# A Unified Framework for Image Set Compression

Barry Gergel    Howard Cheng
*Department of Mathematics & Computer Science*
*University of Lethbridge*
*Lethbridge, Alberta, T1K 3M4, Canada*
*gergel@cs.uleth.ca, cheng@cs.uleth.ca*

Clinton Nielsen    Xiaobo Li
*Department of Computing Science*
*University of Alberta*
*Edmonton, Alberta, T6G 2E8, Canada*
*clintonn@ualberta.ca, li@cs.ualberta.ca*

## Abstract

*While the compression of individual images have been studied extensively, there have been fewer studies on the problem of compressing image sets. A number of schemes have been proposed to compress an image set by taking advantage of the inter-image redundancy between pairs of images. In this paper, we present a unified graph-theoretic framework that includes all such previous schemes. A minimum spanning tree gives the optimal compression method for lossless compression. We also show that while this framework does not necessarily give the optimal compression scheme for lossy compression, we can provide performance guarantees relative to the optimal scheme. Our experiments show that the new minimum spanning tree method performs better than the previous schemes, especially when the image sets are not well suited for any of the previously proposed schemes.*

## 1. Introduction

As the availability and use of digital images increase, the efficient storage of images becomes an important area of research. Traditionally, each image in a set is compressed individually, taking advantage of the coding, interpixel, and psycho-visual redundancies existing *within* the image [8]. In the related area of video compression, a video sequence is decomposed into individual frames. Video compression algorithms take advantage of redundancy existing among consecutive frames as well as the redundancy existing within each frame.

Many applications involve the storage of a large number of *similar* images. This is in sharp contrast to personal photo-album type databases where the images are often drastically different. All the images in the set have identical dimensions and the same color and grayscale range. For example, a medical database may contain a large number of X-ray images of the same body part; a database of satellite images of the same viewing angle of the earth may possess similar characteristics. Unlike video compression, these applications use image sets whose inter-image relationships are unknown. Webcam image databases may also contain a tremendous number of images of similar scenes. Each image is taken minutes or hours apart, instead of 1/30 of a second, thus video coding (e.g., MPEG) is not always suitable.

In some applications such as medical imaging, compressed images must be identical to the original images, therefore lossless compression must be used. On the other hand, lossy compression can be tolerated in other applications, such as agriculture imaging.

Compared with traditional image compression, the compression of a set of images has received relatively little attention from researchers [5, 7, 9, 10, 11, 12, 13, 14]. These earlier schemes have only been effective on image sets with certain properties, and it is not clear which scheme is best *a priori*.

In [5, 12], a graph is constructed from the images to represent inter-image redundancy, and a minimum spanning tree (MST) is computed to decide which difference images to compress with a lossy compression algorithm. In this case, however, it is not obvious that the resulting structure is optimal. An overlooked fact is that the coding error in the lossy compression process perturbs the graph, so that the MST computed from the original graph may not be an MST for the modified graph actually used. In fact, if the errors introduced by the lossy compression process is large, the MST for the modified graph may be very different. A further complication is that the errors introduced depend on which edges are used in the compression process, and the edges are in turn chosen based on the graph. This cyclic dependency makes it difficult to guarantee optimality.

The main contributions provided in this paper are the unification of previous schemes under one framework, as well as a theoretical basis for the MST scheme for lossy compression. We represent an image set as a graph and compute its MST to decide which images and differences to encode. By modifying the underlying graph, the Centroid scheme [9, 10] and the previous MST schemes [5, 12] can both be represented as a spanning tree in our graph. For lossless compression, our scheme is *guaranteed* to be no worse than these previous schemes, regardless of the properties of the image sets. For lossy compression, we show that the difference between the computed MST and the actual MST depends on the amount of error introduced by lossy compression. This gives us a performance guarantee relative to the optimal scheme. In particular, this implies that when the errors are small, the MST scheme indeed gives a good approximation to the optimal compression scheme. This also provides a first step in understanding the trade-off between coding optimality and image quality for lossy set compression.

