения приоритетов в анализе наблюдаемой сцены, требуется анализ качества вводимых данных и, при необходимости их корректировка для дальнейшей обработки. Алгоритмы соответствующих операций применительно к используемым сенсорам (построение и анализ, эквализация гистограмм, изменение контраста и яркости и т.п.) хорошо известны, но должны быть применены к большим массивам данных, собираемых со всего поля зрения. Здесь требуется аппаратное ускорение соответствующих процедур. Кроме того, должен быть обеспечен доступ к исходному изображению, в том случае если потребуется другой вид обработки.

На втором этапе – выделение первичных признаков объектов на изображениях. В зависимости от предыстории обработки (объекты интереса в сцене уже были выделены и осуществляется их прослеживание или осуществляется первоначальный поиск или поиск после потери объекта в текущем поле зрения) может потребоваться обработка больших массивов данных, вплоть до обработки полного кадра. Сами алгоритмы выделения тех или иных признаков объектов (сегментация, выделение полей характеристик, нахождение краёв, описание формы областей и т.п.) заранее известны. На этом этапе также требуется аппаратное ускорение соответствующих процедур.

Третий этап – формирование/использование моделей для распознавания объектов и сцен. На этом этапе количество входных данных не так велико, а большее значение приобретает анализ различных комбинаций имеющихся признаков и их допустимых сочетаний в соответствии с известными моделями. В алгоритмическом обеспечении применяется много эвристических алгоритмов, а в последние годы используются нейросетевые модели.

В реализации описанной последовательности этапов (схемы) следует учитывать то, что из-за неверно выбранной стратегии/гипотезы распознавания или изменения условий наблюдения или самой сцены может потребоваться выбор/формирование другой гипотезы и, как следствие, получение других признаков на тех же исходных данных. Это обстоятельство обеспечивается возможностью «отката» к исходным данным на любом этапе обработки – хранению и обеспечению доступа к текущему кадру (кадрам, если речь идёт об обработке зрительных данных с нескольких полей зрения) до принятия решения об окончании обработки.

Компоновка макета интеллектуальной вычислительной среды. В настоящее время известно много попыток использования группировок GPU для обработки зрительных данных нейроподобными алгоритмами при решении задач классификации [16]. Более продвинутыми разработками считаются так называемые тензорные процессоры – TPU. Первые данные об использовании подобных процессоров стали известны от фирмы Google. Архитектура и особенности Google TPU хранились в строжайшем секрете, но теперь стала доступна некоторая информация [4–6]. В описываемых исследованиях мы реализовывали компоновку

отдельных элементов вычислительной среды из доступных элементов. В качестве прототипа бортовой аппаратной платформы для реализации предложенной схемы рассматриваем гетерогенный вычислитель "грифон" [11, 12]. Здесь в качестве системной магистрали также используется шина PCI Express. На первом этапе исследований мы использовали шину GigEthernet, но всё программное обеспечение комплекса допускает переход на целевую шину PCI Express (рис. 1).

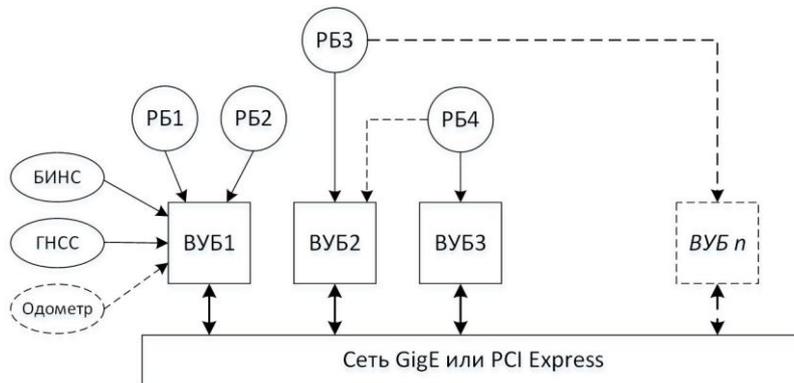


Рис. 1. Общая схема компоновки аппаратных средств СТЗ и традиционных информационных систем для информационного обеспечения мобильных РТК.
 РБ – регистрирующий блок (видеокамера) с определённым полем зрения;
 ВУБ – вычислительно-управляющий блок. ГНСС – глобальная навигационная спутниковая система

Для создания прикладного ПО мы разработали и используем каркас ПО СТЗ реального времени [14, 15]. Этот крупномасштабный программный шаблон (Application Framework), определяет структуру и основные возможности конкретного приложения. Его назначение:

- ◆ повышение степени повторного использования ПО
- ◆ использование типовых решений для сокращения временных затрат на разработку ПО
- ◆ упрощение реализации ПО СТЗ для похожих прикладных задач

На рис. 2 представлена общая структурная схема этого каркаса. На рис. 3 – структура ядра каркаса ПО СТЗ РВ.

