

Simplified Segmentation for Compound Image Compression

Amir Said and Alex Drukarev
Hewlett-Packard Laboratories, Palo Alto, CA.
E-mail: {said, drukarev}@hpl.hp.com

Abstract

There are three basic segmentation schemes for compound image compression: object-based, layer-based, and block-based. This work discusses the relative advantages of each scheme and architecture, and studies the use of fast classification techniques for a segmentation that can be used together with a chosen compression architecture. Particularly, we consider classification techniques working on approximate object boundaries, which reduces the localization and precision of the segmentation, but in exchange allows faster, one-pass segmentation, low memory requirements, and a segmentation map that is better matched to existing compression methods. We show numerical results obtained on a printer application environment, where rigorous standards of visual quality have to be satisfied.

1. Introduction.

Still image compression represents a fairly mature branch of image processing. Most of the efforts in image compression until now have been in developing new algorithms that achieve better compression (often, at the cost of considerable increase in complexity). The analysis of the compression numbers published in the past several years makes one believe that we may be approaching the point of diminishing returns in this area. This requires rethinking of our approach to compression. Whereas in the past the focus of compression research has been on developing *better algorithms*, the future focus is likely to be on the *methods of combining* of various algorithms to achieve the best compression performance for the given type of images.

This approach is especially evident in considering compression of *compound, or mixed, images*. By a compound image we imply an image that contains data of various types, such as photographic images, text, and graphics. Each of these data types has different statistical properties, and is characterized by different level of distortion that a human observer can notice. This structure

of a compound image requires different parameter settings of a compression algorithm used for the entire image. Moreover, evidence dictates that the best compression performance is achieved when different algorithms are applied to different image types. For example, a lossy compression will achieve the best results for photographic images where the human eye may tolerate a fairly large amount of loss. On the other hand, lossless compression algorithms, especially the ones that handle well long runs, give the best compression results on black and white text and some simple graphics. The distortion produced in these types of image data by lossy methods, both DCT- and wavelet-based, is often quite visible, even for moderate compression ratios.

A general approach to compression of a compound image will include 3 major steps:

- image segmentation into the regions of similar data types,
- selection of the best compression algorithm for each region,
- bit allocation (quality setting) among various regions/compression algorithms.

In this paper we will present first a classification of various approaches to compression of compound images. We will then analyze in detail a particular simple scheme, which is based on segmenting the image along approximate object boundaries. This scheme allows fast, one-pass segmentation, has low memory requirements, and a segmentation map that is better matched to existing compression methods. The numerical results and the comparison with conventional (i.e., without segmentation) compression methods will be given.

2. Classification of Methods for Compound Image Compression.

Several approaches to image segmentation and compound document compression have been described in literature [1] - [7]. In order to have a unified way of analyzing all these schemes and to create a framework for the future research, it would be useful to develop a meaningful classification of various methods for

compound image compression. We would like to propose such a classification based on: the *type of segmentation* used; *compression methods* (single algorithm vs multiple algorithms), and the *type of image* (computer-generated vs acquired).

Image segmentation. The segmentation approaches used in compound document compression can be grouped into 3 classes:

- Object-based segmentation (see Fig.1a). In this case, a page is divided into regions, where each region follows exact object boundaries. An object may be a photograph, a graphical object, a letter, etc. In principle, this method may provide the best compression, since it provides the best match between a data type and the compression method most suitable for this data type. In reality, the best compression may not be achievable for the following reasons. Coding the object boundaries requires extra bits, and the typical algorithms, used for lossy image compression, are designed to operate on rectangular objects. They can operate on objects with non-rectangular boundaries, but the compression performance will suffer. Complexity is another drawback of this method, since precise image segmentation may require the use of very sophisticated segmentation algorithms.

