Color Image Segmentation Using Kekre's Algorithm for Vector Quantization

H. B. Kekre, Tanuja K. Sarode, Bhakti Raul

***Abstract*—**In this paper we propose segmentation approach based on Vector Quantization technique. Here we have used Kekre's fast codebook generation algorithm for segmenting low-altitude aerial image. This is used as a preprocessing step to form segmented homogeneous regions. Further to merge adjacent regions color similarity and volume difference criteria is used. Experiments performed with real aerial images of varied nature demonstrate that this approach does not result in over segmentation or under segmentation. The vector quantization seems to give far better results as compared to conventional on-the-fly watershed algorithm.

***Keywords*—**Image Segmentation,, Codebook, Codevector, data compression, Encoding

I. Introduction

IN this paper we deal with the specific problem of segmenting architectural elements such as roofs, walls and pavement in low-altitude aerial images, so that these segmented elements can later be used as the basis to build 3D reconstruction algorithms specifically tailored to recover the geometry of entire metropolitan areas in a fully automatic way[1].

Image segmentation is, by definition, the problem of decomposing images into regions that are semantically uniform. However, since images themselves provide only semantically poor information, image segmentation is essentially an application-oriented problem that demands either strong intervention of human experts or application specific solutions.

In other words, no fully automatic, general-purpose segmentation method exists. The choice of a particular technique depends on the nature of the task to be performed (3D reconstruction, shape recognition, quality control, medical image analysis), on the nature of the images available (presence or not of non-homogeneous illumination, texture, ill-defined contours, occlusions, shadows), on the primitives to be extracted (contours, straight segments, regions, shapes, textures) and on physical limitations (real-time constraints, limitations on computational power and storage capacity) [1].

One of the most traditional forms of segmentation is region growing and merging [4, 5]. These techniques work well in noisy images, but they are sensitive to seed initialization, which is hard to perform in a fully-automatic way, and are prone either to produce jagged boundaries, or to leak through narrow gaps or weak edges, generating under segmentation.

Segmentation by deformable models overcomes some of these shortcomings by describing region boundaries as continuous, piecewise-smooth curves that evolve under a suitable energy functional until they (hopefully) converge to the semantically meaningful borders. Unfortunately, the most traditional techniques of this type, such as the snakes method [6], require very good prior knowledge about the topology and even the approximate position of the actual object boundaries in order to guarantee proper convergence and to avoid being trapped by local minima. Some work [7,8] has been done towards ameliorating this difficulty, for instance by adding an inflation force to the snakes, which helps them escape from local minima. However, this inflation force often pushes the contour over weak edges, which leads back to under-segmentation.

Theoretically, the need for prior knowledge about topology can be avoided by modeling region boundaries within the level-set framework [9, 10], which allows merging of non-significant curves or even the splitting of undersegmented regions. In practice, existing level-set methods [11],[12],[13] require initialization steps that are difficult and time-consuming, such as the manual introduction of polygons around the features of interest. Convergence is also difficult since some curves are still evolving while others have finished their evolution or, worse, have leaked through weak boundaries. Thus, to some extent or another, all techniques mentioned so far have initialization and convergence problems. In this paper, to avoid this kind of difficulty, we use vector quantization technique. This segmentation technique basically has two steps to follow:

Step1: Apply Vector Quantization technique to form regions

Step2: Merge regions

VQ can be defined as a mapping function that maps k-dimensional vector space to a finite set CB = $\{C_1, C_2, C_3, \ldots\ldots, C_N\}$. The set CB is called codebook consisting of N number of codevectors and each codevector $C_i = \{c_{i1}, c_{i2}, c_{i3}, \ldots\ldots, c_{ik}\}$ is of dimension k. The key to VQ is the good codebook. Codebook can be generated in spatial domain by

Dr. H. B. Kekre is Senior Professor working with Mukesh Patel School of Technology, Management and Engineering, NMIMS University, Vile-Parle (West), Mumbai-56. (E-mail: hbkekre@yahoo.com)

Ms. Tanuja K. Sarode, is Ph.D. Scholar from Mukesh Patel School of Technology, Management and Engineering, NMIMS University, Vile-Parle (West), Mumbai-56. Assistant Professor working with Thadomal Shahani Engineering College, Bandra (West), Mumbai-50. (E-mail: tanuja_0123@yahoo.com).

