BINARY IMAGE COMPRESSION USING THE RADON TRANSFORM

G.R. Ramesh and K. Rajgopal,
Department of Electrical Engineering,
Indian Institute of Science,
Bangalore 560 012

ABSTRACT: A new technique for compression of binary images is proposed. The Radon transform is used to convert a binary image to a set of 1-dimensional (1-D) non-binary sequences, which are coded using 1-D techniques. A binary image can be reconstructed from a very small number of projections and this leads to significant compression. The compression ratio for a binary image of size $N \times N$ is inversely proportional to N .

1. Introduction

The need for efficient storage and transmission of graphics and two-tone (binary) images has increased enormously. As a result, several techniques have been developed for binary image compression. Important methods include run-length coding, white block skipping, prediction differential quantization (PDQ), relative address coding and predictive coding [1,2]. In general, all these techniques (except PDQ) do not exploit efficiently, the inter-dimensional correlation in the 2-dimensional (2-D) data. While run length coding and its extensions are simple and efficient, they are very sensitive to channel errors. In 2-D linear predictive coding, the choice of the prediction mask and the complexity of the predictor are major issues [2,3]. Further, the existing techniques for binary image compression differ significantly from those used for compression of gray level images and hence, a separate coding system is required for binary images.

In this paper, we present a new scheme for compression of binary images using the Radon transform (RT). The RT is used as a tool to reduce the problem of coding 2-D binary sequence to that of coding a set of 1-D, non-binary sequences i.e., projections of the image. The procedure for reconstruction consists of decoding the projections, followed by reconstruction of the binary image from these projections. A major advantage of this approach arises from the fact that a binary image can be reconstructed from very few projections [5,6], which results in significant compression. It is possible to apply both predictive and transform techniques in the proposed scheme. In this paper, we consider the application of linear prediction and the discrete cosine transform (DCT).

The organization of the paper is as follows. Section 2 briefly outlines reconstruction of binary images from their projections. Section 3 describes predictive and transform techniques for coding binary images using the

RT. Simulation results are presented in Section 4. Section 5 concludes the paper.

2. Reconstruction of binary images from their projections

The Radon transform of an image $f(x,y)$ is the integral of $f(x,y)$ along all possible lines in the plane containing the image:

$$p(\theta, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \delta(t - x \cos(\theta) - y \sin(\theta)) dx dy \quad \dots\dots\dots (1)$$

where θ and t are the parameters of the straight line in the normal form:

$$t = x \cos(\theta) + y \sin(\theta) \quad (2)$$

The term 'projection' refers to the Radon transform integral (1) evaluated at angle θ , denoted as $p_\theta(t)$ to emphasize the fact that it is a 1-D function. Thus, the Radon transform is used to produce a set of 1-D signals (projections) from a 2-D signal $f(x,y)$. This dimension reducing property of the Radon transform allows the conversion of a 2-D processing task to a set of simpler 1-D processing tasks independently on each of the projections.

The inverse Radon transform mathematically corresponds to finding the function $f(x,y)$ from its projections. There are several algorithms that approximate the inverse Radon transform, to reconstruct an image from a set of its projections, such as the convolution backprojection (CBP) or the Fourier reconstruction technique. However, binary image reconstruction algorithms are basically iterative in nature and resemble the algebraic techniques for reconstructing gray level images. A striking feature of such algorithms is that they require very few projections for a satisfactory reconstruction [5,6].

In this paper, we use the algorithm proposed in [6], to reconstruct a digital binary image from its (coded) projections, at $\theta = 0, \pi/4, \pi/2$ and $3\pi/4$. The projections of an image of size $N \times N$, where $N=2M+1$, are computed as follows:

$$p_0(t) = \sum f(t, n) \quad -M \leq t < M \quad (3 a)$$

$$p_{\pi/4}(t) = \sqrt{2} \sum f(n, t-n), \quad -2M \leq t < 2M \quad (3 b)$$

$$p_{\pi/2}(t) = \sum f(n, t), \quad -M \leq t < M \quad (3 c)$$

$$p_{3\pi/4}(t) = \sqrt{2} \sum f(n, t+n), \quad -2M \leq t < 2M \quad (3 d)$$

The summation is performed over the $N \times N$ image

support. Note that, while the projections at $\theta = 0$ and $\pi/2$ are spaced at unit distances, those at $\theta = \pi/4$ and $3\pi/4$ are spaced at $1/\sqrt{2}$ units.

