Fast Encoding Method for Vector Quantization Using Modified L2-Norm Pyramid

著者	小谷　光司
journal or publication title	IEEE signal processing letters
volume	12
number	9
page range	609-612
year	2005
URL	http://hdl.handle.net/10097/34917

Fast Encoding Method for Vector Quantization Using Modified L_2-Norm Pyramid

Zhibin Pan, *Member, IEEE*, Koji Kotani, *Member, IEEE*, and Tadahiro Ohmi, *Fellow, IEEE*

***Abstract*—The L_2-norm pyramid has already been investigated as a promising data structure for the fast search of vector quantization (VQ) encoding in the previous work. Because the distortion at the top level is always tested first when using such a conventional L_2-norm pyramid, the top level is most important. In order to enhance the capability of achieving a rejection decision at the top level, a modification is introduced into the conventional L_2-norm pyramid in this letter by using both the mean and the variance of a vector simultaneously to replace the L_2-norm of the vector for distortion computation at the top level. Two issues are made clear as 1) why this modification is beneficial to the distortion test is proved and 2) why only the top level of a conventional L_2-norm pyramid should be modified is interpreted as well. Experimental results confirmed that the performance of VQ encoding by using the modified L_2-norm pyramid can be improved obviously.**

***Index Terms*—Conventional L_2-norm pyramid, fast encoding, modified L_2-norm pyramid, vector quantization (VQ).**

I. Introduction

Vector quantization (VQ) [1] is a popular method for image compression. In a conventional block-based VQ method, an $N \times N$ image is first divided into a series of nonoverlapping smaller $2^n \times 2^n$ image blocks. Then, for an image block $x(u, v)$, $u \in [1, 2^n]$, $v \in [1, 2^n]$, the distortion between x and a candidate codeword is usually measured by the squared Euclidean distance as

$$d^2(x, y_i) = \sum_{u=1}^{2^n} \sum_{v=1}^{2^n} [x(u, v) - y_i(u, v)]^2, \qquad i = 1, 2, \ldots, N_c \tag{1}$$

where $y_i(u, v)$, $u \in [1, 2^n]$, $v \in [1, 2^n]$ is the ith codeword in the codebook $Y = \{y_i | i = 1, 2, \ldots, N_c\}$, and N_c is the codebook size. It requires $(2 \times 2^n \times 2^n - 1)$ addition $(\pm)$ and $2^n \times 2^n$ multiplication $(\times)$ operations to compute $d^2(x, y_i)$.

Therefore, the winner can be determined directly by

$$d^2(x, y_w) = \min_{y_i \in Y} \left\lfloor d^2(x, y_i) \right\rfloor, \qquad i = 1, 2, \ldots, N_c \tag{2}$$

where the best-matched codeword y_w is called as the winner, and the subscript "w" is the winner index. Then, the index "w" instead of y_w is transmitted to the receiver for realizing data compression because "w" uses far fewer bits than y_w. This is a full search (FS) process.

II. Related Previous Work

Suppose the "so far" minimum real Euclidean distance obtained during a winner search process be $d_{\min}$. Conventionally, $d_{\min}$ can be determined by computing $d(x, y_{bm})$, where y_{bm} is the "so far" best-matched candidate codeword that has already been found during the winner search process. The previous work [2] proposed to use L_2-norm information of the k-dimensional $(k = 2^n \times 2^n)$ original vectors to roughly measure the distortion between the input vector x and a candidate codeword y_i as

$$d^2(x, y_i) \geq (\|x\| - \|y_i\|)^2, \qquad i = 1, 2, \ldots, N_c. \tag{3}$$

Therefore, if $(\|x\| - \|y_i\|)^2 \geq d^2_{\min}$, $i = 1, 2, \ldots, N_c$ is true, it guarantees that $d^2(x, y_i) \geq d^2_{\min}$, $i = 1, 2, \ldots, N_c$ is also true so that y_i can be rejected safely. When all $\|y_i\|$, $i = 1, 2, \ldots, N_c$ in a codebook are offline computed and stored, (3) needs one extra memory to store $\|y_i\|$ for each y_i and one "$\pm$," one "$\times$," and one "Cmp" (i.e., comparison) operation for a rejection test. This search method is FS equivalent.

