Hierarchical Minimum Spanning Trees for Lossy Image Set Compression

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

Abstract

A number of minimum spanning tree algorithms have been proposed for lossy compression of image sets. In these algorithms, a complete graph is constructed from the entire image set and possibly an average image, and a minimum spanning tree is used to determine which difference images to encode. In this paper, we propose a hierarchical minimum spanning tree algorithm in which the minimum spanning tree algorithm is first applied to clusters of similar images and then it is applied to the average images of the clusters. It is shown that the new algorithm outperforms the previous image set compression algorithms for image sets which are not very similar, especially at lower bitrates. Furthermore, the computational requirement for a hierarchical minimum spanning tree is significantly lower than the previous minimum spanning tree algorithms when the cost of clustering can be neglected.

Keywords: image set compression, clustering, minimum spanning tree.

1. Introduction

Traditional image compression algorithms for individual images, such as predictive coding and transform coding, have been shown to effectively reduce coding, inter-pixel, and psycho visual redundancy within an image [5]. Image sets, however, may contain inter-image redundancy, or "set redundancy" [6], which is not reduced by these algorithms. Some work has been done to address this issue. The centroid [6], MST [1, 10], and MST_a [3, 4] algorithms have been shown to reduce inter-image redundancy in sets of similar images.

In cases where images in a set form multiple clusters of similar images, there is potential for improvement. With the centroid and MST_a algorithms, only one average image is calculated for the entire set of images. As the number of distinct clusters in a set increases, the average image becomes "less similar" to any single image in the set, and is therefore a less effective predictor for the images in the set. This can negatively impact compression performance of the image set compression schemes.

The hierarchical approach presented here combines the MST_a algorithm of Gergel *et al.* [3, 4] with the clustering algorithm of Nielsen *et al.* [11] by partitioning a set of images into clusters and performing the MST_a scheme on each of the clusters. The MST_a scheme is then applied to a set containing the average images of each cluster. In some cases, the clustering may be done without any computation using *a priori* knowledge. For example, an image set containing X-ray images may be clustered based on gender, age, and body parts. A set of images from a webcam can be clustered based on the time of the day each image is taken.

Compression performance of the hierarchical algorithm is examined and compared with the compression performance of the traditional, centroid, MST, and MST_a compression schemes. It is shown that the new hierarchical algorithm outperforms the previous algorithms for image sets which have distinct clusters especially at lower bitrates.

2. Related Work

Karadimitriou and Tyler proposed the centroid and min-max "set mapping" schemes to reduce "set redundancy" for lossless compression [6, 7]. The centroid scheme involves computing an average image for a set of similar images, calculating the difference between the average image and each image in the set, and coding the average image and the difference images. In the min-max scheme, a minimum image and a maximum image are created from the minimum and maximum pixel values across all images. Several methods may be used to predict each original image from the minimum and maximum images. The minimum image, maximum image, and the prediction error for each image are coded. This algorithm gave significant improvement in compression ratios compared to compressing individual images. However, the images in the set must be quite similar if the centroid and min-max algorithms are to perform well, and image sets that contain dissimilar images are not considered. To ensure similarity of the test images, clusters of ten images were selected from a larger set using a genetic algorithm, and each cluster was compressed independently. This algorithm runs quickly and produces clusters that are quite similar. The clustering algorithm, although very effective for experimental purposes, is not practical in all environments. It requires the desired number of images in the cluster as input, which may be impractical to determine for large or unpredictable sets of images.

Nielsen *et al.* proposed a clustering strategy that is adaptive to image sets containing dissimilar images [11]. In their approach, the root mean square error (RMSE) between images in the set and the average image is used to partition the set into clusters of similar images. Each cluster is compressed independently using the centroid scheme. JPEG2000 (lossless and lossy) [2] is used to compress the average and difference images. Their results were compared to “traditional” JPEG2000, which refers to using JPEG2000 to compress each image individually, and showed a 13% to 25% improvement over traditional JPEG2000. Compression performance is clearly improved, but their experiments did not compare the clustered centroid scheme with the centroid scheme on the entire image set, and did not consider other set mapping strategies.

