Artif Intell Rev (2018) 50:539–568
https://doi.org/10.1007/s10462-017-9551-9

A review on document image analysis techniques directly in the compressed domain

Mohammed Javed[1] · **P. Nagabhushan**[2] · **Bidyut B. Chaudhuri**[3]

Published online: 21 March 2017
© Springer Science+Business Media Dordrecht 2017

Abstract The rapid growth of digital libraries, e-governance, and internet based applications has caused an exponential escalation in the volume of 'Big-data' particularly due to texts, images, audios and videos that are being both archived and transmitted on a daily basis. In order to make their storage and transfer efficient, different data compression techniques are used in the literature. The ultimate motive behind data compression is to transform a big size data into small size data, which eventually implies less space while archiving, and less time in transferring. However, in order to operate/analyze compressed data, it is usually necessary to decompress it, so as to bring back the data to its original form, which unfortunately warrants an additional computing cost. In this backdrop, if operating upon the compressed data itself can be made possible without going through the stage of decompression, then the advantage that could be accomplished due to compression would escalate. Further due to compression, from the data structure and storage perspectives, the original visibility structure of the data also being lost, it turns into a potential challenge to trace the original information in the compressed representation. This challenge is the motivation behind exploring the idea of direct processing on the compressed data itself in the literature. The proposed survey paper specifically focuses on compressed document images and brings out two original contributions. The first contribution is that it presents a critical study on different image analysis and image compression techniques, and highlights the motivational reasons for

✉ Mohammed Javed
javedsolutions@gmail.com; javed@nitte.edu.in

P. Nagabhushan
pnagabhushan@hotmail.com

Bidyut B. Chaudhuri
bbc@isical.ac.in

1 Department of Computer Science and Engineering, (An Autonomous Institute Under VTU, Belagavi), NMAM Institute of Technology, Nitte 574110, India

2 Department of Studies in Computer Science, University of Mysore, Mysuru 570006, India

3 Computer Vision and Pattern Recognition (CVPR) Unit, Indian Statistical Institute, Kolkata 700108, India

pursuing document image analysis in the compressed domain. The second contribution is that it summarizes the different compressed domain techniques in the literature so far based on the type of compression and operations performed by them. Overall, the paper aims to provide a perspective for pursuing further research in the area of document image analysis and pattern recognition directly based on the compressed data.

Keywords Compressed document · Compressed domain · Compressed image processing · Compressed data analysis

1 Introduction

Since ancient times, writing has been in use as a tool to communicate and disseminate information in forms which could be stored and accessed (Baird and Tombre 2014; Baird et al. 1992). These included the use of papyrus, palm leaves, clay tablets, stones, and metal plates for the purposes of communication and documentation. However, since the invention of paper in 105 A.D., paper documents have become a popular source for the incorporation of all types of written communication (Kasturi et al. 2002; Marinai 2008a; Baird and Tombre 2014). Later, with the advent of computers and new types of storage technologies, along with an increasing demand for the preservation of precious/irreplaceable documents, digital versions of paper documents have been introduced, which are popularly called as 'document images' (Kasturi et al. 2002; Baird and Tombre 2014). Today we consider scanners, fax machines, mobile cameras and frame grabbers as important sources of acquiring document images (Jain 1989; Rehman and Saba 2014). In an endeavor to move towards the paperless office (Kasturi et al. 2002; Marinai 2008a), large volumes of printed documents are being digitized and archived as document images over the internet and over different image databases (Jawahar et al. 2004b; Lee and On 2011; Doermann and Tombre 2014). Since the manual processing of such a vast number of documents became a cumbersome task, a need was perceived to automate the process of Document Images Analysis (DIA). As a consequence, with the birth of the OCR in the 1950s, automatic reading and analysis of document images became feasible (Kasturi et al. 2002; Marinai 2008a; Jawahar et al. 2004b; Doermann and Tombre 2014). Since then, DIA has seen tremendous growth in the form of many successful applications like postal automation (Angadi 2007), form processing (Vasudev 2007), bank cheque processing (Marinai 2008a; Jayadevan et al. 2012), and invoice reading (Klein et al. 2004).

Digitally acquired document images generally occupy a large amount of storage space and thus require an excess of download/transfer time (Kia 1997; Gonzalez and Woods 2009; Miano 1999). Therefore, for efficient storage and transmission, document images are subjected to compression before being archived in digital libraries or when transmitted over the internet (Sayood 2012; Salomon et al. 2010). As a result, huge volumes of compressed documents get generated on a daily basis. In addition, for the same reason, the trend at present is to keep and store the documents in a compressed form for as long as possible (Adjeroh et al. 2013; Mukhopadhyay 2011). The only way available in the literature to work with compressed document images is through decompression. Hence, direct analysis and processing of compressed document images has now become a potential research issue in the DIA (Javed et al. 2015a, b). This novel idea of carrying out both operations and analysis directly in the compressed representation is called compressed domain processing (Javed et al. 2013; Adjeroh et al. 2013; Mukhopadhyay 2011; Kia 1997). An operation is said to be in the compressed domain, if the required analysis and processing are carried out with minimal decompression

or even without decompression of compressed data. In the recent literature, due to increased importance affixed to direct processing of compressed data (Mukhopadhyay 2011; Adjeroh et al. 2013; Lu and Jiang 2011; de Queiroz and Eschbach 1998; Kia 1997), many research works have been reported that anticipate opening up of a new vista in digital image analysis.

The current survey paper brings out two major contributions to the research community. First it presents a study on image analysis and image compression techniques and subsequently portrays the motivational reasons for pursuing DIA in the compressed domain. Second, it critically reviews the related techniques proposed in the literature based on the type of compression and the type of operations performed. In all, the paper provides a perspective for pursuing further research in the areas of image processing, pattern recognition and computer vision by employing directly the compressed data. Rest of the paper is organized as follows. In Sect. 2, we introduce document images and its related operations from the perspective of DIA. Further in the same section, we provide discussion on compressed representation of document images as a solution to the Big-data problem, and also mention the pros and cons of such a representation. We then differentiate document images from document files in Sect. 3, and also highlight in the same section the merits of pursuing research in the compressed domain. In Sect. 4, we provide a detailed literature review on document images from the perspectives of image analysis, image compression, and processing of images in the compressed domain. In Sect. 5 we give future research directions and in Sect. 6 we conclude the paper.

2 Document images and compressed representation

In this section, we introduce document images from the perspective of image analysis and image compression, and present the motives/reasons for pursuing document image analysis in the compressed domain.

2.1 Document images and analysis

In the phrase 'document image', the word 'document' is derived from the Latin word documentum which in Medieval Latin is referred to as a 'written instruction' or 'an official paper' that is normally used to communicate and store information (Javed 2016). Text is the main source of information communicated through paper documents which could be either in a printed or a handwritten form (Jawahar et al. 2004a, b; Jayadevan et al. 2012; Rehman and Saba 2012). The text contents in a document can also be accompanied by tables, graphs, images, etc., as additional source of information, which could be presented in different types of layouts (see Fig. 1; Antonacopoulos et al. 2009; Marinai 2008a; Ogier et al. 2009). Two such typical printed documents containing text and non-text formatted in double and triple column layouts are shown in Fig. 1, which are extracted from Prima Layout Analysis Dataset (Antonacopoulos et al. 2009) published in ICDAR2009. In an effort to move towards a paperless office, thanks to the contributions made by the DIA community in the past few decades (see the conference series of International Conference on Document Analysis and Recognition-ICDAR, since the year 1991), that today, automatic analysis and processing of such documents have become a reality (Kasturi et al. 2002; Marinai 2008b; Doermann and Tombre 2014). In real life we encounter document images in both printed and handwritten forms.

A decade ago, DIA was restricted only to static document images originating from scanners and fax machines (Jain 1989; Baird et al. 1992; Baird and Tombre 2014). However, with

Fig. 1 Two sample documents from the Prima Layout Analysis Dataset, **a** 00000676, **b** 00000263

advancements in computers and digital technology today, we find instances where document images are captured from mobile phones, digital cameras, and surveillance video cameras too (Jawahar et al. 2004b; Jain 1989). Therefore, with the rapid pace of technology, the problem of DAR (Document Analysis and Recognition) has spread to other domains too like natural scene image processing and video processing (Mukhopadhyay 2011). Since 2005, a separate workshop called CBDAR (Camera-Based Document Analysis and Recognition) has been dedicated to the processing of camera based document images in ICDAR. In addition, due to extensive usage of video cameras in generating document images, we also anticipate VBDAR (Video Based Document Analysis and Recognition) (Jathanna and Nagabhushan 2015) in the near future. Overall, we can say that the scope for DIA is increasing day by day. In the current research paper, the DIA on static compressed document images is reviewed; however, the issue pertaining to natural scene images and video frames are not chosen to be discussed.

