Multi-dimensional Selectivity Estimation Using Compressed Histogram Information*

Ju-Hong Lee† Deok-Hwan Kim† Chin-Wan Chung‡

†Department of Information and Communication Engineering ‡Department of Computer Science
Korea Advanced Institute of Science and Technology

{jhlee,dhkim,chungcw}@islab.kaist.ac.kr

ABSTRACT

The database query optimizer requires the estimation of the query selectivity to find the most efficient access plan. For queries referencing multiple attributes from the same relation, we need a multi-dimensional selectivity estimation technique when the attributes are dependent each other because the selectivity is determined by the joint data distribution of the attributes. Additionally, for multimedia databases, there are intrinsic requirements for the multi-dimensional selectivity estimation because feature vectors are stored in multi-dimensional indexing trees. In the 1-dimensional case, a histogram is practically the most preferable. In the multi-dimensional case, however, a histogram is not adequate because of high storage overhead and high error rates.

In this paper, we propose a novel approach for the multi-dimensional selectivity estimation. Compressed information from a large number of small-sized histogram buckets is maintained using the discrete cosine transform. This enables low error rates and low storage overheads even in high dimensions. In addition, this approach has the advantage of supporting dynamic data updates by eliminating the overhead for periodical reconstructions of the compressed information. Extensive experimental results show advantages of the proposed approach.

1. INTRODUCTION

The database query optimizer chooses an efficient execution plan among all possible plans by estimating the cost of each plan. One of the most important factors for computing the cost of a plan is the selectivity, which is defined as the ratio of the number of data in a query result to the total number of data in a database. The accuracy of the selectivity estimation significantly affects the selection of an efficient plan. The selectivity can be estimated using a variety of statistics that are kept in a database catalog. The statistics for the selectivity estimation usually approximates the data distribution of a database.

There are two classes in selectivity estimation problems according to the dimensionality. One is the 1-dimensional selectivity estimation and the other is the multi-dimensional selectivity estimation. The estimation of the result size of a query with a single attribute predicate depends on the data distribution of the attribute. This case is the 1-dimensional selectivity estimation problem. Regarding the multi-dimensional selectivity estimation, there are several applications that require it. The optimization of a query referencing multiple attributes from the same relation needs it, because the result size of the query depends on the joint data distribution of the attributes that is represented as a multi-dimensional space [PI97]. So do the optimization of fuzzy queries for multimedia repositories [CG96, Fa96, Fa98] and database ranking for selecting resources in a distributed environment such as the World Wide Web [CSZS97], because the feature vectors of multimedia data are stored in multi-dimensional index trees.

A variety of techniques were proposed based on how to approximate the data distribution. An excellent survey and the taxonomy of various selectivity estimation techniques appeared in [MCS98, CR94, PIHS96]. 1-dimensional selectivity estimation techniques are classified into four categories: the parametric, the curve fitting, the sampling, and the non-parametric. Among these classes, the histogram method in the non-parametric class is the most preferable because it approximates any data distribution and requires reasonably small storage with low error rates. And it does not incur run-time overheads. Several histogram techniques were proposed in order to reduce estimation errors[PIHS96]. For the multi-dimensional selectivity estimation, several estimation techniques were proposed: the method using the multilevel grid file(MLGF)[WKW94], the singular value decomposition(SVD), Hilbert numbering, PHASED, and MHIST [PI97]. These are all based on histogram techniques. And these were proposed under the assumption that a histogram method is also efficient in the multi-dimensional selectivity estimation as it is so in the 1-dimensional case. However, the situation of the multi-dimensional case is very different from that of the 1-dimensional case. In order to achieve low error rates, the size of histogram buckets must be small. As the dimension increases, the number of histogram buckets that can achieve low error rates increases explosively. This is because the number of histogram buckets is in inverse proportion to the dimension'th power to the normalized one-dimensional length of a partitioned multi-dimensional bucket as expressed by an equation below. It causes a severe storage overheads problem.

$$\#\text{of buckets} \propto \frac{1}{a^{\text{dim}}},\ 0<a<1,$$

where a is the 1-dimensional length of a bucket.

*This research was supported by the National Geographic Information Systems Technology Development Project and the Software Technology Enhancement Program 2000 of the Ministry of Science and Technology of Korea.

Therefore, it is impossible to maintain a reasonably small storage with low error rates in high dimensions. Also it is difficult to partition a multi-dimensional space into disjoint histogram buckets efficiently so that the error rates are kept small. From a practical point of view, these methods cannot be used in dimensions higher than three. Another problem is that all methods except the MLGF method cannot reflect dynamic data updates immediately to the statistics for the estimation. This leads to an additional overhead such as the periodical reconstruction of statistics for the estimation.

In this paper, motivated from the above problems, we propose a novel approach for the multi-dimensional selectivity estimation. The contents and contributions are as follows: Compressed information from a large number of small-sized buckets is maintained using the discrete cosine transform (DCT). This enables low storage overheads and low error rates even in high dimensions. This can be achieved from the fact that DCT can compress the information remarkably. That is, low error rates can be achieved by small-sized buckets and low storage overheads can be achieved by compressing a large amount of histogram bucket information. As another contribution, as far as we know, this is the first application in which DCT is used in high dimensions. DCT has been widely used in the image and signal processing area usually in 2-dimensional domain. Therefore, we also extend DCT from two dimension to high dimensions. In addition, this method has the advantage that it is not necessary to reconstruct statistics for selectivity estimation periodically, because it reflects dynamic data updates into the statistics for the estimation immediately. An extensive set of experiments show that the method proposed in this paper requires low storage overheads, achieves low error rates, and provides fast computations of the estimation even in high dimensions.

The paper is organized as follows: In Section 2, we describe 1-dimensional and multi-dimensional selectivity estimation techniques as well as their advantages and disadvantages. In Section 3, we introduce the discrete cosine transform. In section 4, we explain how discrete cosine transform can be used in the multi-dimensional selectivity estimation. In Section 5, we show experimental results and discuss them in detail. Finally, conclusions are made in Section 6.

2. RELATED WORK

First, we briefly describe 1-dimensional selectivity estimation techniques and explain multi-dimensional estimation techniques, and then discuss their problems.

2.1 One-dimensional Selectivity Estimation

Selectivity estimation techniques can be classified into four categories: the parametric method by model functions [Chri83], the curve fitting method by general polynomial functions [CR94, SLRD93], the sampling method [HNSS95], and the non-parametric method by histograms [Io93, IP95, PIHS96, JKMPSS98]. The parametric method approximates the data distribution of an attribute to a model function such as normal, exponential, Pearson, Zipf function, and computes free parameters for the model function under the assumption that the data distribution well fits the selected model function. The advantage of this method is that it requires a little storage, incurs low computation overheads, and provides accurate results when the data distribution fits the selected model function. However, if the data distribution does not fit the model function, the error rates of estimation results will be very high. And we must know a priori which model function fits the actual data distribution. If the actual data distribution does not fit any known model function, we cannot use this method. The curve fitting method was proposed to get more flexibility than the parametric method. This method uses a general polynomial function in fitting the actual data distribution. The advantage of this method is that it can approximate any data distribution. However, it has the negative value problem and the rounding error propagation problem. So, we must be careful to use this method. The sampling method is mainly used for statistical queries that have aggregate functions. It retrieves sample data from a database and applies the sample data to a query in order to get statistics of the query. The sampling method must take enough sample data to achieve the desired accuracy. The query optimization that requires frequent selectivity estimations cannot use this method due to its high performance overheads. The histogram method is the most common non-parametric method. The histogram method divides the data distribution into a set of small disjoint intervals, in other words, buckets, to approximate the data distribution, and stores some statistics in each bucket such as value range and the number of data in a bucket. The histogram method is based on the uniform distribution assumption which means that data in a bucket are uniformly distributed. The selectivity estimation using a histogram is as follows: First, all buckets overlapping with the query are selected. The statistics in each bucket is used to compute the number of data that satisfy the query. The numbers of the satisfied data from each bucket are summed up to get the final estimation result.

