EFFICIENT SEARCH SUPPORTING SEVERAL SIMILARITY QUERIES BY REORDERING PIVOTS

Raisa Socorro
Instituto Superior Politécnico José Antonio Echevarría
La Habana. Cuba
raisa@ceis.cujae.edu.cu

Luisa Micó and José Oncina
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
P.O. box 99. E-03080 Alicante. España
mico@dlsi.ua.es, oncina@dlsi.ua.es

ABSTRACT
Effective similarity search indexing in general metric spaces has traditionally received special attention in several areas of interest like pattern recognition, computer vision or information retrieval. A typical method is based on the use of a distance as a dissimilarity function (not restricting to Euclidean distance) where the main objective is to speed up the search of the most similar object in a database by minimising the number of distance computations. Several types of search can be defined, being the *k-nearest neighbour* or the *range search* the most common.

AESA is one of the most well known of such algorithms due to its performance (measured in distance computations). *PiAESA* is an *AESA* variant where the main objective has changed. Instead of trying to find the best nearest neighbour candidate at each step, it tries to find the object that contributes the most to have a bigger lower bound function, that is, a better estimation of the distance.

In this paper we extend and test *PiAESA* to support several similarity queries. Our empirical results show that this approach obtains a significant improvement in performance when comparing with competing algorithms.

KEY WORDS
k-nearest neighbour, approximation, elimination, metric spaces, pivot, range search.

1 Introduction

Pattern recognition [1], image retrieval [2], or multimedia databases [3] are some examples of fields where methods based on similarity search are having an increasing interest due to its simplicity and adaptability to work with complex objects. Of particular importance is the most general approach to similarity search when it is modelled in metric spaces. Accordingly, the metric indexing techniques can be applied to many search problems, allowing different forms of complex queries. Moreover, this generality allows that the functionality of the techniques can be modified or increased.

The aim of many fast search algorithms is to reduce the search time, by reducing the number of distance computations. This is specially interesting when the computation of the distance is particularly expensive. Some examples are the context shape distance [4], distances between histograms [5], edit distance between strings [6], trees [7] or graphs [8], etc. In order to achieve such objective, it is usual to exploit some type of restricting property that the similarity measure should meet, being the triangular inequality the most popular and effective.

One of the most cited algorithms in this general context is the Approximating and Eliminating Search Algorithm *(AESA)*, introduced by E. Vidal in 1986 [9]. This algorithm is classified as a pivot-based metric space search algorithm in some taxonomies [10][11]. A pivot-based technique is a method that uses a subset of objects in the database for speeding up the search. Usually, the distances between the pivots and the rest (or some of them) of the points in the database are stored in preprocess time and used during the search. *AESA*, originally defined for searching the Nearest Neighbour (*NN*), works by iterating two steps: first it searches (heuristically) for a candidate to nearest neighbour (the approximating step), and then it uses this candidate to discard all the objects in the database than can not be nearest than the current candidate to *NN* (the elimination step). To carry out both steps, the approximating and eliminating steps, a lower bound of the distance function is used as an approximation of the true distance. This lower bound function is updated as new information becomes available. *AESA* focuses on searching good *NN* candidates, the effectivity of the elimination step is just a consequence of that.

The main drawback of *AESA* is its quadratic space complexity with respect to the size of the database. In fact, the space complexity, becomes a bottleneck when the algorithm is applied to large databases. Consequently, several solutions have been proposed in the last years to weaken this problem, for example, splitting the database [12], reducing and selecting the stored distances, [13], [14], etc. Despite the storage requirements of *AESA* can be very high, there are applications for which *AESA* is a feasible solution. Moreover, some authors have focused their work in reducing furthermore the number of distance computations of this algorithm ([15]) but at the expense of a significant increase in searching time.

PiAESA is a recently proposed fast *NN* search algorithm. This algorithm reduces significantly the number of distance computations with respect to *AESA* without increasing the overhead of the search [16][1]. The idea behind the algorithm consists on changing the focus with respect to *AESA*. On the first iterations it searches for the pivots that contribute the most to have an accurate lower bound function (but they can be bad candidates to *NN*), switching to the usual *AESA* behaviour when the lower bound function is precise enough.

