A Study of Prediction Measures for Lossy Image Set Compression

Barry Gergel[1], **Anthony Schmieder**[2], **Xiaobo Li**[1], **Howard Cheng**[2]
[1]Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada
[2]Department of Mathematics and Computer Science, University of Lethbridge, Lethbridge, Alberta, Canada

Abstract

The automatic compression strategy proposed by Gergel et al. *is a near-optimal lossy compression scheme for a given collection of images whose inter-image relationships are unknown. Their algorithm uses the root mean square error (RMSE) as a measure of the similarity between two images, in order to predict the compressibility of the difference image. Gergel* et al. *found that RMSE performed well at high compression ratios, but it did not perform as well at lower compression ratios. This paper explores the choice of prediction measure by analyzing the performance of a number of different measures. The experimental results show that entropy performs better than RMSE at lower compression ratios. Furthermore, an adjusted L_1-norm offers nearly the same performance as RMSE at high compression ratios but is easier to compute.*

Keywords: image set compression, difference image.

1. Introduction

The rapid growth in the use of digital images requires new compression strategies for efficient storage. Many modern applications, such as medical imaging centers, store and generate enormously large volumes of images [21]. The automatic lossy compression strategy of Gergel *et al.* [9, 11] allows for the efficient storage of image collections without any prior knowledge of the images. Instead of storing n original images, a subset of $n-1$ difference images are stored. The subset of difference images are selected by studying the compressibility of each of the $\binom{n}{2}$ difference images. The root mean square error (RMSE) is used to predict the compressibility of the difference images. The accuracy of this prediction directly affects the performance of the image set compression scheme. It was assumed that the RMSE between two images is small if and only if the difference image is easy to compress. This paper takes a closer look at this issue and examines four measures, including RMSE, in terms of their prediction power and computation efficiency.

Traditionally, image compression research has focused on reducing the redundancies contained within an image [13]. A large collection of images may contain other redundancies, and a number of different strategies to remove inter-image redundancies have been proposed [2, 3, 4, 15, 16, 18, 17, 20, 22, 23, 24, 25, 26]. Many of these techniques, such as the Centroid method [16, 18], perform well on image sets with particular inter-image relationships, but are less effective on others. It is not clear which method will perform best *a priori*.

The unifying graph theoretical framework proposed by Gergel *et al.* allows for the comparison of all previous techniques that look at the relationship between pairs of images [9, 10, 11]. Their framework led to the discovery of an automatic compression strategy performing no worse than any previous strategy, and often performing better.

The remainder of the paper is structured as follows. Section 2 summarizes the theoretical framework of Gergel *et al.* used for compressing sets of images. Four different measures for predicting compression performance are studied in Section 3. Section 4 concludes the paper with a review of the results of this paper.

2. Graph Theoretical Framework

The graph theoretical framework of Gergel *et al.* facilitates analyzing and comparing compression strategies that reduce inter-image redundancies between pairs of images [9, 10, 11]. The framework defines a complete undirected weighted graph $G=(V,E)$. Each vertex $v \in V$ in the graph represents an image.

An additional two images are added to the image set, and subsequently to the vertex set, to allow the framework to accommodate modeling various proposed compression strategies. These two images are the *zero image*, which is an image composed of pixels with a value of zero, and the *average image*, which is a global average for the input image set.

The edges of the graph define the relationship between the images. Each edge $e \in E$ is assigned a weight value $w(e)$ that is computed from the difference between a pair of images. The weight of an edge is the cost to reconstruct one image given the other image, and it can be approximated using a measure such as the root mean square error (RMSE). This graph structure allows compression strategies to be modeled and compared within an unified framework.

Using this graph representation, all previous compression strategies can be represented as a spanning tree $T = (V, E')$ of the graph G where $E' \subseteq E$. The storage cost for a compression strategy is the total weight of all the edges $e \in E'$ of the spanning tree T. Intuitively, choosing the edges with the smallest weights should yield the best compression for an image set. The strength of this strategy is that a smaller edge weight represents a smaller difference between a pair of images and that this difference takes less space to store. Gergel *et al.* showed that the MST strategy will be no worse than any previous strategy. In some cases, they showed up to a 72% improvement of overall compression for the set when compared to traditional compression [11].

