Journal of Information Hiding and Multimedia Signal Processing ©2012 ISSN 2073-4212
Ubiquitous International Volume **3**, Number **1**, January **2012**

Fast Image Artistic Style Learning Using Twin-Codebook Vector Quantization

Hua Chen

School Of Aeronautics and Astronautics
Zhejiang University
Hangzhou 310027, P.R. China
chen2013571@126.com

Bei-Bei Liu

Department of Computer Science
Korean Advanced Institute Of Science and Technology
Daejeon 305-701, Korean
yesbbl@126.com

Hao Luo

School Of Aeronautics and Astronautics
Zhejiang University
Hangzhou 310027, P.R. China
luohao@zju.edu.cn

Zhe-Ming Lu*

School Of Aeronautics and Astronautics
Zhejiang University
Hangzhou 310027, P.R. China
zheminglu@zju.edu.cn

Received August 2010; revised November 2011

Abstract. *Recently, the Twin-codebook Vector Quantization (TBVQ) has been developed for image colourization. It adopts two different but related codebooks in the encoder and decoder respectively, instead of the same codebooks used in traditional vector quantizers. In this paper, a novel image artistic style learning scheme is proposed based on TBVQ. During the learning process, the target image is first encoded by the natural codebook and then decoded by the stylized codebook. Effectiveness of the proposed method is shown by experimental results.*

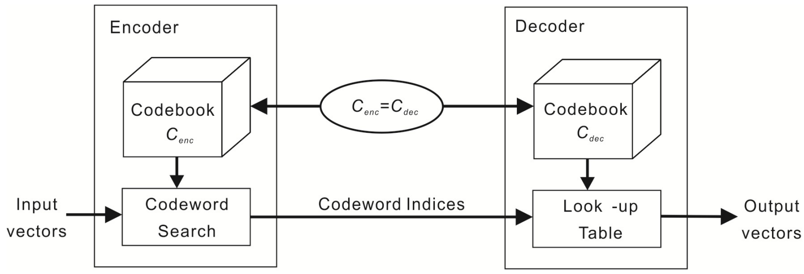
Keywords: Image artistic style learning, Twin-codebook vector quantization.

1. **Introduction.** Image artistic style learning is a technique that learns a filter by examples and applies it to process a target image, so that the output image maintains the artistic style analogous to the target one. Available image artistic style learning methods can be divided into two categories. One is based on brush strokes [1] and the other is based on image analogy [2, 3]. In the brush strokes based scheme, the original image is first converted to black and white and low pass filtered. Then the direction of the gradient of this image is used to control the brush stroke direction while painting. Another interesting technique is to blend the brush strokes onto the canvas. The framework of image analogy includes two stages. In the first stage, a pair of images is presented as

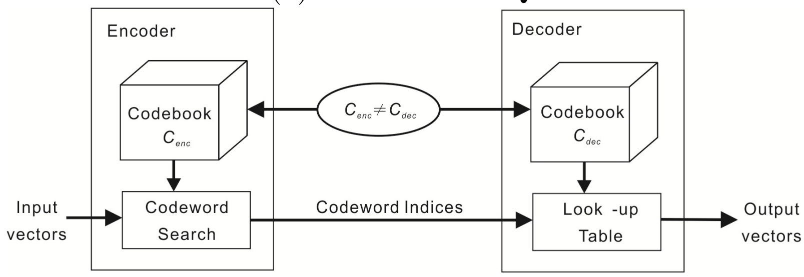
training data, where one image is purported to be a filtered version of the other. In the second stage, the learned filter is applied to some new target image so as to create an analogous filtered result. Image analogies are based on a simple multi-scale autoregression, inspired mainly by recent results in texture synthesis. By choosing different kinds of source image pairs as input, the framework supports a wide variety of image filtering effects. By using this framework, we can perform traditional image filters, such as blurring or embossing. We can perform the improved texture synthesis where some textures are synthesized with higher quality than by previous approaches. We can also implement the super-resolution where a higher-resolution image is inferred from a low-resolution source. By performing the texture transfer, images can be texturized with some arbitrary source texture. By using artistic filters, various drawing and painting styles can be synthesized based on scanned real-world examples. In addition, realistic scenes, composed of a variety of textures, can be created using a simple painting interface. As machine learning is involved, the image analogy based methods are usually time-consuming. Image analogy is a reasoning process according to a pair of template images Tem_1 and Tem_2. It uses training data of Tem_1 and Tem_2 in order to learn a filter that can be applied to a given target image Tar_1 to produce an analogous image Tar_2. This means that Tar_2 relates to Tar1 in the same way Tem_2 relates to Tem_1.

