Similarity between Euclidean and cosine angle distance for nearest neighbor queries

Gang Qian[†]   Shamik Sural[‡]   Yuelong Gu[†]   Sakti Pramanik[†]

[†]Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824, USA
{qiangang,guyuelon,pramanik}@cse.msu.edu

[‡]School of Information Technology
Indian Institute of Technology
Kharagpur 721302, India
shamik@cse.iitkgp.ernet.in

ABSTRACT

Understanding the relationship among different distance measures is helpful in choosing a proper one for a particular application. In this paper, we compare two commonly used distance measures in vector models, namely, Euclidean distance (EUD) and cosine angle distance (CAD), for nearest neighbor (NN) queries in high dimensional data spaces. Using theoretical analysis and experimental results, we show that the retrieval results based on EUD are similar to those based on CAD when dimension is high. We have applied CAD for content based image retrieval (CBIR). Retrieval results show that CAD works no worse than EUD, which is a commonly used distance measure for CBIR, while providing other advantages, such as naturally normalized distance.

Keywords

Vector model, Euclidean distance, Cosine angle distance, Content based image retrieval, Inter-feature normalization

1. INTRODUCTION

Distance measure is an important part of a vector model. Among all distance measures that are proposed in the literature, some have very similar behaviors in similarity queries, while others may behave quite differently. Understanding the relationship among distance measures can help us to choose a proper distance measure for a particular application.

One way of comparing distance measures is to study their retrieval performance in terms of precision and recall in a particular application area, such as content-based image retrieval (CBIR) [18] and video copy detection [6]. One concern in choosing a particular distance measure is the impact of computational overhead on system performance. When feature vectors are large, some distance measures may consume more computing resources than the others. One possible approximation of EUD is proposed in [5]. On the other

hand, it is also important to choose a similarity measure that is consistent with human ideas of similarity. The authors of [17] have proposed a similarity measure based on noise distribution of the image database. In [16], a mathematical analysis of EUD and CAD has been done. It was shown that CAD has a special property to favor relatively larger component in a vector.

In this paper, we compare two commonly used distance measures in vector models, namely, Euclidean distance (EUD) and cosine angle distance (CAD), for nearest neighbor (NN) queries in high dimensional data spaces. From theoretical analysis and experimental results, we find that the retrieval results based on EUD are similar to those based on CAD. We use a high dimensional geometrical model to analyze how similar these two distance measures are under the assumption of uniform data distribution. We find that the *NN* of EUD is also ranked high by CAD when dimension is high. We define *NN* as the first nearest neighbor of the query. Our experimental results have corroborated the correctness of our model. We have also compared these two distance measures experimentally using normalized datasets and clustered datasets. Our conclusions are that:

1. In high dimensional data spaces, the NN query results by EUD and CAD are very similar.
2. For clustered data, the NN query results by EUD and CAD are more similar.
3. When vectors are normalized by its size, the NN query results by EUD and CAD are also more similar.

One application of the above properties is to combine features that are semantically different (e.g. color and texture) in CBIR as explained below. As EUD is often used as a distance measure for individual features in CBIR, inter-feature normalization is needed to combine EUD values of different scales into an over-all score for an image. Based on the property that NN query results of EUD and CAD are very similar in high dimensional spaces, we propose to use CAD instead of EUD for individual features. As CAD values are naturally normalized by norm, there is no need for further inter-feature normalization. Thus, the distance value from different features can be summed up directly as the final score for an image in the database. Our experimental results show that our proposal works not only no worse than other commonly used methods but also has some favorable advantages.

There are a number of methods proposed for combining features in CBIR. They can be divided into two categories:

rank based [8, 10] and distance value based [14, 15]. Rank based methods are also called "voting methods" in [12]. In rank based methods, the rank of an image for different features are calculated first, then these ranks are summed to derive the final rank of the image. Distance value based methods use distance value of individual features directly. Since distance values from different features have different scales, inter-feature normalization is necessary for computing the final rank of an image. One simple method in this category is to divide all distance values of a feature by the maximum distance value [15]. Another widely used method for combining features are based on the assumption that distance values have a Gaussian distribution [14]. Some discussion and comparisons of methods for combining features can be found in [3, 7, 14].

