Locality-Preserving Hashing in Multidimensional Spaces*

Piotr Indyk †
Department of Computer Science
Stanford University
indyk@cs.stanford.edu

Rajeev Motwani ‡
Department of Computer Science
Stanford University
rajeev@theory.stanford.edu

Prabhakar Raghavan
IBM Almaden Research Center
pragh@almaden.ibm.com

Santosh Vempala §
School of Computer Science
Carnegie-Mellon University
vempala@cs.cmu.edu

In a recent paper, Linial and Sasson [2] proved the following theorem about hash functions:

Theorem 1 *There exists a family $\mathcal{G}$ of functions from an integer line $[1, \ldots, U]$ to $[1, \ldots, R]$ and a constant C such that for any $S \subset [1, \ldots, U]$ with $|S| \leq C\sqrt{R}$:*

- $\Pr_{f \in \mathcal{G}}(f_{|S}$ *is one to one*$) \geq \frac{1}{2}$
- *all $f \in \mathcal{G}$ are* non-expansive*, i.e., for any $p, q \in U$ $d(f(p), f(q)) \leq d(p, q)$.*

The family $\mathcal{G}$ contains $O(|U|)$ functions, each of which is computable in $O(1)$ operations.

Their result gives a family of hash functions with the surprising property that points close to each other in the domain are hashed to points close to each other in the range. A potential application of their result is to find near neighbors to points in one-dimensional space. Given a set of points on the line, one can hash these points so that we can search for the points closest to a query point as follows: we hash the query q to $h(q)$, then search the neighborhood of $h(q)$ in the hash table to retrieve, say, the nearest k points within some distance δ in the domain in time $\min\{k, \delta\}$. The advantage of the locality-preserving property is that it affords good paging performance: since the neighborhood of q (in the domain) is not scattered all over the range, the neighborhood of $h(q)$ in the hash table exhibits good locality of reference. This is counter to Knuth's suggestion that

> *In a virtual memory enviroment we probably ought to use tree search or digital tree search, instead of creating a large scatter table that requires bringing a new page nearly every time we hash a key.*

Of course, there are many other good ways of retrieving near neighbors in one dimension. However, efficient near-neighbor retrieval is considerably harder, and of growing importance, in higher dimensions. The main application comes from information retrieval: the process of retrieving text and multimedia documents matching a specified query. Other instances of near-neighbor search appear in algorithms for pattern recognition, statistics and data analysis, machine learning, data compression, data mining, and image analysis.

*A preliminary version of this paper was published in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 618–625, 1997

†Supported by NSF Award CCR-9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation.

‡Supported by an Alfred P. Sloan Research Fellowship, an IBM Faculty Partnership Award, an ARO MURI Grant DAAH04-96-1-0007, and NSF Young Investigator Award CCR-9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation.

§Supported in part by NSF National Young Investigator grant CCR-9357793.

Generalizing the approach of Linial and Sasson to higher dimensions offers promise as a way of tackling this problem. We present such a generalized construction of locality-preserving hash functions in higher dimensions, together with negative results suggesting that our construction is theoretically the best possible. In addition to the theoretical value of these results, our construction will in principle afford fast retrieval and good paging behavior during search. Realistically, though, it will in practice only work for modest values of the dimension d (say, 10-20). Nevertheless it offers a simple approach for indexing problems (if the feature space after dimensionality reduction has moderate dimension) and in iterative computations for sparse finite-element relaxation methods.

Summary of Results: In our work [1], we introduce the notion of *locality-preserving* hashing. More specifically, a function $h : D \to I$ is said to be *c-expansive* under a distance metric d if for any $p, q \in D$, $d(h(p), h(q)) \leq d(p, q) + c$. Our first result is a construction of a family locality-preserving hash functions in two dimensions; this is then extended to higher dimensions. For d-dimensions, our functions are $\sqrt{d}$-expansive under the l_2 norm, d-expansive under the l_1 norm, and non-expansive under the l_∞ norm. The constants in our guarantees (bucket size, collision probability) grow with the dimension, roughly as $O(c^d)$ where c is a fixed constant for l_2 norm and $O(d^d)$ for l_1 and l_∞.

We also present several negative results exploring the intrinsic limitations of locality-preserving hash functions. First, we establish a lower bound of $\Omega(1/R)$ on the collision probability for any family of non-expansive functions in $d \geq 2$ dimensions; this implies that obtaining low collision probability is essentially impossible for higher dimensions. Then we restrict ourselves to polynomial sized families of natural subclass of non-expansive hash functions. We show that even if we allow to store up to c elements in each bucket, no such family is able to hash more than roughly $O(R^{1-1/c})$ elements. Both results suggest that relaxation of the non-expansiveness constraint is essential in order to provide good bounds.

References

[1] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala. Locality-Preserving Hashing in Multidimensional Spaces. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 618–625, 1997.

[2] N. Linial and O. Sasson. Non-Expansive Hashing, In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 509–517, 1996.