Nearest Neighbor Search using Sketches as Quantized Images of Dimension Reduction*

Naoya Higuchi[1], Yasunobu Imamura[1], Tetsuji Kuboyama[2], Kouichi Hirata[1] and Takeshi Shinohara[1]
[1]*Kyushu Institute of Technology, Kawazu 680-4, Iizuka 820-8502, Japan*
[2]*Gakushuin University, Mejiro 1-5-1, Toshima, Tokyo 171-8588, Japan*

Keywords: Similarity Search, Sketches, Ball Partitioning, Hamming Distance, Dimension Reduction, Distance Lower Bound, Quantized Images.

Abstract: In this paper, we discuss sketches based on ball partitioning (BP), which are compact bit sequences representing multidimensional data. The conventional nearest search using sketches consists of two stages. The first stage selects candidates depending on the Hamming distances between sketches. Then, the second stage selects the nearest neighbor from the candidates. Since the Hamming distance cannot completely reflect the original distance, more candidates are needed to achieve higher accuracy. On the other hand, we can regard BP sketches as quantized images of a dimension reduction. Although quantization error is very large if we use only sketches to compute distances, we can partly recover distance information using query. That is, we can compute a lower bound of distance between a query and a data using only query and the sketch of the data. We propose candidate selection methods at the first stage using the lower bounds. Using the proposed method, higher level of accuracy for nearest neighbor search is shown through experimenting on multidimensional data such as images, music and colors.

1 INTRODUCTION

A *similarity search* is one of important and famous tasks for information retrieval in multidimensional data. The purpose of the similarity search is to find data near to a query with respect to a given distance. In order to avoid "the curse of dimensionality," *dimension reduction* techniques have been developed. One of the most important properties of dimension reduction is that the distance between any two data is *not expanded* after transformation. In other word, dimension reduction is a Lipschitz continuous mapping. This property establishes the safety of pruning strategy that excludes any data, which is far away from a query in the lower dimensional space, from the search target without examining the distance in the original space. For example, K-L transformation (or equivalently principal component analysis (PCA)) (Fukunaga, 1990) and FastMap (Faloutsos and Lin, 1995) are known as dimension reductions for a Euclidean space. Also H-Map (Shinohara et al., 1999) and Simple-Map (S-Map, for short) (Shinohara and Ishizaka, 2002) are dimension reductions applicable to any metric space such as L_1 metric space and strings with edit distance (Wagner and Fischer, 1974).

As another technique for efficient similarity search in multidimensional spaces, *sketch* (Müller and Shinohara, 2009; Mic et al., 2016; Dong et al., 2008; Mic et al., 2015; Wang et al., 2007) has also been developed. Sketch is a compact bit sequence representing multidimensional data. *Ball partitioning* (Uhlmann, 1991) (BP, for short) is a method to make sketches by assigning a bit 0 or 1 to a data, such that 0 if it is in a ball and 1 otherwise. BP is also used in vantage point tree (Yianilos, 1993).

Conventionally, the similarity search using sketches consists of two stages. The first stage selects candidates depending on their Hamming distances between sketches. Here, we should note that the computational cost of the Hamming distance is very small if we use efficient bit operations such as exclusive or and bit count. The second stage selects the nearest neighbor by comparing the candidates with the query using distances in the original space. As the Hamming distance cannot preserve the information of distances between data, similarity search using sketches

*This work is partially supported by Grant-in-Aid for Scientific Research 17H00762, 16H02870, 16H01743 and 15K12102 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

Higuchi, N., Imamura, Y., Kuboyama, T., Hirata, K. and Shinohara, T.
Nearest Neighbor Search using Sketches as Quantized Images of Dimension Reduction.
DOI: 10.5220/0006585003560363
In *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2018)*, pages 356-363
ISBN: 978-989-758-276-9

is only an approximation method. One of the most important tasks for sketches is achieving high accuracy with small number of candidates obtained at the first stage.

When the width w of sketches is considered as the dimensionality, the dimensionality may not be reduced. However, as the size w bit is usually much smaller than the original data, we may consider mapping to sketches as a quasi-dimension reduction. Nevertheless, the Lipschitz continuity of the mapping is not guaranteed as long as the Hamming distance is used.

