IMAGE SET COMPRESSION THROUGH MINIMAL-COST PREDICTION STRUCTURES

Chia-Ping Chen[1], Chu-Song Chen[1], Kuo-Liang Chung[2], Hsueh-I Lu[1,3], Gregory Y. Tang[3]*

[1] Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan.
[2] Comput. Sci. & Info. Eng., National Taiwan University of Science and Technology, Taipei, Taiwan.
[3] Comput. Sci. & Info. Eng., National Taiwan University, Taipei, Taiwan.
* Corresponding author, email: song@iis.sinica.edu.tw

ABSTRACT

We propose a new scheme for compressing an image set by building its minimal-cost prediction structure. Existing prediction-based video coding methods can be easily extended and incorporated into this scheme to achieve higher compression efficiency. According to this prediction structure, we also develop a progressive transmission approach for interactive object movie (OM) browsing.

1. INTRODUCTION

Compactly representing a set of images is a fundamental problem in image processing. In the past, major research issues in image set compression focused on video compression, in which plenty of research results have been done and industrial standards such as MPEG-x, and H.26x have been proposed.

However, when the set of images is not from a video or cannot be represented as a linearly ordered sequence of images, systematic methods are still lacked for its compression via predictive coding. Examples of such image sets include OMs, chest X-ray images in a hospital, face images in an album, and so on. These image sets do not have linear orders apparently inherent in their representations, and thus existing video compression methods may not be suitable for their compression.

In this paper, we propose a new method for compressing a set of images. In particular, we concentrate our study on the compression of an OM [1], which is a set of object-centered images taken for the same 3D object from various viewing angles. Nevertheless, note that our method can be easily generalized for compressing other types of image sets.

We develop a predictive-coding strategy for image set compression. Predictive coding has become a main paradigm in video compression. Although its compression ratio is not the highest, it is widely used because of its flexibilities, easy for progressive transmissions, and also very high compression ratios. In this work, we investigate the problem of how to build a prediction structure with high compression efficiency for an image set. We also introduce a progressive transmission strategy based on the prediction structure.

2. RIVIEW OF OM [1]

The OM considered in this paper is a two-dimensional object image array, as illustrated in Fig. 1. The image array can be taken by using some specially designed devices; for example, Fig. 2 shows Kaidan's Megallen M-2500 which is for OM acquisition of real objects. Since the object is allowed to being interactively rotated along both pan and tilt directions, the OM can be lively browsed like a real 3D object. OM provides probably the most efficient way to achieve photo-realistic exhibition of 3D objects since no complex 3D digitization or 3D model reconstruction procedures are involved.

Unlike polygon-mesh-based technologies, the object is usually allowed to see only from the shooting angles for OM. Nevertheless, most users do not care whether or not an object can be viewed from an 'arbitrary' viewing angle, as long as "dense enough" viewing angles are provided. Typically, several hundreds of images are required for an OM so that a smooth manipulation can be achieved. Reducing such huge data amount is thus important for OM construction.

Apple's QTVR [2] can build and play OMs. In QTVR, a row of images contains the images captured at a particular tilt angle, and a column of images contains the images captured at a particular pan angle. These images are stored sequentially in a row-major order, and video compression techniques were used in QTVR for OM compression.

However, treating an OM simply as a row-major-ordered image sequence is not the most appropriate way to exploit the redundancies inherent in an OM. Since an OM is a two-dimensional image array, correlations among the images of an OM cannot be simply represented in a linearly ordered manner. How to organize the images in an OM so that more redundancies can be removed is a central issue to achieve effective compression. We will investigate this problem in the next section.

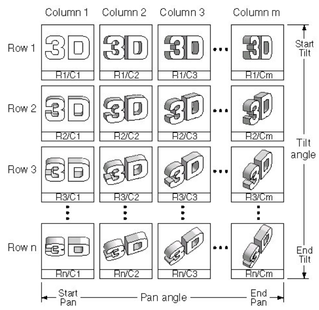

Figure 1. Illustration of an OM.

