Scanned Document Compression Using a Block-based Hybrid Video Codec

Alexandre Zaghetto, Member, IEEE, and Ricardo L. de Queiroz, Senior Member, IEEE

Abstract—This paper proposes a hybrid pattern matching/transform-based compression method for scanned documents. The idea is to use regular video interframe prediction as a pattern matching algorithm that can be applied to document coding. We show that this interpretation may generate residual data that can be efficiently compressed by a transform-based encoder. The efficiency of this approach is demonstrated using H.264/AVC as a high quality single- and multi-page document compressor. The proposed method, called Advanced Document Coding (ADC), uses segments of the originally independent scanned pages of a document to create a video sequence, which is then encoded through regular H.264/AVC. The encoding performance is unrivaled. Results show that ADC outperforms AVC-I (H.264/AVC operating in pure intra mode) and JPEG2000 by up to 2.7 dB and 6.2 dB, respectively. Superior subjective quality is also achieved.

Index Terms—Scanned document compression, advanced document coding, pattern matching, H.264/AVC.

I. INTRODUCTION

COMPRESSION of scanned documents can be tricky. The scanned document is either compressed as a continuous-tone picture, or it is binarized before compression. The binary document can then be compressed using any available two-level lossless compression algorithm (such as JBIG [1] and JBIG2 [2]), or it may undergo character recognition [3]. Binarization may cause strong degradation to object contours and textures, such that, whenever possible, continuous-tone compression is preferred [4]. In single/multi-page document compression, each page may be individually encoded by some continuous-tone image compression algorithm, such as JPEG [5] or JPEG2000 [6], [7]. Multi-layer approaches such as the mixed raster content (MRC) imaging model [8]–[12] are also challenged by soft edges in scanned documents, often requiring pre- and post-processing [13].

Natural text along a document typically presents repetitive symbols such that dictionary-based compression methods become very efficient. For continuous-tone imagery, the recurrence of similar patterns is illustrated in Fig. 1. Nevertheless, an efficient dictionary-based encoder relying on continuous-tone pattern matching is not that trivial. We propose an encoder that explores such a recurrence through the use of pattern-matching predictors and efficient transform encoding of the residual data.

It is important to place our proposal within the proper scenario. Three premises are assumed. Firstly, we want to avoid complex multi-coder schemes such as MRC. Secondly, the decoder should be as standard as possible. Since we are dealing with scanned compound documents (mixed pictures and text), natural image encoders, such as JPEG2000, are the most adequate. Non-standard encoders, based on fractals [14]–[16], texture prediction [17], [18], template matching [19] or multiscale pattern recurrence [20], [21], are good options out of the scope of what is being proposed. Thirdly, one should provide high quality reconstructed versions of scanned documents. This is especially important if rare books of historical value must be digitally stored, thus discarding optical character recognition (OCR) and token-based methods [2]. In summary, we want a standard single coder approach that operates on natural images and delivers high-quality reconstructed compound documents.

The proposed coder makes heavy use of the H.264/AVC standard video coder [22]. H.264/AVC has been well explained in the literature [23]–[27]. H.264/AVC leads to substantial performance improvement when compared to other existing standards [25], [28], such as MPEG-2 [29] and H.263 [30]. Among such improvements we can mention [22], [31]: interframe variable block size prediction; arbitrary reference frames; quarter-pel motion estimation; intraframe macroblock prediction; context-adaptive binary arithmetic coding; and in-loop deblocking filter. Results point to at least a factor of two improvement over previous standards. The many coding advances brought into H.264/AVC not only set a new benchmark for video compression, but they also make it a formidable compressor for still images [32]. The intraframe macroblock prediction, combined with the context-adaptive
binary arithmetic coding (CABAC) [33] turns the H.264/AVC into a powerful still image compression method (i.e. working on a video sequence composed of one single frame). We refer to this coder as AVC-I. Gains of the AVC-I over JPEG2000 are typically in the order of 0.25 dB to 0.5 dB in PSNR for pictorial images [31], [32], [34]. For compound images (mixture of text and picture) [4], the PSNR gains are more substantial, even surpassing the mark of 3 dB improvement over JPEG2000, in some cases [31].