The histogram method is practically the most preferable among the ones in four classes because it is possible to make a histogram that approximates any data distribution with reasonably small storage and low error rates. Therefore it is widely used in many commercial databases. The histogram method is again classified into various methods according to how to partition the data distribution into buckets in order to minimize the estimation error: the Equi-width, the Equi-depth, the MaxDiff, the V-optimal method, etc. In the Equi-width, the widths of the buckets are equal, and the number of data in each bucket approximates the data distribution. In the Equi-depth, each bucket has the same number of data, so the widths of the buckets are different. In the MaxDiff, there is a bucket boundary bewteen two adjacent values when the difference of these values are among the largest. In the V-optimal, the sum of weighted variances of buckets is minimized. The V-optimal method has been shown to be the most accurate histogram method [IP95, JKMPSS98].

2.2 Multi-dimensional Selectivity Estimation

The optimization of fuzzy queries for multimedia repositories needs a multi-dimensional selectivity estimation technique. Chaudhuri[CG96] used the result using the correlation fractal dimension [BF95] as the selectivity estimation. However, the selectivity using the correlation fractal dimension is the average of the estimation results for the same shape and size queries and can be practically used in two and three dimensions. For queries with multiple attributes, there is an estimation method that uses a multi-dimensional file organization called the multilevel grid file (MLGF) [WKW94]. MLGF partitions the multi-dimensional data space into several disjoint nodes, called grids, that act as histogram buckets. A new field, count, is added to each grid node for saving the number of data in the grid. The selectivity is estimated by accessing grid nodes overlapping with a query. This method supports dynamic data updates because MLGF itself is a dynamic access method. And it accurately estimates the result size of a query. However, MLGF suffers from the dimensionality curse [BBK98] that means severe performance degradation in high dimensions. Also the method has the maintenance overhead of MLGF. So, the method can not be applied in dimensions higher than three.

Recently, Poosala et al. proposed several useful methods for the multi-dimensional selectivity estimation [PI97]: The Singular Value Decomposition (SVD), the Hilbert numbering, the PHASED, and the MHIST methods. These methods are based on the 1-dimensional histogram method under the assumption that the histogram can also be used in the multi-dimensional selectivity estimation. So, these methods partition the joint data distribution into disjoint buckets. The SVD method decompose the joint data distribution matrix *J* into three matrices *U, D*, and *V* that satisfy $J=UDV^T$. Large magnitude diagonal entries of the diagonal matrix *D* are selected together with their pairs, left singular vectors from *U* and right singular vectors from *V*. These singular vectors are partitioned using any one-dimensional histogram method so as to be used as histogram buckets of the attributes. There are many efficient SVD algorithms, but the SVD method can be used only in two dimension. The Hilbert numbering method converts the multi-dimensional joint data distribution into the 1-dimensional one and partitions it into several disjoint histogram buckets using any one-dimensional histogram method. The buckets made by this method may not be rectangles. Therefore, it is difficult to find the buckets that overlap with a query. The estimates may be inaccurate because it does not preserve the multi-dimensional proximity in 1-dimension. The PHASED method partitions an n-dimensional space along one dimension chosen arbitrarily by any one-dimensional histogram method, and repeats this until all dimensions are partitioned. The MHIST is an improvement to the PHASED method. It selects the most important dimension in each state and partitions it. From the V-optimal point of view as an applied partitioning method in MHIST, the dimension that has the largest variance is the most important dimension. The experiments in [PI97] showed that MHIST technique is the best among a variety of multi-dimensional histogram techniques. However, even though it produces low error rates in 2-dimensional cases, it has relatively high error rates in the 3-dimensional space (20-30 %) and the 4-dimensional space (30-40%). This demonstrates that it is not easy to segment multi-dimensional spaces into disjoint histogram buckets efficiently. These methods cannot be used in dimensions higher than three. In addition, the database system must reconstruct the statistics periodically in an environment where data is updated frequently because the method do not support dynamic data updates.

3. DISCRETE COSINE TRANSFORM

The discrete cosine transform has been widely used in the image and signal processing areas usually in the 2-dimensional domain because it has the power to compress information. However, we should use the multi-dimensional DCT for compressing the histogram information. Therefore, we briefly describe the definition of the 1-dimensional DCT, the 2-dimensional DCT and extend them to the multi-dimensional DCT.

3.1 Definition of Discrete Cosine Transform

For a series of data $\vec{F}$ = (*f*(0),*f*(1),…,*f*(*N*-1)), DCT coefficients, $\vec{G}$ = (*g*(0),*g*(1),…,*g*(*N*-1)), are defined as follows:

$$g(u)=\sqrt{\frac{2}{N}}k_u\sum_{n=0}^{N-1}f(n)\cos\left(\frac{(2n+1)u\pi}{2N}\right)$$

$$k_u=\begin{cases}\dfrac{1}{\sqrt{2}} & \text{for } u=0\\ 1 & \text{for } u\neq 0\end{cases}\qquad ,\ u = 0,\ldots,N\text{-}1$$

$\vec{F}$ = (*f*(0),*f*(1),…,*f*(*N*-1)) is recovered by the inverse DCT defined as follows:

$$f(n)=\sqrt{\frac{2}{N}}\sum_{u=0}^{N-1}k_u g(u)\cos\left(\frac{(2n+1)u\pi}{2N}\right),\ n=0,\ldots,N\text{-}1$$

1-dimensional DCT was extended to 2-dimensional DCT as follows: Let $[F]_2$ be an $M\times N$ matrix representing the 2-dimensional data and $[G]_2$ be the 2-dimensional DCT coefficients of $[F]_2$. Then the element (*u*,*v*) of $[G]_2$ is given by

$$g(u,v)=\frac{2k_u k_v}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}f(m,n)\cos\left[\frac{(2m+1)u\pi}{2M}\right]\cos\left[\frac{(2n+1)v\pi}{2N}\right]$$

where *u* = 0,…,*M*-1 and *v* = 0,…,*N*-1

By the separability property [RY90, Lim90] of the 2-dimensional DCT, *g*(*u*,*v*) can be rewritten as follows:

$$g(u,v)=\sqrt{\frac{2}{M}}k_u\sum_{m=0}^{M-1}\left\{\sqrt{\frac{2}{N}}k_v\sum_{n=0}^{N-1}f(m,n)\cos\left[\frac{(2n+1)v\pi}{2N}\right]\right\}\cos\left[\frac{(2m+1)u\pi}{2M}\right]$$

