

УДК 004.896

Разработка алгоритма выделения маршрута для мобильного робота

Гемуев Ш.Ш., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Специальная робототехника и мехатроника»*

Научный руководитель: Михайлов Б.Б., к.т.н., доцент

*Россия, 105037, г. Москва,
НУЦ «Робототехника» МГТУ им. Н.Э. Баумана
robot@bmstu.ru*

Введение

На современном автоматизированном складе объекты работы (товары, грузы и т.п.) располагаются на универсальных паллетах. Транспортировка паллет осуществляется при помощи автоматизированных транспортных роботов. Очевидно, что такие роботы должны иметь систему, позволяющую осуществлять движение по заданным маршрутам.

Какие варианты реализации такой системы возможны? Возможно передвижение роботов по рельсам, прокладывание металлических трасс, для ориентации по магнитному полю, либо использование контрастной линии для ориентации посредством телекамеры.

Рельсы – решение хорошее, надёжное, но не универсальное и при изменении конфигурации маршрута требуют существенного изменения системы. Способ с использованием металлических полос подвержен электромагнитным помехам и может осуществлять сканирование только на небольших расстояниях, что снижает скорость движения.

Использование технического зрения представляется наиболее гибким решением данной задачи. Оно позволяет добиться высокой скорости передвижения робота благодаря тому, что возможно сканирование маршрута перед роботом. Кроме того, возможно быстрое изменение маршрута передвижения роботов путём изменения конфигурации линии.

Назначение системы

Система технического зрения, использующая данный алгоритм, рассчитана на мобильных роботов, движущихся по нанесённым на поверхность линиям. Область применения таких роботов довольно широка: на складах разного типа; на производствах,

где доставка необходимых материалов и грузов требует преодоления больших расстояний; для перевозки персонала в пределах предприятия и в других областях.

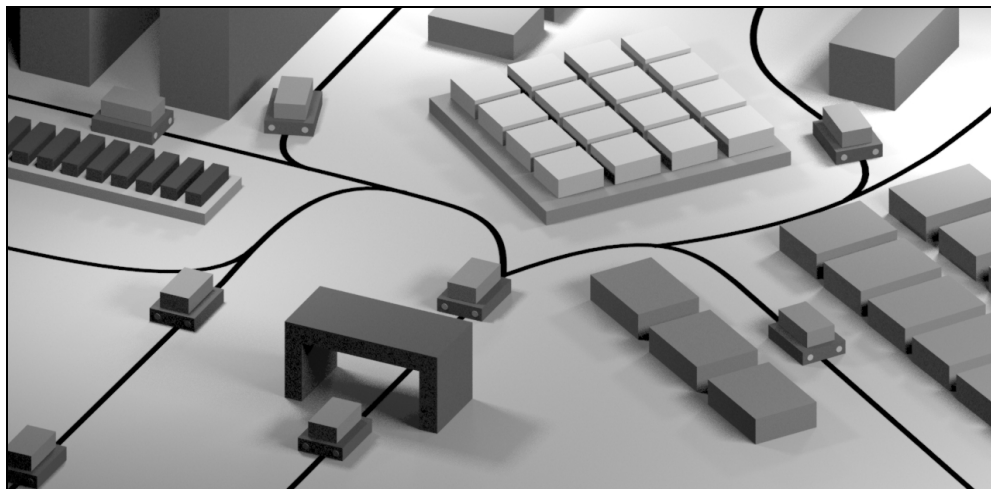


Рис. 1. Пример автоматизированного склада

Для использования данной системы на роботе должна быть установлена телекамера, считывающая изображения и компьютер с быстродействием, достаточным для обработки в масштабе реального времени. Результатом работы программы является массив информации обо всех найденных на изображении линиях маршрута. В массиве содержатся данные о начальной и конечной точках линии, углах поворота, точках пересечения и информация о последовательности следования линий.

Отметим, что разработанная программа не предъявляет требований к протяжённости и количеству разветвлений маршрута следования робота, поэтому возможна организация крупных разветвлённых сетей с любым количеством роботов.

