Supermetric Search with the Four-Point Property

Richard Connor[1], Lucia Vadicamo[2], Franco Alberto Cardillo[3], and Fausto Rabitti[2]

[1] Department of Computer and Information Sciences,
University of Strathclyde, Glasgow, G1 1XH, United Kingdom
[2] Institute of Information Science and Technologies (ISTI)
CNR, Via Moruzzi 1, 56124 Pisa (Italy)
[3] Institute of Computational Linguistics (ILC)
CNR, Via Moruzzi 1, 56124 Pisa (Italy)
richard.connor@strath.ac.uk
{lucia.vadicamo,fausto.rabitti}@isti.cnr.it
francoalberto.cardillo@ilc.cnr.it

Abstract. Metric indexing research is concerned with the efficient evaluation of queries in metric spaces. In general, a large space of objects is arranged in such a way that, when a further object is presented as a query, those objects most similar to the query can be efficiently found. Most such mechanisms rely upon the triangle inequality property of the metric governing the space. The triangle inequality property is equivalent to a finite embedding property, which states that any three points of the space can be isometrically embedded in two-dimensional Euclidean space. In this paper, we examine a class of semimetric space which is finitely 4-embeddable in three-dimensional Euclidean space. In mathematics this property has been extensively studied and is generally known as the *four-point* property. All spaces with the four-point property are metric spaces, but they also have some stronger geometric guarantees. We coin the term *supermetric*[4] *space* as, in terms of metric search, they are significantly more tractable. We show some stronger geometric guarantees deriving from the four-point property which can be used in indexing to great effect, and show results for two of the SISAP benchmark searches that are substantially better than any previously published.

1 Introduction

To set the context, we are interested in searching a (large) finite set of objects S which is a subset of an infinite set U, where (U, d) is a metric space. The general requirement is to efficiently find members of S which are similar to an

[4] This term has previously been used in the domains of particle physics and evolutionary biology as a pseudonym for the mathematical term *ultra-metric*, a concept of no interest in metric search; we believe our concept is of sufficient importance to the domain to justify its reuse with a different meaning.

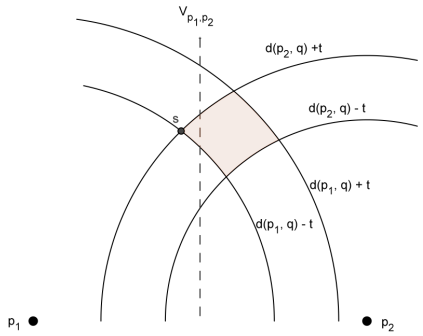

Fig. 1: In any metric space, two pivot points and any solution to a query can be isometrically embedded in ℓ_2^2. The point q cannot be drawn in the same diagram. Given its distance from p_1 and p_2, any solution in the original metric space must lie in the region bounded by the four arcs shown. If the point s lies to the right of V_{p_1,p_2}, there is therefore no requirement to search to the left of the hyperplane in the original space.

arbitrary member of U, where the distance function d gives the only way by which any two objects may be compared. There are many important practical examples captured by this mathematical framework, see for example [3, 8]. Such spaces are typically searched with reference to a query object $q \in U$. A threshold search for some threshold t, based on a query $q \in U$, has the solution set $\{s \in S \text{ such that } d(q, s) \leq t\}$.

1.1 Metric Spaces and Finite Isometric Embeddings

An isometric embedding of one metric space (V, d_v) in another (W, d_w) can be achieved when there exists a mapping function $f : V \to W$ such that $d_v(x, y) = d_w(f(x), f(y))$, for all $x, y \in V$. A finite isometric embedding occurs whenever this property is true for any finite selection of n points from V, in which case the terminology used is that V is isometrically n-embeddable in W.

The first observation to be made in this context is that any metric space is isometrically 3-embeddable in ℓ_2^2. This is apparent from the triangle inequality property of a proper metric. In fact the two properties are equivalent: for any semi-metric space (V, d_v) which is isometrically 3-embeddable in ℓ_2^2, triangle inequality also holds. It is interesting to consider the standard exclusion mechanisms of pivot-based exclusion and hyperplane-based exclusion in the light of an isometric 3-embedding in ℓ_2^2; Figure 1 for example shows a basis for hyperplane exclusion using only this property rather than triangle inequality explicitly.

