Distance Quantization Method for Fast Nearest Neighbor Search Computations with applications to Motion Estimation

Hye-Yeon Cheong and Antonio Ortega
Signal and Image Processing Institute
University of Southern California
Los Angeles, CA 90089

***Abstract*—The problem of given a query vector finding its nearest neighbor within a large set of vectors in high dimensional space arises in many applications. It often poses serious computational challenges due to the size of data point set (database), dimensionality of the search space, and the metric complexity. There exists a wealth of results in the literature that reduce complexity primarily based on altering the data set while still computing the chosen distance metric to full precision. However further significant simplification is attainable with our proposed approach which reduces the search metric computation resolution by applying non-uniform quantization within the metric computation process, in such a way that the minimum distance ranking is most likely to be preserved. This paper provides analytical and experimental studies of our proposed approach. We present an analytical formulation of the search performance measure that gives insight into understanding this approach. Based on this formulation we present a quantizer optimized to minimize the impact of quantization on identifying the nearest neighbor. The main advantages of this approach are that: i) it can reduce the number and complexity of required arithmetic operations significantly, ii) complexity does not increase with the order of l_p norm or input bit size, and increases only slowly with dimensionality, and most importantly iii) the penalty to be paid in performance for the complexity reduction is very small if designed optimally. Motion estimation and compensation for video coding is chosen as an example application. Without requiring any filtering, transform, or sorting process, using a simple hardware oriented mapping, our experimental results show on average 0.05dB loss using only 1 bit per pixel distance (0.01 dB when 2 bits are used) instead of typical 8 or 16 bits metrics.**

I. Introduction

Searching for the closest point in high dimensional spaces is among the most fundamental problems in the study of searching algorithms and it is central to a wide range of applications. However, many real world search problems present serious computational challenges as they involve high dimensional search spaces with often largely varying non-deterministic data sets. Many algorithms have been proposed to reduce the computation complexity of these problems.

Most existing studies and algorithms focus on altering a data set/database S so that only parts of the database are searched, or only part of the query data is used for matching, using for example data-restructuring [1] [2], filtering [3], sorting, sampling [4] [5], transforming, bit-truncating [6] [7], quantizing [8] etc. Many among these algorithms provide good reductions in search complexity while maintaining acceptable performance. Our work is based on the observation that further significant reduction in complexity is attainable by i) focusing on preserving the fidelity of the minimum distance ranking instead of that of source data S, and ii) reducing the search metric computation resolution/precision instead of blindly computing the metric to full precision. This is because the metric is computed *only* in order to compare different candidate points; thus the metric value itself is not important, as long as it still allows identifying the candidate closest to the query vector.

In this paper, we introduce a novel approach based on these observations by applying non-uniform scalar quantization within the metric computation process. This approach leads to significant complexity savings by reducing the number (e.g., total number of additions) and complexity (e.g., bit depth of adders) of required arithmetic operations. More importantly, these computational savings can have minimal impact on performance since quantization is designed to preserve the minimum distance ranking fidelity, rather than based on a source data fidelity criterion. Finally, because the focus of our proposed approach is not on modification of S but only on the metric computation, it can be used in combination with other existing approaches which efficiently exploits redundancies in a data set S to further improve complexity reduction.

This paper provides analytical and experimental studies of our proposed approach. We present an analytical formulation of the search performance measure that provides insight into understanding the impact of this approach on performance. Based on this formulation we present a quantizer optimized to minimize the impact of quantization on identifying the nearest neighbor. As an example application, we apply our techniques to the motion estimation process in video compression systems.

The rest of the paper is organized as follows. In Section II, our proposed approach and its implementation are described in detail. In Section III, we formulate the performance measure of this approach and provide the optimal quantizer design. In Section IV, detailed experimental results are provided to validate our study and to demonstrate the performance of our proposed approach when applied to motion estimation process used in video coding system as an example application. Concluding remarks follow in Section V.

