

行政院國家科學委員會專題研究計畫 成果報告

以 PSO 作向量量化的碼簿設計

計畫類別：個別型計畫

計畫編號：NSC93-2213-E-032-007-

執行期間：93年08月01日至94年07月31日

執行單位：淡江大學電機工程學系

計畫主持人：余繁

計畫參與人員：陳慶逸,李易聰

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 94 年 10 月 12 日

以 PSO 作向量量化的碼簿設計

VQ codebook design using Particle Swarm Optimization

計畫編號：NSC 93-2213-E-032-007

執行期限：93 年 08 月 01 日至 94 年 07 月 31 日

主持人：余 繁 教授 淡江大學電機工程系

計畫參與人員：陳慶逸 淡江大學電機工程系

李易聰 淡江大學電機工程系

e-mail : fyee@mail.tku.edu.tw

一、摘要

向量量化(VQ, vector quantization) 在影像壓縮技術中是一種非常基本的失真影像壓縮法，許多重要的影像壓縮技術，例如靜態影像壓縮中最有名的標準 JPEG 等都應用到 VQ 的基本觀念；但是在 VQ 演算法中，訓練向量值(vectors)以及最小化碼簿向量(codebook)間的 Means Square Error 卻是一個非線性問題，傳統的 LBG 型態演算法在迭代過程中常會收斂於局部最小值處，其結果與初始碼簿向量的決定具有很大的關聯性，許多的研究工作也都在致力於設計有效的碼簿設計及搜尋方法的改善。本計畫提出一個結合梯度下降法 (Gradient descent) 的 APSO-VQ 碼簿設計方法，該架構經由梯度下降來法來加快系統收斂的速度，並透過全域最佳解的搜尋能力建立高品質的碼簿向量。從實驗的模擬結果中也可以確定所提方法的有效性。

關鍵詞：向量量化，LBG, APSO-VQ

Abstract

The traditional LBG method for vector quantization (VQ) codebook design is a pure iterative optimization procedure, where an initial codebook is continually refined at each iteration to reduce the distortion involved in coding vector and a given training set. However, such interactive type learning algorithms will easily converge the final result to a local minimum while the quality of the initial codebook is not good. In our research, a new PSO-based codebook design method is proposed for building VQ scheme. The presented method combines the advantages of

gradient descent and PSO, called APSO-VQ algorithm, can apply the gradient descent part to improve PSO to quickly achieve a near-optimal solutions. The proposed APSO-VQ method provides the higher quality than conventional LBG methods in the application of building image compressed system. Several experimental comparisons of both discussed methods in image compressed design for “Lena”, “Airplane”, “Cameraman” and “peppers” are also presented in our research.

Keywords: VQ, LBG, APSO-VQ

二、簡介

1.基本定義

VQ (Vector Quantization) 在影像壓縮技術中算是一種非常基本的失真影像壓縮法 [1-3]，它比起純量量化器 (scalar quantization scheme) 具有更優越的碼率-誤差效能 (Rate-Distortion Performance)。關於 VQ 的數學表示式可描述如下：

$$\begin{aligned} Q: R^n &\rightarrow C \\ C &= \{c_1, c_2, \dots, c_K\}, c_i \in R^n \quad \forall i = 1, 2, \dots, K \\ Q(x) &= c_j \quad \text{if} \quad d(x, c_j) \leq d(x, c_i) \quad \forall i, j = 1, 2, \dots, K \end{aligned} \quad (1)$$

此處 R^n 是指 n 維的向量空間， $d(x, c_j)$ 指 x 與 c_j 間的距離量測，而 $C \subset R^n$ 則是由 K 個 n 維的碼向量 $c_j = (c_{j1}, c_{j2}, \dots, c_{jn})$ 所構成的碼簿。向量量化器可分解為兩個搭配的子系統：編碼器 E 以及解碼器 D ；編碼器映射 $E: R^n \rightarrow I$ ，且 $I = \{1, 2, \dots, K\}$ ，而解碼器則是映射 $D: I \rightarrow C$ 。依此定義我們可得到 $Q(x) = D(E(x))$ 。一個具有 K -point 的向量量

化器將 R^n 的向量空間切割為 K 個區域，其中 j -th 個區域 $S_j, j \in I$ 定義為

$$S_j = \{x \in T : Q(x) = c_j\}, \quad (2)$$

其中 $T = \{x_i, i = 1, 2, \dots, N\}$ 代表訓練向量集。

2. 最佳化法則

對於如何決定合適之 $C = (c_1, c_2, \dots, c_K)$ 以及 $P = (S_1, S_2, \dots, S_K)$ 以使得平均誤差 $D_{ave} = \frac{1}{N \cdot n} \sum_{i=1}^N \|x_i - Q(x_i)\|^2$ 達到最小的求解問題，它必須滿足下述兩個條件。