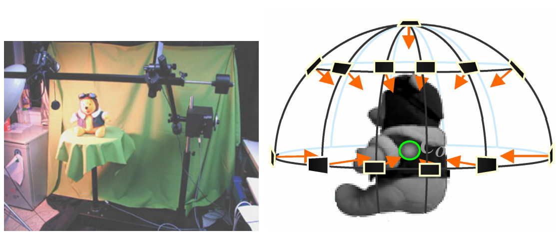
Figure 2. OM acquisition.

3. THE PROPOSED COMPRESSION METHOD

Consider n images I_1, I_2, …, I_n of an object. In our work, predictive coding techniques are used, so we define a measure, *prediction cost* (PC), between a pair of images to indicate how well one image predicts the other. Let p_{ij} be the PC of using I_i to predict I_j. A small value of p_{ij} indicates that I_i is suitable to be the reference to predict I_j and can yield a compact representation for I_j.

Then we construct a *prediction graph* $G = (V, E)$, where V contains n nodes representing n images, and each edge (i, j) of E is denoted by an edge cost p_{ij}. The prediction graph represents prediction-capability relationships between each ordered pair of images. We try to utilize this prediction graph to find some optimal prediction structure, which determines how images can be predicted by others.

The optimal prediction structure considered in our work is a directed subgraph of G which should contain all the nodes (images). The existence of edge (i, j) in the optimal prediction structure indicates that I_i serves as the reference image to predict I_j. In other words, to decode I_j, we have to decode I_i at first because I_j is predicted by I_i. To find the optimal prediction structure, we consider the following constraints:

(C1) There cannot be any cycle in the optimal structure, or decoding is impossible.

(C2) There exists an image, called the root image, which has no reference images.

(C3) Every image, except the root images, should be predicted by another reference image.

We hope the sum of edge costs to be minimized. Evidently, the optimal prediction structure should be the minimum spanning tree (MST) of G.

When a symmetric PC measure is adopted (i.e., $p_{ij} = p_{ji}$), standard algorithms (e.g., the PRIM algorithm [3]) can be used for finding the MST. However, in general, the PC measure is not symmetric, and common textbook algorithms cannot be directly applied because they solve the MST problem only for undirected graphs. Fortunately, in both [4] and [5], Chu/Liu and Edmond have independently given efficient algorithms for finding the MST on a directed graph, respectively. In our work, the Chu-Liu algorithm [4] is used.

3.1. Choose Appropriate PC Measures

Given a predictive coding technique, there are many alternative definitions of PC. When PC is defined as the bit length required to encoding the image being predicted, (i.e., p_{ij} is set as the bit length required to encode I_j when using I_i as the reference image), finding the MST implies that the required bit length for encoding the OM is also minimized. Hence, bit length is the ideal one for defining PC, and it yields a lower bound for compressing an image set by predictive coding if the prediction strategy has been selected.

However, constructing the prediction graph G with its edge costs being bit lengths is usually time-consuming. In practice, we would like to use other measures instead and, hopefully, this measure can approximate the bit-length-measure performance. For OMs, one possible measure is the difference between (θ_i, φ_i) and (θ_j, φ_j), where θ_i and φ_i denote the pan and tilt angles relative to the object where the image I_i was taken, respectively. Although this strategy is simple and fast, it usually provides poor approximations of bit lengths.

In our work, motion estimation and compensation are used for prediction, and the mean absolute residual after motion estimation serves as the PC measure. In our experience, this measure is a good alternative of bit length and its computation time is highly reduced in practice.

3.2. Encode OM According to the MST

When a prediction structure has been found for an image set, many existing predictive-coding algorithms can be easily modified or extended for coding the image set. Compared with video coding, the only difference is that a video is a linear array of images, while the minimal-cost prediction structure of an image set is a tree of images. Essential components in a video coding algorithm, such as transform coding, motion estimation/compensation, and entropy coding, can still be applied since these components are independent of the prediction structure.

In our work, MPEG-1 is extended and used as the predictive coding technique. Nevertheless, we would like to emphasize that it is not a must for our framework because other video coding methods can also be extended and used in our encoding scheme.

Once the MST is found, the root image is denoted as an I-image (intra-image). Images other than I-images are P-images. As the coding methods in MPEG-1, I-images are encoded independently using block-based discrete cosine transform (DCT) and coefficient quantization, and P-images are compressed by motion estimation/compensation with their parent images as reference images.