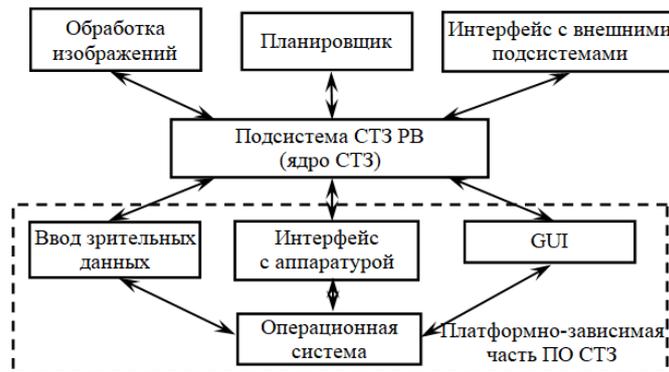


Рис. 2. Схема общей структуры каркаса ПО СТЗ РВ

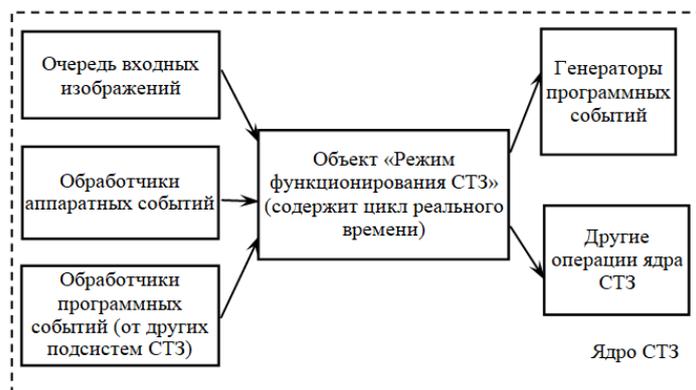


Рис. 3. Структура ядра каркаса ПО СТЗ РВ

Результаты первых экспериментов. Автономной подвижной платформы, способной перемещаться по разнообразной местности и имеющей хорошо описанный интерфейс авторам не было доступно, поэтому эксперименты с прототипом бортовой системы информационного обеспечения предлагаемой архитектуры проводились на различных не автономных транспортных средствах. Алгоритмы для детального исследования и сравнительного анализа способов реализации на нетрадиционных вычислителях отбирались из числа хорошо зарекомендовавших себя алгоритмов в традиционной фон Неймановской реализации при решении типовых задач информационного обеспечения целенаправленных перемещений РТК. На описываемом этапе исследований в перечень алгоритмов мы включили алгоритмы: построения гистограммы заданного фрагмента или всего поля зрения; вычисления плотного и разреженного оптического потока; сегментации.

Данные собранные в реальных условиях полигона служили исходным материалом для сравнительного анализа и отладки технологии реализации алгоритмического обеспечения с использованием нетрадиционных вычислительных ресурсов. На рис. 4 представлен общий вид одного из транспортных средств с установленными модулями СТЗ. На рис. 5 – примеры полей зрения.