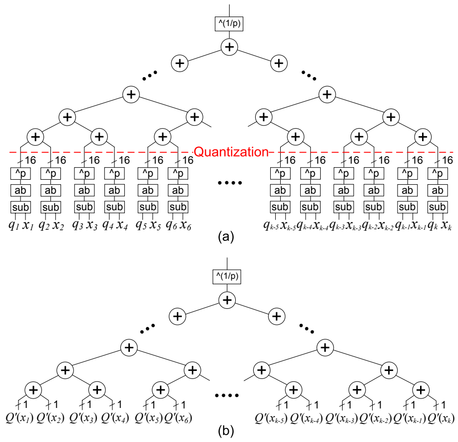
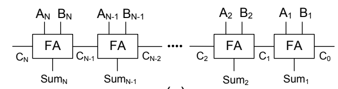
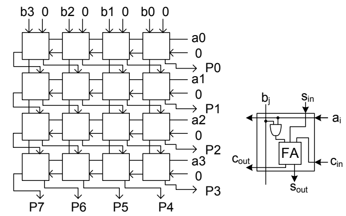


Fig. 1. Illustration of our proposed approach: (a) applying non-uniform quantization within the Minkowski type metric computation architecture (b) its equivalent form to (a).



Adder Type	# of gates			avg. # of logic transitions/add		
	adder size (bits)			adder size (bits)		
	16	32	64	16	32	64
Ripple Carry	144	288	576	90	182	366
Carry Lookahead	200	401	808	100	202	405
Carry Skip	170	350	695	108	220	437
Carry Select	284	597	1228	161	344	711
Conditional Sum	368	857	1938	218	543	1323

(b)



Multiplier Type	# of gates			avg. # of logic transitions pipelined		
	multiplier size (bits)			multiplier size (bits)		
	8	16	32	8	16	32
Array	528	2336	9792	548	7191	99062
Modified Array	567	2405	9918	583	7348	99102
Wallace	613	2569	10417	573	3874	19548
Dadda	612	2477	10051	557	3389	16638

(d)

Fig. 2. Diagrams of arithmetic circuits for (a) N-bit ripple carry adder and (c) 4 bit array multiplier. Tables (b),(d) provide the average number of transitions (a measure of dynamic power consumption) and the number of gates (a measure of circuit size, static power consumption) for various types of adders and multipliers for different input bit sizes. [9]

II. Proposed Approach

The closest point search problem, or nearest neighbor search (NN), is that of locating the data point inside a data set S in metric space M that is closest to the given query point $q \in M$ based on the distance metric $d : q \times S \to \Re$. In many cases of interest M is the k-dimensional Euclidean space $\Re^k$.

$$NN(q) = \{\bar{x} \in S | \forall x \in S \subset M, q \in M : d(\bar{x}, q) \leq d(x, q)\} .$$

The most commonly used proximity index is the Minkowski metric. The Minkowski metric of order p (p-norm distance) for measuring the dissimilarity between two k-dimensional data points q and x, is defined as:

$$d(q, x) = (\sum_{j=1}^{k} |q_j - x_j|^p)^{1/p}$$

This metric computation comprises two basic processes, i) the distance computation in each dimension (which we will refer to as "dimension-distance"): $dist_j(q, x) = |q_j - x_j|^p$ and ii) the summation of all such distances: $\sum_{j=1}^{d} dist_j(q, x)$ followed by $1/p$-th power computation at the final output.

Most approaches focus on the data set S in order to achieve search complexity reduction. Popular techniques include searching a smaller number of candidates, computing the distance metrics based on a subset of dimensions, or reducing the accuracy of the representation of data in each dimension. However further simplification is attainable when focusing on preserving the fidelity of the minimum distance ranking. We propose a novel metric computation method which achieves this by applying non-uniform scalar quantization (where integer values are assigned as reconstruction to each interval) on the partial distance terms in each dimension, $|x_{qj} - x_{rj}|^p$, prior to the summation process (Fig. 1(a)), where the quantizer is chosen with the goal to preserve as well as possible the minimum distance ranking. We next evaluate the complexity of this approach, while in Section III we develop the optimal quantizer design to maximize performance.

