

# Optimizing Massive Distance Computations in Pattern Recognition

Andras Farago

**Abstract**—It is a common task in pattern recognition to evaluate the similarity of large data objects. These are often represented by high dimensional vectors. A frequently used mathematical model for evaluating their similarity is to view them as points (vectors) in a high dimensional space, and compute their distances from each other. The “distance,” however, can be defined in a very complicated way, it may be much more complex than the well known Euclidean distance. Therefore, the algorithmic bottleneck often becomes the number of distance computations that need to be carried out. We consider the case when we have to compute all the distances between  $n$  objects, where  $n$  is large. Without any shortcuts it takes  $n(n - 1)/2 = O(n^2)$  distance computations. In those applications where the distances are complicated, being defined by sophisticated algorithms (such as in speech and image recognition), a quadratically growing number of distance computations becomes a severe bottleneck. We prove the following general result that can help eliminating the bottleneck: for a large and general class of distances it is possible to obtain a very close approximation of each of the  $O(n^2)$  pairwise distances of  $n$  objects by doing only a linear number distance computations, which is optimal with respect to the order of magnitude. Moreover, the approximation factor can be made arbitrarily close to 1, making the approximation error negligible. The needed side computations to achieve this reduction can also be done in polynomial time.

**Index Terms**—Pattern recognition, approximate distance computation, metric space, normed space.

## I. INTRODUCTION

It is a typical task in pattern recognition that various objects are represented by high dimensional vectors and some kind of measure of similarity is defined among them. To visualize the situation, we refer to this measure as *distance*, even if it may not satisfy all the usual axioms of the mathematical distance in metric spaces.

In some cases these distances can be quite complicated to compute. Such complex similarity measures occur, for example, in speech recognition, image recognition, handwriting recognition, and a number of other pattern recognition tasks. Even if we restrict ourselves to those metrics that actually satisfy the distance axioms, they may still be much harder to compute than the classic Euclidean distance.

Let us recall the well-known distance axioms, denoting the distance function by  $d(x, y)$ : (1) Symmetry:  $d(x, y) = d(y, x)$  for every  $x, y$ ; (2) Positivity:  $d(x, y) > 0$  for every  $x \neq y$ ; (3)

Identity:  $d(x, x) = 0$  for every  $x$ ; (4) Triangle inequality:  $d(x, y) + d(y, z) \geq d(x, z)$  for every  $x, y, z$ .

A classical example that satisfies the axioms is the Euclidean distance in vector spaces, which is, of course, easy to compute. Generally, if we have a set and a function  $d(x, y)$  defined on it, satisfying the above axioms, then it is called a metric space.

In pattern recognition tasks one often uses much more complicated distances than the Euclidean. An example of a significantly harder distance is the Hausdorff-distance, which is used, for example, in computer vision to identify a template in a given target image [1], [2]. The Hausdorff-distance is much more time consuming to compute, yet it is of practical importance, as it is actually used in computer vision and image processing.

Let us now consider the following general task. Given a metric space and  $n$  points in this space. Note that the points, represented by vectors, may actually be encoded descriptions of much more complicated objects, such as images. We want to compute all the distances between the  $n$  points. This may be important for several practical reasons: in pattern recognition algorithms, clustering, analyzing the geometry, such as finding nearest and farthest pairs etc.

There are altogether  $n(n - 1)/2 = O(n^2)$  distances among  $n$  points. As complex distance computation can be quite demanding, the computation of a quadratically growing number of distances can become prohibitively slow. Therefore, we are interested in the question: can these  $O(n^2)$  distances be obtained using only a *linear* number of distance computations?

It is clear that sublinear cannot be enough, if we can only obtain information about the data objects through computing their distances. The reason is that with strictly less than  $n$  distance computations there must be a data object among the  $n$  ones, which is not involved in any distance computation. Such an object does not influence the results at all, so it may be changed arbitrarily, without changing the result. This is clearly impossible, if we want to compute *all* distances among  $n$  data objects, as *some* of the distances must depend on this particular data object. Thus, if the task can be solved with a linear number of distance computations, that solution is optimal with respect to the order of magnitude.

In the rest of the paper we show the surprising fact that, under some quite general conditions, we can indeed achieve this optimal order of magnitude. The price is that we need to accept approximate distances, instead of the exact ones. The approximation factor, however, can be brought arbitrarily close to 1, so the error can be made negligible.