The paper is organized as follows. We review the previous schemes in Section 2. In Section 3, we introduce our framework for lossless compression. The framework is then extended to lossy compression in Section 4. In Section 5, we show how previous schemes can be modelled within our framework. We show our experimental results in Section 6. Our concluding remarks follow in Section 7.

## 2. Previous Work

Karadimitriou and Tyler investigated a lossless scheme, called the Centroid scheme, of compressing a set of medical images around an average image [9, 10]. In this scheme, an average image of the set is computed; only the average and the difference between the average and each image are encoded. A similar idea compresses the difference between each image and a template [14]. Karadimitriou and Tyler also showed that a clustering algorithm can be applied to partition the set into clusters of similar images, and each cluster is compressed independently. This idea was examined by Nielsen *et al.* [13], who used the global average approach and proposed a lossy compression algorithm based on different clustering criteria. This work was mainly focused on parameter selection and trade-off analysis. Their experiments showed up to 25% performance improvement over JPEG2000. Note that an average image is introduced for each cluster, so the overhead increases with the number of clusters. As a result, this scheme is suitable when the image set contains few tight clusters. Furthermore, redundancy among the clusters is not exploited as they are encoded independently.

Another way to exploit inter-image redundancy was proposed by Chen *et al.* [5] and Nielsen *et al.* [12]. In this scheme, an MST is computed from a complete graph. The vertices of the graph represent the images, and the weight of each edge is a measure of the cost of encoding one image given the other image of the edge. Chen *et al.* [5] used this scheme to link different views of 3D objects to represent the prediction relationship among the views. The edge weight used is based on motion estimation and compensation. On the other hand, Nielsen *et al.* [12] used the root-mean-squared error (RMSE) between the two images on the edge as the edge weight. This scheme performs better than the Centroid scheme when the images do not form tight clusters, but worse otherwise. Both of these schemes were proposed for lossy compression. The idea of representing lossless compression schemes as spanning trees of a graph was briefly examined by Gergel *et al.* [7].

Although principal components analysis has been used successfully for the recognition problem in large sets of similar images, it is not suitable for compression especially when images must be coded in a lossless manner [9]. Specifically, this technique can only work for lossy compression on a set of images that are very similar so that the set can be approximated by a subspace of small dimensions. Furthermore, the computational cost is high, and a large number of eigenimages must be stored to accurately reconstruct the images.

## 3. Graph Theoretical Framework

Let $S_n = \{I_1, I_2, \ldots, I_n\}$ be a set of $n$ images of the same dimensions. We define two additional images for our framework:

- the zero image $I_{n+1} = I_z$ with $I_z(i,j) = 0$ for all $(i,j)$;
- the average image $I_{n+2} = I_a$ where $I_a(i,j) =$

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

Additional images may also be introduced if necessary. We define two image sets:

- $S_z = S_{n+1} = S_n \cup \{I_{n+1}\}$;
- $S_a = S_{n+2} = S_{n+1} \cup \{I_{n+2}\}$.

Given an image set $S_* \in \{S_z, S_a\}$, we define a complete undirected, weighted graph $G = (V, E)$. The vertices of $G$ are $V = \{I_i \mid I_i \in S_*\}$, and $E = \{(I_i, I_j) \mid I_i, I_j \in V\}$ defines the graph edges. The weight for each edge $(I_i, I_j)$ is defined by the function $w(I_i, I_j)$, where $w : S_* \times S_* \to \mathbb{R}_{\geq 0}$ is a function that measures the cost to reconstruct $I_j$ assuming $I_i$ is known. We will assume that $w$ is symmetric in this paper, and that the weight of an edge is the cost of encoding the difference image $I_i - I_j$. For example, the function $w$ can be an entropy measure of the difference image, which represents the potential of compressing the difference image. Another choice of $w$ is the actual size of the compressed difference image using compression algorithms such as JPEG2000 [6].