Рис. 2. Регистрирующие блоки модулей СТЗ, установленные на крыше автомобиля «НИВА»

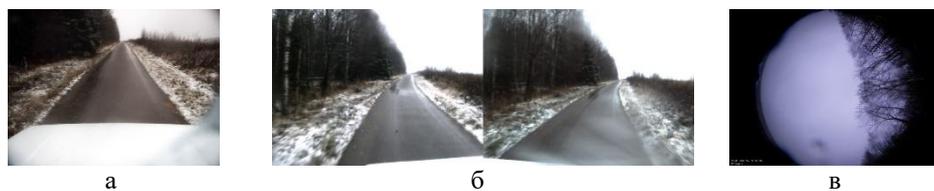


Рис. 3. Примеры полей зрения модулей СТЗ, установленных на автомобиле «Нива»: а – вперёд смотрящего модуля с моно камерой; б – вперёд смотрящего стерео модуля; в – панорамного модуля типа «рыбий глаз»

Работа с FPGA-модулем для вычисления гистограммы. Аппаратная часть этого элемента вычислительно-управляющего блока с нетрадиционным вычислителем включает в себя следующие компоненты:

- ◆ Плата с установленной FPGA Altera (на базе чипа Altera Cyclone V SE (5CSEMA4U23C6N), на которой реализованы операции приема изображений от IP-камеры и вычисление гистограммы.
- ◆ Камера высокого разрешения «Контур HD» производства ИПЗ с поддержкой передачи изображений по протоколу IP в сети Gigabit Ethernet.
- ◆ Компьютер общего назначения под управлением ОС Windows, в среде которой запущено приложение СТЗ для отображения входных изображений и результатов их обработки

Поскольку данный чип основан на технологии System-on-a-chip (SoC), программное обеспечение было разделено на три подсистемы.

Первая подсистема реализована на стороне FPGA, она включает в себя обработку управляющих команд, взаимодействие с микросхемами ЦАП/АЦП, а также сами алгоритмы приема, передачи, и навигации.

Вторая подсистема исполняется на стороне так называемой Hard Processor System (HPS), эта система фактически является драйвером операционной системы Linux и является прослойкой между пользовательским интерфейсом и FPGA.

Третья подсистема реализует непосредственно интерфейс управления устройством, выполнена она в виде веб-интерфейса и включает себя front-end и back-end.

Для взаимодействия между системами была разработана система команд, определяющих режим работы устройства, а также такие внутренние параметры как частота излучения, величины защитных интервалов, минимальный размер пакета передаваемых данных и т.п. Взаимодействие между подсистемами осуществляется через прерывания, POSIX сигналы и HPS-to-FPGA интерфейса.

В разработке использовались языки программирования C, C++, Verilog, Javascript. В процессе работы были оттестированы алгоритмы работы устройства как в режиме передачи информации, так и в навигационном режиме, для чего было произведено модульное тестирование с использованием модулей-заглушек, а также системное тестирование в лабораторных и натуральных экспериментах. Для снижения энергопотребления были использованы оптимизации кода на стороне FPGA, поскольку только в этой подсистеме существует возможность значительно повлиять на величину энергопотребления.

Особенностью данной модельной гетерогенной вычислительной системы является использование камеры «Контур-HD» с поддержкой одного физического подключения GigE и двух логических каналов, по которым осуществляется передача изображений различного разрешения. Для вычисления гистограммы на вход FPGA-вычислителя поступают изображения разрешением 640x480 пикселей (с кадровой частотой 25 Гц), а на головной компьютер СТЗ – изображения высоко-

го разрешения 2592x1944 (с кадровой частотой 12 Гц). В результате между FPGA-вычислителем и компьютером СТЗ отсутствует обмен данными изображений и производится только передача результатов обработки (расчет гистограммных признаков).

Работа с FPGA-модулем для вычисления оптического потока. Модуль для вычисления оптического потока реализован в виде платы с 8-ю вычислительными блоками FPGA Xilinx. Доступ к этому модулю со стороны компьютера СТЗ осуществляется с использованием промежуточного хост-компьютера под управлением ОС Unix. В качестве интерфейса для обмена данными используется интерфейс сокетов в сети с протоколами TCP/IP. Входные изображения поступают в приложение СТЗ, работающее под управлением ОС Windows. По протоколу TCP/IP через выделенный порт эти изображения через хост-компьютер передаются на вход FPGA-вычислителя. Результатами обработки являются значения компонент векторов оптического потока, которые вычисляются для каждой пары последовательных изображений. Результаты в виде массива двухбайтных целых чисел также передаются посредством сокетов на компьютер с приложением СТЗ.

В данной конфигурации камера не подключена непосредственно к FPGA-вычислителю, что приводит к необходимости дополнительного трафика в используемой сети. В результате при применении метода вычисления оптического потока Лукаса-Кэнада обеспечивается частота обработки 10 Гц при размере входных изображений 1600x1200 пикселей.