DIA entails tools and techniques that allow the contents of the image to be analyzed through the computation of statistics and mathematical measurements using the pixel intensities present in the image (Marinai 2008b; Baird et al. 1992). Computing histogram, line profiles, correlation, and entropy could be a few such analyses related to document images (Marinai 2008b; Nixon and Aguado 2012; Ding et al. 2012). The DIA technique involves carrying out different operations on document images which can be grouped broadly into Textual processing and Graphical processing (see Fig. 2; Kasturi et al. 2002). The overall objective of DIA is to recognize the text and graphic components in the document image, and to generate the expected information required for further decision making.

In DIA, textual processing involves the development of OCR and page layout analysis techniques (Kasturi et al. 2002; Marinai 2008a). The job of OCR is to recognize and convert the text contents printed in the image to the ASCII form which eventually consumes less storage space and becomes convenient for editing and search operations (Doermann et al. 1998; Doermann 1998; Doermann and Tombre 2014). While the page layout analysis techniques complement OCR by feeding only the text contents that are automatically extracted from document images which involve a complex layout (Antonacopoulos et al. 2009; Baird et al.

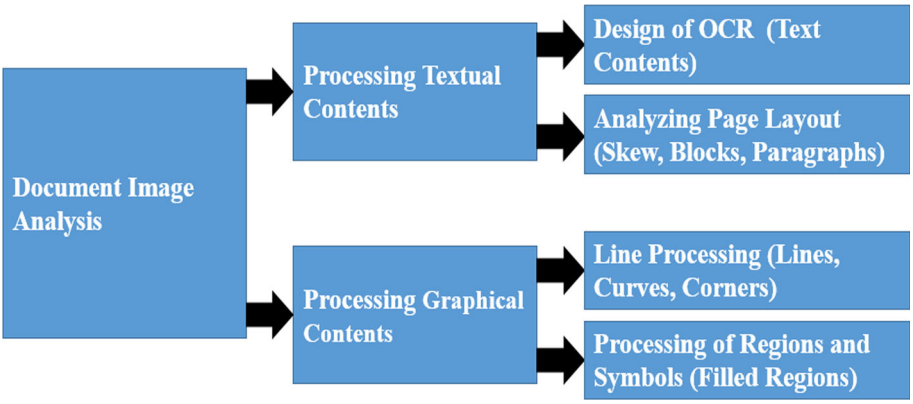

Fig. 2 Block diagram representing DIA techniques

1992), the technique also includes the detecting of skew, text blocks and paragraphs in the document. On the other hand Graphical processing involves the techniques for the detection and processing of non-text components such as logos, images, tables and graphs (Ogier et al. 2009). All of the operations in conventional DIA are carried over the uncompressed version of the document image. In the present research paper, the DIA techniques that operate directly with the compressed document images are reviewed.

2.2 Compression of document images

Digital images generally occupy a large amount of storage space, and hence consume more time during transmission and downloading (Sayood 2012; Salomon et al. 2010; Miano 1999). For example, the number of uncompressed colour images that an 8 Mega pixel camera can store on a 16 GB flash memory card is about 41 images of 24 Mega Bytes (MB) each (Gonzalez and Woods 2009). To elaborate further, consider the sample document image (00000676) shown in Fig. 1, whose size in the uncompressed mode is 21.6 MB. By compressing the document image with JPEG (Joint Photographic Experts Group) and TIFF (Tagged Image File Format), the file sizes are reduced to 1.24 MB and 131 KB (Kilo Bytes) respectively. This also implies that, with a typical modem of speed 56 kbps (7 kBps), the entire document can be transmitted in approximately 52.66 min in an uncompressed mode, 3.02 min in the JPEG mode, and just 18.71 s (seconds) in the TIFF mode. That's a big difference! Therefore, it has become very essential and the usual trend to compress digital images before archiving or communicating over the network, from the point of efficiency of both storage and transfer (Javed 2016; Kia 1997).

Image compression is a technique that is used to identify internal data redundancy and to subsequently come up with a compact representation, which occupies less storage space than the original image data (Sayood 2012; Salomon et al. 2010; Miano 1999). The reverse of this process is called decompression. There are different techniques of image compression available in the literature such as Run-length coding, Huffman coding, Arithmetic coding, Transform coding, Predictive coding, Sub-band coding, Vector quantization, and Wavelet transforms. A detailed review of these compression algorithms is provided in Sayood (2012), Salomon et al. (2010) and Miano (1999). In general, the compression algorithms can be classified into lossless and lossy types (Gonzalez and Woods 2009). Lossless algorithms compress the data in a lossless (without losing any information) manner and give back the

exact data when decompressed. Out of the above listed algorithms, Run-length, Huffman, and Arithmetic belong to a group of the lossless type of compression, and the rest come under the lossy type of compression (Sayood 2012; Salomon et al. 2010). Lossy algorithms refer to a loss of information through compression which cannot be recovered during the decompression stage.

In the literature, there are different compressed image file formats (Miano 1999) available, and among them, the popular ones are TIFF, PDF (Portable Document Format), JPEG, PNG (Portable Network Graphics), and BMP (Windows Bitmap) (Javed 2016; Sayood 2012; Salomon et al. 2010). Each file format has internal support for several types of compression algorithms as is generally found in TIFF and PDF. The file formats can be grouped into continuous tone still-image compression formats and bi-level still-image compression formats (Miano 1999). The following are the compression formats under the category of continuous tone still-images which are used specifically to compress natural scene images and video frames.

JPEG—A JPEG (Gonzalez and Woods 2009; Miano 1999; Mukhopadhyay 2011) is a standard format that is used to represent colour images of a photographic quality. It is a lossy baseline coding system that uses quantized Discrete Cosine Transformations (DCT) on 8×8 image block. It also uses Huffman and run-length coding during entropy encoding. It is one of the most popular methods for compressing images on the internet (Mukhopadhyay 2011).
JPEG-LS—It is a lossless to near-lossless standard for compressing continuous tone images. It uses adaptive prediction, context modelling, and Golomb coding (Sayood 2012) techniques.
JPEG-2000—This is an extension to JPEG for increased compression of photographic quality images (Sayood 2012; Salomon et al. 2010). It uses Arithmetic coding and quantized Discrete Wavelet Transforms (DWT). The compression can be lossy or lossless.
BMP—Windows Bitmap (color) (Sayood 2012; Salomon et al. 2010), is a file format used mainly for creating simple uncompressed images in Microsoft Windows platform.
GIF—Graphic Interchange Format (Sayood 2012; Salomon et al. 2010) is a file format that uses lossless LZW (Lempel–Ziv–Welch) (Adjeroh et al. 2013) coding for 1 through 8-bit images. It is used popularly for making small animation films and short low resolution films for internet applications.
PDF—Portable Document Format (Sayood 2012; Salomon et al. 2010) is a format representing 2D documents in portable devices. It has the capacity to incorporate the JPEG, JPEG2000, CCITT, and other compression algorithms. Some PDF versions are also recognized as ISO standards.
PNG—Portable Network Graphics (Sayood 2012; Salomon et al. 2010) is a file format that losslessly compresses full colour images. It produces a transparency (up to 48 bits/pixel) by coding the differences between each pixel value and a predicted value based on past pixels.
TIFF—Tagged Image File Format (Sayood 2012; Salomon et al. 2010) is a file format supporting varieties of image compression standards. They include CCITT, PackBits, JPEG, JPEG-LS and others.
ZIP—This is a compression format (Sayood 2012; Salomon et al. 2010) used for greyscale or colour images (smarter version of LZW). It is a lossless algorithm.

Following are the compression formats under bi-level image which are specifically used to compress black and white document images.

CCITT Group 3—CCITT (T.4-Recommedation 1985) stands for Consultative Committee for International Telephone and Telegraph, now known as ITU-T (Telecommunication Standardization Sector of the International Telecommunications Union). It is popularly used as a

facsimile (FAX) method for transmitting binary documents over telephone lines. It supports 1D (MH-Modified Huffman), 2D (MR-Modified Read) run-length and Huffman coding. TIFF with CCITT Group 3 compression is considered to be the universal benchmark for FAX and multi-line documents (Cvision Technologies 2015).
CCITT Group 4—An extended version of CCITT Group 3 (T.6-Recommendation 1985) standard supporting 2D (MMR-Modified Modified Read) run-length coding only.
JBIG or JBIG1—A Joint Bi-level Image experts Group standard (Salomon et al. 2010) for progressive, lossless compression of binary images. Context sensitive arithmetic coding is used and the image gets enhanced gradually from low resolution progressively as additional compressed data is added.
JBIG2—A replacement for JBIG1 (Salomon et al. 2010), for bi-level images in desktop, internet and FAX applications. The compression techniques used here are content based. The dictionary based methods are used for text and halftone regions, whereas the Huffman or Arithmetic coding are for other image contents. These can be either lossy or lossless.