Before presenting our solution, let us provide a brief historical survey on related results, to put our method in perspective.

Manuscript received October 10, 2013; revised December 7, 2013. The author is grateful for the support of NSF Grant CNS-1018760.

Andras Farago is with the Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080, USA (e-mail: farago@utdallas.edu).

## II. HISTORICAL REVIEW OF RELATED RESULTS

It is a common situation in many applications that the data is represented by a large number of points in a metric space. This can be viewed such that the data itself forms a finite metric space. In these situations it is very useful if we can simplify the data by mapping the source points into a significantly simpler target metric space, such that the pairwise distances between data points are preserved. If they are preserved exactly, then it is called *isometric embedding*. Unfortunately, perfect isometric embedding into a simpler space is rarely possible. Therefore, one usually looks for the relaxed case, when the distances are preserved only approximately, but with small error. In this case we talk about *low distortion metric embedding*. If the error can be made arbitrarily small, then it is referred to as *almost isometric embedding*. Such solutions have provided tremendous help in many algorithms and applications.

While our method aims at computing all-pairs distances, rather than an embedding, it is still methodologically related to the issue of almost isometric embedding that is why we review the main embedding results here.

The embedding task usually comes in two typical flavors. The first one is when the distance metric in the source space is too complicated. Here the main goal is to replace the complicated metric with a much simpler one, which is typically the well-known  $l_p$  metric with  $p=1, 2$ , or  $\infty$ . Recall that the  $l_p$  distance of two vectors  $x = (x_1, \dots, x_n)$  and  $y = (y_1, \dots, y_n)$  is defined as:

$$d_p(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

In the special case of  $l = 1$  we get the  $l_1$  metric:

$$d_1(x, y) = \sum_{i=1}^n |x_i - y_i|$$

while for  $l = 2$  the Euclidean distance is obtained:

$$d_2(x, y) = \left( \sum_{i=1}^n (x_i - y_i)^2 \right)^{1/2}$$

If  $l = \infty$ , then we get the maximum norm based distance:

$$d_\infty(x, y) = \max_i |x_i - y_i|$$

It may also happen that the data points themselves are not structured (or we may not be interested in their intrinsic structure, so they can be represented just by abstract labels), then the key information is carried solely by the metric. In that case we are led to the task of embedding a general finite metric space into  $l_p^d$ , which is the  $d$ -dimensional real vector space  $\mathbf{R}^d$ , equipped with the  $l_p$  norm.

The other flavor is when the source data may reside in a space of simple structure, such as  $\mathbf{R}^d$  with the Euclidean norm  $l_2$ , but the dimension is excessively high. Even if the metric is simple, the high dimension can dramatically slow down algorithms that often have running times exponential in the dimension (known as the “curse of dimensionality”). In this case the goal of embedding is *dimensionality reduction*: the

mapping into a much lower dimensional space can provide significant help, even if the metric is not simplified.

Since finding good algorithms for metric embedding with low distortion is far from trivial, it is one of the rare fields where clear practical importance meets the intrinsic mathematical beauty of the question.

Embedding problems of finite metric spaces have been the subject of extensive research for a very long time, yielding a large number of results. Below we briefly review some fundamental results about low distortion embedding of  $n$ -point metric spaces into  $l_d^p$  with  $d \ll n$ , as these cases relate most closely to our work.

A classic result is Bourgain’s Theorem [3], which says that any  $n$ -point finite metric space can be embedded into an Euclidean space, with distortion  $O(\log n)$ . Linial, London and Rabinovich [4] showed that it also works with target norm  $l_p$  for any  $p$ . The dimension of the target space was originally exponential, but was later reduced to  $O(\log^2 n)$  by Linial, London and Rabinovich, and Matoušek [5]. Linial, London and Rabinovich also proved an  $\Omega(\log n)$  lower bound for the distortion. Further improvement was obtained by Abraham, Bartal and Neiman [6], who reduced the target dimension to  $O(\log n)$ , which is optimal, and also showed that the *average* (but not the worst case!) distortion can be made constant. A lesson from this chain of results is that when a *general*  $n$ -point metric space is embedded into  $l_p^d$ , then neither the dimension  $d$ , nor the distortion can remain bounded as  $n$  grows.