- Layer-based segmentation [7] (see Fig. 1b). This approach can be regarded as a simplified version of the full object-based segmentation. The original page is divided into rectangular layers, where each layer can have one or more objects, and “mask” planes. A mask plane tells which pixels of a particular layer should be included in the final composite page. Each layer is compressed with a specific compression method. The advantages of this approach are simplified segmentation (when the number of layers is less than the number of objects), and a better match between layer boundaries and the compression algorithms (since layers are rectangular regions). Standard, off-the-shelf compression methods can be easily incorporated into this structure. The drawbacks of this method are: mismatch between the compression method used for a particular layer and the data types (when several various objects are included in the same layer), mismatch between the object boundaries and the compressed region boundaries, and an intrinsic redundancy, due to the fact that the same parts of the original image appear in several layers.

- Approximate block-based segmentation (see Fig. 1c). This also can be regarded as a simplified version of the full object segmentation. Each region follows approximate object boundaries, and is made of rectangular blocks. The size of the blocks may vary within the same region to better approximate the actual object boundary. The advantages of this approach are: simplified segmentation, better match between region boundaries and the compression algorithms, and the lack of redundancy, which may be present in the layer-based

approach. The potential drawbacks are the potential loss in the compression performance compared to the true object-based segmentation, and the need to slightly modify the off-the-shelf algorithms to work on non-rectangular regions. Note that the segmentation performed in this case is done with the purpose of optimizing the compression performance, and may not be appropriate for other uses, such as OCR, image enhancement, etc.

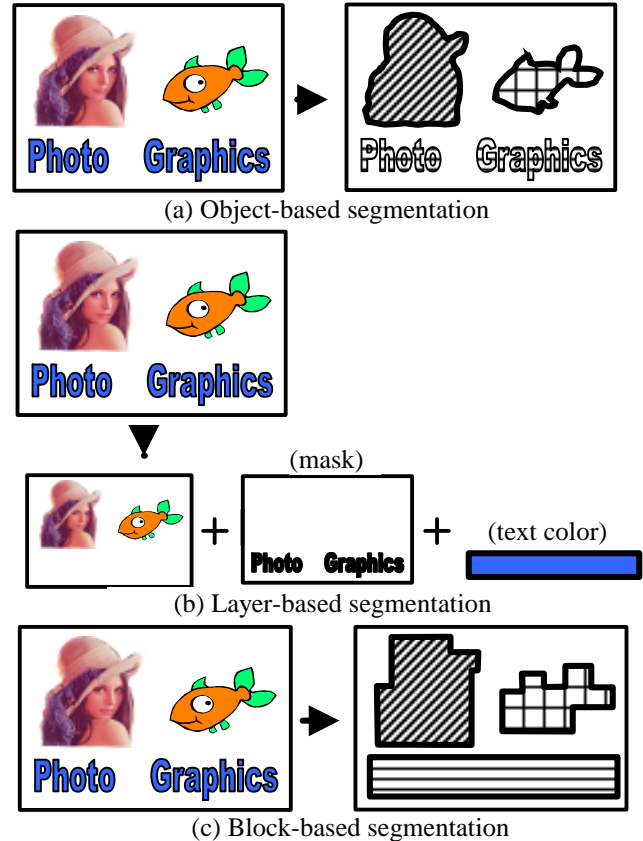


Figure 1. Segmentation approaches

Compression methods. As was mentioned above, the best compression will be achieved if different compression methods are used for different image regions. However, support for multiple compression algorithms may increase complexity of implementation. An alternative approach is to use a single adaptive algorithm, whose parameters can be adjusted for different image regions. The variable quantization JPEG, described in the Part III of the JPEG standard, is an example of such approach.

Type of image. From a practical point of view it is important to differentiate between computer-generated images and scanned, or otherwise acquired images. The main difference is that the acquired images will have a higher level of inherent noise. What may look like a flat region in a scanned image will actually have variable

pixel values. This will impact both the segmentation strategy, and the selection of the compression method.

3. Block Classification

Bandwidth is a very important limiting factor in application of image segmentation. Several segmentation schemes require morphological analysis of the different regions, and multiple passes over the image being segmented. However, each pass normally requires loading memory data from slow to fast memory (L1 cache, etc.), which is a slow process. Segmentation solutions based on multiple passes are much slower (or costly) than what can be expected by, for instance, counting the number of operations. Thus, an ideal solution would use a single pass to decide on the type of image region. Such solution would be very difficult with arbitrary shapes of segmentation regions, but it is feasible if we consider only a pre-defined shape.