Its inverse is as follows:

$$f(m,n)=\sqrt{\frac{2}{M}}\sum_{u=0}^{M-1}k_u\left\{\sqrt{\frac{2}{N}}\sum_{v=0}^{N-1}k_v g(u,v)\cos\left[\frac{(2n+1)v\pi}{2N}\right]\right\}\cos\left[\frac{(2m+1)u\pi}{2M}\right]$$

Now we generalize the above to the *k*-dimensional DCT recursively as follows:

Let $[F]_k$ be $N_1\times N_2\times\ldots\times N_k$ *k*-dimensional data. Let $u(t)=(u_1,\ldots,u_t)\subseteq(u_1,\ldots,u_k)$ and $n(t)=(n_1,\ldots,n_t)\subseteq(n_1,\ldots,n_k)$ for $1<t\leq k$ and $u_i = 0,\ldots,N_i\text{-}1$, $n_i = 0,\ldots,N_i\text{-}1$ for $1\leq i\leq k$. Let $[G]_k$ be DCT coefficients of $[F]_k$. We define *G*(*u*(*t*)) , *F*(*u*(*t*)) as follows:

$$G(u(t))=\sqrt{\frac{2}{N_t}}k_{u_t}\sum_{n_t=0}^{N_t-1}G(u(t-1))\cos\left(\frac{(2n_t+1)u_t\pi}{2N_t}\right)$$

$$G(u(1))=\sqrt{\frac{2}{N_1}}k_{u_1}\sum_{n_1=0}^{N_1-1}f(n_1,\ldots,n_k)\cos\left(\frac{(2n_1+1)u_1\pi}{2N_1}\right)$$

$$F(n(t))=\sqrt{\frac{2}{N_t}}\sum_{u_t=0}^{N_t-1}k_{n_t}F(n(t-1))\cos\left(\frac{(2n_t+1)u_t\pi}{2N_t}\right)$$

$$F(u(1))=\sqrt{\frac{2}{N_1}}\sum_{n_1=0}^{N_1-1}k_{n_1}g(u_1,\ldots,u_k)\cos\left(\frac{(2n_1+1)u_1\pi}{2N_1}\right)$$

Then, *k*-dimensional DCT coefficients is given by $g(u_1,\ldots,u_k)$ = *G*(*u*(*k*)). And the inverse DCT transform is given by $f(u_1,\ldots,u_k)$ = *F*(*u*(*k*)).

3.2 Properties of Discrete Cosine Transform

DCT has many desirable properties as follows:

(1) DCT is a linear transform. Let F_C be DCT and α,β be the scalar values, and let *x*,*y* be the general *k*-dimensional data. Then the following linearity holds:

$$F_C(\alpha x+\beta y)=\alpha F_C(x)+\beta F_C(y)$$

(2) DCT is separable. This means that the 2-dimensional DCT can be reduced to the 1-dimensional DCT which enables the row-column decomposition which is the basis of fast algorithms.

(3) DCT preserves the energy in the transformed domain as Parseval's theorem says that

$$\sum\nolimits_{(n_1,\ldots,n_k)}\left|f(n_1,\ldots,n_k)\right|^2=\sum\nolimits_{(u_1,\ldots,u_k)}\left|g(u_1,\ldots,u_k)\right|^2$$

n_i , $u_i = 0,\ldots,N_i\text{-}1$, $i=1,\ldots,k$

(4) DCT has the property of energy compaction. DCT reduces the correlation among transformed coefficients. This property is related to the energy compaction. That is, if data adjacent to each other in the data distribution are highly correlated, DCT can reduce the correlation between adjacent transformed coefficients. And if the frequency spectrum of a data distribution is skewed in which the magnitudes of low frequency coefficients are large while those of high frequency coefficients are small, we can discard the high frequency coefficients without seriously affecting the original dada distribution [AFS93]. Since discarding the high frequency coefficients causes an error, we measure this error as the mean square error (MSE).

$$MSE = \sum_{(n_1,\ldots,n_k)} (f(n_1,\ldots,n_k) - f^*(n_1,\ldots,n_k))^2$$

$$n_i = 0,\ldots,N_i\text{-}1\ ,\ i=1,\ldots,k$$

where $f^*(n_1,\ldots,n_k)$ is computed by applying the inverse DCT with truncated DCT coefficients.

There are many other transforms such as the discrete Fourier transform (DFT), the Harr transform, the Hadamard Transform, and the Karhunen Loeve Transform (KLT). They differ in energy compaction and in computational requirements. From the energy compaction point of view, KLT is the best transform. That is, KLT is the transform that minimizes the MSE for truncated coefficients. However KLT has a serious practical problem. There is no computationally efficient algorithm for KLT. However, DCT has a good energy compaction property as well as computationally efficient algorithms. Also the energy compation power of DCT is superior to all other transforms except KLT [RY90,Lim90]. Therefore DCT is most widely used in various applications. Typical applications of DCT are the visual telephony and the joint photographic expert group (JPEG).

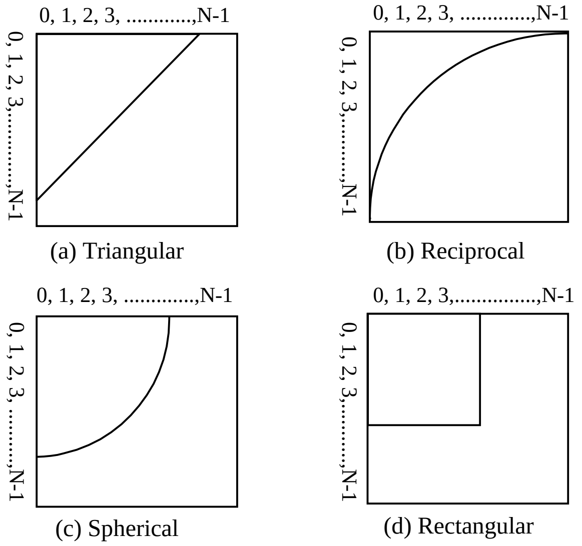
4. SELECTIVITY ESTIMATION USING DISCRETE COSINE TRANSFORM

As explained in Section 1 and 2, a histogram method cannot be directly used in the multi-dimensional selectivity estimation. As alternatives, we can consider parametric and curve-fitting methods. The former has the same constraint in a multi-dimensional space as in the 1-dimensional space, that is, the model function should fit the data distribution in some degree. When the constraint does not hold, the accuracy degrades. The latter uses a polynomial function for fitting a curve. But it uses an independent variable for every dimension and the number of coefficients in a multi-variable polynomial function increases rapidly as the dimensionality increases. It also suffers from the problems of the oscillation (negative values) and rounding errors.

We propose a curve-fitting method using DCT. In this method we use a uniform grid as histogram buckets in a multi-dimensional space. From now, this grid is called a uniform histogram bucket. In case a data distribution is highly correlated, DCT makes it possible for a few data items to represent the whole data by compressing information of the data distribution. We also can get the original distribution by the inverse transformation with low error rates. This method solves the problem of the high storage overheads and higher error rates in high dimensional spaces, since it uses a large number of small-sized multi-dimensional histogram buckets while compressing information from histogram buckets. There are various considerations to estimate the multi-dimensional selectivity by using DCT: coefficients sampling, data distribution, DCT computation and maintenance, and selectivity computation. First, we consider the efficient sampling method to select low-frequency coefficients that have large values. Second, we describe what is the constraint of the data distribution to compress the histogram information efficiently. Third, we explain how to support dynamic data updates to reflect it to the statistics immediately. Fourth, we describe how to simply calculate the selectivity estimation.