Алгоритм работы

Предлагаемый алгоритм работы показан на рис. 2.

После ввода изображения и определения порога яркости между фоном и линией происходит сканирование изображения вдоль его нижней границы. При нахождении отрезка, имеющего определённый цвет и длину больше заданной, отрезок сохраняется как предполагаемое начало пути. После окончания сканирования имеем список предполагаемых начал пути.

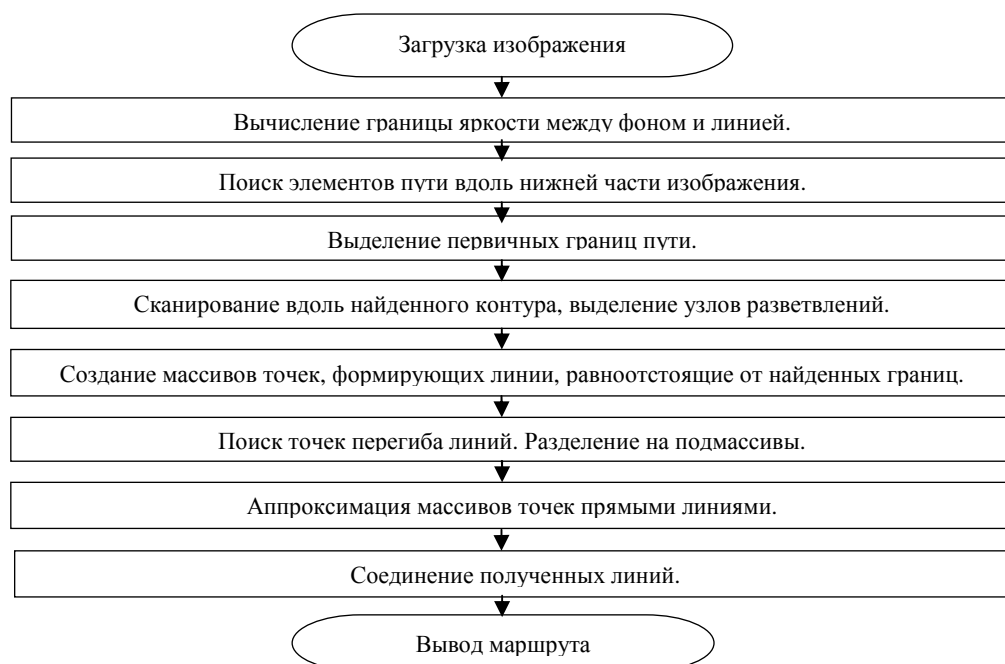


Рис. 2. Блок-схема алгоритма

Затем от крайних точек найденных отрезков при помощи алгоритма Жука [1] проводится выделение границ предполагаемого пути. Следование по границе прекращается, если она пришла в своё начало, либо вышла за пределы изображения. Замкнутые границы помечаются как помехи и далее не рассматриваются. Остальные границы сохраняются.

Возможны случаи, когда остаются ненайденные границы, вследствие существования разветвлений на линии. Для их выделения создаётся временное изображение – копия реального. На него наносятся все найденные границы. Далее осуществляется проход по найденной границе (либо слева, либо справа). Каждый раз, сдвигаясь на несколько точек вдоль границы, определяется прямая, перпендикулярная к ней. Следуя по направлению прямой, вычисляется ширина линии на реальном изображении, а также предполагаемая ширина линии на временном изображении. Две полученные величины сравниваются:

- Если две величины приблизительно равны, и больше предыдущих измерений на установленную величину, то ширина игнорируется как возможная помеха.
- Если же ширина линии на исходном изображении меньше, чем полученная на временном изображении, то считается, что в этой точке произошло разветвление пути. Вычисляется координата точки по другую сторону от линии, выполняется выделение новой границы на другой стороне. Новая граница сохраняется и наносится на временное изображение.

- Если оба условия не выполняются, то осуществляется переход к следующей точке.

Таким образом, находятся все ранее не обнаруженные границы пути.

