

利用 LBG 群聚法之向量量化影像壓縮演算法

A VQ Image Compression Algorithm by Using LBG Clustering

林俊宏

Chun-Hong Lin

中華技術學院電機系副教授

Department of Electrical Engineering China Institute of Technology

陳苑任

Yuan-Ren Chen

中華技術學院電機系二技部學生

摘 要

本論文將以向量量化(VQ)為影像特徵，探討以下兩個問題：(1) 在 LBG 訓練時，碼簿的大小對其所得之向量量化壓縮還原之影像失真、壓縮率有何關係？(2) 碼簿維度的大小對向量量化 LBG 壓縮之影像有何影響？針對這兩個問題，利用 MATLAB 設計程式，並實際地做出向量量化壓縮後的影像來做比較。

關鍵詞：向量量化、LBG 群聚法、影像失真、壓縮率。

ABSTRACT

Two problems were discussed in this paper. One is the relation between quantities of codebook and image distortion and the relation between quantities of codebook and compression rate. The other is the relation between dimensions of codevector and image distortion and the relation between dimensions of codevector and compression rate. Detailed experiments will explain the former two problems and give good results.

Keywords: vector quantization, LBG clustering, image distortion, compression rate.

I. 簡介

近年來，影像壓縮已成為影像處理領域中，極為重要的一環。諸多影像的儲存與傳送場合，如全球網路、視訊會議、影音電話、醫學影像、遙測等，都需要使用影像壓縮技術，以節省儲存空間或傳送時間。

向量量化(Vector Quantization, 簡稱 VQ)為影像壓縮演算法中很常出現的特徵量表示方式，VQ 根據輸入影像製造碼簿，再將輸入影像各個區塊以碼簿內的碼字來近似，藉此達到壓縮位元率的目的，但也因此造成影像的失真；如何在提高壓縮位元率和減低影像的失真之間取得平衡，就成了影像壓縮的一大課題。

LBG 分類演算法常用於製造碼簿，LBG 演算法使用的種子碼簿對分類結果的代表性影響很大，從而影響還原影像的失真度；此外，碼簿的大小對其所得之向量量化壓縮還原之影像失真、壓縮率的影響，碼簿維度的大小對向量量化 LBG 壓縮之影像的影響也必須正視。

因此，本論文將以向量量化(VQ)為影像特徵，探討以下兩個問題：(1)在 LBG 訓練時，碼簿的大小對其所得之向量量化壓縮還原之影像失真 壓縮率有何關係？(2)碼簿維度的大小對向量量化 LBG 壓縮之影像有何影響？針對這兩個問題，利用 MATLAB 設計程式，並實際地做出向量量化壓縮後的影像來做比較。

本論文章節如下：先簡單介紹向量量化和 LBG 演算法，然後詳述實驗方法與實驗結果，並針對上述兩個問題，根據實驗結果，加以討論。

II. 向量量化

設 $\mathbf{f} = [f_1, f_2, \dots, f_N]^T$ 表示一 N 維向量，它是由 N 個實數連續純量值 f_i 所組成。在 VQ 中， $\mathbf{f}$ 將被映射到另一個 N 維向量 $\mathbf{r} = [r_1, r_2, \dots, r_N]^T$ 。 $\mathbf{f}$ 的 VQ 是將一個 N 維向量空間分割成 L 個決策區域 $C_i, 1 \leq i \leq L$ ，每一個決策區域包圍一個重建向量 $\mathbf{r}_i$ 。在編碼的文獻中常將 $\mathbf{r}_i$ 稱為碼向量 (codevector)，並將碼向量所形成的集合稱為碼簿 (codebook)。

設 $\mathbf{f}_p$ 代表 $\mathbf{f}$ 量化的結果，則可寫成

$$\mathbf{f}_p = \text{VQ}(\mathbf{f}) = \mathbf{r}_i, \mathbf{f} \in C_i$$

其中 VQ 代表向量量化的動作。[1]

在向量量化編碼時，不同的碼向量分別給予不同的指標，藉由指標而非向量本身的傳送，可以降低表達此影像時所需花費的位元數，這就是以向量量化做影像壓縮減少資料量的原因。

圖 1 為向量量化的流程圖。其中編碼端及解碼端的碼簿均相同。每輸入一個向量，編碼端會到碼簿中去尋找和此向量有最短距離的碼向量，並將代表此一碼向量的指標傳送至解碼端。解碼端主要為查表的動作，將此一指標所對應的碼向量取出當作輸出向量，完成解碼動作。