On the other hand, since the sketch of w bits is defined by using w BP's, we can regard w-bits sketches as quantized images of S-Map (Ohno, 2011). That is, BP sketches are quantized images of S-Map, where each axis value is quantized to one bit depending on whether or not greater than thresholds. Note that the L_∞ distance should be used to guarantee the Lipschitz continuity of S-Map. Any L_∞ distance between sketches is 0 or 1, that is, the quantization error is very large. As for data in the database, we should use only sketches at the first stage because the original high dimensional data are too large. However, as for queries, we can use the original queries as well as their sketches. Hence, in this paper, we show that, for each sketch bit, a lower bound of distance between a query q and data x can be calculated using q and the sketch of x without the original x. If we take $score_\infty$ defined by the maximum of distance lower bounds as the aggregation like L_∞ distance, the distance estimation using sketch is not expanded and a BP sketch mapping can be considered as a quasi-dimension reduction. Similar idea is also found in *asymmetric distance estimation* (Dong et al., 2008; Jain et al., 2011; Balu et al., 2014), where sketches are constructed by *generalized hyperplane partitioning*(GHP), some of them assume the Euclidean distance or cosine distance, and their estimation may expand the distance.

For w bit sketches, we have w distance lower bounds. To guarantee distance lower bounds, we have to use the aggregation by maximum $score_\infty$. In approximate nearest neighbor search using sketches, we propose $score_1$ and $score_2$, which are the aggregations by sum and square sum respectively, which are no longer distance lower bounds. By experimenting on images, music and colors databases, we observe that $score_1$ and $score_2$ achieve a more accurate nearest neighbor search compared to the Hamming distance and $score_\infty$. Naïve implementation of aggregation needs more computational cost than the Hamming distance using bit operations. For each query we can precompute aggregations to construct a table function. The cost for aggregation using table function is almost comparable with the cost for the Hamming distance using bit operations. We can ignore this preprocess cost if we search large database.

We believe that our contribution lies in the following three points. First, in any metric space, based on the observation that BP sketches are quantized images of dimension reduction S-Map, we point out that the sketch mapping can be considered as a quasi-dimension reduction by aggregating distance lower bounds between the query and a sketch for each bit of the sketch in L_∞ manner. Similar methods are used elsewhere (Charikar, 2002; Jain et al., 2011), where sketches are based generalized hyperplane partitioning (GHP) in Euclidean distance and cosine distance, and aggregated distance lower bounds is just a distance estimation but lower bound. Here, we also point out that GHP sketches are quantized images of H-Map. Thus, our approach is easily extended to be applicable to any GHP based sketches. Second, we propose a low cost method to compute aggregations using precomputed table functions. This contribution is not very significant but important for practical problems. In our setting, we assume that data are made by feature extraction function and they are not very high dimension. Therefore, we cannot ignore the computational cost for the first stage. Third, we propose sum or square sum aggregations of distance lower bounds for the priority at the first stage. Such aggregations are no longer distance lower bound. Nevertheless they are more useful than maximum aggregation. Similar techniques are found elsewhere, where GHP based sketches are considered in Euclidean distance or cosine distance. Our method is applicable to BP sketches in any metric space and easily extended to GHP.

2 PRELIMINARIES

Here, we briefly introduce some necessary concepts for our discussion.

2.1 Dimension Reduction and Simple-Map

We assume two metric spaces (U,D) and (U',D'), where D and D' are distance functions satisfying triangle inequality. Let $dim(x)$ for a data x denote the dimensionality of x. Then, we say that a mapping $\phi : U \to U'$ is a *dimension reduction* if it satisfies the following conditions for every $x, y \in U$:

$$\dim(\phi(x)) < \dim(x) \tag{1}$$

$$D'(\phi(x),\phi(y)) \leq D(x,y) \tag{2}$$

Condition (1) means that ϕ reduces the dimensionality of data. However, it is not handled very strictly in this paper. For example, the size required to represent data is regarded as the dimensionality. Condition (2) means that D' provides the lower bound of a distance $D(x,y)$, which guarantees to ignore a data without computing D in similarity search. For example, if $D'(\phi(q),\phi(x))$ exceeds the current search diameter of a query q, then x can be ignored without computing $D(q,x)$.

