CONstrained AND unconstrained simplification of image partitions encoded with the method of transition points

Armando J. Pinho
Dept. de Electrónica e Telecomunicações / IEETA
Universidade de Aveiro, 3810–193 Aveiro, Portugal
ap@det.ua.pt — http://www.ieeta.pt/~ap

ABSTRACT
The encoding of image partitions plays a key role in the overall efficiency of the so-called region-based image coding techniques. Recently, a method based on the novel concept of “transition point” was proposed for the lossless encoding of arbitrary contour maps. This paper addresses the problem of partition simplification in the context of this new encoding method. A new simplification approach is introduced and results concerning its compression efficiency under constrained and unconstrained conditions are presented.

1. INTRODUCTION
Partition coding has been addressed by several authors [1–9]. For the purpose of image coding, it is possible to use either lossless or lossy image partitions. However, lossy representations should be used carefully, because even small errors can produce very annoying visual artifacts. There are partition coding techniques that, despite allowing lossless representations, are more suitable for lossy compression (among them, we find those based on contour approximation using splines, polygons or Fourier descriptors [10]). On the other hand, there are methods that are lossless by nature, although allowing lossy representations by means of contour simplification.

One of the partition coding techniques of the latter type is the recently proposed method based on the concept of “transition point” of a contour map [9, 11, 12]. The superiority of this technique in relation to differential chain coding [1, 13–16] has been shown to be attained for a wide range of contour complexity [9, 12].

The method of the transition points relies on the recognition and encoding of three classes of configurations of contour elements, namely, corners (C), vertical tees (V) and horizontal tees (H). They are generically referred to as the “transition points” of the contour map. As can be seen by the example presented in Fig. 1, the set of transition points can take the form of a 2 bits per pixel image (to which we call the “map of transition points”), whose pixels store symbols from the set {C, H, V, •} (“•” denotes “no transition point present”). To achieve compression, this image should be losslessly encoded using, for example, an image coding technique, such as JBIG [17].

Fig. 1. Representation of image partitions using the method of transition points: example of conversion from an image partition into the map of transition points.

2. THE SIMPLIFICATION APPROACH
Contrarily to encoding methods such as chain coding, the encoding complexity of the method based on transition points cannot be accurately represented by the number of active contour elements of the contour map, but instead by the number of transition points that it contains. Therefore, in this case, the simplification process should be headed towards the reduction of the number of transition points of the contour map. It worths noting that this contrasts with the frequent assumption that less active contour elements always leads to a reduction in code length.

The simplification technique proposed in this paper relies on local simplifications performed on a context window of 3×3 pixels (corresponding to 12 contour elements) that is slide across the image according to a raster scan movement. As a result of a local simplification, the region to which the central pixel of the context window belongs is changed (see Fig. 2). This change is performed if:

• The total number of transition points inside the context window is reduced (if there are more than one

simplification possibility, choose that with less tee transition points, i.e., V’s and H’s).

- The total number of transition points inside the context window remains the same, but the number of tees is reduced (this also produces a reduction in complexity because, generally, tees are less frequent than corners).

The simplification process is iterated until no more simplifications are performed. When there are configurations that are considered equivalent in terms of encoding complexity, we choose the one that introduces less distortion in the reconstructed image. Finally, we note that even in cases where the contour complexity can be decreased, the creation of new regions is not permitted.

3. EXPERIMENTAL RESULTS

Partitions of the “Lena” and “bridge” images (256 × 256, 8 bpp) were generated by applying an uniform quantization to the gray-level images, according to the following pointwise transformation:

\[
\tilde{g}(r,c) = \frac{g(r,c)}{2q+1} (2q+1) + q
\]

where \(g(r,c)\) and \(\tilde{g}(r,c)\) denote, respectively, the original and quantized pixel values of the image, and \(q\) denotes the maximum quantization error, i.e.,

\[
|g(r,c) - \tilde{g}(r,c)| \leq q, \quad \forall (r,c)
\]

After quantization, the boundaries of the regions are found, the image partition is constructed, simplified and finally encoded.

As explained above, each local simplification implies moving a pixel from one region to another neighbor region. If the error introduced by the simplification process is not controlled, then the final reconstruction error, \(e(r,c) = g(r,c) - \tilde{g}(r,c)\) where \(\tilde{g}\) denotes the gray-level image after simplification, can be unbounded, i.e., can be as much as the maximum allowed pixel value (255, for 8 bpp images).

When all simplifications are allowed, despite the value of \(e(r,c)\), then we refer to them as “unconstrained simplifications” of the image partition. However, although providing the greatest compression, do not limiting the maximum error may be unacceptable in some situations. Therefore, we also include results of constrained simplification, for which the maximum of \(|e(r,c)|\) is kept below some predefined value, i.e.,

\[
|e(r,c)| \leq q + s, \quad \forall (r,c)
\]

Figure 3 displays plots of the number of bits required to encode the image partitions, as a function of the maximum quantization error \(q\) required to encode partitions of two test images, “Lena” and “bridge”, for several values of \(s\) (i.e., maximum simplification error).

When all simplifications are allowed, despite the value of \(e(r,c)\), then we refer to them as “unconstrained simplifications” of the image partition. However, although providing the greatest compression, do not limiting the maximum error may be unacceptable in some situations. Therefore, we also include results of constrained simplification, for which the maximum of \(|e(r,c)|\) is kept below some predefined value, i.e.,

\[
|e(r,c)| \leq q + s, \quad \forall (r,c)
\]
(s = \{0,1,3,9,u\}, where “0” indicates lossless compression and “u” means unconstrained simplification).

It is particularly interesting to observe the evolution of the plots corresponding to the partitions of the “bridge” image. These plots exhibit the absolute maximum for $q \approx 5$, showing a decreasing behavior for $q < 5$, when $q \to 1$. This effect was previously pointed out and explained in [11], and it is a characteristic of the method of transition points when applied to contour maps corresponding to high detail images, such as the “bridge” image.

The importance of imposing constraints to the simplification process is illustrated by the examples depicted in Figs. 4–5. Figure 4 shows a $128 \times 128$ segment of the “Lena” image, and corresponding partition, after uniform quantization using (1) and $q = 10$. Figure 5 displays the simplified versions of the images of Fig. 4. As expected, the compressed size of the partitions decreases as $s$ increases. However, the quality of the reconstructed images also degrades. Therefore, the $s$ parameter can be used to establish a balance between compression efficiency and image quality, guaranteeing, additionally, an upper bound on the maximum absolute ($L_{\infty}$) reconstruction error.

4. REFERENCES


Fig. 4. (a) “Lena” image after uniform quantization using (1) and \(q = 10\): \(L_\infty\) error = 10, PSNR = 32.5dB; (b) Boundaries of the corresponding partition encoded with the method of the transition points: 66 240 bits. Note: although only a 128 \(\times\) 128 segment of the image is shown, the numbers refer to the complete image.

Fig. 5. Simplification results of the partition shown in Fig. 4: (a) \(s = 3\), \(L_\infty\) error = 13, PSNR = 32.2dB, 56 368 bits (14.9% reduction); (b) \(s = 9\), \(L_\infty\) error = 19, PSNR = 31.7dB, 51 608 bits (22.1% reduction); (c) \(s = u\), \(L_\infty\) error = 181, PSNR = 28.7dB, 46 592 bits (29.7% reduction).