III. LBG 演算法

有關影像壓縮之 LBG 演算法整理如下：[1]

(1) LBG 演算法是語出 Linde、Buzo 和 Gray 三位學者所提出的方法。

(2) 主要精神是從一堆編碼字的樣本中以分群法找出代表每個群的編碼字，再由這些具有代表性的編碼字形成一本編碼簿。

(3) 有關 LBG 演算法其步驟如下：

步驟 1: 初始化: 定出碼簿大小 L 、失真臨界值 ϵ 、初始碼簿 $R_0 = \{r_{i,0}; 1 \leq i \leq L\}$ 及訓練序列 $T = \{f_n; n=1, 2, \dots, N\}$, $N \gg L$ 。設疊代次數 $m=0$, 初始失真 $D_{-1} = \infty$ 。

步驟 2: 對碼簿 $R_m = \{r_{i,m}; 1 \leq i \leq L\}$, 找出訓練序列 T 的最小誤差分割, 即若

$$d(f_n, r_{i,m}) \leq d(f_n, r_{j,m}) \quad \forall j \neq i$$

$$\text{則 } f_n \in C_i$$

步驟 3: 計算平均失真 D_m

$$D_m = \frac{1}{N} \sum_{n=1}^N \min(d(f_n, r_{i,m}))$$

若 $(D_{m-1} - D_m) / D_m \leq \epsilon$, 則輸出碼簿 R_m , 退出疊代過程; 否則繼續。

步驟 4: 設屬於 C_i 的向量有 M_i 個, 則新估測的 r_i 是使 $\frac{1}{M_i} \sum_{f \in C_i} d(f, r_i)$ 為最小的值。若 $d(f, r_i) = (f - r_i)^T (f - r_i)$, 即平方誤差, 則 r_i 正是 M_i 個向量的算術平均值。

步驟 5: 取 $m=m+1$, 回到步驟 2。

(4) LBG 的訓練結果取決於初始的向量是否足夠分散; 若初始向量太過集中, 所訓練出來的編碼簿通常不具有很好的代表性。

IV. 實驗結果與討論

演算法程式以 IBM PC 為實驗平台,作業系統為 Windows XP,使用的程式語言為 Matlab。影像壓縮的好壞有兩個參數可作為判定的準則,其一為峰值訊雜比 PSNR :

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{\sum_i \|f_i - \hat{f}_i\|^2}} \right)$$

另一個為壓縮率 CR :

$$CR = \frac{\text{原始影像所需位元數}}{\text{壓縮影像所需位元數}}$$

由表一可以得知,在 LBG 訓練時,碼簿愈大時,其壓縮還原之後所得到的影像其失真越小,但相對的其壓縮率會越低;反之則是碼簿越小,其壓縮還原之後所得到的影像其失真越大,但其壓縮率則會越高。

另外由表一也可以得知,當碼簿維度愈大時,其壓縮率越高,也由於壓縮率越高,其影像壓縮所花費的時間也越低,但是其所付出的代價則是影像失真較高,這是因為每一個指標所代表的像素越多,則記錄一幅影像所必須花費的指標數目也就越少,相對的所需花費的總位元數也就越少。

最後在做完 LBG 演算法的程式後發現,碼簿大小越大,其付出的 LBG 演算法訓練的時間越長,這是由於 LBG 演算法的計算複雜度與訓練向量數 向量的維度、碼簿的大小、訓練循環次數的乘積成正比;實驗時, LBG 演算法的程式總共花費快一個小時的時間,大多時間花費在 VQ 函數副程式裡的迴圈程式,迴圈程式動輒要做上萬次,而且還不僅僅只有一個迴圈而已,再加上每一個碼簿都要做一次 LBG 演算法,真是要讓人等的快要瘋掉了。所以,以下探討有什麼更好的影像壓縮的方法可以加快演算時間,並且也可以得到很好的影像壓縮效果。

未來的研究課題將針對以下問題:[表二]