A Simple-Map (S-Map, for short) (Shinohara and Ishizaka, 2002) is a kind of *Fréchet embedding* (Matousek, 2002) that any finite metric space of n points can be embedded isometrically into n-dimensional L_∞ normed space. A similar idea has also been proposed by Hjaltason and Samet (Hjaltason and Samet, 2003). For a point $p \in U$ called a *pivot*, we define an S-Map ϕ_p of $x \in U$ with p as follows.

$$\phi_p(x) = D(p,x).$$

By triangle inequality, the following inequality holds for every $x, y \in U$:

$$|\phi_p(x) - \phi_p(y)| \leq D(x,y).$$

Furthermore, for a set $P = \{p_1, \ldots, p_m\}$ of pivots, we define S-Map ϕ_P with P as follows.

$$\phi_P(x) = (\phi_{p_1}(x), \ldots, \phi_{p_m}(x)).$$

Suppose that we give D' as follows:

$$D'(\phi_P(x),\phi_P(y)) = \max_{i=1}^{m} |\phi_{p_i}(x) - \phi_{p_i}(y)|$$

.

In other words, if the projected space U' is considered as an L_∞ metric space, then, an S-Map is a dimension reduction.

In H-Map (Shinohara et al., 1999), using a pair of two points $(p_1, p_2) \in U$ as a pivot, we define an H-Map $\phi_{(p_1,p_2)}$ of $x \in U$ as follows.

$$\phi_{(p_1,p_2)}(x) = \frac{D(p_1,x) - D(p_2,x)}{2}.$$

Then, H-Map is also a dimension reduction.

2.2 Nearest Neighbor Search using Sketches

We assume that data in the given database are indexed by natural numbers 1 to n. Thus, let $db = \{x_1, \cdots, x_n\}$ be the given database of n data. The dissimilarity between two data x_i and x_j is defined as distance $D(x_i, x_j)$. The *nearest neighbor search* for a query q is to find $x \in db$ such that $D(q,x) \leq D(q,y)$ for all $y \in db$. Let s be a function which maps data to its sketch. We can realize the k-nearest neighbor search using sketches as follows, where $K > k$.

1. Preparation stage:
   Calculate all the sketches $s(x_1), \ldots, s(x_n)$.
2. First stage (Filtering using the Hamming distances of sketches):
   Calculate the sketch $s(q)$ of query q.
   Calculate all the Hamming distances of sketches from $s(q)$.
   Select the closest K sketches $s(x_{i_1}), \ldots, s(x_{i_K})$ to $s(q)$.
3. Second stage (Nearest neighbor search using actual distances):
   Select the k nearest neighbor data from the candidates $x_{i_1}, \ldots, x_{i_K}$.

Sketches are relatively small structures with respect to their original feature data. For example, we use sketches of 32 bits for image feature data of 64 bytes in our experiments. At the first stage of searching process, we use the Hamming distances, which can be more easily calculated using bit operations than the actual distances between features. However, sketches cannot preserve all the distance relation. Therefore, we use them as a filter. The larger K of the number of candidates at the first stage achieves a more accurate but slower search. Thus, one of the most important subjects on sketch is to achieve higher accuracy with smaller K, or equivalently, to speed up search within acceptable error.

2.3 Sketches based on Ball Partitioning

In this paper, we consider sketches based on *ball partitioning* (BP). A pair (p,r) of a point and a radius is called a *pivot*. A ball partitioning $BP_{(p,r)}$ is defined as follows:

$$BP_{(p,r)}(x) = \begin{cases} 0 & \text{if } D(p,x) \leq r \\ 1 & \text{otherwise} \end{cases}$$

A BP based sketch function s_P of *width* w is defined by a set of w pivots $P = \{(p_1,r_1), ..., (p_w,r_w)\}$ as follows:

$$s_P(x) = BP_{(p_1,r_1)}(x) ... BP_{(p_w,r_w)}(x)$$

Consider 4 points A, B, C, D on a Euclidean plane in Figure 1. Using a set of two pivots $P = \{(p_1,r_1),(p_2,r_2)\}$, their sketches are $s_P(A) = 01$, $s_P(B) = 00$, $s_P(C) = 10$, $s_P(D) = 11$.

