

一種全域與區域搜尋整合模型的向量量化演算法

蔡篤校
修平技術學院
資訊管理系
dtsai@mail.hit.edu.tw

黃暢卿
國立中興大學
資訊科學與工程學系
phd9308@cs.nchu.edu.tw

洪國寶
國立中興大學
資訊科學與工程學系
gbhorng@cs.nchu.edu.tw

摘要

近年來，影像壓縮已成為影像處理領域中極為重要的一環，在各種影像壓縮方法中向量量化(Vector Quantization, VQ)為最常見的影像壓縮方法之一，而 LBG 演算法亦常成為訓練編碼簿(codebook)的測試標準，由於 LBG 演算法的成果過度依賴初始編碼字(initial codeword)，初始編碼字的不當選擇將導致無法產生好的編碼簿。近年來雖然有許多改良 LBG 的文獻陸續提出，然而絕大多數的演算法仍然與 LBG 同樣採取隨機方式(random)來選擇初始編碼字，所以仍然可能無法得到好的編碼簿或者增加收斂的運算時間。因此，本文將提出全域取樣模型(Global Sampling)，以整幅影像全域觀點進行取樣，作為初始的編碼字以降低局部初始值的不當選擇，然後再經由區域 LBG 演算法訓練出最終的編碼簿。經實驗結果證實我們所提的方法皆優於最近文獻 ELBG 與 AILBG 等演算法，能有效提高影像的壓縮品質。

關鍵詞：向量量化、全域取樣、LBG 演算法、初始編碼簿

Abstract

The vector quantization (VQ) method has been recently applied to image compression which is an important research area of image processing. The LBG algorithm is one of the most popular methods and is always used by researchers in VQ as the standard codebook design model for benchmarking. The codeword generation process in LBG is computation intensive and the distortion rate is affected by the initial codewords and may be terminated in suboptimal codebooks. Therefore, many fast algorithms for codebook search have been developed. However, these algorithms depend on the randomized initial codebook, and often obtain a local optimal codebook with intensive computation. In this paper, a

global search method based on histogram threshold is proposed to find better initial codewords for local search algorithm such as LBG. The experimental results show that our method obtains better image quality than other VQ algorithm, such as ELBG and AILBG.

Keywords: vector quantization、global sampling、LBG algorithm、initial codebook

1. 前言

在影像壓縮的領域中，向量量化編碼法(vector quantization, VQ) [1-4]是最常見的影像壓縮技術之一，向量量化編碼法是由 Linde 等學者[1]於 1980 年所提出，其主要精神在於將一張或若干張數位影像切割成數個不相交的影像區塊(image block)，從這些影像區塊中，挑選出一些具有代表性的影像區塊所組成的集合，稱為編碼簿(codebook)，編碼簿中的每個元素區塊則被稱為編碼字(codeword)。

當我們要對一張數位影像壓縮時，首先將該數位影像切割為一群不重複的影像區塊。對於每個影像區塊，我們會將該影像區塊與事先產生的編碼簿作比對，從編碼簿的所有編碼字中，找出一個和該影像區塊最接近的編碼字，接著記錄該編碼字位於編碼簿中的索引值(index)來當作壓縮碼，通常我們使用歐基里德距離(Euclidean distance)來計算兩個影像區塊之間的相似度。

當我們依序為每一個影像區塊找到編碼簿中的最相近碼向量，也就表示已經找到個別影像區塊所對應的最近似索引值，便完成對該數位影像的壓縮程序。也就是說當我們完成一張數位影像中所有影像區塊的壓縮工作後，我們便會得到很多的索引值，而這些索引值所構成的集合，稱之為索引表(index table)。