A. Complexity

The proposed approach leads to complexity reductions in i) the summation process after quantization (upper part of the dashed quantization line in Fig. 1(a)) and ii) the dimension-distance computation process prior to quantization (lower part of the line in Fig. 1(a)).

Fig. 2 provides useful insight to understand the computational complexity of the search at the circuit level [9]. Fig. 2 illustrates the structure of arithmetic circuits for a representative adder (a) and multiplier (c), the size of which increases with the input bit size. Figs. 2 (b), (d) essentially demonstrate that the computational complexity, circuit size, static and dynamic power consumption, computation delays of most basic arithmetic elements including adder or multiplier are all directly influenced by, and increase polynomially with, the input bit size. Therefore quantization applied to partial distance terms in each dimension, as shown in Fig. 1 (a), leads to significant simplification of the summation process. Moreover, as will be shown in Section IV for motion estimation in video coding, very coarse quantization is possible (e.g., to 1 bit) leading to much reduced complexity in the summation process while leaving video coding performance nearly unchanged (average 0.05dB loss).

Moreover, quantizers can be designed in such a way that the per dimension distance computation $|q_j - x_j|^p$ is not required (Fig. 1 (b)). This is because both the quantizer thresholds $\{\theta_i\}$ and the query vector q are fixed for a given search query, so that only the $x \in S$ being tested for their proximity to q vary. Therefore, candidate data x can be quantized directly with a quantizer $Q' : \left\{q \pm \theta_i^{1/p}\right\}$, which would lead to the same result as computing $|q_j - x_j|^p$ followed by quantization by $Q : \{\theta_i\}$, but at a fraction of the complexity. Our proposed metric computation $\bar{d}$ can be represented as:

$$\bar{d}(q, x) = (\sum_{j=1}^{k} Q(|q_j - x_j|^p))^{1/p} = (\sum_{j=1}^{k} Q'(x_j))^{1/p}$$

Fig. 1 (b) shows the metric computation architecture of proposed method, which is considerably simplified but equivalent to Fig. 1 (a). Fig. 3 illustrates the complexity increase as a function of the input bit size, dimensionality, and order p of metric (p-norm distance) for both conventional and proposed metric computations. We measure complexity in units of number of full-adder operations (basic building blocks of arithmetic logic circuits), under the assumption that n-bit addition, subtraction,

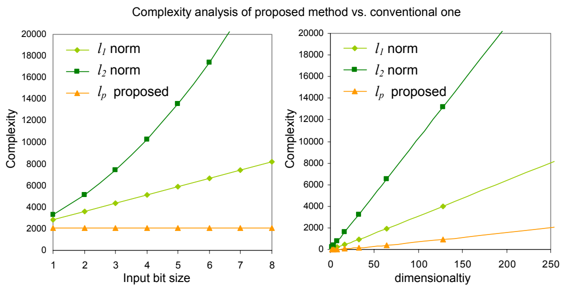


Fig. 3. Complexity behavior comparison of conventional l_1 and l_2 norm metric computation vs. proposed distance quantization based l_p norm metric computation with relationship to the input bit size and dimensionality.

and absolute value operations have the same complexity and that a square operation has equivalent complexity to that of an n^2-bit addition. For motion estimation example, the dimensionality represents the number of pixels per matching block while input bit size represents pixel bit-depth. Note that the complexity of the proposed method remains constant over different input bit sizes and l_p norms, while it slowly increases with dimensionality as compared to the conventional metric computation.

This approach obviously imposes extra complexity for quantization process. However, this quantization implementation can be integrated with the following adder block such that its overhead cost is kept negligible as compared to the complexity reduction achieved elsewhere.

The complexity reduction comes at the expense of some performance loss due to the quantization process. As expected, there is a trade-off between complexity and performance such that coarser quantization will lead to further complexity reductions, while increasing degradation in search performance. To maximize the performance for a fixed number of quantization levels or steps, it is critical to determine the optimal quantizer.