This paper aims to deal with the image artistic style learning problem from a new perspective, i.e., employing a novel twin-codebook vector quantizer. As a lossy data compression method, Vector Quantization (VQ) [4] is based on the principle of block coding. Given a set of input vectors, a distortion measure and the number of codevectors, the traditional vector quantizer design is a task of finding a codebook and a partition which result in the smallest average distortion. VQ can be defined as a mapping from a k-dimensional Euclidean space R_k into a finite subset (codebook) $C = \{c_i | i = 0, 1, , N-1\}$, where c_i is a codeword and N is the codebook size. VQ first divides the input image into vectors, and then each vector is encoded by its best-matched codeword in C. Then the input vector is encoded with the index of its best-matched codeword. Besides data compression, the applications of VQ have been extended to data hiding [5], image retrieval [6] and image colourization [7]. Recently, TBVQ [7] has been developed for image colourization. The comparison of traditional VQ and TBVQ is shown in figure 1. In the encoding process, the input vectors are encoded using the codebook C_{enc} obtained by some given training data in advance. The codeword indices corresponding to their input counterparts are transmitted to the decoder. Then a simple look-up table process based on the codebook C_{dec} is needed to decode the received indices. The key difference between traditional VQ and TBVQ lies in the two codebooks used in the encoder and decoder. In traditional VQ, the codebook for the decoder is the same as the codebook for the encoder, i.e., $C_{enc} = C_{dec}$. However, in TBVQ, the two codebooks used in the encoder and decoder are different, i.e., $C_{enc} \neq C_{dec}$, but their codewords with the same index are closely related. In fact, TBVQ is designed not for data compression but for special mapping or transferring applications. The key problem in TBVQ is to generate the encoding codebook together with the decoding codebook based on an application-oriented design scheme. For example, in the image colorization application, the codebook C_{enc} used in the encoding process should be a grayscale codebook for the input target image to be encoded is a grayscale image, while the codebook C_{dec} used in the decoding process should be a chromatic codebook since the output image is the colorized result.

2. **Proposed Image Artistic Style Learning Method.** Based on the definition of TBVQ, we can easily find that TBVQ is suitable for the image artistic style learning



(a) Traditional VQ



(b) TBVQ

FIGURE 1. Comparison of traditional VQ and TBVQ

problem, i.e., to stylize a target grey-level image I_{tar} according to a pair of template grey-level images composed of a natural image I_n and its stylized version I_s given in advance. As shown in Fig. 2, our scheme consists of three stages, i.e., codebook design, encoding and decoding. First, I_n and I_s are used as the training images to generate two codebooks C_n and C_s for the encoder and decoder respectively. In general, I_n and I_{tar} should be of the same category, e.g., they are both natural building images. The codebook C_n used in the encoding process should be a natural codebook since the input target image to be encoded is also a natural grey-level image, while the codebook C_s used in the decoding process should be a stylized codebook for the output image is the stylized result. Each codeword in C_n is required to have a twin codeword with the same index in C_s. The joint codebook design operations are described below.

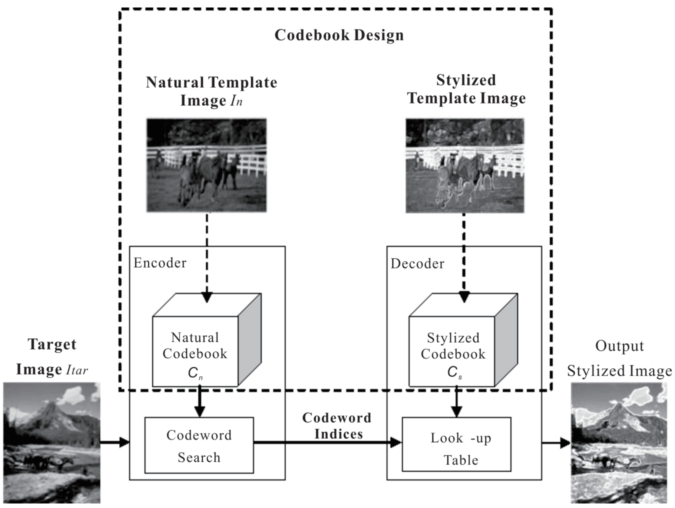


FIGURE 2. Proposed TBVQ-based image artistic style learning scheme

Step 1: The template image I_n is segmented into non-overlapping blocks of size 2×2. Pixel values in each block are transformed into a vector y_j. The same operations are performed on I_s and a vector z_j is obtained for each block. All vectors y_j compose the training set $T_Y = \{y_1, y_2, , y_N\}$ to generate the codebook C_n, and all vectors z_j compose the training set $T_Z = \{z_1, z_2, ..., z_N\}$ for generating the codebook C_s. Here, N is the number of training vectors.