The various important compression schemes and supporting compressed image file formats are listed out in Table 1. Also the Table 2 summarizes different compressed image file formats based on three modes of compression namely, uncompressed, lossy, and lossless. Depending on the application one can choose the file type and the file format.

With respect to document images, the popular compressed file formats available are BMP, TIFF, PDF, JPEG and JBIG (Javed 2016; Salomon et al. 2010; Miano 1999). Each format has an internal support for different compression algorithms. As observed from Tables 1 and 2, TIFF is a very flexible image format which supports all the three modes of image compression, and as a result it is one of the most popular formats when it comes to the compression of document images especially the text image data. PackBits (a form of run-length compression), CCITT Group 3 1D (MH), CCITT Group 3 2D (MR), CCITT Group 4 (MMR), and JPEG are some of the popular compression schemes embedded in the TIFF format (TIFF 1992; Javed 2016).

Run Length Encoding (RLE) and its variant algorithms constitute the backbone of document image compression in TIFF (TIFF 1992). Compression schemes such as RLE, Huffman coding, MH, MR, MMR, PackBits and JPEG encoding are all based on different forms of run-length data compression (Javed 2016). The performance of run-length based compression algorithms on a sample text document image (see Fig. 1a) is shown in Table 3.

From the Table 3, it can be observed that though the JPEG is a popular compression algorithm (in the internet world), the compression ratio with regard to text documents is very low in comparison to run-length based compression algorithms (Javed 2016). Moreover, JPEG is a lossy compression scheme suitable for compressing color images, and as a result it is the least preferred format when it comes to the compression of text document images (TIFF 1992; Javed 2016; Miano 1999). On the other side, JBIG is the recent document compression standard developed by CCITT now ITU-T (Salomon et al. 2010). Though JBIG gives a good compression ratio when used with text documents, it is still not popular for compression purposes because of many patent-related issues and implementation issues linked to JBIG technology (Salomon et al. 2010; Javed 2016). Moreover, JBIG uses Arithmetic coding internally which makes it less suitable for compressed domain processing (Kia 1997; Adjeroh et al. 2013), and JBIG is yet to find its place in popular image formats like TIFF.

From the study presented so far, it can be seen that most of the data compression techniques in the literature have been invented either from the perspective of efficiency of data storage and its transfer or are based on the nature of the compressed data which may be either lossy or lossless (Sayood 2012; Salomon et al. 2010; Kia 1997; Adjeroh et al. 2013). Less importance was given to the direct processing of compressed data during the compression process (Kia

Table 1 Different compression schemes and supporting compressed file formats

Compression	CCITT	JBIG1	JBIG2	JPEG	JPEG-LS	BMP	GIF	PDF	PNG	TIFF	JPEG-2000
Huffman	Yes		Yes	Yes							
LZW							Yes	Yes		Yes	
Arithmetic		Yes	Yes								Yes
Run-length	Yes		Yes	Yes		Yes		Yes		Yes	
Symbol based			Yes								
Bit plane		Yes									Yes
Predictive			Yes	Yes	Yes				Yes		
Wavelet											Yes

Table 2 Compressed image file formats in different compression modes

File type	File format
Uncompressed mode	TIFF, BMP
Lossy compression mode	TIFF, JPEG
Lossless compression mode	TIFF, PNG, GIF, RAW, ZIP, CCITT, JBIG2

Table 3 Effect of different document image compression algorithms on a sample document image (shown in Fig. 1a)

	File format	Compression	File size	Compression ratio
1	TIFF	NO	21.6 MB	1
2	JPG	JPEG	1.24 MB	17
3	BMP	RLE	925 KB	23
4	TIFF	PackBits	331 KB	65
5	TIFF	CCITT Group 3 (1D)	196 KB	110
6	TIFF	CCITT Group 3 (2D)	168 KB	128
7	TIFF	CCITT Group 4 (2D)	131 KB	165
8	JBG	JBIG	195 KB	110

1997). However, in the present day scenario due to the availability of a large volume of documents in a compressed form, it has become crucial to think of providing direct access to the objects present in the compressed data. In this direction, the literature has made a few initial attempts to develop operation-aware compression schemes in document images (Kia 1997; Garain et al. 2006a, b), which are friendly enough to support a few pre-defined operations in the compressed data. To the best of our knowledge, the first attempt to develop an operation-aware compression scheme was made by Kia (1997). The researcher(s) have presented a customized symbol-based compression scheme which is similar in principle to JBIG compression. Here, the compression algorithm is made intelligent so as to handle compressed domain operations. The novel idea of developing an object-aware compression scheme was standardized in the form of JPEG2000 by the Joint Photographic Experts Group committee in the year 2000 and it is based on Wavelet compression (Salomon et al. 2010; Kia 1997). The other discussions related to compressed domain processing are presented in the upcoming sections.

2.3 Document images as a Big-data problem and compression as a solution

Big-data is the problem related to handling enormous amount of data that are generally characterized by 3 V's (Volume, Velocity, and Veracity) (Javed 2016). According to the statistics published by United Nations Economic Commission for Europe (UNECE), on behalf of the international statistical community, it identifies three major sources of Big-data such as Social Networks, Traditional Business Systems and Websites, and also Automated Systems which generate huge amounts of data in the form of texts, images, audios and videos. As per the recent statistics by UNECE, the data growth is going to become exponential in the coming years as shown in Fig. 3 (Javed 2016). Therefore, due to the predicted exponential growth of data, the efficient handling of Big-data, its processing, storage and transmission

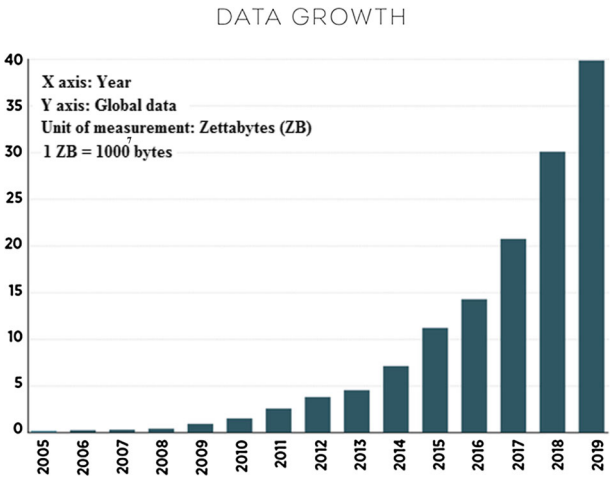

Fig. 3 Exponential growth of Big-data predicted by UNECE

are all becoming issues seen as a challenge and are the main focus in the present day scenario (Javed 2016).

In the literature, data compression techniques have been used extensively to overcome the volume aspects of Big-data (Mukhopadhyay 2011; Salomon et al. 2010; Javed 2016). As observed from Table 3, the compression techniques make a big difference in reducing the volume of uncompressed data (for the sample text document in Fig. 1a 99% data reduction is achieved). The overall objective of data compression is to transform big size data to small size data for efficient storage and transmission. The data compression techniques not only make the Big-data space efficient, but also communication efficient (download/upload); this is because transferring of a large volume of data over the internet or personal networks is expensive in terms of time and bandwidth (Kia 1997; Miano 1999). Therefore, in order to maintain the efficiency of Big-data (texts, images, audios and videos) different compression strategies have been invented in the literature of data compression (Anantharaman 2001; Sayood 2012; Salomon et al. 2010).

In the year 2012, it was reported by Venter and Stein (2012) that images and image sequences (videos) make up about 80% of all corporate and public Big-data. Since compression techniques have become an integral part of image and video acquisition systems, we can today confidently state that more than 80% of Big-data consists of compressed images and videos that are spread across digital libraries, internet and personal databases (Javed 2016). Therefore an automatic mode of analysis of compressed images and videos is becoming a challenging research issue in Big-data. This is because of the traditional method of processing of compressed data, which is through decompression that indents additional computing resources and demands a large buffer memory (Mukhopadhyay 2011; Adjeroh et al. 2013; Javed 2016). Thus for a huge volume of Big-data, decompression based processing may become very expensive in terms of time and space in the near future. Since compressed document images are also an integral part of the Big-data problem, the research study on direct processing of compressed document images is also very much relevant from the Big-data perspective (Javed 2016). In today's scenario, with the rapid growth of digital libraries, e-governance, and internet based applications, a huge volume of document images are being archived and communicated in a compressed form (Mukhopadhyay 2011). The discussion

of the issues related to compressed domain processing of natural scene images and videos is not intended to be a part of this current research study; however, it is touched upon briefly in the Sect. 4.

