

4. «КАМАЗ» начнет полигонные испытания беспилотного автомобиля. [Электронный ресурс] – URL: <http://www.interfax.ru/business/445663>. Международная информационная группа «Интерфакс». – 2015. (03.11.2017).

5. Павлов Б.П., Гортышов Ю.Ф., Мингалеев Г.Ф., Мельничнов В.В., Салимов Р.И. Формирование профессиональных компетенций выпускников КГТУ им. А.Н. Туполева в полипрофессиональных учебно-проектных группах с участием предприятий-работодателей / Павлов Б.П., Гортышов Ю.Ф., Мингалеев Г.Ф., Мельничнов В.В., Салимов Р.И. // Изд-во КГТУ им. А.Н. Туполева. – Казань, 2008. – 88 с.

УДК681.786.23

УПРАВЛЕНИЕ МОБИЛЬНЫМ РОБОТОМ В РЕЖИМЕ ТЕЛЕОПЕРАЦИИ: СОЗДАНИЕ НОВОГО ROS-ПАКЕТА ДЛЯ ОДНОВРЕМЕННОЙ ПЕРЕДАЧИ ВИДЕОПОТОКА С НЕСКОЛЬКИХ КАМЕР

Сафин Р.Н.; Лавренов Р.О.; Магид Е.А., профессор, Лаборатория интеллектуальных робототехнических систем, Высшая школа информационных технологий и информационных систем (ИТИС), Казанский федеральный университет, г. Казань, Россия

MOBILE ROBOT CONTROL IN TELEOPERATIONAL MODE: NEW ROS-PACKAGE FOR SIMULTANEOUS VIDEO STREAMING FROM MULTIPLE CAMERAS

Safin R.; Lavrenov R.; Magid E., PhD, professor, Laboratory of Intelligent Robotic Systems, Higher Institute for Information Technology and Information Systems (ITIS), Kazan Federal University, Kazan, Russia

Аннотация

Передача видео в режиме реального времени позволяет решать множество важных задач в сфере робототехники, включая удаленное управление роботом в режиме телеоперации. Мы разработали новый ROS-пакет, позволяющий независимо захватывать видео с четырех камер мобильного робота «Сервосила Инженер». Конфигурация и получение исходных данных с камер осуществляется при помощи библиотеки V4L2. Метод проецирования буферов устройства в память приложения повысил производительность, устранив излишнее копирование информации. Сравнительный анализ производительности нашего пакета и основанного на библиотеке OpenCV стандартного пакета показал существенное преимущество нашего пакета.

Abstract

Capturing and processing of a video stream in real time is important for a variety of tasks in robotics, including robot teleoperation. We created a new ROS-based package that captures concurrent video stream from four different cameras of a mobile robot «Servosila Engineer». We applied V4L2 API for video devices configuring and retrieving raw data from robot cameras. Memory mapping approach of mapping device buffers increased overall performance by eliminating redundant memory copies. We compared performance of our new package with a standard OpenCV library based package, and our package demonstrated significantly better results.

Ключевые слова: видеостриминг, V4L2, ROS, OpenCV, мобильный робот.

Key words: videostreaming, V4L2, ROS, OpenCV, mobile robot.

Работа осуществлена при финансовой поддержке Российским фондом фундаментальных исследований (РФФИ) и Правительства Республики Татарстан в рамках научного проекта № 17-48-160879. Часть работ выполнена при финансовой поддержке РФФИ и Министерства науки и технологий и Государством Израиль (совместный проект ID 15-57-06010).

This work was supported by the Russian Foundation for Basic Research (RFBR, project ID 17-48-160879). Part of the work was supported by RFBR and Ministry of Science Technology & Space State of Israel (joint project ID 15-57-06010).