Using CAD as a normalized distance measure was mentioned in [4], but no analysis or experimental results were presented in the context of vector models or CBIR. We have done experiments to compare the CAD based method we proposed with two widely used methods, namely, rank based method by EUD and distance value based method with Gaussian assumption.

The rest of this paper is organized as follows. Section 2 presents the theoretical analysis of NN queries by EUD and CAD using a geometrical model in high dimensions. Section 3 and 4 present experimental results for comparing EUD and CAD and combining multiple features, respectively. Section 5 discusses the conclusion and future work.

2. THEORETICAL ANALYSIS OF EUD AND CAD FOR NN QUERIES

The similarity between EUD and CAD for NN queries can be measured by the average rank of the *NN* of EUD (represented as NN_e) in CAD. The two distance measures are considered similar if NN_e is also ranked high by CAD. The theoretical (probability) analysis compares EUD and CAD using a high dimensional geometrical model. A similar model has been used in [2] for the derivation of the cost model of high dimensional NN queries. Without losing generality, our analysis is based on a d-dimensional unit hypercube data space. We assume that data points are uniformly distributed within the space and there is no dependence between dimensions.

2.1 Notations and definitions

Table 1 is a summary of notations that we have used in this paper.

Explanation of some of the notations listed in Table 1 is given as follows:

1. The d-dimensional unit data space can be deemed as the Cartesian product $[0, 1]^d$. It also implies that every data point (vector) P in Ω has no negative element.

2. The value of $angle(P_1, P_2)$ is defined as follows:

$$angle(P_1, P_2) = \cos^{-1} \frac{\vec{P_1} \cdot \vec{P_2}}{\sqrt{(\vec{P_1} \cdot \vec{P_1})(\vec{P_2} \cdot \vec{P_2})}} \quad (1)$$

Since $angle(P_1, P_2)$ is defined based on CAD between P_1 and P_2 and has a better geometrical meaning than CAD, we use $angle(P_1, P_2)$ in place of CAD in the following discussion.

3. A hyper-cone $cone(P, \theta)$ for a given point P and angle θ is defined as follows:

The vertex of $cone(P, \theta)$ is the origin O of the unit space Ω. Let P be a point in Ω that is not O. Every point P' of $cone(P, \theta)$ satisfies $angle(P', P) \le \theta$. Figure 1 shows a 2-dimensional hyper-cone $cone(P, \theta)$, which is the Quadrangle $OABC$.
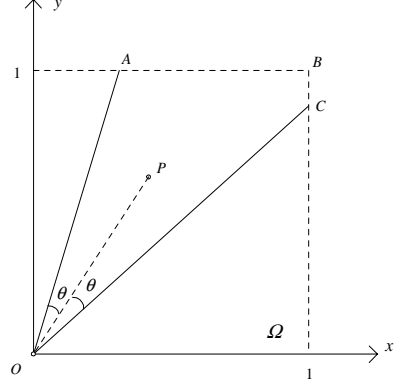

Figure 1: 2-dimensional hyper-cone $cone(P, \theta)$

4. A hyper-region is a close geometrical object such as a hyper-sphere or a hyper-cone.

2.2 Comparison of EUD and CAD

We first illustrate our approach to compare EUD and CAD using a 2-dimensional space. Figure 2 shows a 2-dimensional unit space Ω, where Q is a query point and
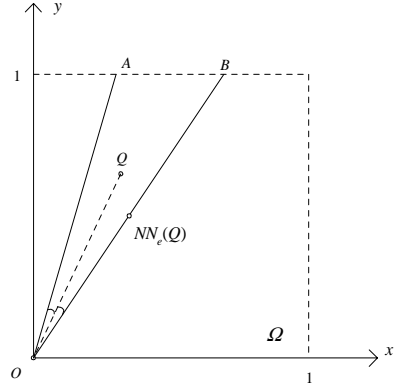

Figure 2: $NN_e(Q)$ **and the hyper-cone**

$NN_e(Q)$ is the nearest neighbor (i.e., the first nearest neighbor) of Q by EUD. Let $\triangle OAB$ be the hyper-cone $cone(Q, angle(Q, NN_e(Q)))$ with the property that $angle(Q, A)$ ($\angle QOA$) equals $angle(Q, B)$ ($\angle QOB$). Note that $\angle QOA$ and $\angle QOB$ correspond to the CAD between query Q and $NN_e(Q)$. It is clear that the rank of $NN_e(Q)$ in the NN query of Q by CAD is given by the number of data points within Hyper-cone $\triangle OAB$. The same observation can be extended to high-dimensional spaces, where the rank of $NN_e(Q)$ in the NN query of Q by CAD is determined by the number of data points within the hyper-cone $cone(Q, angle(Q, NN_e(Q)))$.