In this work we extend the *PiAESA* to other types of similarity queries: k-Nearest Neighbours (*kNN*) and *range search*. We also compare experimentally the effectiveness of our extensions with *AESA* and other state of the art techniques.

2 The algorithm

Given a new object q (*query*) and a database T, in each step, *AESA* (Approximating Eliminating Search Algorithm) searches for a good candidate to *NN* avoiding the computation of as many distances as possible. Instead of searching the object $t \in T$ that minimises the distance by computing all the distances $d(q,t)$ for each $t \in T$, it uses a lower bound function $G(q,t) = \max_{p \in V} |d(q,p) - d(p,t)|$, where $V \subset T$ is the set of objects used as *NN* candidate in the previous iterations of the search. Since p and t are objects in the database, the distances $d(p,t)$ are computed in preprocess time and stored in a table. Each time a new candidate $p \in T$ is selected, the distance $d(q,p)$ is computed and then G is updated. Then, applying the triangle inequality, all the objects $t \in T$ whose lower bound distance to the query ($G(q,t)$) is larger than the distance to the current nearest neighbour candidate (d_{min}) can be eliminated of the search. On the other hand, the object $t \in T$ that minimizes the value of $G(q,t)$, is selected as new candidate in each step. The aim of this selection is to find a good candidate for: first, reduce d_{min} allowing the elimination of more objects in the database and second, update $G(q,t)$ $\forall t \in T$. Note that in the earlier steps of the search V is very small, and then, the lower bounds are a very bad estimates of the true distance d. This behaviour worsens when the dimensionality increases.

In [16] *PiAESA* (see Algorithm 1) was proposed. The idea is to focus on selecting objects in the database that are going to contribute increasing the most the lower bound function. When no further increases are expected the algorithm switches to the usual *AESA* behaviour: focus on obtaining good *NN* candidates. In *PiAESA* the objects of the database are sorted in a list, in preprocess, by their expected contribution to increase the lower bound functions. This list of pivots (P in the algorithm) is used by the search algorithm. A parameter R is used to assess if the lower bound is a good estimation of the distance. This parameter measures the number of successive iterations the best candidate to *NN* has not changed. If the *NN* candidate does not change in a large number of iterations that means the lower bound has stabilized and no further improvements are expected by increasing the set V. In this moment the algorithm changes its strategy and looks for good candidates to *NN*. Obviously, the switch can be also triggered if the list of pivots is exhausted. This can happen if the list of pivots P does not cover the full database.

Algorithm 1: PiAESA-1NN

```
Input: $T$: training set;
       $q$: query;
       $P \subset T$: list of selected ordered pivots;
       $R \in \mathbb{N}$: parameter to control the switch
            of the approximation criterion;
Output: $p_{min} \in T$: nearest neighbour
        $d_{min}$: distance to the nearest neighbour;

1  for $t \in T$ do $G(q,t) = 0$;
2  $i = 0$;
3  while ($P \neq \emptyset$ and $i < R$) do
4      approximating step selecting the next pivot $s$
       in $P$; $T = T - \{s\}$;
5      update the nearest neighbour $p_{min}$; //new NN
6      for $t \in T$ do update $G(q,t)$;
7      $min$ = argmin $G(q,t)$;
8      $i = i + 1$;
9      if $min > min_{prev}$ then $i = 0$; //G has
       changed
10 end
11 while $T \neq \emptyset$ do
12     $s = argmin_{t \in T} G(q,t)$; $T = T - \{s\}$;
13     update the nearest neighbour $p_{min}$;
14     foreach $t \in T$ do
15         update $G(q,t)$;
16         if $G(q,t) \geq d_{min}$ then $T = T - \{t\}$;
17     end
18 end
```

Note that when $R = 0$ the algorithm is exactly the *AESA*, since the first loop (lines 3 to 10) is skipped.

