

## ВИЗУАЛЬНАЯ ЛОКАЛИЗАЦИЯ НА ОСНОВЕ ГЕОМЕТРИЧЕСКИХ ПРИЗНАКОВ И 3D МОДЕЛИ ОКРУЖАЮЩЕЙ СРЕДЫ В СИСТЕМЕ ROS

А.К. Буйвал<sup>1</sup>, М.А. Гавриленков<sup>2</sup>

<sup>1</sup>К.т.н., Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Брянский государственный технический университет» (БГТУ), доцент, 241035, г. Брянск, бул. 50-летия Октября, 7, тел.: +7(4832)56-09-84, alexbuyval@gmail.com

<sup>2</sup>БГТУ, аспирант, 241035, г. Брянск, бул. 50-летия Октября, 7, тел.: +7(4832)56-09-84, gavrilencov@umlab.ru

### Аннотация

Описаны основные компоненты системы визуальной локализации малого БПЛА внутри помещения для системы ROS, основанной на сопоставлении граней изображения с видеокамеры и изображения, полученного на основе известной 3D модели окружающей среды (помещения).

### Ключевые слова

Визуальная локализация, фильтр частиц, ROS, Gazebo.

## VISUAL LOCALIZATION BASED ON GEOMETRICAL FEATURES AND 3D ENVIRONMENT MODELS IN ROS

A.K. Buyval<sup>1</sup>, M.A. Gavrilencov<sup>2</sup>

<sup>1</sup>PhD in Technical Sciences, State Educational Institution of Higher Professional Education «Bryansk State Technical University» (BSTU), Assistant Professor, 7, Bulvar 50-letiya Oktyabrya, Bryansk, 241035, Russia, tel.: +7(4832)56-09-84, alexbuyval@gmail.com

<sup>2</sup>BSTU, Postgraduate Student, 7, Bulvar 50-letiya Oktyabrya, Bryansk, 241035, Russia, tel.: +7(4832)56-09-84, gavrilencov@umlab.ru

### Abstract

Basic components of system for small UAV's visual localization indoors for ROS, based on comparison of edges of video camera images and images, obtained from known 3D model of environment (indoor space), are described.

### Key words

Visual localization, particle filter, ROS, Gazebo.

### Введение

Одной из основных проблем мобильных роботов остается локализация в пространстве. Для ее решения беспилотные летательные роботы используют множество датчиков. Для локализации внутри помещений, к сожалению, невозможно использовать GPS/ГЛОНАСС, очень часто используются инерциальные датчики и лазерные дальномеры, системы, основанные на визуальной информации. Ниже описана подобная система для локализации БПЛА. В ней использовано сравнение изображения, полученного с камеры, со смоделированным изображением, полученным на основе 3D модели помещения.

Предлагается для локализации в помещении использовать ROS и среду моделирования GAZEBO. Разработанная система оперирует с данными от камеры и данными из модели, из чего вытекает главное требование: необходимость в модели помещения. В прикладном использовании, зачастую, проще создать такую модель, опираясь на чертежи, чем использовать визуальные маркеры внутри помещения.

### 1 Обоснование выбора ROS для проектирования подсистемы локализации

Математический аппарат для локализации в пространстве совершенствуется с каждым годом. Одним из главных критериев при этом является масштабируемость. Для упрощения задачи система строится

на базе Robot Operating System (ROS) – набора библиотек и средств разработки роботизированных систем, который приобретает всё большую популярность. Основной концепцией ROS является создание программного обеспечения, которое бы работало с различными роботами при наименьшем изменении в программном коде. Подобная концепция позволяет разграничивать функциональность системы.

Проектирование программ под ROS подразумевает написание «узлов», которые и выполняют программный код. Узлы ROS общаются между собой посредством «topic» – тем, содержащих сообщения. При необходимости отправки данных из одного узла в другой, узел-отправитель создает «topic» с определенным именем, после чего он публикуется в общую очередь сообщений. Узлы-слушатели используют специальный механизм для прослушивания этой очереди и в случае нахождения сообщения, подписанного необходимым топиком, передают сообщение модулю обработки сообщения узла.

В описываемой системе используется несколько узлов, основными из них являются:

- узел интерфейса управления;
- узел визуальной локализации;
- узел моделирования окружения.