The graph theoretical set compression algorithm consists of three main stages.

1. Compute the complete graph.
2. Compute a minimum spanning tree of the graph.
3. Compress the edges chosen in the MST.

The second stage involves computing an MST of the graph. This can be done using any of the well known algorithms such as Kruskal's algorithm [6]. Once an MST has been found, the difference images for the chosen edges can be compressed using a standard compression algorithm in the third stage.

The first stage is the most time-intensive part of the algorithm. A complete graph has $\binom{n+2}{2}$ edges, which has a computational complexity of $O(n^2)$ operations where n is the number of images. The weight for each edge is the cost to store the difference image. Ultimately, the compressibility of a difference image (the byte size of the compressed image) should be used as the weight of the corresponding edge. However, this results in $\binom{n+2}{2}$ calls to the underlying compression algorithm and it is computationally intensive. We consider four possible prediction measures below. Let $f_1(x,y)$ and $f_2(x,y)$ be two $M \times N$ images with $d(x,y) = f_1(x,y) - f_2(x,y)$ their difference image, and $\bar{d} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} d(x,y)$ is the mean pixel value of the difference image.

Root mean square error (RMSE):

$$rmse = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[d(x,y)]^2}. \tag{1}$$

Standard deviation:

$$\sigma = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[d(x,y)-\bar{d}]^2}. \tag{2}$$

Standard deviation is the same as RMSE except that the mean $\bar{d}$ is removed.

Adjusted L_1-norm:

$$A = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}|d(x,y)-\bar{d}|. \tag{3}$$

The adjusted L_1-norm is the usual L_1-norm with the mean $\bar{d}$ removed. Although computing RMSE and standard deviation was quicker than computing the actual byte costs of a compressed difference image, the adjusted L_1-norm is even easier to compute because there is no need to square and take square roots. This measure is similar to the weighted distance of [7].

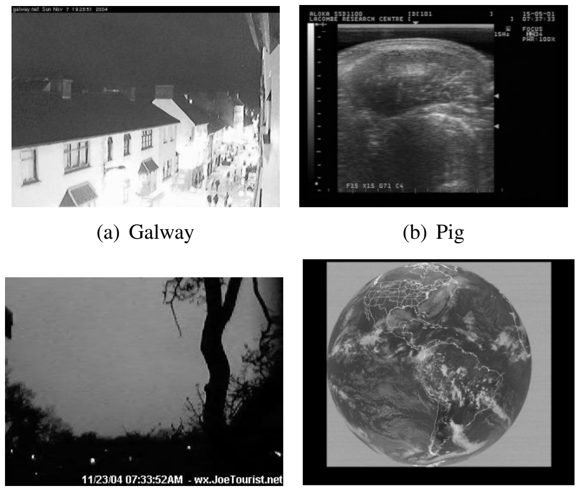
Entropy: let $P(a_1), \ldots, P(a_J)$ be probabilities of the symbols $a_1, \ldots, a_J$ occurring in an image. The entropy is defined as

$$H = -\sum_{j=1}^{J} P(a_j)\log_2 P(a_j). \tag{4}$$

When a_j's are the gray level of individual pixels, H is the first-order entropy and shows how much coding redundancy can be reduced from an image. More generally, blocks of k pixels can be used to obtain a k-th order entropy. When $k > 1$, inter-pixel redundancies can be detected as well. While the first-order entropy is easy to compute, higher-order entropies are somewhat more difficult. In our experiments, we use fourth-order entropy with 2×2 pixel blocks. This provides a balance between performance and computational complexity.

3. Experimental Results

Five image sets were used in the experiments. Each test set was composed of twelve randomly chosen images from each of the larger sets used by Gergel *et*



(a) Galway (b) Pig

(c) Joe (d) GOES

Figure 1. Typical images from each set.

al. [9, 11] and Nielsen *et al.* [22, 23]. The smaller subsets were chosen to speed up testing. Figure 1 shows a typical image from the first four image sets. The Galway set contains webcam images from a street in Galway City, Ireland [8]. The Pig set is composed of ultrasound images of pig rib cages. The Joe set is another webcam image set taken from a camera directed at a beach in Victoria, British Columbia [14]. Satellite images from the GOES project [12] make up the GOES set. The final set is the Combination set and it was composed of 6 images from the Galway set and 6 images from the Pig set. All the images were 8-bit gray scale images.