We are going to study several ways of building the list of pivots P. The use of pivot selection techniques in fast pivot-based search algorithms seems a good choice.

On the following we review some techniques:

- *Random Pivot Selection, RPS.*

  This is the straightforward technique where pivots are selected randomly.

- *Outliers Selection Techniques.*

  They refer to incremental selection methods locating objects far away from each other and to the rest of objects. Starting with a randomly selected pivot (p_1), two strategies are commonly used to select the next pivot [13].

  - *Maximum of Minimum Distances, MMD*
    $$p_i = \text{argmax}_{s \in (T-P_i)} \min_{j=1}^{i-1} d(s, p_j)$$

[1]Note to the referees: this paper is not publicly available. It was sent some time ago to a journal and now is still under review.

  - *Maximum Sum of Distances, MSD*
    $p_i = \text{argmax}_{s \in (T - P_i)} \sum_{j=1}^{i-1} d(s, p_j)$

    where T is the training set and $P_i = \{p_1, \ldots, p_i\}$. The list of pivots is then, $P = (p_1, \ldots, p_{|T|})$.

    Note that all the objects in the database can appear in the pivot list (like in the *RPS* method), that means the algorithm is just an ordering of the objects in the database.

- *Sparse Spatial Selection, SSS.*

  This method [17] dynamically selects a set of pivots whose distance to any already selected pivot is greater than a percentage of the maximum distance from the database. If two objects do not fulfil the previous condition, one of them is eliminated from the pivot list and then it can not be an enumeration of all the objects in the database.

- *Dynamic Pivot Selection, DPS.*

  This technique, proposed in [18], is a dynamic extension of the *SSS* method where the deletion of pivots is allowed if it can be proved a pivot becomes redundant. As in the previous method, it can not be an enumeration of all the objects in the database. This is the reason why a second condition was added in the new approach to switch the strategy (see line 3 in Algorithm 1) when the list of pivots was empty.

In this work we use several types of queries. In next sections, we introduce the major modifications to be made in the *Pi-AESA-1NN* algorithm to apply them.

2.1 Extension to k-nearest neighbour search

To adapt the *PiAESA-1NN* to a *kNN* search strategy, it is necessary to change two main elements in the algorithm:

- Output: the k nearest objects to the query and their distances should be recovered (instead of p_{min} and d_{min})
- the elimination criterion in line 16 should be changed by $G(t) \geq d_{kNN}$, where d_{kNN} is the distance to the k nearest neighbour.

2.2 Extension to range search

To adapt *PiAESA-1NN* to a *range search*, a specific value of the distance from the query is defined (distance to the query) and it is necessary to change two main elements in the algorithm:

- Output: given a value for the range (radius), r, all the objects whose distance to the query is lower than r are recovered
- the elimination criterion in line 16 should be changed by $G(t) \geq r$

3 Experimental results

The primary purpose of this work is to study the extendibility of *PiAESA-1NN* when the *range search* or the *kNN* search is used. In order to check that, we evaluate these techniques in several metric spaces, such as synthetic and real vectors, and a string database. In all the experiments the distance table is stored in main memory.

In this work we have experimented with artificial and real data to check the performance of *PiAESA* when it is extended to different types of search. Moreover, we have compared our proposal with other algorithms competing *AESA* (computing less distances or saving space).

The used datasets are:

- Data extracted from a uniform distribution in the unit hypercube with database sizes ranging from 500 to 15 000 and dimensionality from 2 to 24.
- Real image databases with vectorial representation (*NASA* represented by features vector of 20 components, and *COLORS* represented by vectors of 112 components, both can be found in `http://www.sisap.com`)
- Contour strings from the *MNIST* database, a collection of 60 000 images of handwriting digits (`http://yann.lecun.com/exdb/mnist/`).

In this work, and for all the collections, a subset of 15 000 objects were used for training and 1 000 for testing. Moreover, for databases with a vectorial representation of the data, the Minkowsky L_1 distance were used. We have used the edit distance for the *MNIST* database, where the strings extracted represent the contour of the images.