1.2 Supermetric Spaces: Isometric 4-embedding in ℓ_2^3

It turns out that many useful metric spaces have a stronger property: they are isometrically 4-embeddable in ℓ_2^3. In the mathematical literature, this has been referred to as the *four-point property*. We have studied such spaces in the context of metric indexing in [4], where we develop in detail the following outcomes:

1. Any metric space which is isometrically embeddable in a Hilbert space has the four-point property.
2. Important spaces with the property include, for any dimension, spaces with the following metrics: Euclidean, Jensen-Shannon, Triangular, and (a variant of) Cosine distances.
3. Important spaces which do not have the property include those with the metrics: Manhattan, Chebyshev, and Levenshtein distances.
4. However, for any metric space (U, d), the space $(U, \sqrt{d})$ does have the four-point property.

In terms of practical impact on metric search, in [4] we show only how the four-point property can be used to improve standard hyperplane partitioning. We consider a situation where a subspace is divided according to which of two selected reference points p_1 and p_2 is the closer. When relying only on triangle inequality, that is in a metric space without the four-point property, then for a query q and a query threshold t, the subspace associated with p_1 can be excluded from the search only if $d(q, p_1) - d(q, p_2) > 2t$. As the region defined by this condition when projected onto the plane is a hyperbola (see Figure 1), we name this Hyperbolic Exclusion.

If the space in question has the four-point property, however, we show that, for the same subspaces, there is no requirement to search that associated with p_1 whenever $\frac{d(q,p_1)^2 - d(q,p_2)^2}{d(p_1,p_2)} > 2t$; this is a weaker condition and therefore allows, in general, more exclusion. We name this condition Hilbert Exclusion.

In this paper, we examine a more general consequence of four-point embeddable spaces and show some interim results including new best-performance search of SISAP data sets.

2 Tetrahedral Projection onto a Plane

In a supermetric space, any two reference points p_1 and p_2, and query point q, *and* any solution to that query s where $d(q, s) \leq t$, can *all* be embedded in 3D Euclidean space. As such, they can be used to form the vertices of a tetrahedron. It seems that, while simple metric search is based around the properties of a triangle, there should be corresponding tetrahedral properties which give a new, stronger, set of guarantees.

Assume that for some search context, points $p_1, p_2 \in U$ are somehow selected and a data structure is built for a finite set $S \subset U$ where, for $s \in S$, the three distances $d(p_1, p_2), d(s, p_1)$ and $d(s, p_2)$ are calculated during the build process and used to guide the structuring of the data. At query time, for a query q, the

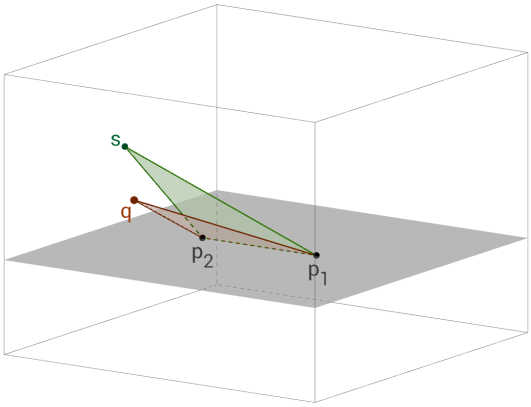

Fig. 2: Two triangles in 3D space

two distances $d(q, p_1)$ and $d(q, p_2)$ are calculated and may be used to make some deduction relating to this structure.

This situation gives knowledge of two adjacent faces of the tetrahedron which can be formed in three dimensions. Five of the six edge lengths have been measured, and the final edge is upper-bounded by the value of t. Therefore, for a point s to be a solution to the query, it must be possible to form a tetrahedron with the five measured edge lengths, and a last edge of length t.

Figure 2 shows a situation where five edge lengths have been embedded in 3D space. The edge p_1p_2 is shared between the two facial triangles depicted. However the distance $d(s, q)$ is not known, and therefore neither is the angle between these triangles. The observation which gives rise to the results presented here is that, if both triangles are now projected onto the same plane, which can be achieved by rotating one of them around the line p_1p_2 until it is coplanar with the other, then for any case where the final edge of the tetrahedron (qs) is less than the length t, then the length of this side in the resulting planar tetrahedron is upper bounded by t, as illustrated in Figure 3.