Во время поиска найденных границ после определения толщины линии вдоль прямой, перпендикулярной границе, вычисляется точка, расположенная по центру пути (как середина перпендикуляра). Найденная точка добавляется в массив точек, из которых формируется путь. Стоит отметить, что один массив содержит точки от одного разветвления пути до другого.

Для определения критических перегибов линии создаётся массив углов между каждым отрезком ломанной, полученной из массива точек. Из этого массива углов выделяются локальные максимумы. Если какой-либо из максимумов превышает критическое значение, то массив точек разделяется на два. Одновременно углы суммируются. Если суммарный угол превышает критическое значение, то массив точек тоже разделяется на два подмассива.

Полученные массивы точек линейно аппроксимируются по методу наименьших квадратов (МНК) [2] в виде $y = a \cdot x + b$. Для оценки погрешностей определяется дисперсия каждой из прямых

$$a = \frac{n \cdot \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \cdot \sum_{i=1}^n y_i}{n \cdot \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}, b = \frac{\sum_{i=1}^n y_i - a \cdot \sum_{i=1}^n x_i}{n}, D = \frac{\sum_{i=1}^n (y_i - a \cdot x_i - b)^2}{n},$$

где x_i , y_i – координаты i -ой точки, D – дисперсия. Для построения итогового пути определяются точки пересечений прямых, т.е. из прямых находятся отрезки траектории движения робота.

Полученная траектория движения предоставляется в систему управления роботом.

Моделирование алгоритма

В соответствии с разработанным алгоритмом была создана программа для проверки его работоспособности. Также была написана вспомогательная программа для создания тестовых изображений пути. В процессе отладки алгоритма было создано 69 изображений, каждое из которых использовалось для тестирования отдельных элементов алгоритма и их действие в целом. Проверке подвергались как «чистые», так и зашумлённые изображения.

На рис. 3 представлена последовательность анализа одного из изображений: само изображение с отмеченными найденными точками вдоль нижней части изображения (Рис. 3а); первичные выделенные границы (Рис. 3б); все выделенные границы – в том числе не найденные в предыдущем этапе (Рис. 3в); финальная ломаная линия пути, наложенная на

первоначальное изображение (Рис. 3г). На Рис. 3д показаны полученные в процессе анализа перпендикуляры к границам линии и линия, проведенная через середины перпендикуляров.

