Научная статья

Original article

УДК 681.5

ПРИМЕНЕНИЕ АЛГОРИТМЫ РАСПОЗНАВАНИЯ ЖЕСТОВ ДЛЯ УПРАВЛЕНИЯ МОБИЛЬНОГО РОБОТА

USING GESTURE RECOGNITION TO CONTROL A MOBILE ROBOT

Ван Вэньюэ, студент факультета специального машиностроения, Московский государственный технический университет имени Н. Э. Баумана, г. Москва. 18761875608@163.com

Сун Бисянь, студент факультета специального машиностроения, Московский государственный технический университет имени Н. Э. Баумана, г. Москва. 13390773394@163.com

Wang Wenyue, student of the Faculty of Special Engineering, Bauman Moscow State Technical University, Moscow. 18761875608@163.com

Song Bixian, student of the Faculty of Special Engineering, Bauman Moscow State Technical University, Moscow. 13390773394@163.com

Аннотация. В этой работе объединяются распознавание жестов и управление роботом. Распознавание жестов было выполнялся на основе MediaPipe Hands. Описываются основные структуры MediaPipe Hands. Были определены пять команд жестов для управления движением мобильного робота (Turtlebot3) с дифференциальным приводом. Точности распознавания жестов оценивался в работе. На конце эксперименты для управления робота с помощью командами жестов провелся на платформе ROS в системе Ubuntu 18.04.

Abstract. This work combines gesture recognition and robot control. Gesture recognition has been performed based on MediaPipe Hands. The basic structures of MediaPipe Hands are described. Five gesture commands were defined to control the motion of a mobile robot (Turtlebot3) with a differential drive. Gesture recognition accuracies were evaluated in the paper. At the end, experiments to control the robot with gesture commands were conducted on the ROS platform on the Ubuntu 18.04 system.

Ключевые слова: распознавание жестов, оценка точности, MediaPipe Hands, система ROS

Keywords: gesture recognition, accuracy assessment, MediaPipe Hands, ROS system

Введение

За последние несколько десятилетий графический пользовательский интерфейс (GUI) быстро продвинулся в качестве интерфейса связи между человеком и компьютером, но использование графического интерфейса по-прежнему требует ручного управления, и в некоторых случаях это будет неудобно. Задача ручного управления робототехнической системой еще полностью не решена. Основной проблемой является отсутствие удобного и простого в обращение, особенно для неподготовленного оператора, пользовательского интерфейса. Даже для проведения простых манипуляций с физическими предметами может потребоваться несколько десятков комбинаций команд. Создание же интерфейса, понятного и эргономичного для человека, в таком ключе становиться почти невыполнимой задачей.

Одним из решений данной проблемы может стать управление устройством при помощи команда жестов, подаваемых человеком. Хотя каналы связи между людьми и роботами по-прежнему ограничены, распознавание жестов эффективно применяется в качестве интерфейса между людьми и компьютерами в течение длительного периода времени.

Основной целью данной работы является использование MediaPipe, предоставляемого компанией Google, для простого распознавания жестов. Затем используйте свои собственные определенные жесты для управления движением робота Turtlebot3 в системе Ubuntu 18.04.

1. MediaPipe Hands

MediaPipe – это фреймворк для создания кросс-платформенных (Android, iOS, web, оконечные устройства) мультимодальных (видео, аудио, временные последовательности данных) конвейеров (pipelines) машинного обучения (Machine Learning), который состоит из алгоритмов машинного обучения, классического компьютерного зрения и обработки медиа информации (то есть декодирования видео). MediaPipe включает в себя пакеты, написанные на python и других языках программирования), для решения следующих задач [1]:

• обнаружение объектов;

• обнаружение лиц;

• отслеживание рук (как одной, так и нескольких);

• оценка положения тела;

• сегментации волос.

MediaPipe Hands - это высококачественное решение для отслеживания рук и пальцев. Он использует машинное обучение (ML) для определения 21 ориентира руки всего из одного кадра.

Здесь 3 модели используются для выполнения распознавания жестов в MediaPipe hand [2].

- Модель обнаружения ладони. Эта модель работает со всем изображением и возвращает ограничивающую рамку с направленной областей.
- Модель ориентиров руки, которая работает с обрезанной областью изображения, определенной детектором ладони, и возвращает высокоточные 3D ключевые точки руки.
- Распознаватель жестов, который классифицирует ранее вычисленную конфигурацию ключевых точек в дискретный набор жестов.