Criterion 1) 最近臨域條件:

在第 t 次迭代過程中，編碼區域 S_j^t 應由訓練向量中距離 c_j^t 比其他碼向量更近的所有向量所組成：

$$S_j^t = \mathbf{V}(c_j^t) \\ = \{x \in T : \|x - c_j^t\|^2 \leq \|x - c_u^t\|^2, \quad u = 1, 2, \dots, K\} \quad (3)$$

其中 \mathbf{V} 代表最近臨域切割。

Criterion 2) 群聚中心條件:

碼向量 c_j^{t+1} 應為所有歸屬於編碼 S_j^t 中的訓練向量之平均值：

$$c_j^{t+1} = \mathbf{C}(\mathbf{V}(c_j^t)) \\ = \frac{\sum_{x_i \in S_j^t} x_i}{\sum_{x_i \in S_j^t} 1}, \quad j = 1, 2, \dots, K \quad (4)$$

此處 $\mathbf{C}(\mathbf{V}(c_j^t))$ 代表基於距離量測公式 $d(x, c_j)$, $x, c_j \in R^n$ 所得到之切割區域 $\mathbf{V}(c_j^t)$ 的群聚中心。

而廣為人知的 LBG VQ 演算法即為基於輪替求解上述兩個最佳化法則的一種迭代式學習演算法[3]。

三、以 PSO 設計向量量化碼簿

在 PSO 演算法中，族群中的每一個個體在解空間(solution space)的移動位置是根據自我過去最佳經驗 ($pbest_p$) 與群體最佳行為 ($gbest$) 進行機率式的修正調整，其式子可描述如下[4,5]:

$$V_{p,d}^{t+1} = \tau \cdot V_{p,d}^t + c_1 * rand() * (pbest_{p,d}^t - X_{p,d}^t) \\ + c_2 * rand() * (gbest_d^t - X_{p,d}^t) \quad (5)$$

此處 X_p 是個體的位置向量(解集合), p 是指向族群中某一個個體的指標, d 是向量維

度, τ 用來決定迭代過程中移動速度遞減的幅度, t 意指現在狀態而 $t+1$ 代表下一次迭代的狀態, c_1 和 c_2 則是控制個體受到自我過去經驗 ($pbest_p$) 與群體最佳解 ($gbest$) 比率的參數。當個體的移動速度 V_p 確定之後, 其位置向量 X_p 則可據以更新為

$$X_{p,d}^{t+1} = X_{p,d}^t + V_{p,d}^{t+1}. \quad (6)$$

接下來我們將敘述 APSO-VQ (Alternative PSO-VQ algorithm) 的設計步驟如下[6,7]:

Step1) 隨機產生初始群中所有個體的位置向量 X_p 以及移動速度 V_p , 此處位置向量 $X_p = \{c_{p,j}, j = 1, 2, \dots, K\}$ 乃尺寸大小為 K 的碼簿, 而每一個碼向量 $c_{p,j} = (c_{p,j1}, c_{p,j2}, \dots, c_{p,jn})$ 均為 n 維的向量。

Step2) 計算每一個個體之適應函數值, 我們使用指數距離來度量資料集之中 N 筆資料樣本與 K 個群聚之間的距離[8,9]。對於訓練資料 $x_i \in T$, 我們依下述法則分派 x_i 到與其最相似的碼向量 $c_{p,j}, j \in I = \{1, 2, \dots, K\}$ 所歸屬的區域:

$$1 - \exp(-\beta \|x_i - c_{p,u}\|^2) = \min_u \{1 - \exp(-\beta \|x_i - c_{p,u}\|^2)\}, \\ u = 1, 2, \dots, K. \quad (7)$$

$$\text{此處 } \beta = \left(\frac{\sum_{i=1}^N \|x_i - \bar{x}\|^2}{N} \right)^{-1}, \quad \bar{x} = \frac{\sum_{i=1}^N x_i}{N} \quad (8)$$

而 APSO-VQ 演算法之適應函數 ψ 設計如下:

$$\psi = \lambda \cdot \left(\sum_{j=1}^K \sum_{x_i \in S_{p,j}} (1 - \exp(-\beta \|x_i - c_{p,j}\|^2)) \right) + \gamma)^{-1} \quad (9)$$