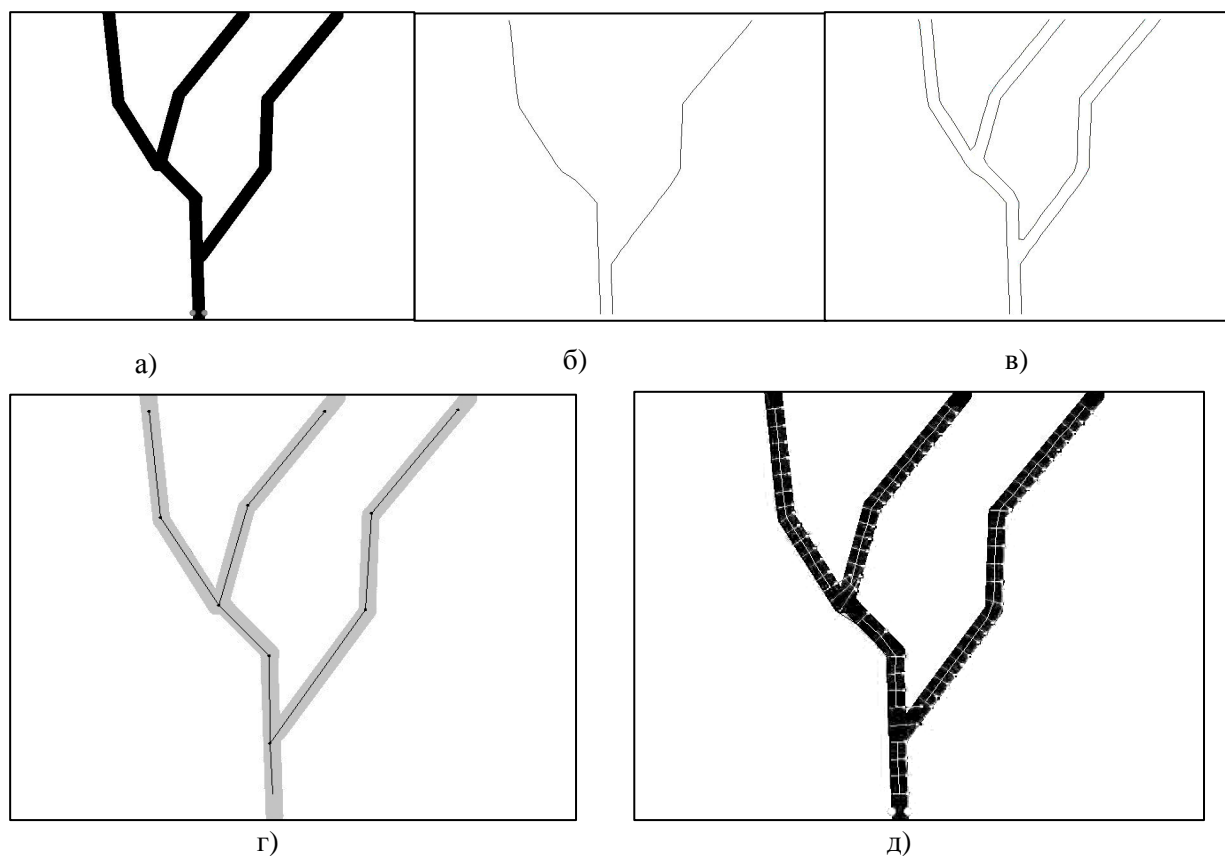


Рис. 3. Последовательность анализа изображения

В случае сильного зашумления изображения вычисленный программой путь может иметь деформации. Линии могут сдвигаться, дублироваться (Рис. 4). Некоторые участки могут быть упущены.

Для проверки точности работы программы было проведено моделирование. На тестовое изображение наносилась одна прямая с известными углом наклона и сдвигом по горизонтали. ($x = ay + b$). Были получены зависимости абсолютной погрешности от степени зашумлённости изображения; от сдвига прямой по горизонтали относительно левого края изображения и от угла наклона прямой. Результаты измерений представлены в таблице 1. Формат используемого изображения – 640:480, средний размер помех – 10 пикселей. При тестировании зависимостей от сдвига и угла количество помех бралось равным 300.

