

Exhibit 47

LSH Algorithm and Implementation (E2LSH)

Locality-Sensitive Hashing (LSH) is an algorithm for solving the (approximate/exact) Near Neighbor Search in high dimensional spaces. On this webpage, you will find pointers to the newest LSH algorithm in Euclidean (l_2) spaces, as well as the description of the **E2LSH package**, an implementation of this new algorithm for the Euclidean space.

- **Algorithm description:**

- **CACM survey of LSH (2008):** "Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions" (by Alexandr Andoni and Piotr Indyk). *Communications of the ACM*, vol. 51, no. 1, 2008, pp. 117-122.
[directly from CACM](#) (for free). [local copy](#) (see [CACM disclaimer](#)).

- **Most recent algorithm (2006):** "Near-Optimal Hashing Algorithms for Near Neighbor Problem in High Dimensions" (by Alexandr Andoni and Piotr Indyk). In *Proceedings of the Symposium on Foundations of Computer Science (FOCS'06)*, 2006.

Slides: Here are [some slides](#) on the LSH algorithm from a talk given by Piotr Indyk.

- **Earlier algorithm for Euclidean space (2004):** a good introduction to LSH, and the description of affairs as of 2005, is in the following book chapter

[Locality-Sensitive Hashing Scheme Based on p-Stable Distributions](#) (by Alexandr Andoni, Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab Mirrokni), appearing in the book *Nearest Neighbor Methods in Learning and Vision: Theory and Practice*, by T. Darrell and P. Indyk and G. Shakhnarovich (eds.), MIT Press, 2006.

See also the [book introduction](#) for a smooth introduction to NN problem and LSH.

- **Original LSH algorithm (1999):** the best algorithm for the Hamming space remains the one described, e.g. in [\[GIM'99\] paper](#).

- **Implementation of LSH:** Currently, we only have an alpha-version available - the E2LSH package. The code is based on the algorithm described in the book chapter (2006) from above. [Download the code](#).

You can also download the [manual](#) for the code to see its functionality. The code has been developed by [Alex Andoni](#) in 2004-2005.

This research is supported by NSF CAREER Grant #0133849 "Approximate Algorithms for High-dimensional Geometric Problems".