

УДК 681.3

РАЗРАБОТКА 3D СИСТЕМЫ ТЕХНИЧЕСКОГО ЗРЕНИЯ И ИДЕНТИФИКАЦИЙ ОБЪЕКТОВ ДЛЯ МОБИЛЬНЫХ РОБОТОВ

К.т.н. В.В. Евсеев¹, Г.В. Фоменко¹, И.Н. Малая²

1. Харьковский национальный университет радиоэлектроники

2. Государственное предприятие "Южный Государственный проектно-конструкторский научно-исследовательский институт авиационной промышленности", г. Харьков

В статье предложен новый подход до реализации систем технического зрения для мобильных роботов на базе виртуальной реальности дополненной методом идентификации объектов. Данное решение позволит создать эффект 3D погружение в окружающую среду мобильного робота, что даст возможность упростить управление и принятие решений оператором.

У статті запропоновано новий підхід до реалізації систем технічного зору для мобільних роботів на базі віртуальної реальності доповненої методом ідентифікації об'єктів. Дане рішення дозволить створити ефект 3D занурення в навколишнє середовище мобільного робота, що дасть можливість спростити управління і прийняття рішень оператором.

This article proposes a new approach to the implementation of technical systems for mobile robots based on virtual reality and method of identifying objects. This solution will create the effect of 3D immersion into the surrounding environment of mobile robot that will make possible to simplify the management and decision-making by the operator.

Ключевые слова: 3D, системы технического зрения, мобильные роботы

Введение

Использование мобильных роботов (МР) в последнее время получило широкое применение, особенно востребованными МР стали в решениях задач, в которых невозможно использовать человеческий ресурс. Исследователи с Польши, Германии работают над МР, которые позволяют проводить разведку в шахтах при возникновении чрезвычайных ситуаций (ЧС). Вследствие чего, возникает задача адекватности представления окружающего пространства МР. Для решения такой задачи используются системы технического зрения (СТЗ). Существующие СТЗ позволяют передавать оператору 2D изображения, что вызывает неполное погружение в окружающую среду и добавляет элементы управления, контроля над внешними параметрами окружающей среды (наличие вредных примесей, расстояние до преграды) и управления МР с помощью пульта управления (движением МР, поворотом камерой, и т.д.) [1]. Все это усложняет управление МР и снижает эффективность принятых решений за счет рассеивания внимания оператора, визуального распознавания образов, особенно мелких деталей, которые могут быть незаметны на общем фоне и нести в себе ключевые моменты в решении поставленных задач. Вследствие чего возникает актуальная задача разработки новых подходов

реализации СТЗ и автоматизации обработки видео потока, с выделением и распознаванием элементов окружающей среды МР. Для решения данной задачи предлагается использовать подход получения изображения и информации с датчиков, передавая их на шлем виртуальной реальности и управлять поворотом камеры в 4 степенях свободы [2].

Разработка структуры СТЗ для получения стерео 3D изображения на базе ОС Android

В ходе анализа аппаратной составляющей, необходимой для реализации и разработки СТЗ на базе стерео 3D, были выбраны следующие компоненты:

- шлем виртуальной реальности VR Box 2.0;
- беспроводная Wi-Fi камера;
- смартфон 6 дюймовым экраном на базе ОС Android 4.0.4 (API level 15, NDK 8) и выше;
- мобильная платформа.

Для программной реализации выбраны:

- OpenGL ES;
- FFmpeg;
- OpenCV.

Разработанная общая структурная схема приложения со следующими информационными потоками для ОС Android представлена на рисунке 1. Приложение состоит из двух модулей, Activity и Native Code [5]. Данное решение позволит ускорить процесс обработки потокового видео в связи с ограничениями, которые накладываются программными возможностями операционной системы.

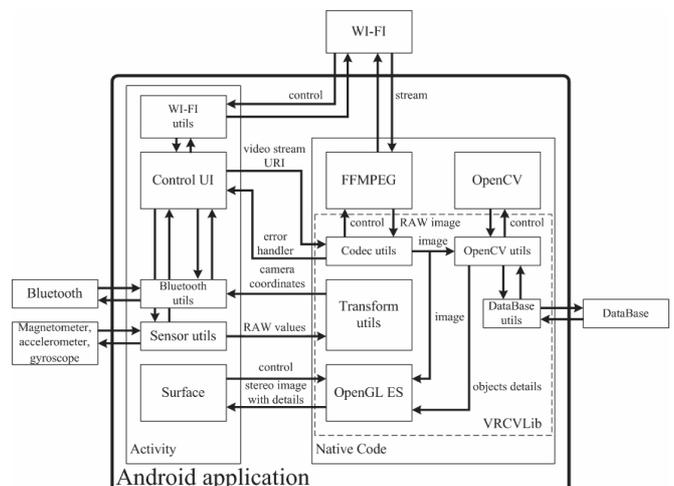


Рис. 1. Структурная схема приложения управления мобильной платформой

Модуль Activity представляет собой java код и содержит в себе модули:

- Wi-Fi utility;
- Control UI;
- Bluetooth utility;
- Sensor utility;
- Surface.