III. Optimal Quantizer Design

A. Performance Measure for General Search Algorithm

We first propose a cost function to quantify the difference in performance between an arbitrary search algorithm and a chosen benchmark search algorithm. This metric is based on computing the average difference in distance between query point and the (possibly different) nearest neighbors identified by each algorithm. We denote the search dataset, a query, a metric, and a resulting nearest neighbor data point of the benchmark algorithm as S, q, d, and $NN(q)$, respectively. We denote $\bar{S}$, $\bar{q}$, $\bar{d}$, and $\overline{NN}(q)$ similar quantities obtained for a different target algorithm.

$$NN(q) = \{\bar{x} \in S | \forall x \in S \subset M, q \in M : d(\bar{x}, q) \leq d(x, q)\}$$
$$\overline{NN}(q) = \{\bar{x} \in \bar{S} | \forall x \in \bar{S} \subset S, q \in M : \bar{d}(\bar{x}, \bar{q}) \leq \bar{d}(x, \bar{q})\}$$

We define the search performance cost measure as[1]:

$$E_{NN} = E\left\{d(\overline{NN}(q), q) - d(NN(q), q)\right\} \quad (1)$$

where the expectation is with respect to the query data when S and $\bar{S}$ are fixed, or with respect to the set $\left\{(q, S, \bar{S})_i\right\}_i$ otherwise. This equation can be further written as:

[1]Note that the benchmark algorithm is assumed to identify the nearest neighbor for the given search, so that E_{NN} is non-negative.

$$E_{NN} = \int_{R^+} \bar{\mu}(a)\, \bar{f}(a) da - \int_{R^+} a\, f(a) da \quad (2)$$

where $\bar{\mu}(a) = E\left\{d(x, q) | \bar{d}(x, q) = a, x \in \bar{S}\right\}$, and minimum distance distribution functions $\bar{f}(a) = Pr(\bar{d}(\overline{NN}(q), q) = a)$, $f(a) = Pr(d(NN(q), q) = a)$.

If the goal is to find a target search algorithm minimizing this cost E_{NN}, we only consider the first term of above equation (which we denote as $\bar{E}$), since the target algorithm affects only this term.

$$\bar{E} = E\left\{d(\overline{NN}(q), q)\right\} = \int_{R^+} \bar{\mu}(a)\, \bar{f}(a) da \quad (3)$$

B. Performance Cost Function for the Proposed Method

Our focus is not on modifying S or q but only on simplifying the metric computation d by embedding a quantizer within the Minkowski metric computation architecture. Thus, the Minkowski metric (p-norm distance) becomes our benchmark and we assume $\bar{S} = S$. Our goal to find the quantizer that, for a given number of quantization levels N, can minimize E_{NN}. Instead of considering statistical information of $x \in S$ and q separately, our cost function is based on the statistical characteristics of Y, a k-dimensional multivariate random variable representing the input data on which a quantizer is applied:

$$Y_i = (y_{i1}, y_{i2}, ..., y_{ik}) = (|q_1 - x_{i1}|^p, |q_2 - x_{i2}|^p, ..., |q_k - x_{ik}|^p)$$

We further denote quantized input as:

$$Z_i = (z_{i1}, z_{i2}, ..., z_{ik}) = Q(Y) = (Q(y_{i1}), Q(y_{i2}), ..., Q(y_{ik}))$$

Its corresponding benchmark and proposed target metrics are:

$$d(Y_i) = (\sum_{j=1}^{k} y_{ij})^{1/p} \qquad \bar{d}(Y_i) = d(Q(Y_i)) = (\sum_{j=1}^{k} z_{ij})^{1/p}$$

The number of candidates M and their dimensions k are assumed to be fixed over the search process. We define a quantizer operating on y as a set of N non-overlapping intervals that cover all possible values of y: $S = \{s_n; s_n = [\theta_n, \theta_{n+1}), n \in \Phi\}$, where Φ is a set of consecutive integers from 0 to $N-1$, and $\{\theta_n\}$ is an increasing sequence of thresholds. Therefore for all $y_{ij} \in s_n$, we assign $z_{ij} = Q(y_{ij}) = n$, and the probability mass function (pmf) p_{ij} and centroid μ_{ij} of z_{ij} can be computed using $f_{y_{ij}}$, the probability density function (pdf) of y_{ij} as:

$$z_{ij} = Q(y_{ij}) = \sum_n n\, 1_{s_n}(y_{ij}) \quad (4)$$

$$p_{ij}(n) = \int_{s_n} f_{y_{ij}}(y) dy, \qquad \mu_{ij}(n) = \frac{\int_{s_n} y\, f_{y_{ij}}(y) dy}{\int_{s_n} f_{y_{ij}}(y) dy}$$