Then, in order to use L_2-norm information in a more detailed or a finer way, a L_2-norm pyramid data structure is proposed in the previous work [3], as shown in Fig. 1(a). Taking the input x as an example, it is clear that a $(n + 1)$ level L_2-norm pyramid can be constructed for x. The top level is L_0, which stores the real L_2-norm of x. The bottom level L_n actually stores the original block x. Then, each pixel value at a higher $(m - 1)$th level can be computed from the corresponding four pixel values at a lower mth level, as shown in Fig. 1(a), to construct its L_2-norm pyramid by using (4), shown at the bottom of the next page.

Obviously, the relation in (5) holds according to the definition of the L_2-norm pyramid given in (4)

$$\begin{cases} (\|x\|_m)^2 = (\|x\|)^2, & m = 0, 1, \ldots, n \\ (\|x\|_m)^2 \overset{\text{Def}}{=} \sum\limits_{u=1}^{2^m} \sum\limits_{v=1}^{2^m} [x_m(u, v)]^2 \end{cases} \tag{5}$$

where $\|x\|_m$ physically refers to the L_2-norm of the purposely constructed $(2^m \times 2^m)$-dimensional vector at the mth level in a L_2-norm pyramid.

Then, the distortion between x and y_i can be measured level by level from the top level L_0 toward the bottom level L_n instead of an immediate real Euclidean distance computation by using (1). Because all levels of L_m, $m \in [0, n - 1]$ above the bottom level L_n have a much lower and exponentially decreasing dimension of $2^m \times 2^m$ compared to the original dimension of $2^n \times 2^n$, the distortion computation at these levels of L_m,

Manuscript received December 9, 2004; revised February 28, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dipti Prasad Mukherjee.

Z. Pan and T. Ohmi are with the New Industry Creation Hatchery Center, Tohoku University, Sendai 980-8579, Japan (e-mail: pzb@fff.niche.tohku.ac.jp).

K. Kotani is with the Department of Electronic Engineering, Graduate School of Engineering, Tohoku University, Sendai 980-8579, Japan.

Digital Object Identifier 10.1109/LSP.2005.851263

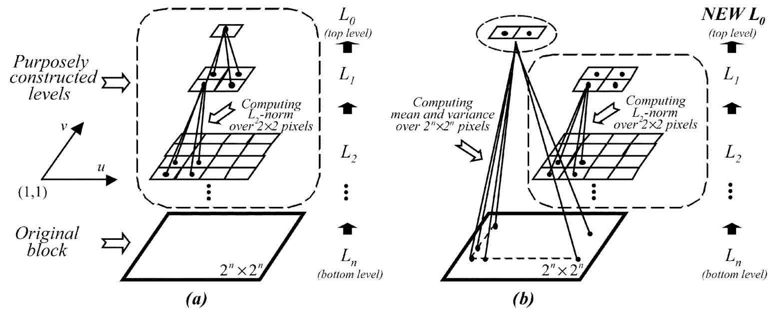

Fig. 1. Concept to show how to construct (a) a conventional L_2-norm pyramid data structure and (b) a modified L_2-norm pyramid data structure for the input x.

$m \in [0, n-1]$ must be computationally inexpensive. If a rejection decision to y_i can be made before the distortion test reaches the bottom level L_n, the computational cost can be reduced in practice. According to the concept of Euclidean distance, a direct definition of the distortion for the mth level between two pyramids of x and y_i should be

$$d_m^2(x, y_i) \stackrel{\text{Def}}{=} \sum_{u=1}^{2^m} \sum_{v=1}^{2^m} [x_m(u,v) - y_{i,m}(u,v)]^2 \qquad m = 0, 1, \ldots, n. \tag{6}$$

Equation (6) requires $(2 \times 2^m \times 2^m - 1)$ addition $(\pm)$ and $2^m \times 2^m$ multiplication $(\times)$ operations.