Введение

Захват видео потока с камер робота является важной задачей для многих целей, например, таких как визуальная одновременная локализация и картографирование (SLAM) [6], планирование маршрута [4], человеко-машинное взаимодействие [2] и управление в режиме телеоперации, в том числе при проведении поисково-спасательных операций в условиях городской застройки. В некоторых ситуациях центру управления или оператору робота необходимо получать видео данные от мобильного робота и обрабатывать их в режиме реального времени. Более того, если для беспилотных летательных аппаратов (БЛА) и примитивных роботов одной камеры может быть достаточно для успешного функционирования (например, при навигации на основе моно SLAM-алгоритмов [6]), то в сложных робототехнических системах (РТС) могут одновременно использоваться несколько камер (например, стереопара) для реализации более комплексных алгоритмов управления и навигации. Так как алгоритмы навигации сложных РТС могут требовать больших вычислительных ресурсов, нередко мобильный робот нуждается в передаче сенсорных данных на более мощное вычислительное устройство для их последующей обработки и анализа.

В данной статье мы представляем разработку нового ROS-пакета, предоставляющего функционал независимого захвата видео данных с четырех камер мобильного робота. ROS-пакет представляет собой небольшую библиотеку, работающую совместно с робототехнической операционной системой РОС (англ. ROS, Robot Operating System [8]). Наш ROS-пакет создан для использования в составе системы управления российского мобильного гусеничного робота «Сервосила Инженер» [8]. Робот оборудован четырьмя камерами и имеет клиент-серверный программный интерфейс для управления. Функционал встроенной системы управления предоставляет возможность переключения между всеми камерами в графическом интерфейсе, однако возможность одновременного получения видео данных со всех камер робота отсутствует. Для устранения этого недостатка мы создали ROS-пакет, позволяющий одновременно захватывать видео поток с четырех камер. Нашей следующей целью является передача видео потока через беспроводные сети в режиме реального времени и реализация RTP сервера, что позволит клиентам получать видео данные с робота и улучшит процесс телеоперации (то есть управления телеоператором в режиме удаленного доступа), так как телеоператор сможет получать больше информации об окружающей среде.

Аппаратный комплекс системы технического зрения робота

Мобильный робот «Сервосила Инженер» (рис. 1) оснащен четырьмя камерами, что обеспечивает ему хорошую информированность об окружающей среде. Три из четырех камер на голове робота расположены в передней части и одна камера – в задней (рис. 1):

- фронтальная камера с оптическим увеличением;
- две фронтальные широкоугольные камеры (стереопара);
- широкоугольная камера заднего вида.



Рис. 1. Мобильный робот «Сервосила Инженер»

На роботе установлена операционная система Ubuntu 14.04 LTS (Trusty Tahr). В роли бортового вычислителя используется процессор Intel® Core™ i7-3517UE 1.70ГГц с двумя физическими ядрами.

Разработка нового ROS-пакета

Библиотека V4L2

Для захвата видео с камер мы использовали вторую версию библиотеки Video4Linux. Данная библиотека была предустановлена на оригинальной системе создателями робота. Программирование V4L2 устройства состоит из следующих шагов [10], показанных на рис. 2:

- открытие устройства;
- конфигурация параметров устройства;
- выбор видеостандарта;
- конфигурация формата видеоданных;
- конфигурация методов ввода-вывода;
- цикл чтения/записи;
- закрытие устройства.