A compression scheme is represented by a subset of edges in $G$, so that the chosen edges correspond to the difference that is coded. The chosen edges must form a spanning tree in order for the entire set to be decompressed. The zero image $I_z$ (called the virtual node in [5]) allows the original images to be represented as the difference image $I_i - I_z$ on the edges. This allows spanning forests to be included in our framework as a spanning tree.

Our framework differs from that of Chen *et al.* [5] in two ways. First, we introduce an average image, which allows one to include previous schemes in this framework as well (see Section 5). Also, we assume that the edge weights are symmetric, which is necessary for our extension to lossy compression (see Section 4).

Any compression scheme that takes advantage of inter-image redundancy between pairs of images can be represented as a spanning tree within our framework. For each compression scheme, the total weight of the spanning tree represents the storage cost for the scheme. For lossless compression, a minimum spanning tree (MST) gives us the optimal compression scheme given the image set. We remark also that the spanning tree for each scheme has to be encoded either implicitly in the algorithm or explicitly, but the cost is negligible.

## 4. Extension to Lossy Compression

The previous MST schemes [5, 12] were actually proposed for lossy compression. If the MST computation is performed on the graph constructed from the original images, errors introduced by lossy compression may change the actual graph. As a result, it is not necessarily true that the MST computed is the optimal scheme. Furthermore, it is difficult to compute the MST of the "optimal perturbed graph"—how each image is perturbed in the final graph depends on its path from $I_z$ in the spanning tree, and the path in turn depends on the tree edges chosen.

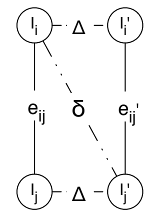



**Figure 1. Using an edge in $T_G$ as an edge of a spanning tree in $G'$.**

In this section, we give a bound on the difference between the optimal MST compared to the computed MST. This bound depends on the maximum distortion introduced by the lossy compression process. Our result is valid for an MST of any graph that results from perturbing each vertex in the original graph by less than the maximum distortion. Therefore, the bound gives a performance guarantee of the MST scheme based on the original graph compared to the optimal MST scheme, without explicitly computing the optimally perturbed graph. The optimally perturbed graph is difficult to compute because the cyclic dependency between the choice of tree edges and the perturbation on vertices.

The derivation of our results assumes that the edge weight function is a metric. For example, the root-mean-squared error (RMSE) is a metric commonly used to measure differences between images, and corresponds reasonably well to actual compression cost.

In the following, we let $N$ be the number of vertices in the graph $G$. Let $I_i'$ be images such that the RMSE between $I_i$ and $I_i'$ is bounded by $\Delta$, and $G'$ be the graph constructed from $\{I_i'\}$. We can think of $\Delta$ as a "quality" parameter in our lossy compression algorithm, and $I_i'$ as the reconstructed images. We will use $T_G$ and $T_{G'}$ to denote an MST of $G$ and $G'$, respectively. In addition, for any spanning tree $T$ we denote its cost (the sum of edge weights) by $w(T)$.

Suppose $e_{ij}$ is the edge between $I_i$ and $I_j$ in $G$ and it is included in $T_G$. Let $e_{ij}'$ be the edge connecting $I_i'$ and $I_j'$ in $G'$ (Figure 1). Since the edge weight function is a metric, we may apply the triangle inequality to see that

$$\delta - \Delta \leq w(e'_{ij}) \leq \delta + \Delta,$$
$$w(e_{ij}) - \Delta \leq \delta \leq w(e_{ij}) + \Delta,$$

so that

$$w(e_{ij}) - 2\Delta \leq w(e'_{ij}) \leq w(e_{ij}) + 2\Delta. \quad (1)$$

Now, if $I_i = I_z$, there is no reconstruction error so $I_z = I'_z$. In that case, we may refine the bound to be

$$w(e_{ij}) - \Delta \leq w(e'_{ij}) \leq w(e_{ij}) + \Delta. \quad (2)$$

Since there are $N-1$ edges in a spanning tree, we arrive at the following performance guarantee on using the MST computed from $G$ in $G'$.