Furthermore, in order to simplify the derivation and computation, the previous work [3] defined another new distortion measurement at the mth level by subtracting the common term $(\|x\|)^2$ from (6) and using $(\|x\|_m^2) = (\|x\|)^2, (\|y\|_m^2) = (\|y\|)^2$ to get

$$\begin{aligned} \overline{d}_m^2(x, y_i) &= d_m^2(x, y_i) - (\|x\|)^2, \qquad m = 0, 1, \ldots, n \\ &= \|y_i\|^2 - 2 \sum_{u=1}^{2^m} \sum_{v=1}^{2^m} [x_m(u,v) \times y_{i,m}(u,v)]. \end{aligned} \tag{7}$$

Because $\|y_i\|^2, i = 1, 2, \ldots, N_c$ can be offline computed and stored, (7) needs $[1 + (2^m \times 2^m - 1) + 1] = 2^m \times 2^m + 1$ addition $(\pm)$ and $2^m \times 2^m$ multiplication $(\times)$ operations because the coefficient "2" in (7) can be realized by one addition $(\pm)$ instead of one multiplication $(\times)$ operation.

Then, the previous work [3] proved a progressive rejection test condition, as given in (8), based on the new distortion definition in (7) as

$$\begin{aligned} \overline{d}^2(x, y_i) \equiv \overline{d}_n^2(x, y_i) \geq \overline{d}_{n-1}^2(x, y_i) \geq \cdots \geq \overline{d}_m^2(x, y_i) \geq \cdots \\ \geq \overline{d}_1^2(x, y_i) \geq \overline{d}_0^2(x, y_i) \quad m = 0, 1, \ldots, n \end{aligned} \tag{8}$$

where $\overline{d}^2(x, y_i) \stackrel{\text{Def}}{=} d^2(x, y_i) - (\|x\|)^2$ and $\overline{d}_{\min}^2 \stackrel{\text{Def}}{=} d_{\min}^2 - (\|x\|)^2$. Then, the rejection test is conducted from the L_0 level first. If at a certain mth level $\overline{d}_m^2(x, y_i) \geq \overline{d}_{\min}^2$ becomes true, it guarantees that $\overline{d}^2(x, y_i) \geq \overline{d}_{\min}^2$ or $d^2(x, y_i) \geq d_{\min}^2$ is also true so that the current codeword y_i cannot be the winner. Therefore, the distortion tests can be terminated for all of the following $(m+1)$th level, $(m+2)$th level, and so on to save computational cost. Equation (8) is the core of the previous work [3]. Obviously, it is equivalent to use either(3) or (8) at the top level L_0. This search method is also FS equivalent.

It is obvious that (8) needs $\sum_{m=0}^{n-1}(2^m \times 2^m) = (2^n \times 2^n - 1)/3$ extra memories to store the L_2-norm pyramid for each y_i. Meanwhile, (8) needs $\sum_{m=0}^{n-1}(2^m \times 2^m + 1) = (2^n \times 2^n - 1)/3 + n$ "$\pm$", $\sum_{m=0}^{n-1}(2^m \times 2^m) = (2^n \times 2^n - 1)/3$ "$\times$," and (n) "Cmp" operations for a complete rejection test till the $(n-1)$th level. For the most common $4 \times 4 = 2^2 \times 2^2$ block size, it needs five extra memories for each y_i. Also, it needs seven "$\pm$", five "$\times$," and two "Cmp" for a complete rejection test till the L_1 level.

It is clear that (3) and (8) only use L_2-norm information for rejection tests. However, for an original k-dimensional vector, there are three characteristic values as the mean, the variance, and L_2-norm to statistically describe the property of the vector. Independently, the previous work [4] proposed another codeword rejection rule by using both the mean and the variance of a vector simultaneously as

$$d^2(x, y_i) \geq k(Mx - My_i)^2 + (Vx - Vy_i)^2 \tag{9}$$

where $Mx = \sum_{u=1}^{2^n} \sum_{v=1}^{2^n} x(u,v)/k$,

$$Vx \stackrel{\text{Def}}{=} \sqrt{\sum_{u=1}^{2^n} \sum_{v=1}^{2^n} [x(u,v) - Mx]^2}$$

is the mean and the variance of an input x, respectively. $My_i = \sum_{u=1}^{2^n} \sum_{v=1}^{2^n} y_i(u,v)/k$,