當我們要對一張數位影像解壓縮時，解碼端根據索引表所記錄的索引值對應到編碼簿中的編碼字，也就是使用查表法就可重建其對應的影像區塊，如此一來便完成解碼的工作。

了解有關向量量化編碼法的編碼和解碼的

過程之後，我們知道向量量化編碼法具有低位元率的優點並且其解壓縮的程序非常簡潔。但是它的缺點在於解壓縮後的影像品質會因編碼簿的訓練迥異，而有不同的還原品質。過去多年來，有許多文獻針對編碼簿的設計提出許多方法，其中最具有代表性、最被廣泛使用的是由 Linde、Buzo 與 Gray 三位學者於 1980 年所提出的 LBG 演算法[1]，由於 LBG 演算法的訓練成果與初始向量的選取有極密切的關係，若是初始向量的選擇太過集中，此往往會陷入局部的最佳化 (local optimal)，所訓練出來的編碼簿通常不具有很好的代表性。所以陸續有多種改良的方法被提出[5-11]，其中 Hu 與 Chang 在 1999 年提出細胞分裂法[9]，其主要精神是將初始編碼簿儘量分開，以期望打散各個初始向量，來改善 LBG 演算法初始值過於集中的缺點。Patane 與 Russo 在 2001 年提出 ELBG 方法[10]，首先它在 LBG 方法中定義了編碼字效率 (utility of codeword)，於演算法中將編碼字效率低於 1 的低效率編碼字 (low-utility codewords)，尋找與其鄰近而且編碼字效率接近 1 者，將兩群合併，在此同時再選擇一個效率高於 1 的高效率編碼字 (high-utility codewords) 進行分裂，也就是將失真較小的一群與接近平均失真的一群，併成編碼字效率靠近 1 的失真群，再選擇失真大的一群進行分裂，如此循環計算一直到整體失真符合收斂條件為止，該演算法能在不改變整體群組數目的前提下同時降低整體的失真。Shena 與 Hasegawa 於 2006 年提出 AILBG 方法[11]，在 AILBG 演算法中定義了調適性距離函數 (adaptive distance function) 有別於一般常用的歐基里德距離，以確保群組與群組間的距離大於群組內每個元素間的距離，也就是要讓訓練後的編碼簿能夠維持群組間相互遠離、群組內相互凝聚的性質，以求得整體性的最佳分布 (globally optimal)。

上述各種 LBG 的改良方法雖然有改進編碼簿的還原品質，然而各演算法仍然與 LBG 同樣採取隨機方式來選擇初始編碼字，所以仍然可能無法得到很好的編碼簿或者增加收斂的運算時間。因此，本篇文章提出全域取樣模型 (Global Sampling Model) 對全幅影像區塊的相關性進行取樣，以整幅影像取樣作為初始的編碼字來降低局部初始值的不當選擇，然後再經由 LBG 演算法訓練出最終的編碼簿。接下來的章節內容如下：第 2 章我們將回顧相關技術，第 3 章我們將介紹本篇論文所提出的方

法，第 4 章我們將提供相關的實驗數據，第 5 章則對本篇論文的貢獻作結論以及思考未來的研究方針。

2. 相關技術回顧

在這一章節裡，我們將簡單介紹 VQ 中最常見的 LBG 演算法。

2.1 LBG 演算法

一般要進行向量量化之前，必須事先建構好一個適當的編碼簿，而編碼簿訓練成果的好壞將會決定影像還原品質的高低，所以編碼簿訓練的目的就是要萃取出那些最能代表影像特色的碼向量集合，來構成量化用的編碼簿，其演算法如下：

Step 1: (初始化)

由 N_p 個訓練集之中 $X = \{x_1, x_2, \dots, x_{N_p}\}$ ，隨機選取 N_c 個向量 $C_0 = \{c_1, c_2, \dots, c_{N_c}\}$ 作為初始的編碼字，其中 C_0 表示原始疊代的訓練集合， c_i 表示集合的元素總共有 N_c 個。

Step 2: (歸群計算)

將所有原始影像的向量碼 x_p 分別與編碼字

$$c_q \text{ 計算歐氏距離 } d(x_p, c_q) = \sum_{i=1}^k ((x_p)_i - (c_q)_i)^2,$$

然後將 x_p 歸屬於與其距離最小的群組 G_q ，其中 k 表示向量的維度。

Step 3: (重新計算或結束)

若是整體失真未能降低則結束執行。

$$\text{計算各群的新重心 } c_m = \frac{1}{|G_m|} \sum_{x_i \in G_m} x_i,$$

go to Step 2.