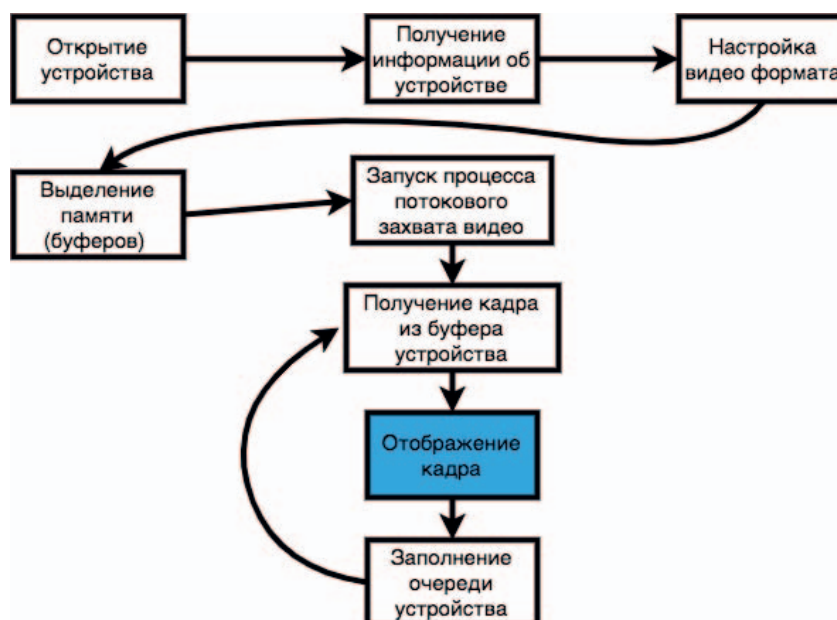


Рис. 2. Принцип работы V4L2 приложения

Параметры видеоустройства настраиваются путем вызова специальных запросов (ioctl). Запрос представляет из себя системный вызов, задающий параметры устройства (количество кадров в секунду, разрешение кадра). Процесс работы V4L2 программы (рис. 2) может изменяться в зависимости от ситуации. Например, наша реализация не отображает получаемые видео данные на дисплее. Первоначальным шагом является открытие устройства для последующей настройки его параметров для потокового захвата видео. Для того, чтобы использовать системные вызовы ioctl, файловый дескриптор устройства (камеры) должен быть открыт для операций чтения и записи.

Необходимо также настроить формат получаемых данных с каждой камеры. В нашем случае, робот «Сервосила Инженер» имеет три камеры (стереопара и камера заднего вида), которые могут выдавать кадры в формате шаблона Байера [7]. Кадры с передней камеры с оптическим увеличением представлены в цветовом пространстве yuv (yuuv422).

После настройки формата кадров видеопотока выделяется память под буферы устройства. Буфер содержит изображения с камеры и служит путем обмена данными между приложением и драйвером устройства. Для управления процессом обмена данными между приложением и устройством используется метод проецирования памяти устройства в память приложения (mmap). Это позволяет увеличить производительность захвата видео с камеры, избавляясь от избыточного копирования данных. Приложение и драйвер обмениваются указателями на данные, которые становятся доступными приложению.

Далее, совершается запрос на получение информации о выделенных буферах. Эта информация включает в себя расположение буферов в памяти (указатели) и их количестве (может отличаться от запрашиваемого размера). Процесс получения кадров состоит из двух этапов (рис. 3):

- этап заполнения очереди пустым буфером;
- этап получения буфера с данными.

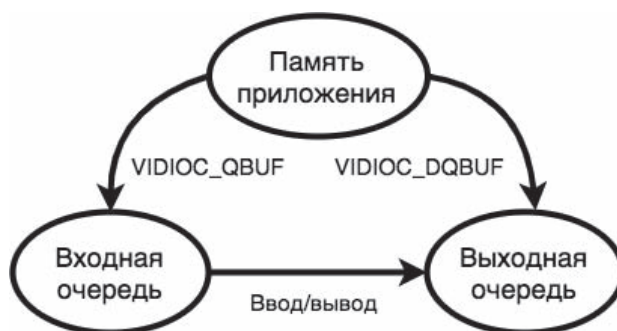


Рис. 3. Процесс получения кадров в V4L2

Для каждого этапа вызывается специальный запрос (VIDIOC_QBUF и VIDIOC_DQBUF). На первом этапе буфер вставляется в очередь драйвера устройства. Буфер будет ожидать момента, когда драйвер устройства заполнит его данными. На втором этапе обработанный буфер (с видео данными) извлекается из очереди [9].

Реализация ROS-пакета

Изначально в системе управления робота не была установлена операционная система ROS. Мы установили совместимый с Ubuntu 14.04 дистрибутив ROS Indigo. Архитектура созданного нами ROS-пакета состоит из следующих основных частей (рис. 4):

- РОС узел, независимо публикующий данные с каждой камеры;
- V4L2 прослойка, через которую осуществляется захват видеок кадров с камер.