Let q be any query outside of both balls. Since $s_P(q) = 11$, Hamming distances between sketches of q and A, B, C, D are 1, 2, 1, 0, respectively. The order

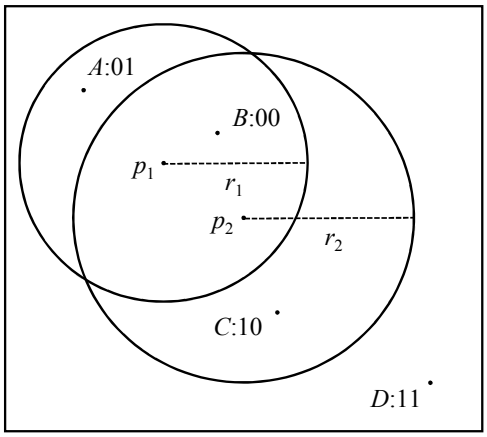

Figure 1: 2 bit sketches by two balls.

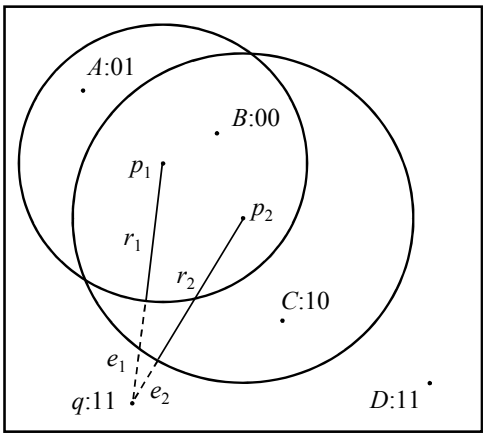

Figure 2: Distance lower bound between query and sketches.

of conventional priority at the first stage is $D < A = C < B$. Note that A and C cannot be distinguished by Hamming distances from q.

2.4 Pivot Selection for Sketches using Binary Quantization

The pivot selection for sketch is a very important problem but outside of our scope. Here, we briefly introduce a heuristic algorithm SELECTPIVOTQBP, which is not the best search algorithm but can find a relatively good set of pivots within a small computation time.

Binary quantization (BQ) selects one of the 2^n corners of n dimensional feature space, which are too many to efficiently be selected. BQ randomly chooses a data from database and quantizes it to a corner according with the median.

In our experiments, we adopt probability of collisions between sketches as the evaluation function of pivots. If two different data x and y have the same sketch, we say a *collision* occurs. From experience, sketches with less collisions provide more accurate nearest neighbor search. The algorithm SELECTPIVOTQBP in Algorithm 15 illustrates the algorithm for selecting pivots of QBP, where D is the distance function and *eval* is the evaluation function for pivot.

3 DISTANCE LOWER BOUNDS BETWEEN QUERIES AND SKETCHES

In this section, we introduce a method to compute distance lower bounds between queries and sketches.

First, let us explain the idea using an example. Consider four points A,B,C,D and a query q in Figure 2.

The sketch $s_P(q)$ has 1 at the leftmost bit while $s_P(A)$ has 0. Therefore,

$$D(p_1,A) \leq r_1.$$

By triangle inequality,

$$D(q,A) \geq D(p_1,q) - D(p_1,A).$$

From these two inequalities,

$$D(q,A) \geq D(p_1,q) - r_1.$$

The lower bound of distance between q and any point whose leftmost sketch bit is 0 is given by (as e_1 in Figure 2),

$$e_1(q,0*) = D(p_1,q) - r_1.$$

Similarly, the lower bound of distance between q and C is

$$e_2(q,*0) = D(p_2,q) - r_2.$$

Here, note that the above lower bounds can be calculated only from $D(p_1,q)$, $D(p_2,q)$, r_1 and r_2 without $D(q,A)$ and $D(q,C)$.

More formally, we can compute the lower bounds as explained below. Let p_i and r_i be the center and radius of i-th pivot. Assume i-th sketch bits for a query q and data x are different from each other, that is, $D(p_i,q) > r_i$ and $D(p_i,x) \leq r_i$ (or $D(p_i,q) \leq r_i$ and $D(p_i,x) > r_i$). This difference is extremely quantized as 1 in their Hamming distance. Calculating the distances $D(p_i,x)$ for all x in database require more computational costs. On the other hand, the distance $D(q,x)$ is partly recovered using $D(p_i,q)$ only.