4.1 Geometrical Zonal Sampling

The size of the histogram bucket should be maintained small enough to get a low error rate in high dimensionality. The number of DCT coefficients transformed, however, increases exponentially as the dimensionality increases. If we choose appropriate coefficients after all coefficients are computed, it causes a severe computation overhead. Therefore, we must choose and compute only the coefficients that are estimated to have large values. To select the appropriate DCT coefficients, we use the 2-dimensional geometric zonal sampling technique that is used frequently in the area of digital signal processing [RY90, Lim90] and extend it to a multi-dimensional technique. Only those transformed coefficients within a specified zone are processed, with the remaining ones set to zero. This selection corresponds to low frequency filtering. There are several zonal sampling techniques: The triangular, the reciprocal, the spherical, and the rectangular zonal sampling. Fig.1 (a)~(d) shows only 2-dimensional cases of 4 geometrical zonal sampling methods for easy visualization. The triangular method is to select the coefficients within the triangle in a 2-dimensional case as shown in Fig.1(a). It selects DCT coefficients, $g(u_1,u_2)$, such that the sum of u_1 and u_2 is less than or equal to a given value b, that is, $u_1+u_2\leq b$ for $u_1=0,\ldots,N_1\text{-}1$ and $u_2=0,\ldots,N_2\text{-}1$. In a multi-dimensional case, it selects DCT coefficients, $g(u_1,\ldots,u_n)$, such that $\sum_{i=1}^{n} u_i \leq b$ for $u_i = 0,\ldots, N_i\text{-}1$. We know the number of DCT coefficients by this sampling with lemma 1.


0, 1, 2, 3, ...,N-1
0, 1, 2, 3,...........,N-1
0, 1, 2, 3, ...,N-1
0, 1, 2, 3,...........,N-1
0, 1, 2, 3, ...,N-1
0, 1, 2, 3, ...,N-1
0, 1, 2, 3,...............,N-1
0, 1, 2, 3,...........,N-1

(a) Triangular (b) Reciprocal

(c) Spherical (d) Rectangular

Fig 1. Geometrical Zonal Sampling in 2-dimensional case

Lemma 1) The number of DCT coefficients selected by the triangular zonal sampling is given by $_{n+b}C_{min(n,b)}$, if the condition

$b \le N_i$ is satisfied.

Table 1 shows various values of n and b.

	b=1	*b*=2	*b*=3	*b*=4	*b*=5	*b*=6
n=1	${}_2C_1=2$	${}_3C_1=3$	${}_4C_1=4$	${}_5C_1=5$	${}_6C_1=6$	${}_7C_1=7$
n=2	${}_3C_1=3$	${}_4C_2=6$	${}_5C_2=10$	${}_6C_2=15$	${}_7C_2=21$	${}_8C_2=28$
n=3	${}_4C_1=4$	${}_5C_2=10$	${}_6C_3=20$	${}_7C_3=35$	${}_8C_3=56$	${}_9C_3=84$
n=4	${}_5C_1=5$	${}_6C_2=15$	${}_7C_3=35$	${}_8C_4=70$	${}_9C_4=126$	${}_{10}C_4=210$
n=5	${}_6C_1=6$	${}_7C_2=21$	${}_8C_3=56$	${}_9C_4=126$	${}_{10}C_5=252$	${}_{11}C_5=462$
n=6	${}_7C_1=7$	${}_8C_2=28$	${}_9C_3=84$	${}_{10}C_4=210$	${}_{11}C_5=462$	${}_{12}C_6=924$

Table 1. The number of DCT coefficients selected by the triangular zonal sampling

The reciprocal method is to select the coefficients such that the multiplication of indices is less than or equal to a given value *b*. That is, the selection is made by the constraint $\prod_{i=1}^{n}(u_i+1) \le b$ for $u_i = 0,..,N_i-1$. This method chooses more high-frequency values in each dimension than the previous method. The spherical zonal sampling method is to select the coefficients such that the sum of the square of indices is less than or equal to a given value *b*, that is, $\sum_{i=1}^{n} u_i^2 \le b$ for $u_i = 0,\ldots,N_i-1$. It chooses the coefficients within the area of a circle in the 2-dimensional case and a sphere in the 3-dimensional case. The rectangular zonal sampling method chooses the coefficients such that the maximum value of indices is less than or equal to a given value *b*, that is, $\max(u_1, u_2, ..., u_n) \le b$ for $u_i = 0,\ldots,N_i-1$. It chooses the coefficients within the area of a rectangle.

Table 2 shows the sampling ratio of each zonal sampling methods. As the dimensionality increases, the number of coefficients chosen by the triangular zonal sampling and the reciprocal zonal sampling increases slowly, while the total number of histogram buckets increases explosively. However, the number of selected coefficients by the spherical and rectangular zonal sampling method increases somewhat rapidly.

4.2 Data Distributions

In order to be able to compress a great number of histogram buckets into a small amount of information with low estimation error rates by using DCT, the data distribution should have certain characteristics. The distribution should have high correlation among data items. That is, the frequency spectrum of the distribution should show large values in its low frequency coefficients and small values in its high frequency coefficients [AFS93]. If the data distribution does not follow the above characteristics, that is, data are totally independent of adjacent data, we cannot have the benefits of energy compaction and cannot reduce the number of coefficients without distorting the original data distribution. We believe that data in a real data distribution are highly correlated. There are many cases that data are correlated. It is natural for the joint data distribution of multiple attributes from a relation to have clusters in most cases, since the attributes are in general closely dependent each other [PI97]. Actually in the areas like data mining, the techniques to find such clusters are practically used for extracting useful knowledge from a large volume of databases [GRS98, EKSWX98, ZRL96, NH94]. The clustering effect can also be seen in multimedia databases like images and in spatial databases [EKSX96, SCZ98]. The large-sized shapes of a cluster correspond to large-valued low frequency coefficients while small-sized variations in it correspond to small-valued high frequency coefficients. Therefore, the mean square error between the actual data distribution and the distribution recovered by selected low coefficients is usually small. Based on these observations, we can reduce the number of multi-dimensional histogram buckets remarkably. In general, as the skewness of data distributions grow or the number of clusters increases, the number of large-valued high frequency coefficients tends to increase. It means more coefficients are needed to keep low error rates.