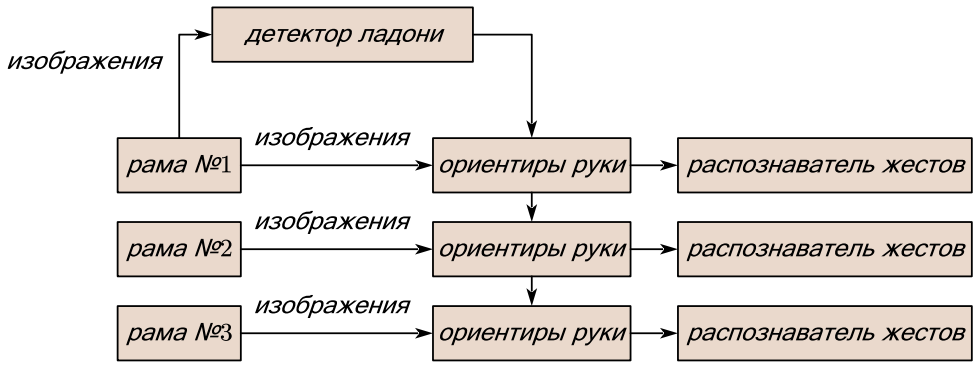


Рис.1 Схема распознавания жестов в MediaPipe hand

После обнаружения ладони на всем изображении наша последующая модель ориентиров руки выполняет точную локализацию ключевых точек 21 3D координат костяшек кисти внутри обнаруженных областей руки с помощью регрессии [3], полученные результаты показаны на рисунке ниже.

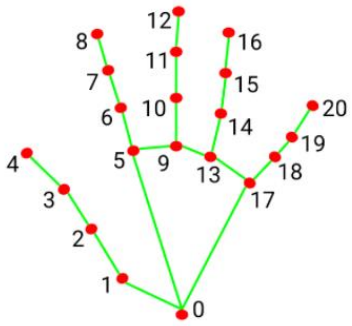


Рис.2 ориентиры руки

Здесь на рис.2 :

Точка 0 - ориентир на запястье;

Точки 1,2,3,4 - ориентиры на большом пальце;

Точки 5,6,7,8 - ориентиры на указательном пальце;

Точки 9,10,11,12 - ориентиры на среднем пальце;

Точки 13,14,15,16 - ориентиры на безымянном пальце;

Точки 17,18,19,20 - ориентиры на мизинце.

На основе ориентиров руки, мы используем простой алгоритм для определения жестов в блоке распознаватели жестов. С начало мы можем получить координаты этих 21 ориентира на изображении и отобразить координаты текущего местоположения на изображении.

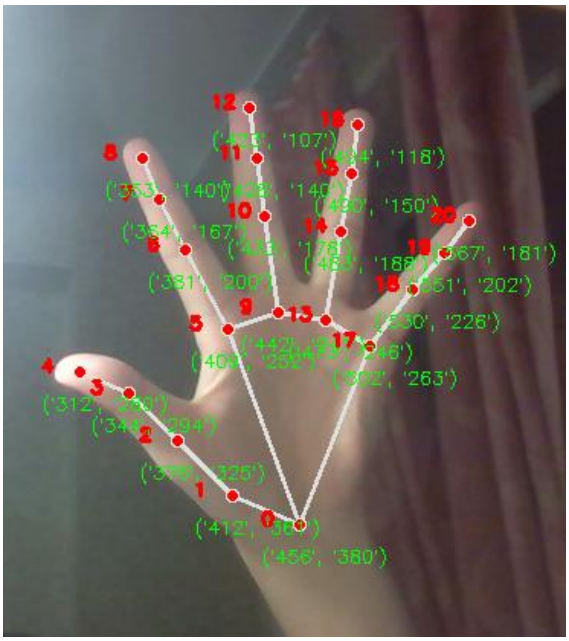


Рис.3 координаты ориентиров

После того, как мы получим координаты ориентира, мы сможем получить два двумерных вектора на наших пальцах. Определите, согнуты ли пальцы, рассчитав угол между векторами, чтобы можно было выполнить простое распознавание жестов. Например, Мы можем использовать вектор $\overrightarrow{0\ 2}$ и вектор $\overrightarrow{3\ 4}$, чтобы определить, согнут ли большой палец. Векторы, которые мы используем для определения того, согнут ли каждый палец, показан в следующей таблице.