When

$$D(p_i,q) > r_i \text{ and } D(p_i,x) \leq r_i,$$

```
/* M : the dimension of data, N : the number of data */
/* db[N][M] : database, med[M] : the median of feature values */
/* T : the number of random trials for each axis */
/* MIN : the minimum feature value, MAX : the maximum feature value
*/
1  procedure SELECTPIVOTQBP(p[W][M], r[W], W)
       /* p[W][M], r[W] : the centers and radiuses of pivots */
       /* W : the width of sketches */
       /* eval : evaluation of pivots to be minimized */
2      for i = 1 to W do
3          best ← ∞;
4          for t = 1 to T do
5              c ← random();
               /* binary quantization */
6              for j = 1 to M do
7                  if db[c][j] ≤ med[j] then
8                      p[i][j] ← MIN;
9                  else
10                     p[i][j] ← MAX;
11             r[i] ← the median of D(p[i], db[1]), ..., D(p[i], db[N]);
12             current ← eval(p[1], r[1], ..., p[i], r[i]);
13             if current < best then
14                 best ← current; temp ← p[i]; rtemp ← r[i];
15         p[i] ← temp; r[i] ← rtemp;
```

Algorithm 1: SELECTPIVOTQBP.

by triangle inequality, as shown in the left of Figure 3,

$$D(q,x) \geq D(p_i,q) - D(p_i,x) \geq D(p_i,q) - r_i.$$

Similarly, when

$$D(p_i,q) \leq r_i \text{ and } D(p_i,x) > r_i,$$

we have, as shown in the right of Figure 3,

$$D(q,x) \geq D(p_i,x) - D(p_i,q) \geq r_i - D(p_i,q).$$

Thus, we have the lower bound $e_i(q,s(x))$ of $D(q,x)$ by

$$e_i(q,s(x)) = \begin{cases} 0 & \text{if } s_i(q) = s_i(x) \\ |D(p_i,q) - r_i| & \text{if } s_i(q) \neq s_i(x) \end{cases}$$

We propose priorities using the distance lower bound $e_i(q,x)$ as the criteria to select candidates at the first stage. When we use as the priority

$$score_\infty(q,s(x)) = \max_{i=1}^{w} e_i(q,s(x))$$

which is the maximum lower bound, we can safely prune some of candidates because it is really a distance lower bound.

On the other hand, the Hamming distance is the *sum* of the differences. We also examine the sum and the square sum of the distance lower bounds

$$score_1(q,s(x)) = \sum_{i=1}^{w} e_i(q,s(x))$$

$$score_2(q,s(x)) = \sum_{i=1}^{w} (e_i(q,x))^2$$

as the priorities. Although they are no longer theoretical lower bounds, experiments show that they bring a more accurate nearest neighbor search.

In the conventional sketch search, the Hamming distance is used as the priority at the first stage, and it is advantageous that high speed calculation is possible by bit operations. In the proposed method, since the distance lower bounds are obtained for a large number of data for each question, the cost increase can be reduced by preparing a table function. The cost increase can almost be ignored.

4 EXPERIMENTS

In this section, we report experiments using several data, which are images, music and colors, as follows:

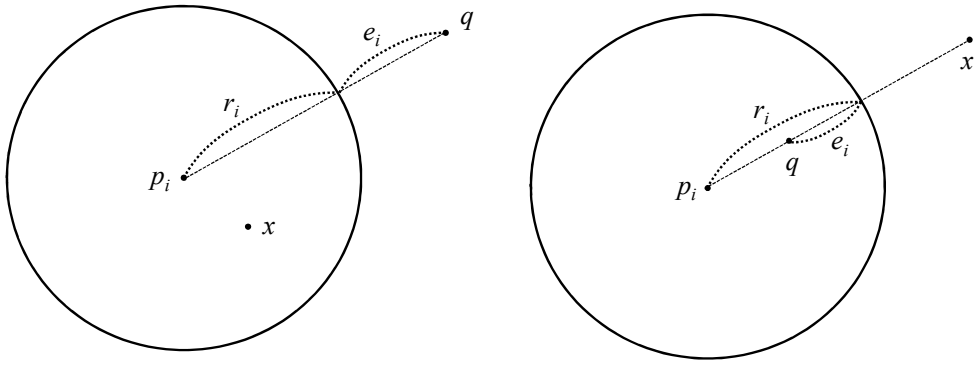

Figure 3: Distance Lower Bound.