Under the assumption of uniform data distribution and based on the unit space Ω, the probability of a point existing in $cone(Q, angle(Q, NN_e(Q)))$ is equal to the volume

d	Number of dimensions
Ω	d-dimensional unit hyper-cube data space
$P/\vec{P}$	Data point/vector in Ω
O	Origin of Ω
N	Size of the dataset
$sp(C,r)$	d-dimensional hyper-sphere with center C and radius r
$ssp(C,r))$	Surface of a hyper-sphere with center C and radius r
$|P_1,P_2|_e$	EUD between points P_1 and P_2
$angle(P_1,P_2)$	Hyper-angle between points P_1 and P_2 with respect to O
$cone(P,\theta)$	Hyper-cone with vertex O, axis $\vec{P}$ and angle θ
$NN_e(Q)$	NN to a query point Q by EUD
$vol(R)$	(Hyper-)Volume of a hyper-region R

Table 1: Summary of notations

$vol(cone(Q, angle(Q, NN_e(Q))))$. Therefore, the expected number of data points within the hyper-cone is equal to the product of the size (N) of the dataset and the (hyper-)volume of the hyper-cone $cone(Q, angle(Q, NN_e(Q)))$.

The volume of a hyper-cone $cone(Q, \theta)$ is computed by integrating a piecewise function defined over data space Ω as follows:

$$vol(cone(Q,\theta)) = \int_{P \in \Omega} \left(\left\{ \begin{array}{ll} 1 & \text{if } angle(Q,P) \leq \theta \\ 0 & \text{otherwise} \end{array} \right\} \right) \mathrm{d}P \tag{2}$$

A good approximation of Equation 2 can be obtained by the Monte-Carlo method [9].

To estimate the expected number of data points within the hyper-cone $cone(Q, angle(Q, NN_e(Q)))$, we first calculate its expected volume by the following steps:

1. Suppose that the EUD $(|Q, NN_e(Q)|_e)$ between query point Q and $NN_e(Q)$ is r, the expected value of $vol(cone(Q, angle(Q, NN_e(Q))))$ equals the average volume of all hyper-cones given by Q and points that are on the surface $(ssp(Q,r))$ of the hyper-sphere $sp(Q,r)$. The situation is illustrated in a 2-dimensional data space in Figure 3, where Q is the query point, P is one of the points that are on $ssp(Q,r)$, and $\triangle AOB$ is the corresponding hyper-cone.
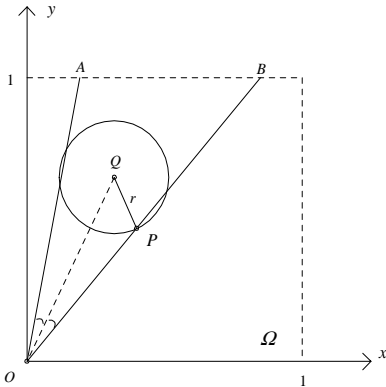

Figure 3: $sp(Q,r)$ **and the hyper-cone**

Thus for any given Q and r, based on the uniform distribution assumption, the expected volume satisfies the following function:

$$v(Q,r) = \int_{P \in (ssp(Q,r) \cap \Omega)} vol(cone(Q, angle(Q,P))) \, \mathrm{d}P \tag{3}$$

2. For a given query Q, the expected volume of $cone(Q, angle(Q, NN_e(Q)))$ can be obtained by integrating Equation 3 over all possible values of r as follows:

$$\begin{aligned} v(Q) &= \int_0^\infty v(Q,r) p_r(Q,r) \, \mathrm{d}r \\ &= \int_0^\infty (\int_{P \in (ssp(Q,r) \cap \Omega)} vol(cone(Q, \\ &\quad angle(Q,P)))\, \mathrm{d}P) p_r(Q,r)\, \mathrm{d}r \end{aligned} \tag{4}$$