Ms. Bhakti Raul is Lecturer working Engineering at K.J. Somaiya College of engineering, Mumbai (E-mail: bhakti_raul@yahoo.com).

clustering algorithms [14], [16]-[18], [48]-[51] or using transform domain techniques [20]-[22].

In Encoding phase image is divided into non overlapping blocks and each block then converted to the training vector $X_i = (x_{i1}, x_{i2}, \ldots\ldots, x_{ik})$. The codebook is then searched for the nearest codevector C_{min} by computing squared Euclidean distance as presented in equation (1) with vector X_i with all the codevectors of the codebook CB. This method is called exhaustive search (ES).

$$d(X_i, C_{\min}) = \min_{1\le j\le N}\{d(X_i, C_j)\}$$

$$\text{Where } d(X_i, C_j) = \sum_{p=1}^{k}(x_{ip}, c_{jp})^2 \qquad (1)$$

Although the Exhaustive Search (ES) method gives the optimal result at the end, it involves heavy computational complexity. If we observe the above equation (1) to obtain one nearest codevector for a training vector requires N Euclidean distance computation where N is the size of the codebook. So for M image training vectors, will require M*N number of Euclidean distances computations. It is obvious that if the codebook size is increased to reduce the distortion the searching time will also increase.

In order to reduce the searching time there are various search algorithms available in literature. So far, Partial Distortion search (PDS) [19], equal-average nearest neighbor search (ENNS) [23], the equal average equal variance nearest neighbor search (EENNS) [24], nearest neighbor search algorithm based on orthonormal transform (OTNNS) [25]. Partial Distortion Elimination (PDE) [38], triangular inequality elimination (TIE) [39-41], mean distance ordered partial codebook search (MPS) algorithm [42] ,double test algorithm (DTA) [35], fast codebook search algorithm based on the Cauchy-Schwarz inequality (CSI) [43], fast codebook search based on subvector technique (SVT) [44], the image encoding based on L_2-norm pyramid of codewords [45] and the fast algorithms using the modified L_2-norm pyramid (MLP) [46], fast codeword search algorithm based on MPS+TIE+PDE proposed by Yu-Chen, Bing-Hwang and Chih-Chiang (YBC) in 2008 [47], Kekre's fast search algorithms [52], [53], Eigen vector method (EVM) [34], and others [28], [32], [33], [35] are classified as partial search methods. Some of the partial techniques use data structure to organize the codebook for example tree-based [26], [27], [30], [31], [36] and projection based structure [26], [29], [37]. All these algorithms reduce the computational cost needed for VQ encoding keeping the image quality equivalent to Exhaustive search algorithm

In section II, we first describe the on-the-fly approach based on watershed segmentation. Proposed methodology is explained in Section III. Evaluation function is given in section IV. The segmentation results and discussion are provided in Section V and we conclude the paper in Section VI.

II. On-the-fly approach [1]

Catchment basin merging algorithm (CBMA) [3] embeds an on-the-fly merging mechanism into Vincent and Soille watershed segmentation (VSWT) [2], in order to reduce over-segmentation. The merging process is guided by a set of rules that take into account geometric attributes of the catchment basins such as depth, area and volume. A typical CBMA, would create new segmentation edges only between catchment basins whose volumes (or depths, or areas) in the previous iteration are larger than pre-defined thresholds Units

III. Segmentation using Fast codebook generation algorithm

A. Kekre's Fast codebook generation algorithm (KFCG) [14], [49]

In reference [49] we have proposed this algorithm for image data compression. This algorithm reduces the time of code book generation. Initially we have one cluster with the entire training vectors and the codevector C_1 which is centroid. In the first iteration of the algorithm, the clusters are formed by comparing first member of training vector with first member of code vector C_1. The vector X_i is grouped into the cluster 1 if $x_{i1} < c_{11}$ otherwise vector X_i is grouped into cluster2. In second iteration, the cluster 1 is split into two by comparing second member x_{i2} of vector X_i belonging to cluster 1 with that of the member c_{12} of the codevector C_1. Cluster 2 is split into two by comparing the member x_{i2} of vector X_i belonging to cluster 2 with that of the member c_{22} of the codevector C_2.