3.1 Analysis of the parameter R using the *kNN* search strategy

As mentioned in section 2, in our approach we define a parameter (R) that lets to decide when to switch the approximation strategy. Our first experiment aims at comparing the behaviour of the pivot selecting technique *MMD* for *PiAESA-kNN* with some databases. The performance is shown in Fig. 1 for a 10 and 20-dimensional space in the unit hypercube, and Fig. 2 where two real datasets were evaluated.

These figures show that there exists a value of the parameter R for which the average number of distances is minimum (the behaviour is similar independently of the pivot selecting techniques that is used). This result confirms that the achievement of our first objective (to obtain a good estimation of the lower bound function) is fulfilled when the minimum number of distances is computed. Then, if we use a higher value of R, marginal improvements will be obtained in the function, losing the opportunity to update the solution.

The optimum value of R was obtained for every dataset and used in the remaining experiments.

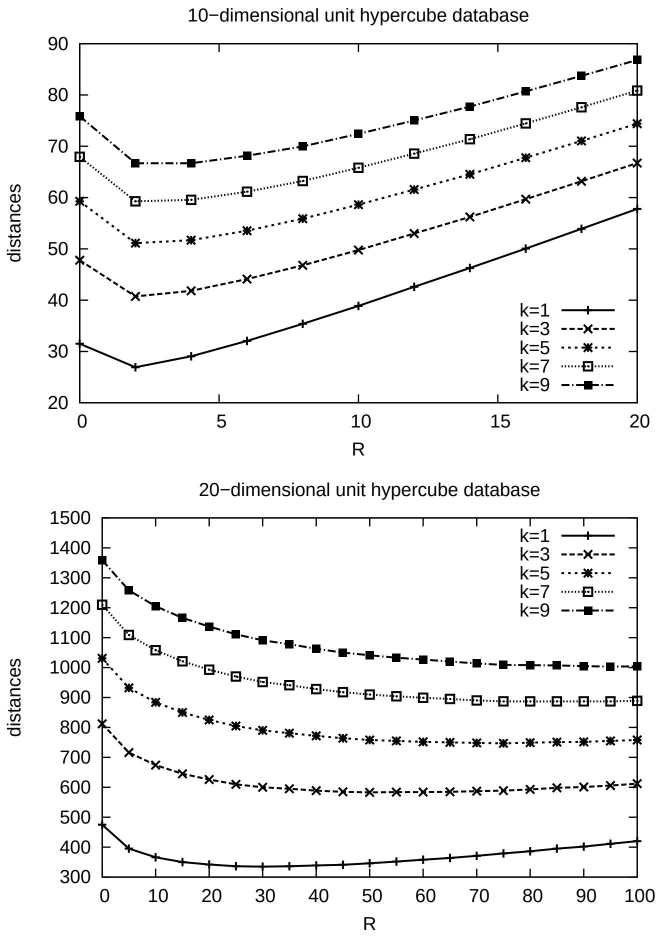

Figure 1. Average number of distance computations for increasing values of R. A $15\,000$ points database training set, L_1 Minkowski distance and the *MMD* pivot selection technique were used.