In Equation 4, the function $p_r(Q,r)$ is the density function of r for a given query point Q. Note that r is the EUD between Q and $NN_e(Q)$. Following [2], for a given query point Q, the distribution function of r, $P_r(Q,r)$, is:

$$P_r(Q,r) = 1 - (1 - vol(sp(Q,r) \cap \Omega))^N \tag{5}$$

Note that N in Equation 5 represents the size of the dataset. The corresponding density function $p_r(Q,r)$ can be derived as follows:

$$\begin{aligned} p_r(Q,r) &= \frac{\partial}{\partial r} P_r(Q,r) \\ &= \frac{\partial}{\partial r} vol(sp(Q,r) \cap \Omega) \cdot N \\ &\quad \cdot (1 - vol(sp(Q,r) \cap \Omega))^{N-1} \end{aligned} \tag{6}$$

3. From Equation 4, for a given query Q, we can calculate the expected number of points in $cone(Q, angle(Q, NN_e(Q)))$ as $N \cdot v(Q)$. Thus the overall expected number of points in cone, i.e., the expected rank of NN_e of NN query by CAD, can be computed by averaging over all possible Q in Ω. Based on Equations 4 and 6, under the assumption of uniform data distribution, we obtain the following equation for a given N:

$$\begin{aligned} &\text{expected rank of } NN_e \text{ of NN query by CAD} \\ =\ & N \cdot \int_{Q \in \Omega} v(Q) \, \mathrm{d}Q \\ =\ & N \cdot \int_{Q \in \Omega} (\int_0^\infty (v(Q,r)) p_r(Q,r)\, \mathrm{d}r) \, \mathrm{d}Q \\ =\ & N^2 \cdot \int_{Q \in \Omega} (\int_0^\infty (\int_{P \in (ssp(Q,r) \cap \Omega)} vol(cone(Q, \\ &\quad angle(Q,P)))\, \mathrm{d}P) \cdot \frac{\partial}{\partial r} vol(sp(Q,r) \cap \Omega) \\ &\quad \cdot (1 - vol(sp(Q,r) \cap \Omega))^{N-1}\, \mathrm{d}r)\, \mathrm{d}Q \end{aligned} \tag{7}$$

From Equation 7, we have calculated the expected rank of NN_e of NN query by CAD at different dimensions using $N = 50,000$. As shown in Table 2, expected rank of NN_e by CAD increases drastically from dimension 2 to dimension 4, which shows that NN query results between EUD and CAD become similar even at lower dimensions. Note that as dimension gets even higher, EUD and CAD eventually becomes less similar again. However, the rate of decrease

d	2	4	8	16	32	64	128
rank	157	13.5	4.3	2.5	2.6	3.1	4.3

Table 2: Expected NN rank of NN_e by CAD at different dimensions

of similarity is very slow. Within a range of high dimensions, the claim of the similarity between EUD and CAD is reasonable. Our experimental results have corroborated the results of our theoretical analysis. In the following sections, we will also show that when vectors are normalized by size or clustered, the NN query results of EUD and CAD are even more similar in high dimensional spaces.

3. EXPERIMENTAL RESULTS FOR NN AND K-NN QUERIES

This section is divided into three subsections. The first subsection shows that the experimental results corroborate the results of our theoretical analysis. The second subsection shows the similarity between EUD and CAD with a different measure, i.e., the percentage of the same results (intersection) in the result sets of k nearest neighbor (k-NN) queries by EUD and CAD. The datasets used in this subsection include normalized as well as un-normalized data, uniformly distributed data, clustered data, and real image data. We summarize and discuss our experimental results in the third subsection.

3.1 Comparison of experimental and theoretical results

The experiments are conducted using a dataset of 50,000 randomly generated data points. The average rank of NN_e of NN query by CAD is computed based on 30 query points selected randomly from the dataset. Dimensions used in the comparison are 2, 4, 8, 16, 32, 64, and 128.