4.3 Dynamic Data Update

It is important to reflect dynamic data updates to the statistics for estimating selectivity immediately in the environment where data are frequently inserted or deleted. Except the MLGF method, most of multi-dimensional selectivity estimation techniques, such as MHIST, SVD, PHASED, and Hilbert numbering, cannot reflect dynamic data updates into the histogram immediately. In other words, when the number of data updates reaches a certain threshold, the histogram should be reconstructed entirely. In

			# of selected coefficients (% ratio to # total buckets)			
dim	N_i	# of total buckets	Triangular	Reciprocal	Spherical	Rectangular
			b=6	*b*=14	*b*=22	*b*=3
2	50	2500	28(1.1%)	41(1.6%)	22(0.44%)	16(0.64%)
3	25	15625	84(0.54%)	86(0.56%)	87(0.56%)	64(0.41%)
4	15	50625	210(0.41%)	153(0.3%)	305(0.6%)	256(0.51%)
5	10	100000	462(0.46%)	226(0.23%)	973(0.97%)	1024(1%)
6	8	262114	924(0.35%)	333(0.13%)	2882(1.1%)	4096(1.6%)
7	7	823543	1716(0.21%)	477(0.058%)	8080(0.98%)	16384(2%)
8	6	1679616	3003(0.18%)	601(0.036%)	21772(1.3%)	65536(3.9%)

Table 2. The ratio of the number of selected coefficients by the zonal sampling to the total number of uniform histogram buckets

Selectivity of a query $q_k = \int_{a_k}^{b_k} ... \int_{a_2}^{b_2} \int_{a_1}^{b_1} f(x_1, x_2, ..., x_k) dx_1 dx_2 ... dx_k$ (1)

$$\approx \sqrt{\frac{2}{N_1}} ... \sqrt{\frac{2}{N_k}} \sum_{g(u_1,...,u_k)\in Z} k_{u_1} ... k_{u_k} g(u_1,...,u_k) \int_{a_1}^{b_1} \cos(u_1 \pi x_1) dx_1 .... \int_{a_k}^{b_k} \cos(u_k \pi x_k) dx_k \qquad (2)$$

contrast, our proposed method can reflect dynamic data updates to the statistics for estimating the selectivity with reasonable overheads. This is enabled because DCT is a linear transform. Its process is as follows: When data is newly inserted, the values of its DCT coefficients are computed and added into existing DCT coefficients. In case of deletion, the values of DCT coefficients of the deleted data are computed and subtracted from existing DCT coefficients. Therefore, we can immediately reflect data insertions and deletions into the statistics for estimating the selectivity by processing only the update data.

Example 1) We show an example for a 2-dimensional case. Let $[F]_2$ be the current uniform histogram buckets and $[G]_2$ be the current DCT coefficients of $[F]_2$. Let $[F']_2$ be some data updates which represents that one data in (0,1) and two data in (1,2) are deleted and two data in (2,0) are newly added. And let $[G']_2$ be DCT coefficients of $[F']_2$. Let $[F'']_2$ be the final uniform histogram buckets and $[G'']_2$ be final DCT coefficients of $[F'']_2$. Then $[F'']_2 = [F]_2 + [F']_2$ and $[G'']_2 = [G]_2 + [G']_2$.

$$[F]_2 = \begin{pmatrix} 10 & 15 & 13 \\ 14 & 20 & 16 \\ 9 & 13 & 11 \end{pmatrix} \xrightarrow{DCT} [G]_2 = \begin{pmatrix} 40.333 & -2.858 & -5.421 \\ 2.041 & -0.500 & -0.289 \\ -6.835 & -0.289 & 1.167 \end{pmatrix}$$

$$[F']_2 = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -2 \\ 2 & 0 & 0 \end{pmatrix} \xrightarrow{DCT} [G']_2 = \begin{pmatrix} -0.333 & 1.633 & 0.471 \\ -1.225 & -1.000 & 0.000 \\ 1.179 & -0.577 & 1.333 \end{pmatrix}$$

$$[F'']_2 = \begin{pmatrix} 10 & 14 & 13 \\ 14 & 20 & 14 \\ 11 & 13 & 11 \end{pmatrix} \xrightarrow{DCT} [G'']_2 = \begin{pmatrix} 40.000 & -1.225 & -4.950 \\ 0.816 & -1.500 & -0.289 \\ -5.657 & -0.866 & 2.500 \end{pmatrix}$$

4.4 Selectivity Estimation of Range Queries

There are two kinds of methods to compute the selectivity of a range query. The first method finds all histogram buckets within the query range using the inverse DCT, and then computes the selectivity as the histogram method does. It assumes the uniform data distribution within a bucket like the existing histogram methods. The second method computes the selectivity using the integral of the inverse DCT function since the function is a continuous cosine function. The former method needs the inverse DCT computation for each bucket information while the latter simply computes the selectivity without the computation for each bucket information count since it computes the integral of the inverse DCT function only for the interval of the query range. Since the inverse DCT function naturally supports the continuous interpolation between contiguous histogram buckets, the second method provides accurate results. The following is the expression of the integral to estimate the selectivity of a range query.

First, we show the 2 dimensional case and generalize it to the *k*-dimensional case. Let q_2 be a 2-dimensional query. The range of q_2 is $a \le x \le b$, $c \le y \le d$, which is represented as (a~b, c~d). We assume the data space is normalized as $(0,1)^n$. The *x* coordinate is divided into *N* partitions and *y* coordinate is divided into *M* partitions. Then *i*'th positions of *x*,*y* (x_i and y_i) are as follows:

$$x_i = \frac{2i+1}{2N}, y_i = \frac{2i+1}{2M}$$

Then we can rewrite the inverse DCT function *f*(*m*,*n*) in section 3.1 as follows:

$$f(x,y) = \sqrt{\frac{2}{M}} \sum_{u=0}^{M-1} k_u \left\{ \sqrt{\frac{2}{N}} \sum_{v=0}^{N-1} k_v g(u,v) \cos(xv\pi) \right\} \cos(yu\pi)$$

Selectivity of a query $q_2 = \int_c^d \int_a^b f(x,y) dxdy$

$$= \int_c^d \int_a^b \sqrt{\frac{2}{M}} \sum_{u=0}^{M-1} k_u \left\{ \sqrt{\frac{2}{N}} \sum_{v=0}^{N-1} k_v g(u,v) \cos(xv\pi) \right\} \cos(yu\pi)\, dxdy$$

$$= \int_c^d \sqrt{\frac{2}{M}} \sum_{u=0}^{M-1} k_u \left\{ \int_a^b \sqrt{\frac{2}{N}} \sum_{v=0}^{N-1} k_v g(u,v) \cos(xv\pi) dx \right\} \cos(yu\pi) dy$$

$$\approx \sqrt{\frac{2}{M}} \sqrt{\frac{2}{N}} \sum_{g(u,v)\in Z} k_u k_v g(u,v) \int_c^d \cos(u\pi y) dy \int_a^b \cos(v\pi x) dx$$

where *Z* is the set of selected coefficients from zonal sampling

Now, we generalize the above integral to the *k*-dimensional case. Let q_k be a *k*-dimensional range query. The range of the query q_k is $a_i \le x_i \le b_i$ for $1 \le i \le k$, which is represented as (a_1~b_1,…,a_k~b_k). The x_i coordinate is devided into N_i partitions. Then the selectivity is expressed as formula (1), (2).

5. EXPERIMENTAL EVALUATION

In order to measure the accuracy of the proposed method in estimating the result sizes of queries, we conducted comprehensive experiments over an environment containing various synthetic data distributions and various queries. All data are generated in the normalized data space $(0,1)^n$. We were not able to make detailed comparisons with the previous results[WKW94, PI97] because the existing methods showed high errors in high dimensions beyond 3 dimension. For example, MHIST shows somewhat high errors in the 3-dimension (20~30%) and the 4-dimension (30~40%), and the MLGF method cannot be used in dimensions higher than three.