The problem of reconstructing $f(m,n)$ from the set of 4 projections, is formulated as that of solving the set of linear equations, (3). The algorithm is an error-reduction scheme, and begins by assuming an initial estimate for $f(m,n)$. In each iteration, $f(m,n)$ is first estimated based on projections at one of the four angles, $\theta = \theta_i$, which forms the initial estimate for estimation based on the projection at a different angle, θ_{i+1} , as follows:

$$\hat{f}_{i+1}(m,n) = f_{e_i}(m,n) + e_{e_{i+1}}[t(m,n)] \quad (4)$$

where,

$$e_{e_{i+1}}(t) = [p_{e_{i+1}}(t) - \hat{p}_{e_{i+1}}(t)] / L_{e_{i+1}}(t) \quad (5)$$

and $\hat{p}_{e_{i+1}}(t)$ is the projection of $\hat{f}_{e_i}(m,n)$ at $\theta = \theta_{i+1}$. $L_{e_{i+1}}(t)$ is the number of pixels along the line (2) at index perpendicular distance t and $\theta = \theta_{i+1}$. $e_{e_{i+1}}(t)$ is the estimation error in terms of projections. $t(m,n)$ denotes the index t corresponding to the line (2), passing through the point (m,n) , at $\theta = \theta_{i+1}$. The above step is repeated to utilize the projections at all the angles.

The final step in each iteration consists of applying the constraint that, $0 \leq f \leq 1$. The constrained values of $f(m,n)$ form the initial estimate for the next iteration. The last step of the reconstruction algorithm involves thresholding i.e., setting the values of f to 1 if $f \geq 0.5$, and zero otherwise.

3. Binary Image Coding

The basic idea here is to use the Radon transform to reduce the binary image to a set of 1-D sequences i.e., projections at various angles. The projections of a binary image are 1-D non-binary sequences, which are coded by using well known 1-D techniques.

The first step is therefore, to compute the Radon transform (the set of projections) of the digital binary image $f(m,n)$, using (3). Each of the projections $p_e(t)$, is made zero mean by subtracting from it, its mean μ_e .

$$d_e(t) = p_e(t) - \mu_e \quad (6)$$

In this paper, we consider the application of LPC and the DCT for coding the data sequences, $d_e(t)$. The algorithm is described in the following section.

3.1 Linear predictive coding (LPC)

The principles of 2-D linear predictive coding have been applied to binary images [1-4]. The difference is that, in the case of binary images, the error signal is also a binary sequence and therefore, no quantizer is required. Redundancy in the binary image will be reflected in long runs of zeros in the error signal, which may be then coded using run-length techniques. However, direct 2-D LP techniques considered so far, do not exploit efficiently, the inter-dimensional correlation in the 2-D data. Further, the choice of

the prediction mask (memory set), complexity of the predictor, and stability, remain as major issues [2,3,4]. We propose an algorithm which uses simpler 1-D LP techniques on the set of projections (i.e., the Radon transform) of the binary image. Stability is guaranteed by using the so called autocorrelation method or estimating the 1-D LP parameters.

Each of the data sequences $d_e(t)$, is independently modeled as an autoregressive (AR) sequence of order m_e :

$$d_e(t) = \tilde{d}_e(t) + e_e(t) \quad (7)$$

$$\text{where, } \tilde{d}_e(t) = - \sum a_e(k) d_e(t-k) \quad (8)$$

is an estimate of $d_e(t)$; $a_e(k)$, $k=1,2,\dots,m_e$, are the AR model (LP) parameters and $e_e(t)$, the error signal. The AR parameters $\{a_e(k)\}$ are computed from the data $d_e(t)$, using the constrained minimization procedure due to Burg. This method guarantees the stability of the inverse prediction-error filter. These parameters are used to compute $\tilde{d}_e(t)$ and hence, the error signal $e_e(t)$. The error signal sequences $e_e(t)$ are quantized and stored or coded for transmission, along with $\{a_e(k)\}$ and μ_e . Due to its inherent low variance, $e_e(t)$ requires few bits to represent and this is the essence of linear predictive data compression.

At the receiver, the data sequences $d_e(t)$ are retrieved by driving the inverse prediction-error filter by $e_e(t)$ (equation 7). The projections $p_e(t)$ are obtained by adding μ_e to $d_e(t)$, from which, the binary image is reconstructed using the algorithm described in the previous section.