- images: about 70 million 2D frequency spectrums of 64 dimension data extracted from 2,900 videos.
- music: about 70 million mel-frequency spectrums of 96 dimension data extracted from 1,400 music CD.
- colors: about 110,000 data of 112 dimension from SISAP database.

We adopted 32 bit as the width of sketches. 32 pivots are selected by SELECTPIVOTQBP with TRIAL = 1000. For each database, we selected 10 different sets of pivots. The experimental results of precision show the average of these 10 sets of pivots.

Randomly generated data are not appropriate for experiments of nearest neighbor search, because in higher dimensional spaces it is rare to find near data. Therefore, we prepare five types of queries: very-near, near, middle, far, very-far which are generated from randomly selected pairs from database with mixing noise ratio of $5-10\%$, $15-20\%$, $25-30\%$, $35-40\%$, $45-50\%$, respectively. For example, a very-near query q is a weighted sum of randomly selected data x and y from database with weight 10% and 90%, respectively. For each noise level, we prepare 100 queries. The average of nearest neighbor distances for queries are shown in Figure 4.

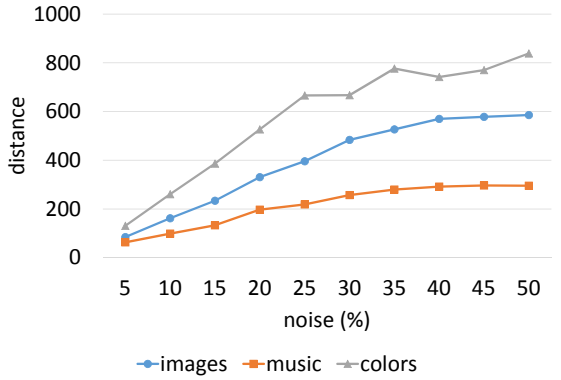

Figure 4: The average of nearest neighbor distances.

First, we compare run times using priorities of the Hamming distance, $score_\infty$, $score_1$, and $score_2$. Table 1 illustrates the computer environment.

Table 1: The computer environment.

CPU	Intel(R) Xeon(R) CPU E5-2640 2.5 GHz
memory	64GBytes

Table 2 shows the average run time per query in millisecond. From this table, no significant difference between methods is observed. Thus, computational costs for distance lower bounds and their aggregation are almost the same as that for the Hamming distance. In this table, only the results for images and music databases are shown, because the run time for colors is very small.

Table 2: Run time.

DB	images		music	
K	0.10%	1.0%	0.10%	1.0%
Hamming	19.5	71.5	21.1	87.5
$score_1$	16.8	85.5	23.0	111
$score_2$	22.5	89.1	23.4	121
$score_\infty$	15.2	63.4	16.9	74.2

The *precision* of the search is defined as the provability that the top K candidates of the first stage include the exact nearest neighbor. We report the results for K of three sizes 0.01%, 0.1% and 1.0% relative to database size.

Figure 5, 6 and 7 summarize the precisions of search for databases images, music and colors, respectively, where $score_2$ is omitted, because it is very similar to $score_1$.

5 CONCLUDING REMARKS

From the experiments, we confirmed that the precision is improved by the proposed methods using the

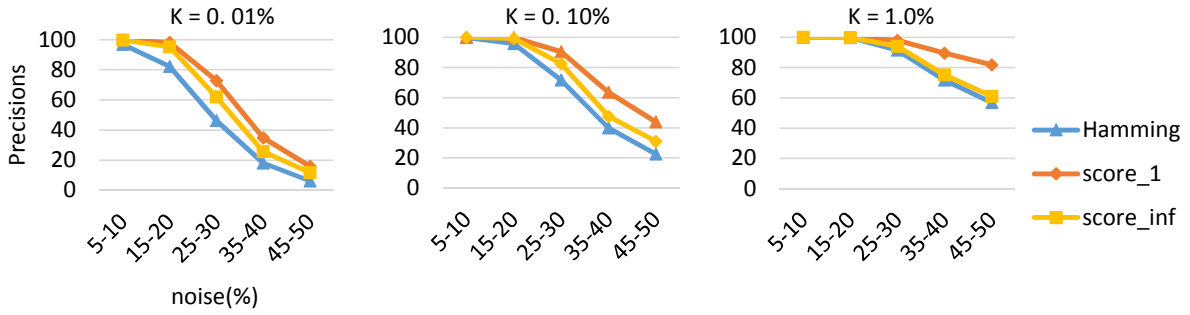

Figure 5: Precisions for images.

