

ОБНАРУЖЕНИЕ ЛЮДЕЙ НА ИЗОБРАЖЕНИИ С ИНФРАКРАСНОЙ КАМЕРЫ И ЛИДАРА

Шмелева Екатерина

МЭСИ, гр. ДКО-1216

Москва 2014

ВВЕДЕНИЕ

Задача локализации людей на изображении имеет множество приложений в системах машинного зрения в робототехнике, системах вождения и интеллектуальных системах безопасности и контроля доступа.

Целью курсовой работы является реализация программы, анализирующей изображения, полученные с помощью инфракрасной камеры переднего обзора (FLIR) и сканирующего лидара, с целью определения местоположения людей на снимках.

Поставленная цель определила необходимость решения следующих задач:

- Провести анализ существующих методов локализации людей на изображении, а также подходов к построению алгоритмов улучшения изображений;
- Создать программу, локализующую людей на изображениях с инфракрасной камеры и лидара;
- Провести экспериментальные исследования.

ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ

Для извлечения областей интереса на изображении используется снимок с инфракрасной камеры. Процесс основывается на предположении, что тело человека теплее окружающей среды.

Для получения расстояния до локализованных областей интереса используется сканирующий лидар. Области интереса, выделенные на предыдущем этапе, масштабируются таким образом, чтобы найденный объект вписывался в прямоугольник один на два метра в координатах реального мира.

Последним шагом к обнаружению людей на изображении является классификация найденных объектов.

1. Максимально устойчивые области экстремума

Для локализации областей интереса необходимо найти так называемые максимально устойчивые области экстремума (англ. *Maximally Stable Extremal Regions*).



Рисунок 1. Исходное изображение



Рисунок 2. Максимально устойчивые области экстремума

Под максимально устойчивыми областями экстремума понимаются области, размер которых остается практически неизменным при некотором количестве изменений порога бинаризации. Нахождение максимально устойчивых областей экстремума происходит следующим образом:

1. Все пиксели сортируются в порядке убывания интенсивности. Значение интенсивности пикселя изменяется в пределах от 0 до 255, поэтому для сортировки удобно использовать алгоритм блочной сортировки, вычислительная сложность которого составляет $O(n)$.
2. Пиксели последовательно возвращаются на изображение. Одновременно с этим происходит обновление системы непересекающихся множеств (англ. *disjoint-set data structure*) и построение дерева вложенных областей экстремума.

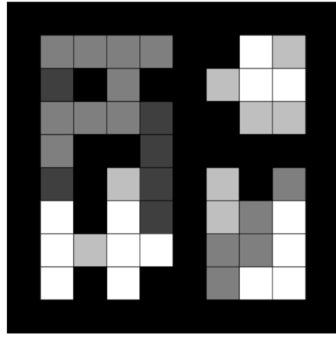


Рисунок 3. Исходное изображение

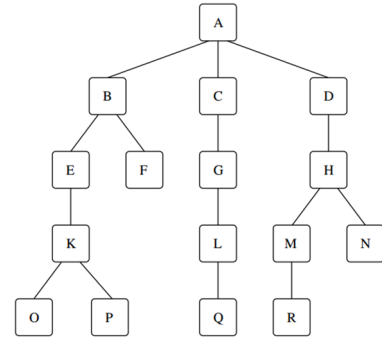


Рисунок 4. Дерево вложенных областей экстремума

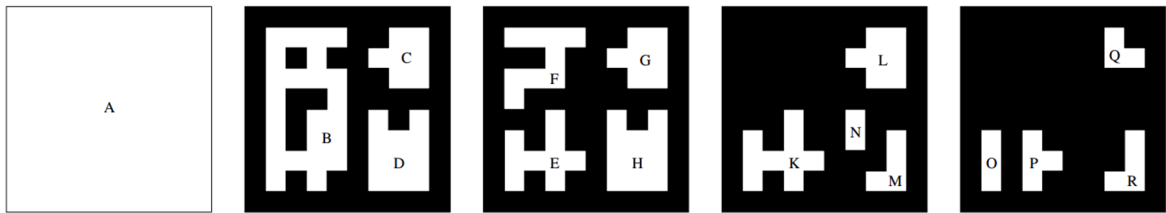


Рисунок 5. Пороговая бинаризация

Сложность алгоритма составляет $O(n \times \alpha(n))$, где $\alpha(n)$ – функция, обратная к крайне быстро растущей функции Аккермана. При анализе асимптотики можно считать, что для всех практически встречающихся n значение функции $\alpha(n)$ меньше 5, поэтому данный шаг выполняется за квазилинейное время.