Many such coplanar triangles can be depicted, representing many points in a single space, in a single scatter plot as in Figure 4. This shows a set of 500 points, drawn from randomly generated 8-dimensional Euclidean space, and plotted with respect to their distances from two fixed reference points p_1 and p_2. The distance between the reference points is measured, and the reference points are plotted on the X-axis symmetrically either side of the origin. For each point in the rest of the set, the distances $d(s, p_1)$ and $d(s, p_2)$ are calculated, and used to plot the unique corresponding point in a triangle above the X-axis, according to these edge lengths. In this figure, in consideration with our observations over Figure 3, it can be seen that, if any two points are separated by less that some constant

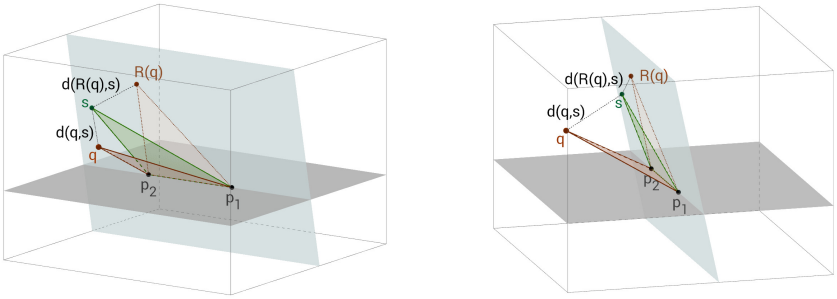

Fig. 3: Projection of the two triangles onto the same plane by rotation around p_1p_2. Note that $\ell_2^2(R(q), s) \leq \ell_2^3(q, s)$

t in the original space, and thus also in the 3D embedding, then they are also within t of each other in this scatter plot.

It is important to be aware, in this and the following figures, of the importance of the four-point property. The same diagram can of course be plotted for a simple metric space, but in this case no spatial relationship is implied between any two points plotted: no matter how close two points are in the plot, there is no implication for the distance between them in the original space. However if the diagram is plotted for a metric with the four-point property, then the distance between any two points on the plane is a lower bound on their distance in the original space; two points that are further than t on the plot cannot be within t of each other in the original space. This observation leads to an arbitrarily large number of ways of partitioning the space and allowing these partitions to excluded based on a query position, and has many potential uses in metric indexing.

3 Indexes Based on Tetrahedral/Planar Projection

During construction of an index, the constructed 2D space can be arbitrarily partitioned according to any rule based on the geometry of this plane, calculated with respect to the distances $d(s_i, p_1), d(s_i, p_2)$ and $d(p_1, p_2)$. At query time, if the query falls in any region of the plane that is further than the query threshold t from any such partition, points within that partition cannot contain any solution to the query. Since, as will be shown, different spaces give quite different distributions of points within the plane, build-time partitions can be chosen according to this distribution, rather than as a fixed attribute of an index mechanism.

There is much potential for investigating partitions of this plane, and our work is ongoing. The simplest such mechanism to consider is the application of this concept to normal hyperplane partitioning. Suppose that a data set S is simply divided according to which of the points p_1 and p_2 is the closer, which

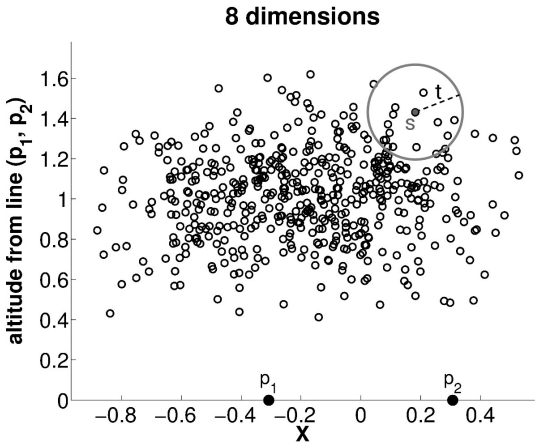

Fig. 4: Scatter diagram for 8-dimensional Euclidean Space. The distance δ between two selected reference points p_1 and p_2 is measured, and an embedding function is chosen which maps these to $(0, -\delta/2)$ and $(0, \delta/2)$ respectively. Other points s_i in the space are then plotted to preserve the distances $d(s_i, p_1)$ and $d(s_i, p_2)$. For metric spaces with the four-point property, the ℓ_2 distance between the corresponding points in this diagram is a lower bound on $d(s_i, s_j)$ in the original space. Hence, any point within t of a point s in the original space cannot lie outside the circle of radius t centered around s in the scatter plot.