Synthetic data are generated with 50K records which ranged from 2 to 10 dimensions. We generated data with various distributions:

1. Normal distribution : The data points follow $N(0,\sigma^2)$ where $\sigma = 0.4$ for 2~4 dimensions, $\sigma = 1.0$ for 5~10 dimensions.
2. Zipf distribution: The data points follow the Zipf distribution where z = 0.3 for 2~5 dimensions, z = 0.2 for 6~10 dimensions. The Zipf distribution is defined as follows:

$$f(i) = \frac{\frac{1}{i^z}}{\frac{1}{1^z} + \frac{1}{2^z} + \ldots + \frac{1}{N^z}} \quad \text{where } i = 1,2,\ldots..,N$$

3. Clustered distribution: 5~15 normal distributions are overlapped in a data distribution.

DCT coefficients are calculated as follows: A multi-dimensional space is partitioned into a large number of uniform histogram buckets such that the number of partitions in each dimension is the same as those of others. The total number of buckets is in proportion to the dimension'th power of the number of partitions in one dimension. In low dimensions, if the total number of buckets is not quite large, we read data sequentially and count the number of data in each bucket and store them in the array of main memory. Then we calculate only DCT coefficients that are selected by the zonal sampling using DCT. In high dimensions, since the number of buckets is very large, we cannot afford the memory space for counting the number of data in all buckets. So, we used an X-tree[BKK96] to get groups of data that are close to each other by accessing nodes of the X-tree. This enables to get the number of data in a small group of buckets at a time for calculating DCT coefficients.

The selectivity estimation method proposed in this paper is evaluated for range queries of the form $(a_1 \le X_1 \le b_1)$&…&$(a_n \le X_n \le b_n)$, where $0 \le a_i, b_i \le 1$. Four sets of 30 queries were made such that each set represents a different range of selectivity: large(≈0.3), medium(≈0.067), small(≈0.0067), very small (≈0.0013). There are two query models for the probability distribution of queries [PSTW93, BF95]: the random model, the biased model. The random model assumes that queries are uniformly distributed in the data space. That is, every part of data space is equally likely to be queried. The biased model assumes that queries are more highly distributed in high-density regions. That is, each data is equally likely to be queried. Most applications follow the latter model. For example, in GIS applications, users are not likely to query the area of a dessert but are likely to query populated areas like a city. In image database applications, most of users may browse the images from a database and pick up the most similar image that they want from the browsed images and search images similar to it. This means that queries are located more frequently in dense area in the data space. So, we adopt the biased model as a query model in these experiments. For each query, we generated 30 biased queries. The query results are compared with the estimations using the proposed method in this paper. A percentage error is used for the accuracy of an estimation result:

$$\text{Percentage error} = \frac{\left|\text{query result size - estimated result size}\right|}{\text{query result size}} \times 100\,\%$$

5.1 Storage Requirements and Selectivity Estimation Time

The proposed method requires the storage of the statistics for estimating the selectivity. The amount of the storage for the method is proportional to the number of DCT coefficients selected by zonal sampling. We convert the multi-dimensional indices of a DCT coefficient to an one-dimensional value and vice versa. Therefore, one DCT coefficient needs 4 bytes for storing its value and 4 bytes for storing its index. 8 bytes are required for storing one DCT coefficient. If one use 100 DCT coefficients for estimating the selectivity, 800 bytes and some book keeping bytes are required.

From the selectivity calculation formula (2), we can estimate the the selectivity computation time as follows: If k is the dimension and α is the time to compute the sine function, the time to compute the selectivity is given by $2*k*\alpha*$(*the number of selected DCT coefficients*). Table 3 shows the typical selectivity estimation time. In Sun Ultra II, α is measured as about 1 μ sec.

dimension	# DCT= 50	# DCT = 100	# DCT = 200
3	300 μ sec	600 μ sec	1.2 m sec
6	600 μ sec	1.2 m sec	2.4 m sec
9	900 μ sec	1.8 m sec	3.6 m sec

Table 3. The selectivity computation time in Sun Ultra II

It follows that the proposed method is efficient for time and space.

5.2 Effect of Zonal Sampling

The zonal sampling selects low frequency coefficients. That is, it acts as a low frequency filter. Its effectiveness can be measured by the mean square error. But this requires all values of uniform histogram buckets by the inverse DCT, which is a very time consuming job. So, instead we measure the effectiveness of the zonal sampling by percentage errors of queries. We make 30 queries for each test and averaged their results. The efficiency of the zonal sampling is affected by distributions. We made experiments for 3 different distributions in the 6-dimension: (1) Normal distribution (2) Zipf distribution (3) Clustered 15 distribution (that has 15 clusters). We apply the three zonal sampling methods to these data. We drop the rectangular zonal sampling in the 6-dimension because the number of selected DCT coefficients by rectangular zonal sampling increases very rapidly with a small b value as indicated in Table 2. The results are shown in Fig. 2~4. The results show that the reciprocal zonal sampling is the best for all distributions. The triangular zonal sampling method is the second. The spherical zonal sampling showed the worst performance. However, there are some threshold after which there is no difference between three zonal methods. Therefore, when we use a few DCT coefficients, the reciprocal zonal sampling is the best.

5.3 Effect of Dimension and Query Size

In Fig. 5~7, we show the results of various query sizes in various dimensions. Query sizes are large, medium, small, very small. The dimensions are varied as 2, 4, 6, 8, 10. The data distribution is the clustered 15 distribution. We use the reciprocal zonal sampling method as section 5.2 shows that the reciprocal zonal sampling is the best. Fig. 5 shows the results for using only 100 DCT coefficients. Fig. 6 for 500 DCT coefficients and Fig. 7 for 2000 DCT coefficients. As the dimension increases, the error rates increase slightly, but the average error of queries is below 10 %. This results show that the method in this paper can be used for high dimensional data spaces. As the query size is decreased, the error rates increase. This is a natural result because the percentage error is magnified by the slight difference between an estimation size and a query result size when the query result is small.

5.4 Effect of Data Distributions

The data distribution has impacts on the error rates for estimating the selectivity. Fig. 8~10 shows the results for various distributions. The Zipf is a skewed distribution. As the dimension increase, the skewness of the Zipf also increase exponentially. Therefore, the error rates increase. However, as expected, we verified the fact that the more we use DCT coefficients, the more accurate the results are. The error rates of the normal and the clustered 5 distributions increase very slightly. This means that the skewness of the normal and the clustered distribution increases very slightly as the dimension increases. In addition, since the clustered distribution is the most common phenomenon in many applications, the proposed method can be widely used in real world.

5.5 Effect of Data Space Partition

A multi-dimensional space is partitioned into a large number of uniform histogram buckets. The number of DCT coefficients is the same as that of the histogram buckets. But 2000 DCT coefficients that are selected by the triangular zonal sampling are computed and *sorted*. To show the effects of the number of partitioned buckets, we partition a multi-dimensional space into several different ways. The p in Fig. 11~14 means the number of partitions in one dimension. We find the average result size of 30 medium-size queries and estimate the size of the queries with only the indicated number of DCT coefficients in the X-coordinate (numDCT) in Fig. 11~14. Then we calculate percentage errors. We found some interesting facts. As the number of partitions (p) increases, the accuracy also increase. The more DCT coefficients we use for estimating the selectivity, the more accurate the result is. There is some threshold after which the accuracy is not changed. In 3 dimensional case, if p=15, the threshold of the number of DCT coefficients is 30 with less than 1% error. That is, it is sufficient to have 30 DCT coefficients for estimating the selectivity with low error rates.