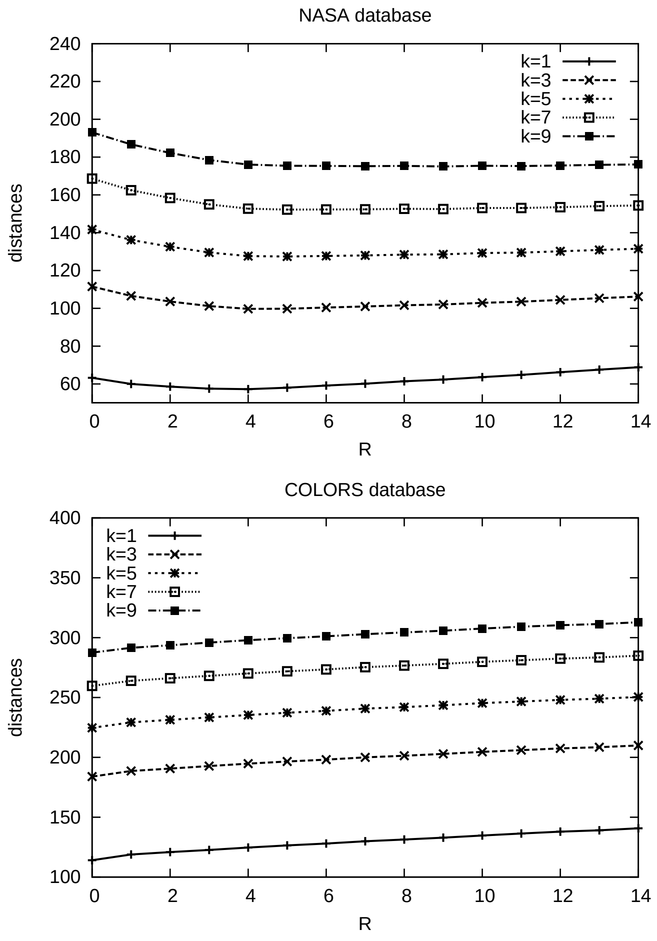

Figure 2. Average number of distance computations for increasing values of R. A $15\,000$ points database training set, L_1 Minkowski distance and the *MMD* pivot selection technique were used.

It can be shown that *PiAESA* always outperform the *AESA* algorithm (when $R = 0$) for some value $R > 0$, and this result does not depend on the type of search (even *range search* is included). Although it seems that with database *COLORS* the algorithm does not improve *AESA* results (see Fig.2), if a pivot selection technique is appropriately chosen, the result can be improved. For example, for this database, *PiAESA* improves *AESA* if the *SSS* pivot technique is used.

3.2 Analysis of the parameter R using different pivot techniques

In a second set of experiments, we have evaluated the behaviour of *PiAESA* when the pivot selection techniques described in section 2 were used to make the approximation step in the earlier steps of the search. Both, experiments with *kNN* and *range search* were made. Results for *kNN* search in Fig 3 with $k = 1$ (on the left) and $k = 9$ (on the right) confirm that the optimum value of R increases with k. Moreover, the different pivot methods have the same behaviour regardless the value of k. One can view also in Fig. 3 that the performance of *PiAESA* with *SSS* and *DPS* pivot selection techniques are independent of R when $R > 5$ in dimension 10 and $R > 25$ in dimension 20, due to the number of pivots is fixed and lower than the training set size, ie, the condition $P = \emptyset$ is fulfilled in the algorithm (line 3). Results for the *range search* in Fig 4 confirm that the optimum value of R increases with r.

These results show that also when k is larger than 1, *PiAESA-kNN* always outperforms *AESA* algorithm for some pivot techniques (when comparing the average number of distance computations). It must be noted that these results have been obtained without any extra computational cost in search time as the pivots are ordered in preprocessing time.

Similar results have been obtained when the *range search* is applied. In Fig.4 we show two experiments applying *range search* in a 18-dimensional space in the unit hypercube for radius 1 and 4. In this case, *MMD* and *MSD*

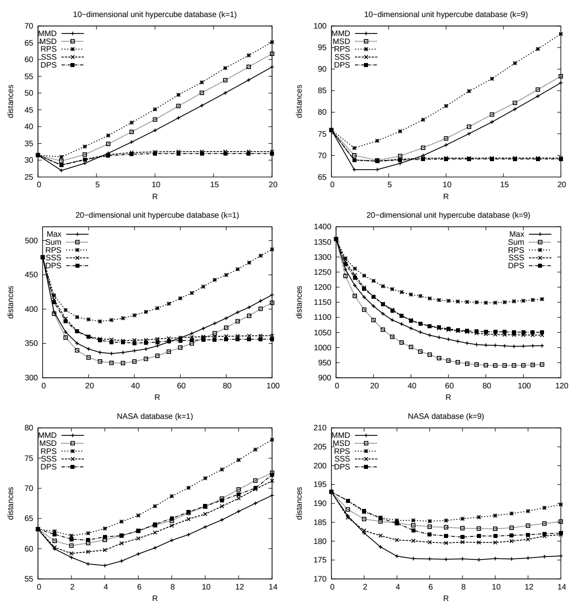

Figure 3. Average number of distance computations for increasing values of R. A $15\,000$ points database training set and L_1 Minkowski distance were used.

obtain the best results, and *MSD* outperforms *MMD* when the value of the range increases, as with increasing dimensionality.