Figure 4 shows the comparison of experimental results with the theoretical results. Allowing for some statistical
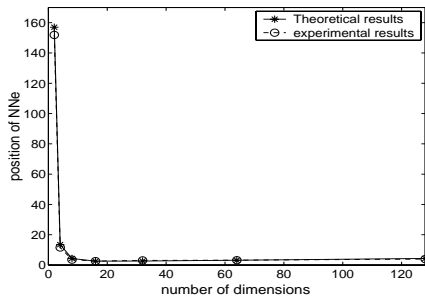

Figure 4: Theoretical and experimental results

precision error, Figure 4 shows that the results of theoretical analysis matches very well with those of the experiments. When dimension is low, the difference between EUD and CAD are very large. But when dimension gets higher, they become very similar. EUD and CAD become less similar again as dimension increases further. However, the rate of decrease of similarity is very slow.

3.2 Experimental results of kNN queries

k-NN queries are often used in real world applications. Thus, for different datasets, such as real world data, we

have done experiments to measure the similarity between EUD and CAD using the percentage of the same results (intersection) in the EUD answer set and CAD answer set (k-NN). Results of 10, 20, 100, 500, and 1000 NN queries are presented. If not specifically mentioned, experimental results presented in the following tables are obtained using datasets of 50,000 data points and 30 query points picked randomly from their corresponding datasets.

Table 3 shows experimental results based on random data.

k-NN	10	20	100	500	1000
			%		
$d = 2$	11	7.83	7.2	14.3	19.7
4	31	28.5	37.7	48.8	54.2
8	55	57	61.6	67.2	69.8
16	68	68	68.3	69.8	70.6
32	69.3	67.8	68.2	70.9	72.9
64	64.7	63.7	64.6	68.1	69.4
128	54.7	56.3	59	63.4	65.4

Table 3: Experimental results based on random data

As dimension gets higher than 8, more than 50 percent of the 10 NN query results of EUD and CAD are the same. The percentage of the intersection are even greater for larger k-NN queries from 20 NN to 1000 NN. Note that EUD and CAD eventually becomes less similar as dimension gets even higher (≥ 128). However, the rate of decrease of similarity is very slow.

Table 4 shows experimental results based on normalized

k-NN	10	20	100	500	1000
			%		
$d = 2$	100	100	100	99.9	99.7
4	95.7	95.5	96.3	96.1	96.1
8	96.3	95.2	95	95.2	95.1
16	91.3	94.8	94.4	93.6	93.6
32	90.3	92.8	91	91.4	91.7
64	89.7	88.2	89.7	90.2	90.7
128	87.3	88	86.9	89.4	89.9

Table 4: Experimental results based on normalized random data

random data. Normalization is an important process when vector model is applied for similarity queries. Its purpose is to normalize each element in a vector to be in the same range so that individual element gets the same weight when distance measures are applied. Depending on application, there are different methods for vector normalization such as those described in [1, 14]. In our experiment, vectors are normalized by their size, i.e., for each vector $v = <e_1, e_2, \ldots, e_d>$, its corresponding normalized vector is $v' = <e_1', e_2', \ldots, e_d'>$ where:

$$e_i' = \frac{e_i}{\sum_{j=1}^{d} e_j} \tag{8}$$

where $1 \leq i \leq d$. Table 4 shows that, after normalization, the EUD and CAD becomes very similar even for lower dimensions.

Table 5 shows experimental results based on clustered data with 50 clusters. Even for dimension as low as 4, the k-NN query results by EUD and CAD are very similar. We have also done experiments using datasets with different number of clusters. The results are similar to that of Table 5.

Table 6 shows experimental results based on real image data. The image dataset is generated from an image database

k-NN	10	20	100	500	1000
	%				
$d = 2$	15.7	15	16.2	21	27.7
4	93	85	69.2	80.6	84.4
8	86	87	79.3	89.4	92.7
16	91	91	89.1	97.4	99.1
32	93	97.2	89.9	98.4	100
64	92.3	90.8	93.9	99.2	100
128	91	90.7	98.2	99.7	100

Table 5: Experimental results based on clustered data (50 clusters)

of more than 30,000 color images. It contains 64-dimensional QBIC [13] color feature vectors. We can see from Table 6 that, for real data, the EUD and CAD are also very similar.