2.4 Limitations of the compressed representation of document images

No doubt, in the literature, data compression techniques have been considered to be one the ways to overcome the volume aspects of Big-data, especially the issues related to data storage and transmission; but there are also some limitations in having data in the compressed form which are projected hereunder (Javed 2016). The overall objective of data compression is to transform big size data into small size data, so that the data gets handled efficiently (Miano 1999). On the one hand, data compression techniques overcome the issue of data storage and transmission; but on the other hand, they make the processing of compressed data (small size data) expensive in terms of time and space (Mukhopadhyay 2011; Adjeroh et al. 2013). This is because of the traditional way of processing compressed data that is carried out through decompression, which implies that in order to operate on the compressed data, the data needs to be decompressed first, and then processed, which requires additional memory and computing resources (Kia 1997; Javed 2016). Therefore, the traditional model of decompression-based processing is generally considered to be expensive (Mukhopadhyay 2011; Adjeroh et al. 2013).

From the perspective of the Big-data problem, decompression implies that the compressed data (small size data) is again brought back to the state of Big-data (large size data), which is not desirable. Therefore, the issue of volume in the Big-data problem still remains unsolved when it comes to the processing of compressed data via decompression (Javed 2016). From the viewpoint of storage and transmission, decompression is expensive in terms of requirement of memory for the decompressed data, and further, once the decompressed data is processed, it needs to be re-compressed to make it storage and transmission friendly (Kia 1997; Mukhopadhyay 2011; Javed 2016). Therefore, in a decompression based processing model, the compressed data needs to repeatedly go through the stages of compression and decompression, depending upon the number of times the compressed data requires processing.

Consequently, based on the different issues highlighted in this section, the overall concern in the present research theme is to keep the data in its compressed form and yet be able to directly analyze and process the data without the involvement of the decompression stage. In DIA, OCR (Baird and Tombre 2014; Baird et al. 1992) is used to solve many problems related to document images such as searching, editing, indexing and retrieval (Doermann et al. 1998; Doermann 1998). However, applying the existing OCR on a compressed document requires the data to be in an uncompressed form, which is possible only when the compressed data is decompressed. Therefore, decompression becomes an additional burden while using OCR on compressed documents (Javed 2016). This issue has motivated the researchers to investigate for a decompression-less and OCR-less model of processing compressed document images which is discussed in Sect. 4.

3 Document images vis-a-vis document files

The dictionary meaning of the word ‘document’ is a piece of written, printed, or electronically typed matter that provides some information or evidence or serves as an official record. Though in the present day, we use documents in both printed and electronic versions, paper

continues to be the preferred medium of communication, due to the ease in navigation and adding of annotations. The electronic version of the document is called document file or ASCII file, which requires less memory, and supports editing and searching operations with a minimum effort (Adjeroh et al. 2013; Lloret and Palomar 2012; Gambhir and Gupta 2016). When the same document file is printed or typed on a paper, we get a document in the paper form. Any operation or information extraction from the paper document requires manual processing which is considered a laborious task when it involves huge volumes of paper documents (Jawahar et al. 2004a; Javed 2016). Therefore in real time, computers are the preferred choice for automating the process of document analysis (Marinai 2008a). Computers cannot directly handle paper documents: hence, documents have to be digitized with the help of scanners or fax machines (Jain 1989). A paper document in the digital form is called a document image. A document image generally requires more storage space than a document file (Salomon et al. 2010; Adjeroh et al. 2013). For example, a typical document page which in ASCII form can be easily stored in 2–3 KB requires a storage space between 500 KB and 2 MB (Kia 1997).

In the current digital era, due to an excessive use of digital technology, a huge amount of data is being generated in the form texts and images (Venter and Stein 2012) through corporate, social media, and the world of the internet. The text-contents are the most informative component in document files and images. Moreover, the text contains a rich source of information in text books, journals, newspapers, magazines, legal documents, and so on (Baird and Tombre 2014). Therefore, the automatic extraction and analysis of text in both files and document images has been an important research issue since many decades (Adjeroh et al. 2013). Automatic analysis of text in files includes operations such as searching, editing, indexing, classification, text mining, information extraction, and text retrieval. A detailed survey of different operations and techniques related to text files are available in Salton (1988), Berry (2013), Lloret and Palomar (2012) and Gambhir and Gupta (2016). In the literature, many strategies have been evolved to solve the problem of conventional pattern matching in text files. The issue of exact pattern matching (Faro and Lecroq 2013; Crochemore et al. 2007; Adjeroh et al. 2008) and inexact pattern matching (Navarro 2001) are reported in detail. The other variants of pattern matching with swaps (Lee et al. 1997), with fusion (Adjeroh et al. 1999), with don't cares (Akutsu 1994), with scaling (Amir et al. 1992), with rotation (Amir et al. 2006), multiple patterns (Aho and Corasick 1975; Amir and Calinescu 1996), super patterns (Knight and Myers 1999), multi-dimensional patterns (Giancarlo and Gross 1997), parameterized pattern (Fredriksson and Mozgovoy 2006) have also been attempted. In case of compressed pattern matching, the key works reported are on LZW (Lempel–Ziv–Welch) (Amir et al. 1996; Adjeroh et al. 2013), LZ77 (LZ refers to Lempel–Ziv coding and 77 refers to 1977, the year when this coding was introduced) (Farach and Thorup 1995; Salomon et al. 2010), hybrid approach using LZ77 and LZ78 (LZ78 refers to advanced version of LZ77 coding introduced in the year 1978) (Navarro and Raffinot 1999, 2004; Salomon et al. 2010), BWT (Burrows Wheeler Transform) (Bell et al. 2002), Grammar based compression (Kieffer and Yang 2000; Yang et al. 2001), Run-length encoding (Eilam-Tzoreff and Vishkin 1988; Apostolico et al. 1997; Bunke and Csirik 1993, 1995; Adjeroh et al. 1999), and Huffman codes (Mukherjee and Acharya 1994; Moura et al. 2000; Ziviani et al. 2000).

However, extracting information from a document image is not quite simple as has been found with document files (Jawahar et al. 2004a; Adjeroh et al. 2013; Doermann and Tombre 2014). Before the document image is ready for analysis, it has to go through different stages such as pre-processing, feature extraction, segmentation, and finally recognition or analysis (Kasturi et al. 2002; Marinai 2008a). The invention of OCR was a major breakthrough in DIA, and its prime task was to transform the printed text image into a printed text file (Doermann

1998). A detailed survey on OCR and other automatic document image processing techniques with respect to uncompressed documents are discussed in Tang et al. (1996), Kasturi et al. (2002) and Doermann (1998). In today's scenario as a huge volume of text document images are being generated on a daily basis in digital libraries, e-governance, and internet based applications, the trend at present is to store these documents in a compressed form for the purposes of efficiency in storage and transmission. Therefore automatic processing and analysis of texts in compressed document images is becoming an important research issue (Mukhopadhyay 2011; Adjeroh et al. 2013).

4 State of the art

In this section we present a detailed review of the literature on document images from three perspectives- image analysis, image compression, and image processing in the compressed domain.

4.1 Document image analysis

The objective of DIA is to automatically analyze and extract different kinds of information from the document images (Marinai 2008a, b; Baird et al. 1992; Baird and Nagy 1994). The different stages generally identified in any typical DIA system are pre-processing, segmentation, recognition, and post processing (Kasturi et al. 2002; Marinai 2008a; Rehman and Saba 2014). The pre-processing stage improves the quality of the image, and increases the possibility of accurate detection of the objects of interest. The segmentation stage detects and extracts the objects from the image. The recognition/classification stage identifies the objects of interest and categorizes them into certain classes. The post-processing stage involves techniques that actually boost the accuracy and performance of DIA applications. All these stages involve feature extraction (Ding et al. 2012) which exhibits the distinctive features or characteristics of the document images. In the literature, many operations have been defined which tackle various issues arising through the stages of DIA, from among them a few important ones are listed out in Fig. 4. A detailed survey is available in Kasturi et al. (2002), Marinai (2008a) and Ding et al. (2012). DIA involves the processing of both handwritten and printed document images.