Control UI отвечает за интерфейс управления и настроек приложения, Bluetooth, Wi-Fi, IP камерой а так же настройки базой данных, датчиков (рис. 2).



Рис. 2. Окно настроек приложения

Wi-Fi utility отвечает за подключение и связь с IP камерой. При входе в приложение выбираются настройки приложения и настройки IP камеры (рис. 3). Нажав подключение к сети, откроется стандартное окно подключения к беспроводной сети Wi-Fi, в котором необходимо выбрать IP камеру или сеть, в которой она находится. После чего необходимо ввести URI адрес камеры, включая протокол передачи данных (http, rtsp, rtp, rtcp и другие), IP адрес, порт.



Рис. 3. Окно настроек IP камеры

Bluetooth utility необходим для управления Bluetooth модулем, получения списка модулей в сети, подключения к ним и передачу данных. Так же обеспечивает управление мобильной платформой.

Sensor utility используется для получения данных с датчиков, таких как компас, акселерометр и гироскоп. Данные получаются без применения фильтров и передаются для дальнейшей обработки библиотеке VRCVLib.

Surface используется для рисования сцены OpenGL ES.

Native Code включает в себя три библиотеки:

- FFMPEG – набор свободных библиотек с открытым исходным кодом, которые позволяют записывать, конвертировать и передавать цифровые аудио- и видеозаписи в различных форматах;
- OpenCV – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом;

- VRCVLib (Virtual Reality and Computer Vision Library) – разрабатываемая библиотека для объединения виртуальной реальности и компьютерного зрения. Позволяет получать видео поток (применяется FFMPEG) преобразовывать в стерео картинку и получать данные об объектах (применяется OpenCV).

VRCVLib представляет собой кроссплатформенную библиотеку, написанную на C/C++. В нее входят модули:

- Codec utility;
- OpenCV utility;
- OpenGL ES;
- DataBase utility;
- Transform utility.

Codec utility представляет собой утилиты для получения видео потока с IP камеры, преобразования в необходимый формат данных [3], и вывода преобразованных данных с помощью OpenGL ES [4] и дальнейшую обработку модулем OpenCV utility. Структура модуля изображена на рисунке 4.

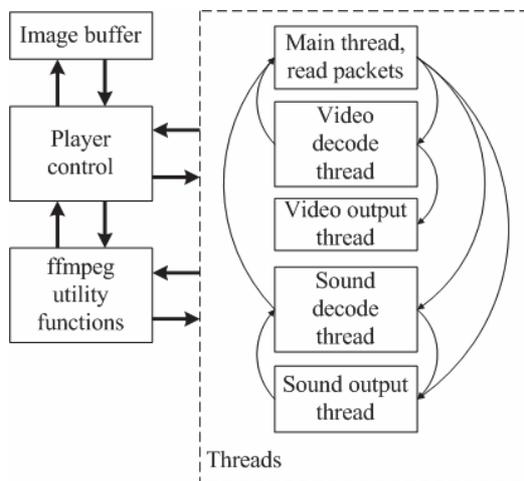


Рис. 4. Структура модуля Codec utility

В ходе запуска проигрывателя (Player control), необходимо передать ему URI адрес, указать буфер изображения (Image buffer), задать дополнительные параметры, после чего выполнить функцию воспроизведения. Пример кода:

```
player = new HPlayer();
player->setGLBuffer(glbuf);
player->openFile("http://192.168.0.68:8080/video");
player->startDecode();
player->play(HPLAYER_STATE_PLAY);
```

В случае ошибки, каждая функция возвращает HPLAYER_OK в случае нормального завершения функции и HPLAYER_ERROR в случае ошибки, при этом детальное описание ошибки выводится в Logcat (Android) или stdout (Linux, Windows) в зависимости от конфигурации откомпилированной библиотеки.