6. CONCLUSION

In this paper, we proposed a novel approach for estimating the multi-dimensional selectivity. The histogram is not adequate in high dimensions because the desired high accuracy requires small-sized histogram buckets, however we have a tremendous storage overhead as the dimension increases. To solve this problem, we used the discrete cosine transform which is an information compression technique in order to compress the information of a large number of histogram buckets. We achieved the high accuracy by using small-sized buckets, and also low storage overhead by a small amount of compressed information. Extensive experiments showed the proposed method is superior to the previous ones with the following advantages:

(1) The previous methods could not support multi-dimensional selectivity estimation, particularly, more than three dimensions. But our method supports high dimensional selectivity estimation with high accuracy.

(2) Our method eliminates the periodical reconstruction of the statistics for estimating the selectivity because it can reflect dynamic data updates to the statistics immediately.

(3) Our method simply calculates the selectivity using the integral of cosine functions. It also calculates the estimation accurately because it naturally supports the interpolation between the adjacent buckets.

For the future research, we plan to investigate the selectivity estimation of the nearest neighbor query.

7. REFERENCES

[AFS93] R. Agrawal, C. Faloutsos, A. Swami. Efficient Similarity Search In Sequence Databases. *Foundations of Data Organizations and Algorithms Conference*, 1993.

[BBK98] S. Berchtold, C. Bohm, H. Kriegel. The Pyramid Technique: Towards Breaking the Curse of Dimensionality. *ACM SIGMOD Conference*, pp.142-153, 1998.

[BKK96] S. Berchtold, D. Keim, H. Kriegel. The X-tree: An Index Structure for High-Dimensional Data. *22th VLDB Conference*, pp. 28-39, 1996

[BF95] A. Belussi, C. Faloutsos. Estimating the Selectivity of Spatial Queries Using the ‘Correlation’ Fractal Dimension. *21th VLDB Conference*, 1995.

[CR94] C. Chen. N. Roussopoulos. Adaptive Selectivity Estimation Using Query Feedback. *ACM SIGMOD Conference*, pp. 161-172, 1994.

[CSZS97] W. Chang, G. Sheikholeslami, A Zhang, T. Syeda-Mahmood. Efficient Resource Selection in Distributed Visual Information Systems. *ACM Multimedia Conference*, pp. 203-213, 1997.

[CG96] S. Chaudhuri, L. Gravano. Optimizing Queries over Multimedia Repositories. *ACM SIGMOD Conference* pp. 91-102, 1996,.

[Chri83] S. Christodoulakis. Estimating record selectivities. *Information Systems Journal*, 8(2) , pp105-115, 1983.

[EKSWX98] M. Ester, H. Kriegel, J. Sander, M. Winner, X. Xu. Incremental Clustering for Mining in Data Warehousing Environment. *24th VLDB Conference*, pp. 323-333, 1998.

[EKSX96] M. Ester, H. Kriegel, J. Sander, X. Xu. A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *In Proc. 2nd Int. Conf. on Knowledge Discovering and Data Mining*, 1996.

[Fa96] R. Fagin. Combining Fuzzy Information from Multiple Systems. *In Proc. of the 5th ACM Symposium on Principles of Database Systems*, 1996.

[Fa98] R. Fagin. Fuzzy Queries in Multimedia Database Systems. *In Proc. of the 7th ACM Symposium on Principles of Database Systems*, pp. 1-10, 1998.

[GRS98] S. Guha, R. Rastogi, K. Shim. CURE: An Efficient Clustering Algorithm for Large Databases. *ACM SIGMOD Conference*, pp. 73-84, 1998.

[HNSS95] P.J. Haas, J.F. Naughton, S. Seshadri, and L. Stokes. Sampling based estimation of the number of distinct values of an attribute. *21th VLDB Conference*, 1995.

[Io93] Y. Ioannidis. Universality of Serial Histograms. *19th VLDB Conference*, pp. 256-267, 1993.

[IP95] Y. Ioannidis, V. Poosala. Balancing Optimality and Practicality for Query Result Size Estimation. *ACM SIGMOD Conference*, pp. 233-244, 1995.

[JKMPSS98] H. Jagadish, N. Kouda, S. Muthukrishnan, V. Poosala, K. Sevcik, T. Suel. Optimal Histograms with Quality Gurantees. *24th VLDB Conference*, pp. 275-286, 1998.

[Lim90] J.S. Lim. Two Dimensional Signal And Image Processing. *Prentice Hall*, 1990.

[MCS98] M.V. Mannino, P. Chu, and T. Sager. Statistical profile estimation in database systems. *ACM Computing Surveys*, 20(3), 1988.

[NH94] R. Ng, J. Han. Efficient and Effective Clustering Methods for Spatial Data Minig. *20th VLDB Conference*, 1994.

[PSTW93] B. Pagel, H. Six, H. Toben, P. Widmayer. Towards an Analysis of Range Query Performance in Spatial Data

Structures. *In Proc. of the 2nd ACM Symposium on Principles of Database Systems*, 1993.

[PIHS96] V. Poosala, Y.E. Ioannidis, P.J. Haas, E.J. Shekita. Improved Histograms for Selectivity Estimation of Range Predicates. *ACM SIGMOD Conference*, pp. 294-305, 1996.

[PI97] V. Poosala, Y.E. Ioannidis. Selectivity Estimation Without the Attribute Value Independence Assumption. *23th VLDB Conference*, pp. 486-495, 1997.

[RY90] K.R. Rao, P. Yip. Discrete Cosine Transform Algorithms, Advantages, Applications. *Academic Press*, 1990.

[SCZ98] G. Sheikholeslami, W. Chang, A. Zhang. Semantic Clustering and Querying Heterogeneous Features for Visual Data. *ACM Multimedia Conference*, pp. 3-12, 1998.

[SLRD93] W. Sun, Y. Ling, N. Rishe, and Y. Deng. An Instant and accurate size estimation method for joins and selections in a retrieval-intensive environment. *ACM SIGMOD Conference*, 1993.

[WKW94] K.Y. Whang, S.W. Kim, G. Wiederhold. Dynamic Maintenance of Data Distribution for Selectivity Estimation, *VLDB Journal*. Vol.3, No.1, pp. 29-51, 1994.

[ZRL96] T. Zhang, R. Ramakrishnam, M. Linvry. BIRCH: An Efficient Data Clustering Method for Very Large Databases. *ACM SIGMOD Conference*, pp. 103-114, 1996.