The important operations involved in pre-processing stage are filtering, geometrical transformations, and thinning (Nixon and Aguado 2012; Marinai 2008a). Image filtering gives an image where each pixel has a value that is influenced by the pixels in its neighborhood in some strategy or the other. Important filtering operations include binarization (Lu et al. 2010; Bolan 2012), noise removal (Farahmand et al. 2013), and image enhancement (Bolan 2012; Hendahewa 2010). Geometrical transformations (Saragiotis and Papamarkos 2008; Na and Jinxiao 2011; Shiraishi et al. 2013) include operations such as detection and correction of skew in the document image. A thinning operation (Gonzalez and Woods 2009; Kasturi et al. 2002) is used to compute the features based on the skeleton of the image components. Overall, the pre-processing stage improves the quality of the image, and helps in the escalation of performances of the subsequent stages and eventually of the entire system itself.

Any typical printed document image may contain texts, images, tables, graphs, etc (Namboodiri and Jain 2007; Ogier et al. 2009). Therefore, the first task in any document image analysis is to understand the layout of the document, namely the physical layout and the logical layout (Marinai 2008a). Segmented page layout is called the physical layout and the corresponding labeled page layout is called the logical layout. Page segmentation techniques

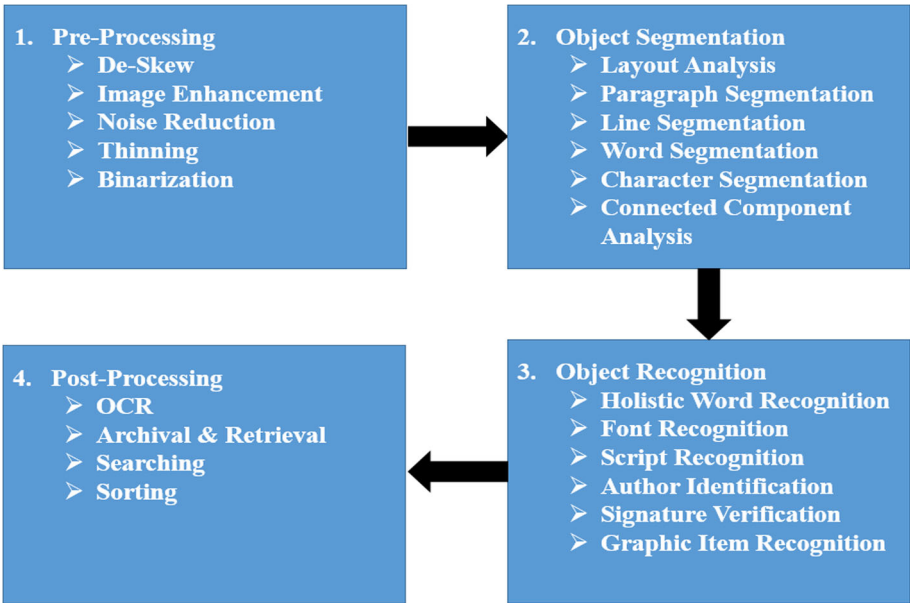

Fig. 4 Important operations through the different stages of a typical DIA system

in the literature can be grouped into bottom up and top down type approaches (Namboodiri and Jain 2007). The recent developments in the area are reported in Zirari et al. (2013), Chen et al. (2013) and Breuel (2003). The bottom up approach (Zirari et al. 2013) involves repeated aggregation of lower level components to achieve segmentation, whereas in the top down approach (Chen et al. 2013; Breuel 2003), to begin with larger components are detected initially, and then broken down into smaller components to the required level. Once the page is segmented, the important features are next extracted from the segmented components, and their logical meaning is then attached to them (Nagy et al. 1992; Chen et al. 2013); for example the decision regarding text or non-text is done here. The other aspects in page segmentation are segmentation of text into paragraphs, lines, words and characters which are very critical in deciding the performance of the OCR (Rehman and Saba 2012; Dash et al. 2016). A detailed survey on text segmentation and recognition issues are reported in the research work of Nagy (2000) and Marinai et al. (2005). The other important research concerns related to text document images are script identification (Ghosh et al. 2010; Dhandra et al. 2006), font detection (Ramanathan et al. 2009; Slimane et al. 2013), reading order determination (Meunier 2005; Yucun et al. 2010), document classification (Chen and Blostein 2007), etc. Another important application in text document images is word spotting, that involves the searching of specified keywords within large repositories of document images. Today word spotting (Ahmed et al. 2017) is a challenging problem in historical documents (Rath and Manmatha 2007), degraded documents (Yong et al. 2010), and handwritten documents (Wshah et al. 2012a, b; Ahmed et al. 2017). A detailed survey on keyword spotting techniques is available in Murugappan et al. (2011) and Ahmed et al. (2017). On the other hand, processing of graphical components present within the document image is also an important aspect of DIA. A detailed work on graphics recognition can be traced from the series of workshops on GREC (Graphics RECognition) (Ogier et al. 2009) in ICDAR conferences.

All through the stages of DIA, the different features involved need to be extracted from the document images in order to carry out the desired operation (Nixon and Aguado 2012; Marinai 2008b; Ding et al. 2012). The feature extraction techniques reported in the literature can be grouped broadly into three classes (Nixon and Aguado 2012) namely, the low level features, the shape based features, and the flexible shape based features. The low level features are those features that can be extracted without having the shape information sourced from the image (Nixon and Aguado 2012; Ding et al. 2012). All the point based features (projection profiling (Javed et al. 2013), entropy (Javed et al. 2015b), histogram, template operations, thresholding, correlation (Javed et al. 2015a)), the edge based, the line based, the curvature based, and the motion based features come under the category of low level features (Javed 2016; Nixon and Aguado 2012). The shape based features are high level features which include techniques such as image subtraction, image thresholding, intensity template, binary templates, and Hough transform (Nixon and Aguado 2012). The flexible shape features include techniques such as the deformable templates, the snakes, the symmetry operations, and the active shape models (Nixon and Aguado 2012). The different DIA techniques discussed so far have the limitation of working only with uncompressed document images. This implies that the techniques proposed may not be directly suitable for carrying out operations on compressed document images. However, in the present research paper compressed domain techniques related to feature extraction, segmentation, text-font characterization and word spotting directly in compressed document images are reported.

4.2 Document image compression

In the literature, we have many types of compression techniques proposed which are used to compress images either with redundancy reduction or lossy coding techniques so as to achieve a high compression ratio. The redundancy reduction methods are run-length coding and Huffman coding (Sayood 2012; Salomon et al. 2010; Miano 1999), where the compression in run-length coding is achieved by replacing the sequence of similar elements with the value of the element and its count, and in Huffman coding, the frequently occurring elements in the image are substituted with shorter codes that are dynamically computed after going through the whole data once (Sayood 2012). The redundancy reduction methods are of a lossless type of compression, and are the most preferred in the compression of text document images. On the other hand, lossy coding methods are lossy (incurring loss of information) in nature due to which a high compression ratio is achieved, and are usually preferred in compressing scene images and image frames (in videos) (Miano 1999; Sayood 2012). The different types of compression algorithms under this category are Transform based coding (Jain 1989), Sub-band coding (Vetterli 1984; Woods and O'Niel 1986), Predictive coding (Habibi 1977; Pirsch 1982), Vector quantization (Lim 1990), and Wavelets (Mazzarri and Leonardi 1995).

The transform based coding popularly uses Fourier transforms (Jain 1989), Sine/Cosine transforms (Ahmed et al. 1974; Jain 1989), KL transforms (Andrews 1970; Jain 1989), etc. The coding techniques are pixel based, and they involve the extraction of an orthonormal basis to get a high level of compression. The technique is made lossy by coding only the high frequency elements and discarding the low frequency elements. In Sub-band coding (Vetterli 1984; Woods and O'Niel 1986), the technique of coding is similar to transform coding, with just one difference, that is, the different frequency elements are treated at different resolutions. The concept of the Predictive coding (Habibi 1977; Pirsch 1982) method is based on the prediction of the next element by using the knowledge of the previous element and the error between the prediction and the given sample element. The Vector quantization (Lim 1990) based methods achieve compression by dividing the image into different blocks which are

then transformed into vectors having a higher dimension. Each vector is then treated as a sample class in the form of a prototype which actually resembles the vectors that belong to this class. However, we at present have compression algorithms based on Wavelets (Mazzarri and Leonardi 1995) which have higher compression capabilities. Fractal coding (Zhang et al. 1995) is one such example that is based on Wavelets. Though all these compression methods have a good compression ratio with respect to scene images, they are the least preferred when it comes to compressing text document images. This is because when text based document images are compressed using lossy techniques, they tend to lose information in the presence of low resolution and degradation. Therefore, the lossless compression algorithms are preferred while compressing text document images, and today we have many such standard algorithms that are specifically used for document images (Sayood 2012; Salomon et al. 2010).