The hypothesis presented in this paper is that a scanned document encoder that employs state-of-the-art video coding techniques and generates an H.264/AVC-decodable bit-stream yields the best rate-distortion performance compared to other continuous-tone still image compressors. 1

II. THE PROPOSED METHOD AND ITS IMPLEMENTATION USING AVC

The proposed document coder has a generic concept and an implementation based on a stock H.264/AVC video coder. We now describe the desired features and how one can implement them using AVC. The generic description may help the reader to adapt other video coders for that purpose or to develop non-standard-based (proprietary) variations.

A. Block-based pattern matching

The encoder is based on pattern matching. The document image is segmented into blocks of pixels. Each block is matched to an existing pattern in a dictionary which is populated by the previous contents of the same document. In order to do that, we partition the scanned document, which may be made of one or more scanned pages of \(H \times W\) pixels, into \(N_p\) sub-pages or frames. Hence, a scanned book may be decomposed into many frames. Figure 2 illustrates the page pre-processing (partition) algorithm, while Fig. 3 shows an example of a frame sequence built from a 3-pages set. 1

Fig. 2. A document page is partitioned into segments (labeled in sequence). Each one is considered a frame and can be sequentially encoded.

Fig. 3. Example of a frame sequence, built from a 3 pages set, \(N_p = 4\) frames/page. Frames 1 to 4, 5 to 8, and 9 to 12 are built from pages 1, 2 and 3, respectively.

Blocks have for example \(16 \times 16\) pixels and each one is matched to an existing pattern in a previous frame. In this way, the previous frames make a dynamic dictionary of patterns to look when encoding the present frame, which is continuously being updated as more frames are encoded. Once a match is found, the matching pattern is used to predict the block and the prediction error (residue) is encoded along with the frame number and position where the match was found (reference vector). A block can be partitioned into smaller blocks to ease prediction at the cost of spending more bits to encode reference vectors. Figure 4 illustrates the effect of using the pattern matching prediction algorithm. Figures 4 (a) and (b) show examples of a reference and a current text area, respectively. Figures 4 (c), (e) and (g) represent the predictions of the current text using \(16 \times 16\), \(8 \times 8\) and \(4 \times 4\)-pixel block partitions. Figures 4 (d), (f) and (h) are the corresponding residual data. Notice that the \(4 \times 4\)-pixel prediction generates a lower-energy residual, when compared to the \(16 \times 16\) and \(8 \times 8\) prediction, however, they require encoding more reference vectors.

In this context, video coders often use motion estimation techniques which are essentially the same as pattern matching. The H.264/AVC is capable of partitioning macroblocks of \(16 \times 16\) pixels into any valid combination of blocks of \(16 \times 8\), \(8 \times 16\), \(8 \times 8\), \(4 \times 8\), \(8 \times 4\), and \(4 \times 4\) pixels. Our algorithm is then to feed the document frames as video frames into AVC since its motion estimation algorithm will take care of the pattern matching search for us. However, motion estimation algorithms always take advantage of the fact that video content at the same frame position in neighbor frames are typically very correlated. Since this is not our case, it is advisable to make the search window to cover as much as possible of the reference frames, or the whole frame, in order to enrich the dictionary and to remove the spatial dependency.
Fig. 4. Illustration of approximate pattern matching using interframe prediction: (a) reference text; (b) current text; (c), (e) and (g) predicted text (block size: 16 × 16, 8 × 8 and 4 × 4 pixels, respectively); and (d), (f) and (h) prediction residue (block size: 16 × 16, 8 × 8 and 4 × 4 pixels, respectively). Each zoomed image patch has 178 × 178 pixels.

