J. Kittler and M.S. Nixon (Eds.): AVBPA 2003, LNCS 2688, pp. 343–350, 2003. (c) Springer-Verlag Berlin Heidelberg

Orientation Scanning to Improve Lossless Compression of Fingerprint Images

Johan Thärnå, Kenneth Nilsson, and Josef Bigun

School of Information Science, Computer and Electrical Engineering (IDE)
Halmstad University, P.O. Box 823, 301 18 Halmstad, Sweden
{Kenneth.Nilsson,Josef.Bigun}@ide.hh.se

Abstract. While standard compression methods available include complex source encoding schemes, the scanning of the image is often performed by a horizontal (row-by-row) or vertical scanning. In this work a new scanning method, called ridge scanning, for lossless compression of fingerprint images is presented. By using ridge scanning our goal is to increase the redundancy in data and thereby increase the compression rate.

By using orientations, estimated from the linear symmetry property of local neighbourhoods in the fingerprint, a scanning algorithm which follows the ridges and valleys is developed. The properties of linear symmetry are also used for a segmentation of the fingerprint into two parts, one part which lacks orientation and one that has it.

We demonstrate that ridge scanning increases the compression ratio for Lempel-Ziv coding as well as recursive Huffman coding with approximately 3% in average. Compared to JPEG-LS, using ridge scanning and recursive Huffman the gain is 10% in average.

1 Introduction

The compression performance is sensitive to how data is presented to the sequential encoders, which inherently assume data being one dimensional. Scanning corresponds to the order that the pixels of the fingerprint are read. Horizontal scanning (row-by-row) is a common scanning method adopted in many lossless image compression algorithms such as for example JPEG-LS [8], and PNG [3]. To the best of our knowledge, published lossless compression schemes adopted to fingerprints are lacking.

The aim of scanning by following the ridges in the fingerprint is to reduce the local variance in the resulting data input to the encoder, and thereby increase the performance of the compression. We do not present a full lossless compression scheme but rather present an important element of such a scheme: scanning. We also present results that show JPEG-LS does not present an advantage as compared to a straight forward use of Lempel-Ziv or recursive Huffman utilized together with ridge scanning.

J. Kittler and M.S. Nixon (Eds.): AVBPA 2003, LNCS 2688, pp. 343–350, 2003.

1.1 Linear Symmetry and Local Orientations

Linear symmetry [2, 1] is a way of describing an image by simple patterns, i.e. line patterns, and how they are rotated. The complex moments I_{20} and I_{11} can be used to estimate the linear symmetry of an image neighbourhood and are defined in the frequency domain as

$$I_{11} = \int\int (\omega_x + j\omega_y)(\omega_x - j\omega_y)|F(\omega_x, \omega_y)|^2 d\omega_x d\omega_y. \quad (1)$$

$$I_{20} = \int\int (\omega_x + j\omega_y)^2 |F(\omega_x, \omega_y)|^2 d\omega_x d\omega_y. \quad (2)$$

I_{20} and I_{11} can be computed in the spatial domain by [2]

$$I_{11} = |z(x, y)| * h(x, y). \quad (3)$$

$$I_{20} = z(x, y) * h(x, y). \quad (4)$$

where $h(x, y)$ is a gaussian window function, described by

$$h(x, y) = e^{-\frac{x^2+y^2}{2\sigma_h^2}}. \quad (5)$$

and $z(x, y)$ is the squared complex valued gradient image, computed as

$$z(x, y) = (\hat{f}_x + j\hat{f}_y)^2. \quad (6)$$

Thus, I_{20} and I_{11} are formed by gaussian averaging $z(x, y)$ and $|z(x, y)|$ respectively in an image neighbourhood defined in size by σ_h.

Equations 3 and 4 state that if all the gradient vectors in the local neighbourhood, defined by σ_h, points in the same direction, the magnitude of I_{20} will be the same as I_{11}. If instead the variance in direction of the vectors within the neighbourhood is large, then the magnitude of I_{20} will decrease compared to if they all had the same direction. Thus a certainty measure of linear symmetry for the neighbourhood, and for the angular information contained within I_{20}, is $|I_{20}|/I_{11}$.