$$Vy_i \stackrel{\text{Def}}{=} \sqrt{\sum_{u=1}^{2^n} \sum_{v=1}^{2^n} [y_i(u,v) - My_i]^2}$$

means the same for y_i. If $k(Mx - My_i)^2 + (Vx - Vy_i)^2 \geq d_{\min}^2$ holds, then reject y_i safely. In practice, (9) uses a two-step test flow to check 1) $k(Mx - My_i)^2 \geq d_{\min}^2$ first and 2) $k(Mx - My_i)^2 + (Vx - Vy_i)^2 \geq d_{\min}^2$ in order to avoid a

$$\begin{cases} x_{m-1}(u,v) = \sqrt{x_m^2(2u-1, 2v-1) + x_m^2(2u, 2v-1) + x_m^2(2u-1, 2v) + x_m^2(2u, 2v)} \\ \qquad v = 1, 2, 3, \ldots 2^{m-1} \quad u = 1, 2, 3, \ldots, 2^{m-1} \quad m = 1, 2, 3, \ldots, n \\ x_n(u,v) \equiv x(u,v). \end{cases} \tag{4}$$

possible computational overhead of accumulating $(Vx - Vy_i)^2$ at the first step because a lot of codewords can simply be rejected by the first check. Equation (9) needs two extra memories to store My_i and Vy_i for each y_i. Meanwhile, (9) needs three "±," two "×," and two "Cmp" operations for a complete two-step rejection test.

III. Modified L_2-Norm Pyramid Data Structure

In the previous work [3], two facts are essential. The first fact is that the rejection tests always start from the most inexpensive top level L_0 according to (8), which implies that the L_0 level is most important. If a rejection test becomes successful at the L_0 level, it becomes unnecessary to conduct all the following tests at L_m, $m \in [1, n]$ anymore. In addition, the second fact is that the rejection tests at any L_m, $m \in [0, n]$ are completely independent to each other according to (8). This fact provides a possibility to use different rejection tests at different levels. In other words, it is unnecessary to always use the same L_2-norm information at all levels for rejection tests as given in (8). Any information, such as the mean and the variance of a vector, can also be integrated into a conventional L_2-norm pyramid. Then, a natural consideration could be to combine these two facts to find out a new and more powerful rejection test at the L_0 level instead of using L_2-norm information only.

Therefore, the first key problem in this letter is whether it is valuable to integrate the mean and the variance of a vector by using (9) into the top level of a conventional L_2-norm pyramid to realize a new rejection test. It can be proved (see the Appendix)

$$k(Mx - My_i)^2 + (Vx - Vy_i)^2 \geq (\|x\| - \|y_i\|)^2. \quad (10)$$

Because the left-hand term is larger, (10) guarantees that it is really more powerful to use (9) for a rejection test to replace (8) at the top level. Equation (10) is the core of this letter because (10) can answer why the modification at the top level is profitable by using both the mean and the variance of a vector simultaneously. As a result, a conventional L_2-norm pyramid shown in Fig. 1(a) can be modified to the data structure given in Fig. 1(b). Clearly, only the top level L_0 has been modified, which stores the mean and the variance instead of the L_2-norm of an original $(2^n \times 2^n)$-dimensional vector.

Based on (7)–(10), the rejection tests in this letter for all levels can be summarized in (11). If (11) becomes true at any mth level, reject y_i safely

$$\begin{cases} k(Mx - My_i)^2 + (Vx - Vy_i)^2 \geq d_{\min}^2, & m = 0 \\ \bar{d}_m^2(x, y_i) \geq \bar{d}_{\min}^2, & m = 1, 2, \ldots, n. \end{cases} \quad (11)$$