We further denote the cumulative mass and centroid functions of z_{ij} as $P_{ij}(n) = \sum_{s_0}^{s_n} p_{ij}(n)$ and $U_{ij}(n) = \sum_{s_0}^{s_n} p_{ij}(n)\mu_{ij}(n)$.

We first consider a simple case with M random samples from a k-variate distribution f_Y with iid dimensions, i.e., all y_j following the same pdf f_y and are independent of each other. The k-dimensional space can be partitioned into hypercubes through quantization, so that each input sample $Y = (y_1, y_2, ..., y_k)$ falls into one of the hypercubes. Each hypercube can be represented by a vector $Z = (z_1, z_2, ..., z_k) = (Q(y_1), Q(y_2), ..., Q(y_k))$ and all z_j have the same pmf p and the same centroid function μ. Each hypercube Z can be described by i) a probability mass

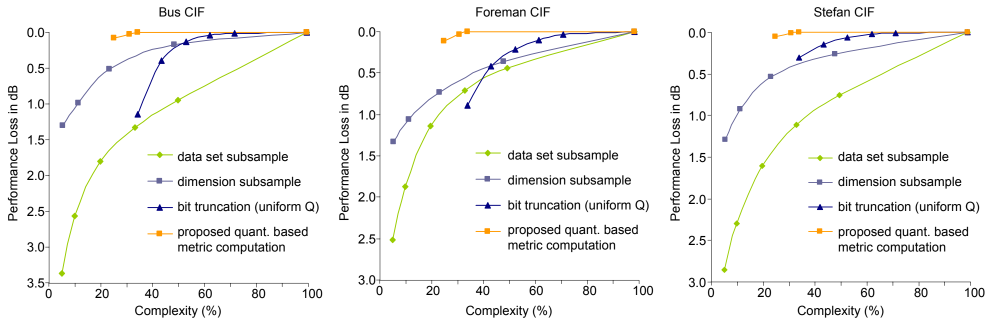


Fig. 4. Comparisons of Complexity-Performance trade-offs of four different scenarios: each scenario reduces i) size of a data set S, ii) dimensionality of each data $x \in S$, iii) bit depth of each data dimension by truncating least significant bits (equally seen as uniform quantization on each data dimension), and iv) resolution of each dimension-distance (proposed distance quantization based metric computation. X axis represents complexity percentage to that of original full computation. Y axis represents the RD performance loss measured in dB, thus zero represents no performance degradation.

M_Z, ii) a centroid C_Z, and iii) its corresponding total metric S_Z:

$$M_Z = \prod_j p(z_j), \quad C_Z = \sum_j \mu(z_j), \quad S_Z = \|Z\|_1 = \sum_j z_j$$

The pmf $P_{\|Z\|_1}$ represents the probability of a sample Y falling in one of the hypercubes having a given S_Z:

$$p_{\|Z\|_1}(x) = \sum_{\|Z\|_1 = x} M_Z = \sum_{\|Z\|_1 = x} \prod p(z_j) = p^{*k}(x)$$

where p^{*k} is the k-fold power convolution of p. Note that our ultimate goal is to minimize the cost function $\bar{E}$ in Eq.(3):

$$\bar{E} = \sum_a \bar{\mu}(a)\bar{p}(a), \tag{5}$$

where $\bar{p}$ is the pmf of the minimum S_Z value among M samples of Y:

$$\bar{p}(a) = (\sum_{x=a}^{\infty} p_{\|Z\|_1}(x))^M - (\sum_{x=a+1}^{\infty} p_{\|Z\|_1}(x))^M$$

$$\bar{p} = \nabla(\hat{P}_{\|Z\|_1}(a))^M$$

and where ∇ is a backward difference operator and we define a reverse cmf $\hat{P}(x) = 1 - P(x) = Pr[X \geq x]$.