This procedure is repeated till the codebook size is increased to the size specified by user. It is observed that this algorithm gives minimum error and requires least time to generate codebook as compared to other algorithms [14], [49], [51].

B. Proposed technique (KFCG+RM)

Given image is divided into regions using vector quantization techniques and then the regions are merged based on color threshold and volume threshold values. The proposed technique has two steps to follow:

1. Region forming using vector quantization technique
2. Region merging

Region forming:

To form regions Kekre's Fast codebook generation algorithm (KFCG) is used. In each vector quantization technique 2 types of training vectors are formed.

- Each training vector is of dimension three consisting of R, G, B components of one pixel.
- Each training vector is of dimension twelve consisting of R, G, and B components of 2 ×2 adjacent pixels.

The size of codebook is set to eight. Training vectors are reassigned to encoding regions in every iteration. Once the code book size reaches eight the process is stopped. In the original image pixel value is replaced by the encoding region number to which the pixel is assigned.

Region merging:

Region merging based on color similarity is performed after region formation as a posteriori step. All pixels pertaining to each segmented region have exactly the same label. Thus, a single scan through the labeled image suffices to compute the mean color and volume of each region. The labeled image is then scanned successively to combine two adjacent regions whose mean colors differ by less than a preset threshold and to merge a small region whose volume is less than a preset threshold into larger region.

IV. Evaluation function Q(I) [15]

The evaluation function is defined as

$$Q(I) = \frac{1}{10000(N \times M)} \sqrt{R} \times \sum_{i=1}^{R} \left[\frac{ei^2}{1+\log Ai} + \left(\frac{R(Ai)}{Ai} \right)^2 \right] \quad (2)$$

where I is the segmented image, N×M the image size, and R the number of regions of the segmented image, while Ai and ei are, respectively, the area and the average color error of the i^{th} region; ei is defined as the sum of the Euclidean distances between the RGB color vectors of the pixels of region i and the color vector attributed to region i in the segmented image. While R(Ai), represents the number of regions having an area equal to Ai .The smaller the value of Q(I), the better the segmentation result should be. The body of the sum is composed of two terms: the first is high only for non-homogeneous regions (typically, large ones), while the second term is high only for regions whose area A is equal to the area of many other regions in the segmented image (typically, small ones). We may expect that the number of regions of area Ai in given an image will be small if area Ai has a high value; and in this case $R(Ai)^2/Ai$ contributes little to the sum. On the other hand, the number of regions of area Ai may be large if the area Ai has a low value; in this case $R(Ai)^2/Ai$ contributes strongly to the sum. Heuristically we can say that R(Ai) is almost always 1 for large regions, and can be much larger than 1 for small regions. In any case, the denominator Ai drastically forces the term R(Ai) /Ai to near zero for large regions, and lets it grow for small regions.

V. Results

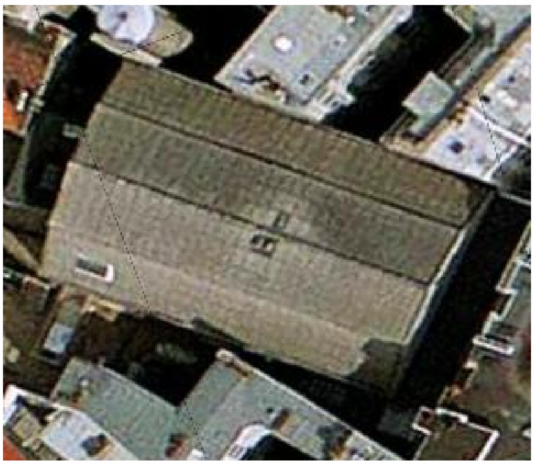
Here we have used image segmentation as an application of vector quantization. The algorithms are implemented on Celeron processor 1.73 GHz, 1MB cache, 1GB RAM machine to obtain result. We have tested these algorithms on 9 images of different sizes. We compare two approaches – on-the-fly and KFCG+RM explained in sections II and III respectively using the evaluation function described in section I.

Table 1 shows the values of Evaluation function 'Q' given in equation (2) obtained after the algorithms are applied on various scenes. The value shown in bold is the least value among all the values of evaluation function 'Q' obtained using RGB 1 pixel and RGB 2×2 block. The least value presents the best segmentation result.