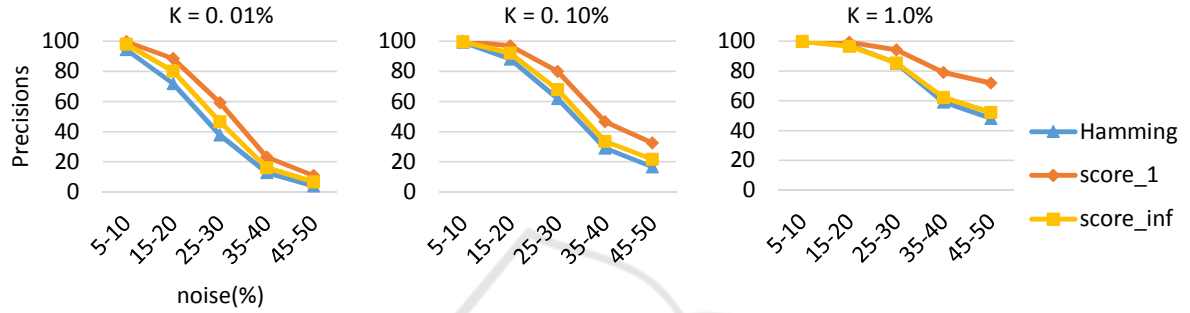

Figure 6: Precisions for music.

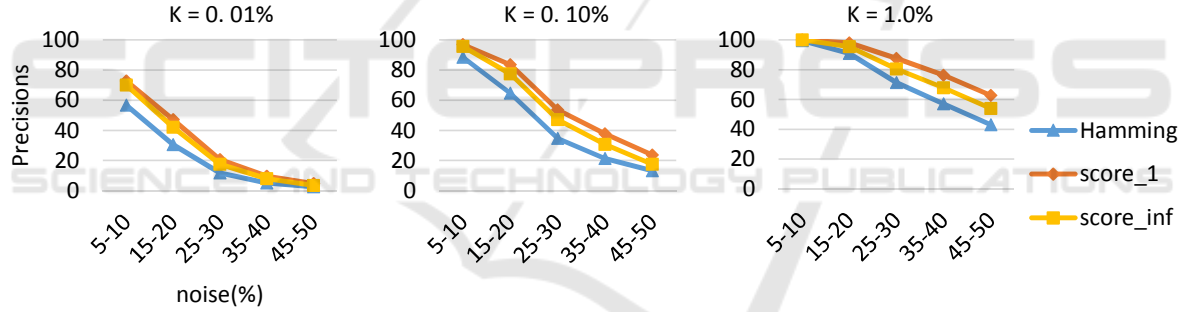

Figure 7: Precisions for colors.

distance lower bounds, their maximum $score_\infty$, their total sum $score_1$, and their total square sum $score_2$, compared to using the Hamming distance. One of the most important tasks for the future work of this research may be to examine why $score_1$ and $score_2$ give better priorities than $score_\infty$ and Hamming distance. Since all priorities in these experiments have a similar tendency to any of the three databases of image, music and colors, we think that the dependency of database is unlikely. We should consider the influence of the width w of sketches, because $score_\infty$ will improve by the larger w, but $score_1$ and $score_2$ may get worse. There is a possibility that the same phenomenon may also occur in S-Map, where L_∞ is used in projected space. This is also a future issue.

In the experiments we reported, we optimized sketches by random trials using binary quantization. We can apply optimization techniques such as local search and simulated annealing. We run *annealing by increasing resampling* (AIR) introduced by Imamura et.al (Imamura et al., 2017) on pivot selection for sketches to minimize collisions. Although sketches with smaller collision probability are found by using them, the search precision is not improved. Therefore, we have to consider other evaluation function for sketches rather than the probability of collision. For example, it is conceivable to use the correlation coefficient of distances before and after projection between sample pairs and the distance preservation ratio of the distance lower bounds as the evaluation function.