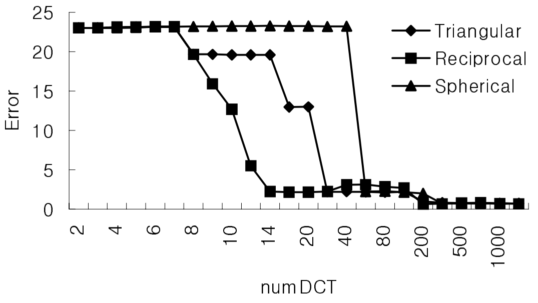


Fig. 2. Normal distribution, dimension=6, one-dimensional partition=10

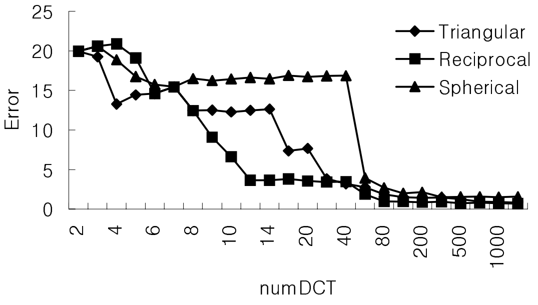


Fig. 3. Zipf distribution, dimension=6, one-dimensional partition=10

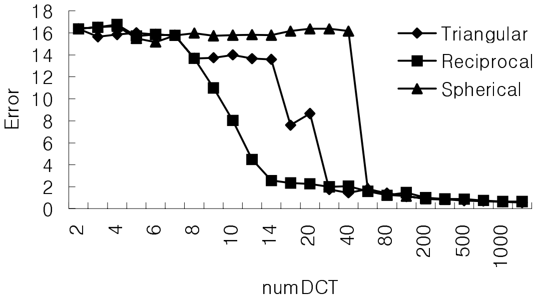


Fig. 4. Clustered 15 distribution, dimension=6, one-dimensional partition=10

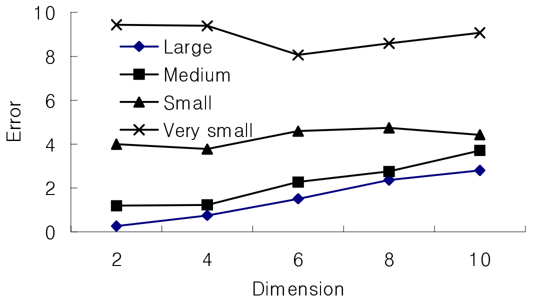


Fig. 5. Clustered 15 distribution, number of DCT coefficients = 100

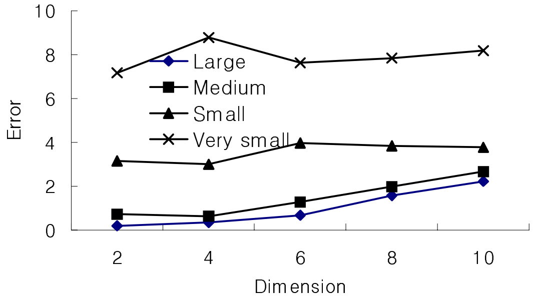


Fig. 6. Clustered 15 distribution, number of DCT coefficients = 500

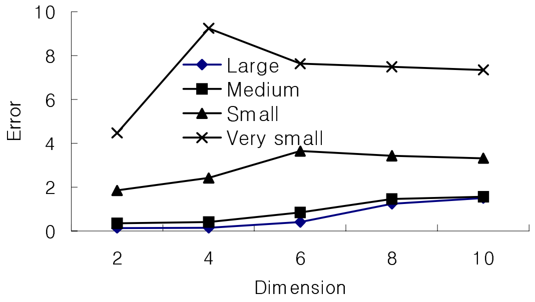


Fig 7. Clustered 15 distribution, number of DCT coefficients = 2000

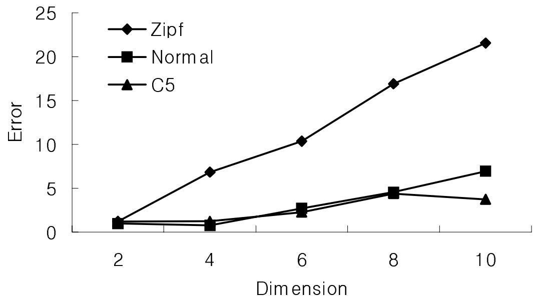


Fig. 8. number of DCT coefficients = 100

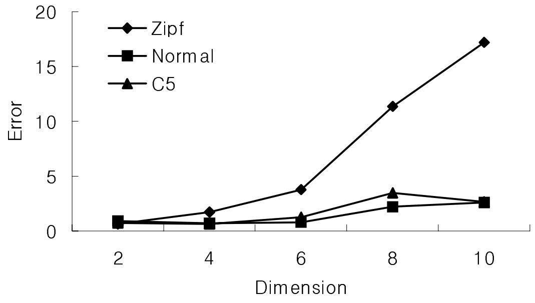


Fig. 9. number of DCT coefficients = 500

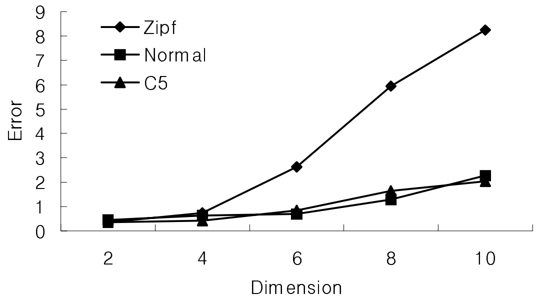


Fig. 10. number of DCT coefficients = 2000

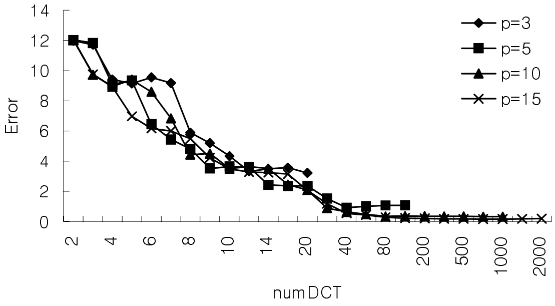


Fig. 11. dimension = 3, Clustered 5 distribution Query size = medium

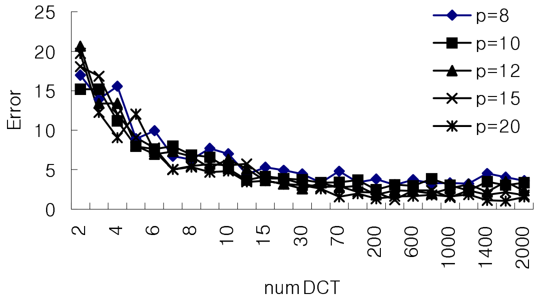


Fig. 12. dimension= 5, Clustered 5 distribution, Query size=medium

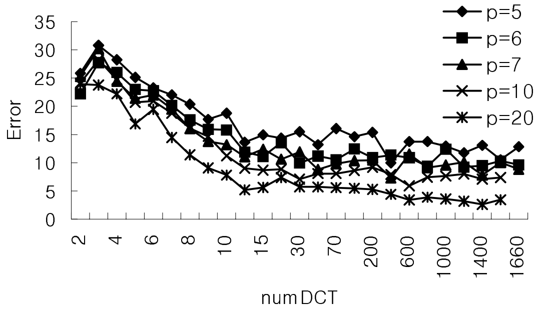


Fig. 13. dimension = 7, Clustered 5 distribution, Query size= medium

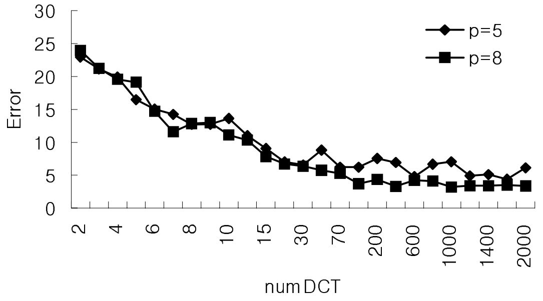


Fig. 14. dimension = 10, Clustered 5 distribution, Query size = medium