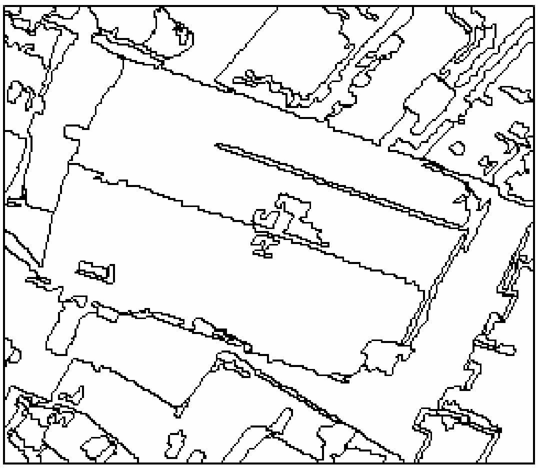
Figure 1 and Figure 2 show the result of applying these algorithms on scene 1 and scene 2 respectively.

TABLE I Values of Evaluation Function 'Q' for various scenes

Scene No.	Algorithms		
	KFCG + RM Vector Dimensions		On The Fly
	RGB 1 Pixel	RGB 2x2 block	
1	**521.2367**	585.6722	1628.30
2	**384.5241**	576.7616	1204.10
3	2647.20	**1553.80**	2756.5
4	654.0498	**570.3052**	1632.5
5	**299.8264**	418.4331	905.602
6	**578.93**	699.13	1818
7	**1544.70**	2079.80	2423.7
8	**325.4873**	570.5526	2112.7
9	**181.6723**	199.1552	865.0008
10	**1402**	6065.4	5057

