On the non-optimality of four color coding of Image partitions

Sameer Aagarwal
sagarwal@cs.ucsd.edu
Department of Computer Science
Univeristy of California, San Diego

1 Introduction

Object based image decomposition and compression is an area of active research. For image and video compression we are witnessing a move from purely transform based coding to methods to methods which decompose the image into multiple layers/objects and compress each of them separately exploiting layer specific properties to get better better distortion rates as well as achieving better overall compression ratios. For documents we now have an ITU standard for Multiple Raster Content (MRC). The DjVu system is an implementation of this standard [3]. The DjVu systems treats the documents as made up of three layers corresponding to color, background, and text. For each of these layers it uses layer specific algorithms. Similarly the MPEG-4 standard for video compression specifically focuses on identifying objects across frames and compressing in different streams.

Segmentation based image compression requires the following two components for successful compression.

1. Robust image segmentation methods.
2. Compression algorithms which can be used based on the content in a particular segment.

A number of robust image segmentation have become available over the past few years [6], and in what follows we will assume that a "good" partitioning of the image into coherent connected segments is available. We consider the image to be composed of multiple layers, each layer made up of texture corresponding to one of the layers. One way of compressing the image would be to compress each of the segments individually, which includes both the texture and the shape information for the segment. Another option is to only encode texture information in the segment layers, and use an additional layer for storing the image partition information. We call this the image partition layer. In this paper we will address the issue of effective compression of the partition layer.

2 Boundary Based Image Partition Encoding

There exists a number of classical approaches to image partition encoding. We mention a few prominent ones [4].

2.1 Run Length Encoding

The simplest method of shape encoding uses the fact that all information about the image partitions is contained in the boundary pixels of the shapes. Hence we can consider a bitmap image with the boundary pixels with value 1 and the pixels internal to each segment with value 0. This bitmap has a large amount of redundancy and can be efficiently compressed using Run Length Encoding.

2.2 Chain Coding

Chain coding refers to storing the shape of each segment in terms of its contour. A chain code describes a walk along the contour and the relative direction in which one should walk to traverse it. This description is then RLE and entropy encoded. Optimizations for whole image partitions include storing shared boundaries between segments only once.

2.3 Fourier Descriptors

The contour of a segment is a closed 2-d curve. We consider the boundary to be a continuous curve and consider a parametric decomposition of the curve in terms of the curve length. Considering the plane of the image as the complex Argand plane we can represent the set of points (x, y) as

$$z(p) = x(p) + iy(p) \tag{1}$$

the periodic curve $z(p)$ can now be written in terms of its Fourier transform

$$z(p) = \sum_{u=-\infty}^{\infty} \hat{z}_u e^{\frac{2\pi iup}{P}} \tag{2}$$

here P is the perimeter of the curve and $\hat{p}_u$ is the set of Fourier transform coefficients of z.

The Fourier coefficients can now be quantized and entropy encoded to created a compressed representation of the image. A similar method can be employed by approximating the boundary with splines and storing the spline coefficents instead.

3 Region Based Boundary Encoding

Another class of methods that have recently been used for partition encoding are region based methods, which depend on using non-lossy compression methods on image bitmaps created by associating a single color with all the pixels in a segmeent [1].

The simplest way of constructing such a bitmap is to assign pixels in each region the corresponding segment index. This methods while simple has an obvious difficulty. The number of bits required to represent the image depends on the number of regions in the image and grows with the number of segments in the image.

3.1 The Four Color Theorem

An interesting solution to this problem is possible using the four color theorem [2]. The four color theorem states that any planar map can be colored using only four colors, such that no adjacent region has the same color. An image partition is nothing more than a planar map and hence we can use just four colors to describe the image partition. This places an upper limit of 2 bits per pixel for any image partitioning [1].