The minimum spanning tree (MST) set mapping strategy, proposed by Nielsen and Li, is based on a graph data structure [10]. A complete weighted graph is constructed, using images as the vertices and the RMSE between adjacent images as the edge weights. An MST for the graph is calculated, and one image is chosen as the root. The root image and difference images represented by the edges with the lowest total cost are encoded using lossy JPEG2000 [2]. The results of these experiments showed a clear improvement in average distortion (RMSE) when using the MST scheme over compressing each image individually, especially at lower bitrates. These experiments focused on sets of similar images, and did not examine performance on sets containing dissimilar images. Other authors have applied a similar strategy to specific applications such as object movies [1], multiview video coding [9], multispectral images [12], and map images [8]. However, the image sets in these applications are known to be similar and the image sets are usually small.

Gergel *et al.* generalized this work with the MST_a scheme [3, 4]. An MST is computed on a complete graph that includes a zero image and an average image, using RMSE as edge weight. The MST_a scheme is a unified framework that adaptively chooses the best scheme among the traditional, centroid, MST, and other schemes. Gergel *et al.* compared lossy and lossless compression results between the traditional, centroid, MST, and MST_a schemes. The MST_a scheme is shown to be highly effective, outperforming the other schemes in many cases. For image sets which are very similar, the MST_a strategy makes use of the average image to arrive at a strategy very close to the centroid strategy. On the other hand, for image sets which have clusters of similar images, the MST_a strategy essentially chooses to compress each image independently because the average image for the entire set is not a good predictor of the images in the set. Furthermore, it may not be practical to construct the complete graph for a large image set.

3. Approach

The existing set mapping strategies have been shown to be effective on sets of similar images, but the images may not be similar in all cases. In the hierarchical MST_a scheme, we partition the image set into clusters of similar images, and apply the MST_a algorithm to each cluster. Since the average image for each cluster should be very similar to all images in the cluster, it is a good predictor for images in the cluster. This should produce difference images with a small range of pixel values that will compress well. If the image set contains tight clusters, the increased compression performance for the difference images offsets the added cost of storing multiple average images.

Figure 1 shows one situation commonly encountered in the compression of image sets. There are two clusters of very similar images in this set. If r is the “radius” of the cluster as measured from the cluster average, then the magnitude of r is an indication of how similar the images are inside the cluster. When r is small in relation to the inter-cluster distance d, using a global average image does not lead to smaller difference images between the average and the remaining images. On the other hand, the difference between each image and the average of its cluster is small, so that the overall compression performance is improved when cluster averages are used. In addition, the edges between images in different clusters have weight at least d while the edges between an image and its cluster average is bounded by r. Therefore, the inter-cluster edges will never be used in an MST of the complete graph, and it is possible to improve the complexity of the graph

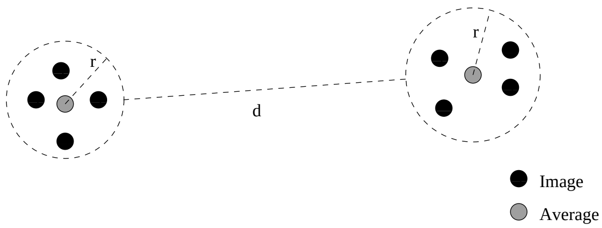

Figure 1. Image set containing two distinct clusters.

construction and MST computation by ignoring inter-cluster edges.