k-NN	10	20	100	500	1000
	%				
$d = 64$	78.7	76.7	78.0	78.1	78.1

Table 6: Experimental results based on real image data (QBIC)

3.3 Discussion

The vector model is used for an approximate generalization of the real world objects. The definition of "similar" is subjective and depend on the way feature vectors are generated. Based on the above theoretical analysis and experimental results, we consider EUD and CAD very similar when applied to NN queries in high dimensional data spaces. We explain this phenomena based on the norm of the vectors. When all vectors have the same norm, the NN query results by EUD and CAD will be exactly the same. Based on the assumption of uniform distribution, as dimension gets higher, the variance of the norms of the vectors in the dataset becomes smaller. As the norms become similar, EUD and CAD also become similar. For clustered data and normalized data in high dimensional spaces, the norms of the vectors are even more similar which causes EUD and CAD to behave more similarly.

4. EXPERIMENTAL RESULTS FOR COMBINING FEATURES

We have shown that when dimension is high, which is usually the case for a CBIR application, EUD and CAD are similar. EUD is widely used in CBIR. However, CAD has the unique property that the distance value is inherently normalized for a given feature. This makes combining semantically different features as easy as summing up the CAD values from different features. As an application for our analysis, we propose to use CAD measure instead of EUD for CBIR where multiple features are combined to create a single distance value. Experimental results for this analysis are presented in this section.

We use an image database of 6344 color images of animals and natural scenes. Two image features, color and texture, are used for image retrieval. The color feature is a 64-dimensional vector generated by QBIC [13]. The texture feature is a 24-dimensional vector generated using the algorithms proposed in [11]. 18 images of animals are chosen from the database as the query images due to their clear semantic meaning. The answer set of each query image is

decided solely by its semantic content, e.g., if the query image is a tiger, the answer image should also contain a tiger. Note that the purpose of this experiment is not to capture all possible semantics of the query image (e.g. tigers), but to show the effectiveness of the CAD-based method. The size of the answer set (relevant set) of each query image ranges from 4 to 40. We use recall and precision to measure the performance of each combining method. Figure 5 and Figure 6 show the average recalls and precisions of 10, 20, 50 and 100 NN query results. In Figures 5 and 6, "Rank Euclidean"
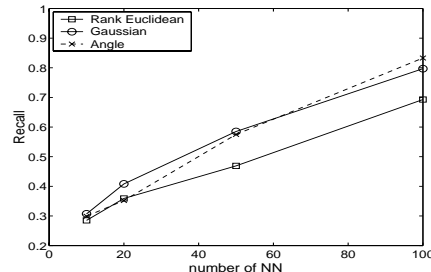

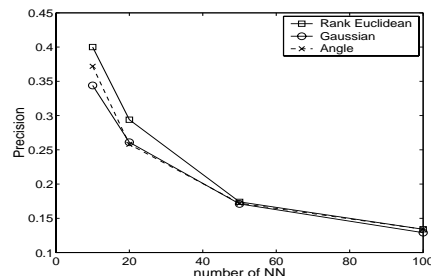**Figure 5: Recalls of different feature combining methods at various k-NN**

Figure 6: Precisions of different feature combining methods at various k-NN

is the rank based method using ranks of individual features by EUD. "Gaussian" is the distance value based method using Gaussian assumption [14]. For "Gaussian", EUD values computed for individual features are normalized using the following equation:

$$d' = \frac{d - m}{6\sigma} + \frac{1}{2} \qquad (9)$$

In Equation 9, d and d' are the original and normalized distance value, respectively. m and σ are the mean and standard deviation of pair-wise distances over all images in the database. Any value greater than 1 is considered as 1 in the experiments as described in [14]. "Angle" is the distance value based method we proposed, which uses CAD directly for inter-feature normalization.