其中 λ 是正實數值, 而 γ 是一個極微小的數值。

Step3) 將每一個個體目前求得之適應函數值與其所記憶之最佳適應函數值進行比較, 若目前之適應函數值較之前最佳結果為佳, 則以目前位置取代個體記憶之最佳所在位置, 以目前適應函數值取代個體記憶之最佳值:

$$pbest_p^{t+1} = \begin{cases} X_p^{t+1} & \text{if } \psi(X_p^{t+1}) \geq \psi(pbest_p^t) \\ pbest_p^t & \text{if } \psi(X_p^{t+1}) < \psi(pbest_p^t) \end{cases} \quad (10)$$

Step4) 比較個體最佳解所求得的適應函數是否優於群體所記憶的最佳值, 若判斷條件成立則將群體所記憶之最佳位置與最佳值重

設為目前的結果，反之群體最佳解維持原先狀態：

$$gbest^{t+1} = \begin{cases} pbest_p^{t+1} & \text{if } \psi(pbest_p^{t+1}) \geq \psi(gbest^t) \\ gbest^t & \text{if } \psi(pbest_p^{t+1}) < \psi(gbest^t) \end{cases} \quad (11)$$

Step5) 將 $gbest^{t+1} = \{c_{gbest,j}^{t+1}, j=1,2,\dots,K\}$ 更新為 $gbest^{t+1*}$ [8-10]：

$$gbest^{t+1*} = gbest^{t+1} + s \cdot (gbest_{cu}^{t+1} - gbest^{t+1}), \quad (12)$$

而 $gbest_{cu}^{t+1}$

$$= \{C(V(c_{gbest,j}^{t+1})), \quad j=1,2,\dots,K\},$$

$$= \left\{ \frac{\sum_{x_i \in S_{gbest,j}^{t+1}} (\exp(-\beta \|x_i - c_{gbest,j}^{t+1}\|^2)) \cdot x_i}{\sum_{x_i \in S_{gbest,j}^{t+1}} \exp(-\beta \|x_i - c_{gbest,j}^{t+1}\|^2)}, \quad j=1,2,\dots,K \right\} \quad (13)$$

式中 s 是介於 1~2 之間的尺度因子，此處我們所使用的尺度更新策略(scaled centroid update scheme)是讓 $gbest$ 沿著比傳統群聚分析演算法更大步距的區域梯度 (local gradient) 方向進行調整。

Step6) 根據 Equation 5 和 Equation 6 調整所有個體移動的速度與位置。

Step7) 重覆 Step2)-Step6) 直至達到預設的迭代次數為止。

三、實驗結果

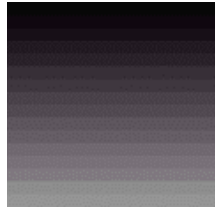
為驗證上述所提架構之有效性，我們將分別使用一些影像資料進行測試；在實驗中，我們預設 PSO 之參數 $c_1 = c_2 = 1.5$ ，而且為了達

到較佳的收斂效果，我們在每次迭代過程中令 $\tau = 0.75$ ，以逐漸調整個體移動的速度值。在實驗中我們使用 Means Square Error 來對重建後的影像品質進行鑑定，其定義如下：

$$MSE = \left(\frac{1}{M \times N} \right) \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - \hat{f}_{ij})^2 \quad (14)$$

此處 f_{ij} 是指原始影像在座標 (i, j) 的像素值，而 \hat{f}_{ij} 是編碼影像在座標 (i, j) 的像素值， $M \times N$ 為影像大小。

Example 1. Fig.1(a)以及 Fig. 2(a)所示的圖形分別為具有 16 種灰度值的橫條狀與格狀圖案之灰階影像。當我們將取值範圍區分成 2^B 個區間，而且分別設定 $B=1,2,3,4$ 時，可以把像素的灰度值分成 $K=2,4,8,16$ 個區間以量化原始影像；Fig.1 (b)-(e)以及 Fig. 2 (b)-(e) 所示為所提方法分別針對 $K=2,4,8,16$ 進行影像量化所得到的結果，從圖中可知其重建影像與原始影像之間的差距會隨著切割區間增大的逐漸逼近。Fig.1 (f)以及 Fig. 2 (f)則是顯示 APSO-VQ 在結合 step 5 所述之 scaled centroid update scheme 之後，收斂速度確實得到明顯的提升，有助於架構的快速收斂。