3.1. Graph Theory and MST$_a$

Gergel *et al.* described the MST$_a$ set mapping scheme as follows [3, 4]. Let $S = \{I_1, I_2, \ldots, I_n\}$ represent a set of n images of identical dimensions, and $I_k(i, j)$ represent the pixel value at location (i, j) in I_k. Two additional images are defined: the zero image $I_{n+1} = I_z$ where $I_z(i, j) = 0$ for all values of (i, j), and the average image $I_{n+2} = I_a$ where

$$I_a(i,j) = \frac{1}{n}\sum_{k=1}^{n} I_k(i,j).$$

The average and zero images are added to S to create a new set as follows:

$$S_a = S \cup \{I_z\} \cup \{I_a\}.$$

Next, a complete, undirected, and weighted graph $G = (V, E)$ is defined from S_a, where $V = \{I_i | I_i \in S_a\}$ and $E = \{(I_i, I_j) | I_i, I_j \in V, i < j\}$. The weight for each edge is defined as $w(I_i, I_j)$ where w is a function that measures the cost to reconstruct I_j assuming I_i is known. For this paper, RMSE is used as the weight function and it is symmetric.

The minimum spanning tree T of G is calculated, and the difference images represented by the edges in T are coded.

3.2. MST$_a$ and Clustering: The Hierarchical MST$_a$

In this work, we add clustering to the MST$_a$ scheme to form a hierarchical MST$_a$ (HMST$_a$). The set of images S is partitioned into k clusters $S_1 \cup S_2 \cup \ldots \cup S_k = S$ where $S_i \cap S_j = \emptyset$ for $i \neq j$. Each cluster S_i contains n_i images, such that $S_i = \{I_{i,1}, I_{i,2}, \ldots, I_{i,n_i}\}$ and $n_1 + n_2 + \ldots + n_k = n$. The MST$_a$ algorithm is applied to each cluster. In the first step of the HMST$_a$ algorithm, the average images are computed and the zero image and average image are added to each cluster S_i to form the set $S_{a,i}$. The average image for $S_{a,i}$ is

$$I_{i,a}(i,j) = \frac{1}{n_i}\sum_{l=1}^{n_i} I_{i,l}(i,j).$$

Next, a new cluster S_A of the average images of all clusters is created as $S_A = \{I_{1,a}, I_{2,a}, \ldots, I_{k,a}\}$. The average image is computed and the zero image and average image are added to S_A to form $S_{a,A}$. The average image for $S_{a,A}$ is

$$I_{A,a}(i,j) = \frac{1}{k}\sum_{m=1}^{k} I_{m,a}(i,j).$$

An MST is computed on the complete graph constructed from each cluster $S_{a,i}$ as well as $S_{a,A}$. The difference images represented by the edges in the MSTs are coded. Notice that the resulting edges may not be a spanning tree for the complete graph constructed from the image set $\left(\cup_{i=1}^{k} S_{a,i}\right) \cup S_{a,A}$ because there may be cycles involving the average images $I_{i,a}$ and the zero image. These cycles are broken by removing edges connecting $I_{i,a}$ to obtain a spanning tree. The HMST$_a$ scheme can be viewed as a generalization of the MST$_a$ scheme as the two are identical when every cluster contains a single image.

3.3. The Clustering Algorithm

For these experiments, we implemented the clustering algorithm described by Nielsen and Li [11]. Their algorithm partitions images into clusters based on both the percentage of pixels outside of the interval $[-127, 127]$ in the differences between the images and the average image, and the RMSE between the images and the average image. Let $\Delta(I_1, I_2)$ be the RMSE between images I_1 and I_2, $I_1 - I_2$ be the difference between I_1 and I_2, and $\%(I)$ be the percentage of pixels in image I that are in the interval $[-127, 127]$. A percentage threshold ϕ is chosen. For each image $I \in S_k$, if $\%(I_{k,a} - I) < \phi$, then $\Delta(I_{k,a}, I)$ is computed. The image I in cluster S_k with the highest $\Delta(I_{k,a}, I)$ is moved to cluster S_{k+1}. $I_{k,a}$ is then recalculated, and the comparison is repeated for images remaining in S_k. These steps are repeated until a pass is made through S_k where no image is removed. This process is repeated for all clusters. The clustering algorithm described is presented in Algorithm 1 [11].

4. Experimental Results

It may or may not be obvious that a set of images indeed contains images that are truly similar enough to each other. In order to effectively remove or reduce set

Algorithm 1 Nielsen and Li's clustering algorithm.