Также были проведены эксперименты по вычислению плотного оптического потока методом Фарнбека. Достигнута производительность порядка 10 Гц при размере входных изображений 320x240 пикселей.

Анализ и подготовка алгоритма для сегментации изображений. Для исследования способов ускорения этого активно используемого алгоритма был выбран алгоритм сегментации изображений Фельценсвальба (и Хуттенлохера) [17]. Имеются весьма успешные реализации этого алгоритма для CPU (последовательно работающего процессора с памятью произвольного доступа) [18, 19].

Этот алгоритм, в свою очередь использует алгоритм Крускала, который строит остовный лес минимальной стоимости для неориентированного взвешенного графа. Алгоритм Крускала, виртуозно использует все возможности и особенности классического CPU. Именно из-за этого он плохо переносится на вычислительные устройства других типов (вроде GPU или схем, реализуемый в FPGA). Известен алгоритм Борувки, который является более “параллельным”, хотя был опубликован в 1926 году, когда о вычислительной технике речи ещё не шло и никак не учитывались свойства CPU (= фон-неймановской машины). В результате, у Борувки получился алгоритм, в котором гораздо больше “естественного” параллелизма, чем в алгоритме Крускала [20]. Позднее этот алгоритм переоткрывался (и публиковался) несколько раз. Сейчас существуют “параллельные” реализации алгоритма Борувки. Мы тщательно изучили вариант реализации алгоритма Борувки предложенный Васконселос (с соавторами) [21].

В отличие от других вариантов, для алгоритма Васконселос имеются исходные тексты, из которых можно узнать некоторые тонкости и детали реализации, которые не очевидны из текста статьи. Доступны два варианта реализации: для CPU и для GPU. Можно эти два варианта сравнивать.

Вариант для GPU реализован на основе платформы CUDA, которая даёт возможность реализовывать параллельные вычисления для GPU фирмы NVidia.

Отсюда возникла идея модифицировать алгоритм Фельценсвальба, вырезав из него алгоритм Крускала и заменив его на алгоритм Борувки в варианте Васконселос.

При этом выяснилось, что вариант Васконселос невозможно просто взять – и чисто механически вставить в алгоритм Фельценсвальба. Причины этого – следующие.

Результат работы алгоритма Васконселос – дерево минимальной стоимости (для связного графа). А результат сегментации изображения – это несколько сегментов. Поэтому, алгоритм анализа графа должен выдавать не дерево, а лес (каждое дерево в котором соответствует некоторому сегменту изображения).

У Васконселос работа алгоритма заканчивается в тот момент, когда все компоненты склеиваются в одну компоненту. Но если результат работы – несколько компонент (лес, а не одно дерево), то критерий завершения работы должен быть другой. Результатом работы алгоритма Васконселоса является список рёбер составляющих остовное дерево минимальной стоимости. Но, в случае сегментации изображений, эта информация – бесполезна! А вместо этого требуется получить отображение вершин исходного графа (соответствующих пикселям изображения) на номера сегментов.

У Фельценсвальба исходный граф является связным, но результат сегментации графа – несколько компонент графа (соответствующих сегментам изображения). Это достигается за счёт того, что связный граф превращается в несвязный путём удаления из графа некоторых “тяжёлых” рёбер. Для этого в алгоритм Крускала сделана вставка, которая удаляет рёбра, так сказать, “на лету”, прямо в процессе работы алгоритма Крускала. Логически проще было бы удалить “тяжёлые” рёбра заранее, до запуска алгоритма Крускала. Но это сделать нельзя, поскольку требуется учитывать не только вес самого ребра, но ещё и размеры компонент, соединяемых этим ребром. Чем больше компонента – тем менее “охотно” она склеивается с другими компонентами. Но в алгоритме Васконселос удаление “тяжёлых” рёбер не предусмотрено. Поэтому, необходим метод удаления “тяжёлых” рёбер, совместимый с алгоритмом Васконселос (и поддающийся распараллеливанию). В алгоритме Фельценсвальба, помимо удаления “тяжёлых” рёбер реализовано и противоположное действие – принудительное склеивание маленьких компонент друг с другом, а так же – принудительное приклеивание маленьких компонент к большим. Поэтому, необходимы соответствующие дополнения к алгоритму Васконселос (поддающиеся распараллеливанию).