The procedure may be summarized as follows:

1. Compute the projections $p_e(t)$ of the digital binary image $f(m,n)$.
2. Code the projections $p_e(t)$ using 1-D LPC.
3. Reconstruction: Retrieve $p_e(t)$ by driving the inverse prediction-error filter by the (quantized) error signal $e_e(t)$. The retrieved projections are then used to reconstruct the image $f(m,n)$ using a binary image reconstruction algorithm.

The procedure outlined above suggests the implementation of an intra-frame adaptive predictive image coding scheme (adaptive differential pulse code modulation or ADPCM). The scheme is illustrated in fig.1. The block 'REC' indicates image reconstruction from projections and 'VAR' corresponds to computation of variance. Calculation of the variance and transmission of the quantizer step-size information can be avoided by using Jayant's 1-word memory adaptive quantizer [7]. A major advantage of this quantizer consists in increasing the dynamic range of the quantizer.

3.1.1 Compression Ratio

Let N_p be the number of projections required to satisfactorily reconstruct the binary image, m_e the order of the predictor and B , the number of bits required to quantize the error signal $e_e(t)$. Then, in general

Fig. 1(a) ADPCM Coder

Fig. 1(b) Decoder

(assuming for simplicity, that the same order is used for all the projections) the compression ratio, defined as the ratio of the number of bits representing the coded image to that of the original image, is given by:

$$CR = \frac{N_p (\text{max} 8 + N/2 \cdot B)}{N \times N} \quad (9)$$

8 bits have been assumed for the prediction coefficients. The contribution of the quantizer step-size information and the mean are neglected. $N/2$ corresponds to the length (in number of points) of the projection data and hence that of the error signal. If $N_e \ll N$, which is normally the case, then the compression ratio reduces to:

$$CR \approx \sqrt{2} N_p B / N \quad (10)$$

Thus, the compression ratio depends upon the number of projections required for satisfactory reconstruction of the image, and B is the number of bits required to quantize the error signal. The factor N_p in general depends upon the complexity of the image under consideration. However, as mentioned earlier, a binary image can be reconstructed from very few projections and this leads to a significant compression. In the particular case considered in this paper, with 4 projections (N points for horizontal and vertical projections, and $2N-1$ points for those along the two diagonals), the CR is given by:

$$CR \approx 6 B / N \quad (11)$$

Thus, the compression ratio is seen to be inversely proportional to N . That is, as the size of the image increases, the achievable compression also increases linearly, and this looks very impressive. In practice however, when N is very large and/or the image has finer details, the image will have to be divided into smaller blocks (subimages) and each of them coded separately. This also serves to make the coder adapt to the local image characteristics. If an $N \times N$ image is divided uniformly into N_b blocks, the overall compression ratio is:

$$CR_o = CR / N_b \quad (12)$$

where, CR is the compression ratio with the $N \times N$ image considered as a whole. Thus, compression is scaled down by the factor $\sqrt{N_b}$. However, significant compression can still be

achieved in the case of large images, by an appropriate choice of the parameters. For example, if $N=1024$, $N_b=16$, then, for LPC with $B=3$ and 2, CR_o is approximately 1/14 and 1/22 respectively.

3.2 Transform compression

Transform techniques have not been applied for binary image compression, since they do not offer significant compression. However, it is possible to use efficient transforms such as the DCT to compress the (non-binary) projections of the image. Compression is achieved by applying the DCT to the signal (projection, in the present case) and retaining and encoding only a fraction of the rest of the coefficients does not seriously degrade the signal. This is due to the excellent energy compaction property of the DCT. Advantages of transform techniques for compression include higher efficiency and robustness to transmission degradations. The algorithm is summarized below:

- (i) Compute the projections $p_e(t)$ of the image $f(m,n)$.
- (ii) Code the 1-D, non-binary sequences $p_e(t)$, using DCT. Note that the DCT can be computed via an FFT algorithm.
- (iii) At the receiver, the projections $p_e(t)$ are retrieved by IDCT (via IFFT) and used to reconstruct the image $f(m,n)$.

The scheme is illustrated in Fig. 2.

3.2.1 Compression ratio

Let C be the fraction of the DCT coefficients (corresponding to each of the projections) retained. Assuming that 4 projections represent the image (as in the case of LPC), the compression ratio is:

$$CR_{DCT} = 6 B_{DCT} C / N \quad (13)$$

B_{DCT} is the number of bits required to represent each of the DCT coefficients (assuming a uniform bit allocation). Better compression can be achieved by efficient bit allocation of the DCT coefficients. As discussed in the case of LPC, large images have to be divided into smaller blocks and each of them coded separately. Apart from making the coder adapt to local image characteristics, this also serves to reduce storage and computational

Fig. 2: Transform compression scheme

requirement. If the image is divided uniformly into N_b blocks, the overall compression ratio CR_o will be CR/N_b , where CR is the compression ratio with the $N \times N$ image considered as a whole. With $N=1024$ and $N_b=16$, the values of CR_o with $Bpct=8$ and $C=0.4$ and 0.3 , are $1/18$ and $1/24$ respectively.