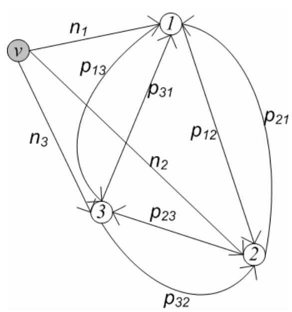

Figure 3. Modifying G for MSF.

3.3. Extend MST to MSF

In the above, we focus on the case in which only one root image is used. How to construct the minimum-cost prediction structure when multiple root images are allowed will be investigated in the following. Unlike the single-root case, the node cost (defined as the bit length for encoding the I-image) should be taken into account in the multi-root case. We further denote n_i to be the node cost for I_i. Formally, the constraint (C2) defined before is modified as follows for the multi-root case:

(C2') There exist *some* root images that have no reference image.

The constraints (C1), (C2') and (C3) yield the minimum spanning forest (MSF) of G. More specifically, we try to find the MSF which consists of several disjoint trees. The sum of the edge costs of these trees as well as their root node costs is expected to be minimized.

To our best knowledge, no algorithms have been proposed before for solving the MSF problem for a directed graph. Nevertheless, we found that the MSF problem can be easily solved by using the algorithm for finding the MST, if we augment G with a virtual node as follows:

1. Add an additional virtual node v to G.
2. Add n edges, $(v, 1), \ldots, (v, n)$, to G.
3. Denote the edge cost of (v, i) as n_i.

Let G' denote the augmented graph of G. We then find the MST of G' by using the Chu-Liu algorithm with node v being the single root node of G'. An example is given in Fig. 3. After the MST is found (Fig. 4a), removing node v and all edges originating from node v results in k disjoint trees. The k disjoint trees make up the MSF we want (Fig. 4b).

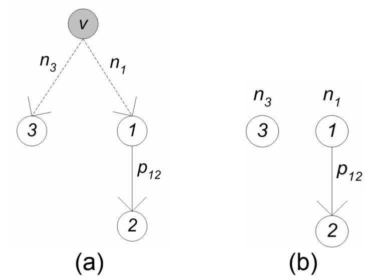

Figure 4. Finding the MSF.

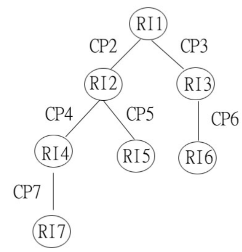

Figure 5. Progressive transmission.

4. PROGRESSIVE TRANSMISSION

After encoding an OM, each node stores a residual image (but root nodes store the original images), and each edge stores coding parameters (including macroblock types, motion vectors ...) used to reconstruct the predicted image. Since the size needed for coding parameters is far smaller than that needed for the residual images, transmitting coding parameters is much faster than transmitting the residual images. The key idea of our progressive transmission scheme is to use the coding parameters to reconstruct images whose residual images are still not available; we call these images as *pre-rendered images*.

Residual images and coding parameters are transmitted according to breadth-first-search (BFS) order of the MST, and coding parameters can be transmitted earlier than residual images by specifying the number of pre-rendered layers. For example, if the number of pre-rendered layers is 2, coding parameters are sent 2 layers earlier than residual images. The transmission sequence is CP2, CP3, CP4, CP5, CP6, RI1, CP7, RI2, RI3, RI4, RI5, RI6, RI7, where CP and RI denote coding parameters and residual images, respectively (Fig. 5). When the first residual image RI1 is received, images 1~ 6 are visible, where images 2~6 are pre-rendered. When RI2 is received, all images are visible but images 3~7 are pre-rendered, which will be refined one after one when RI3 ~ RI7 are received successively.

Determining the number of pre-rendered layers is a trade-off between interactivity and image quality. More pre-rendered layers provide a higher interactivity with lower- quality pre-rendered images. According to our experience, even when only one pre-rendered layer is specified, the interactivity can be greatly improved for OM transmission.

5. EXPERIMENTAL RESULTS

We used four OMs in our experiments. Two of them, RABBIT and TEAPOT are rendered from synthetic 3D models. Another two, BOTTLE and CLAY are acquired from real objects. They are sampled by every 15 degrees pan or tilt, resulting 168 images totally. One view of each OM is shown in Fig 6. The compression results are shown in Fig. 7, where the PSNR remains higher than 30dB when the bpp is reduced to 0.1.