Text contents in a document image can be losslessly represented and reproduced using a black and white pixel image (Javed 2016). The important characteristic of a text document is that it contains a large amount of continuous black and white pixels. A continuous sequence of similar pixels is known as a run. Therefore in a text document image, we get many such sequences of white and black pixel runs. These pixel runs were observed by CCITT (T.4-Recommedation 1985; T.6-Recommendation 1985) and a new standard of compression was introduced in the form of CCITT Group 3 1D (Modified Huffman) that specifically targeted documents originating from FAX machines and digital libraries. Run based features also form the backbone of many text document compression algorithms like Run-Length Encoding (RLE), PackBits, Huffman, Modified Read (MR), and Modified Modified Read (MMR) which are widely supported by TIFF and PDF image formats (Salomon et al. 2010).

In the literature, we find compression algorithms like JBIG and JPEG that are also used in compressing document images (Salomon et al. 2010; Sayood 2012). However, these compression schemes are not as popular as TIFF for the following reasons. Though JPEG gives good compression ratio with color images, the compression ratio with binary text documents is low (see Table 3) and also lossy in nature, and hence they are usually not preferred for the compressing of text documents (Javed 2016). On the other hand, we have JBIG which gives a good compression ratio with binary text documents, but the compression scheme is not very popular presently, and it is yet to find its position in the TIFF image format (Javed 2016). An experimental analysis is reported in Table 3 for a sample printed text document in Fig. 1 that clearly shows the performances of different compression algorithms.

We have recently witnessed a few document image compression schemes that are based on the repetitive nature of text components (Ascher and Nagy 1974; Inglis and Witten 1994; Kia 1997). The coding scheme (Ascher and Nagy 1974) uses symbol level coding instead of the pixel level found in most of the previous methods. The method creates a catalogue of characters that are present in the document image; and any new character that is encountered gets added to the library consisting of a collection of character-images. By using such a created library, compression can be attained. This compression was further enhanced by Inglis and Witten (1994), where they used pattern matching approaches to extract the unique image components from the text document, then compressing the indexed components to achieve a high compression ratio, and finally coding the residues to achieve lossless compression. In the same direction, (Kia 1997) developed a new compression algorithm that promoted compressed domain processing based on the symbols present in the text document image. The method consists of three stages such as symbol prototype generation, symbol coding, and residue coding. With prototype images and component location table, one can generate the lossy version of the document; however, the residue coding could be used to get a lossless version. In the next section we review different techniques that have been developed to operate directly with compressed document images.

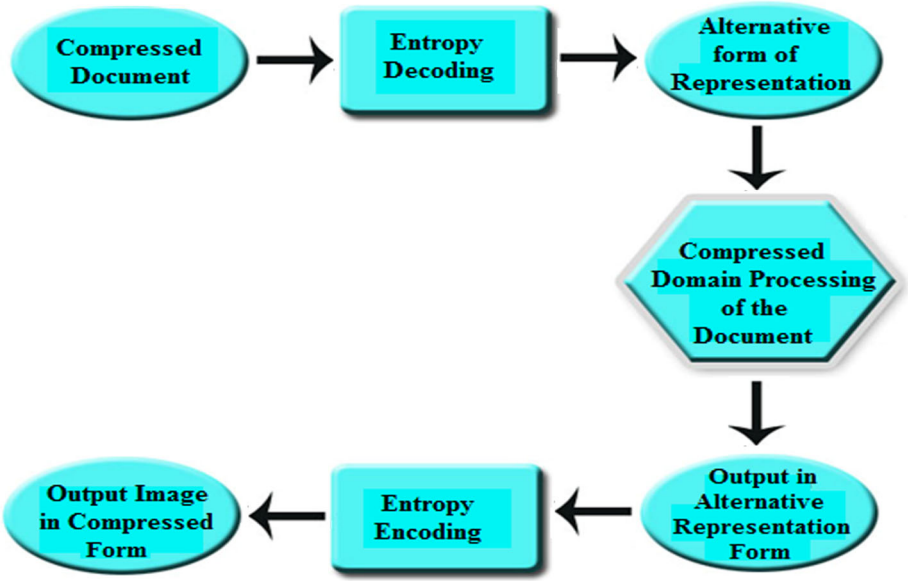

Fig. 5 A general model of processing a compressed document image in compressed domain

4.3 Processing of document images directly in the compressed domain

With an excessive use of the internet, computers, and personal gadgets, we are witnessing an explosion of information in the form of texts, images, audios, and videos (Bell et al. 2002). For efficacious storage and transfer, the present day trend is to keep the data in a compressed form (Mukhopadhyay 2011). Since, more and more data is being made available in the compressed form; there is a need for direct analysis of the compressed data without involving the stage of decompression. This novel idea of operating directly on the compressed data without undergoing the stage of decompression is known as Compressed Domain Processing (CoDoP) (see Fig. 5) (Javed 2016; Mukhopadhyay 2011; Adjeroh et al. 2013; Kia 1997).

The problem of CoDoP was introduced initially in the field of text data (ASCII files) by Amir and Benson in the year 1992. Their whole idea was to search for a pattern directly within the compressed text data without the use of decompression, and reporting the number of occurrences of the pattern in the file (Amir and Benson 1992). In recent days, compressed domain methods are gaining increased momentum as the amount of data being handled is very large, and the pattern matching techniques are comparatively faster than decompression based methods (Adjeroh et al. 2013). Based on this research theme, many interesting articles and books have been published in the area of compressed texts (Adjeroh et al. 2013), images and videos (Mukhopadhyay 2011). However, in this section, we focus mainly on compressed domain processing techniques related to document images, and just touch upon the issues related to texts, scene images and videos briefly.

A general model of processing document images in the compressed domain is shown in Fig. 5 (Mukhopadhyay 2011). In the context of document images, different compressed domain methods have been proposed in the literature, based on the significance and the type of data being compressed. For processing text document images, CCITT Group 3 (Javed et al. 2013, 2015a, b, 2016), CCITT Group 4 (Spitz 1998; Hull and Cullen 1997; Lu and Tan

2003b) and JBIG (Kanai and Bagdanov 1998; Regentova et al. 2005) based models have been proposed. For processing grayscale documents, compressed domain models in JPEG (de Queiroz 1998) and symbol based compression (Kia 1997) schemes have been proposed. Detailed remarks regarding these models are given in the following deliberations.

There are plenty of efficient coding schemes available for the compressing of document images. Among them CCITT Group 3 1D (MH), CCITT Group 3 2D (MR), CCITT Group 4 2D (MMR), JBIG, run-length (RLE), and JPEG compression standards are popular for dealing with document images (Kou 1995; Salomon et al. 2010). CCITT is the most preferred compression algorithm for bi-level document images that are widely supported by TIFF and PDF formats. CCITT Group 3 and CCITT Group 4 compressions are used in all facsimile transmissions by modem over analog telephone lines, as well as in digital storage and transmission of documents. Due to its popularity, researchers became interested in working directly with these compressed documents (Latifi and Kanai 1997; Javed 2016). Many initial attempts have been reported in the literature of the compressed domain to perform different operations using these documents which are discussed in the following paragraph.

The first group of work reported are on the CCITT Group 3 and CCITT Group 4 documents. In CCITT Group 3 there have been attempts to extract features like projection profile, run-histogram, and entropy features (Javed et al. 2013). Further, the image processing applications like text segmentation and word spotting have also been successfully attempted (Javed et al. 2015a, b, 2016). The proposed run-length compressed domain model is general and operates directly on the run-length compressed data irrespective of the compression scheme applied for compression. The model is also extended to CCITT Group 4 documents by providing the option of generating run-length data from CCITT Group 4 (Javed et al. 2016a) compressed data. The initial work by Spitz (1998) describes the detection of dominant skew, multiple skew angles, and presence of logotype in compressed documents. It uses the Baird technique (Baird 1987), where the fiducial points are found by using the pass modes in a CCITT Group 4 file. A fiducial point is the bottom center of the bounded box surrounding the connected component. Here, the bottom positions of black and white structures in the compressed image are extracted, and their alignment is determined with the help of peak strength. This technique is extended further so as to detect the logotype which produces a unique alignment signature at different angular distributions in the compressed representation. A new compression technique derived from CCITT Group 4 is proposed in Deng et al. (1998, 1999) for the purpose of skew detection and correction, and the extraction of the connected components in binary documents. The authors claim that their method is efficient in compression and is also flexible for operating in the compressed domain with good speed-up in comparison to the traditional method.