Таблица.1

Пальцы	Векторы
большой палец	$\overrightarrow{0\ 2}$ и $\overrightarrow{3\ 4}$
указательный палец	$\overrightarrow{0\ 6}$ и $\overrightarrow{7\ 8}$
средний палец	$\overrightarrow{0\ 10}$ и $\overrightarrow{11\ 12}$
Безымянный палец	$\overrightarrow{0\ 14}$ и $\overrightarrow{15\ 16}$
мизинец	$\overrightarrow{0\ 18}$ и $\overrightarrow{19\ 20}$

На основе уравнения (1) можно вычислить угол θ между двумя векторами.

$$\theta = \arccos\left(\frac{\vec{A} * \vec{B}}{|A| * |B|}\right) \quad (1)$$

Сравните угол изгиба пальца с пороговым значением, чтобы определить, согнут ли палец. Пороговые значения, используемые здесь, основаны на опыте. Мы определили следующие пять жестов, как показано на рис.4.



Рис.4 команды жестов

Чтобы проверить точность распознавания жестов, мы выполнили распознавание жестов на статичных изображениях. Мы пригласили 5 студентов выполнять различные жесты. При записи жестов поместите одну ладонь в центр экрана, что составляет примерно 1/4-1/3 экрана, чтобы убедиться, что ладони одинаковы по размеру и узнаваемы. Основываясь на 5 разработанных жестах, каждый студент выполнил каждый жест 3 раза, чтобы получить 75 изображений жестов. Затем набор данных расширяется путем переворачивания, вращения, панорамирования, размытия или осветления, в результате чего получается 1432 изображений жестов.



Рис.5 набор данных

Как показано на рисунке 12, мы поочередно вводим каждое изображение жеста, выводим результаты распознавания жестов, наносим на изображение

результаты и ключевые точки руки, и, наконец, подсчитываем матрицу ошибок, как показано в таблице 2.

Таблица.2

Обнаруженные жесты / Реальные жесты	вперед	назад	направо	налево	стоп
вперед	300	1	0	0	1
назад	0	263	0	3	1
направо	0	5	279	0	0
налево	0	0	11	282	2
стоп	0	0	1	0	283

Показатели precision(точность), recall(полнота) и F_1 (среднее гармоническое precision и recall) используются для оценки производительности алгоритма распознавания жестов, и получается таблица 3.

$$\text{precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (3)$$

В уравнениях (2) и (3):

- TP (истинно-положительные) — те категории, которые мы ожидали увидеть и получили на выходе.
- FP (ложно-положительные) — категории, которых быть на выходе не должно и анализатор их ошибочно вернул на выходе.
- FN (ложно-отрицательные) — категории, которые мы ожидали увидеть, но анализатор их не определил [4].

Таблица.3

	точность	полнота	F_1
вперед	94.94%	100.00%	97.40%
назад	93.93%	87.67%	90.69%

направо	92.38%	93.00%	92.69%
налево	90.38%	94.00%	92.16%
стоп	93.09%	94.33%	93.71%

Можно сделать вывод, что уровень точности результатов распознавания жестов очень высок, что может соответствовать нашим требованиям к использованию

2. Эксперименты

Эксперимент проводился в системе Ubuntu 18.04 и поставщик для ROS – Melodic. Содержание эксперимента состоит в том, чтобы использовать пять определенных нами жестов для управления движением робота Turtlebot3.