3. Вычисляется значение вариации для каждой области экстремума:

$$variation_i = \frac{area_{i+\Delta} - area_i}{area_i}.$$

Здесь i представляет собой область с максимальной интенсивностью i , а $i + \Delta$, соответственно, расширенную область с максимальной интенсивностью $i + \Delta$. Значение Δ является параметром алгоритма.

4. На последнем шаге находятся максимально устойчивые области экстремума. Значение вариации этих областей является локальным минимумом, то есть меньшим, чем у непосредственного предка и наследника.

2. Определение размеров объекта

Для вычисления размера объекта на матрице камеры (Y_2) необходимо знать реальный размер объекта (Y_1), расстояние от линзы до объекта (X_1) и фокальное расстояние или расстояние от линзы до матрицы (X_2).

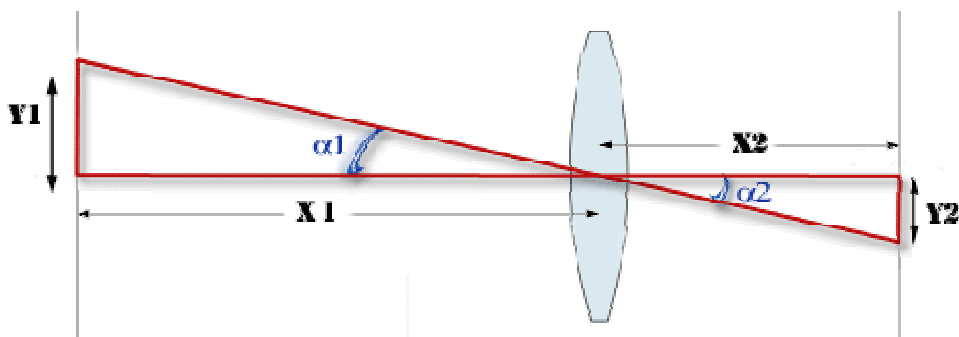


Рисунок 6.

Из равенства вертикальных углов α_1 и α_2 следует равенство тангенсов этих углов. Отсюда получаем соотношение, связывающее физическое расстояние до объекта, его размер на матрице камеры и другие переменные:

$$\frac{X_1}{Y_1} = \frac{X_2}{Y_2}.$$

Выделив из уравнения Y_2 и сделав еще несколько преобразований, получим формулу для определения высоты объекта на изображении.

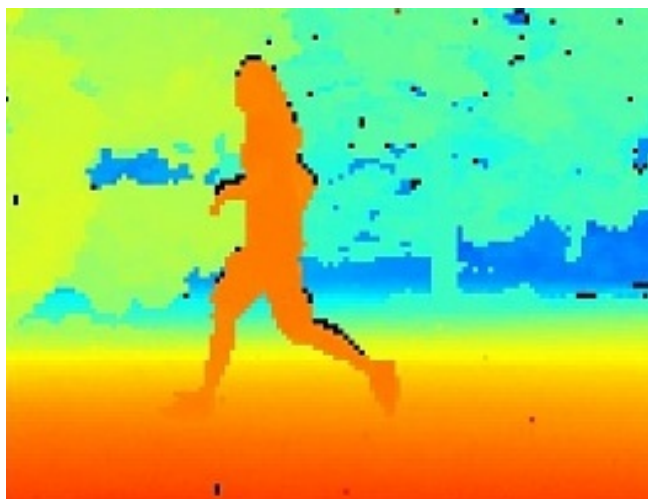


Рисунок 7. Снимок, полученный по данным лидара

На снимке со сканирующего лидара расстояние до объекта определяется исходя из цвета объекта. Цвета изображения являются спектральными, то есть каждый из них имеет максимальную насыщенность в пределах цветового тона, который он имеет. Таким образом, соотнеся метровую и цветовую шкалы, можно определить расстояние до точки на изображении. Для определения расстояния до объекта выбирается его центральная точка.

3. Классификация объектов

После обнаружения регионов интереса необходимо проанализировать, находится ли в данной области человек или иной объект. В качестве метрики для классификации объектов используется дисперсия (*англ. dispersiveness*). Метрика основана на том, что люди, в общем, меньше, чем, например, автомобили, и имеют более сложную форму.