We also compare the compression performances between the following two encoding methods: (1) organizing the images in an OM as a linear row major sequence, and (2) organizing them as the MST. The results are shown in Figs. 8 and 9 for RABBIT and BOTTLE, respectively. MST encoding consistently achieves better PSNRs for both OMs. This comparison indicates how MSTs outperform linear sequences which are used by previous methods, such as Apple's QTVR.

6. CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a new scheme for image set compression. In our scheme, the minimal-cost prediction structure is found for the image set, so that redundancies among images can be greatly exploited. Our method outperforms the methods of encoding an OM based on a linear sequence. We also propose a progressive transmission approach which enables users to interact with OM in a friendlier way. In addition to OMs, our method can be used for compressing other image sets (e.g., images in association with the same category contained in an image database) as well.

Our method uses one reference image for prediction. How to extend our framework such that multiple reference images can be used for prediction is a future work (e.g., the B-frame in MPEG uses two reference images). In essence, multi-reference prediction yields the associated prediction graph to be a hypergraph in which hyper edges (i.e., edges in association with more than two nodes) exist. As we know, no algorithms have been proposed in graph theory to solve the MST or MSF problem in a hypergraph. We will investigate this problem in the future.

Acknowledgement: This paper was supported in part from the Digital Archive Program under NSC 93-2422-H-001-0003, from the National Science Council, Taiwan.

REFERENCES

[1] S. E. Chen, "QuickTime VR: an image-based approach to virtual environment navigation," *Proc. ACM SIGGRAPH 1995*.
[2] *Inside QuickTime, The QuickTime Technical Reference Library*, Apple Incorp., October 2002.
[3] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, The MIT Press, 1990.
[4] Y. J. Chu and T. H. Liu, "On the shortest arborescence of a directed graph," *Sci. Sinica*, vol.14, pp. 724-742, 1965.
[5] J. Edmonds, "Optimum branchings," *J. Res. Nat. Bur. Standards 71B*, 233-240, 1967.

Figure 6. Four OMs used in the experiments.

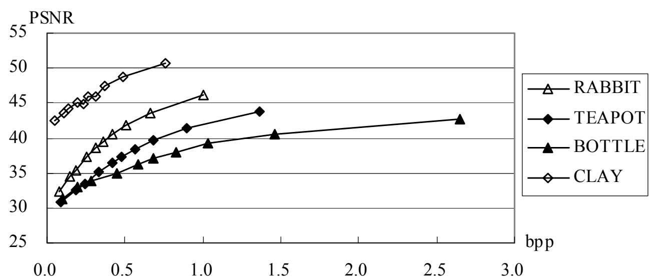

Figure 7. Compression results.

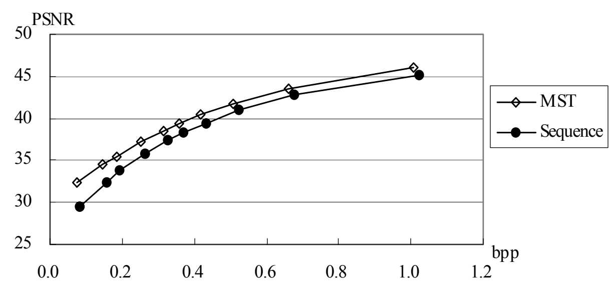

Figure 8. Comparison with sequence (RABBIT).

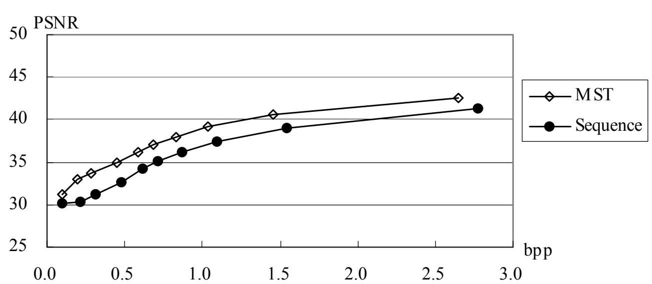

Figure 9. Comparison with sequence (BOTTLE).