3. 本篇論文所提出的方法

由於隨機選擇的初始向量過於集中時，將導致 LBG 演算法陷入局部最佳化，所訓練出來的編碼簿通常不具好的代表性，近年來，各種改良 LBG 的相關文獻雖然有改進編碼簿的還原品質，然而各演算法大多仍然與 LBG 同樣採取隨機方式來選擇初始編碼字，所以仍然可能無法得到很好的編碼簿或者增加收斂的運算時間，為了改進這個缺點，在這個章節，我們提出全域取樣模型 (Global Sampling Model)，運用 GSM 所提理論對整幅影像做全域取樣，由於 GSM 是基於 Otsu[12,13]理論為出發點，於每群組分裂時皆搜尋最佳的門檻值來作為分群的切割點，而此分裂的最佳切割點代表著，該分裂的兩個子群間向量平均能夠相

互遠離、兩個子群內每個向量能夠相互凝聚，所以能保證該次分裂的兩個子群為最佳分布。

GSM 產生編碼簿的方法與 LBG 演算法最大不同處，在於編碼字的選擇並非採用隨機方式挑選，而是運用 Otsu 的理論計算出該群的門檻值，然而，欲得到足夠的群組個數時，必須由原始群組經過一層層的分裂，最後才能得到 N_c 個終端節點，隨著編碼簿尺寸的增大，分裂路徑的平均長度亦隨之增加，而逐層間的累積誤差也就愈來愈大，如此將導致編碼簿的還原品質下降。另一方面，LBG 演算法訓練編碼簿的好壞，則是倚賴於初始向量的選擇，過於局部的初始值，所訓練出來的編碼簿還原品質必定不佳，但若是能有極佳的初始向量，經 LBG 演算法訓練後能保證產生局部最佳化(local optimal)的編碼品質。所以我們提出全域取樣模型的 LBG 演算法(GSMLBG)，就是藉由 LBG 演算法來修正 GSM 逐層間的累積誤差，或者說是倚靠 GSM 的全域取樣提供給 LBG 理想的初始訓練向量，其演算法如下：

Step 1: (全域搜尋)

執行 GSM 挑選出 N_c 個編碼字。

Step 1.1: (初始化)

將整個訓練集 $X = \{x_1, x_2, \dots, x_{N_p}\}$ 視為單一個群組。

Step 1.2: (群組分裂計算)

運用 Otsu 理論計算該群的門檻值 t_m^* 。利用原來重心 c_q 與門檻值 t_m^* 產生虛擬重心 $c_q' = 2t_m^* - c_q$ 。

將該群的向量碼 x_p 分別與原來重心 c_q 以及虛擬重心 c_q' 計算歐氏距離，然後將 x_p 歸屬於與其距離較近的群組 G_q 或 G_q' 。

Step 1.3: (繼續分群或結束)

若是群組個數已達到要求，則輸出所有群組的重心來作為編碼簿。

否則，挑選 Weight 最大者 $\text{Max}(W_m)$ 的群組進行群組分裂， $W_m = |G_m|^2 \times d(c_q, t_m^*)$ ， $|G_m|$ 表示該群的元素個數，go to Step 1.2。

Step 2: (收斂計算)

執行 LBG 訓練迴圈為 15 次。

4. 實驗分析

在這個章節，我們分別以隨機初始值 LBG、GSM 以及 GSMLBG 來訓練編碼簿。我們使用 4 張大小為 512*512 的灰階數位影像，

每張影像分割為互不相交且大小為 4*4 的影像區塊，也就是將每張影像分割成 16384 個 16 維的影像區塊，作為實驗的評估(圖 1)，所訓練的編碼簿大小分別為 128、256、512 以及 1024 等 4 種不同尺寸的編碼簿，所有的實驗都是使用 C++ 程式語言於 AMD Athlon 64 Dual 的個人電腦執行，在解壓縮後的還原品質我們採用 PSNR 值作為評估標準。

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE}$$

$$MSE = \left(\frac{1}{M \times N} \right) \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - \bar{x}_{ij})^2$$

其中 x_{ij} 與 \bar{x}_{ij} 分別代表在座標 (i, j) 的原始像素值與經過處理的像素值， M 與 N 則分別表示影像的寬度與高度。

表 1 為隨機初始值 LBG 演算法的還原品質，其中我們設定訓練迴圈的終止條件為 15 次，且每張影像都執行 10 次隨機測試，以求得最佳、最差以及 10 次平均值，GSM 以及 GSMLBG 演算法的還原品質如表 2 及表 3，表 4 為 GSMLBG 提升 GSM 還原品質的比較表，GSM、LBG 及 GSMLBG 的執行時間列於表 5，最後我們以本篇文章所提的方法與最近文獻 ELBG 與 AILBG 比較還原品質做成表 6。