3.2 Minimum Entropy Coloring

In graph theoretic terms the map coloring problem is formulated as a vertex coloring problem, where individual vertices of a graph represent regions in the map, and edges between the vertices represent adjacency between the region. The minimum number of colors required to color the vertices of a graph G such that no two adjacent vertices do not have the same color called the chromatic number of the graph and is denoted by $\chi(G)$. Now consider the graph G, such that with each vertex v_i we associate a weight p_i, which denotes the proportion of image pixels in the corresponding segment. Since we are interested in using this coloring for compressing the bitmap, where each color is treated as symbol, we are not interested in finding just any coloring of the graph. We are interested in finding that $k-$coloring which will minimize the entropy of the resulting $k-$colored image. We call this quantity the color entropy of the image and define it as :

$$H = -\sum_{i=1}^{k} C_i \log C_i \tag{3}$$

where

$$C_i = \sum_{j=i}^{N} p_j \delta_{\phi(j)i} \tag{4}$$

$phi(j)$ is the function that maps the set of integers $\{1, \ldots, N\}$ representing the segments to $\{1, \ldots, k\}$ the set of colors and δ_{ij} Kronecker's Delta. The lower the color entropy the higher compression we will be able to get.

As we mentioned earlier, planar graphs can be 4-colored. Theoretically there exists a quadratic algorithm for 4-coloring a planar graph [5]. It is extremely impractical to implement and no known implementations of it exist. Th existing four coloring methods are either heuristic which may or maynot return a 4-coloring, or are based on backtracking, which in the worst case can result in exponential performance. But before one gets involved in finding efficient ways of 4-coloring a graph, it is useful to differentiate the problem we are trying to solve from the problem of simply 4-coloring a graph.

While its tempting to think that since a planar graph requires 4 colors to color it, the minimum entropy coloring will also require 4 colors, we shall show in the following that this intuition is wrong.

We begin with the following lemma.

Lemma 1. *There exists a graph G with chromatic number $\chi(G) = n$ and maximum independent set $I(G)$, s.t. the subgraph induced by G-I(G) has chromatic number k. i.e. $\chi(G - I(G)) = n$.*

Proof. The proof is trivial and can be demonstrated by construction of an example. Consider the complete graph K_n. We know that $\chi(K_n) = n$. Consider now a new graph G' constructed as follows :

1. Add n^2 vertices to K_n.

2. Connect vertices $\{i * n, \ldots, (i+1) * n - 1\}$ to the i^{th} vertex of K_n.

The chromatic number of G' is still n. It can now be trivially shown that the maximum independent set for the graph G' is the set of n^2 vertices in $G' - K_n$. Which in turn proves the lemma. □

This leads to our main result,

Theorem 1. *There exist graphs with chromatic number k, s.t. the minimum entropy coloring uses more than k colors.*

Proof. By Lemma 1 we know there exists a graph s.t. that removing the maximum independent from it does not reduce its chromatic number. Let G be such an unweighted graph. Let $I(G)$ denote the max independent set of G. We will now associate weights with the vertices of G and show that there exists a $k+1$-coloring with entropy lower than the best (minimum entropy) k-coloring of G. For an arbitrary $1 > \epsilon > 0$, label each vertex in $I(G)$ with weight $1 - \epsilon/|I(G)|$, label the remaining vertices with the weight $\epsilon/|G - I(G)|$. Now color all vertices in $I(G)$ with the color $k+1$. The rest of the vertices can be colored using k colors independent of the vertices in $I(G)$. The *maximum* entropy for a coloring is when the vertices of $G - I(G)$ are equally divided amongst the k colors.

$$\begin{aligned} H_{k+1}(G) &= -[(1-\epsilon)\log(1-\epsilon) + \sum_{i=1}^{k} \frac{\epsilon}{k} \log(\frac{\epsilon}{k})] \\ &= -[(1-\epsilon)\log(1-\epsilon) + \epsilon \log(\frac{\epsilon}{k})] \qquad (5) \\ & \qquad (6) \end{aligned}$$

since the set I(G) cannot be removed from G and reduce the chromatic number, it means that in any k-coloring of the graph G, there is at least one vertex $x \in I(G)$ with color different from the rest of the vertices in $I(G)$. Hence the lower bound on the minimum entropy k-coloring of G is given by assuming that $1 - \frac{1}{|I(G)|}$ vertices are colored 1, the rest except for $k-2$ vertices are colored 2. The remaining $k-2$ colors are assigned one vertex each.