Step 2: Generate the codebook C_n based on the training set T_Y using the LBG algorithm [4].

Step 3: Encode the training set T_Y into M cells $\{R_1^{(Y)}, R_2^{(Y)}, ..., R_M^{(Y)}\}$ based on the obtained codebook $C_n = \{c_n^1, c_n^2, ..., c_n^M\}$, satisfying

$$R_i^{(Y)} = \{y_i | d(y_j, c_n^i) = \min_{1 \le k \le M} d(y_j, c_n^k), 1 \le j \le N\} \quad 1 \le i \le M \tag{1}$$

Here, M is the predefined codebook size.

Step 4: Apply the partition result as given in Eq.(1) directly to the training set T_Z to obtain its M cells $\{R_1^{(Z)}, R_2^{(Z)}, ..., R_M^{(Z)}\}$ as follows

$$R_i^{(Z)} = \{z_j | y_j \in R_i^{(Y)}, 1 \le j \le N\} \tag{2}$$

Step 5: Calculate the centroid of each cell $R_i^{(Z)}$ to obtain the corresponding codeword c_s^i. All the codewords c_s^i compose the codebook $C_s = \{c_s^1, c_s^2, ..., c_s^M\}$.

After above steps, two different but related codebooks C_n and C_s are generated. Each codeword in the encoding codebook C_n has its twin codeword in the decoding codebook C_s with the same index. Based on these two codebooks, the artistic stylization is realized as follows.

Step 1: First, the input target image I_{tar} is partitioned into non-overlapping blocks of size 2×2. Each block y_k is encoded by the codebook C_n to get the index ID_k.

Step 2: Second, according to ID_k we can find its corresponding codeword in the codebook C_s to get the decoded vector.

Step 3: Third, the output stylized block is easily constructed based on the codeword.

Step 4: At last all the output blocks are recomposed and the final stylized image I_{out} is obtained.

3. **Experimental Results.** The test computer is with a 512M RAM, a 2.8G CPU and the platform is MATLAB. In all experiments, sizes of the target and template images are 406×512 and 366×544 respectively, and the block size is 2×2. The LBG codebook design algorithm [4] is used to generate the codebooks C_n and C_s. Fig. 3 shows the stylization results for two test images. The first column shows the target images and the others show the output stylized images with each column having a certain style. We adopt the fast codeword search scheme [8] during the encoding stage to speed up the stylization. In these experiments, the total time required for codebook training and stylization ranges from 11 to 14 seconds. In fact, the trained codebooks also can be used to stylize multiple images of the same style. Compared with the image analogy based and brush strokes

based methods, our method can obtain better visual effects in a shorter time, while image analogy and brush strokes based methods are time-consuming.

4. **Conclusions.** This paper presents an image artistic style learning method based on TBVQ. The proposed method is fast and effective. Future work will concentrate on the improvement of TBVQ to achieve better visual effects and extend the application for color image artistic style learning.

FIGURE 3. Various artistic stylization results for two target images

REFERENCES

[1] P. Haeberli, Paint by numbers: abstract image representations, *Proc. of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, vol. 24, pp. 207-214, 1990.

[2] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, D. Salesin, Image analogies, *Proc. of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 327-340, 2001.

[3] J. Huttunen, Image analogies, *T-111.500 Seminar on computer graphics*, Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology, 2004.

[4] Y. Linde, A. Buzo, R. M. Gray, An algorithm for vector quantizer design, *IEEE Trans. Communications*, vol. 28, no. 1, pp. 84-95, 1980.

[5] Z. M. Lu, J. X. Wang, B. B. Liu, An improved lossless data hiding scheme based on image VQ-index residual value coding, *Journal of Systems and Software*, vol. 82, no. 6, pp. 1016-1024, 2009.

[6] Z. M. Lu, H. Burkhardt, Colour image retrieval based on DCT-domain vector quantization index histograms, *Electronics Letters*, vol. 41, no. 17, pp. 956-957, 2005.

[7] F. X. Yu, B. B. Liu, Z. M. Lu, Fast image colourization based on twin-codebook vector quantization, *Electronics Letters*, vol. 46, no. 2, pp. 132-134, 2010.

[8] Z. M. Lu, J. S. Pan, S. H. Sun, Efficient codeword search algorithm based on Hadamard transform, *Electronics Letters*, vol. 36, no. 16, pp. 1364-1365, 2000.