The dominant orientation in the local neighbourhood is extracted from I_{20} as

$$\theta = \frac{1}{2} arg(I_{20}). \quad (7)$$

where the factor $\frac{1}{2}$ is due to the double angle representation of $z(x, y)$ in equation 6. Left of figure 1 displays the magnitude of I_{20} for the fingerprint in figure 3. Middle figure shows the magnitude of the low-pass filtered I_{20}-image, while on the right θ is presented.

We use the local orientation θ to estimate the appropriate direction for the scanning.

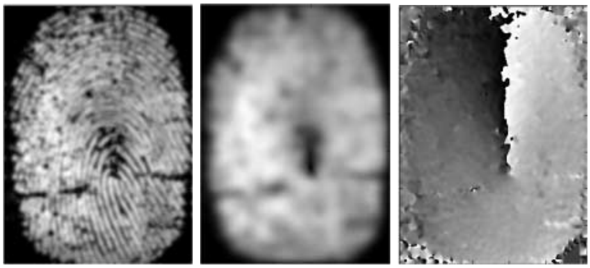

Fig. 1. Left: displaying $|I_{20}|$ for the fingerprint in figure 3. Middle: the low-pass filtered (downsampling followed by upsampling) version of the left image. Right: θ for the fingerprint (not low-pass filtered)

Segmentation. The certainty measure of linear symmetry offers the possibility of segmenting the image in two parts . A boolean (binary) image, L_b, is created, telling whether the certainty measure exceeds a certain threshold or not. Figure 2 displays, from left to right, the binary image L_b, the part of the fingerprint that exceeds, respectively not exceeds the threshold value. When computing the certainty measure a low-pass filtered version of I_{20} is used. This gives the ability to encode the two parts separately as they ideally are uncorrelated.

If the value of $|I_{20}|$ exceeds the threshold, then the corresponding position in L_b is set to the estimated orientation angle θ, if not the corresponding position of L_b is set to a reserved value.

1.2 Ridge Scanning

Scanning corresponds to the order that a set of pixel samples are read. The usual method is the horizontal (row-by-row) scanning. However, we argue that the scanning method should depend on the information theoretic encoding approach taken at the final stage of compression.

Fig. 2. Left: the binary image L_b. Middle: the part of the fingerprint that exceeds the threshold for linear symmetry. Right: the part of the fingerprint that lacks linear symmetry

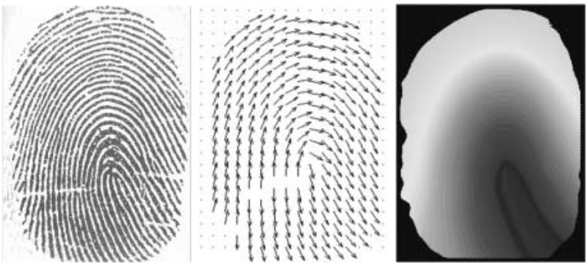

Fig. 3. Left: a fingerprint. Middle: the corresponding scanning directions (downsampled). Right: the scanning order for ridge scanning for the part of the fingerprint with distinct linear symmetry

The scanning order can be defined by any function, as long as there is a one-to-one mapping. Using the ridge scanning approach, the scanning order function is defined by the local orientations of the image. Figure 3 shows a fingerprint with corresponding scanning directions.

To reduce the overhead cost a downsampled version of the estimated orientation image, which is upsampled by the decoder to determine the scanning order and thereby decode the information, is used.

Figure 4 shows the scanning order calculation for the encoding/decoding procedures using ridge scanning. The encoder calculates a downsampled version of the local orientation image, L, upsamples it again to make the scanning consistent with the decoder. When the encoder has determined the scanning order matrix, $\boldsymbol{V}$, the pixels within the image can be scanned and put in a vector $\boldsymbol{y}$. The value of $\boldsymbol{y}$ at position n is determined by $y(n) = A(V(n))$, where $\boldsymbol{A}$ is the input image. The vector $\boldsymbol{y}$ can then be compressed.

The decoder receives the downsampled orientation image together with the compressed vector, upsamples the orientation image and calculates the scanning order. The decompressed data $\boldsymbol{y}$, can then be re-ordered in the correct way.