3.3 Comparison with other methods

The performances of *PiAESA* and other state of the art algorithms (*AESA*, *iAESA* and *LAESA*) for *kNN* and *range search* are compared in Table 1 and Table 2 respectively. Usually the performance of these algorithms when answering both range and *kNN* queries worsens as the dimension of the space grows. We have designed some experiments to study these settings.

As expected, increasing the dimension of the data makes the problem more difficult. However *PiAESA* performs consistently (up to 47% in dimension 24) better than *AESA*. One can view that this reduction increases with the dimensionality (in dimension 12 only a 15% of distances were saved). Moreover, in the *kNN* case, this reduction also increases with k (up to dimension 20). This is an encouraging feature for the extensibility of the method to other types of search.

Moreover, it can be seen than *MSD* method outperforms *MMD* both when increases the dimensionality of the space and the radius used in *range search*(see Table 2).

Tables 3 and 4 show the results of the experiment when the objects belong to real databases (*COLORS*, *NASA* and *MNIST*). Tables show that *PiAESA* has the best performance, with slight improvements. Moreover, it was significant that the best results using the *MNIST* database were obtained using the random technique *RPS*.

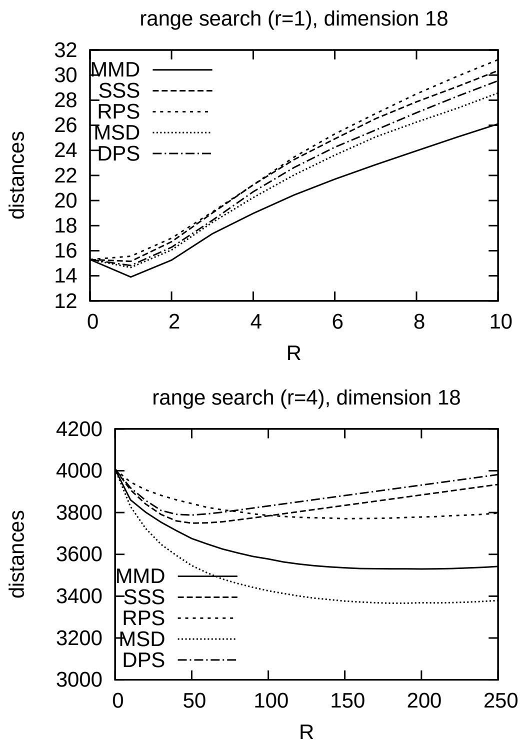

Figure 4. Average number of distance computations for increasing values of R. A $15\,000$ points database training set and L_1 Minkowski distance in a 18-dimensional space were used.

4 Conclusions

In this work we have analized experimentally the behaviour of the *PiAESA* algorithm for several similarity queries. The experimental evaluation shows that *PiAESA* outperforms *AESA* and other state of the art fast methods independently of the type of search. Moreover, this approach improves significantly their behaviour with the dimensionality: saving 15% of distance computations in a 12-dimensional space up to 46% of distance computations in a 24-dimensional space using the nearest neighbour search. This improvement is even bigger with *kNN* and *range* searches up to 20-dimensional spaces.

The disparity of results depending on the pivot selection technique suggests that there is room for improvement in this point. As a consequence, we are interested in exploring other pivot selection techniques studying how the use of different set of pivots can affect the behaviour of our approach.

Acknowledgements.