Thus, if we hope to achieve the ideal case of almost isometric embedding into bounded dimension, then we must impose some restriction on the source metric space, rather than allowing an arbitrary source metric. A classic result in this direction is the Johnson-Lindenstrauss Lemma [7], which says that for every  $\epsilon > 0$ , any  $n$  points from an  $l_2$  space can be embedded with  $1 + \epsilon$  distortion (i.e., almost isometrically) into  $l_2^d$ , where  $d = O(\log n / \epsilon^2)$ . Since it is known (see, e.g., Indyk and Matoušek [8]) that the *isometric* embedding of the same points is not possible into  $l_2^d$  with  $d < n - 1$ , therefore, it is an appealing and extremely useful fact that allowing a very slight distortion can bring down the target dimension to  $O(\log n)$ , resulting in exponential dimension reduction. Another useful feature is that the embedding can be computed via a simple random projection. On the other hand, it is not possible to reduce the target dimension to a constant, as there is a known almost matching lower bound of  $\Omega(\log n / (\epsilon^2 \log(1/\epsilon)))$  for the dimension, due to Alon [9], see also Matoušek [10].

The lower bound on the dimension shows that we cannot achieve our ideal goal with  $l_2$  target metric, when the source is an arbitrary normed space. How about  $l_1$  as target metric? Then the situation is even worse, as Brinkman and Charikar [11] proves that for every  $n$ , the embedding of an  $n$ -point  $l_1$  metric into  $l_1^d$  with distortion  $c > 1$  requires  $d = n^{\Omega(1/c^2)}$  in the worst case.

Regarding  $l_\infty$  as the target metric, Matousek [12] showed that any  $n$ -point metric can be embedded into  $l_\infty^d$  With distortion  $c$  and of dimension  $d = O((c + 1)n^{2/(c+1)} \log n)$ . Furthermore, an almost matching lower bound can be proved, so the target dimension cannot remain bounded for a general

source metric with  $l_\infty$  target.

One may ask at this point: is there any nontrivial embedding result with constant target dimension, independent of the number of source data points? While such results do exist, under special conditions, they are quite rare. A folklore example (see Indyk [13]) is that  $l_1^d$  isometrically embeds into  $l_\infty^{d'}$  with  $d' = 2d$ , regardless of the number of source points. Another example is the theorem of Gupta, Krauthgamer and Lee [14], which says that every doubling tree metric embeds into  $l_p^d$  with constant  $d$  and constant distortion (which may not be arbitrarily close to 1), for every  $p \in [1, \infty]$ . Here a *tree metric* is the shortest path metric on the vertices of a tree. A metric is called *doubling*, if its doubling dimension is finite, where the *doubling dimension* is the smallest integer  $k$ , such that every ball can be covered by  $2^k$  balls of half the radius.

Another embedding result into constant dimension is Assouad's Theorem [15]. It says that if the source metric has constant doubling dimension, and  $0 < \gamma < 1$  is a constant, then there exist constants  $d, C$ , such that the "snowflake version" of the source metric embeds into  $l_2^d$  with distortion  $C$ . Here the *snowflake version* of the source metric is the original metric raised to the power of  $\gamma$  (which remains a metric for  $0 < \gamma < 1$ ). The target dimension and the distortion in Assouad's Theorem depend on the doubling dimension and  $\gamma$ , but not on the number of input points. On the other hand, the embedded metric is a modified version of the source metric, not the original itself. Note that the difference between the original and the snowflake version is not bounded for any  $0 < \gamma < 1$ . Assouad conjectured that the embedding also applies to the original source metric (corresponding to  $\gamma = 1$ ), but that was disproved by Semmes [16]. As a further step, Gottlieb and Krauthgamer [17] extended Assouad's Theorem by showing that if the source metric is also Euclidean (beyond having constant doubling dimension), then the embedding of the snowflake version into constant dimension can be done with  $1+\epsilon$  distortion with arbitrary  $\epsilon > 0$ . This again applies to the modified source metric, not the original.

Apparently, the most general result so far on embedding the original  $n$ -point source metric (not its snowflake version) into constant dimension is the theorem of Abraham, Bartal and Neiman [18]. It says that any  $n$ -point metric space of doubling dimension  $k$  can be embedded into  $l_p^d$  with  $d = O(k/\theta)$  and distortion  $O(\log^{1+\theta} n)$ , where  $\theta \in (0, 1]$ . While the target dimension is constant here, the distortion grows to infinity with  $n$ , so it is not an almost isometric embedding.