(1)對於 LBG 的訓練結果取決於初始的向量是否足夠分散;若初始向量太過集中,所訓練出來的編碼簿通常不具有很好的代表性;試著做做看並探討、研究這問題。

(2)有沒有更好的影像壓縮的方法可以改善影像壓縮之 LBG 演算法的缺點並且取代之。

要解決上述問題,細胞分裂法影像壓縮為研究課題。這個方法時常被提出來討論,因為細胞分裂法其主要精神是將初始編碼簿儘量分開,如此可以改善 LBG 演算法初始值過於集中的缺點,並且可以改進製作編碼簿的速度。細胞分裂法影

像壓縮具有以下優點：

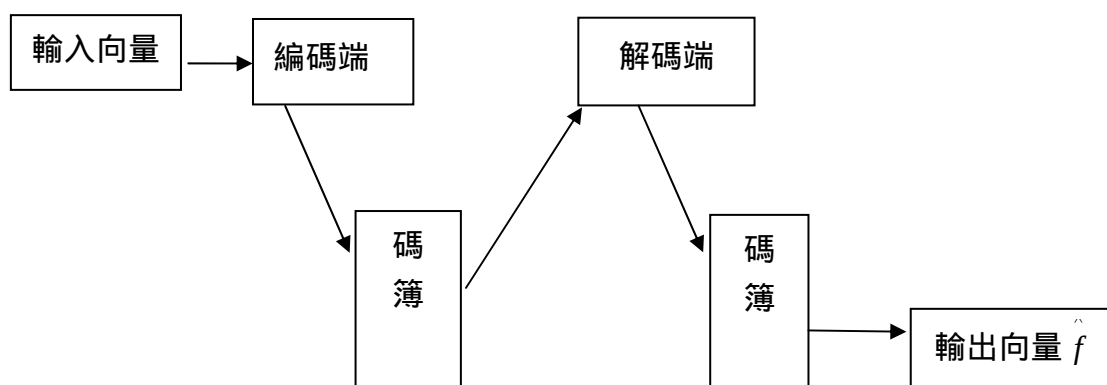
- (1) 利用樹狀結構漸進的方式產生編碼簿，可以不用每次都計算編碼簿個分群重心，因此可以有效的減少計算量。
- (2) 初始向量的選擇對編碼簿的結果，不會有絕對性的影響。
- (3) 較 LBG 演算法具有更好的效能，包括編碼簿可產生較佳的壓縮品質，以及製作編碼簿所花費的時間較少等優點。

