Efficiency of Gaussian Pyramid Compression Technique for Biometric Images

K. C. Chandra sekaran [1], Dr. K. Kuppusamy [2]
[1] Department of Computer Science, Alagappa Govt. Arts College, Karaikudi, Tamilnadu 630 303, India

[2] Department of Computer Science and Engineering, Alagappa University, Karaikudi, , Tamilnadu 630 303, India

Abstract
Images in their uncompressed form require large amount of storage capacity and uncompresses data needs large transmission bandwidth for the transmission. Image compression technique is used to reduce the storage requirement per image, while maintaining image quality. The proposed method investigates biometric image compression using Gaussian Pyramid Compression (GPC) in which the performance is observed down to 0.23 bits/pixel attributed to noise reduction without a significant loss of texture. Objective measures were used such as Signal to Noise Ratio (SNR), Peak Signal to Noise Ratio (PSNR), Mean Squared Error (MSE) and Bit Rate (BR) to evaluate the image quality for both gray and RGB biometric images. The comparisons of Gaussian Pyramid compression technique applied to .jpg, .bmp and .png images.

Keywords: *Biometric, iris, loss-less, fingerprint, palm print, generating kernel, low-pass filter.*

1. Introduction

Nowadays, there are various algorithms used to reduce the large storage capacity for images. The use of images plays a vital role in many fields such as identification of persons via face recognition, password authentication and so on. Recently, fingerprints and iris recognition systems are used issuing AADHAR card to each citizen in India. Therefore, this research work analyses about the reduced form of images using biometric images.

A digital image is basically a 2-dimensional array of pixels. Images form the significant part of data, particularly in remote sensing, biomedical and video conferencing applications. A biometric verification system is designed to verify or recognize the identity of a living person on the basis of his/her physiological characters, such as face, fingerprint and iris or some other aspects of behavior such as handwriting or keystroke pattern. The biometric verification technique acts as an efficient method and has wide applications in the areas of information retrieval, automatic banking and control of access to security areas, buildings and so on. Compared to other biometric verification techniques, iris recognition plays a vital role, as it cannot be altered by the user. When lot of image has to be stored memory required is more. Hence image compression is used to reduce the amount of memory to store an image without much affecting the quality of an image.

2. Related Work

With the rapid growth of the internet, electronic commerce revenue now spent several billion US dollars to avoid fraud and misuse, buyers and sellers desire more secure methods of authentication than today's userid and password combinations. Automated biometrics technology in general, and fingerprints in particular, provide an accurate and reliable authentication methods[10].However, fingerprint-based authentication requires accessing fingerprint images and palm images scanned remotely at the user's workstation, a potentially weak point in the security system. Stored or synthetic fingerprint images and palm images might be fraudulently transmitted, even if the communication channel itself is encrypted [9].

The investigation methods said that there are three schemes for severe compression of iris images in order to assess what their impact would be on recognition performance of the algorithms deployed today for identifying people by this biometric feature. Currently, standard iris images are 600 times larger than the IrisCode templates computed from them for database storage and search; but it is administratively desired that iris data should be stored, transmitted, and embedded in media in the form of images rather than as templates computed with proprietary algorithms. To compromise that goal with its implications for bandwidth and storage, We present schemes that combine region-of-interest isolation with JPEG and JPEG2000 compression at severe levels, and we test them using a publicly available database of iris images. It shows that it is possible to compress iris images to as little as 2000 bytes with minimal impact on recognition performance. Only some 2% to 3% of the bits in the IrisCode templates are changed by such severe image compression, and we calculate the entropy per code bit introduced by each compression scheme. Error tradeoff curve metrics documents very good recognition performance despite this reduction in data size by a net factor of 150, approaching a convergence of image data size and template size.