B. Inter- and intra-frame prediction

AVC also allows for intra-frame prediction, in which a block (partitioned or not) can be predicted from neighboring blocks by means of directional extrapolation of the border pixels. The decision to use or not intra-frame prediction is typically based on rate-distortion optimization (RDO) and we use RDO in all our simulations. However, AVC does not allow for in-frame motion vectors (IFMV), but many variations using such a feature and other sophisticated methods of intra-frame prediction do exist [19]. Apparently, HEVC will also support IFMV [36]. Breaking up the pages into frames allows for some intra-document prediction similar to IFMV, yet using a stock video coder. Furthermore, the information derived from IFMV is typically very small compared to all compressed data, such that the advantage should not be much relevant. Because of that, we do not use IFMV.

Another issue is the random access to different book pages. In order to get to a book page, we are forced to decompress all the frames it uses as reference. So, if random access is an issue, we suggest to periodically use no-reference frames, i.e. frames in which inter-frame prediction is not allowed, relying on pure intra-frame prediction/extrapolation.

In our encoder, using the AVC structure, each block-partitioning combination and prediction mode is tested and the best one is picked through RDO. With RDO within AVC, in the k-th configuration test in a macroblock, AVC computes the rate \( R_k \) (bits spent to encode the block) and distortion \( D_k \) (sum of absolute differences - SAD) achieved by reconstructing the block. One picks the block partition method that minimizes

\[
J_k = R_k + \lambda D_k.
\]

The process is then repeated for every macroblock. As usual, \( \lambda \) controls compression ratios and is varied to find the RD curves in our simulations.

C. Residual coding

The residual macroblock, i.e. the prediction error, is transformed using 4 8×8-pixel discrete cosine transform (DCT) or an integer approximation of it. The transformed blocks are quantized and encoded using arithmetic coding. H.264/AVC uses an integer transform with similar properties as the DCT and the resulting transformed coefficients are quantized and entropy encoded using CABAC.

D. Compound documents and region classification

Compound document compression usually segments the image into regions and classifies each one as containing text and graphics or images (or halftones, for instance). Once a region is classified, it can be encoded using a proper algorithm. This approach is driven by objects such as text characters so that regions of the image are labeled based on our estimate of its contents. Our method, however, is driven by the compression itself. Rather than only testing pattern-matching-based prediction for every block partition, we also test prediction by extrapolating neighboring blocks, as in "intra-prediction" in H.264/AVC. The RD-optimized selection of the best prediction assures that the best option is picked. Text and graphics shall contain recurrent patterns and will be often encoded using patterns from previous regions, while pictorial regions may resort to intra-prediction. In this sense, segmentation is embedded into the encoding process. In fact, the block prediction and RDO may have the same effect of
Fig. 5. Configuration parameters that have greater influence on the encoder performance: $R_f$ (number of reference frames) and $S_r$ (search range).

a segmentation map, even though benefiting the compression process, and not the true identification of image contents.

E. Encoder and decoder summary

In our concept, the frames are fed into AVC in sequence just like in a regular video coder. Because of that relation to AVC, we refer to the proposed method as advanced document coding, or, simply, ADC. In a nutshell, ADC operation can be summarized as (i) break the book pages into frames; (ii) feed all frames to H.264/AVC resulting in an AVC-compatible stream; (iii) decode the bit stream; and (iv) assemble the decoded frames into the final document book pages.

In order to work well, H.246/AVC should operate in "High" profile, following an IPPP... framework. The encoder should periodically use no-reference I-frames in the case random access is desired. RDO should be turned on. Motion estimation should be set to full search over a window that is as large as possible. Note that other video coders such as HEVC and MPEG-2 will also work, even though achieving different performance levels due to their different sophistication levels.

III. EXPERIMENTAL RESULTS

In our tests, different page sets are compressed using JPEG2000, AVC-I (H.264/AVC operating in pure intra mode) and the proposed ADC. The reason we chose JPEG2000 and AVC-I for comparison is that these are the most suitable standards that would meet the three premises presented in Section I.