In this paper, we consider sketches based on ball partitioning, which can be considered as quantized images of S-Map. Generalized hyperplane partitioning (Uhlmann, 1991) (GHP) can also be used to make sketches. GHP based sketches can be considered as quantized images of H-Map. We can calculate

distance lower bounds between queries and sketches even for GHP based sketches. We should also investigate the effectiveness of proposed methods for GHP.

We compare aggregation methods $score_1$ and $score_2$ in addition to $score_\infty$. In our experiments, their performances are better than the traditional method using the Hamming distances. The $score_\infty$ gives a theoretical distance lower bound. Therefore, for K objects selected at the first stage based on $score_1$ or $score_2$, safe pruning based on $score_\infty$ is applicable. At the second stage, any candidate with $score_\infty$ larger than the actual distance of the provisional nearest neighbor can be removed without computing the actual distance. We also have to consider $score_p$ aggregation for p other than 1 and 2, such as 1.5.

REFERENCES

Balu, R., Furon, T., and Jégou, H. (2014). Beyond "project and sign" for cosine estimation with binary codes. In *Proc. ICASPP 2014*, pages 6884–6888. IEEE.

Charikar, M. (2002). Similarity estimation techniques from rounding algorithms. In *Proc. STOC*, pages 380–388. ACM.

Dong, W., Charikar, M., and Li, K. (2008). Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces. In *Proc. ACM SIGIR'08*, pages 123–130.

Faloutsos, C. and Lin, K. (1995). Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proc. ACM SIGMOD'95*, volume 24, pages 163–174.

Fukunaga, K. (1990). *Statistical pattern recognition. (second edition)*. Academic Press.

Hjaltason, G. and Samet, H. (2003). Properties of embedding methods for similarity searching in metric space. *IEEE Transactions on Pat. Anal. Mach. Intel.*, 25(5):530–549.

Imamura, Y., Higuchi, N., Kuboyama, T., Hirata, K., and Shinohara, T. (2017). Pivot selection for dimension reduction using annealing by increasing resampling. In *Proc. LWDA 2017*.

Jain, M., Jégou, H., and Gros, P. (2011). Asymmetric hamming embedding: taking the best of our bits for large scale image search. In *Proc. AMC Multimedia 2011*, pages 1441–1444.

Matousek, J. (2002). *Lectures on discrete geometry*. Springer Verlag.

Mic, V., Novak, D., and Zezula, P. (2015). Improving sketches for similarity search. In *Proc. MEMICS'15*, pages 45–57.

Mic, V., Novak, D., and Zezula, P. (2016). Speeding up similarity search by sketches. In *Proc. SISAP 2016*, pages 250–258.

Müller, A. and Shinohara, T. (2009). Efficient similarity search by reducing i/o with compressed sketches. In *Proc. SISAP'09*, pages 30–38.

Ohno, S. (2011). A study on quantization dimension reduction mapping for similarity search in multidimensional databases. Master's thesis, Kyushu Institute of Technology, (in Japanese).

Shinohara, T., Chen, J., and Ishizaka, H. (1999). H-map: A dimension reduction mapping for approximate retrieval of multi-dimensional data. In *Proc. DS'99, LNAI 1721*, pages 299–305.

Shinohara, T. and Ishizaka, H. (2002). On dimension reduction mappings for approximate retrieval of multi-dimensional data. In *Progress of Discovery Science, LNCS 2281*, pages 89–94. Springer Berlin / Heidelberg.

Uhlmann, J. (1991). Satisfying general proximity/similarity queries with metric trees. *Inform. Process. Let.*, 40(4):175–179.

Wagner, R. and Fischer, M. (1974). The string-to-string correction problem. *J. ACM*, 21:168–178.

Wang, Z., Dong, W., Josephson, W., Q. Lv, M. C., and Li, K. (2007). Sizing sketches: A rank-based analysis for similarity search. In *Proc. ACM SIGMETRICS'07*, pages 157–168.

Yianilos, P. (1993). Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proc. SODA 1993*, pages 311–321. ACM Press.