For example, we can use rectangular blocks, and decide the image type based only on the properties of the pixels inside the block. We call such technique *block classification*. This technique is theoretically sub-optimal, since it must classify all pixels in the block in the same manner, even if the block contains the boundary between two regions. Another potential problem would be the fact that it does not consider the pixels in the block's neighborhood.

Some factors mitigate the sub-optimal performance of a block-based scheme around region boundaries. First, compound images now have fairly high resolutions. If the block size is small enough, we should not expect to have the image type changing for every block. (We are considering types like text, graphics, photos, etc.; we do not assume a fine level segmentation that would split, for instance, letters and their background). Therefore,

boundary blocks consist of a small fraction of the image. We can identify those boundary blocks doing a simple analysis on a *block level*. Such analysis would not require as much bandwidth, because the number of blocks is much smaller than the number of pixels, and it is required only when some transition is found.

Using those techniques we can obtain a segmentation that is good enough for image compression, allowing high text quality, and with a complexity that is practically the same as required by a one-pass segmentation. Furthermore, the resulting segmentation has very attractive features. For instance, many image compression methods are efficient on rectangular regions, but do not work well on arbitrary regions.

Figure 2 shows how the block classification is applied to an individual block. First, the block is classified according to the distribution of its pixels. Next, the class of the neighboring blocks is analyzed. At this stage we can identify if a block is in the boundary between two regions, or change the classification if its confidence is low. To avoid adding the burden of analysis to the decoder, the final classification is added to the compressed stream. Finally, the block is compressed according to the identified type. The identification of boundaries is important because we can change compression parameters for those blocks. For instance, we can set a different quantization when moving from a lossy to lossless region. By introducing extra buffering, it is possible to use the block neighborhood analysis for merging blocks that are classified in the same manner. This can be useful for some compression methods that work better in large blocks (e.g., it allows longer run-length, overlapping transforms, etc.).

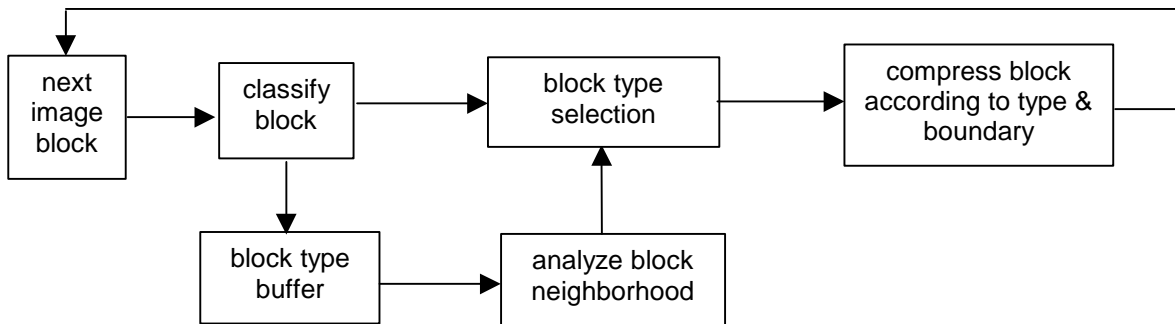


Figure 2. One pass block classification using neighborhood analysis.

4. Numerical results

We analyzed the performance of block classification in several experiments. The main objective was to test it for hardcopy systems, which should produce excellent quality images. To minimize hardware costs, the classification engine considered only two types: photo, which is compressed with a lossy DCT-based scheme (similar to JPEG), and graphics/text, compressed with a near-lossless method (max. error = 2), similar to JPEG-LS.

Figure 3 shows a test document: a simulated magazine page, rendered at 600 ppi, RGB. Table I shows the compression for different compression methods. Note that we are comparing lossy and lossless compression methods, but it is still a valid comparison.



