

## Алгоритм робастной детекции отрезка в кубе данных на основе быстрого преобразования Хафа

Е.Н. Асватов<sup>1</sup>, Е.И. Еришов<sup>1,2</sup><sup>1</sup>Московский физико-технический институт (государственный университет)<sup>2</sup>Институт проблем передачи информации им. А.А. Харкевича

Одной из популярных задач математической статистики является задача линейной регрессии. Существует множество различных алгоритмов, реализующих решение данной задачи. Однако известно, что сложность работы подобных алгоритмов полиномиально возрастает с увеличением количества данных. Более того, существенная часть из них не обладает робастностью к нежелательным выбросам в этих данных. В данной работе предлагается новый метод робастной линейной регрессии в трехмерном массиве данных. К преимуществам метода можно отнести робастность к выбросовым данным, а также независимость от размера обучающей выборки. В отличие от аналогов сложность данного алгоритма определяется линейными размерами куба.

Алгоритм с подобными свойствами актуален при решении различных задач обработки изображений и компьютерного зрения. К таким задачам можно отнести анализ карты глубины с целью детекции препятствий, ухода от столкновений и пр. Другим примером потенциального применения алгоритма является компьютерное зрение в медицине: детекция иглы для люмбальной пункции по трехмерным УЗИ данным. Корректное и робастное решение данной задачи может быть использовано как для улучшения уже существующих систем, так и для разработки новых автономных систем.

Рассмотрим дискретный куб с ребром длиной  $n$  вокселей. Задача: найти в кубе прямую с наибольшей кумулятивной суммой вдоль нее. Всего таких дискретных линий в кубе  $O(n^4)$ . Для того, чтобы параметризовать линию в трехмерном пространстве необходимо зафиксировать 4 степени свободы. Также, необходимо произвести суммирование вдоль каждой линии, что составляет еще  $O(n)$  операций. Таким образом сложность наивного алгоритма  $O(n^5)$ .

В работе предлагается алгоритм с асимптотической сложностью  $O(n^3 \log n)$ . Объем необходимой памяти составляет  $O(n^3)$ . Данный алгоритм основан на последовательном применении быстрого трехмерного преобразования Хафа (ЗБПХ) [2] и быстрого плоского преобразования Хафа (БПХ) [1].

На первом шаге алгоритма вычисляется сумма по всем возможным плоскостям в кубе с использованием ЗБПХ. Результат работы алгоритма (пространство Хафа) хранится в массиве памяти размером  $O(n^3)$ . На следующем шаге определяется плоскость, сумма вдоль которой является наибольшей. Это осуществляется путем полного перебора по пространству Хафа, что не влияет на общую асимптотическую сложность всей алгоритмической схемы. Стоит отметить, что в настоящее время не найдено таких свойств пространства Хафа, которые бы позволяли осуществить глобальную оптимизацию с гарантией правильного ответа быстрее, чем полный перебор. Таким образом эти два шага алгоритма имеют асимптотическую сложность  $O(n^3 \log n)$ .

На следующем шаге в найденной плоскости исходного куба также вычисляется преобразование Хафа с использованием БПХ. В полученном двумерном пространстве Хафа, аналогично, полным перебором осуществляется поиск прямой с наибольшей суммой. Найденная прямая и будет ответом алгоритма. Для реализации этих двух шагов необходимо потратить  $O(n^2 \log n)$  вычислений. Стоит заметить, что данная схема может быть применена для детекции первых  $N$  прямых с наибольшей суммой. Для этого достаточно занулять значения в кубе в пикселях, соответствующим найденной прямой на данном шаге, после чего повторить вышеизложенный алгоритм. Таким образом итоговая асимптотическая сложность алгоритма составляет  $O(n^3 \log n)$  и для константного количества детектируемых прямых.

Предложенное решение можно обобщить на многомерный случай (четыре и более). Стоит отметить, что в отличие от существующих вероятностных алгоритмов данный гарантирует глобальную оптимальность результата.

Сравнительный график времени работы предложенного алгоритма и обычного трехмерного преобразования Хафа для разных размеров кубов данных приведен на рис. 1.

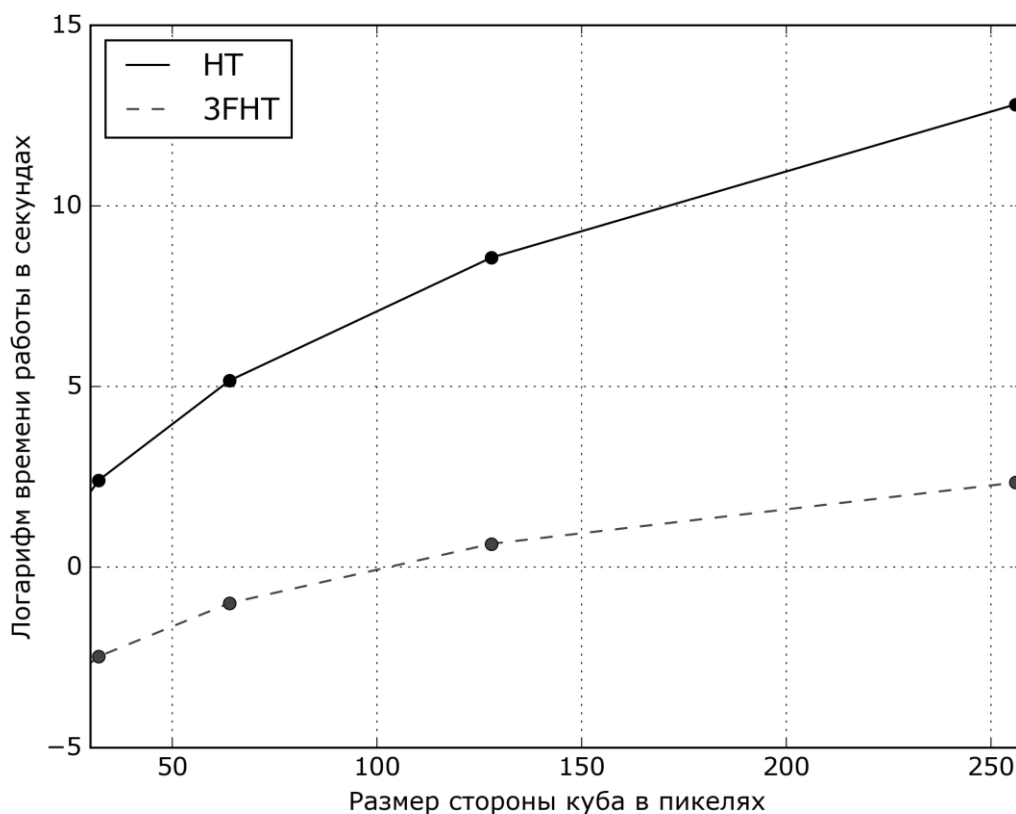


Рис. 1 Графики времени работы наивного преобразования Хафа (HT) и быстрого преобразования Хафа (3FHT).

#### Литература

1. Nikolaev D.P., Karpenko S.M., Nikolaev I.P., Nikolayev P.P. Hough transform: underestimated tool in the computer vision field // Proceedings of the 22th European Conference on Modelling and Simulation. Vol. 238. 2008.
2. Ershov E., Terekhin A., Karpenko S., Nikolaev D., Postnikov V. Fast 3D Hough Transform computation // 30th European Conference on Modelling and Simulation. (In printing)