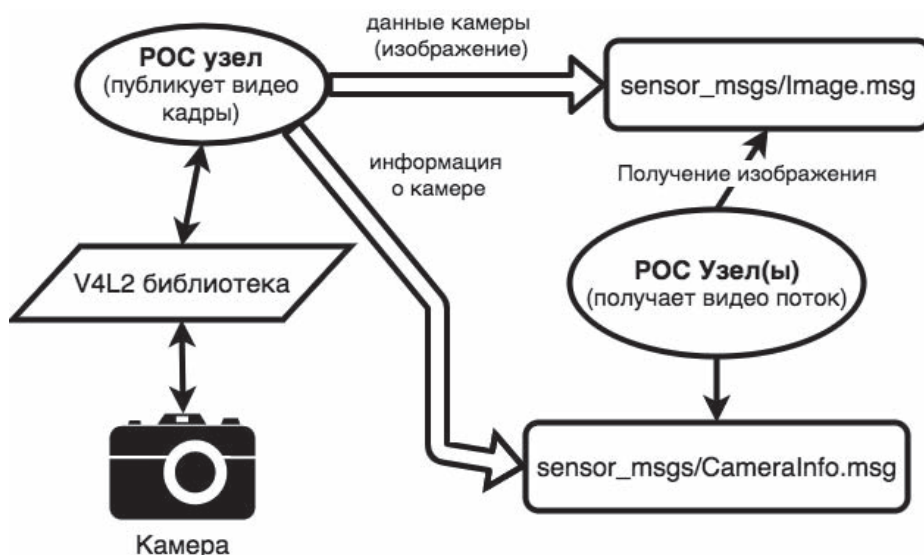


Рис. 4. Архитектура созданного ROS-пакета

Узел, публикующий данные с камеры, транслирует видеопоток в созданные топики. Любой другой узел, подписанный на созданный топик с видеоданными, может получать данные, обрабатывать их или ретранслировать. Используются следующие параметры публикующего узла:

- количество кадров в секунду задает количество видеок кадров, публикуемых в топик каждую секунду;
- разрешение кадра – высота и ширина видео кадра;
- формат ROS-кадра (например, `bayer_grbg8`);
- видеоустройство (например, `/dev/video0`).

Следует отметить, что исходное необработанное изображение с камер искажено и для дальнейшей успешной работы с этим изображением требуется предварительная калибровка камер, что является важной частью наших планов на будущее в рамках работы над проектом по созданию системы управления для робота «Сервосила Инженер» на базе ROS.

В процессе разработки мы столкнулись с несколькими практическими проблемами. Наиболее существенной из них являлась проблема получения некорректного времени захвата кадра. Данный параметр необходим для реализации алгоритмов навигации (семейства визуальных стерео SLAM алгоритмов) для синхронизации двух камер по времени. Несмотря на то, что в V4L2 есть возможность получить время заполнения буфера данными с камеры, в оригинальной имплементации библиотеки данный параметр находился в некорректном состоянии.

Решение данной проблемы было найдено, используя GitHub репозитории библиотеки OpenCV. Данные о времени захвата кадра должны быть извлечены до того, как вновь класть буфер с данными в очередь драйвера устройства.