3.3 Remarks:

- 1) Errors due to quantization and channel degradations tend to get distributed over the entire image during the process of reconstruction from projections.
- 2) As the RT reduces a binary image to a set of non-binary sequences, binary as well as gray level images can be coded using the proposed scheme.
- 3) Coding of the projections can be performed in parallel.
- 4) The projection process tends to have an averaging effect on background noise, reducing its effect.
- 5) Major disadvantages of the proposed scheme are:
 - (i) errors due to reconstruction from projections.
 - (ii) unsuitability for continuous on-line image transmission, since the algorithms for the reconstruction of a binary image from its projections are iterative and hence time consuming.

4. Simulation

Simulation studies were carried out on several computer generated test images. One example is presented in Figs. 3-5. Fig. 3 shows the original image of size 129×129 . The size of each of the character is 23×15 . This corresponds to a printed character of size $3mm \times 2mm$ sampled at 200 points per inch. Note however, that the entire image does not correspond to a standard document. Four projections of the image were coded by LPC using a simple uniform quantizer with $B_{lpc}=3$ and 2. The compression ratios obtained were $1/7.2$ and $1/10.8$ respectively. The results are displayed in Fig. 4. The corresponding results with DCT coding of the projections with $Bpct=8$, $C=0.4$ and $C=0.3$, are displayed in Fig. 5. The compression ratios are $1/6.7$ and $1/9$ respectively. It is possible to achieve better performance by a proper bit allocation of the DCT coefficients.

In the present case, we have considered four projections to represent the binary image. However, it is possible to consider a greater number of projections by using a discretization of the Radon transform, followed by a suitable reconstruction algorithm.

As mentioned earlier, a major disadvantage of the proposed technique is the error resulting from the process of image reconstruction from projections; degradation can be relatively severe on documents with finer details.

5. Conclusions

A new technique for binary image compression has been presented, where the Radon transform is made use of as a tool to reduce the 2-D binary image coding problem to a set of 1-D problems of coding non-binary sequences. Both predictive and transform techniques have been discussed. The compression ratio for an $N \times N$ image has been shown to be inversely proportional to N . Simulation results showing the performance of the algorithm have been presented. Various other features of the technique have been highlighted. Application of different reconstruction algorithms are being investigated. The effects due to quantization and channel errors on the quality of the reconstructed image are being studied.

Acknowledgment

The authors wish to thank Ganesh Murthy C.N.S. for useful suggestions, encouragement and support.

REFERENCES

- [1] R.B. Arps, "Bibliography on Binary Image Compression", Proc. IEEE, Vol.68, No.7, July 1980, pp:922-924.
- [2] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.
- [3] H. Kobayashi and L.R. Bahl, "Image Data Compression by Predictive Coding I: Prediction Algorithms", IBM J. of Res. and Dev., March 1974, pp:164-171.
- [4] T.S. Huang, "Coding of Two-Tone Images", IEEE Trans. Communications, COM-25, Vol.11, November 1977.
- [5] G.T. Herman, "Reconstruction of Binary Patterns From a Few Projections", in A Gunther et al. (eds), International Computing Symposium 1973, North-Holland Publishing Co., 1974.
- [6] M. Soumekh, "Binary Image Reconstruction from Four Projections", Proc. ICASSP'88, pp:1280-1283.
- [7] N.S. Jayant, "Adaptive Quantization With a One-Word Memory", The Bell System technical Journal, Vol.52, No.7, Sept. 1973, pp:1118-1144.

CAN
YOU
SEE

Fig. 3: Test image

CAN
YOU
SEE

(a)

CAN
YOU
SEE

(b)

Fig. 4: Reconstruction of the test image from its projections coded by LFC:
(a) $B=3$ ($CR=1/7.2$) and (b) $B=2$ ($CR=1/10.8$)

CAN
YOU
SEE

(a)

CAN
YOU
SEE

(b)

Fig. 5: Reconstruction of the test image from its projections coded using the DCT, with $Bocr=8$: (a) $C=0.4$ ($CR=1/8.7$) and (b) $C=0.3$ ($CR=1/9$)