In the case of forensic biometric systems, compression of fingerprint images has already been applied in automatic fingerprint identification systems applications, where the size of the digital fingerprint archives would be tremendous for uncompressed images. Iris recognition is a proven, accurate means to identify people. Commercial iris recognition systems are currently employed to allow passengers in some airports to be rapidly processed through security, to allow access to secure areas, and for secure access to computer networks. With the growing employment of iris recognition systems and associated research to support this, the need for large databases of iris images is growing. If required storage space is not adequate for these images, compression is an alternative. It allows a reduction in the space needed to store these iris images, although it may be at a cost in some amount of information lost in the process.

Many image compression methods are compared on public palm print image databases. Effect of lossy compression algorithms on biometric samples has been well studied [10]. However, lossless compression algorithms on the compression ratios have little been appreciated. The performance shows that a guide is given to choose which lossless palm print image compression algorithm. At last, to find better solutions on how to improve image compression performance, this gives some examples and suggestions.

3. Proposed Work

Biometrics-based authentication has many usability advantages over traditional systems such as passwords. Specifically, users can never lose their biometrics and the biometric signal is difficult to steal or create. We have shown that the intrinsic bit strength of a biometric signal can be quite good, especially for fingerprints, when compared to conventional passwords. Yet, any system, including a biometric system, is vulnerable when attacked by determined hackers. We have highlighted eight points of vulnerability in a generic biometric system and have discussed possible attacks. We suggested several ways to alleviate some of these security threats. Replay attacks have been addressed using data-hiding techniques to secretly embed a telltale mark directly in the compressed finger print image. A challenge/response method has been proposed to check the liveliness of the signal acquired from an intelligent sensor. Finally, we have touched on the often neglected problems of privacy and revocation of biometrics. It is somewhat ironic that the greatest strength of biometrics the fact that the biometrics does not change over time, is at the same time its greatest liability once a set of biometric data has been compressed, it is compromised forever. To address this issue, we have proposed applying repeatable non invertible distortions to the biometric signal. Cancellation simply requires the specification of a new distortion transform. Privacy is enhanced because different distortions can be used for different services and the true biometrics are never stored or revealed to the authentication server. In addition, search intentionally distorted biometrics cannot be used for searching legacy data bases and will thus alleviate some privacy violations concern however this Gaussian pyramid compression algorithm is compared with other best suited biometric compression algorithms and the results will be compared.

With the use of digital cameras, requirements for storage, manipulation and transfer of digital images has grown explosively. These image files contain data with the combination of information and redundancy. Information is the portion of data that must be preserved in its original form in order to correctly interpret the meaning of the data. Redundancy [5] is that portion of data that can be removed when it is not needed or can be reinserted to interpret the data when needed. Data compression [7] is the technique to reduce the redundancies in representation in order to decrease storage requirement and hence communication costs. The redundancy in data representation is reduced such a way that it can be subsequently reinserted to recover the original data which is called decompression of data. There exist two types of image compression lossless and lossy. In lossless compression technique [4], coding eliminates only redundant information that one can recover the exactly upon decomposition of file. The reconstructed image is nearly identical to the original image and thus it achieves less compression. While in lossy image compression, as the name lossy implies some amount of data will be lost during compression. It eliminates redundant as well as irrelevant information and thus permits only an approximate reconstruction of original image than an exact duplicate.

4. IMAGE COMPRESSION USING GPC

Gaussian Pyramid compression techniques are applied to images. With two-dimensional images the pyramid is split into several layers with each layer being of a fraction of the original images's resolution. One such algorithm works by taking the original image and passing a filter over it, such as a guassian blur. Other such methods can include scaling down the original image to a quarter of its original size and then scaling it up again to the origianl size using various interpolation methods.

4.1 GPC Structure

The GP structure can be used for compression purposes. Using the pyramid coding scheme, we decompose the original image into several sub-images depending on the signal characteristics. The initial step in pyramid coding is to low-pass filter the original image GP_0 to obtain image GP_1. Actually GP_1 is a reduced version of GP_0 in that both resolution and sample density are decreased. In a similar way, we form GP_2 as a reduced version of GP_1 and so on. Filtering is performed by a procedure equivalent to convolution with one of a family of local, symmetric weighing functions.