One method to evaluate the performance of a measure is to calculate the correlation coefficient between the value given by the measure and the actual compression performance. Since we are performing lossy compression, it is not sufficient to measure the performance simply by examining the resulting file size. Instead, a compression ratio is fixed and the RMSE between the original and decompressed images is used. An image is easy to compress if this error is small, and hard to compress if this error is large.

The correlation coefficient is defined as

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}. \quad (5)$$

Two values are computed for each of the n difference images. The value of X_i is the computed measure of the difference image. The value of Y_i is the RMSE between the original difference image and the decompressed difference image using a compression algorithm such as JPEG2000 at a fixed compression ratio. If $r \approx 0$ then the chosen measure is not good at predicting the compression performance. Conversely, if $r \approx 1$ then the chosen measure predicts the compression performance well.

The experiments computed the correlation coefficients for the five image sets using two different compression algorithms; Jasper [1], which is an implementation of the JPEG2000 specification [5], and the wavelet packet tools from Meyer *et al.* [19]. Each measure was tested at six compression ratios $\{3:1, 10:1, 20:1, 40:1, 60:1, 100:1\}$. The resulting correlation coefficients are shown in Table 1 through Table 5.

3.1. RMSE Results

The first measure evaluated was RMSE, and this served as the base case for comparison of the other tested measures. As the compression ratio increased, the correlation coefficient for each image set increases. Also, the wavelet packet algorithm outperformed the JPEG2000 algorithm. These results agree with the previous works [9, 11]. The GOES set was difficult to predict by RMSE as indicated by the low correlation coefficients for both compression algorithms.

3.2. Standard Deviation Results

Overall, standard deviation slightly outperformed RMSE. It was better at higher compression ratios, but standard deviation also performed poorly at lower compression ratios. Although standard deviation nearly doubled the performance of RMSE on the GOES set, the correlation coefficients were still very low.

3.3. Adjusted L_1-norm Results

The results were very similar to the RMSE results. Therefore, the adjusted L_1-norm could be used as a substitute for RMSE, which would reduce the time required to compute the edge weights.

3.4. Entropy Results

The correlation coefficients showed that fourth-order entropy performed better at lower compression ratios. As the compression ratio increased, the other measures tested performed better than entropy. This is because the compression is close to lossless at the lower compression ratios. At higher compression ratio it is not sufficient to consider entropy alone. The performance of entropy on the GOES set was better than the other measure, but again, the overall performance is poor for this image set.

Table 1. The correlation coefficients for the Galway image set.

Ratio	JPEG2000				Wavelet Packets			
	RMSE	σ	Adj. L_1	Entropy	RMSE	σ	Adj. L_1	Entropy
3:1	0.157	0.120	0.149	**0.799**	0.093	0.058	0.085	**0.865**
10:1	0.074	0.039	0.067	**0.868**	0.197	0.231	0.204	**0.959**
20:1	0.103	0.137	0.109	**0.930**	0.424	0.455	0.430	**0.973**
40:1	0.421	0.452	0.425	**0.957**	0.731	0.753	0.733	**0.874**
60:1	0.673	0.697	0.675	**0.902**	0.864	**0.880**	0.865	0.762
100:1	0.854	**0.870**	0.855	0.773	0.944	**0.954**	0.945	0.640

Table 2. The correlation coefficients for the Pig image set.