Distortion metrics based on visual models such as Structural Similarity (SSIM) [37] and Video Quality Metric (VQM) [38] have been extensively tested for pictorial content. However, they are unproven for text and graphics, which rely more on resolution than on number of gray levels. Readability is very important and some alternative metrics such as OCR efficiency are considered. A good objective metric to reflect subjective perception of text has not been well explored yet. Hence, we opted to stick to the traditional PSNR as a distortion metric.

In JPEG2000 and AVC-I compression, the pages are separately encoded. As for ADC, the first frame of the sequence is encoded as an I-frame (only intraframe prediction modes are used) and all the remaining frames are encoded as P-frames (in addition to intraframe prediction, only past frames are used as reference by the interframe prediction). We also considered that each page may be segmented into $N_p$ = 4 frames, $N_p$ = 16 frames, or not segmented at all ($N_p$ = 1, for multi-page documents only). Two configuration parameters have greater influence on the encoder performance. One is the number of reference frames, $R_f$, the other is the search range, $S_r$, as illustrated in Fig. 5. Initially, we evaluated the effect of choosing different values for $S_r$ and $R_f$. Figure 6 shows PSNR plots comparing JPEG2000, AVC-I and ADC ($N_p$ = 4 frames/page), for different combinations of $S_r$ and $R_f$. The PSNR was calculated using the global mean square error (MSE). The higher the $S_r$ and $R_f$ values, the better the rate-distortion performance. In particular, for $S_r$ = 32 pixels and $R_f$ = 5 frames, ADC outperforms AVC-I by more than 2 dB and JPEG2000 by more than 5 dB, at 0.5 bit/pixel (bpp).

Our test set is composed by 18 documents divided into the
In addition, the intraframe prediction, the DCT-based transform and the CABAC contribute to improve the encoding efficiency.

In essence, our work can be summarized as splitting the document into many pages, forming frames, and feeding the frames to AVC. Despite the simplicity of the idea, the performance for scanned documents is unrivaled, to our knowledge. Results show that ADC objectively outperforms AVC-I and JPEG2000 by up to 2.7 dB and 6.2 dB, respectively, with more significant gains observed for multi-page text-only documents. Furthermore, the encoder outputs documents with superior subjective quality. Replacing H.264/AVC by HEVC in ADC would yield even larger gains.

REFERENCES


![Table 1: Average objective (PSNR in dB) improvement over existing standards for the 4 document test sets.](image)

<table>
<thead>
<tr>
<th>Document Set</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG2000</td>
<td>6.26</td>
<td>5.61</td>
<td>4.47</td>
<td>2.41</td>
</tr>
<tr>
<td>AVC-I</td>
<td>2.75</td>
<td>2.58</td>
<td>1.56</td>
<td>0.89</td>
</tr>
</tbody>
</table>

Following 4 classes:

- Class 0: 6 multi-page text-only documents;
- Class 1: 6 single-page text-only documents;
- Class 2: 3 multi-page compound documents; and
- Class 3: 3 single-page compound documents.

Figure 7 illustrates one example of each class. PSNR plots for the whole test set are shown in Figs. 8, 9, and 10. Figure 11 shows average PSNR improvement of ADC over JPEG2000 and AVC-I for each of the document class test sets. In all cases, JPEG2000 and AVC-I are objectively outperformed, considering $S_r = 32$, $R_f = 5$ and $N_p = 16$. Figure 12 (a) shows a zoomed part of the original “cerrado” sequence. Its encoded and reconstructed versions using AVC-I, JPEG2000 and ADC, at approximately 0.25 bits/pixel, are shown in Figs. 12 (b), (c) and (d), respectively. ADC also yields superior subjective quality. As a reference, in Table 1 we present the gains of the proposed method over AVC-I and JPEG2000 using Bjontegaard’s method [39] applied to the curves in Fig. 11. As one can see from the table and from the graphics, the gains are very substantial.