В настоящий момент, вышеупомянутые переделки алгоритма Васконселос разработаны и реализованы [22]. Собственно говоря, основным “результатом” является не реализация как таковая, а то, что показана реализуемость вышеупомянутых доработок. И что, в результате соединения алгоритма Фельценсвальба и доработанного алгоритма Васконселос получается алгоритм сегментации, который работоспособен и производит результаты похожие на результаты исходного алгоритма Фельценсвальба. Возникает естественный вопрос «Почему не удастся добиться полной тождественности результатов исходного и переработанного алгоритма Фельценсвальба»? Основная проблема состоит в том, что полной тождественности результатов (при разном способе вычисления) можно добиться только в том случае, если желаемый результат имеет однозначное математическое определение.

Например, если веса всех рёбер в ориентированном взвешенном графе различны – то существует только один остовный лес минимальной стоимости. Поэтому, если есть два алгоритма нахождения этого леса, и эти алгоритмы корректны, то они выдадут один и тот же результат, даже если вычисляют его абсолютно разным способом. Если же веса разных рёбер могут совпадать, то всё равно можно добиться единственности результата с помощью искусственного приёма. А именно – пронумеровав рёбра. И в случаях, когда два ребра имеют одинаковый вес, будем считать более “лёгким” то ребро, номер у которого меньше.

Но в случае сегментации изображений нет критерия, который однозначно бы определял, что следует считать “правильным” результатом сегментации. Неформально задача ставится так, что результат сегментации должен “выглядеть разумно” с точки зрения человека. Т.е., этот результат должен соответствовать психологии человеческого восприятия. А восприятие у разных людей – разное. Поэтому, Фельценсвальб взял за основу алгоритм Крускала (результат работы которого определён однозначно и математически) и приделал к этому алгоритму некоторые “эвристические” добавки. Эти добавки не имеют строгого обоснования, и их оправданием является исключительно то, что они – “работают”, т.е. приводят к результатам сегментации, которые выглядят “разумно” с человеческой точки зрения. При этом, результат работы алгоритма Фельценсвальба не определён однозначно! У алгоритма есть несколько параметров, “подкручивая” которые можно влиять на результаты сегментации. Параметры – такие.

float sigma.

Стандартное отклонение по Гауссу. Влияет на степень размытия изображения при предварительной обработке.

float scale.

“Ширина поля зрения”. Увеличение этого параметра приводит к увеличению размеров сегментов.

int min_size.

Минимальный размер сегмента. Сегменты, размер которых меньше, чем min_size, принудительно склеиваются друг с другом или приклеиваются к сегментам, размер которых – больше, чем min_size. Таким образом, уменьшая scale, мы можем волонтаристски сдерживать рост размеров сегментов, а увеличивая min_size - подавлять маленькие сегменты.

Понятно, что при таких условиях можно не соблюдать строгую эквивалентность между исходным алгоритмом и его “распараллеленной” версией. Оправданием разумности такого нарушения эквивалентности должна служить “разумность” получаемых результатов. Например, в исходном алгоритме Фельценсвальба, каждая вершина приписана к компоненте, размер которой известен. А при склеивании компонент учитывается размер компонент. И при этом мгновенно вычисляется размер компоненты, которая получилась в результате склеивания двух компонент. Но в случае алгоритма Васконселос было бы проблематично реализовать мгновенный пересчёт размеров компонент, поскольку в каждый момент времени рассматривается много рёбер и может попарно объединяться много компонент. И это делается с помощью нескольких итераций. После каждой итерации, список компонент перестраивается. При этом, размеры компонент перевычисляются между итерациями, но не в процессе каждой итерации. Это является некоторой аппроксимацией к тому, что делается в исходном алгоритме Васконселос. Поэтому, полного совпадения результатов не получается. Но результаты сегментации получаются похожими. Таким образом, реализован работоспособный “гибрид” алгоритма Фельценсвальба и алгоритма Васконселос (для CPU). Он конструировался с расчётом на последующее распараллеливание на GPU. Вариант для GPU на платформе CU Vasconcellos DA был проверен. Следующим шагом предполагается переделка его на платформу OpenCL для реализации на FPGA. Каковы отличия CUDA и OpenCL?