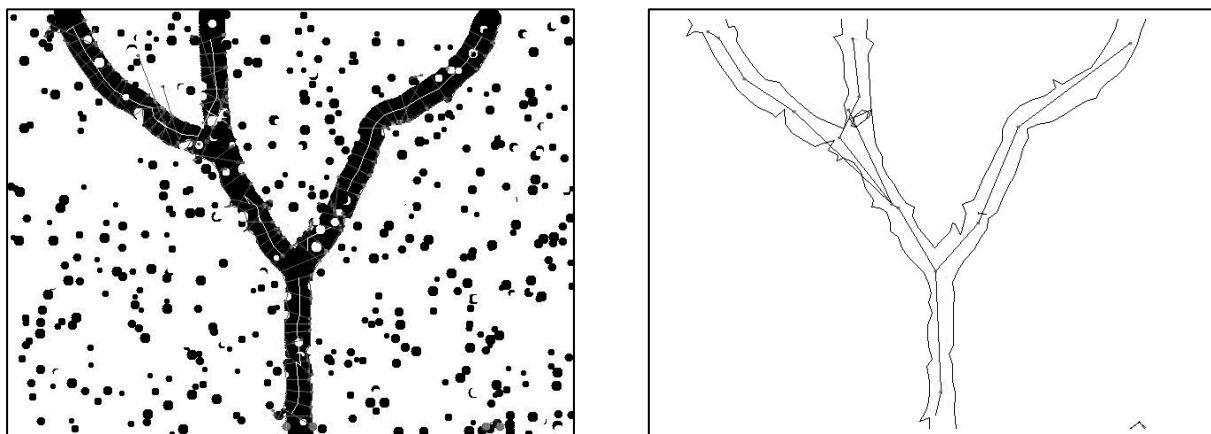


Рис. 4. Зашумлённое изображение. Полученные границы и финальная ломаная пути

Таблица 1

Погрешности измерений

Количество помех	Погрешность угла, °	Погрешность сдвига, пикс.	Сдвиг, пикс.	Погрешность угла, °	Погрешность сдвига, пикс.	Угол, °	Погрешность угла, °	Погрешность сдвига, пикс.
0	0,00	0,25	200	0,09	0,46	-45	4,97	6,54
300	-0,80	2,22	300	-0,69	-3,13	-30	-1,89	10,18
500	-1,34	0,08	400	-0,10	0,58	0	2,63	26,00
900	0,39	-2,71	500	-0,37	-2,02	30	0,09	-0,33
1100	2,89	-24,77	600	0,01	0,48	45	-1,71	11,44

При тестировании влияния помех при большом их количестве также возникали участки с непредсказуемым искривлением линий маршрута. В этих случаях брался самый длинный отрезок с наибольшей погрешностью.

К линиям маршрута не предъявляются жёсткие требования: они могут быть произвольной длины и кривизны. Максимальная ширина линии не должна превышать ширину изображения; минимальная ширина линии должна быть больше размера помех. Линии могут произвольно разветвляться, но следует избегать разветвления на четыре и более линии в одной точке поскольку снижается точность измерения. Линии должны быть контрастными, а цвет линий должен быть одинаковым на всём своем протяжении.

Анализируя результаты, можно сказать, что алгоритм показал себя как в целом нечувствительный к помехам. Решающее влияние на идентификацию линий маршрута оказывали лишь помехи, разрывающие или сильно искажающие часть линии. Именно они и вносили наибольшую погрешность в результат измерений и даже приводили к тому, что алгоритм переставал работать.

Экспериментальные исследования

Для экспериментальной проверки работы алгоритма в помещении был построен участок линии маршрута предполагаемого робота. Линия наносилась на поверхность, имитирующую пол складского помещения. Было получено изображение построенного участка в формате 640:480 и затем обработано (Рис. 5).



Рис. 5. Исходное изображение линии маршрута и результат его обработки

Характеристики системы, осуществляющей обработку изображений, представлены в Таблице 2.

Таблица 2

Характеристики системы

Характеристики системы		Характеристики программы	
ОС	Windows 7	Объем программы	81,5 КБ
ОЗУ	4 ГБ	Среднее время обработки	436 мс
Частота процессора	2,4 ГГц	Формат анализируемых изображений	640x480
Количество ядер	2 ядра		

В результате проведенного эксперимента была подтверждена работоспособность системы.

Выводы

В ходе проведенной работы был разработан алгоритм выделения маршрута для мобильного робота. Работоспособность алгоритма проверена множеством экспериментов на изображениях со сложной конфигурацией линии, с разной степенью зашумленности изображения, а также экспериментами с использованием изображения реальной линии маршрута. Исследования показали, что алгоритм является помехоустойчивым: время обработки изображения и погрешность измерений существенно не зависят от количества помех. Алгоритм теряет работоспособность только при частичном или полном обрыве.

Список литературы

1. Визильтер Ю.В., Желтов С.Ю., Бондаренко А.В., Ососков М.В., Моржин А.В. Обработка и анализ изображений в задачах машинного зрения: курс лекций и практических занятий. М.: Физматкнига, 2010. 672 с.
2. Бронштейн И.Н., Семендяев К.А. Справочник по математике для инженеров и учащихся втузов: учеб. пособие. СПб.: Лань, 2010. 608 с.
3. Метод наименьших квадратов (МНК). Режим доступа: <http://www.cleverstudents.ru/articles/mnk.html> (дата обращения 04.06.2014).