致謝

感謝中華技術學院提供實驗設備與圖書、期刊，使本文得以順利完成。

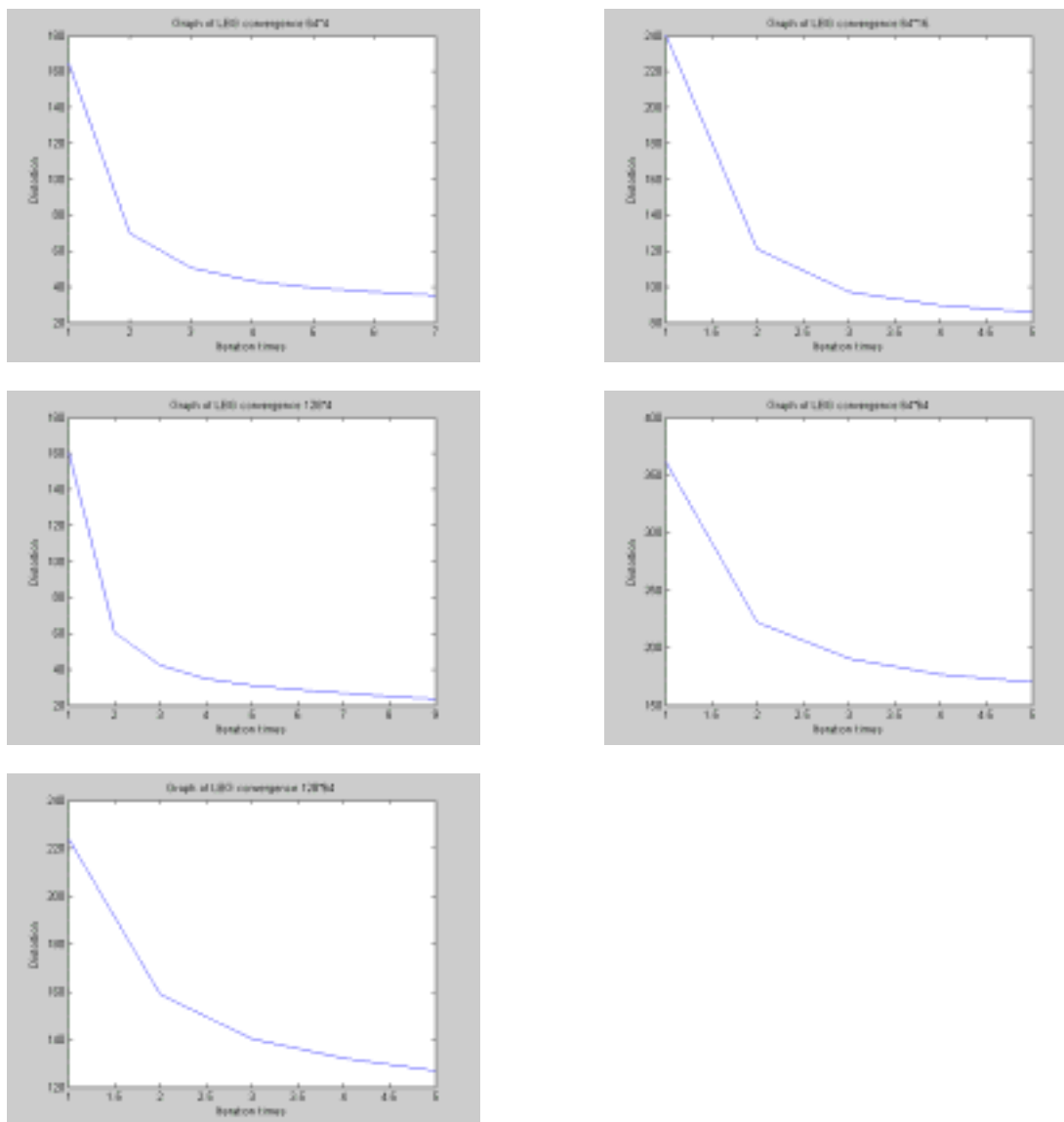
參考文獻

- [1] 繆紹綱, “數位影像處理-活用 MATLAB,” 全華科技圖書, 1999.
- [2] W.K.Pratt, “Digital Image Processing,” New York: Wiley, 1991.