CUDA – это “фирменная” платформа. И она позволяет распараллеливать алгоритмы только для GPU фирмы NVidia.

OpenCL – это открытый стандарт, который позволяет распараллеливать алгоритмы для GPU нескольких фирм (в частности – фирм NVidia и Intel).

Кроме того, OpenCL позволяет распараллеливать алгоритмы для изделий от фирмы Xilinx.

Заключение. В работе описан первый этап исследований в компоновке и программно-алгоритмическом обеспечении перспективных бортовых гетерогенных вычислительных средств мобильных робототехнических комплексов, оснащённых системами технического зрения. К настоящему моменту промакетированы отдельные элементы предлагаемой программно-аппаратной архитектуры и осуществлены эксперименты по реализации алгоритмического обеспечения. Предложено объединяющее ПО, но использованы разнотипные аппаратные реализации. Получены первые обнадеживающие результаты. С поддержкой заинтересованных организаций, предоставивших аппаратную базу, в ближайшей перспективе будет осуществлён перенос макетных решений на единую платформу ГРИФОН. Будет продолжена работа по автоматизации диспетчерских функций при распределении задач обработки между специализированными вычислителями. Естественно, большое поле приложения сил открывается в продолжение оптимизации и настройки алгоритмов обработки зрительных данных под нетрадиционного исполнителя, разработке методики и инструментальных средств создания информационных систем для конкретного типа мобильных РТК.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Марьясин А.В., Маниленко И.Н., Михайлов М.А., Перов И.А., Самарин О.Ф., Савостьянов В.Ю., Ильин Е.М.* Цифровая вычислительная машина для современных малогабаритных бортовых радиолокационных станций // Вестник СибГУТИ. – 2016. – № 3. – С. 15-31.
2. *Дашевский В.П., Бизин М.М.* Обзор возможностей бортовых вычислителей // Доклады ТУСУРа. – Сентябрь 2015. – № 3 (37). – С. 924-929.
3. *Лукин Н.А.* Бортовые функционально-ориентированные процессоры на основе однородных вычислительных сред для мобильных систем реального времени // Фундаментальные исследования. – 2015. – № 12.
4. Чип Google Tensor Processor упростит процесс машинного обучения и вернёт силу закону Мура. 3DNews. – URL: <https://3dnews.ru/933239>.
5. Блок Google TPU второго поколения в задачах машинного обучения демонстрирует производительность выше, чем у GPU Nvidia GV100. iXBT.com. – URL: <https://www.ixbt.com/news/2017/05/19/google-tpu-gpu-nvidia-gv100.html>.
6. Подробности о тензорном сопроцессоре Google TPU. Servernews. – URL: <https://servernews.ru/957543>.
7. Molaletsa Namoshe, Oduetse Matsebe and Nkgatho Tlale. Sensor Fusion and Its Applications // Feature extraction: techniques for landmark based navigation system. – P. 347-374.
8. *Nützi G. et al.* Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM // Journal of Intelligent & Robotic System. – 2011. – Vol. 61. – P. 287-299.
9. *Engel J., Schöps T., Cremers D.* LSD-SLAM: Large-Scale Direct Monocular SLAM // Proc. of the ECCV 2014. – 2014. – P. 834-849.
10. *Labbé M., Michaud F.* RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library // Journal of Field Robotics. – 2018. – P. 1-31.
11. *Gomez-Ojeda R. et al.* PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments // IEEE Transactions on Robotics. – 2019. – Vol. 35, No. 3. – P. 734-746.
12. *Галаган П., Кузьминский Л., Сорокин А.* Решение задач машинного зрения на базе гетерогенной платформы ГРИФОН // Control Engineering Россия. – 2018. – № 1 (73). – С. 38-43.
13. *Галаган П.* Платформа ГРИФОН для решения задач встраиваемых систем специального назначения // Современные технологии автоматизации. – 2015. – № 4.
14. *Boguslavsky A.A., Sokolov S.M.* Component Approach to the Applied Visual System Software Development // 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003), July 27-30, Orlando, Florida, USA, 2003.
15. *Sokolov S.M., Boguslavsky A.A.* Intellectual Images Processing for a Realtime Recognition Problem // Proc. The 2nd Intern. Multi-Conf. on Complexity, Informatics and Cybernetics (IMCIC2011), Orlando, Florida, USA, March 27th-30th, 2011, Orlando, Florida, USA. – Vol. II. – P. 406-411.