Then, the second key problem in this letter is whether it is valuable to continuously integrate the mean and the variance into the remaining L_m, $m \in [1, n-1]$ levels to replace the L_2-norm. From Fig. 1(a) and (7), it is clear that the extra memory requirement and the computational cost until the L_m level increase exponentially like $(1 + 2^1 \times 2^1 +, \ldots, +2^m \times 2^m)$, $m \in [1, n-1]$. Because it doubles the extra memory requirement and the computational cost further at each level if the mean and the variance of a vector are used to replace the L_2-norm, it is impractical to integrate them into (8) for the remaining L_m, $m \in [1, n-1]$ levels. The modification to the top level, as given in (11), is sufficient. Because only the top level in (8) is modified, (11) needs $2 + \sum_{m=1}^{n-1}(2^m \times 2^m) = (2^n \times 2^n - 1)/3 + 1$ extra memories to store the modified L_2-norm pyramid for each y_i. Meanwhile, (11) needs $3 + \sum_{m=1}^{n-1}(2^m \times 2^m + 1) = (2^n \times 2^n - 1)/3 + (n+2)$ "±," $2 + \sum_{m=1}^{n-1}(2^m \times 2^m) = (2^n \times 2^n - 1)/3 + 1$ "×," and $(n+1)$ "Cmp" operations for a complete rejection test until the $(n-1)$th level. For the most common $4 \times 4 = 2^2 \times 2^2$ block size, it needs six extra memories for each y_i. Meanwhile, (11) needs nine "±," six "×," and three "Cmp" operations at maximum for a complete rejection test until the L_1 level.

TABLE I
Comparison of the Reduced Search Space or the Remaining Real Euclidean Distance Computations Per Input Vector

CB	Method	Level	Test	Lena	F-16	Pepper	Baboon
256	C-L_2NP	L_0	$\bar{d}_0^2 < \bar{d}_{\min}^2$	16.27	14.19	18.90	49.64
		L_1	$\bar{d}_1^2 < \bar{d}_{\min}^2$	7.68	5.95	8.54	24.18
	M-L_2NP	L_0	*TC-1*	16.27	14.17	18.58	49.60
			TC-2	6.44	6.24	6.94	30.09
		L_1	$\bar{d}_1^2 < \bar{d}_{\min}^2$	4.54	3.55	4.58	17.64
512	C-L_2NP	L_0	$\bar{d}_0^2 < \bar{d}_{\min}^2$	29.85	27.23	36.34	98.34
		L_1	$\bar{d}_1^2 < \bar{d}_{\min}^2$	12.47	10.60	15.22	46.64
	M-L_2NP	L_0	*TC-1*	29.81	27.40	35.83	98.29
			TC-2	10.91	10.74	12.83	55.56
		L_1	$\bar{d}_1^2 < \bar{d}_{\min}^2$	7.13	5.77	7.91	32.41
1024	C-L_2NP	L_0	$\bar{d}_0^2 < \bar{d}_{\min}^2$	52.32	52.67	70.12	190.85
		L_1	$\bar{d}_1^2 < \bar{d}_{\min}^2$	17.94	18.10	25.68	81.77
	M-L_2NP	L_0	*TC-1*	52.06	52.55	68.86	189.97
			TC-2	16.21	19.49	22.07	107.57
		L_1	$\bar{d}_1^2 < \bar{d}_{\min}^2$	9.76	9.44	12.47	57.23

IV. Experiment Results

Simulation experiments are conducted to compare the performance of VQ encoding by using a conventional L_2-norm pyramid (i.e., C-L_2NP) or a modified L_2-norm pyramid (i.e., M-L_2NP). Four typical 8-bit, 512×512 standard images are used. The block size is $4 \times 4 = 2^2 \times 2^2$. The codebook size (i.e., CB) is chosen as 256, 512, and 1024, which are generated with the Lena image as a training set. The search flow is similar to that used in [3] except the top level, which uses a two-step rejection test as given in (9). In the case of using a C-L_2NP data structure, the codebook is offline rearranged along L_2-norm values of all codewords that are already sorted in an ascending order. For an input x, the search process starts from the initial best-matched codeword y_{int}, which is found by a binary search to let $|\|x\|_0 - \|y_{\text{int}}\|_0| \Longrightarrow \min$. Also, in the case of using a M-L_2NP data structure, the codebook is similarly offline rearranged along the mean values of all codewords that are already sorted in an ascending order. Then, for an input x, the search process starts from the initial best-matched codeword y_{int}, which is determined by a binary search to let $|Mx - My_{\text{int}}| \Longrightarrow \min$.