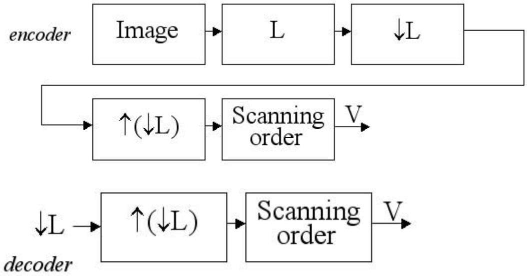

Fig. 4. Determining the scanning order for the encoder/decoder using ridge scanning

Figure 3 right displays the determined scanning order for the part of the fingerprint which was captured by the segmentation. Low gray intensity indicates a low scanning-order number, the scanning-order number increases with intensity.

1.3 Compression Algorithms

Lempel-Ziv Coding - ZIP. The Lempel-ziv coding algorithm [9] is a lossless compression algorithm, which is categorized as variable-to-fix length coding, meaning that the input block is of variable size, while the output size is fix. The output size depends on the size of the dictionary. The Lempel-Ziv algorithm exploits redundant patterns present in the input data.

Huffman Coding. Static Huffman coding [5] is a fix-to-variable length encoding algorithm. Huffman coding provides shorter codes for frequently used symbols at the price of longer codes for symbols occurring at a lower frequency. When the probabilities p_i for different symbols are known, it is possible to assign a different number of bits to each symbol, based on the symbols probability of occurrence.

The theoretical minimum average number of bits required to encode a data source with Huffman coding is given by the entropy of the data source, where the entropy η is defined as [4]

$$\eta = -\sum_{i=1} p_i \cdot log_2(\frac{1}{p_i}). \quad (8)$$

Recursive Huffman. Skretting et.al [6] has proposed a recursive splitting scheme of signals before entropy coding them using Huffman coding. The algorithm splits the signal into shorter parts with lower entropy, which can then be separately Huffman encoded. The algorithm includes a balancing procedure, such that the splitting does not continue if the size of the compressed data increases. The scheme seems to suit well for the ridge-scanning approach, as the scanning optimally scans in the ridges and valleys, where the variance is locally low.

In this paper we test the hypothesis that ridge scanning, compared to horizontal scanning, should increase the redundancy in data and therefore further increase the compression rate.

JPEG-LS. JPEG-LS [8] is an image compression algorithm which can be run in two different modes, lossless mode or near lossless mode. Near lossless corresponds to visually small changes between the reconstructed and the original data set that nevertheless can damage a fingerprint, e.g. a faint minutiae point can disappear. The algorithm features two different run modes. *Run Mode*, is a run-length encoder, and *Regular Mode*, is a predictive encoder which predicts the value for the current pixel from four of it's surrounding pixels. Results of JPEG-LS (in lossless mode) is provided as a benchmark to other results.

Fig. 5. The fingerprints used for testing. First row: 6_3, 7_4, 9_2, 10_8, 12_2 and second row: 18_3, 64_3, 86_8, 91_3, 99_6

2 Experiments and Results

Segmenting the image using the L_b-representation, two ideally uncorrelated parts are created which are scanned differently. The part with lack of linear symmetry is scanned vertically while the part with distinct local orientations is scanned using ridge scanning. The vector formed from the part of the image with lack of linear symmetry, $\mathbf{r}$, consists mainly of scanned areas with low variance in pixel colour. The variance within the ridge scanned part, $\mathbf{w}$, is typically higher (though the local variance is low). We are encoding the vectors independently.

The origin of the scanning is determined by the global orientation of the fingerprint, while the ridge scanning order is determined by the local orientation.

2.1 Data Set

The fingerprints used in the tests are from the database "db2a", used in the "Fingerprint Verification Contest 2000", "FVC2000", [7] available from the university of Bologna, Italy. The database consists of 8 fingerprints/person from 100 persons giving a total number of 800 fingerprints. The images are of size 364x256 pixels with a resolution of 500 dpi (dots per inch). The sensor used to capture the fingerprints was a low-cost capacitive sensor. Images are of the format uncompressed Tagged Image File Format (TIFF). The fingerprints 6_3, 7_4, 9_2, 10_8, 12_2, 18_3, 64_3, 86_8, 91_3, and 99_6 shown in figure 5 were used in the tests.