(a)



(b)



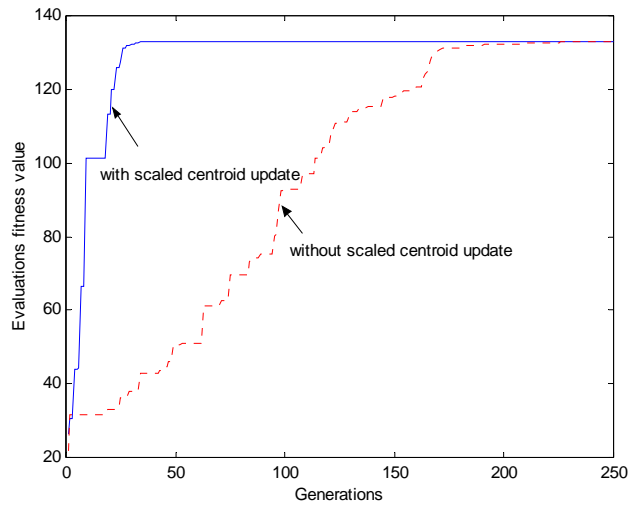
(c)



(d)



(e)



(f)

Fig. 1 具有 16 種灰度值之橫條狀灰階影像(a) 原始影像，(b) $n=1, K=2$ ，(c) $n=1, K=4$ ，(d) $n=1, K=8$ ，(e) $n=1, K=16$ ，(f) 收斂性能比較($K=16$)。



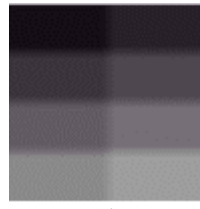
(a).



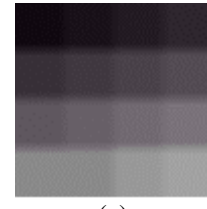
(b)



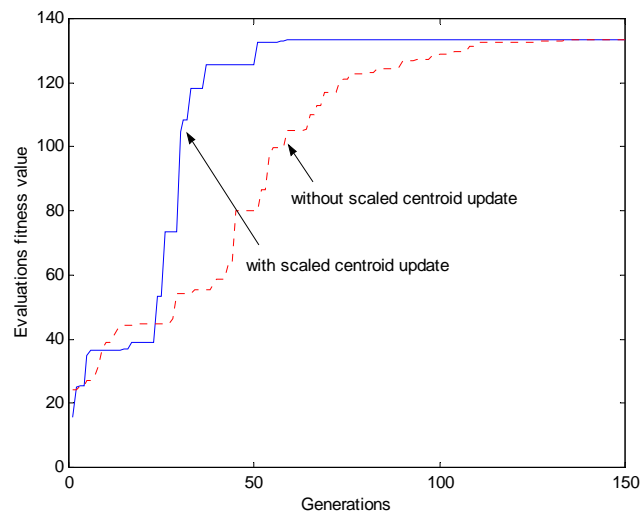
(c)



(d)



(e)



(f)

Fig. 2 具有 16 種灰度值之格狀灰階影像(a) 原始影像，(b) $n=1, K=2$ ，(c) $n=1, K=4$ ，(d) $n=1, K=8$ ，(e) $n=1, K=16$ ，(f) 收斂性能比較($K=16$)。

Example 2. 本實驗中，我們以知名的“Lena”，“Airplane”，“Cameraman” and “Peppers” 4 張 128×128 的灰階影像作為測試的對象(Fig. 3 (a)-(d))，所有的影像在進行 VQ 編碼時均被切割為 4096 筆 2×2 的子區塊。而在模擬的過程中，我們分別設定碼簿大小為 $K=16, 32, 64, 128, 256$ ，並各自執行 APSO-VQ 以及 LBG 三次以得到其 MSE 對應碼簿大小的曲線圖，如 Fig. 4(a)-(d) 中所示。很明顯的，

從實驗結果中我們發現 LBG 很容易受到初始值的影響而掉入區域最佳解中，因此在不同的實驗中它可能得到差異很大的結果；相對的，APSO-VQ 並不受到初始值的影響，它在不同的實驗中都能穩定的得到近似最佳解；而且即使面對碼簿大小的增加和改變，該方法仍能得到良好的品質，因此它也適合應用在大型資料集中。