$\bar{\mu}(a)$ is the centroid of all hypercubes with the same $S_Z = a$.

$$\bar{\mu}(a) = \frac{\sum_{\|Z\|_1 = a} M_Z C_Z}{\sum_{\|Z\|_1 = a} M_Z} \qquad \bar{\mu} = \frac{k\, p^{*(k-1)} * (p\mu)}{p^{*k}}$$

The above formulation assumes p=1. Alternatively it would be valid for cases when the benchmark metric does not include $1/p$-th power computation, as is the case in most real search applications. Otherwise, redefining C_Z as $C_Z = (\sum_j \mu(z_j))^{\frac{1}{p}}$ allows us to use the same procedure.

Extending this to the more general case, we can consider candidates Y_i to be drawn each from different f_{Y_i}; similarly we can consider that the of each vector dimensions have non-identical distributions. However we assume that vector data is independent across dimensions and that candidates with similar distance in terms of the benchmark metric d also share a similar distribution. We denote $f_\lambda(\lambda) = Pr(d(x,q) = \lambda, x \in S)$, a distribution of M candidates Y_i in terms of benchmark distance (λ). Representing candidates having same λ as $Y_\lambda = (y_{\lambda 1}, y_{\lambda 2}, ..., y_{\lambda k})$ with $y_{\lambda j}$ following a pdf $f_{y_{\lambda j}}$, we have $Z_\lambda = (z_{\lambda 1}, z_{\lambda 2}, ..., z_{\lambda k})$ with $z_{\lambda j}$ following a pmf $p_{\lambda j}$ and its centroid function $\mu_{\lambda j}$. Thus, for each hypercube Z_λ,

$$M_{Z_\lambda} = \prod_j p_{\lambda j}(z_{\lambda j}) \quad C_{Z_\lambda} = \sum_j \mu_{\lambda j}(z_{\lambda j}) \quad S_{Z_\lambda} = \|Z_\lambda\|_1$$

We define a new operator $\prod_{i=n}^{m} p_i^* \equiv p_n * p_{n+1} * \cdots * p_m$ with which we represent $p_{\|Z_\lambda\|_1}$ as,

$$p_{\|Z_\lambda\|_1}(x) = \sum_{\|Z_\lambda\|_1 = x} M_{Z_\lambda} \qquad p_{\|Z_\lambda\|_1} = \prod_{j=1}^{k} p^*_{\lambda j}$$

Consequently, $\bar{p}$ and $\bar{\mu}$ of $\bar{E} = \sum_a \bar{\mu}(a)\bar{p}(a)$ becomes

$$\bar{p} = \nabla(E_\lambda[\hat{P}_{\|Z_\lambda\|_1}(x)])^M$$

$$\bar{\mu} = E_\lambda[\sum_{i=1}^{k}(\prod_{j \neq i}^{1..k} p^*_{\lambda j}) * (p_{\lambda i}\mu_{\lambda i})/p_{\|Z_\lambda\|_1}]$$

Given the cost function quantifying the performance loss, our goal is to find a quantizer that leads to the minimum $\bar{E}$. Considering the case when data is assumed to be iid across dimensions, for a given input distribution f_y, a quantizer is uniquely defined by two vectors $\mu, p \in \Re^N$, where p satisfies the probability axioms (i.e., it is uniquely defined given the set of centroids and the probability masses of each quantization bin.) Note that given f_y, $\bar{E}$ is a function of p. Note also that $\bar{E}$ can be represented in terms of P and U, defined previously as cumulative mass and centroid functions of z_j, where $P \in C$ such that $\bar{E}(P) : \Re^N \mapsto \Re$ and C is a convex subset of $\Re^N$, $C = \left\{x | x_i \leq x_{i+1}, 0 \leq x_i \leq 1, \forall i,\ x \in \Re^N\right\}$. It can be shown that

$$\bar{E}(\hat{P}) \geq \bar{E}(P) + (\hat{P} - P)'\nabla\bar{E}(P), \quad \forall \hat{P}, P \in C$$

where a gradient of $\bar{E}$: $\nabla\bar{E}(F_z) = (\frac{\partial\bar{E}(P)}{\partial P(0)}, \cdots, \frac{\partial\bar{E}(P)}{\partial P(N-1)})'$, proving that $\bar{E}$ is convex over C.