16. *Barlas G.* Multicore and GPU Programming: An Integrated Approach. – Morgan Kaufmann, 2014. – 698 p.
17. *Vasconcellos J.F. de A., Cáceres E.N., Mongelli H. and Song S.W.* A Parallel Algorithm for Minimum Spanning Tree on GPU // 2017 International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW), Campinas, 2017. – P. 67-72.
18. *Vasconcellos J.F. de A., Cáceres E.N., Mongelli H., Song S.W., Dehne F., & Szwarcfiter J.L.* New BSP/CGM algorithms for spanning trees // The International Journal of High Performance Computing Applications. – 2019. – Vol. 33 (3). – P. 444-461.
19. *Pedro F. Felzenszwalb and Daniel P. Huttenlocher.* Efficient Graph-Based Image Segmentation // *Int. J. Comput. Vision* 59, 2 (September 2004). – P. 167-181.
20. Реализация алгоритма Коускена. – URL: <http://cs.brown.edu/people/pfelzens/segment/>.
21. *Borůvka Otakar.* O jistém problému minimálním [About a certain minimal problem]. *Práce Moravské přírodovědecké společnosti. V Brně III* (in Czech and German). – 1926. – No. 3. – P. 37-58.
22. Проекты Gitlab: Реализация алгоритма Vasconcellos – URL: https://gitlab.com/sergei-romanenko/vasconcellos-new-mst/-/tree/master/img_seg_seq.

REFERENCES

1. *Mar'yasin A.V., Manilenko I.N., Mikhaylov M.A., Perov I.A., Samarin O.F., Savost'yanov V.Yu., Il'in E.M.* Tsifrovaya vychislitel'naya mashina dlya sovremennykh malogabaritnykh bortovykh radiolokatsionnykh stantsiy [Digital computer for modern small-size onboard radar stations], *Vestnik SibGUTI* [Vestnik SibGUTI], 2016, № 3, pp. 15-31.
2. *Dashevskiy V.P., Bizin M.M.* Obzor vozmozhnostey bortovykh vychisliteley [An overview of the capabilities of onboard computers], *Doklady TUSURa* [Proceedings of TUSUR], Sentyabr' 2015, No. 3 (37), pp. 924-929.
3. *Lukin N.A.* Bortovye funktsional'no-orientirovannye protsessory na osnove odnorodnykh vychislitel'nykh sred dlya mobil'nykh sistem real'nogo vremeni [Onboard functional-oriented processors based on homogeneous computing environments for mobile real-time systems], *Fundamental'nye issledovaniya* [Fundamental study], 2015, No. 12.
4. Chip Google Tensor Processor uprostit protsess mashinnogo obucheniya i vernet silu zakonu Mura. 3DNews [The Google Tensor Processor chip will simplify the machine learning process and restore the power of Moore's law. 3DNews]. Available at: <https://3dnews.ru/933239>.
5. Blok Google TPU vtorogo pokoleniya v zadachakh mashinnogo obucheniya demonstriruet proizvoditel'nost' vyshe, chem u GPU Nvidia GV100. iXBT.com [The second-generation Google TPU block in machine learning tasks demonstrates higher performance than the Nvidia GV100 GPU. iXBT.com]. Available at: <https://www.ixbt.com/news/2017/05/19/google-tpu-gpu-nvidia-gv100.html>.
6. Podrobnosti o tenzornom soprotessore Google TPU. Servernews [Details about the Google TPU tensor coprocessor. Servernews]. Available at: <https://servernews.ru/957543>.
7. Molaletsa Namoshe, Oduetse Matsebe and Nkgatho Tlale. Sensor Fusion and Its Applications, *Feature extraction: techniques for landmark based navigation system*, pp. 347-374.
8. *Nützi G. et al.* Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM, *Journal of Intelligent & Robotic System*. 2011. Vol. 61. pp. 287-299.
9. *Engel J., Schöps T., Cremers D.* LSD-SLAM: Large-Scale Direct Monocular SLAM, *Proc. of the ECCV 2014*, 2014, pp. 834-849.
10. *Labbé M., Michaud F.* RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library, *Journal of Field Robotics*, 2018, pp. 1-31.
11. *Gomez-Ojeda R. et al.* PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments, *IEEE Transactions on Robotics*, 2019, Vol. 35, No. 3, pp. 734-746.
12. *Galagan P., Kuz'minskiy L., Sorokin A.* Reshenie zadach mashinnogo zreniya na baze geterogennoy platformy GRIFON [Solving machine vision problems based on the heterogeneous GRYPHON platform], *Control Engineering Rossiya* [Control Engineering Russia], 2018, No. 1 (73), pp. 38-43.
13. *Galagan P.* Platforma GRIFON dlya resheniya zadach vstraivaemykh sistem spetsial'nogo naznacheniya [Gryphon platform for solving problems of embedded special purpose systems], *Sovremennye tekhnologii avtomatizatsii* [Modern automation technologies], 2015, No. 4.