Our result (to be presented in the next sections) can technically be viewed as an almost isometric embedding, when the source points are from a fixed finite dimensional normed space, and the target space is a constant dimensional metric space. This constant target dimension makes it possible to reduce the number of distance computations to linear order.

### III. THE TASK

In this section we specify the task we are going to solve.

First, we have to impose some restriction on the considered distances, to make the task tractable. The restriction still allows that many different distances are incorporated, including ones that are hard to compute.

We restrict ourselves to distances that are generated by a *norm* in a vector space. To recall the well known concept of a norm, let us review its axioms. A norm is a function that assigns a real number to each vector in a vector space and satisfies the properties listed below. The norm of a vector  $x$  is denoted by  $\|x\|$ .

- Positivity:  $\|x\| > 0$  for each vector  $x \neq \vec{0}$ , where  $\vec{0}$  is the zero vector
- Linearity:  $\|cx\| = |c| \|x\|$  for any vector  $x$  and any scalar  $c$  (this also implies  $\|\vec{0}\| = 0$ )
- Triangle inequality:  $\|x\| + \|y\| \geq \|x + y\|$  for every  $x, y$ .

A vector space with a norm defined on it is called a *normed space*. If the underlying vector space is finite dimensional, then we call it a *finite dimensional normed space*. Note that one may define many different norms on the same vector space. Once we have a norm, it directly generates a distance by

$$d(x, y) = \|x - y\|$$

It is easy to prove that this is indeed a distance, satisfying the distance axioms. It is known, however, that not every distance can be generated by a norm. The restriction we impose is that we only deal with norm-generated distances.

One may suspect at this point that perhaps only the algorithmically "easy" distances are generated by a norm, so by restricting ourselves to norm-generated distances, we may exclude all the hard cases. This is, however, not so at all. The reason is that a norm itself can also be very hard to compute. Here is an example. Let  $C$  be a convex set in a vector space, such that  $C$  contains the origin in its interior. It is known [19] that the following construction generates a norm:

$$\|x\| = \inf \{1/\gamma : \gamma x \in C\}$$

Now choosing the convex set  $C$  such that it is hard to carry out the above optimization makes the norm hard to compute. For example, we can create a norm that is NP-hard to compute, if we choose  $C$  as the convex hull of the incidence vectors of cliques in a graph (with a little shifting to include the origin in its interior).

Now the specific task we are going to solve is detailed below.

#### Task All-Pairs Distance Computation (APDC)

**Input:**  $n$  vectors in a finite dimensional normed space

**Goal:** Compute (approximately) all the  $n(n-1)/2$  pair-wise distances among the vectors

**Assumption:** The algorithm can access the norm of any vector of the space via a (black box) subroutine. Calling the subroutine is counted as a single step in the running time of the algorithm.

#### Requirements:

- 1) **Approximation quality.** The computed approximate

distances can deviate from the exact ones at most by a constant factor. That is, there must exist constants  $0 < \alpha \leq 1$  and  $\beta \geq 1$  (independent of  $n$ ), such that for any vectors  $x, y$

$$\alpha d(x, y) \leq d(x, y) \leq \beta d(x, y) \quad (1)$$

holds, where  $d(x, y) = \|x - y\|$  is the exact distance (defined by the norm), and  $d(x, y)$  is the computed approximation. Moreover, we require that the value of  $\beta - \alpha$  can be made an arbitrarily small positive number, yielding an approximation factor arbitrarily close to 1.

- 2) **The algorithm can call the norm subroutine only  $O(n)$  times.** This makes the problem hard (otherwise we could just compute all the  $O(n^2)$  distances). This requirement reflects the view that the main bottleneck is the computation of a complex distance function.
- 3) **No excessive hidden costs.** If all the vectors fall in a fixed bounded subset (independent of  $n$ ), then the algorithm must run in polynomial time (counting the norm subroutine calls as single steps). This means, the linear number of distance (norm) computations cannot come at the price of exponentially high side-computations.

#### IV. RESULT

Now we prove that an algorithm that satisfies all the above requirements indeed exists. Before presenting the result formally, let us note that for number representation and operation we adopt the *unit cost model*: each number is considered a single data unit and each elementary number operation (addition, subtraction, multiplication, division and comparison) is counted as a single step. That is, the bit length of numbers is not considered.