$S_1 \leftarrow$ set of all images, $k \leftarrow 1$
repeat
  repeat
    compute $I_{k,a}$
    let $I \in S_k$ be such that
      $\%(I_{k,a} - I) < \phi$ and $\Delta(I_{k,a}, I)$ is maximum
    move I to S_{k+1}, creating S_{k+1} if necessary
  until no image is removed
  $k \leftarrow k+1$
until there are no more clusters

redundancy in an image set, sometimes it is necessary to analyze the image set and determine whether there exist one or more clusters of truly-similar images. Our experiments include two image sets—the Combination set and the Joe image set [3, 4]. The Combination set is made of two distinctly different image sets. The Joe set contains images from a single original source, so clustering is necessary in dividing this set into a number of clusters.

Results for the Combination set can be seen in Figure 2, which plots bitrate against average distortion. For these experiments, we coded the image sets using each set mapping scheme at varying bitrates, and measured the distortion of the reconstructed images using RMSE for each image in the set at each bitrate. Plotted values represent average distortion across all images at a specific bitrate, therefore, a curve that is lower and to the left represents better compression performance. For both image sets, the HMST$_a$ scheme outperforms the other set mapping strategies.

The Combination image set contains 29 images from the Pig image set and 28 images from the Galway image set [3, 4]. Clearly, the images form two tight, distinct clusters (the images in a single cluster are quite similar to each other, but quite dissimilar to the images in the other cluster. See Figures 3 and 4).

With the other set mapping schemes such as centroid and MST$_a$, only one average is formed for the entire set. In the case of the Combination set, the average image contains elements of both the Pig and Galway images, and is not a good predictor for any image in the set. Figure 5 shows the average image from the application of MST$_a$ on the Combination set. Obviously, this "global" average image gives a poor prediction for the Pig images and the Galway images.

With the HMST$_a$ strategy, the two clusters are identified and separated, and an average image is calculated for each cluster. The average images are much better predictors for the images in the clusters, because they only contain elements from a set of similar images. As

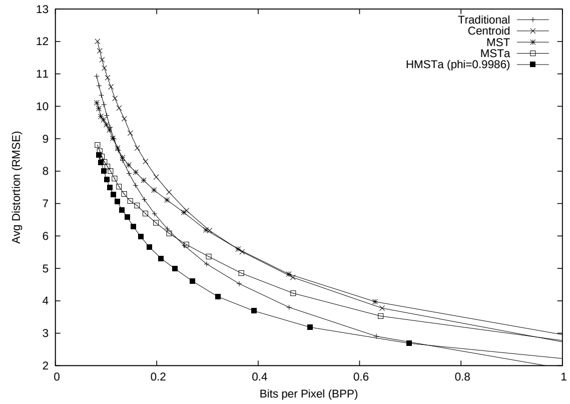

Figure 2. Results for the Combination set.

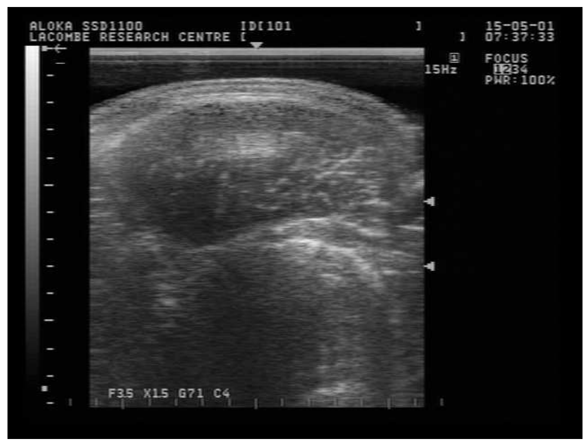

Figure 3. Sample image from the Pig set.

Figure 4. Sample image from the Galway set.

Figure 5. Average image from MST$_a$ on the Combination set.