corresponds in the scatter diagram to a split over the Y axis. Then at query time, if the corresponding plot position for the query is further than t from the Y axis, no solutions can exist in the subset closer to the opposing reference point. Figure 5 shows the same points, but now highlighted according to this distinction. Those drawn in solid, either side of the Y-axis, are guaranteed to be on the same side of the corresponding hyperplane partition in the original space; therefore, if they were query points, the opposing semi-space would not require to be searched. If the same diagram is drawn for a simple metric space, a query point can be used to exclude the opposing semi-space only according to a condition algebraically derived from triangle inequality: $|d(q, p_1) - d(q, p_2)| > 2t$, which describes a hyperbola with foci at the reference points and semi-major axis of the search threshold. For the same data and search threshold, the difference in exclusion capability is shown in Figure 5; of the 500 randomly selected queries, only 160 fail to exclude the opposing semi-space, whereas with normal hyperbolic exclusion, the number is 421. The query threshold illustrated, 0.145, is chosen to retrieve around one millionth of the space and is not therefore artificially large.

As stated, this particular situation has been extensively investigated and is fully reported in [4]. Here we will concentrate further on other properties of the planar projection, of which the derivation of Hilbert exclusion turns out to be a special case.

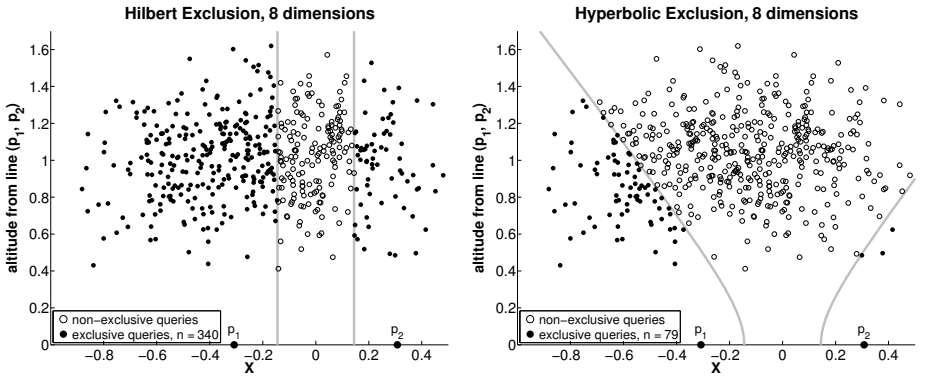

Fig. 5: Scatter diagram for 8-dimensional Euclidean Space. The data is divided into two subsets according to which side of Y-axis they lie; if the solidly-coloured points represent queries, the data on the opposing side cannot contain a solution. The left-hand side illustrates use of tetrahedral/planar projection, the right hand side illustrates use of the normal hyperbolic condition.

4 Partitions of the 2D Plane

For the purposes of this analysis only, for reasons of simplicity, we seek to divide a data set into precisely two partitions. This is without reference to details of any indexing structure which may use the concepts, although in all cases by implication there exists a simple binary partition tree structure corresponding to the partitioning. In all cases the partition is defined in terms of the 2D plane onto which all points are projected as described above. A few points are important to note for such structures:

1. For any such strategy, other more complex ways of indexing the data exist; by analogy, for example, various forms of SAT[2], GNAT[1], M-Index [7] etc. will exist. Mechanisms normally associated with single reference point pivoting may also have equivalents. In our initial analysis we do not have time or space to investigate all of these forms.
2. There is an apparent disadvantage for any of these techniques when compared with any technique based on single-point pivoting, which is that for any conceptual tree node, two distances need to be calculated as against one. This is not the case in fact, as it is always possible to re-use one reference point from the node directly above, without significantly affecting any spatial properties of the distribution, using a technique first proposed for the monotonous bisector tree [6].
3. Furthermore, any such mechanism has a further advantage, as whenever the space is partitioned, it is also possible to store internal and external radii for the partitions, from both of the reference points, which allow further exclusions to be made at effectively no extra cost.

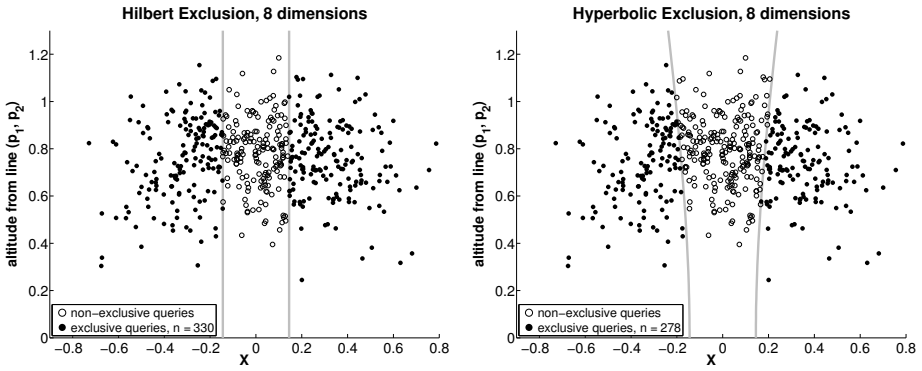

Fig. 6: Scatter diagram for 8-dimensional Euclidean Space with widely separated reference points. (The distance between reference points is such that the reference points themselves do not appear on the plot.)