$$
\begin{aligned}
H_k(G) = &-\Bigg[\left((1-\epsilon)(1-\frac{1}{|I(G)|})\right)\log\left((1-\epsilon)(1-\frac{1}{|I(G)|})\right)+\\
&\left(\frac{1-\epsilon}{|I(G)|}+\epsilon-\frac{\epsilon(k-2)}{|G-I(G)|}\right)\log\left(\frac{1-\epsilon}{|I(G)|}+\epsilon-\frac{\epsilon(k-2)}{|G-I(G)|}\right)+\\
&\sum_{i=1}^{k-2}\frac{\epsilon}{|G-I(G)|}\log\left(\frac{\epsilon}{|G-I(G)|}\right)\Bigg]\\
&=-\Bigg[\left((1-\epsilon)(1-\frac{1}{|I(G)|})\right)\log\left((1-\epsilon)(1-\frac{1}{|I(G)|})\right)+\\
&\left(\frac{1-\epsilon}{|I(G)|}+\epsilon-\frac{\epsilon(k-2)}{|G-I(G)|}\right)\log\left(\frac{1-\epsilon}{|I(G)|}+\epsilon-\frac{\epsilon(k-2)}{|G-I(G)|}\right)+\\
&\frac{(k-2)\epsilon}{|G-I(G)|}\log\left(\frac{\epsilon}{|G-I(G)|}\right)\Bigg]\\
&>-\left[\left(1-\frac{1}{|I(G)|}\right)\log\left(1-\frac{1}{|I(G)|}\right)+\alpha\log\alpha\right]
\end{aligned}
\tag{7}
$$

where

$$
\alpha = max\{\frac{1}{|I(G)|}, 1-\frac{k-2}{|G-I(G)|}\} \tag{8}
$$

now since ϵ was chosen arbitrarily we can always choose ϵ to be sufficiently small so that the upper bound on H_{k+1} is smaller than the lower bound on H_k. Hence for the graph G with the above weight structure has a lower entropy with $k+1$ colors than the best k-coloring. Hence proved.

□

4 Remarks

Theorem 1 proves that even though the chromatic number of an unweighted graph is k, the minimum number of colors required to achieve the lowest entropy maybe more than k. This has some interesting implication and raises some interesting questions.

1. What is the optimal number of colors which will minimize the entropy ? in our proof we have only shown that there exist graphs for which the optimal number of colors which minimizes entropy is larger than the chromatic number of the graph. We conjecture that the optimal color entropy number for a graph with chromatic number k is bounded above by $k+1$.
2. We are not aware of any work on finding coloring which will explicitly reduce the color entropy. We conjecture that this problem is NP-Hard.

We believe further work in using map coloring for partition compression should focus on the above two settling the above two questions.

5 Conclusions

In this work we looked at some classical methods of image partition encoding. We also proved that for the class of new generation methods which depend on graph coloring for constructing bitmaps for compression, the intuition that the minimum number of colors required to color the graph is the same as what is required to minimize its color entropy is wrong.

6 Acknowledgement

The author would like to thank Prof. Serge Belongie for all his help, suggestions, time and unwavering faith in graph theory.

7 References

1. Accame, M., De Natale, F.G.B, Granelli, F., Efficient labeling procedures for image partition encoding. Signal Processing, vol.80, (no.6), pp 1127-31, June 2000.
2. Appel, K., Haken, W., Koch, J., Every planar map is four colorable. Part II. Reducibility, Illinois J. Math. 21 (1977), 491–567.
3. Bottou, L.,Haffner, P., Howard, P., G.,Simard, P., Bengio Y., Cun, Y. L., High Quality Document Image Compression with DjVu. Journal of Electronic Imaging, vol 7, no 3, pp 410-425, SPIE, 1988.
4. Jahne, B., Digital Image Processing. Springer Verlag, Berlin. 1998.
5. Robertson R. , Sanders, D. P., Seymour P., Thomas R., Efficiently four-coloring planar graphs. In Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, pages 571-575, May 1996
6. Shi, J., Belongie, S., Leung, T., Malik, J. Image and Video Segmentation: the Normalized Cut Framework. Proceedings of ICIP 1998.