**Theorem 1.** Let  $B$  be a fixed bounded subset in a finite dimensional normed space and let  $x_1, \dots, x_n \in S$  be arbitrary input vectors. Assume that a subroutine is available to compute the norm of any vector in the space. Then there exists a polynomial-time algorithm that finds a constant factor approximation of all the pairwise distances between the input vectors, such that the algorithm requires only  $O(n)$  norm computations. Moreover, the approximation factor can be made arbitrarily close to 1.

**Proof.** Let  $S$  be a finite dimensional normed space with norm  $l \cdot l$ . Let  $U$  be the closed unit ball in this space, that is,  $U = \{u \in S : \|u\| \leq 1\}$ . Fix a real number  $\alpha$  with  $0 < \alpha < 1$ . First we show that one can fix  $k$  vectors  $z_1, \dots, z_k \in S$  for some constant  $k$ , such that

$$\alpha \|x - y\| \leq \max_i \| \|x - z_i\| - \|y - z_i\| \| \leq \|x - y\| \quad (2)$$

holds for every  $x, y \in U$ .

For the proof, let us define a new normed space on  $S^2 = S \times S$ . The operations on  $S^2$  are defined as  $c(x, y) = (cx, cy)$  and  $(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$ , and let us introduce the norm  $\|(x, y)\| = \|x\| + \|y\|$  on  $S^2$ . One can directly check that this way again a finite dimensional normed vector space

is obtained. Now fix a real number  $0 < \alpha < 1$  and for each  $z \in S$  define the set  $A_z \subset S^2$  by

$$A_z = \left\{ (x, y) \in S^2 : x \neq y, \frac{\|y - z\| - \|x - z\|}{\|x - y\|} > \alpha \right\}$$

Furthermore, let  $H \subset S^2$  be the following set.

$$H = \{(x, y) : \|x\| \leq 2, \|y\| \leq 1, \|x - y\| \geq 1/2\}$$

We will use the following properties of these sets:

- 1)  $H$  is closed and bounded;
- 2)  $A_z$  is open;
- 3)  $H \subset \bigcup_{z \in S} A_z$ .

Clearly, 1) and 2) follow from the definition. To see 3) it is enough to observe that for any  $x \neq y \in S(x, y) \in A_x$  must hold, since with  $x = z$  we have

$$\frac{\|y - z\| - \|x - z\|}{\|x - y\|} = \frac{\|y - z\| - 0}{\|x - y\|} = 1 > \alpha$$

Thus the sets  $\{A_z, z \in S\}$  form an open cover of  $H$ . Given that the space is finite dimensional, it follows from the well-known Heine-Borel theorem that there exists a *finite* subcover, that is, there are points  $z_1, \dots, z_k$  with  $H \subset \bigcup_{i=1}^k A_{z_i}$  means that for any  $(x, y) \in H$ .

$$\frac{\|y - z_j\| - \|x - z_j\|}{\|x - y\|} > \alpha \quad (3)$$

holds for some  $1 \leq j \leq k$ .

We now show that (3) also holds for any  $x, y \in U$ ,  $x \neq y$  If  $\|x - y\| \geq 1/2$ , then  $(x, y) \in H$ , so then we already know it. Now take two points  $x, y \in U$  at distance  $\lambda$  apart, for some  $0 < \lambda < 1/2$ . That is  $\|x\| \leq 1$ ,  $\|y\| \leq 1$ , and  $0 \leq \|x - y\| = \lambda < 1/2$ . Set  $x' = \frac{1}{\lambda}(x - (1 - \lambda)y)$  Then  $x = \lambda x' + (1 - \lambda)y$  holds, that is,  $x$  divides the line segment  $\overline{x'y}$  such that  $\|x - y\| = \lambda \|x' - y\|$ . As we have chosen  $\|x - y\| = \lambda$ , it implies  $\|x' - y\| = 1$ . This, together with  $y \in U$  and by the triangle inequality, gives  $2 \geq \|x' - y\| + \|y\| \geq \|x'\|$ , yielding  $\|x'\| \leq 2$ . Collecting these facts, they imply  $(x', y) \in H$ . But then by (3) there is a  $z_j$  with

$$\frac{\|y - z_j\| - \|x' - z_j\|}{\|x' - y\|} > \alpha \quad (4)$$

Now using  $x - z_j = \lambda(x' - z_j) + (1 - \lambda)(y - z_j)$ , as well as the fact that the norm axioms imply that every norm is a convex function, we can write

$$\frac{\|y - z_j\| - \|x - z_j\|}{\|x - y\|} \geq$$

$$\frac{\|y - z_j\| - (\lambda \|x' - z_j\| + (1 - \lambda) \|y - z_j\|)}{\lambda} = \|y - z_j\| - \|x' - z_j\| = \frac{\|y - z_j\| - \|x' - z_j\|}{\|x' - y\|} > \alpha \quad (5)$$