Figure 3. Test Document A

Compression method	Compression ratio
GZIP (LZ77)	14.3:1
baseline JPEG	15.4:1
JPEG-LS (max. error = 2)	16.5:1
Block classification	51.5:1

Table I - Compression ratios obtained for Test Document A

The objective is to guarantee a visually lossless reproduction (a must, for instance, in printer applications). Thus, we set the quality factor of the lossy methods to guarantee such high quality. Under such stringent requirements, we can see that the performance of a lossy method like JPEG is comparable to the lossless/near-lossless methods.

This fact is even more important for documents that contain only graphics, and no photos, like the example shown in Figure 4. As we can see in Table II, in such pages lossless compression methods like Lempel-Ziv have exceptional performance.

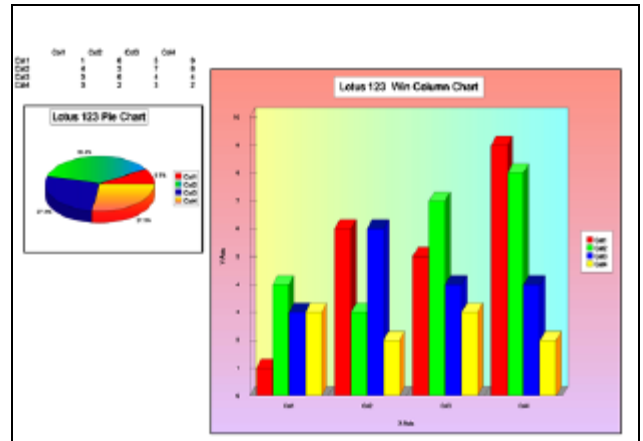


Figure 4. Test Document B

Compression method	Compression ratio
GZIP (LZ77)	165:1
baseline JPEG	71:1
JPEG-LS (max. error = 2)	90:1
Block classification	150:1

Table II - Compression ratios obtained for Test Document B

Table I shows that even with this simplest form of one-pass segmentation, and with only two compression methods, the block classification technique can yield compression ratios much larger than those obtained with the most commonly used image compression methods.

The block classification result in Table II is slightly inferior to LZ. This can be explained by the fact that the page is composed of only graphics. Thus, no segmentation would be necessary. The block classification has the overhead of coding information about segmentation, but we can see from the results that it is a small overhead, and a small price to pay for a much greater flexibility.

5. Conclusions

We analyzed the different techniques for segmentation and compression of compound documents. We analyzed a low-complexity segmentation technique based on a one-pass classification of image blocks. We considered the limitations of this technique, and how they can be circumvented by using simple block neighborhood analysis.

The numerical results show that, although it has very low complexity, such technique can yield quite significant gains in the compression of compound documents.

6. References

- [1] T. Pavlidis and J. Zhou, "Page segmentation and classification," *CVGIP: Graph. Models Image Proc.*, vol. 54, Nov. 1992, pp.484-496.
- [2] R. de Queiroz and R. Eschbach, "Fast segmentation of JPEG compressed documents," *J. Electron. Imag.*, vol.7, Apr. 1998, pp.367-377.
- [3] P. Haffner, L. Bottou, P. Howard, P. Simard, Y. Bengio, Y. LeCun, "High quality document image compression with 'DjVu'," *J. Electron. Imag.*, vol. 7, July 1998, pp.410-425.
- [4] A. Benedetti, N. Scarabottolo, "Towards a dedicated compression pipeline for document image archiving," *Proc. Workshop on Doc. Image Analysis (DIA'97)*, IEEE Comput. Soc., 1997.
- [5] H. T. Fung, K. J. Parker, "Color, complex document segmentation and compression," *Proc. of the SPIE*, vol.3027, San Jose, CA, 1997.
- [6] R. Sennhauser, K.W. Ohnesorge, "Document image compression using document analysis and block-class-specific data compression methods," *Proceedings of the SPIE*, vol. 2186, 1994, pp. 146-55.
- [7] International Telecommunication Union (ITU-T), "Mixed Raster Content (MRC)," Draft recommendation T.44.