4.1 Reference Point Separation

An important observation is that the shape of the 2D "point cloud", upon which effective exclusion depends, is not greatly affected by the choice of reference points. In comparison with normal Hyperbolic exclusion this is a huge advantage. The hyperbola which bounds the effective queries, i.e. those which can be used to exclude the opposing semispace, is defined only by the (fixed) query radius, and the distance between the reference points, where the larger the separation of the reference points, the better the exclusion. In the extreme case where the separation is no larger than twice the query radius, which can readily occur in high-dimensional space, it is impossible for any exclusions to be made. This effect can be ameliorated by choosing widely separated reference points, but in an unevenly distributed set this in itself can be dangerous: if one point chosen is an outlier, then the point cloud will lie close to the other point, and again no exclusions will be made. Finding two reference points which are well separated, and where the rest of the points is evenly distributed between them, is of course an intractable task in general.

Figures 6 and 7 show this effect. In these diagrams, the reference points have been selected as the furthest, and nearest, respectively out of 1,000 sample pairs of points drawn from the space. It can be seen that, when exclusion is based on tetrahedral properties allowed from the four-point property, the exclusive power remains fairly constant, as the size and shape of the point cloud is not greatly affected. However, when the hyperbolic condition is used, the exclusive power is hugely affected; in this case the query threshold is only slightly less than half the separation of the reference points, and the resulting hyperbola diverges so rapidly from the separating hyperplane that no exclusions are made from the sample queries. From Figure 6 it should also be noted that, no matter how far the reference points are separated, the four-point property always gives more exclusions; in this case, although the separating lines do not appear visually

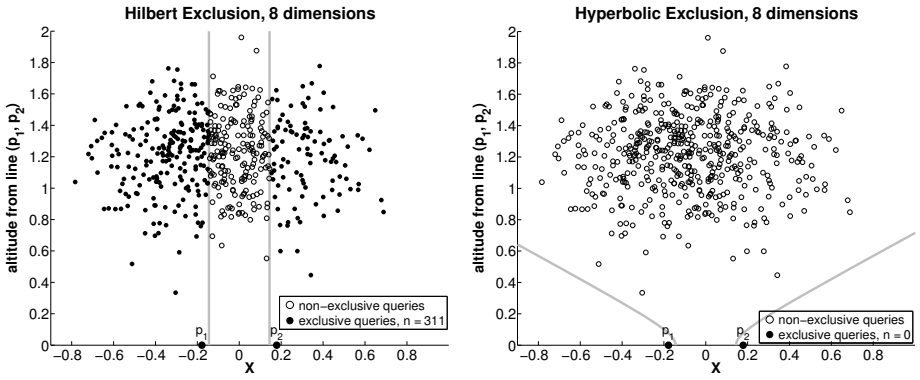

Fig. 7: Scatter diagram for 8-dimensional Euclidean Space with close reference points. Note from comparison of the left-hand graphs of this figure with Figure 6 that the separation of the reference points has no apparent effect on the power of the four-point exclusion, whereas normal metric exclusion becomes completely useless.

to be very different, the implied probability of exclusion in for the four-point property is 0.66, against 0.58.

To allow most partition structures to perform well, a very large part of the build cost is typically spent in the selection of good reference points and this cost is largely avoidable with any such four-point strategy.

4.2 Arbitrary Partitions

Again we stress the fact that, given the strong lower bound condition on the projected 2D plane, we can choose arbitrary geometric partitions of this plane to structure the data. For randomly generated, evenly distributed points there seems to be little to choose. However it is often the case that "real world" data sets do not show the same properties as generated sets; in particular, they tend to be much less evenly distributed, with significant numbers of clusters and outliers. These factors can significantly affect the performance of indexing mechanisms. Figures 8 and 9 show a sample taken from the SISAP *colors* data set with Euclidean distance applied, showing four different partitions. Four different partitions of the plane have been arbitrarily selected and applied. The query threshold illustrated is 0.052 corresponding to a query returning 0.001% of the data.