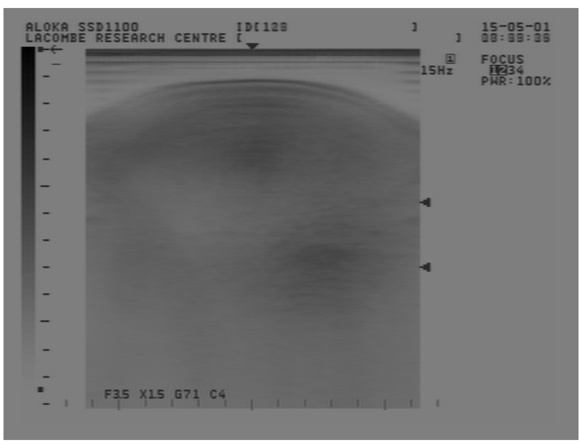

Figure 6. Sample average image from HMST$_a$ on the Combination set.

a result, the difference images are easier to compress. See Figure 6 for a sample cluster average image from the HMST$_a$ algorithm.

Results for the Joe set can be seen in Figure 7. The Joe image set contains time lapsed photographs of an outdoor scene, captured from a webcam. Since the images are taken minutes or hours apart from each other, video compression algorithms such as MPEG cannot be applied directly here, treating the images as consecutive frames. The images used in these experiments were captured at different times throughout the day, and in different weather conditions, so the sky portion of the images is significantly different among the images (see Figure 8 for sample images). The drastic variance in the sky portion of the images has a strong effect on the average image, and as a result, the average image is a poor predictor for the images, and the difference images contain a wide range of pixel values. The average

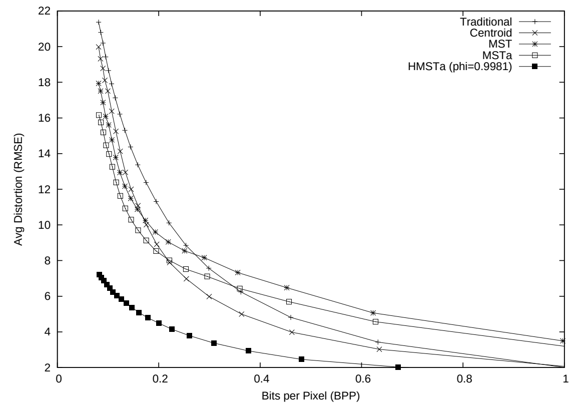

Figure 7. Results for the Joe set.

Figure 8. Sample images from the Joe set.

image of the entire set is shown in Figure 9, and a sample difference image from the MST$_a$ scheme is shown in Figure 10. The HMST$_a$ strategy performs well on the Joe set for reasons similar to why it performs well on the Combination set. Images in similar clusters show similar sky conditions. This means that the average image for each cluster will be a better predictor for images in that cluster, so that difference images are easier to compress. Figure 11 shows a sample difference image from the application of HMST$_a$ on the Joe image set. It is clear that the pixel values in Figure 11 are significantly smaller than those in Figure 10. Furthermore, the HMST$_a$ strategy significantly outperforms a more sequential prediction scheme that would have been chosen by a video compression algorithm.

5. Complexity Analysis

The complexity analysis is done in terms of operations per pixel in a single image. For example, if an operation is performed once for each pixel on n images, it would be considered to be performed n times per pixel in a single image. We will only examine the complexity for the graph construction as the MST calculations are negligible. Also, we will not examine the cost of clustering because it depends on the chosen clustering algorithm. In some cases, clustering may be “free” if

Figure 9. Average image from MST_a on the Joe set.

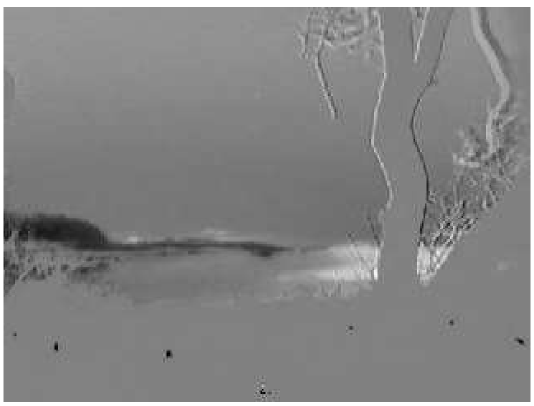

Figure 10. Sample difference image from MST_a on the Joe set.