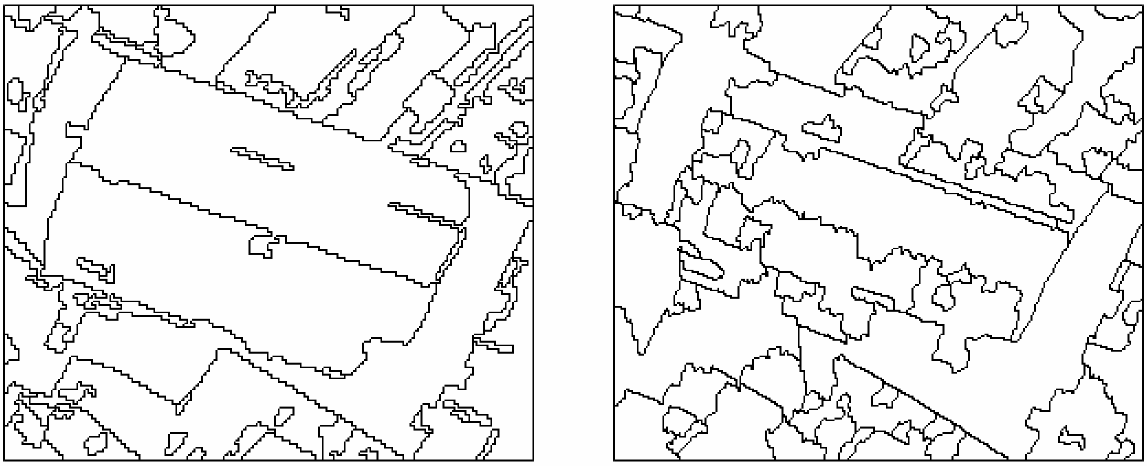


(a) Input Image: Suburban scene 1



(b) KFCG+RM- Training vector size=3

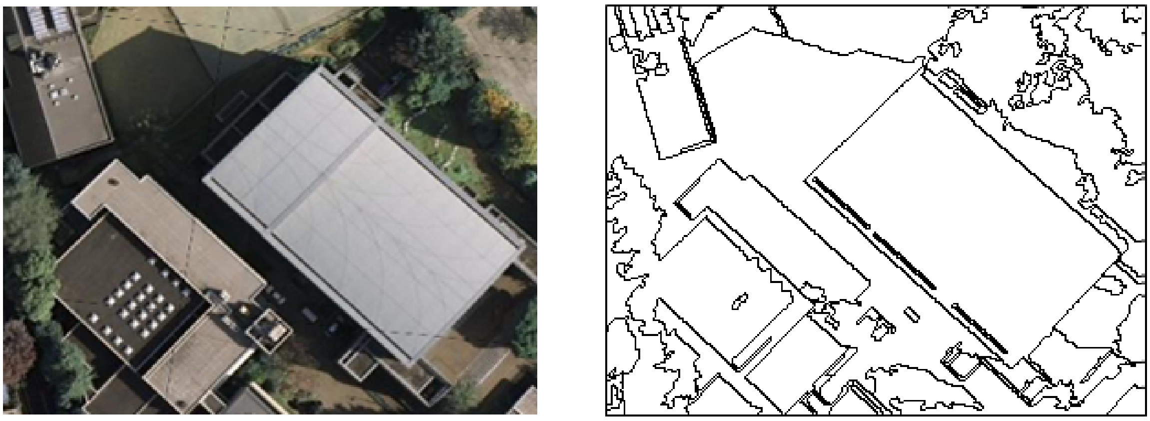
International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:2, No:9, 2008



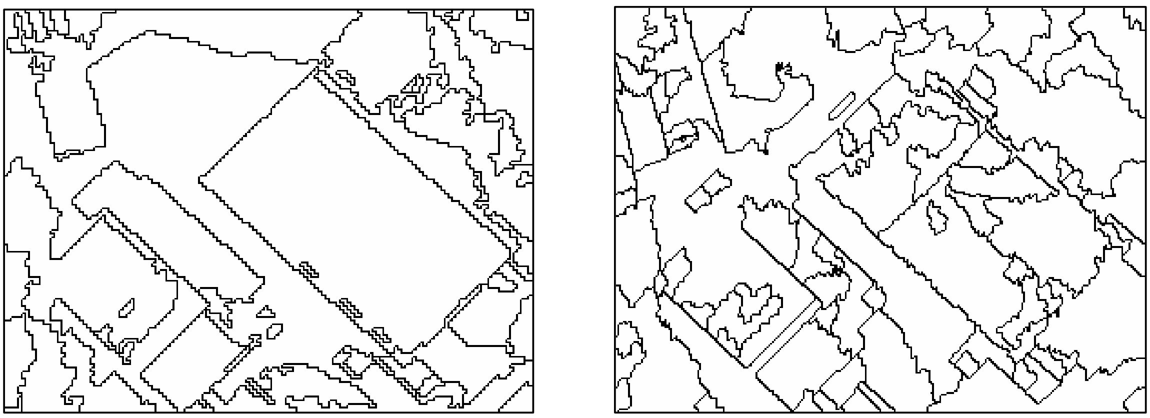
(c) KFCG+RM- Training vector size=12 (d) On-The-Fly

Fig. 1 Segmentation of suburban scene 1

Note: Figure 1(a) shows the input image, Figure 1(b) shows the best segmentation result having least value for quality factor, Figure 1(c) has zigzag effect over the contours and Figure 1(d) shows the over segmented image generated using on-the-fly algorithm.



(a) Input Image: Suburban scene 2 (b) KFCG+RM- Training vector size=3



(c) KFCG+RM- Training vector size=12 (d) On-the-fly

Fig. 2 Segmentation of suburban scene 2

Note: Figure 2(a) shows the input image, Figure 2(b) shows the best segmentation result having least value for quality factor, Figure 2(c) has zigzag effect over the contours and Figure 2(d) shows the over segmented image generated using on-the-fly algorithm.

VI. Conclusion

In this paper we have demonstrated that Kekre's fast code book generation algorithm can be used as the basis to construct fully automatic, reliable technique for segmenting architectural elements in low-altitude aerial images of urban scenes. The evaluation function 'Q' correctly retrieves the best segmentation result. It has been found that when the training vector size is 12 i.e. when a block of 4 pixels is taken into a training vector, a zigzag effect appears over the contours. When the training vector size is 3, we don't see the zigzag effect over the contours. Performance of this algorithm is far better than on-the-fly watershed algorithm which generates over segmented image

References

[1] Ariano B. Huguet1, Marcos C. de Andrade2, Rodrigo L. Carceroni1, Arnaldo de A. Ara ujo1, Color-Based Watershed Segmentation of Low-Altitude Aerial Images", *Proceedings of the XVII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'04),* pp. 138-145, 17-20 Oct 2004.