In all cases, it can be noted that the partitions are even, leading to balanced indexing structures. It is very likely that skewed partitions may perform better, an aspect we have not yet investigated. However one important balanced partition is illustrated on the left hand side of Figure 8, implying that a balanced hyperplane tree can be efficiently constructed.

It can be seen that, in this case, partitioning the plane according to the height of individual points above the X-axis is the most effective strategy. The

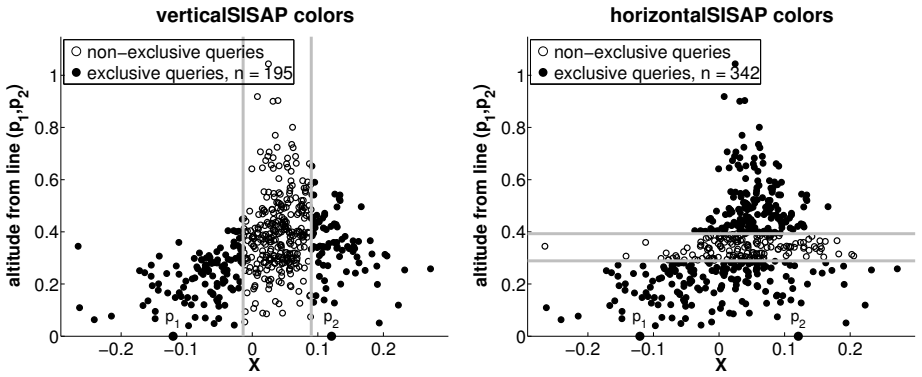

Fig. 8: Scatter diagrams dividing the plane equally in X and Y dimension, either can be used for partitioning a hyperplane tree structure. We show results for the "horizontal" pattern in Figure 12 where it is the best available partitioning.

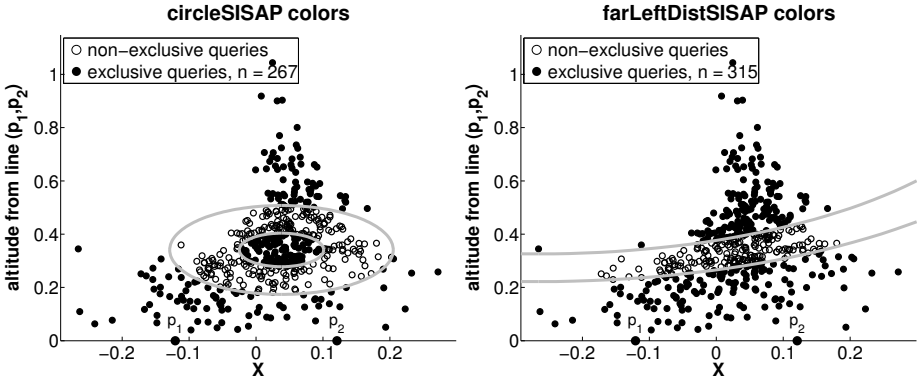

Fig. 9: Two more binary partitions, based now on median distance from arbitrary points in the plane (centre and top-left respectively); we have not yet found a use for these but include the diagrams to make the point that any such partition may be used.

disadvantage with this is that a little more calculation is required to plot the height of the point, rather than its offset from the Y-axis; however this is a very minor effect when significantly more distance calculations can be avoided.

4.3 Balance

As already noted, any of the partitions shown above can be simply used to bisect the data and thus produced a balanced indexing structure. These examples are all defined using a single real value with respect to the planar geometry. This can be calculated for each object within the subset to be divided, and the median can be found very efficiently using the QuickMedianSort algorithm; for a random distribution of points, the practical cost of balancing a binary tree at construction time appears similar to performing QuickSort once on all the data. While balanced structures are often slower than unbalanced ones for relatively small data sets, they become rapidly more desirable as the size of the data increases, and again more so if it is too large to fit in main memory and requires to be stored in backing store pages. The ability to balance the data without reducing the effectiveness of the exclusion mechanism therefore seems important. One further area of investigation, not yet performed, would be the effect of controlling the balance, which once again is arbitrarily possible simply by selecting different offset values. In general this will increase the probability of exclusion at cost of excluding smaller subsets of the data, and the effectiveness will depend on the individual distributions of the different strategies.