The pyramid scheme codes an input image in a multi-resolution representation in the same way as the generation of sub-images of various scales. Here, how resolution sub-images GP_k are created by passing GP_{k-1} through a low-pass filter H. In the encoder scheme [1], we transmit sub-images $\{L_0, L_1, L_K, \quad GP_{k+1}\}$ obtained by

$$
\begin{aligned}
&L_0=GP_0-GP_{1i}\\
&L_1=GP_1-GP_{2i}\\
&....\\
&.... \qquad\qquad (1)\\
&....\\
&L_k=GP_k-GP_{(k+1)i}\\
&GP_{k+1}
\end{aligned}
$$

Where L_k is the difference sub-image at the k^{th} level, GP_k is the low-resolution sub-images of the k^{th} level and GP_{ki} is the interpolated version of GP_k (using filter F)

In the decoding scheme, we reverse the equation (1) to get the original signal GP_0. The Pyramid representation has been introduced in the literature for coding purposes as it was shown to be a complete representation. Perfect reconstruction is guaranteed if there is no quantization of the transmitted data, regardless of the choice of filters H and F.

Suppose the image is represented initially by the array g_0 which contains C columns and R rows of pixels. Each pixel represents the light intensity at the corresponding image point by an integer I between 0 and k-1. This image becomes the bottom or zero level of the Gaussian Pyramid. Pyramid level 1 contains image g_1, which is reduced version of g_0. Each value within level 1 is computed as a weighted average of values in level 0 within a 2-by-2 window. Each value within level 2, representing g_2, is then obtained from values within level 1 by applying the same portion of weights. A graphical representation of this process in one dimension is given in fig. 1. The size of the weighting function is not critical. We have selected the 2-by-2 pattern because it provides adequate filtering at low computational cost.

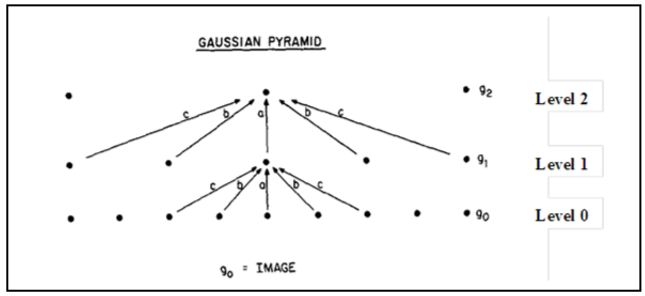

Fig.1: Gaussian Pyramid

Each row of dots represents nodes within a level of the pyramid. The value of each node in the zero level is just the gray level of a corresponding image pixel. The value of each node in a high level is the weighted average of node values in the next lower level. Note that node spacing doubles from level to level, while the same weighting pattern or “generating kernel” is used to generate all levels.

The GPC algorithm has been implemented in Matlab to progressive image transmission using jpeg image up to 2 levels is shown in the fig. 2. The GP consists of low-pass filtering (LPF) of the previous stage and corresponding sub-sampling of the filtered output.

Input image is labeled as G_0; the LPF versions are labeled G_1 through G_{K+1} with decreasing resolutions. A recursive procedure allows for the creation of the pyramid, as follows:

$$
\begin{aligned}
G^0_{n+1} &= W * G_n\\
G_{n+1} &= \text{subsampled}\, G^0_{n+1} \qquad (2)
\end{aligned}
$$

where Gn is termed the nth level Gaussian image in equation (2). Generally, the weighting function, W, is Gaussian in shape and normalized to have the sum of its coefficients equal to 1.

The original image level 0 measures 255 by 255 pixels and each higher-level array is roughly half the dimension of its predecessor. Thus, level 2 measures just 64 by 64 pixels. We have selected the 2-by-2 patterns because it provides adequate filtering at low computational cost. The low-pass filter effect of the Gaussian pyramid, which is of the order of 1.5:1 for RGB image and 1.6:1 for gray image.