The authors thank the Spanish CICyT for partial support of this work through projects TIN2009-14205-C04-01, the IST Programme of the European Community, under the PASCAL Network of Excellence, IST–2002-506778, and

Table 1. The three first columns represent the average number of distances (nd) computed by some state of the art algorithms for different values of k with unit hypercube database and different dimensions. The two last columns represent the improvement (in %) of *PiAESA* versus *AESA* (Pi/A) and versus *iAESA* (Pi/iA). The third column shows in brackets the optimum value of R or each k. The pivot selection technique used in *PiAESA* was *MSD*.

	dimension 12				
k	**AESA** nd	**iAESA** nd	**PiAESA** nd	**Pi–A** % impr.	**Pi–iA** % impr.
1	54.3	49.9	**47.0(5)**	15.5	6.2
3	83.9	76.2	**70.2(5)**	19.5	8.5
5	105.6	95.7	**88.4(5)**	19.5	8.7
7	124.1	112.2	**103.9(5)**	19.4	8
9	141.1	127.6	**118.2(5)**	19.4	8
	dimension 18				
k	**AESA** nd.	**iAESA** nd	**PiAESA** nd	**Pi–A** % impr.	**Pi–iA** % impr.
1	274.9	231.7	**200.6(20)**	37.0	15.5
3	462.9	401.7	**332.7(30)**	39.1	20.7
5	588.3	518.6	**423.9(40)**	38.8	22.3
7	691.2	616.6	**496.9(50)**	39.1	24.1
9	780.6	701.0	**561.5(50)**	39.0	24.8
	dimension 24				
k	**AESA** nd	**iAESA** nd	**PiAESA** nd	**Pi–A** % impr.	**Pi–iA** % impr.
1	1277.8	1120.2	**869.7(85)**	46.9	28.8
3	2103.8	1901.8	**1486.0(145)**	41.6	28.0
5	2589.6	2364.9	**1880.4(180)**	37.7	25.8
7	2953.9	2717.2	**2190.0(200)**	34.9	24.1
9	3264.8	3003.2	**2462.3(220)**	32.6	22.0

the program CONSOLIDER INGENIO 2010 (CSD2007-00018).

References

[1] M. Potamias and V. Athitsos. Nearest neighbor search methods for handshape recognition. In *Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2008.

[2] S. Batiato, G. Di Blasi, and D. Reforgiato. Advanced indexing schema for imaging applications: three case studies. *Image Processing*, 1(3):249–268, 2007.

[3] T.S. Huang, C.K. Dagli, S. Rajaram, E.Y. Chang, M.I. Mandel, G.E. Poliner, and D.P.W. Ellis. Active learning for interactive multimedia retrieval. volume 96, pages 648–667, April 2008.

Table 2. Average number of distance computations by several state of the art algorithms for different values of the radius r using *range search* with a unit hypercube database and several dimensions. The third and fourth column show in brackets the optimum value of R for each r and technique.

	dimension 12			
r	**AESA**	**iAESA**	**PiAESA MSD**	**PiAESA MMD**
1	24.3	23.8	22.8(1)	**21.1(2)**
2	325.8	301.0	**288.9(15)**	293.8(10)
3	3776	3743	**3382(135)**	3497(125)
4	11727	11624	**11387(235)**	11510(240)
	dimension 18			
r	**AESA**	**iAESA**	**PiAESA MSD**	**PiAESA MMD**
1	15.3	15.3	14.7(1)	**13.9(1)**
2	73.1	62.4	59.6(5)	**55.5(5)**
3	584.2	510.7	**449.9(35)**	467.2(30)
4	4007	3887	**3365(185)**	3529(195)
	dimension 24			
r	**AESA**	**iAESA**	**PiAESA MSD**	**PiAESA MMD**
1	12.4	12.5	12.1(1)	**12.0(1)**
2	41.2	38.2	36.6(5)	**35.5(5)**
3	200.7	156.5	**136.8(10)**	138.8(10)
4	1168	1013	**821.0(75)**	874.0(70)

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[5] Sung-Hyuk Cha and Sargur N. Srihari. On measuring the distance between histograms. *Pattern Recognition*, 35:1355–1370, 2002.