5 Experiments and Results

To illustrate the effects discussed, we have implemented a generic partition tree which can be specialised according to a number of criteria. All of the core code executed is the same[5], allowing fair comparisons to be made for both distance measurement counts and elapsed time. The generic partition tree can be parameterised according to the following criteria:

Hilbert or Hyperbolic: the essence of our investigation

Hilbert partition type: horizontal or vertical; we have not yet experimented with any other partition of the plane

Balanced, or Unbalanced: as explained in the text, all the partitioned spaces can be balanced, in these tests we choose an even left/right split. Unbalanced spaces tested are split according only to which reference point is closer; both Hilbert and Hyperbolic exclusion are tested for these.

Reference point selection: Three different strategies are tested. The first reference point is arbitrarily selected, and the second is one of: the closest (non-identical) value within the subset; a randomly-selected value from the subset; and the furthest value from within the subset.

[5] All of the (Java) code for these experiments can be downloaded from `https://bitbucket.org/richardconnor/metric-space-framework/`.

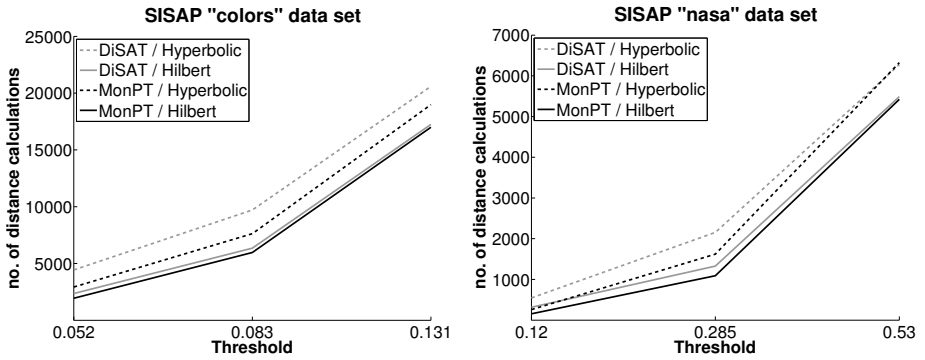

Fig. 10: Number of distance calculations per query for two SISAP benchmark sets. The best case for each data is Hilbert exclusion with a monotonous partition tree.

In all cases, with each partition two pivot values are kept and used in conjunction with each other exclusion policy: a cover radius is stored for the respective left/right reference points, and also the minimum radius between each reference point and the closest point in the opposing semi-space.

All tests are perfomed over SISAP *colors* and *nasa* benchmark data sets [5], using Euclidean distance, taking 10% of the set to act as queries over the remainder and measuring only the number of distance calculations performed per query ($n = 101.5k, 36.1k$ respectively.) In the remaining text we highlight some of the more interesting results.

5.1 Results

The smallest number of distances required for indexing was achieved by the unbalanced monotonous tree using Hilbert exclusion, with the reference points separated as far as possible. Figure 10 shows these results, in each case the bottom line on the graph indicating the best performance of our Hilbert exclusion mechanism in this context[6]. For comparison, the DiSAT [2] results with the two exclusion mechanisms are shown with grey lines; DiSAT/Hyperbolic, the top line on this chart, at the time of its publication was the best known general-purpose indexing mechanism.

Figure 11 shows the relative effect of reference point separation when using Hilbert and Hyperbolic exclusion. Clearly, the furthest separation works best as would always be expected. The point here to note is the relative disadvantage suffered by the four-point metric with a cheaper choice of reference point. As collection size increases, the selection of multiple good reference points becomes relatively more expensive; with the four-point properties, building a

[6] which we therefore believe makes the best performance yet published for these metric/dataset combinations

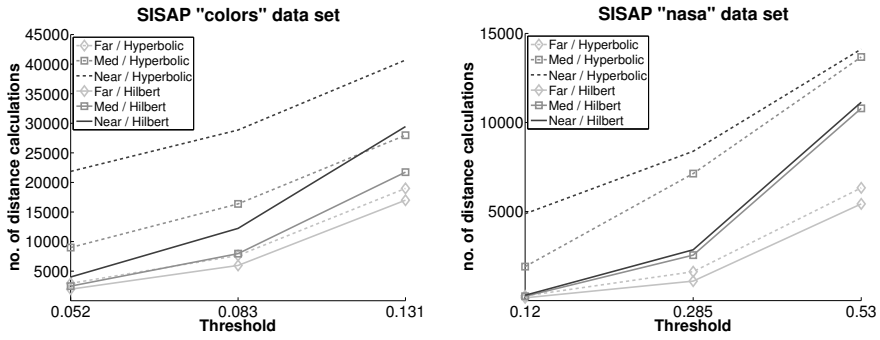

Fig. 11: Unbalanced hyperplane tree, different reference point separations. Choice of reference point is far less important for Hilbert exclusion, potentially allowing dramatic reductions to build time performance.