Therefore, finding the optimal quantizer can be formulated as a constrained convex optimization problem with the goal to minimize $\bar{E}(P)$ subject to $P \in C$. The global minimum value represents the optimal performance attainable given input distribution and can be obtained using standard convex optimization techniques. From the P vector corresponding to the global minimum, the optimal quantizer can be uniquely determined.

For more general situation considering different statistical distribution for candidates as well as dimensions with $\bar{E}$ function shown above, essentially similar procedure can be developed and optimal quantizer can be found using standard optimization techniques.

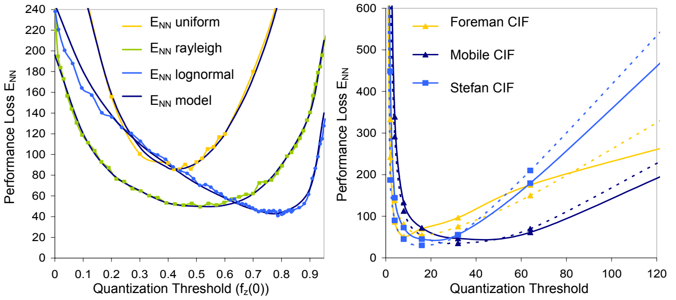


Fig. 5. (Left) compares our cost function $\bar{E}$ (Eq.(5)) with numerically simulated experiments for different input distribution settings. (Right) compares the cost function $\bar{E}$ (Eq.(6)) based on the collected ME data (dashed lines) with simulated experiments excluding intra and skip modes (solid lines) measured in E_{NN}. Both used 1bit quantization thus a single threshold (x-axis).

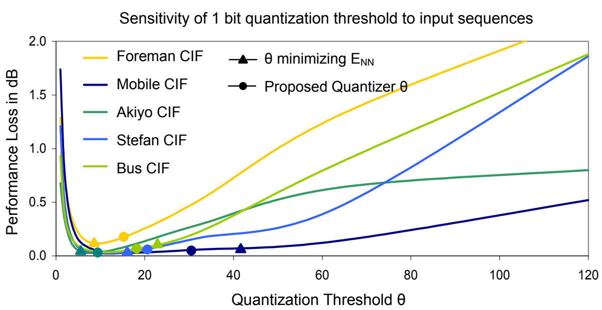


Fig. 6. Performance degradation in coding efficiency with relation to the 1-bit quantizer threshold value for various input sequences, illustrating low sensitivity of the optimal threshold to different input. Thresholds minimizing E_{NN} and our search performance measure model are also shown. The range of threshold is from 0 to 255 while only 0 to 120 range is shown.

IV. Motion Estimation Experiments

In this section, our proposed approach and its analytical study are applied to motion estimation (ME) process used in video coding system as an example application. Experimental results are provided to validate our study and to demonstrate the performance of our proposed approach. Various sequences were tested for simulation using a H.264/MPEG-4 AVC baseline encoder with 16×16 block partitions (256 dimensional vectors), a single reference, full pel resolution search, 8-bit depth pixel, and l_1 norm (sum of absolute difference) for search metric, and the search window of ±16 resulting a data set size of 1089.