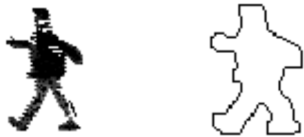


Рисунок 8.
dispersiveness: 61.8



Рисунок 9.
dispersiveness: 41.0

Дисперсия вычисляется по следующей формуле:

$$dispersiveness = \frac{perimeter^2}{square}.$$

Таким образом, изображения людей обладают большим значением дисперсии, чем изображения частей других объектов. Для нахождения значения дисперсии необходимо знать площадь и периметр объекта.

Для получения контура объекта применяется многоступенчатый детектор границ Кэнни.

4. Детектор границ Кэнни

В отличие от других алгоритмов, детектор границ Кэнни не ограничивается вычислением градиента сглаженного изображения. В контуре границы оставляются только точки максимума градиента изображения, а не максимальные точки, лежащие рядом с границей, удаляются. Алгоритм состоит из следующих шагов:

1. *Сглаживание.* Оператор Кэнни использует фильтр который может быть хорошо приближен к первой производной гауссианы.
2. *Поиск градиентов.* Границы отмечаются там, где градиент изображения приобретает максимальное значение. В данной работе для поиска градиентов применяется оператор Собеля.
3. *Подавление не-максимумов.* Только локальные максимумы отмечаются как границы.
4. *Двойная пороговая фильтрация.* Потенциальные границы определяются порогами.
5. *Трассировка области неоднозначности.* Итоговые границы определяются путём подавления всех краёв, несвязанных с сильными границами.



Рисунок 10. Исходное изображение


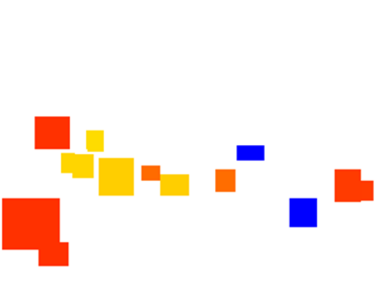









Рисунок 11. Границы объектов

СПИСОК ЛИТЕРАТУРЫ И ИНТЕРНЕТ-РЕСУРСОВ

1. Фролов В. Обнаружение пешеходов [Электронный ресурс] // Хабрахабр URL <http://habrahabr.ru/post/100820/> (дата обращения: 26.12.2014).
2. Laurent Najman, Michel Couprie "Building the Component Tree in Quasi-Linear Time" // Image Processing, IEEE Transactions on (Volume: 15, Issue: 11). - IEEE, 2006 - С. 3531 - 3539
3. John Canny "A Computational Approach to Edge Detection" // Pattern Analysis and Machine Intelligence, IEEE Transactions on (Volume: PAMI-8, Issue: 6). - IEEE, 1986 - С. 679 - 698
4. Michael Donoser, Horst Bischof "Efficient Maximally Stable Extremal Region (MSER) Tracking" // Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on (Volume: 1). - IEEE, 2006 - С. 553 - 560
5. J. Matas, O. Chum, M. Urban, T. Pajdla "Robust wide baseline stereo from maximally stable extremal regions" // Proceedings of the British Machine Conference. - BMVA Press, 2002 - С. 384 - 396.
6. R.E. Tarjan "Efficiency of a good but not linear set union algorithm" // Journal of the ACM (Vol. 22, No. 2). - ACM, 1975 - С. 215 - 225.
7. A.L. Hironobu, A.J. Lipton, H. Fujiyoshi, R.S. Patil "Moving Target Classification and Tracking From Real-Time Video" // Applications of Computer Vision, 1998. WACV '98. Proceedings, Fourth IEEE Workshop on. - IEEE, 1998 - С. 8 - 14
8. Dr. Vinayak Ashok Bharadi "Canny Edge Detection in C#" [Электронный ресурс] // CodeProject URL <http://www.codeproject.com/Articles/93642/Canny-Edge-Detection-in-C> (дата обращения: 26.12.2014).
9. Mohammad Reza Khosravi "Range Finder" [Электронный ресурс] // CodeProject URL <http://www.codeproject.com/Articles/35029/Range-Finder> (дата обращения: 26.12.2014).
10. Sven Walter "MetroFramework" [Электронный ресурс] // GitHub URL <https://github.com/thielj/MetroFramework> (дата обращения: 26.12.2014).

Приложение 1. Примеры работы программы

№	ИК-камера	Лидар	Вывод
1			
2			
3			
4	