Рис.6. Turtlebot3

Топики и узлы, которые мы создали для эксперимента в ROS, показаны на рис.7

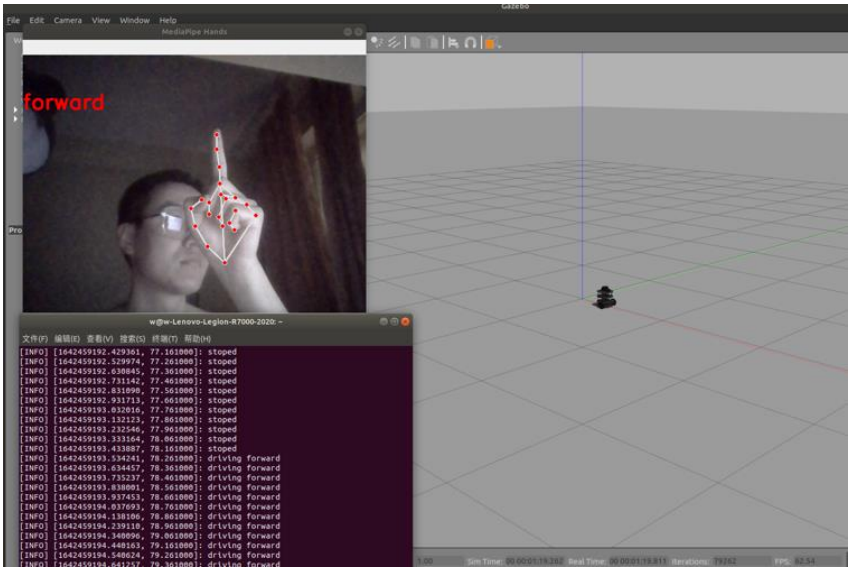


Рис.7. Топики и узлы в ROS

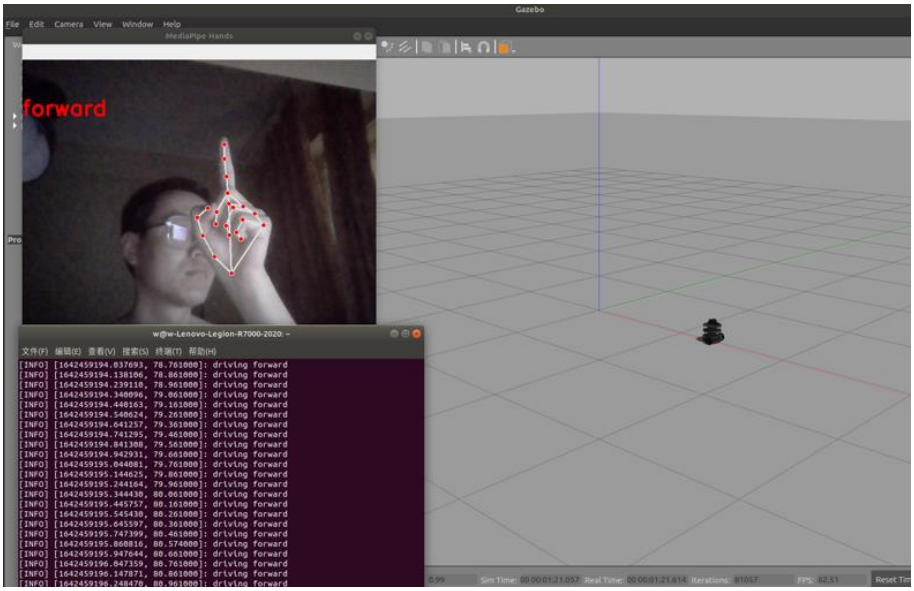
Узел «recognition» - издатель. Он содержит Алгоритм распознавания жестов, основанный на MediaPipe. Здесь мы определяем пять жестов путем обнаружения изгибания пальцев и публикуем распознанные жесты в теме «gesture».

Узел «command» получает топик «gesture», опубликованную узлом «recognition». В этом узле содержит команды управления, соответствующие пяти жестам, а затем команды публикуются в топике «cmd_vel». Когда модель робота получает команду в топике «cmd_vel», она выполняет движение.