Figure 11. Sample difference image from $HMST_a$ on the Joe set.

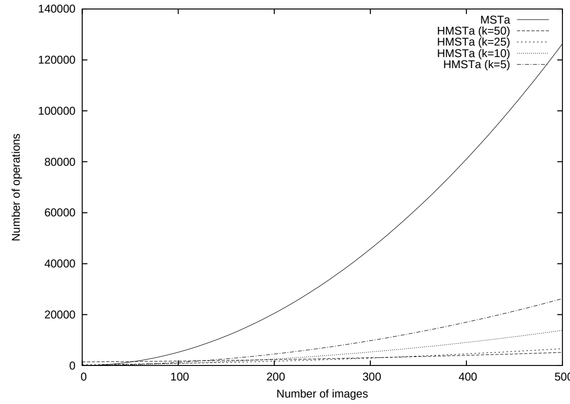

Figure 12. Graph construction operations in MST_a and $HMST_a$ for varying number of clusters.

the image set can be clustered with *a priori* knowledge.

For the MST_a scheme, the average image must be calculated, followed by the complete graph on all images plus the zero and average images. Therefore, the number of operations performed to construct the complete graph is

$$T_{MST_a} = n + \binom{n+2}{2} = \frac{n^2 + 5n + 2}{2}.$$

On the other hand, the number of operations performed by $HMST_a$ depends on the number of clusters and the size of each cluster. For illustration, we assume that the image set is partitioned into k clusters each of size n/k. In that case, the cost for the graph construction in $HMST_a$ is

$$\begin{aligned} T_{HMST_a} &= k\left(\frac{n}{k} + \binom{\frac{n}{k}+2}{2}\right) + k + \binom{k+2}{2} \\ &= \frac{n^2 + 5nk + k^3 + 7k^2 + 2k}{2k}. \end{aligned}$$

Figure 12 shows that the graph construction for the $HMST_a$ algorithm is more efficient than that of the MST_a algorithm on the unclustered set especially for large image sets.

6. Conclusion

The compression performance of the $HMST_a$ strategy is better than that of the traditional, centroid, MST, and MST_a strategies on image sets that contain multiple tight clusters. This is at the computational expense of running a clustering algorithm on the images prior to applying the $HMST_a$ algorithm. In some cases, this computational expense may be avoidable if some *a priori* knowledge about the images is available.

Future experiments will be conducted to determine how well $HMST_a$ performs for image sets that do not contain tight clusters, and to gauge the impact of using wavelet packet coding rather than JPEG2000 to compress the difference images. Future work may also include the application of different clustering algorithms before the application of the $HMST_a$ scheme.

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

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

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

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

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

[6] K. Karadimitriou. *Set Redundancy, the Enhanced Compression Model, and Methods for Compressing Sets of Similar Images*. PhD thesis, Louisiana State University, 1996.

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

[8] P. Kopylov and P. Franti. Compression of map images by multilayer context tree modeling. *IEEE Transactions on Image Processing*, 14(1):1–11, January 2005.

[9] D.-X. Li, W. Zheng, X.-H. Xie, and M. Zhang. Optimising inter-view prediction structure for multiview video coding with minimum spanning tree. *Electronics Letters*, 43(23):1269–1271, November 2007.

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

[11] C. Nielsen, X. Li, and K. Abma. Methods of grouping similar images for compression coding. In *Proceedings of The 2005 International Conference on Computer Vision*, pages 93–99. CSREA Press, 2005.

[12] S. R. Tate. Band ordering in lossless compression of multispectral images. *IEEE Transactions on Computers*, 46(4):477–483, April 1997.