Ratio	JPEG2000				Wavelet Packets			
	RMSE	σ	Adj. L_1	Entropy	RMSE	σ	Adj. L_1	Entropy
3:1	0.180	0.141	0.181	**0.797**	0.388	0.424	0.344	**0.703**
10:1	0.479	0.533	0.448	**0.629**	**0.786**	0.774	0.764	0.039
20:1	0.611	**0.656**	0.586	0.481	0.762	**0.791**	0.719	0.278
40:1	0.591	**0.639**	0.569	0.464	0.734	**0.769**	0.693	0.314
60:1	0.590	**0.639**	0.563	0.453	0.750	**0.785**	0.708	0.284
100:1	0.682	**0.723**	0.640	0.358	0.729	**0.765**	0.676	0.268

Table 3. The correlation coefficients for the Joe image set.

Ratio	JPEG2000				Wavelet Packets			
	RMSE	σ	Adj. L_1	Entropy	RMSE	σ	Adj. L_1	Entropy
3:1	0.251	0.266	0.249	**0.792**	0.261	0.332	0.288	**0.911**
10:1	0.486	0.552	0.506	**0.814**	0.596	0.668	0.622	**0.731**
20:1	0.706	**0.751**	0.715	0.610	0.785	**0.822**	0.793	0.498
40:1	**0.889**	0.883	0.876	0.289	0.882	**0.888**	0.877	0.282
60:1	**0.911**	0.889	0.890	0.161	**0.905**	0.894	0.890	0.199
100:1	**0.924**	0.893	0.900	0.107	**0.901**	0.873	0.876	0.115

Table 4. The correlation coefficients for the GOES image set.

Ratio	JPEG2000				Wavelet Packets			
	RMSE	σ	Adj. L_1	Entropy	RMSE	σ	Adj. L_1	Entropy
3:1	0.060	0.331	0.139	**0.629**	0.106	0.257	0.017	**0.791**
10:1	0.124	0.192	0.043	**0.740**	0.052	0.241	0.029	**0.699**
20:1	0.079	0.201	0.002	**0.677**	0.003	0.258	0.078	**0.632**
40:1	0.008	0.222	0.062	**0.574**	0.087	0.288	0.154	**0.530**
60:1	0.043	0.240	0.108	**0.508**	0.116	0.298	0.180	**0.486**
100:1	0.121	0.269	0.179	**0.426**	0.187	0.324	0.243	**0.412**

Table 5. The correlation coefficients for the Combination image set.

Ratio	JPEG2000				Wavelet Packets			
	RMSE	σ	Adj. L_1	Entropy	RMSE	σ	Adj. L_1	Entropy
3:1	0.694	0.771	0.758	**0.857**	0.720	0.778	0.776	**0.916**
10:1	0.750	0.806	0.802	**0.930**	0.796	0.849	0.845	**0.919**
20:1	0.804	0.869	0.860	**0.899**	0.843	**0.901**	0.892	0.883
40:1	0.843	**0.912**	0.898	0.841	0.881	**0.939**	0.928	0.818
60:1	0.865	**0.932**	0.917	0.806	0.901	**0.953**	0.942	0.785
100:1	0.895	**0.954**	0.940	0.781	0.921	**0.965**	0.955	0.742

4. Conclusion

The automatic strategy of Gergel *et al.* is an effective lossy compression scheme to improve the storage requirements for collections of images. RMSE is a common measure used to predict compression performance, but their results indicated that RMSE did not perform well at low compression. In this paper, the performance of four measures was compared. A number of observations can be made regarding the results of the experiments.

The performance of RMSE at high compression ratios presented by Gergel *et al.* was validated by correlation results given in this paper. At higher compression ratios, RMSE performed well, but it did poorly at lower ratios. The standard deviation measure slightly outperformed RMSE at high compression ratios. On the other hand, the adjusted L_1-norm performed very close to RMSE and it is easier to compute. Therefore, the adjusted L_1-norm makes a good candidate for implementation where high compression ratios are required to compress an image set.

The results for the fourth-order entropy indicate that it performs much better than RMSE at lower compression ratios. Thus, a modified version of the compression strategy may pick the appropriate measure based upon the compression ratio.

The performance of all the measures on the GOES set was poor. This set is known to have images that are not very similar to one another, and its qualities need to be studied further to improve the understanding of difference images and how to compress image sets.

Acknowledgements

We would like to gratefully acknowledge Dr. Alan Tong of Lacombe Research Centre, Agriculture and Agri-Food Canada for making the Ultrasound images and data available to this study.