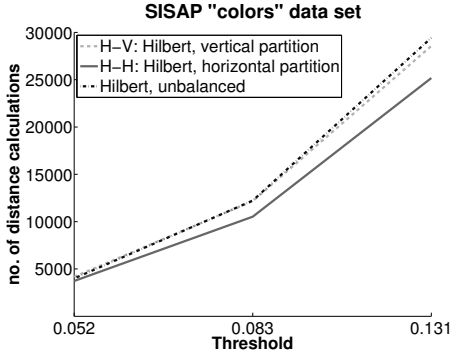

Fig. 12: Hilbert-Horizontal with close reference points; H-H is the bottom line, compared with H-V and unbalanced Hilbert. Hyperbolic exclusion with these reference points is shown as the top line of Figure11.

high-performance index is much, much cheaper as the choice of reference point is much less significant.

Finally, we can show one of our other partitions in action: the "horizontal" partition shown on the right-hand side of Figure 8. Figure 12 shows this partition in comparison with the vertical and unbalanced Hilbert partitions when the close reference points are selected for the *colors* data set. Although, as can be seen in comparison with other graphs, this is not the best way we have found of searching this particular data set, the graph is included as a demonstration that a completely novel partitioning technique can be the best with some selections of data sets and reference points; there is much work still to do here. In fact, if partition exclusion alone is used this technique is the best available, but the way the space is partitions means many less cover radius exclusions are made; definitely the subject of further work.

6 Conclusions

We have presented here a novel observation based on the four-point property that is possessed by many useful distance metrics. We have shown how, if it is guaranteed that any four points from the original space may be embedded in ℓ_2^3 as a tetrahedron, some much tighter geometric properties exist, in particular we have shown a lower-bound distance that can be calculated from knowledge of the sides of two tetrahedral faces. We have shown a few examples of how metric indexes can be constructed from this property and, although at an early stage of investigation, we have already shown a new best-performance for Euclidean distance search over two of the SISAP benchmark datasets. We believe a step change in improvement for exact search is possible; already our improved distance counts represent 29% and 44% of the previously published best results for *nasa* and *colors* respectively, using a structure which is much simpler and has a much smaller build time; we think much greater improvement is yet possible.

Acknowledgements We would like to thank the anonymous referees for helpful comments on an earlier version of this paper. Richard Connor would like to acknowledge support by the National Research Council of Italy (CNR) for a Short-term Mobility Fellowship (STM) in June 2015, which funded a stay at ISTI-CNR in Pisa during which much of this work was conceived.

References

1. S. Brin. Near neighbor search in large metric spaces. In *21th International Conference on Very Large Data Bases (VLDB 1995)*, 1995.
2. Edgar Chávez, Verónica Ludueña, Nora Reyes, and Patricia Roggero. Faster proximity searching with the distal SAT. *Information Systems*, pages –, 2016.
3. Edgar Chávez and Gonzalo Navarro. Metric databases. In Laura C. Rivero, Jorge Horacio Doorn, and Viviana E. Ferraggine, editors, *Encyclopedia of Database Technologies and Applications*, pages 366–371. Idea Group, 2005.
4. R. Connor, F. A. Cardillo, L. Vadicamo, and F. Rabitti. Hilbert Exclusion: Improved Metric Search through Finite Isometric Embeddings. *ArXiv e-prints, accepted for publication ACM TOIS, July 2016*, April 2016.
5. Karina Figueroa, Gonzalo Navarro, and Edgar Chávez. Metric spaces library. `www.sisap.org/library/manual.pdf`.
6. H. Noltemeier, K. Verbarg, and C. Zirkelbach. *Data structures and efficient algorithms: Final Report on the DFG Special Joint Initiative*, chapter Monotonous Bisector* Trees — a tool for efficient partitioning of complex scenes of geometric objects, pages 186–203. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.
7. David Novak, Michal Batko, and Pavel Zezula. Metric index: An efficient and scalable solution for precise and approximate similarity search. *Information Systems*, 36(4):721 – 733, 2011. Selected Papers from the 2nd International Workshop on Similarity Search and Applications {SISAP} 2009.
8. Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko. *Similarity search: the metric space approach*, volume 32 of *Advances in Database Systems*. Springer, 2006.