Тестирование работы и производительности созданного ROS-пакета было проведено в системе управления мобильного робота «Сервосила Инженер» (результаты тестирования см. на рис. 5, 6). При отсутствии подписчиков на топик с видеоданными, использование ресурсов процессора было минимально (0,4%). При наличии подписчиков наш пакет использует 4,4% ресурсов процессора. Никаких дополнительных действий по сжатию или конвертации изображения в другие форматы (например, таких как RGB) не осуществлялось. Таким образом, мы можем снизить нагрузку на процессор мобильного робота, не затрачивая ресурсы на дополнительную обработку данных. Важной особенностью нашего пакета является его полная независимость от библиотеки OpenCV.

```

robot@robot-computer:~$ pidstat 10 -p 9402
Linux 4.4.0-83-generic (robot-computer)      02.11.2017      _x86_64_      (4 CPU)

11:12:07      UID      PID      %usr  %system  %guest   %CPU   CPU  Command
11:12:17      1000     9402     0,00  0,40    0,00   0,40    0  video_stream
11:12:27      1000     9402     0,00  0,40    0,00   0,40    1  video_stream
11:12:37      1000     9402     0,00  0,30    0,00   0,30    2  video_stream
11:12:47      1000     9402     0,00  0,40    0,00   0,40    0  video_stream
11:12:57      1000     9402     0,00  0,40    0,00   0,40    0  video_stream
11:13:07      1000     9402     0,00  0,40    0,00   0,40    2  video_stream
11:13:17      1000     9402     0,00  0,50    0,00   0,50    1  video_stream
robot@robot-computer:~$ pidstat 10 -p 9402
Linux 4.4.0-83-generic (robot-computer)      02.11.2017      _x86_64_      (4 CPU)

11:18:29      UID      PID      %usr  %system  %guest   %CPU   CPU  Command
11:18:39      1000     9402     0,00  3,90    0,00   3,90    1  video_stream
11:18:49      1000     9402     0,00  4,70    0,00   4,70    3  video_stream
11:18:59      1000     9402     0,00  4,20    0,00   4,20    3  video_stream
11:19:09      1000     9402     0,00  4,40    0,00   4,40    0  video_stream
11:19:19      1000     9402     0,00  4,40    0,00   4,40    1  video_stream
    
```

Рис. 5. Использование процессора нашим ROS-пакетом (замеры проведены утилитой *pidstat* в режиме простоя *idle* и в режиме нагрузки)

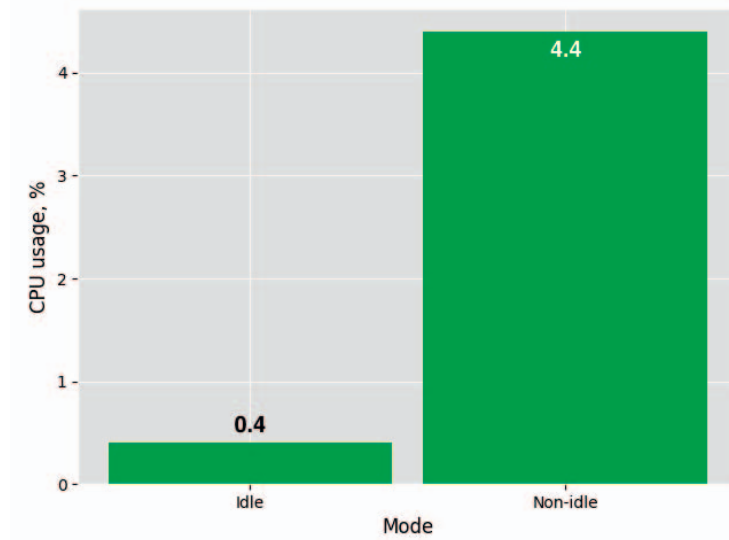


Рис. 6. Использование ресурсов процессора созданным ROS-пакетом (в режиме простоя *idle* и в режиме нагрузки)