**Theorem 1** *Let $T_G$ be an MST of $G$, and $d$ be the degree of $I_z$ in $T_G$. If $T$ is the spanning tree of $G'$ obtained by using the same edges as $T_G$, then*

$$|w(T) - w(T_G)| \leq (2N - 2 - d)\Delta.$$

□

This gives us a bound on the performance of the compression scheme. However, it does not relate to the optimal scheme given by an MST of $G'$.

Now, since $T$ in Theorem 1 is a spanning tree of $G'$, it follows that

$$w(T_{G'}) \leq w(T) \leq w(T_G) + (2N - 2 - d)\Delta. \quad (3)$$

In the derivation of Theorem 1, we note that the only assumptions on $G$ and $G'$ are that $I_z = I'_z$ and the RMSE between $I_i$ and $I'_i$ is bounded by $\Delta$. It does not matter whether $G$ or $G'$ is the graph constructed from the original images. Therefore, we may interchange the role of $G$ and $G'$ to also obtain

$$w(T_G) \leq w(T) \leq w(T_{G'}) + (2N - 2 - d')\Delta. \quad (4)$$

Since $d, d' \geq 1$, we get from (3) and (4) that

$$|w(T_G) - w(T_{G'})| \leq (2N - 3)\Delta. \quad (5)$$

Combining with Theorem 1 gives the following performance guarantee of the MST compression scheme relative to the optimal scheme.

**Theorem 2** *Let $T$ be a spanning tree of $G'$ obtained by using the same edges as $T_G$, and $d$ as defined in Theorem 1. Then*

$$|w(T) - w(T_{G'})| \leq (4N - 5 - d)\Delta.$$

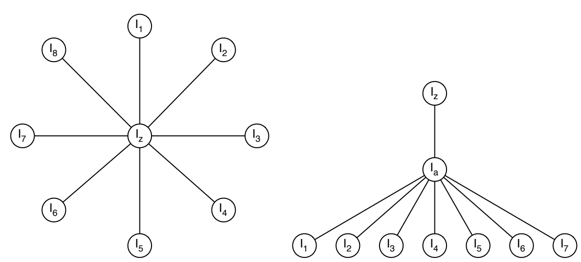
□

We make an important note that there is no assumption on the actual perturbations made on the images for the graph $G'$, so that the performance bound above applies to any graph with the same quality bound $\Delta$. Thus, the performance bound indeed gives a relationship between the quality of compression $\Delta$ and the coding performance relative to the optimal scheme. This is a first step in understanding the trade-off between quality and compression performance.

## 5. Compression Schemes

We examine four compression schemes and model them within our framework. Although we only mention four schemes, any compression scheme that utilizes inter-image redundancy between two images can be represented within our framework.

**Traditional Scheme** The traditional scheme results in a star graph created from the set $S_z$ with $I_z$ as the center as shown in Figure 2(a). The spanning tree represents encoding each image individually as the only edges are $(I_i, I_z)$.




(a) Traditional (b) Centroid

**Figure 2. Two compression schemes in our framework.**

**Centroid Scheme** In our framework, the Centroid scheme with a single cluster is represented by a star graph created from $S_a$. The center for this spanning tree is the average image $I_a$, and each edge $(I_i, I_a)$ is the difference of the image $I_i$ from the average $I_a$ (Figure 2(b)). Similarly, the Centroid scheme with multiple clusters can be represented by a spanning tree with an extra vertex for each cluster average. In addition, template extraction schemes can also be represented by using the template image in place of the average image in the spanning tree.

**MST Scheme (MST)** The previous MST schemes [5, 12] can be adapted for lossless compression to obtain the optimal lossless compression scheme. If it is used for lossy compression, the result is not necessarily optimal as shown in Section 4.