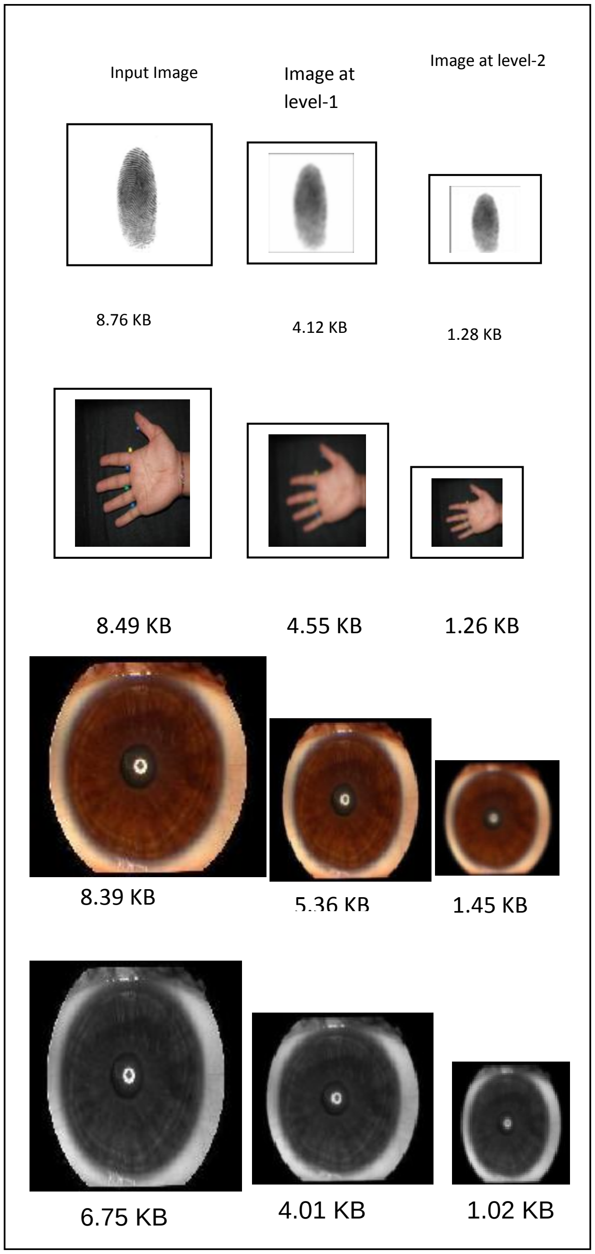

Fig.2 image transmission using jpeg image

4.2 GPC Algorithm

Input (1) Infile is the input file name
(2) Level is a scaling factor (positive integer)
(3) Outfile is output file name
(4) Output:
(5) Compressed file is the outfile
(6) Method:

(1) Accept the input image A can be RGB or GRAY scale
(2) If given image is GRAY image perform step 3
(3) Image compression is done by low pass filter the image using convolution fast fourier transform
(4) if image is RGB, low pass filter the image using convolution fft for red, green and blue separately
(5) Downsampling the image and save image in outfile Image reconstruction is by upsampling the image.
(6) Image reconstruction is by upsampling the image.

5. EXPERIMENTAL RESULTS AND PERFORMANCE

Once image compression is designed and implemented it is important to evaluate the performance. This evaluation should be done in such a way to be able to compare results against other image compression techniques. The compression efficiency is measured by using the CR. The quality of the image is analyzed by measuring PSNR [1] and MSE.

5.1 Compression Ratio (CR)

The performance of image compression can be specified in terms of compression efficiency is measured by the compression ratio or by the bit rate. Compression ratio is the ratio of the size of the original image to the size of the compressed image.

CR=Size of original image / Size of the compressed image
If CR > 1, it is a positive compression.
If CR < 1, it means a negative compression.

Whenever this ratio is large, it indicates that the compression is better otherwise the compression is weak.

5.2 Bits Per Pixel (BPP)/ Bit Rate (BR)

The bit rate is the number of bits per pixel required by the compressed image. Let b be the number of bits per pixel (bit depth) of the uncompressed image, CR the compression ratio and BR the bit rate. The bit rate ratio is given by BR = b / CR.