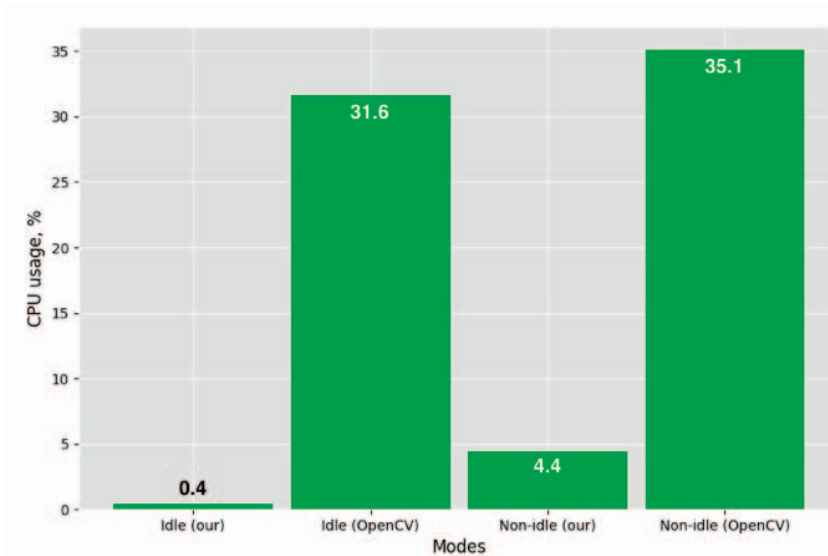


Рис. 7. Среднее использование ресурсов процессора нашим ROS-пакетом и основанным на OpenCV ROS-пакетом (в режиме простоя *idle* и в режиме нагрузки)

Мы проанализировали другие существующие ROS-пакеты потокового видео захвата для сравнительного анализа с полученными нами результатами. Наиболее интересным является пакет, основанный на библиотеке OpenCV [5]. Результаты тестов для его сравнения с нашим пакетом (рис. 7) показали, что наш пакет потребляет существенно меньше системных ресурсов как в режиме простоя idle (0,4% против 31,6%), так и в режиме нагрузки (4,4% против 35,1%). Конечно, если в потоковый захват видео вовлечено больше камер, то и нагрузка на процессор будет расти. Тем не менее, при отсутствии клиентов в режиме простоя, пакет на основе OpenCV продолжает активно использовать ресурсы системы.

Заключение

Потоковый захват видео с нескольких камер является важной задачей. Ключевым моментом является не только получение кадров с камеры, но и уменьшение количества потребляемых ресурсов, нагрузки на процессор. В данной работе мы представили реализацию ROS-пакета для одновременного захвата видео с нескольких камер мобильного робота «Сервосила Инженер».

Созданный пакет показал хорошие результаты по потреблению ресурсов как в режиме простоя, так и в режиме нагрузки, существенно превзойдя производительность стандартного ROS-пакета на основе библиотеки OpenCV. В будущем мы планируем реализовать RTP сервер, который позволит передавать видео с робота режиме реального времени через беспроводную сеть. Исходный код доступен для публичного доступа на сайте системы контроля версий GitHub [7].

Список литературы

1. Buyval A., Afanasyev I. and Magid E. Comparative analysis of ROS-based Monocular SLAM methods for indoor navigation. *Ninth International Conference on Machine Vision. International Society for Optics and Photonics* (2017), pp. 103411K-103411K-6.
2. Karpov A., Carbini S., Ronzhin A., Viallet J. Two similar different speech and gestures multimodal interfaces. *Multimodal User Interfaces*. (Berlin, Heidelberg, 2008), pp. 155-184.
3. Lukin A. and Kubasov D. High-quality algorithm for Bayer pattern interpolation. *Programming and Computer Software* 30.6 (2004), pp. 347-358.
4. Magid E., Lavrenov R., Khusianov A. Modified spline-based path planning for autonomous ground vehicle. *Int. Conf. on Informatics in Control, Automation and Robotics* (Madrid, Spain, 2017), pp.132-141.
5. GitHub video_stream_opencv ROS package: https://github.com/ros-drivers/video_stream_opencv
6. Fuentes-Pacheco J., Ruiz-Ascencio J. and Rendón-Mancha J. M., Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, (2015), v 43(1), pp. 55-81.
7. ROS-пакет одновременного потокового видео захвата с мобильного робота; https://github.com/chupakabra1996/lirs_ros_video_streaming.
8. Sokolov M., Lavrenov R., Gabdullin A., Afanasyev I. and Magid E. 3D modelling and simulation of a crawler robot in ROS/Gazebo. *4th Int. Conf. on Control, Mechatronics and Automation* (2016), pp. 61-65.
9. V4L2; <https://linuxtv.org/downloads/v4l-dvb-apis/uapi/v4l/v4l2.html>
10. Yinli L., Hongli Y., and Pengpeng Zh. The implementation of embedded image acquisition based on V4L2. *In Electronics, Communications and Control (ICECC), 2011 International Conference on IEEE*, pp. 549-552.