**MST Scheme with Average (MST$_A$)** This scheme is similar to the MST scheme above except that $S_a$ is used to construct the graph. There are situations when the Centroid scheme is better than the MST scheme, and vice versa. This scheme automatically chooses the best scheme locally for a subset of the images in the graph $G$. For example, if $S_n$ contains a group of similar images and a few outliers, it may be more efficient to encode the outliers using inter-image differences instead of the differences to the average. As in the Centroid scheme, we may introduce additional average images if there are multiple clusters.

## 6. Experimental Results

For lossless compression, we report the actual compression results using JPEG2000 [6] to represent edge weights in the graph. For lossy compression, RMSE is used as the edge weight function because it is a metric. For each image set, a spanning tree was generated for each of the four compression schemes presented in Section 5. In our experiments, we report the total number of bytes required to store the whole set. JPEG2000 compression is performed using the JasPer software package [4].

Since we are not interested in clustering algorithms in this work, we only show the result of the Centroid scheme with one cluster. We note, however, that the Centroid scheme with multiple clusters can also be modelled within our framework. Thus, our MST scheme is still no worse than the Centroid scheme.

A typical image from each image set is shown in Figure 3. The results are given in Table 6 for lossless compression. The best results are highlighted for each set. Note that the minimum is always achieved by one of the two MST schemes.

For lossy compression, we show the percentage improvement over the traditional scheme at various values of average RMSE, which is controlled indirectly by the bit rate (Table 6). It is not always possible to obtain results for a particular RMSE value (shown as "n/a"). We have not included results for the MST$_A$ scheme. Similar to the lossless case, its performance is not worse than the Centroid scheme. The Centroid scheme (and hence the MST$_A$ scheme) performs very well compared to the traditional scheme in most cases.

The experimental results show the effectiveness of our framework in adapting to any given image set. We now give some remarks on the results for the specific image sets.

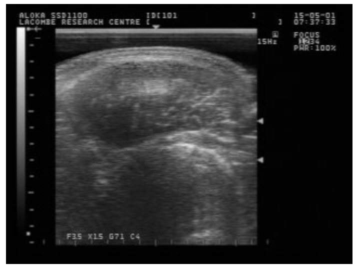
**Galway** The first image set contains 28 images from a webcam [1]. All the images are similar except that the people in the image are moving. Because of the large amount of redundancy among the images, we see that the Centroid method performs the best with this set for lossless compression. Of course, the MST$_A$ scheme cannot perform worse than the Centroid scheme. The MST scheme did not provide much improvement for either lossless and lossy compression.

**Pig** The second test set consists of 304 Ultrasound images of pig ribcages. Most of the images of this set are very similar and therefore form a tight cluster. Here the MST$_A$ scheme provides the best performance for lossless compression, a 39% improvement over the traditional scheme. The majority of images in the set are connected through the average image in the MST computed, but there are also some images connected through inter-image edges. Our framework allows the best compression scheme to be chosen locally in an image set. This is an improvement over using either scheme independently.

**Joe** The third image set contains 162 webcam images from Joe Tourist Weather [3]. The images are taken at intervals throughout the day. This results in interpixel redundancy in the difference images as large portions of each image change in a similar manner due to the lighting conditions. The MST scheme provides the best result for the lossless compression of this image set. For sets of images that do not form a tight cluster but have small inter-image difference, MST without an average image provides the best encoding scheme. Since the image set contains many small clusters, even if we combine clustering with the Centroid scheme, it does not perform as well as the MST scheme without average images. This can also be seen from the results from lossy compression.