Thus, for any  $x, y \in U$  there must be a  $j \in \{1, \dots, k\}$  such that

$$\| \|x - z_j\| - \|y - z_j\| \| \geq \alpha \|x - y\| \quad (6)$$

holds. For  $\|x - y\| \geq 1/2$  this follows from (3) and the case  $\|x - y\| < 1/2$  is covered by the analysis resulting in (5). Since the triangle inequality implies  $\| \|x - z\| - \|y - z\| \| \leq \|x - y\|$  for every  $x, y, z$ , therefore, combining this with (6), we obtain that (2) holds for every  $x, y \in U$ .

Now, let  $B$  be the bounded subset in the statement of the theorem and let  $R = \sup_{x \in B} \|x\| < \infty$ . Then  $x, y \in B$  implies  $\frac{1}{R}x, \frac{1}{R}y \in U$ , so we can apply (2) to  $\frac{1}{R}x, \frac{1}{R}y$ . Multiplying all sides in (2) by  $R$  and replacing each  $z_i$  by  $z'_i = Rz_i$ , we obtain that

$$\alpha \|x - y\| \leq \max_i \| \|x - z'_i\| - \|y - z'_i\| \| \leq \|x - y\| \quad (7)$$

holds for every  $x, y \in B$ .

Thus the value

$$d(x, y) = \max_i \| \|x - z'_i\| - \|y - z'_i\| \| \quad (8)$$

satisfies the requirements of the approximate distance (1) we are looking for, with  $\beta = 1$ . Since  $\alpha$  can be chosen to be arbitrarily close to 1, the difference  $\beta - \alpha = 1 - \alpha$  can be made arbitrarily close to 0, as required. Finally, we also observe from the formula that in order to compute  $d(x_i, x_j)$  for all  $i, j$  it is enough to know the norms  $\|x_i - z'_j\|$  for every  $i, j$ .

Since there are  $k$  vectors  $z'_j$  and for a fixed set  $B$  the value of  $k$  is constant (independent of  $n$ ), therefore, we need only constantly many norm computations for each  $x_i$ , which is altogether  $O(n)$  norm computations, as desired. The other side-computations can clearly be done in polynomial time, thus completing the proof.

## V. DISCUSSION AND OPEN PROBLEMS

A key feature of our result is that it proves only the *existence* of an algorithm with all the desired properties. On the other hand, the proof is not *constructive*, that is, it does not provide an effective method to actually construct the algorithm. In other words, we did not specify in the proof how to effectively find the required  $z'_1, \dots, z'_k$  vectors that

allow the computation of the approximate distance, according to formula (8). Nevertheless, we have proved that these vectors *exist*, so the algorithm with the desired properties must also exist.

A next natural research issue (which we plan to address in a future paper) is how to efficiently find the needed vectors  $z'_1, \dots, z'_k$ , so that the algorithm can be practically implemented. Regarding this issue, we present the following conjecture.

**Conjecture 1:** In a  $d$ -dimensional normed space, let us choose  $d + 1$  vectors independently at random from a continuous probability distribution, and let us use them in the approximate distance formula (8), as the  $z'$  vectors. If the input comes from a continuous probability distribution, then the algorithm (as described in the proof of Theorem 1) will work correctly, with probability approaching 1.

## VI. CONCLUSION

We have presented a method for the optimization of massive distance computations. We have proved that the quadratic all pairs distance computation in any normed space can be carried out with a *linear* number of distance computations. This can provide significant speed-up in those computationally intensive pattern recognition tasks that require many sophisticated distance computations. Let us note that while the rigorous proof of Theorem 1 is not simple, the result is more practical than it appears. Specifically, if Conjecture 1 is true (for which there is good chance), then it results in a quick and practical way of constructing an algorithm that can compute the  $O(n^2)$  all pairs distances of  $n$  points (data objects) in an arbitrary normed space, using only a *linear* number of distance computations, without excessive hidden costs in the side computations.