[2] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal.Mach. Intell.*, vol. 13, no. 6, pp. 583–598, Jun. 1991.

[3] M. Andrade, G. Bertrand, and A. Araujo, "Segmentation of microscopic images by flooding simulation: A catchment basins merging algorithm." *In Proc. SPIE Nonlinear Image Processing VIII,* vol. 3026, pp. 164-175, Feb 1997.

[4] R. Adams and L. Bischof, " Seeded region growing", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15 , no. 6, pp. 641–647, June 1994.

[5] A. Mehnert and P. Jackway, "An improved seeded region growing algorithm", *Pattern Recognition Letters*, vol. 18, no. 10, pp. 1065–1071, Oct. 1997.

[6] M. Kass, A.Witkin, and D. Terzopoulos, "Snakes - active contour models", *Int. J. Computer Vision*, vol. 1, no. 4, pp. 321–331, Jan. 1998.

[7] L. Cohen and I. Cohen , "Finite element methods for active contour models and balloons for 2d and 3d images", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no.11, pp. 1131–1147, Nov. 1993.

[8] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow", *IEEE Trans Image Processing*, vol. 7, no. 3, pp. 359–369, Mar. 1988.

[9] J. A. Sethian, "Tracking interfaces with level sets", *American Scientist*, pp. 254–263, May 1997.

[10] J. A. Sethian, "*Level Set Methods and Fast Marching Methods",* Cambridge U. Press, second edition, 1999.

[11] F. Caselles, "Image selective smoothing and edge detection by nonlinear diffusion", *SIAM J. Numerical Analysis*, vol. 29, no. 1, pp. 182–193, Feb 1992.

[12] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–175, Feb 1995.

[13] R. Malladi, J. A. Sethian, and B. C. Vemuri, "A fast level set based algorithm for topology-independent shape modeling", *J. Math. Imaging and Vision*, vol. 6, no.2-3, pp. 269–290, Jun. 1996.

[14] Dr. H. B. Kekre, Ms. Tanuja K. Sarode, "New Fast Improved Clustering Algorithm for Codebook Generation for Vector Quantization", *International Conference on Engineering Technologies and Applications in Engineering, Technology and Sciences, Computer Science Department, Saurashtra University, Rajkot, Gujarat. (India), Amoghsiddhi Education Society, Sangli, Maharashtra (India), 13th – 14th January 2008*

[15] M. Borsotti , P. Campadelli , R. Schettini, "Quantitative evaluation of color image segmentation results", *Pattern Recognition Letters,* vol. 19, no. 8, pp. 741–747, Jun. 1998..

[16] R. M. Gray.: 'Vector quantization', *IEEE ASSP Mag*, Apr. 1984, pp. 4-29.

[17] Y. Linde, A. Buzo, and R. M. Gray.: 'An algorithm for vector quantizer design," *IEEE Trans. Commun.*', vol. COM-28, No. 1, 1980, pp. 84-95.

[18] A. Gersho, R.M. Gray.: '*Vector Quantization and Signal Compressio'*, Kluwer Academic Publishers, Boston, MA, 1991.

[19] C. D. Bei and R. M. Gray.: 'An improvement of the minimum distortion encoding algorithm for vector quantization', *IEEE Trans. Commun.*,vol. 33, No. 10, pp. 1132–1133, Oct. 1985.

[20] Momotaz Begum, Nurun Nahar, Kaneez Fatimah, M. K. Hasan, and M. A. Rahaman: 'An Efficient Algorithm for Codebook Design in Transform Vector Quantization', *WSCG'2003*, February 3-7, 2003.

[21] Robert Li and Jung Kim: 'Image Compression Using Fast Transformed Vector Quantization', *IEEE Applied Imagery Pattern Recognition Workshop, 2000 Proceedings* 29th Volume , 2000, pp.141 – 145.

[22] Zhibin Pan; Kotani, K.; Ohmi, T., 'Enhanced fast encoding method for vector quantization by finding an optimally-ordered Walsh transform kernel', *ICIP 2005, IEEE International Conference*, Volume 1, pp I - 573-6, Sept. 2005.

[23] Guan, L., and Kamel, M. : 'Equal-average hyperplane partitioning method for vector quantization of image data', Patt. Recognit. Lett., 1992, pp. 693-699.