**2 Теоретическая основа разработанной системы локализации**

Ключевым моментом в предлагаемом методе визуальной локализации является сравнение и численная оценка схожести 2-х изображений: полученного с камеры и изображения, смоделированного на основе модели помещения. Предполагается, что алгоритм строит ряд гипотез о положении робота в пространстве, затем на основе этих гипотез он моделирует изображения, каждое из которых соответствует изображению, которое должен получить робот, если он находится в этой предполагаемой точке. Таким образом, на каждой итерации алгоритма сравнивается одно изображение с камеры со множеством смоделированных изображений и определяется степень их схожести.

В работах [4] и [1] приведены несколько различных методов оценки схожести изображений и обоснован выбор метода «ближайших граней», как наиболее эффективного для данной задачи.

Для обработки данных о схожести изображений, а также данных с других доступных датчиков предлагается использовать алгоритм локализации, основанный на множестве частиц (гипотез о местоположении робота) – фильтр частиц (particle filter). Данный метод хорошо зарекомендовал себя в подобных задачах. Его преимуществом является то, что он позволяет использовать множество гипотез, а также нелинейные модели как самой системы, так и нелинейные модели показаний датчиков.

Более подробно разработанная система описана в [4].

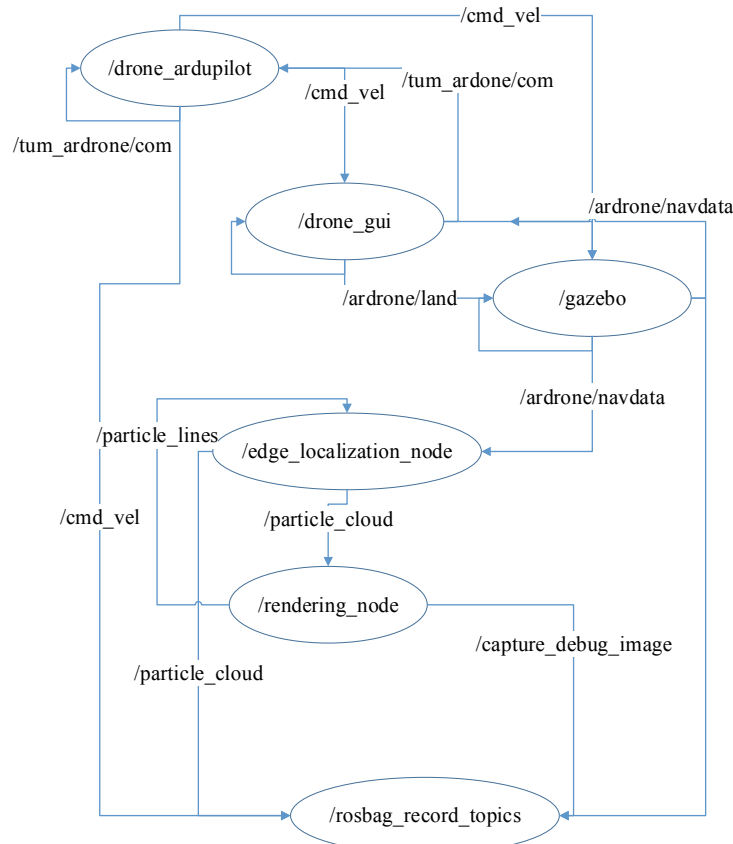
**3 Структура разработанной системы**

Система использует ряд заимствованных пакетов, о которых будет сказано ниже, а также включает в себя ряд ключевых узлов (рис. 1):

1. EdgeLocalizationNode – узел, отвечающий за локализацию робота в пространстве. Этот узел получает сообщения от узла «Gazebo», который используется для моделирования движения робота. Также узел подписан на тему «/particle\_lines», сообщения которого содержат список граней каждой частицы. При получении этого сообщения происходит вычисление весов частиц.

2. RenderingNode – узел, отвечающий за моделирование окружения. Подписан на сообщения от узла «EdgeLocalizationNode». При получении списка частиц в узле производится рендеринг позиции каждой частицы, из результата рендеринга извлекаются линии граней, которые затем публикуются в тему «/particle\_lines».