5.3 The Mean Squared Error (MSE)

MSE [7] indicate the average difference of the original image data and reconstructed data and results the level of distortion in equation (3). The MSE between the original data and reconstructed data is

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[f(i,j) - f'(i,j) \right]^2 \qquad (3)$$

where m and n are the row and column size of the image. f=original image of size m x n and f' = reconstructed image of size m x n.

Table 1: Comparison of image compression using different image type

5.4 Peak Signal to Noise Ratio (PSNR)

Peak Signal to noise is one of the quantitative measures for image quality evaluation which is based on the Mean Square Error (MSE) of the reconstructed image. PSNR is expressed by equation (4).

	Image Type	*Uncompressed Image (KB)*	*Compressed Image (KB)*	
	RGB Image		***Level 1***	***Level 2***
Iris Images	'001L_1.jpg','001L_1_C.jpg' [2]	8.383	5.36	1.45
	'001L_1.png','001L_1_C.png'	77.1	51.8	5.99
	'001L_1.bmp','001L_1_C.bmp'	191	187	12
Palm Prints	'Img_001_1.jpg','Img_001_1_C.jpg' [5]	8.49	4.55	1.26
	'Img_001_1.png','Img_001_1_C.png'	97.2	49.8	5.52
	'Img_001_1.bmp','Img_001_1_C.bmp'	191	187	12
	GRAY Image			
Iris Images	'001LG_1.jpg','001LG_1_C.jpg' [2]	6.72	4.01	1.03
	'001LG_1.png','001LG_1_C.png'	25.4	17.9	2.19
	'001LG_1.bmp','001LG_1_C.bmp'	64.8	64.3	5.05
Finger Prints	'101_2.jpg','101_2_C.jpg' [3]	8.76	5.12	1.28
	'101_2.png','101_2_C.png'	17.4	9.6	1.17
	'101_2.bmp','101_2_C.bmp'	64.8	64.3	5.05

$$\mathrm{PSNR} = 10 \log \frac{\mathrm{MAX}_i^2}{\mathrm{MSE}} \qquad (4)$$

where MAX_i is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. In general, when samples are represented using GPC with B bits per sample, MAX_i is 2B-1. For color images with RGB values per pixel, the definition of PSNR is the same except the MSE is the sum over all the squared value differences divided by image size and three. In this case, the large results mean that there is a small noise in the compression system image quality of the reconstructed and image is better. When value of PSNR is small, it means that the compression performance is weak.

Table 1 shows the size of the uncompressed image and compressed image at two different levels. While increasing the decomposition level, the size of the image is decreased. The PSNR value is maximum at decomposition level 1. Table 2 represents performance summary and investigation of Gaussian Pyramid compression on different image types.

Table 2: Performance Measures on Gaussian Pyramid Compression

	Image Type / *RGB Image*	*SNR*	*PSNR*	*MSE*	*BPP*	*CR*
Iris Images	'001L_1.jpg','001L_1_C.jpg'[2]	11.12	28.95	82.64	0.2252	1.563
	'001L_1.png','001L_1_C.png'	11.28	29.19	78.22	2.1768	1.488
	'001L_1.bmp','001L_1_C.bmp'	11.35	29.26	77.07	7.8876	1.019
Palm Prints	'Img_001_1.jpg','Img_001_1_C.jpg' [5]	15.75	32.26	38.61	0.1912	1.86
	'Img_001_1.png','Img_001_1_C.png'	17.03	33.54	28.76	0.2320	1.75
	'Img_001_1.bmp','Img_001_1_C.bmp'	16.92	33.45	29.35	0.5061	1.02
	GRAY Image					
Iris Images	'001LG_1.jpg','001LG_1_C.jpg' [2]	12.62	22.42	72.01	0.5063	1.674
	'001LG_1.png','001LG_1_C.png'	12.41	22.21	90.69	2.2621	1.42
	'001LG_1.bmp','001LG_1_C.bmp'	12.56	22.36	77.26	8.10	1.008
Finger Prints	'101_2.jpg','101_2_C.jpg' [3]	17.08	22.71	303.6	0.1734	1.71
	'101_2.png','101_2_C.png'	17.18	18.03	907.04	1.2098	1.13
	'101_2.bmp','101_2_C.bmp'	17.18	18.55	907.04	0.6366	1.008

Experiments are conducted on UPOL[2] iris images of various file formats using Matlab 7.0 on an Intel Pentium IV 3.0 GHz processor with 512MB memory. Gaussian Pyramid compression on iris images is performed and it is compared based on SNR, PSNR, MSE and Compression Ratio in percentage for both RGB and gray images.