OpenCV utility перехватывает видео буфер, обрабатывает, преобразуя его в необходимый формат, после чего начинается анализ изображенных объектов [8]. Основой алгоритма является кластеризация объектов, т.е. выделение их в отдельную область, после чего выполняется идентификация по базе данных. Результат реализаций подхода идентификаций и поиска объектов представлен на рисунке 5.

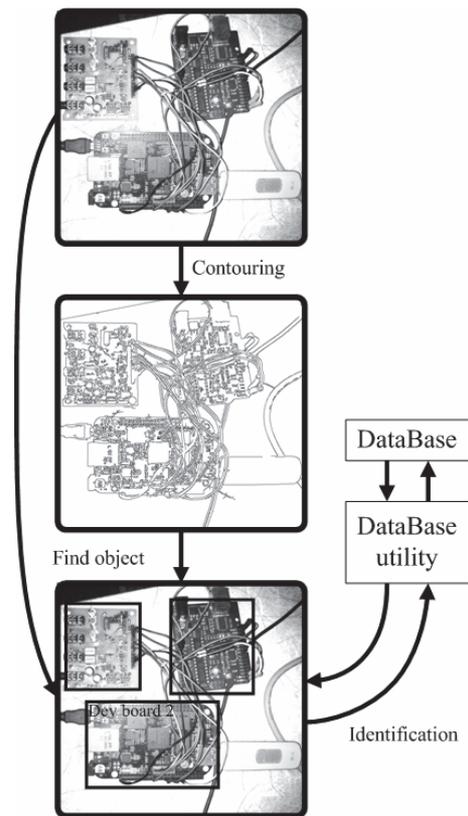


Рис. 5. Структура поиска и идентификации объектов

Функции OpenGL ES представляет собой класс рендера HGLRenderVR, который занимается рисованием стерео сцены, класс занимающийся выводом сцены HGLSurface. Пример кода:

```
glsurface = new HGLSurface(WND);
glrendervr = new HGLRenderVR();
```

```
glsurface->setRender(glrendervr);
```

WND представляет собой указатель на окно, т.е. куда будет выводиться сцена, в системе Android это ANativeWindow, в системе Linux это XWindow (если используется X11), в системе Windows это HWND. Для ускорения вывода на некоторых устройствах возможен вывод изображения через расширение EGL_KHR_image, это позволяет выводить изображения напрямую, минуя копирования памяти, т.е. функции glTexImage2D.

В модуле так же есть функции рисования примитивов (линии, прямоугольники, текст, куб и другие), при этом они рисуются в обеих сторонах стерео изображения, все объекты могут рисоваться как в перспективе, так и на переднем плане, в виде элементов управления, пример отображение примитивов приведен на рисунке 6.

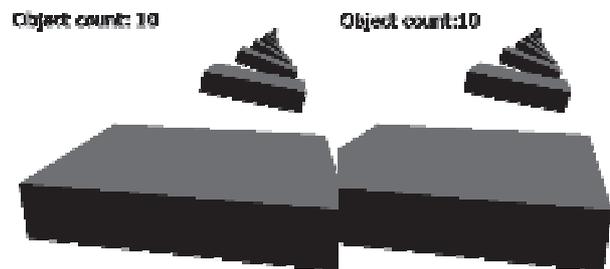


Рис. 6. Демонстрация отображения примитивов в стерео изображений для шлема VR Box 2.0

Рисование стерео сцены происходит по методу асимметричной перспективы видимости [6][7] (рис. 7). В этом методе используется матрица камеры frustum.

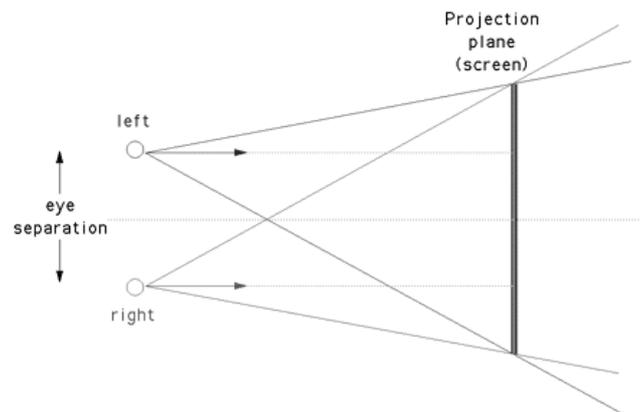


Рис. 7. Метод асимметричной перспективы видимости

DataBase utility представляет собой функции обмена данными с базой данных. В базе данных хранится информация об объектах и их свойствах.