Based on precision and recall, the performance of combining methods are similar, though "Rank Euclidean" is a little behind. As mentioned in [7], rank-based method may not be very effective since it does not directly use the distance value between the query and the retrieved image. The rank of retrieved images may give a false sense of similarity when actually the distance value may be very large. On the other hand, distance value based method using Gaussian assumption may not be effective if the distance distribution pattern among images in the database is not Gaussian. Another problem of this scheme is that it requires the mean and

<div align="center">Angle   Rank Euclidean</div>

Figure 7: CAD favors (retrieves) vectors with dominant component

variance of pair-wise distance values of the whole database. If the database is large and changes dynamically, the cost to maintain such value may be expensive. Thus we believe our simple CAD value based method for combining features is better compared to the two methods mentioned above. Moreover, it will not affect the results much to replace EUD by CAD as we have shown in the previous sections. Besides the benefit of simplicity, the CAD based method also has another special property, i.e., the CAD favors (retrieves) vectors with relatively larger component values [16]. This effect is illustrated in Figure 7, which shows the results of a 20 NN query. The first image at the upper-left corner is the query image. Since the query image has dominant color components blue and brown, nearly all 20 images returned by the CAD based method ("Angle") has blue and brown as their dominant color components. On the other hand, the rank based method by EUD ("Rank Euclidean") only returns about 10 such images out of 20 images.

5. CONCLUSION

Through our theoretical analysis and experimental results, we conclude that EUD and CAD are similar when applied to high dimensional NN queries. For normalized data and clustered data, EUD and CAD becomes even more similar. As an application of this inference, we use a simple CAD based method for combining features in CBIR. We have shown that the method we have proposed works no worse than some commonly used methods while possessing some favorable advantages.

In future work, we plan to extend our geometrical model to analyze other distance measures, such as the Manhattan distance.

6. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. Addison Wesley, 1997.

[2] S. Berchtold, C. Böhm, D. A. Keim, and H. P. Kriegel. A cost model for nearest neighbor search in high-dimensional data space. Proc of ACM PODS, pp. 78–86, 1997.

[3] D. Comaniciu, P. Meer, and D. Foran. Image guided decision support system for pathology. Machine Vision and Applications, 11(4): 213–224, 1999.

[4] R. Duda and P. Hart. Pattern Classication and Scene Analysis. John Wiley & Sons, 1973.

[5] J. Hafner, H. Sawney, W. Equitz, M. Flickner, and W. Niblack. Efficient Color Histogram Indexing for Quadratic Form Distance Functions. IEEE Transactions on PAMI, 17(7): 729–736, 1995.

[6] A. Hampapur and R. Bolle. Comparison of distance measures for video copy detection. Proc of International Conference on Multimedia and Expo, 2001.

[7] A. K. Jain and A. Vailaya. Image retrieval using color and shape. Pattern Recognition, 29(8): 1233–1244, 1996.

[8] S. Jeong, K. Kim, B. Chun, J. Lee and Y. J. Bae. An effective method for combining multiple features of image retrieval. Proc. of the IEEE Region 10 Conference, pp. 982–985, 1999.

[9] M. H. Kalos and P. Whitlock. Monte Carlo Methods. John Wiley & Sons, 1986.

[10] F. Liu and R. Picard. Periodicity, directionality, and randomness: Wold features for image modeling and retrieval. IEEE Transactions on PAMI, 18(7): 722–733, 1996.

[11] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of large image data. IEEE Transactions on PAMI, 18(8): 837–842, 1996.

[12] C. Nastar, M. Mitschke and C. Meilhac. Efficient query refinement for image retrieval. Proc. of IEEE CVPR, pp. 547–552, 1998.

[13] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Pektovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content using color, texture, and shape. Proc. of SPIE Storage and Retrieval for Image and Video Databases, pp. 173–181, 1993.

[14] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. Huang. Supporting similarity queries in MARS. Proc. of ACM Multimedia, pp. 403–413, 1997.

[15] E. Petrakis and C. Faloutsos. Similarity searching in medical image databases. IEEE TKDE, 9(3): 435–447, 1997.

[16] G. Qian, S. Sural, and S. Pramanik. A Comparative Analysis Of Two Distance Measures In Color Image Databases. Proc. of IEEE Int. Conf. on Image Processing, pp. 401–404, 2002.

[17] N. Sebe, M. Lew and D. Huijsmans. Toward improved ranking metrics. IEEE Trans. on PAMI, 22(10): 1132–1143, 2000.

[18] J. Smith. Integrated spatial and feature image systems: retrieval, analysis and compression. Ph.D. Dissertation, Columbia University, 1997.