With respect to CCITT Group 4, some interests are also seen in finding a similarity in documents in the compressed domain. The works reported by Hull (1997) and Hull and Cullen (1997) extract pass codes from a rectangular patch of text and use the modified Hausdorff distance measure to determine similarity in two documents. Later in the work of Hull (1998) and Lee and Hull (2001), it was observed that the extraction of pass codes was not cost effective, and therefore a new feature based on up and down endpoints were proposed for document matching which led to promising results. Further, the problem of document similarity in compressed documents was extended to applications like word searching (Lu and Tan 2003b), document retrieval (Lu et al. 2001; Lu and Tan 2003a), and pattern searching (Maa 1994). A word searching technique for CCITT Group 4 compressed document had been devised based on the features extracted from the black and white pixel changing elements in the compressed file. In the subsequent steps, using the connected components and the word bounding boxes across words, a dissimilarity measure was evolved using which word

searching was attained. The same concept was extended to document retrieval of CCITT Group 4 compressed images (Lu and Tan 2003a). The document similarity here is based on scalar product of document vectors, and the weighted Hausdorff distance is used for the purposes of classification and retrieval. Another interesting work by Maa (1994) on the CCITT Group 4 was to detect the existence of bar codes using the features of the vertical mode coding standard. Based on the repeated sequence of vertical modes present within the compressed file, the existences of bar codes were detected. Further, an algorithm with reduced operations for detecting connected components in CCITT Group 3 2D and CCITT Group 4 2D has been reported by Regentova et al. (2002). The idea of developing an OCR for compressed documents using CCITT Group 3 2D is discussed in Marti et al. (2000). Finally, in the case of CCITT Group 3 1D, a formal model for processing compressed documents is defined in Latifi and Kanai (1997). They use the finite state transducer to perform point operations, and push down transducer for algebraic and spatial transformations. However, it should be noted that most of the compressed domain techniques proposed on CCITT Group 3 and CCITT Group 4 documents use compressed data to extract the location of coded pixels and to generate a prototype of the document which contains fewer pixels in comparison with the actual document image (Lu and Tan 2003a; Javed et al. 2016). This implies that the actual compressed domain operation is not carried out directly on the compressed data but on the uncompressed version of the document.

A few other works which are related to CCITT are based on run-length information extracted directly from uncompressed binary images. A run based connected component detection technique is discussed in Ronse and Devijver (1984), and is also based on the coordinate information of runs. The work by Shima et al. (1989, 1990) extracts connected components by sorting and tracking of runs in each scan line and then labelling the connected runs. The work in Hinds et al. (1990) uses the run length encoding guided by the Hough transform to locate the skew angle present in the run length coded binary image. Recently, the work by Breuel (2008) showed the morphological operations on run-length representations.

JBIG (Salomon et al. 2010) is the binary image compression standard which was recently adopted by CCITT (Kanai and Bagdanov 1998). JBIG supports a lossy model of compression, and provides a strong support for progressive transmission. In the context of JBIG compressed images, the following research works are reported in the literature (Kanai and Bagdanov 1998; Regentova et al. 2005). Similar to the work reported in Spitz (1998), it also uses the concept of extracting fiducial points for skew detection during the decoding stage of JBIG compressed images. Further, with the help of the projection profile, the skew angle is determined. The angle of projection within a search interval that produces maximum alignment with respect to fiducial points is the skew angle. In Regentova et al. (2005), the researchers propose an interesting work with JBIG compressed documents to achieve document layout analysis. They employ the knowledge of JBIG encoding process, and perform horizontal smearing followed by connected component extraction. They use a hybrid strategy where top-down analysis results in structural layout while the bottom up analysis yields the connected components. Then the simple geometrical features extracted from the connected components are used to classify the document into text and non-text blocks. Further, they have extended their layout analysis for detecting the form type and the form dropout.

By using JPEG (Miano 1999) compressed documents, segmentation of different regions has been demonstrated by de Queiroz and Eschbach (1997, 1998). JPEG images have been used to perform scaling, previewing, rotating, mirroring, cropping, re-compressing and segmentation in the work of de Queiroz (1998). The whole idea was to have an encoding map (ECM) for the entire image in terms of the encoding cost over each block. The ECM, along with row-sums and decimated DC maps are embedded inside the JPEG compressed stream

which results in an increased file size. The ECM, which provides an important activity in the block, is decoded to extract just the informative blocks, so as to carry out different operations like segmentation and image cropping. The ECM alone is not sufficient to get good results: therefore, the other details such as DC coefficients are preferred to get better results. In case of JPEG2000, the merging of two different regions from two different image compressed streams is demonstrated in Rosenbaum and Taubman (2003). They use different synchronization codes to mark the code blocks of different regions. Based on the synchronization codes extracted from the compressed stream, region merging has been achieved. Another region segmentation method on compressed gray images using quadtree and shading representation is given in Chung et al. (2004).

An attempt was also made by Kia (1997), to develop a new symbol based compression scheme having the ability to perform compressed domain operations such as keyword searching, sub-document retrieval, skew estimation, and document matching. The compression scheme utilizes the repetitive nature of the text components, and is a symbol based coding scheme. The compression scheme generates a prototype map of the text under compression, the component layout, and the residual map. Using the prototype map and component layout, a lossy version of the image can be generated; however, to get a lossless image, the residual map needs to be decoded.

We have so far discussed different compressed domain methods related to just the processing of compressed document images. However, the literature also has compressed domain techniques proposed for JPEG images, video frames and audio data. We do not intend to discuss these issues in detail, but to just touch upon them in brief.

JPEG (Miano 1999) is the most widely used image compression algorithm which uses Discrete Cosine Transform (DCT). Many algorithms have been developed to work in the DCT compressed domain. Operations such as image filtering (Kresch and Merhav 1999; Yim 2004), image enhancement (Lee 2007; Tang et al. 2003; Mukherjee and Mitra 2008), image resizing (Dugad and Ahuja 2001; Martucci 1995; Merhav and Bhaskaran 1997; Mukhopadhyay and Mitra 2009; Mukherjee and Mitra 2006), and image transcoding (Viswanath 2009; Viswanath et al. 2010) have been reported. The image editing operations such as digital watermarking (Lu 2002; Hernndez et al. 2000; Chen and Wornell 2001), steganography (Avcibas et al. 2005; Chen and Wornell 2001; Provos 2001), indexing (Reeves et al. 1997; Shneier and Mottaleb 1996; Smith and Chang 1994), face detection and identification (Wang and Chang 1997; Chua et al. 2002), and image registration (Lee et al. 2005; Ito and Kiya 2007) are also reported in JPEG compressed domain. Edge enhancement of remote sensing images and image enhancement without decompressing the image are reported in Chen et al. (1999) and Tang et al. (2003). Text and image sharpening is achieved in Bhaskaran et al. (1997) by scaling each element of the encoding quantization table which enhances the image. A set of algorithms for manipulating compressed images like performing algebraic operations, scalar addition, scalar multiplication and pixel wise addition or multiplication is demonstrated in Smith and Rowe (1993). Operations like image shearing and rotation in the compressed domain are seen in Chang et al. (1992), Chang (1995a, b). Using the algebraic operations, the inner block manipulation and scan line methods in the compressed domain were developed by Shen and Sethi (1995, 1996). Image filtering, which is one of the common operations used in the spatial domain has been applied on compressed images by Chiptrasert and Rao (1990), Ngan and Clarke (1980), Lee and Lee (1992) and Mukherjee and Mitra (2006). Algorithms for image translation, overlapping, scaling, linear filtering and pixel multiplication which have been applied to image subtitling, anti-aliasing and masking are discussed in Chang et al. (1992) and Chang and Messerschmitt (1995). Direct feature extraction is achieved in Scotney et al. (2005) by using a Laplacian operator to work directly

on the compressed images. Using DC and AC coefficients of the DCT domain (Miano 1999), the colour, texture, shape and statistical features are identified from skin regions (Ye et al. 2003). Further, the classifier detects the objectionable image. Finding the areas of interest within the image, and detecting the coarse edges is found in Shen and Sethi (1996). Pattern matching on Huffman encoded texts has been attempted by Klein and Shapira (2005). Losslessly compressed images obtained using lossless JPEG can be retrieved using information derived from the Huffman coding tables of the compressed image in Schaefer (2010).

Video compression techniques are used frequently for the purpose of compressing digital videos in different applications (Salomon et al. 2010). Because of the large availability of videos over the internet and other sources of multimedia in a compressed form, there have been several attempts to tackle the problems related to video processing in the compressed domain. Operations such as key-frame extraction (Avrithis et al. 1999; Shen and Delp 1995; Yeo and Liu 1995; Zhang et al. 1995), caption localization (Yeo and Liu 1995b; Gargi et al. 1998), object recognition (Jing and Zhang 2004; Wang and Chang 1997), and video transcoding (Moiron et al. 2009) have been attempted in the literature of video processing in the compressed domain.