Table 1. Abbreviations for table 2

abbreviation	scanning	compression method
hz	horizontal	zip
sH	any	static Huffman
hrH	horizontal	recursive Huffman
rrH	ridge	recursive Huffman
rz	ridge	zip

Table 2. Bits per pixel (bpp). Original fingerprint has 8 bpp

fingerprint	JPEG-LS	hz	sH	hrH	rrH	rz
6_3	6.0510	6.1476	5.7722	5.5273	**5.3656**	5.9847
7_4	4.3854	4.2787	4.3318	4.1798	**4.0034**	4.2790
9_2	4.6481	4.4079	4.3209	4.1678	**3.9503**	4.3316
10_8	4.7169	4.2373	4.1210	4.0251	3.9266	**3.8983**
12_2	5.9876	6.5312	6.4168	5.7771	**5.7119**	6.4357
18_3	5.2967	4.9169	4.7275	4.4775	**4.2844**	4.6476
64_3	3.6756	3.3162	3.5804	3.5852	3.3001	**3.1614**
86_8	6.2515	6.2125	5.8616	5.6176	**5.3872**	5.8939
91_3	6.2286	6.7074	6.6716	6.2085	**6.0462**	6.6700
99_6	5.9128	6.8651	6.3944	6.0276	**5.8786**	6.5146
average	5.3154	5.3621	5.2198	4.9594	4.7854	5.1817

The size of the fingerprints used in the tests were clipped to be of size 352x256 pixels, giving a total file size of 90112 bytes. This was done to get a more suitable image size when divided by two in the downsampling process.

2.2 Compression Results

In the results for the compression algorithms the overhead, i.e. the size of the downsampled orientation image ($\downarrow L$ in figure 4), is not taken into account. If downsampled 16 times, and orientation is quantized to 62 levels the uncompressed orientation image is of size 264 bytes. This represents less than 0.3% of the total image size.

The average number of bits per pixel (bpp) for the best compression methods for the different encoding approaches are displayed in table 2. Lowest number of bits per pixel for each fingerprint displayed in bold. Table 1 shows the abbreviations used. The available implementation of JPEG-LS is provided as a benchmark towards the other algorithms.

From table 2 the gain using ridge scanning compared to horizontal scanning is for zip 3.4% and for recursive Huffman 3.5% in average. Comparing ridge scanning and JPEG-LS the gain for zip (rz) is 2.5% and for recursive Huffman (rrH) 10%, also in average.

The total compression rate is for JPEG-LS 34%, for ridge scanning zip (rz) 35%, and for ridge scanned recursive Huffman (rrH) 40%.

3 Conclusions and Future Works

We have investigated the use of linear symmetry (i.e. local orientations) to determine an alternative way of scanning a fingerprint compared to horizontal (row-by-row) scanning.

The results provides evidence that ridge scanning applied to a lossless compression schemes increases redundancy in data and improves compression by information theoretic techniques. While we could clearly provide evidence for superior performance as compared to JPEG-LS more tests are needed to confirm the advantage of ridge scanning in conjunction with recursive Huffman and Lempel-Ziv compression.

A more specific algorithm needs to be developed to fully take advantage of the structures within the fingerprint. Considering not only the local orientations but also the dominant frequency of the fingerprint, creating separate vectors for different phases of the ridge/valley pattern, will probably further decrease the local variance in data.

Applying different encoding algorithms to the two parts (**r** and **w**) of the segmented image has not been tested, but might be useful as they are uncorrelated.

References

[1] J. Bigun. Recognition of local symmetries in gray value images by harmonic functions. *Ninth International Conference on Pattern Recognition, Rome*, pages 345–347, 1988. 344

[2] J. Bigun and G. H. Granlund. Optimal orientation detection of linear symmetry. In *First International Conference on Computer Vision, ICCV (London)*, pages 433–438, Washington, DC., June 8–11 1987. IEEE Computer Society Press. 344

[3] G. Randers-Pehrson et. al. *PNG (Portable Network Graphics) Specification, Version 1.2*. PNG Development Group, July 1999. 343

[4] Fred Halsall. *Data Communications, Computer Networks and Open Systems*. Addison-Wesley, 1996. 347

[5] D. A. Huffman. A method for the construction of minimum redundancy codes. *Proc. IRE*, 40:1098–1101, 1952. 347

[6] John Håkon Husøy Karl Skretting and Sven Ole Aase. Improved huffman coding using recursive splitting. In *NORSIG-99 conference 9-11 sep.*, 1999. 347

[7] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. Fvc2000: Fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):402–412, March 2002. 348

[8] M Weinberger and G Seroussi. *The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS*. 343, 347

[9] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, IT-23(3), may 1977. 347