## REFERENCES

- [1] W. J. Rucklidge, "Efficiently locating objects using the hausdorff distance," *International Journal of Computer Vision*, vol. 24, issue 3, pp. 251–270, 1997.
- [2] X. Yi and O. I. Camps, "Line feature-based recognition using hausdorff distance," *International Symposium on Computer Vision*, 1995.
- [3] J. Bourgain, "On lipschitz embedding of finite metric spaces in hilbert space," *Israel J. Math.*, vol. 52, pp. 46–52, 1985.
- [4] N. Linial, E. London, and Y. Rabinovich, "The geometry of graphs and some of its algorithmic applications," *Combinatorica*, vol. 15, pp. 215–245, 1995.
- [5] J. Matoušek, "Note on Bi-lipschitz embeddings into low dimensional euclidean spaces", *Comment. Math. Univ. Carolinae*, vol. 31, pp. 589–600, 1990.
- [6] Abraham, Y. Bartal and O. Neiman, "Advances in metric embedding theory," in *Proc. 38<sup>th</sup> Annual ACM Symp. on Theory of Computing*, Seattle, WA, USA, May 2006, pp. 271–286.
- [7] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemp. Math.*, vol. 26, pp. 189–206, 1984.
- [8] P. Indyk and J. Matoušek, "Low distortion embeddings of finite metric spaces," in *Handbook of Discrete and Combinatorial Geometry*, J. E. Goodman and J. O'Rourke, Eds. Chapman and Hall/CRC, 2004, pp. 177–196.
- [9] N. Alon, "Problems and results in extremal combinatorics," *Discrete Math.*, vol. 273, pp. 31–53, 2003.
- [10] J. Matoušek, *Lectures on Discrete Geometry*, New York: Springer-Verlag, 2002.
- [11] B. Brinkman and M. Charikar, "On the impossibility of dimension reduction in  $l_1$ ," *Journal of the ACM*, vol. 52, pp. 766–788, 2005.

- [12] J. Matoušek, "On the distortion required for embedding finite metric spaces into normed spaces," *Israel J. Math.*, vol. 93, pp. 333–344, 1996.
- [13] P. Indyk, "Algorithmic applications of low-distortion geometric embeddings," in *Proc. 42<sup>nd</sup> Annual IEEE Symp. on Foundations of Computer Science*, Las Vegas, NV, October 2001, pp. 10–33.
- [14] A. Gupta, R. Krauthgamer, and J. R. Lee, "Bounded geometries, fractals, and low-distortion embeddings," in *Proc. 44<sup>th</sup> Annual IEEE Symp. on Foundations of Computer Science*, Cambridge, MA, USA, October 2003, pp. 534–543.
- [15] P. Assouad, "Plongements lipschitziens dans  $R^n$ ," *Bull. Soc. Math.*, vol. 111, pp. 429–448, France, 1983.
- [16] S. Semmes, "On the nonexistence bilipschitz parametrizations and geometric problems about  $a_\infty$  weights," *Revista Matemática Iberoamericana*, vol. 12, pp. 337–410, 1996.
- [17] L. A. Gottlieb and R. Krauthgamer, "A nonlinear approach to dimension reduction," Cornell University Library, April 2010.
- [18] I. Abraham, Y. Bartal, and O. Neiman, "Embedding metric spaces in their intrinsic dimension", in *Proc. 19<sup>th</sup> Annual ACM-SIAM Symp. on Discrete Algorithms*, San Francisco, CA, USA, January 2008, pp. 363–372.
- [19] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.



**Andras Farago** received the B.Sc., M.Sc. and Ph.D. degrees in electrical engineering from the Technical University of Budapest, Budapest, Hungary, in 1976, 1979, and 1981, respectively. Until 1997, he was a faculty member at the Department of Telecommunications and Telematics, Technical University of Budapest. In 1996 he obtained the distinguished title "Doctor of Sciences" from the Hungarian Academy of Sciences. In 1998 he moved to the USA, and became a professor of computer science at the University of Texas at Dallas. He also worked as a visiting senior research fellow at the University of Massachusetts at Amherst in 1991/92, and spent a sabbatical year at Boston University in 1996. He is a Senior Member of IEEE, member of ACM, and of the IFIP Working Group 6.3 "Performance of Communication Systems." He serves as editor for the journal *Wireless Networks*. His research focuses on modeling, design and analysis methods of communication networks, as well as on algorithms for various application fields. He authored over 200 research publications.