[6] R. Wagner and M. Fisher. The string-to-string correction problem. *ACM*, 21:168–178, 1974.

[7] P. Bille. A survey on tree edit distance and related problems. *Theor. Comput. Sci.*, 337(1-3):217–239, 2005.

[8] X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis & Applications*, 13(1):113–129, 2010.

Table 3. Average number of distance computations by some state of the art algorithms for different values of k with databases *NASA* and *COLORS*. The third and fourth column show in brackets the optimum value of R for each k and technique.

	COLORS			
k	**AESA**	**iAESA**	**PiAESA MMD**	**PIAESA SSS**
1	114.0	122.4	114.0 (0)	**110.7 (3)**
3	183.9	193.8	183.9 (0)	**177.3 (5)**
5	224.7	235.7	224.7 (0)	**217.1 (5)**
7	259.7	271.6	259.7 (0)	**251.2 (6)**
9	287.5	299.6	287.5 (0)	**278.3 (8)**
	NASA			
k	**AESA**	**iAESA**	**PiAESA MMD**	**PIAESA SSS**
1	63.2	69.3	**57.2 (4)**	59.2 (2)
3	111.5	118.5	**99.7 (4)**	103.1 (4)
5	141.7	149.4	**127.4 (5)**	131.4 (4)
7	168.6	177.9	**152.2 (5)**	156.4 (7)
9	193.1	203.5	**175.1 (9)**	179.5 (7)

Table 4. Average number of distance computations by several state of the art algorithms for different values of k with database *MNIST*.

	MNIST				
k	**AESA**	**iAESA**	**PiAESA MMD**	**PiAESA SSS**	**PiAESA RPS**
1	801.1	833.9	794.8(2)	798.9(4)	**780.1(16)**
3	1115.7	1148.0	1109.2(2)	1113.7(4)	**1090.4(16)**
5	1292.3	1323.7	1286.1(2)	1290.8(4)	**1265.9(16)**
7	1433.3	1460.4	1426.1(2)	1431.8(4)	**1405.7(16)**
9	1541.2	1564.2	1534.6(2)	1550.1(4)	**1512.7(16)**

[9] E. Vidal. An algorithm for finding nearest neighbours in (approximately) constant average time. *Pattern Recognition Letters*, 4(3):145–157, 1986.

[10] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33(3):273–321, 2001.

[11] Gisli R. Hjaltason and Hanan Samet. Index-driven similarity search in metric spaces (survey article). *ACM Trans. Database Syst.*, 28(4):517–580, 2003.

[12] K. Fredriksson. Engineering efficient metric indexes. *Pattern Recognition Letters*, 28(1):75–84, 2007.

[13] L. Micó, J. Oncina, and E. Vidal. A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing time and memory requirements. *Pattern Recognition Letters*, 15(1):9–17, 1994.

[14] L. G. Ares, N. Brisaboa, M. F. Esteller, O. Pedreira, and A. S. Places. Optimal pivots to minimize the index size for metric access methods. In *SISAP '09: Proceedings of the 2009 Second International Workshop on Similarity Search and Applications*, pages 74–80, Washington, DC, USA, 2009. IEEE Computer Society.

[15] K. Figueroa, E. Chavez, G. Navarro, and R. Paredes. Speeding up spatial approximation search in metric spaces. *J. Exp. Algorithmics*, 14:3.6–3.21, 2009.

[16] R. Socorro, L. Micó, and J. Oncina. A fast pivot-based indexing algorithm for metric spaces. *Submitted to Pattern Recognition Letters, under revision*, 2010.

[17] N. R. Brisaboa, A. Farina, O. Pedreira, and N. Reyes. Similarity search using sparse pivots for efficient multimedia information retrieval. In *ISM '06: Proceedings of the Eighth IEEE International Symposium on Multimedia*, pages 881–888, Washington, DC, USA, 2006. IEEE Computer Society.

[18] B. Bustos and T. Skopal. Dynamic similarity search in multi-metric spaces. In *Proc. 8th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'06)*, pages 137–146. ACM Press, 2006.