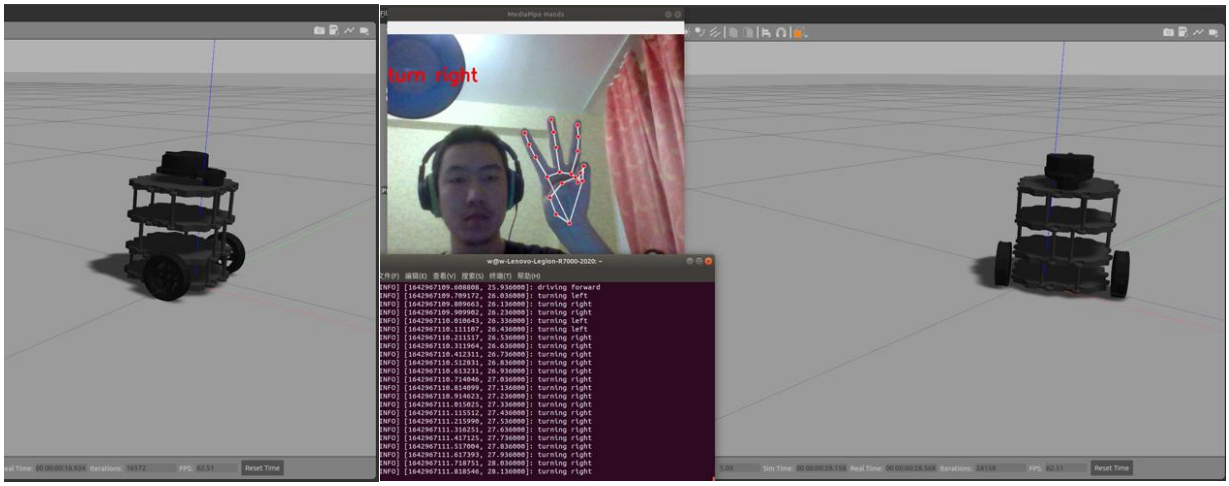
Например, мы использовали команду "вперед" для управления движением робота и наблюдали, точно ли распознаются жесты и точно ли передаются команды роботу. чтобы двигаться вперед. результат выполнения команд «вперед» и «направо» выглядит следующим образом.



(a)



(b)



(c)

Рис.8 команды «вперед» и «направо»

Как вы можете видеть, наш жест был точно распознан, и робот прекрасно выполнил команду, соответствующую жесту.

3. Заключение

В этой статье мы выполнили простое распознавание жестов на основе MediaPipe. Пять различных жестов были определены путем определения того, были ли пальцы согнуты, а затем жесты использовались в системе Ubuntu 18.04 для завершения управления роботом Turtlebot3. Результаты показывают, что метод управления, основанный на распознавании жестов, может точно передавать наши управляющие команды роботу.

Литература

1. Управляемый жестами видеоплеер на Raspberry Pi и MediaPipe [Электронный ресурс]. –URL: https://microkontroller.ru/raspberry-pi-projects/upravlyaemyj-zhestami-videopleer-na-raspberry-pi-i-mediapipe/? (Дата обращения 19.05.2022).
2. Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, Matthias Grundmann. (2020) MediaPipe Hands: On-device Real-time Hand Tracking.
3. MediaPipe Hands [Электронный ресурс]. –URL: https://google.github.io/mediapipe/solutions/hands#max_num_hands (Дата обращения 19.05.2022).

4. Оценка качества в задачах классификации и регрессии [Электронный ресурс]. –URL: https://neerc.ifmo.ru/wiki/index.php?title=Оценка_качества_в_задачах_классификации_и_регрессии (Дата обращения 20.05.2022).

References

1. Gesture-controlled video player on the Raspberry Pi and MediaPipe [Electronic resource]. –URL: https://microkontroller.ru/raspberry-pi-projects/upravlyaemyj-zhestami-videopleer-na-raspberry-pi-i-mediapipe/? (Date of access 19.05.2022).
2. Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, Matthias Grundmann. (2020) MediaPipe Hands: On-device Real-time Hand Tracking.
3. MediaPipe Hands [Electronic resource]. –URL: https://google.github.io/mediapipe/solutions/hands#max_num_hands (Date of access 19.05.2022).
4. Quality assessment in classification and regression tasks [Electronic resource]. –URL: https://neerc.ifmo.ru/wiki/index.php?title=Оценка_качества_в_задачах_классификации_и_регрессии (Date of access 20.05.2022).

© Ван Вэньюэ, Сун Бисянь, 2022 Научно-образовательный журнал для студентов и преподавателей «StudNet» №5/2022.

Для цитирования: Ван Вэньюэ, Сун Бисянь ПРИМЕНЕНИЕ АЛОРИТМЫ РАСПОЗНАВАНИЯ ЖЕСТОВ ДЛЯ УПРАВЛЕНИЯ МОБИЛЬНОГО РОБОТА// Научно-образовательный журнал для студентов и преподавателей «StudNet» №5/2022.