3. Gazebo - узел, использующийся для симуляции движения робота. Т.к. во время разработки системы нецелесообразно использовать реального робота, используется среда моделирования. В рабочей системе этот узел будет заменен узлом реального робота, имеющим идентичный интерфейс. Узел генерирует сообщения с данными одометрии, подписчиков «/ardrone/navdata», а также сообщения, содержащие кадры с камеры робота, и отправляет их узлу «EdgeLocalizationNode». Этот узел подписан на темы, содержащие сообщения с управляющими командами о взлете, приземлении и т.п.



**Рисунок 1 - Структура узлов разработанной системы**

Как видно из схемы, все узлы общаются между собой собственными сообщениями:

1. /ardrone/navdata – сообщение данной темы хранит координаты грани в пикселях; структура: x1, y1, x2, y2 – float;

2. /particle\_lines – сообщение формируется в узле моделирования после определения всех граней на изображении; в поле ID хранится идентификатор частицы, в lines – список граней, обнаруженных для данной частицы; структура: ID – int, lines - LineMsg[];

3. /particles\_cloud – сообщение содержит список элементов ParticleLines, сопоставляющих с каждой частицей список граней; используется для передачи информации из узла моделирования в узел локализации; структура: particles - particleLines[].

Ключевым компонентом узла EdgeLocalization является фильтр частиц. Данный фильтр реализован на базе библиотеки Orococos BFL. Orococos Bayesian Filtering Library является независимой библиотекой, обеспечивающей рекурсивную обработку информации опираясь на правила Байеса. В качестве базового класса выступает BootstrapFilter, оперирующий набором объектов класса Particle и набором прямых, как аргументом функции наблюдения.

Как указано в [1] основными компонентами фильтра частиц являются измерительная модель и модель системы. Здесь измерительная модель реализована на основе класса NonLinearMeasurementPdf, который принимает как аргумент прямые, полученные из изображения с камеры, на основе которых вычисляется вес частицы. Модель системы реализована на основе класса NonLinearSystemPdf, основным аргументом которого являются скорости движения объектов.

Для запуска системы написан launch файл, описывающий все запускаемые узлы. В файле запускаются узлы rendering\_node и edge\_localization\_node. Узел rendering\_node имеет в своем описании свойства углов поворота камеры, а также свойства алгоритма определения граней изображений. Узел edge\_localization\_node имеет в своем описании свойства положения квадрокоптера в среде моделирования и параметры алгоритма определения граней.

Одним из ключевых моментов разработанной системы, является извлечение линий граней из списка частиц. За эту процедуру отвечает узел RenderingNode. Данный узел моделирует отображение видеопотока с камеры квадрокоптера (рис. 2). Основным классом узла является одноименный класс RenderingNode, который наследуется от класса OgreBase, свободного графического движка Ogre [6]. Диаграмма основных классов разработанной подсистемы представлена на рис. 3.

Для получения изображения в узле RenderingNode используется функционал библиотеки OpenCV.

Узел RenderingNode является слушателем узла EdgeLocalizationNode, который отправляет заданному узлу информацию о частицах. Сообщение имеет тип /particle\_cloud.

Результатом обработки и сопоставления частиц и картинки с камеры, является набор линий граней, которые возвращаются узлу EdgeLocalizationNode посредством сообщения типа /particle\_lines.

Для получения линий граней используется вспомогательный класс LineExtractor. Данный класс имеет в своем описании статичный метод GetLineByHough, который, в свою очередь, принимает на вход изображение, которое описывается классом Mat библиотеки OpenCV. Перед отправкой изображения для обработки методом GetLineByHough, изображение обращается в серый цвет. Также, входным параметром является список настроек для фильтра, такие как минимальная и максимальная длина линии и другие.

При работе метода GetLineByHough используется метод Canny, библиотеки OpenCV, который возвращает список обнаруженных граней изображения. Найденные грани становятся входным параметром для функции HoughLinesP, которая описывает преобразования Хафа и возвращает список найденных линий.