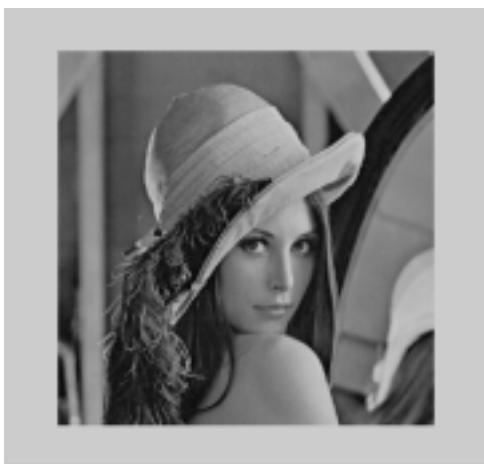


圖一、向量量化流程圖

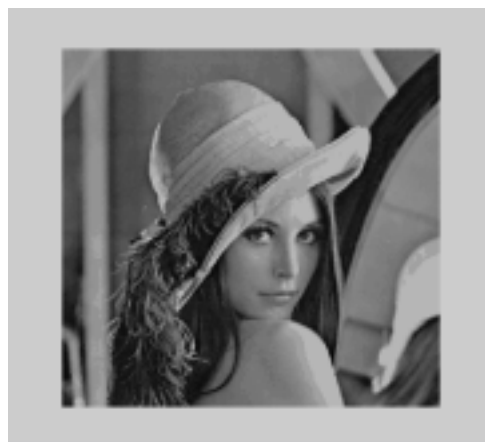
利用 LBG 群聚法之向量量化影像壓縮演算法



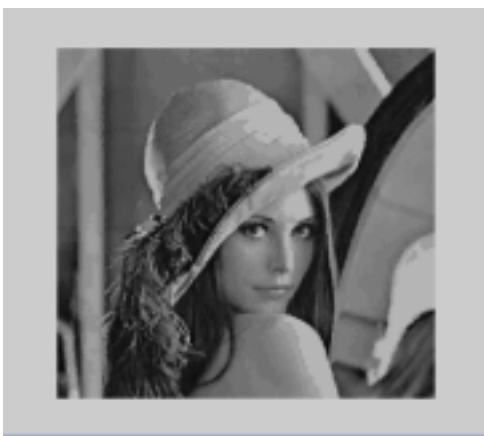
圖二(a-e)、碼簿與碼向量大小和收斂速度的關係



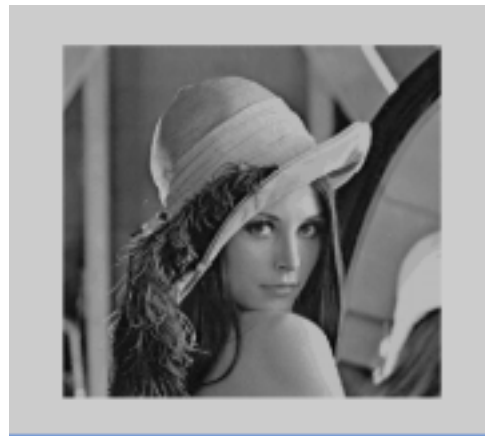
圖三、原始影像



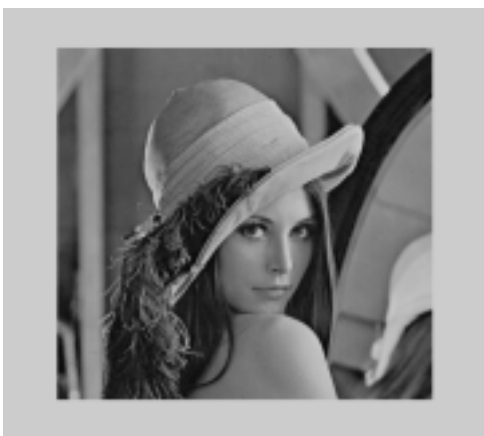
圖四、碼簿為 $cb_64 \times 64$ 之還原影像



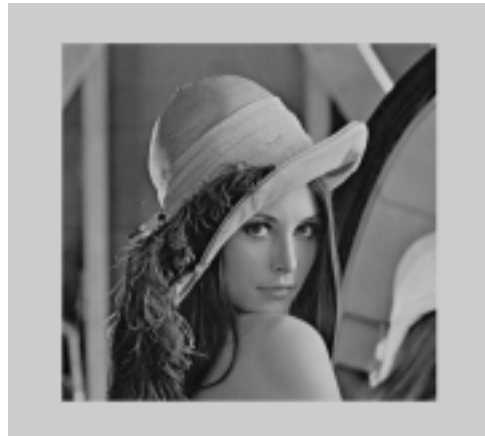
圖五、碼簿為 $cb_128 \times 64$ 之還原影像



圖六、碼簿為 $cb_64 \times 16$ 之還原影像



碼簿： $cb_64 \times 4$ 之還原影像



碼簿： $cb_128 \times 4$ 之還原影像

利用 LBG 群聚法之向量量化影像壓縮演算法

碼簿大小	cb_64x4	cb_64x16	cb_128x4	cb_64x64	cb_128x64
PSNR	32.7984	28.9406	34.5316	25.9375	27.2090
CR	5.3333	21.3333	4.5714	85.3333	73.1429

表一、碼簿做向量量化壓縮後之影像 PSNR、CR 的數值列表

影像 512 512	LBG 編碼簿	細胞分裂法 $\delta = 6$		細胞分裂法 $\delta = 10$	
	PSNR	PSNR	Gain	PSNR	Gain
Girl	31.167	31.872	0.705	31.700	0.533
Gold	29.337	30.203	0.866	30.355	1.018
Lena	29.080	29.927	0.847	29.930	0.850
Pepper	29.942	30.905	0.963	30.869	0.927
Toys	28.618	30.061	1.443	30.104	1.486
Tiffany	28.268	30.669	2.401	30.928	2.268
平均	29.402	30.610	1.204	30.647	1.245

表二、細胞分裂法與 LBG 之比較(網路上所提供的結果比較表格)