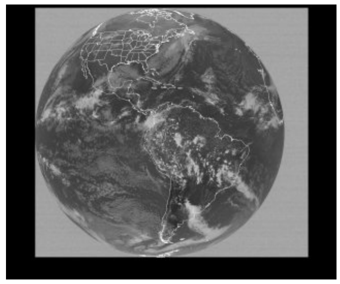
**GOES** Satellite images of earth from the GOES Project are used for the fourth image set [2]. This set contains 128 images. The set contains two distinct clusters of images with the focus on eastern and western North America. Here, the traditional scheme provides the best lossless compression performance because the difference images require more storage than the original images using JPEG2000. In fact, the 4th order entropy indicates that the original images are indeed easier to compress than the difference images. Thus, it is

(a) Galway



(b) Pig


(c) Joe



(d) GOES

**Figure 3. Typical images from each set.**

**Table 1. Lossless compression results (in bytes).**

| Scheme | Galway | Pig | Joe | GOES | Combination |
|---|---|---|---|---|---|
| **Traditional** | 3,792,948 | 47,681,737 | 4,696,170 | **56,234,650** | 9,587,002 |
| **Centroid** | **3,710,070** | 29,109,705 | 4,560,285 | 60,051,481 | 7,879,385 |
| **MST** | 3,792,948 | 31,347,677 | **4,434,349** | **56,234,650** | **7,117,393** |
| **MST$_A$** | **3,710,070** | **29,019,727** | 4,451,390 | 56,503,150 | 7,159,164 |

**Table 2. Lossy compression results: percentage improvement over traditional scheme at various average RMSE values.**

| Image Set | RMSE = 3.0 | | RMSE = 4.0 | | RMSE = 5.0 | |
|---|---|---|---|---|---|---|
| | Centroid | MST | Centroid | MST | Centroid | MST |
| Galway | 14.4% | n/a | 21.3% | 4.4% | 25.1% | 9.3% |
| Pig | 34.4% | n/a | 34.2% | 0.5% | n/a | 6.8% |
| Joe | 3.0% | 5.1% | 15.9% | 16.7% | 25.6% | 23.1% |
| GOES | 25.4% | n/a | 26.7% | n/a | 31.0% | n/a |

not always better to compress difference images instead of the original ones. On the other hand, the centroid scheme performs quite well for lossy compression. It is important to note that our graph-theoretical framework automatically chooses the best scheme depending on the situation.

**Combination** The final set of test images is composed of the Galway image set combined with the first 30 images from the Pig set. The goal was to test the framework with two clusters of images that have no relation to one another. As expected, the resulting spanning tree has two independent subtrees for each cluster. The MST scheme gives the best performance for this set for lossless compression.

## 7. Conclusion

In this paper, we proposed a new framework for all lossless compression schemes that consider inter-image redundancy between two images in a set. Our experimental results have shown that our framework allows us to compute the optimal compression scheme that is guaranteed to be no worse than the previously proposed schemes. In the lossy case, we also showed that although computing the optimal scheme is difficult because of the errors introduced by the compression process, we can compute a compression scheme whose performance is "close to" that of the optimal scheme. Experimental results once again showed significant improvement over the traditional scheme in the lossy case.

## 8. Acknowledgements

We 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] Galway.net. `http://www.galway.net/webcam/images.shtml`.

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

[3] Photos courtesy of joetourist infosystems. `http://www.JoeTourist.net`.

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

[5] 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 (ICIP '04)*, pages 1289–1292, 2004.

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

[7] B. Gergel, H. Cheng, and X. Li. A unified framework for lossless image set compression. To appear in the Proceedings of the Data Compression Conference, 2006. 1 page.

[8] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1993.

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

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

[11] M. Manouvrier, G. Jomier, and M. Rukoz. Storage and management of similar images. *J. Braz. Comp. Soc.*, 6(3):13–25, 2000.

[12] C. Nielsen and X. Li. MST for lossy compression on image sets. To appear in the Proceedings of the Data Compression Conference, 2006. 1 page.

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

[14] J. Wang and H. Yan. Form image compression using template extraction and matching. In *Procs VIP'2000, Sydney, Australia*, December 2000.