Transform utility преобразует данные с датчиков в углы поворота камеры на мобильной платформе, при этом в начале преобразования происходит калибровка и фильтрация данных, так как мобильные устройства

содержат в себе не точные датчики, имеющие зашумленные значения. Для более точного определения поворота используются все возможные датчики, которые находятся в смартфоне. Минимальным вариантом является акселерометр. Матрица калибровки представляет собой матрицу трансформации и смещения. Общий вид матрицы калибровки датчиков представлен на рисунке 8.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \times \begin{bmatrix} X_{nc} \\ Y_{nc} \\ Z_{nc} \end{bmatrix} - \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix}$$

Рис. 8. Матрица калибровки датчиков

Матрица трансформации и смещения задается в настройках приложения, для разных датчиков может принимать разные значения, вследствие чего для разных аппаратных устройств их необходимо получать заранее [9,10], в противном случае, управление будет невозможно, либо приносить неудобства.

Для ускорения фильтрации используется комбинаторный фильтр, который представляет собой комбинацию фильтра низкой и высокой частоты. Если в устройстве есть гироскоп и акселерометр можно применить фильтр Калмана [11], суть которого заключается в предсказании значений. Применяя комбинацию акселерометра и гироскопа, а так же данный фильтр, можно достичь большей точности в управлении и отзывчивости мобильной платформы.

Выводы

Предложенный подход представления информации об окружающем пространстве мобильного робота с использованием эффекта 3D погружения, даст возможность оператору получать объемные данные с видео камеры и сразу проводить процесс идентификации и распознавания объектов. Используя возможности VR Box 2.0 и смартфона с разработанным программным обеспечением, позволит объединить процесс управления поворотом камеры, а также выводить на экран все

необходимые данные с датчиков с минимальными задержками, что улучшает управление мобильным роботом, делая его более удобным для оператора.

Следующим этапом исследований станет создание базы данных с использованием технологий Big Data, данное решение позволит увеличить скорость и обработку большого массива баз знаний с шаблонами изображения для распознавания.

СПИСОК ЛИТЕРАТУРЫ:

1. *Acoustic Model Using for Mobile Robot Control / I. Nevlyudov, O. Tsimbal, S. Miliutina, V. Sharkovskiy // Radiotechnics, 164/2011, Kharkiv, KNURE, 2011. - pp. 120-125.*
2. *Mobile platforms control system development on the base of virtual dive method to the environment / prof. Nevlyudov I., prof. Yevsieiev V., assoc. prof. Miliutina S., Salieva V.*
3. *An ffmpeg and SDL Tutorial [Электронный ресурс]// URL: <http://dranger.com/ffmpeg> (дата обращения 25.12.2016).*
4. *Displaying Graphics with OpenGL ES [Электронный ресурс]// URL: <https://developer.android.com/training/graphics/opengl/index.html> (дата обращения 25.12.2016).*
5. *Getting Started with the NDK [Электронный ресурс]// URL: <https://developer.android.com/ndk/guides/index.html> (дата обращения 25.12.2016).*
6. *Implementing Stereoscopic 3D in Your Applications [Электронный ресурс]// URL: http://www.nvidia.com/content/GTC-2010/pdfs/2010_GTC2010.pdf (дата обращения 25.12.2016).*
7. *Calculating Stereo Pairs [Электронный ресурс]// URL: <http://paulbourke.net/stereographics/stereorender/> (дата обращения 25.12.2016).*
8. *OpenCV Samples [Электронный ресурс]//URL: <https://github.com/opencv/opencv/tree/master/samples> (дата обращения 25.12.2016).*
9. *Scalar calibration of vector magnetometers [Электронный ресурс]// URL: <https://pdfs.semanticscholar.org/9846/06d95a3ab5deae5efbe5be17501e3805f7d.pdf> (дата обращения 25.12.2016).*
10. *Parameters and calibration of a low-g 3-axis accelerometer [Электронный ресурс]// URL: http://www.st.com/content/ccc/resource/technical/document/application_note/a0/f0/a0/62/3b/69/47/66/DM00119044.pdf/files/DM00119044.pdf/jcr:content/translations/en.DM00119044.pdf (дата обращения 25.12.2016).*
11. *Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation [Электронный ресурс]// URL: <https://www.cl.cam.ac.uk/~rmf25/papers/Understanding%20the%20Basis%20of%20the%20Kalman%20Filter.pdf> (дата обращения 25.12.2016).*