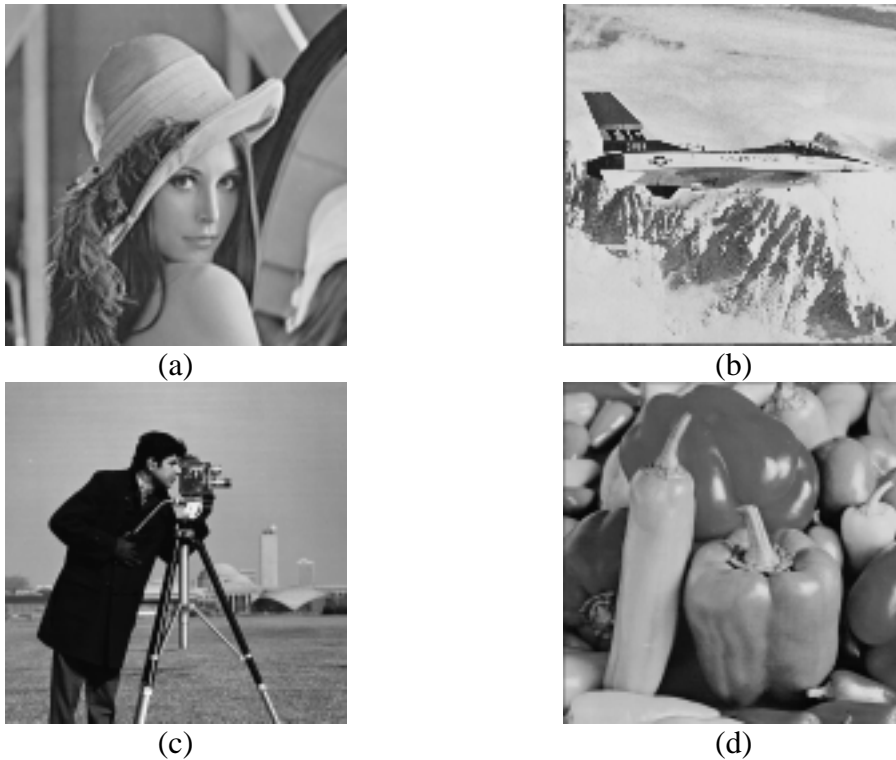


Fig. 3. 訓練影像 (a) Lena, (b) Airplane, (c) Cameraman, (d) Peppers.