The computational complexity is evaluated by two kinds of assessments in this letter, which are 1) the reduced search space after completing each rejection test and 2) the total computational cost in arithmetical operations per input vector.

TABLE II
COMPARISON OF THE TOTAL COMPUTATIONAL COST BY ARITHMETICAL OPERATIONS PER INPUT VECTOR

CB	Method	OP	Lena	F-16	Pepper	Baboon
256	C-L_2NP	Add	339.0	271.2	374.4	999.7
		Mul	185.8	149.1	205.7	549.3
		Cmp	39.6	34.1	44.0	106.0
		Sqrt	5	5	5	5
	M-L_2NP	Add	232.6	192.6	237.4	811.2
		Mul	129.5	107.4	133.1	448.5
		Cmp	39.8	35.5	42.7	123.0
		Sqrt	5	5	5	5
512	C-L_2NP	Add	534.9	461.1	645.8	1901.6
		Mul	295.3	255.3	356.8	1047.1
		Cmp	63.8	57.4	75.8	200.6
		Sqrt	5	5	5	5
	M-L_2NP	Add	353.7	299.3	392.9	1472.1
		Mul	199.4	169.6	222.8	818.1
		Cmp	64.0	58.7	73.5	227.7
		Sqrt	5	5	5	5
1024	C-L_2NP	Add	765.1	771.5	1076.9	3329.1
		Mul	427.2	430.7	599.7	1842.3
		Cmp	98.2	98.9	131.5	364.4
		Sqrt	5	5	5	5
	M-L_2NP	Add	486.2	481.4	617.7	2611.0
		Mul	279.4	276.9	356.3	1458.2
		Cmp	97.8	100.9	125.9	422.0
		Sqrt	5	5	5	5

Obviously, the final reduced search space is a key factor to the encoding efficiency because real Euclidean distance computations must be conducted in this space. A smaller reduced search space is better. For convenience, let the first test condition of $k(Mx - My_i)^2 < d_{\min}^2$ at L_0 level be "*TC-1*" and the second test condition of $k(Mx - My_i)^2 + (Vx - Vy_i)^2 < d_{\min}^2$ at L_0 level be "*TC-2*" when using a M-L_2NP data structure. The reduced search spaces per input vector are summarized in Table I.

From Table I, it is obvious that at the top level L_0, the rejection test by using L_2-norm information of a vector like $\overline{d}_0^2 < \overline{d}_{\min}^2$ and the rejection test of "*TC-1*" by using the mean information of a vector like $k(Mx - My_i)^2 < d_{\min}^2$ can achieve almost the same reduced search space, which can also be observed in [5]. However, the "*TC-2*" rejection test like $k(Mx - My_i)^2 + (Vx - Vy_i)^2 < d_{\min}^2$ is extremely effective so that the final search space can be reduced obviously by using the M-L_2NP data structure. This is because the rejection test by using (11) is much more powerful than using (8) at the top level L_0.

Then, from Table II, it is clear that the total computational cost can be reduced obviously by using the M-L_2NP data structure. This is because (11) can lead to a much smaller reduced search space at the top level L_0 to guarantee much less real Euclidean distance computations.

V. CONCLUSION

In this letter, two contributions are made. First, a new inequality of (10) is proved, which states that it is more powerful to use the mean and the variance of a vector simultaneously for rejection tests than to use the L_2-norm of the vector. Equation (10) is important because it gives out a guideline on how to use the three popular statistical values of the mean, the variance, and L_2-norm of a vector for rejection tests more effectively. Second, when it is necessary to introduce a modification to a conventional L_2-norm pyramid by using the mean and the variance of a vector simultaneously to replace the L_2-norm of the vector, it is sufficient to only conduct this modification at the top level L_0 because the pyramid data structure features an exponentially increasing extra memory requirement and computational cost for each of the following L_m, $m \in [1, n-1]$ levels.