由表 1 與表 2 中不難觀察到，GSM 的影像還原品質皆能優於 LBG 的最佳還原值，而且隨著編碼簿的尺寸增加 GSM 更是具有其優勢，例如“Girl”編碼簿尺寸為 1024 時，GSM 已能優於 LBG 達 4dB 以上(34.979 - 30.898 = 4.081 dB)，由此可知 LBG 演算法受限於隨機初始值的選擇，即使增加編碼簿尺寸也很難有好的還原品質。

由表 2 中的“Lena”與表 7 相比較，GSM 除了高壓縮比的 256 編碼簿尺寸能略優於 ELBG 與 AILBG，其餘都略遜於 ELBG 與 AILBG。然而，GSM 經過 LBG 訓練後的編碼簿已能大幅提升其還原品質，相較於表 2 與表 3 其提升值列於表 4，由表 4 得知，當編碼簿尺寸由 128 增加到 1024 時，其平均提升品質由 0.748 dB 增加至 1.291 dB，由此可驗證 LBG 演算法確實能修正 GSM 逐層間的累積誤差。最後我們以本篇文章所提的方法與最近文獻 ELBG、AILBG 作比較，由表 6 中不難發現，經過我們所提的方法修正 GSM 的缺點後，在各種尺寸編碼簿的影像還原品質皆能優於兩者的還原能力。更值得注意的是，我們所提的方法在“時間”的執行效能上等同於 LBG 的複雜度，皆能

優於執行時間更長久的 ELBG 與 AILBG 等演算法。

5. 結論

在本篇論文中，我們運用 GSM 與 LBG 兩種演算法的互補特性，提出一個簡單且有效率之一全域取樣模型的 LBG 演算法(計算複雜度為 $O(N_p N_c)$)，我們所提的方法 GSMLBG，不但能夠改善 LBG 初始編碼向量的不當選擇導致無法訓練出好的編碼簿，而且改進了 GSM 逐層累積誤差造成無法訓練好的編碼簿之缺點。本篇論文相較於各種隨機取樣的 LBG 演算法，不論是大或小尺寸的編碼簿都有很好的改善表現，甚至在“Girl”編碼簿尺寸為 1024 時，能將傳統 LBG 演算法的還原品質提高 5 dB 以上($36.145 - 30.898 = 5.247$ dB)，同時相較於最近文獻 ELBG、AILBG 也都能優於兩者的還原能力。

參考文獻

- [1] Y. Linde, A. Buzo and R. M. Gray, “An Algorithm for Vector Quantizer Design,” *IEEE Transactions on Communications*, Vol. 28, pp. 84-95, January 1980.
- [2] R. M. Gray, “Vector Quantization,” *IEEE ASSP Magazine*, pp. 4-29, 1984.
- [3] N. M. Nasrabadi, and R. A. King, “Image Coding Using Vector Quantization: A Review,” *IEEE Transactions on Communications*, Vol. 36, pp. 957-971, 1988.
- [4] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Boston, MA: Kluwer Academic Publishers, 1992.
- [5] C. M. Huang and R. W. Harris, “A Comparison of Several Vector Quantization Codebook Generation Approaches,” *IEEE Transactions on Image Processing*, Vol. 2, No. 1, pp. 108-112, 1993.
- [6] C. K. Chan and C. K. Ma, “A Fast Method of Designing Better Codebooks for Image Vector Quantization,” *IEEE Transactions on Communications*, Vol. 42, No. 2, pp. 237-242, 1994.
- [7] C. C. Chang and Y. C. Hu, “A Fast Codebook Training Algorithm for Vector Quantization,” *IEEE Transactions on Consumer Electronics*, Vol. 44, No. 4, pp. 1201-1208, 1998.
- [8] Y. C. Lin and S. C. Tai, “A Fast Linde-Buzo-Gray Algorithm in Image Vector Quantization,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 45, No. 3, pp. 432-435, 1998.
- [9] Y. C. Hu, and C.C. Chang, “A Progressive Codebook Training Algorithm for Image Vector Quantization,” *Proceedings of the Fifth Asia-Pacific Conference on Communications and Fourth Optoelectronics and Communications Conference*, Beijing, China, Vol. 2, pp. 936-939, 1999.
- [10] G. Patane, and M. Russo, “The Enhanced LBG Algorithm,” *Neural Networks*, Vol. 14, No. 9, pp. 1219-1237, 2001.
- [11] F. Shena, and O. Hasegawa, “An adaptive incremental LBG for vector quantization,” *Neural Networks*, Vol. 19, No. 5, pp. 694-704, 2006.
- [12] C. C. Huang, D. S. Tsai and G. B. Horng, “Efficient Vector Quantization Codebook Generation Based on Histogram Thresholding Algorithm,” *IHMSP 2008*, pp. 1141-1145, 2008.
- [13] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Trans. on System Man Cybernetics*, Vol. SMC-9, No. 1, pp. 62-66, 1979.