Statistical characteristics of general ME input data show input dimension distances (pixel distances) to have approximately independent identical distributions while distribution varies with different candidates (distant candidates showed higher variance than nearer ones). Therefore $\bar{p}$ and $\bar{\mu}$ of the cost function $\bar{E} = \sum_a \bar{\mu}(a)\bar{p}(a)$ in Eq. (5) for the general ME data becomes:

$$\bar{p} = \nabla(E_\lambda[\hat{P}_{\|Z\|_1}(x)])^M \qquad \bar{\mu} = E_\lambda[\frac{k\, p^{*(k-1)} * (p\mu)}{p^{*k}}]$$

Fig. 4 illustrates the trade-offs between complexity and performance for proposed and three different representative scenarios. Other sequences tested showed similar results. The proposed approach provides a better trade-off and can also be used together with most of other existing algorithms to further improve the complexity reduction.

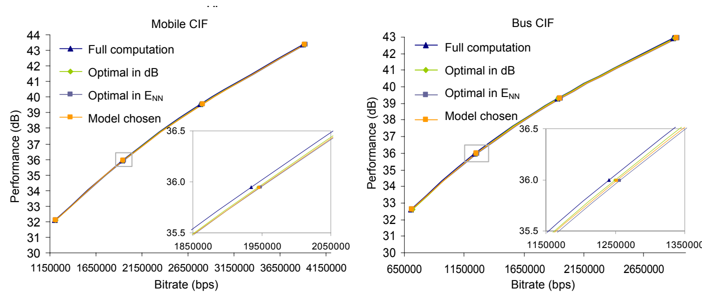


Fig. 7. Rate distortion performance of full metric computation and proposed 1 bit quantization based method with three different thresholds: each one is optimal in i) overall coding efficiency (dB), ii) E_{NN} measure, and iii) our proposed cost model based quantizer. Other sequences showed similar results.

Fig. 5 (Left) compares our cost function $\bar{E}$ with the expected performance error collected from numerically simulated experiments for different input distribution settings f_y. As the number of experiments increases, expected error converges to our cost function, confirming the accuracy of our $\bar{E}$ formulation. Fig. 5 (Right) compares our cost function based on the collected ME data with simulated experiments.

Fig. 6 and Fig. 7 compare the performances of our proposed method with three different thresholds each of which minimizes overall coding efficiency, E_{NN} measure, and our proposed cost model. These results clearly show that quantizers obtained by optimizing our simplified cost function can achieve near optimal performance. Fig. 6 also provides some insight about the sensitivity of optimal threshold to input variation. Despite large variation of the input source characteristics, dimension-distances where quantization is applied exhibit more consistent statistical behavior, leading to overall robustness in our quantization method.

V. Conclusion

We introduced a novel approach to the nearest neighbor search problem achieving significant computational complexity reduction by applying non-uniform quantization within the metric computation. We provide an analytical formulation of the search performance measure, based on which the optimal quantizer is designed to maximize the fidelity of the minimum distance ranking.

References

[1] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, MA, 1980.

[2] T. Sellis, N. Roussopoulos, and C. Faloutsos, "Multidimensional access methods: Trees have grown everywhere," in *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB)*, 1997.

[3] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms," in *IEEE Trans. Circuits Syst. Video Technol. vol. 7*, 1997, p. 702706.

[4] B. Liu and A. Zaccarin, "New fast algorithm for motion estimation of block motion vectors," in *IEEE Trans. CSVT. 3*, 1993, p. 148157.

[5] K. T. Choi, S. C. Chan, and T. S. Ng, "A new fast motion estimation algorithm using hexagonal subsampling pattern and multiple candidates search," in *in Proc. ICIP. 2*, 1996, p. 497500.

[6] Z. L. He et al, "Low-power vlsi design for motion estimation using adaptive pixel truncation," in *IEEE Transaction on CSVT 10*, 2000.

[7] Y. Chan and S. Y. Kung, "Multi-level pixel difference classification methods," in *in Proc. ICIP*, 1995, p. 252255.

[8] S. Lee, J. M. Kim, and S. I. Chae, "New motion estimation algorithm using adaptively quantized low bit-resolution image and its vlsi architecture formpeg video encoding," in *IEEE Trans. CSVT. 8*, 1998.

[9] T. K. Callaway and E. E. Swartzlander, "Optimizing arithmetic elements for signal processing," in *VLSI Signal Processing, V*, 1992, pp. 91–100.