APPENDIX

In statistics, the relations $[\|x\|]^2 = k(Mx)^2 + (Vx)^2$ and $[\|y_i\|]^2 = k(My_i)^2 + (Vy_i)^2$ are true. Based on the basic inequality $a^2 + b^2 \geq 2ab$, the following relation can be derived, which completes the proof of (10):

$$
\begin{aligned}
& k \times (Mx)^2(Vy_i)^2 + k \times (My_i)^2(Vx)^2 \\
&\quad \geq 2k \times MxVxMy_iVy_i \\
&\quad \Leftrightarrow k^2 \times (Mx)^2(My_i)^2 + k \times (Mx)^2(Vy_i)^2 \\
&\qquad + k \times (My_i)^2(Vx)^2 + (Vx)^2(Vy_i)^2 \\
&\quad \geq k^2 \times (Mx)^2(My_i)^2 \\
&\qquad + 2k \times MxVxMy_iVy_i + (Vx)^2(Vy_i)^2 \\
&\quad \Leftrightarrow \left[k \times (Mx)^2 + (Vx)^2\right]\left[k \times (My_i)^2 + (Vy_i)^2\right] \\
&\quad \geq \left[k \times (Mx)(My_i) + (Vx)(Vy_i)\right]^2 \\
&\quad \Leftrightarrow [\|x\|]^2[\|y_i\|]^2 \geq \left[k \times (Mx)(My_i) + (Vx)(Vy_i)\right]^2 \\
&\quad \Leftrightarrow [\|x\|][\|y_i\|] \geq \left[k \times (Mx)(My_i) + (Vx)(Vy_i)\right] \\
&\quad \Leftrightarrow -2 \times [\|x\|][\|y_i\|] \\
&\quad \leq -2k \times (Mx)(My_i) - 2 \times (Vx)(Vy_i) \\
&\quad \Leftrightarrow [\|x\|]^2 - 2 \times [\|x\|][\|y_i\|] + [\|y_i\|]^2 \\
&\quad \leq [\|x_i\|]^2 - 2k \times (Mx)(My_i) \\
&\qquad - 2 \times (Vx)(Vy_i) + [\|y_i\|]^2 \\
&\quad \Leftrightarrow [\|x\| - \|y_i\|]^2 \\
&\quad \leq \left[k \times (Mx)^2 + (Vx)^2\right] - 2k \times (Mx)(My_i) \\
&\qquad - 2 \times (Vx)(Vy_i) + \left[k \times (My_i)^2 + (Vy_i)^2\right] \\
&\quad \Leftrightarrow [\|x\| - \|y_i\|]^2 \\
&\quad \leq \left[k \times (Mx)^2 - 2k \times (Mx)(My_i) + k \times (My_i)^2\right] \\
&\qquad + \left[(Vx)^2 - 2 \times (Vx)(Vy_i) + (Vy_i)^2\right] \\
&\quad \Leftrightarrow [\|x\| - \|y_i\|]^2 \leq k \times (Mx - My_i)^2 \\
&\qquad + (Vx - Vy_i)^2.
\end{aligned}
$$

REFERENCES

[1] P. C. Cosman, R. M. Gray, and M. Vetterli, "Vector quantization of image subbands: A survey," *IEEE Trans. Image Process.*, vol. 5, no. 2, pp. 202–225, Feb. 1996.

[2] C. M. Huang, Q. Bi, G. S. Stiles, and R. W. Harris, "Fast full search equivalent encoding algorithms for image compression using vector quantization," *IEEE Trans. Image Process.*, vol. 1, no. 3, pp. 413–416, Jul. 1992.

[3] B. C. Song and J. B. Ra, "A fast search algorithm for vector quantization using L_2-norm pyramid of codewords," *IEEE Trans. Image Process.*, vol. 11, no. 1, pp. 10–15, Jan. 2002.

[4] S. Baek, B. Jeon, and K. Sung, "A fast encoding algorithm for vector quantization," *IEEE Signal Process. Lett.*, vol. 4, no. 12, pp. 325–327, Dec. 1997.

[5] Z. Pan, K. Kotani, and T. Ohmi, "A fast encoding method for vector quantization using L_1 and L_2 norms to narrow necessary search scope," *IEICE Trans. Inf. Syst.*, vol. E86-D, pp. 2483–2486, 2003.