14. Boguslavsky A.A., Sokolov S.M. Component Approach to the Applied Visual System Software Development, *7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003)*, July 27-30, Orlando, Florida, USA, 2003.
15. Sokolov S.M., Boguslavsky A.A. Intellectual Images Processing for a Realtime Recognition Problem, *Proc. The 2nd Intern. Multi-Conf. on Complexity, Informatics and Cybernetics (IMCIC2011)*, Orlando, Florida, USA, March 27th-30th, 2011, Orlando, Florida, USA, Vol. II, pp. 406-411.
16. Barlas G. *Multicore and GPU Programming: An Integrated Approach*. Morgan Kaufmann, 2014, 698 p.
17. Vasconcellos J.F. de A., Cáceres E.N., Mongelli H. and Song S.W. A Parallel Algorithm for Minimum Spanning Tree on GPU, *2017 International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW)*, Campinas, 2017, pp. 67-72.
18. Vasconcellos J.F. de A., Cáceres E.N., Mongelli H., Song S.W., Dehne F., & Szwarcfinder J.L. New BSP/CGM algorithms for spanning trees, *The International Journal of High Performance Computing Applications*, 2019, Vol. 33 (3), pp. 444-461.
19. Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient Graph-Based Image Segmentation, *Int. J. Comput. Vision* 59, 2 (September 2004), pp. 167-181.
20. Realizatsiya algoritma Kouskena [The implementation of the algorithm Koscina]. Available at: <http://cs.brown.edu/people/pfelzens/segment/>.
21. Borivka Otakar. O jistém problému minimálním [About a certain minimal problem]. *Práce Moravské přírodovědecké společnosti. V Brně III* (in Czech and German), 1926, No. 3, pp. 37-58.
22. Proekty Gitlab: Realizatsiya algoritma Vasconcellos [GitLab projects: implementation of the Vasconcellos algorithm.]. Available at: https://gitlab.com/sergei-romanenko/vasconcellos-new-mst/-/tree/master/img_seg_seq.

Статью рекомендовал к опубликованию д.т.н. М.Ю. Медведев.

Соколов Сергей Михайлович – Федеральное государственное учреждение «Федеральный исследовательский центр Институт прикладной математики им. М.В. Келдыша Российской академии наук» (ИПМ им. М.В. Келдыша РАН); e-mail: sokolism@keldysh.ru 125047, г. Москва, Миусская пл., 4; тел.: 89161220113; д.ф.-м.н.; профессор; г.н.с.

Богуславский Андрей Александрович – e-mail: anbg74@mail.ru; тел.: 89167379448; д.ф.-м.н.; в.н.с.

Романенко Сергей Анатольевич – e-mail: romansa@keldysh.ru; тел.: 84992507873; к.ф.-м.н.; в.н.с.

Sokolov Sergey Mikhailovich – Keldysh Institute of Applied Mathematics Russian Academy of Sciences; e-mail: sokolism@keldysh.ru; 4, Miusskaya sq., Moscow, 125047, Russia; phone: +74992207994; dr. of math. and physics. sc.; professor; principal researcher.

Boguslavsky Andrey Alexandrovich – e-mail: anbg74@mail.ru; phone: +79167379448; dr. of math. and physics. sc.; leading researcher.

Romanenko Sergei Anatolyevich – e-mail: romansa@keldysh.ru; phone: +74992507873; cand. of math. and physics. sc.; leading researcher.