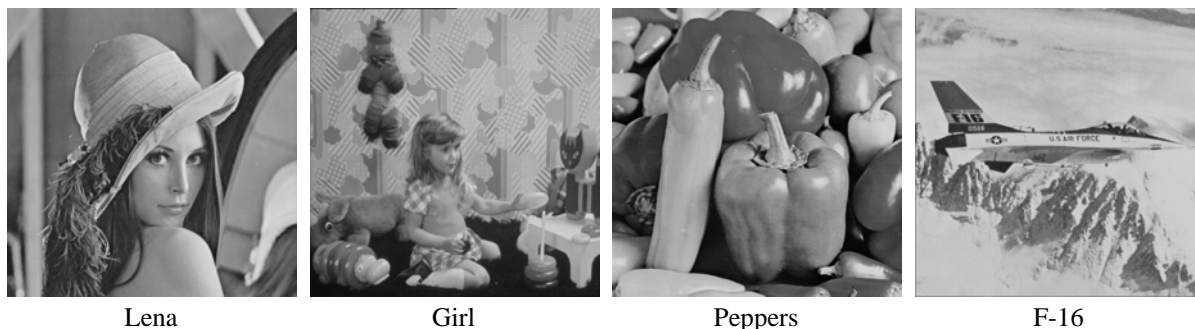


圖 1：四張 512×512 像素的灰階數位影像

表 1：LBG 影像還原品質，每張圖像做 10 次隨機初始值選取，每次做 15 回合歸群計算

編碼簿 大小	PSNR (dB)											
	128			256			512			1024		
Image	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min
Lena	28.966	29.437	28.215	29.714	30.063	29.110	30.136	30.545	29.685	30.499	30.756	30.194
Girl	29.068	29.495	28.332	29.772	30.489	28.990	30.142	30.778	29.586	30.428	30.898	30.178
Peppers	28.842	29.361	27.965	29.676	30.118	29.062	30.126	30.459	29.739	30.493	30.807	30.273
F-16	29.157	29.473	28.593	29.722	29.999	29.259	30.080	30.379	29.738	30.477	30.702	30.154

表 2：GSM 影像還原品質

編碼簿 大小	PSNR (dB)				
	Lena	Girl	Peppers	F-16	Avg
128	30.968	31.587	30.862	30.301	30.930
256	32.099	32.736	32.123	31.444	32.101
512	33.098	33.819	33.172	32.446	33.134
1024	34.244	34.979	34.304	33.685	34.303

表 3：GSMLBG 影像還原品質

編碼簿 大小	PSNR (dB)				
	Lena	Girl	Peppers	F-16	Avg
128	31.556	32.148	31.615	31.391	31.678
256	32.775	33.495	32.816	32.458	32.886
512	34.079	34.770	34.076	33.667	34.148
1024	35.451	36.145	35.541	35.239	35.594

表 4：GSMLBG 提升 GSM 比較表

編碼簿 大小	PSNR (dB)				
	Lena	Girl	Peppers	F-16	Avg
128	0.588	0.561	0.753	1.090	0.748
256	0.676	0.759	0.693	1.014	0.785
512	0.981	0.951	0.904	1.221	1.014
1024	1.207	1.166	1.237	1.554	1.291

表 5：三種演算法的執行時間

編碼簿 大小	RUN time (sec)		
	GSM	LBG	GSMLBG
128	0.465	4.155	4.620
256	0.895	8.295	9.190
512	1.851	16.560	18.411
1024	3.979	33.105	37.084

表 6：GSMLBG 相較於 ELBG、AILBG 以“Lena”作為評估標準

編碼簿 大小	PSNR (dB)		
	GSMLBG	ELBG	AILBG
256	32.775	31.94	32.01
512	34.079	33.14	33.22
1024	35.451	34.59	34.71