The software-based tests using the popular and efficient x246 implementation of the H.264/AVC standard and the Kakadu implementation of JPEG 2000, indicate that ADC (AVC running with 5 reference frames, slowest search, $32 \times 32$ window, in IPPPP... mode) is near $10 \times$ slower than JPEG 2000. x264 in an Intel Core i7 platform has been shown to encode RGB video at a rate of 3 to 30 million pixels per second (Mps). The variation is due to the various x264 settings that affect speed and quality. Of course, in order to do that, the system was dedicated to the task, as an appliance. A scanned letter-sized (8.5 $\times$ 11 in) page at 600 pixels per inch (ppi) yields about 33 million pixels. Hence, we can expect a page compression speed roughly in the order of 5 to 50 pages per minute (ppm). This page rate may be acceptable for many on-the-fly applications and is definitely reasonable for off-line compression of books and such. A rigorous complexity study of the encoding algorithms presented here is beyond the scope of this paper.

IV. CONCLUSIONS

In this paper, we presented a pattern matching/transform-based encoder for scanned documents named ADC. The reason why we decided to use H.264/AVC tools to implement the proposed method is because its interframe prediction scheme allied with RDO yield an efficient pattern matching algorithm.

The entire test set is available at [http://image.unb.br/queiroz/testset](http://image.unb.br/queiroz/testset)
Fig. 7. Examples of documents used in our experiments: (a) class 0: multi-page text-only documents; (b) class 1: single-page text-only documents; (c) class 2: multi-page compound documents; and (d) class 3: single-page compound documents.
Fig. 8. PSNR plots for class 0 (multi-page, text-only) documents: (a) “guita” (number of pages: 2, size: 1568 × 1024 pixels); (b) “cerrado” (number of pages: 2, size: 1568 × 1056 pixels); (c) “krishnamurti” (number of pages: 2, size: 1280 × 800 pixels); (d) “samkhya” (number of pages: 3, size: 1568 × 1008 pixels); (e) “fedon” (number of pages: 5, size: 1408 × 1056 pixels); (f) “principia” (number of pages: 8, size: 1600 × 1152 pixels). Search range and number of reference frames are $S_r = 32$ and $R_f = 5$, respectively.
Fig. 9. PSNR plots for class 1 (single-page, text-only) documents: (a) page 2 of “guita” (1568 × 1024 pixels); (b) page 1 of “cerrado” (1568 × 1056 pixels); (c) page 1 of “krishnamurti” (1280 × 800 pixels); (d) page 2 of “samkhya” (1508 × 1008 pixels); (e) page 3 of “fedon” (1408 × 1056 pixels); (f) page 8 of “principia” (1600 × 1152 pixels). Search range and number of reference frames are $S_r = 32$ and $R_f = 5$, respectively.
Fig. 10. PSNR plots for class 2 (multi-page, compound) documents: (a) “paper” (number of pages: 4, size: 2304 × 1632 pixels); (b) “spectrum” (number of pages: 5, size: 2304 × 1632 pixels); (c) “scientific” (number of pages: 5, size: 2152 × 1632 pixels). PSNR plots for class 3 (single-page, compound) documents: (d) “carta” (2152 × 1632 pixels); (e) “spore” (1360 × 1024 pixels); (f) “avatar” (2060 × 1488 pixels). Search range and number of reference frames are $S_r = 32$ and $R_f = 5$, respectively.
Fig. 11. Comparison of ADC against JPEG2000 and AVC-I in terms of PSNR averaged for documents in: (a) class 0; (b) class 1; (c) class 2; and (d) class 3.
Fig. 12. Subjective comparison among coders: (a) zoomed part of “cerrado” sequence; reconstructed versions using (b) AVC-I, (c) JPEG2000 and (d) ADC, at approximately 0.25 bits/pixels. ADC yields superior subjective quality.