6.CONCLUSION

This research work demonstrates that the potential of Gaussian Pyramid compression technique with respect to SNR, PSNR, MSE and Compression Ratio on both RGB and gray scale biometric images. When compared with different image types, RGB of jpg image is the best image type with respect to bits per pixel. During comparison of compression ratios, Pyramid method has the least value in gray scale image. Thus, on comparing the various image types, Gaussian Pyramid compression plays an excellent role in RGB image in case of bpp and best Compression Ratio in gray scale images.

References

[1] Afshan Mulla, Namrata Gunjikar and Radhika Naik, "Comparison of Different Image Compression Techniques", IJCA, Vol 70, pp 7-11, 2013.

[2] Burt P.J and Adelson E.A., "The Laplacian Pyramid as a Compact Image Code", IEEE Transactions on Communications, Vol. COM-31,pp. 532-50,1983.

[3] Daugman .J and Downing .C , "Effect of severe image compression on iris recognition performance", IEEE Transactions on Information Forensics and Security, 3(1):52–61, 2008.

[4] Doaa Mohammed, Fatma Abou-Chadi, "Image Compression Using Block Truncation Coding" JSAT, pp 9-13, 2011.

[5] Guo, Z.H., D. Zhang, L. Zhang and W.M. Zuo, "Palm print verification using binary orientation cooccurrence vector", Patt. Recogn. Lett., 30: 1219-1227, 2009

[6] Khuwaja, G. A.; Tolba, A. S., "Fingerprint image compression", Neural Networks for Signal

[7] Kiran Bindu, etal., "A Comparative Study of Image Compression Algorithms", IJRCS, Vol 2, Issue 5,pp 37-42,2012.

[8] Michal Dobes and Libor Machala, UPOL Iris Image Database, 2004.

[9] Nigel M. Allinson, Jeevandra Sivarajah, Ian Gledhill, Michael Carling, and Lesley J. Allinson Robust Wireless , "Transmission of Compressed Latent Fingerprint Images, IEEE Transactions on Information Forensics and Security", Volume: 2 Issue:3. pp. 331–340. 2007.

[10] Pardeep Singh, et al., "A Comparative Study: Block Truncation Coding, Wavelet, Embedded Zerotree and Fractal Image Compression on Color Image", Vol 3, Issue-2, pp.10-21, 2012.

[11] Weinhandel .G, H. St¨ogner, and A. Uhl, "Experimental study on lossess compression of biometric sample data", Proceedings of the 6th International Symposium on Image and Signal Processing and Analysis, ISPA '09, Salzburg, Austria, Sep. 2009.

K.C.Chandrasekaran is working as a Associate Professor and Head, Department of Computer Sciencce, Alagappa Government Arts College, Karaikudi Sivagangai (Dt).T.N. India He has 20 years of teaching experience for both U.G. and P.G. courses . One Research paper published in National Conference organized by Presidency College, Chennai.

Dr.K.Kuppusamy is working as a Professor in the Department of Computer Science and Engineering, Alagappa University, Karaikukdi, Tamilnadu, India. He received his Ph.D degree in Computer Science from the dept of computer science and Engineering Alagappa University, Karaikudi, Tamilnadu in the year 2007. He has 25 years of teaching experience in the field of Computer science. He has published many papers in International & National Journals and presented in National and International conferences. His areas of research interests includes Information/Network Security, Algorithms, Neural Networks, Fault Tolerant Computing, Software Engineering and Optimization Techniques.