[24] Lee, H., and Chen, L. H. : 'Fast closest codevector search algorithms for vector quantization', Signal Process., vol. 43, 1995, pp. 323-331.

[25] Z. Li, and Z.- M. Lu. : 'Fast codevector search scheme for 3D mesh model vector quantization', Electron. Lett., vol. 44, 2008, pp. 104-105.

[26] Chin-Chen Chang, Wen-Chuan Wu, " Fast Planar-Oriented Ripple Search Algorithm for Hyperspace VQ Codebook*", IEEE Transaction on image processing,* vol 16, No. 6, June 2007.

[27] C. C. Chang and T. S. Chen, "New tree-structured vector quantization with closest-coupled multipath searching method," *Opt. Eng.*, vol. 36, No. 6, pp. 1713–1720, Jun. 1997.

[28] C. C. Chang and I. C. Lin, "Fast search algorithm for vector quantization without extra look-up table using declustered subcodebooks," *IEE Proc. Vis., Image, Signal Process.*, vol. 152, No. 5, pp. 513–519, Oct.2005.

[29] C. C. Chang, D. C. Lin, and T. S. Chen, "An improved VQ codebook search algorithm using principal component analysis," *J. Vis. Commun. Image Represent.*, vol. 8, No. 1, pp. 27–37, Mar. 1997.

[30] C. C. Chang, F. J. Shiue, and T. S. Chen, "Tree structured vector quantization with dynamic path search," in *Proc. Int. Workshop on Multimedia Network Systems*, Aizu, Japan, pp. 536 –541, Sep. 1999.

[31] R. M. Gray and Y. Linde, "Vector quantization and predictive quantizers for gauss-markov sources," *IEEE Trans. Commun.*, vol. 30, No. 2, pp. 381–389, Feb. 1982.

[32] 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.

[33] Y. C. Hu and C. C. Chang, "An effective codebook search algorithm for vector quantization", *Imag. Sci. J.*, vol. 51, No. 4, pp. 221–234, Dec. 2003.

[34] C. H. Lee and L. H. Chen, "High-speed closest codeword search algorithm for vector quantization," *Signal Process.*, vol. 43, No. 3, pp.323–331, May 1995.

[35] L. Torres and J. Huguet, "An improvement on codebook search for vector quantisation", *IEEE Trans. Commun.*, vol. 42, No. 2, pp. 208–210, Feb. 1994.

[36] S. J. Wang and C. H. Yang, "Hierarchy-oriented searching algorithms using alternative duplicate codewords for vector quantization mechanism," *Appl. Math. Comput.*, vol. 162, No. 234, pp. 559–576, Mar. 2005.

[37] S. C. Tai, C. C. Lai, and Y. C. Lin, "Two fast nearest neighbor searching algorithms for image vector quantization," *IEEE Trans. Commun.*, vol. 44, No. 12, pp. 1623–1628, Dec. 1996.