References

[1] M. Adams. JasPer project. http://www.ece.uvic.ca/ mdadams/jasper/.

[2] S. Ait-Aoudia and A. Gabis. A comparison of set redundancy compression techniques. *EURASIP Journal on Applied Signal Processing*, 2006:1–13, 2006.

[3] J.P. Cheiney and A. Touir. Fi-quadtree: A new data structure for content-oriented retrieval and fuzzy search. In *Proceedings of the International Symposium on Large /Spatial Databases*, pages 23–32, 1991.

[4] C.-P. Chen, C.-S. Chen, K.-L. Chung, H.-I. Lu, and G. Tang. Image set compression through minimal-cost prediction structures. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1289–1292, 2004.

[5] The JPEG Committee. JPEG 2000. http://www.jpeg.org/jpeg2000/index.html.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001.

[7] A. M. Eskicioglu and P. S. Fisher. Image quality measures and their performance. *IEEE Transactions on Communications*, 43(12):2959–2965, December 1995.

[8] Galway.net. http://www.galway.net/webcam/images.shtml.

[9] B. Gergel. Automatic compression for image sets using a graph theoretical framework. Master's thesis, University of Lethbridge, March 2007.

[10] B. Gergel, H. Cheng, and X. Li. A unified framework for lossless image set compression. In *Proceedings of the Data Compression Conference*, page 448, 2006.

[11] B. Gergel, H. Cheng, C. Nielsen, and X. Li. A unified framework for image set compression. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition*, pages 417–423, 2006.

[12] GOES project science. http://rsd.gsfc.nasa.gov/goes.

[13] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, 2007.

[14] JoeTourist Infosystems. http://www.JoeTourist.net.

[15] G. Jomier, M. Manouvrier, and M. Rukoz. Storage and management of similar images. *Journal of the Brazilian Computer Society*, 6(3):13–25, 2000.

[16] K. Karadimitriou. *Set redundancy, the enhanced compression model, and methods for compressing sets of similar images*. PhD thesis, Louisiana State University, 1996.

[17] K. Karadimitriou and J. M. Tyler. Min-max compression methods for medical image databases. *ACM Special Interest Group on Management of Data*, 26(1):47–52, 1997.

[18] K. Karadimitriou and J. M. Tyler. The centroid method for compressing sets of similar images. *Pattern Recognition Letters*, 19(7):585–593, 1998.

[19] F. Meyer, A. Averbuch, and J. Strömberg. Fast adaptive wavelet packet image compression. *IEEE Transactions on Image Processing*, 9(5):792–800, 2000.

[20] Y. S. Musatenko and V. N. Kurashov. Correlated image set compression system based on new fast efficient algorithm of Karhunen-Loeve transform. In C.-C. J. Kuo, S.-F. Chang, and S. Panchanathan, editors, *Proceedings of The International Society for Optical Engineering: Multimedia Storage and Archiving Systems III*, pages 518–529, October 1998.

[21] H. Nagata and H. Tanaka. Estimate of total volume of medical data in a year in Japan. In *Proceedings of the APAMI and CJKMI-KOSMI Conference*, pages 117–121, 2003.

[22] C. Nielsen and X. Li. MST for lossy compression on image sets. In *Proceedings of the Data Compression Conference*, page 463, 2006.

[23] C. Nielsen, X. Li, and K. Abma. Methods of grouping similar images for compression coding. In *Proceedings of the International Conference on Computer Vision (WCAC'05)*, pages 93–99, Las Vegas, June 20–23, 2005.

[24] M. Vassilakopoulos and Y. Manolopoulos. Dynamic inverted quadtree: a structure for pictorial databases. *Information Systems*, 20(6):483–500, 1995.

[25] M. Vassilakopoulos, Y. Manolopoulos, and N. Economou. Overlapping quadtrees for the representation of similar images. *Image and Vision Computing*, 11(5):257–262, 1993.

[26] J. Wang and H. Yan. Form image compression using template extraction and matching. In *Proceedings of Visual Information Processing*, December 2000.