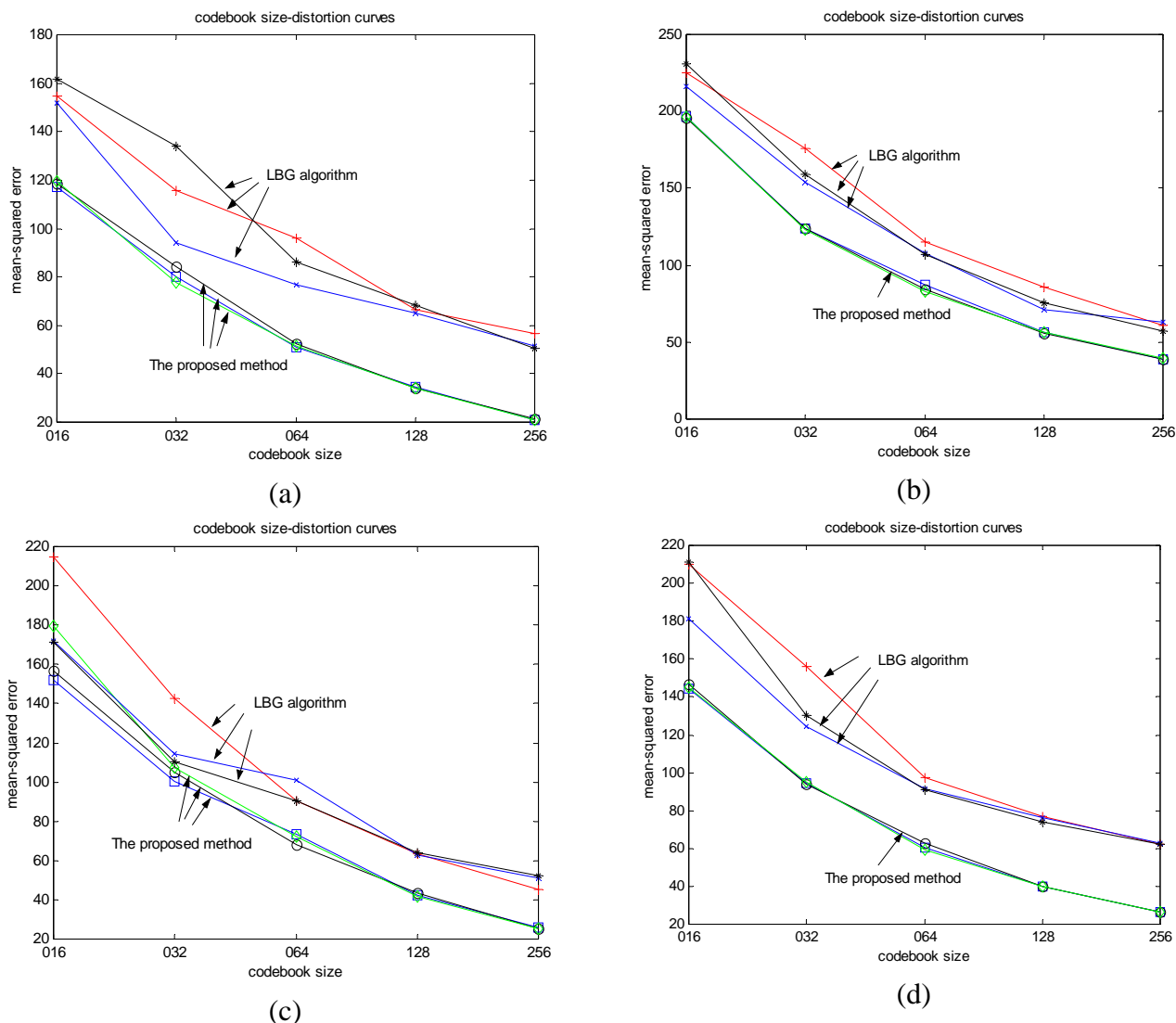


Fig. 4. MSE 對應 codebook size 的曲線圖 (a) Lena , (b) Airplane , (c) Cameraman , (d) Peppers

四、結論

本計畫提出一個以演化式計算為基礎的向量量化演算法 APSO-VQ，我們以更具健性的指數距離進行訓練資料與碼向量之間相似性的量測，並將總體誤差指標轉化成引導 PSO 群體行動中每一個個體 (particle) 移動方向的適應函數 (fitness function) 評估值，透過多個體平行搜尋的方式來快速找到能滿足問題的近似最佳解，以得到高品質的向量量化碼簿設計。而為了進一步加快向量編碼的搜尋速度，在計畫中我們也結合梯度下降法 (Gradient descent) 實現混合式架構，以加速系統的收斂速度。數張灰階影像的實際驗證說明了所提架構的有效性與良好性能表現。

五.參考文獻:

- [1] H. Abut, *Vector Quantization*, IEEE Press, 1990.
- [2] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer, 1992.
- [3] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for Vector Quantizer Design," *IEEE Trans. Communications*, pp. 84-95, Vol.28, 1980.
- [4] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," *Proc. of IEEE international Conference on Neural Networks (ICNN)*, Vol. IV, pp.1942-1948, Perth, Australia, 1995.
- [5] R. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," *Proc. 6th Int. Symposium on Micro Machine and Human Science*, pp.39-43, 1995.
- [6] Ching-Yi Chen, Kuang-Yuan Chen, and Fun Ye, "Evolutionary-based Vector Quantizer Design," *ICSS2005 International Conference on System & Signals*, I-Shou University, Taiwan, April, 2005, pp.649-654..
- [7] Ching-Yi Chen and Fun Ye, "K-means Algorithm

Based on Particle Swarm Optimization”, in *ICICS 2003*, pp.1470-1475, Dec. 2003.

- [8] Kuo-Lung Wu, Miin-Shen Yang, “Alternative c-means clustering algorithms, ” *Pattern Recognition*, vol. 35, pp. 2267-2278, 2002.
- [9] Fun Ye, Ching-Yi Chen, “Alternative KPSO-Clustering Algorithm”, *Tamkang Journal of Science and Engineering*, Vol. 8, No. 2, pp. 165-174,

2005.

- [10] Daeryong Lee, Seongjoon Baek, and Koengmo Sung, “Modified K-means Algorithm for Vector Quantizer Design”, *IEEE Signal Processing Letters*, pp. 2-4, Vol. 4, NO. 1, JANUARY 1997.

附件一

已投稿至 ICSS 2005 研討會並且獲得接受及於會議中
發表論文

Evolutionary-based Vector Quantizer Design

Ching-Yi Chen, Kuang-Yuan Chen, and Fun Ye

ICSS 2005 International Conference on Systems & Signals

APRIL 28-29, 2005, Kaohsiung, Taiwan, R.O.C.