[38] C. Bei, R.M. Gray, ''An improvement of the minimum distortion encoding algorithm for vector quantization'', IEEE Trans. Commun.33, 1985, pp. 1132–1133.
[39] S.H. Huang, S.H. Chen, ''Fast encoding algorithm for VQ-based image coding'', Electron. Lett. Vol. 26, No. 19, 1990, pp. 1618–1619.
[40] W. Li, E. Salari, ''A fast vector quantization encoding method for image compression'', IEEE Trans. Circ. Syst. Vid. Vol 5, 1995, pp. 119–123.
[41] C.H. Hsieh, Y.J. Liu, ''Fast search algorithms for vector quantization of images using multiple triangle inequalities and wavelet transform'', IEEE Trans. Image Process. Vol. 9, No. 3, 2000, pp. 321–328.
[42] S.W. Ra, J.K. Kim, ''A fast mean-distance-ordered partial codebook search algorithm for image vector quantization'', IEEE Trans. Circuits-II, vol. 40, No. 9, 1993, pp. 576–579.
[43] K.S. Wu, J.C. Lin, ''Fast VQ encoding by an efficient kick-out condition'', IEEE Trans. Circ. Syst. Vid., vol.10, No. 1, 2000, pp. 59–62.
[44] J.S. Pan, Z.M. Lu, S.H. Sun, ''An efficient encoding algorithm for vector quantization based on subvector technique'', IEEE Trans. Image Process. Vol 12, No.3, 2003, pp. 265–270.
[45] B.C. Song, J.B. Ra, ''A fast algorithm for vector quantization using L2-norm pyramid of codewords'', IEEE Trans. Image Process. Vol. 4, No.12, 2002, pp. 325–327.
[46] Z. Pan, K. Kotani, T. Ohmi, ''Fast encoding method for vector quantization using modified L2-norm pyramid'', IEEE Signal Process. Lett. Vol. 12, issue 9, 2005, pp. 609–612.
[47] Y. Chen, B. Hwang, C. Chiang, "Fast VQ codebook search algorithm for grayscale image coding", Image and Vision Compu., vol. 26, 2008, pp. 657-666.
[48] H. B. Kekre, Tanuja K. Sarode, "New Fast Improved Codebook Generation Algorithm for Color Images using Vector Quantization," International Journal of Engineering and Technology, vol.1, No.1, pp. 67-77, September 2008
[49] H. B. Kekre, Tanuja K. Sarode, "Fast Codebook Generation Algorithm for Color Images using Vector Quantization," International Journal of Computer Science and Information Technology, Vol. 1, No. 1, pp: 7-12, Jan 2009.
[50] H. B. Kekre, Tanuja K. Sarode, "An Efficient Fast Algorithm to Generate Codebook for Vector Quantization," First International Conference on Emerging Trends in Engineering and Technology, ICETET-2008, held at Raisoni College of Engineering, Nagpur, India, 16-18 July 2008, Avaliable at online IEEE Xplore.
[51] H. B. Kekre, Tanuja K. Sarode, "Speech Data Compression using Vector Quantization", WASET International Journal of Computer and Information Science and Engineering 2;4 © www.waset.org Fall 2008 (IJECSE), Volume 2, Number 4, 251-254, 2008. available: http://www.waset.org/ijcise
[52] H. B. Kekre, Tanuja K. Sarode, ""Centroid Based Fast Search Algorithm for Vector Quantization", International Journal of Imaging (IJI), Volume 1, Number A08, pp. 73-83, Autumn 2008, available: http://www.ceser.res.in/iji.html
[53] H. B. Kekre, Tanuja K. Sarode, ""Fast Codevector Search Algorithm for 3-D Vector Quantized Codebook", WASET International Journal of Electrical Computer and Systems Engineering (IJCISE), Volume 2, Number 4, pp. 235-239, Fall 2008. available: http://www.waset.org/ijcise

BIOGRAPHIES

Dr. H. B. Kekre has received B.E. (Hons.) in Telecomm. Engineering. from Jabalpur University in 1958, M.Tech (Industrial Electronics) from IIT Bombay in 1960, M.S.Engg. (Electrical Engg.) from University of Ottawa in 1965 and Ph.D. (System Identification) from IIT Bombay in 1970. He has worked Over 35 years as Faculty of Electrical Engg. and then HOD Computer Science and Engg. at IIT Bombay. For last 13 years worked as a Professor in Department of Computer Engg. at TSEC, Mumbai. He is currently Senior Professor working with Mukesh Patel School of Technology, Management and Engineering, NMIMS University, Vile-Parle (W), Mumbai. His areas of interest are Digital Signal processing and Image Processing and Computer Networks. He has more than 200 papers in National / International Conferences / Journals to his credit. Recently five students working under his guidance have received best paper awards.

Ms. Tanuja K. Sarode has Received M.E.(Computer Engineering) degree from Mumbai University in 2004, currently Perusing Ph.D. from Mukesh Patel School of Technology, Management and Engineering, NMIMS University, Vile-Parle (W), Mumbai. She has more than 9 years of experience in teaching. Currently working as a Assistant Professor in Department of Computer Engineering at Thadomal Shahani Engineering College, Mumbai. Her areas of interest are Image Processing, Signal Processing and Computer Graphics. She has 22 papers in National /International Conferences/journal to her credit.

Ms. Bhakti C. Raul has received B.E (Computer science and engineering) in 2001, currently persuing M.E from Mumbai University. She has more than 6 years of experience in teaching. Currently working as a lecturer in Department of Computer Engineering at K.J. Somaiya College of engineering, Mumbai. Her areas of interest are Image processing, Database management systems, Compiler construction.

.