Рисунок 2 - Окно рендеринга узла rendering\_node

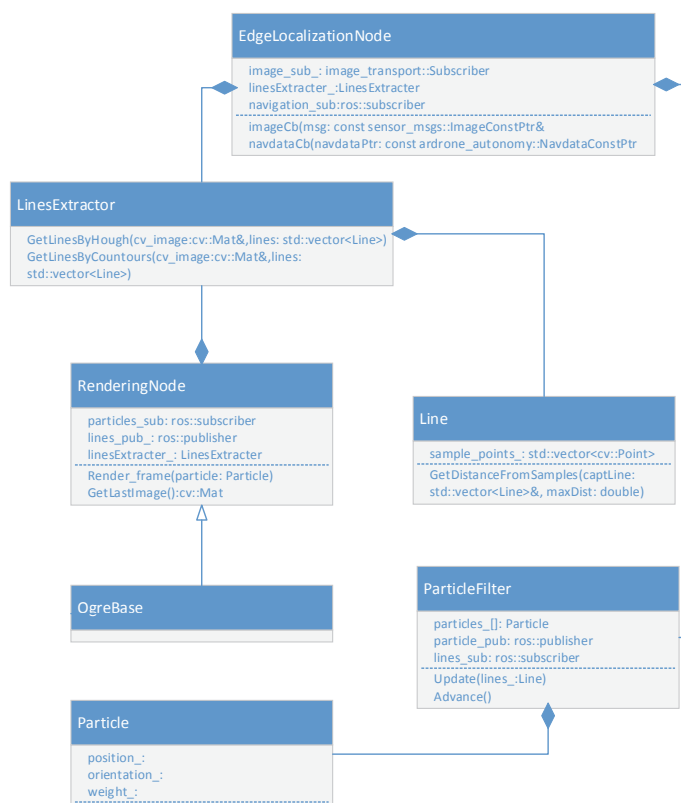


Рисунок 3 - Диаграмма классов

#### 4 Дальнейшее развитие системы

Планируется испытать систему на реальном квадрокоптере. При этом модернизировать обработку изображения для улучшенного нахождения граней

при изменяющихся внешних условиях, а также оптимизировать работу системы на графических сопроцессорах для ее ускорения, в частности узла рендеринга.

#### Литература

1. Nuske, S., Roberts, J., & Wyeth, G. (2008, May). Outdoor visual localization in industrial building environments. // IEEE International Conference on Robotics and Automation, Pasadena, CA. 2008. – Pp. 544-550.
2. Nuske, S., Roberts, J., & Wyeth, G. Robust outdoor visual localization using a three-dimensional-edge map // Journal of Field Robotics. – 2009. - №26(9). – Pp. 728-756.
3. Klein, G., & Murray, D. Full-3D edge tracking with a particle filter // In British Machine Vision Conference. 2006. Edinburgh, UK.
4. Буйвал А.К. Локализация беспилотного летательного аппарата внутри помещений на основе визуальных геометрических маркеров и известной 3D модели окружающей среды. // Робототехника и техническая кибернетика. - 2015. - №3(8). - С. 71-75.
5. Fernandez, E., Crespo S., Mahtani A., Martinez A., Learning ROS for Robotics Programming Second Edition. // Packt Publishing. - 2015.
6. Kerger F. OGRE 3D 1.7 Beginner's Guide // Packt Publishing. – 2010.

#### References

1. Nuske, S., Roberts, J. and Wyeth, G. (2008). Visual localisation in outdoor industrial building environments. *2008 IEEE International Conference on Robotics and Automation*.
2. Nuske, S., Roberts, J. and Wyeth, G. (2009). Robust outdoor visual localization using a three-dimensional-edge map. *Journal of Field Robotics*, 26(9), pp.728-756.
3. Klein, G. and Murray, D. (2006). Full-3D edge tracking with a particle filter. In: *British Machine Vision Conference*.
4. Buyval, A. (2015). Lokalizatsiya bespilotnogo letatel'nogo apparata vnutri pomeshcheniy na osnove vizual'nykh geometricheskikh markerov i izvestnoy 3D modeli okruzhayushchey sredy [Indoors UAV localization based on visual geometric features and known environment 3D model]. *Robotics and Technical Cybernetics*, 3(8), pp.71-75.
5. Fernández, E. (2015). *Learning ROS for robotics programming*. 2nd ed. Birmingham, UK: Packt Publishing.
6. Kerger, F. (2010). *OGRE 3D 1.7 beginner's guide*. Birmingham: Packt Pub.