MPEG (Moving Picture Experts Group) (Mukhopadhyay 2011) audio compression techniques are in popular use so as to overcome the storage and transmissions issues of high quality audio signals. Therefore, several attempts have been reported to directly process the MPEG data in the compressed domain. A detailed study on MPEG audio compressed data, and subsequent analysis and processing strategies are reported in the research work of Anantharaman (2001). The other significant contributions on MPEG audio compressed domain processing are by Tzanetakis and Cook (2000), Schuller et al. (2011), Rizzi et al. (2006), Shao et al. (2004) and Li and Han (2009).

In the literature, there are also attempts reported to do pattern matching in compressed text document files (Adjeroh et al. 2013). Searching for patterns is a crucial activity in many applications, which both humans and machines do most of the time. The fundamental concept of pattern matching in a compressed text is also meant to overcome the problem of storage, and computation at run time (Adjeroh et al. 2013). In conventional pattern matching, text and pattern are both given in an uncompressed form and the problem is to check the occurrence of the pattern in the given text which may be an exact/inexact matching. In case of a compressed pattern matching problem, the text, the pattern, or both the text and the pattern are found in a compressed form (Amir and Benson 1992; Gasieniec and Rytter 1999).

The first attempt was made by Amir et al. (1996) to search for patterns in LZW (Limpel-Ziv-Welch) compressed files. LZW is a dictionary based compression method where data compression is achieved by replacing strings in the source text with a unique code generated for the strings in the dictionary. In practicality, the LZW compression is carried out using a tree-like data structure called as dictionary trie. The tree structure maps the unique code words generated for the strings. Using this compressed trie dictionary, the compressed patterns are matched and the occurrences reported. The detailed discussion is available in Amir et al. (1996) and Adjeroh et al. (2013), and the extended works in Tao and Mukherjee (2005) and Gawrychowski (2012). On LZ77 compressed text, (Farach and Thorup 1995) were the first to devise an algorithm for conducting a search by introducing informers which indicated the regions that were likely to contain the pattern. This scheme assists in decompressing and analyzing only the regions identified by the informers. Later a hybrid compression scheme using LZ77 and LZ88 was introduced by Navarro and Raffinot (2004) which allowed for a more efficient pattern matching than that of Farach and Thorup (1995). The latest developments on the LZ family of compression can be traced through the work of Gawrychowski (2011, 2012). Finally, a Grammar based compression which is closely related to the dictio-

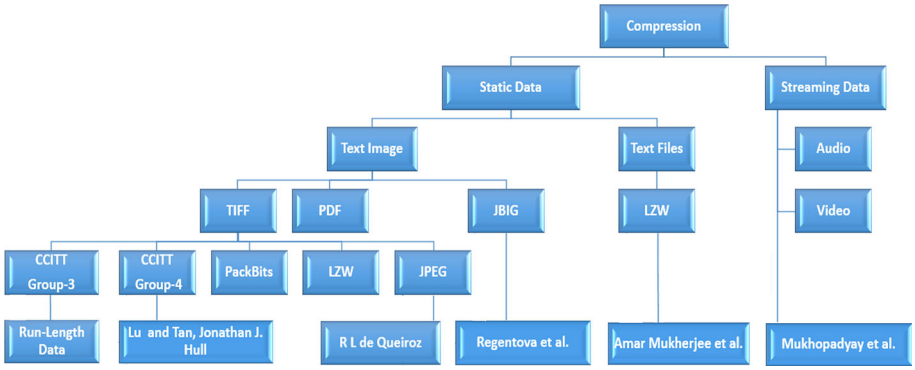

Fig. 6 Overview of compressed domain techniques in the literature

nary based method was investigated for compressed pattern matching by Kieffer and Yang (2000) and Yang et al. (2001). In BWT (Borrow Wheeler Transform), the text is broken into blocks and then compressed. The pattern searching requires the entire block to be traced; therefore, Bell et al. (2002) proposes to maintain an inverted index at the locations where the patterns may most likely occur, and then carry out the compressed pattern match. A detailed comparative study is provided by Adjeroh et al. (2008).

Run-length encoding is the compression algorithm that replaces a sequence of symbols with the count of the occurrences of that particular symbol. The initial idea of pattern searching on run-length encoded text were given by Apostolico et al. (1997) and Bunke and Csirik (1993, 1995). In Adjeroh et al. (1999) video sequences were represented as strings, which resulted in similar strings. During the matching of video sequences, a special edit operation had been defined to detect the repeated strings based on the principle of run-length encoding. Pattern matching on the other variations of run-length coded text were studied by Makinen et al. (2003) and Tamakoshi et al. (2013), and on 2D run-length by Amir and Benson (1992) and Amir et al. (2006).

Direct searching techniques were also introduced in Huffman compressed files by Mukherjee and Acharya (1994). In order to search patterns, they introduced a new data structure that separates byte boundaries and allows pattern matching with reference to the compressed pattern. The other strategies to search patterns in the compressed text are discussed in Moura et al. (2000), Ziviani et al. (2000) and Klein and Shapira (2005, 2011). Compressed pattern matching is also reported using the compression techniques of Arithmetic coding (Bell et al. 2002), Anti-dictionary (Shibata et al. 1999), PPM (Prediction by Partial Matching) (Cleary and Teahan 1997), Byte pair encoding (Mandal et al. 1999; Manber 1997; Shibata et al. 1999).

An overview of the compressed domain techniques in the literature is given in Fig. 6.

5 Future research direction of DIA in the compressed domain

In the previous state-of-the-art section, we saw that only few DIA related operations have been tried in the compressed domain and most them have been proven to be computationally efficient (Mukhopadhyay 2011; Javed et al. 2013; Adjeroh et al. 2013). Hence there is a large scope for research in the field of document images analysis to be carried out directly in

the compressed domain. The important applications like graphics recognition (Ogier et al. 2009; Doermann and Tombre 2014), handwriting recognition (Wshah et al. 2012a, b; Rehman and Saba 2012; Ahmed et al. 2017; Dash et al. 2016; Jayadevan et al. 2012), image analysis (Kasturi et al. 2002; Marinai 2008a), texture based image representation (Dong et al. 2015a, b; Alvarez et al. 2012; Li et al. 2010), clustering analysis of document images (Li et al. 2016; Zeng et al. 2014; Xu et al. 2003; Shahnaz et al. 2006; Gambhir and Gupta 2016; Asghari and KeyvanPour 2015), document understanding (Ceci et al. 2005), camera based document processing (Iwamura and Shafait 2013), document image mining (Marinai 2008b), document forensics (Saini and Kaur 2016), historical documents (Rath and Manmatha 2003, 2007), segmentation and restoration (Yong et al. 2010) and other innovative applications can be tried. Further, due to digitization, there is also scope for automatic processing of digitally born documents (Doermann and Tombre 2014; Lee and On 2011; Asghari and KeyvanPour 2015) in the compressed domain.

6 Conclusion

Through this survey paper, we presented a study on document image analysis techniques from the perspectives of image processing, image compression and compressed domain processing. The motivation behind pursuing research directly on compressed document images were discussed. Different document image processing techniques that have been attempted in the literature were reviewed on the basis of data compression technique employed during compression and the type of operations performed. Overall, the paper projects the importance of compressed domain processing of document images and gives a perspective for further research in the area of image processing and pattern recognition by employing directly the compressed data.

References

Adjeroh D, Bell T, Mukherjee A (2008) The burrows-wheeler transform: data compression, suffix arrays and pattern matching. Springer, New York

Adjeroh D, Bell T, Mukherjee A (2013) Pattern matching in compressed texts and images. Now Publishers, Hanover

Adjeroh DA, Lee MC, King I (1999) A distance measure for video sequence similarity matching. Comput Vis Image Underst 75(1):25–45

Ahmed N, Natarajan T, Rao K (1974) Discrete cosine transform. IEEE Trans Comput 23:90–93

Ahmed R, Al-Khatib WG, Mahmoud S (2017) A survey on handwritten documents word spotting. Int J Multimed Inf Retr 6(1):31–47

Aho AV, Corasick M (1975) Efficient string matching: an aid to bibliographic search. Commun ACM 18(6):333–340

Akutsu T (1994) Approximate string matching with dont care characters. In: Proceedings combinatorial pattern matching, LNCS, vol 807, pp 240–249

Alvarez S, Salvatella A, Vanrell M, Otazu X (2012) Low-dimensional and comprehensive color texture description. Comput Vis Image Underst 116(1):54–67

Amir A, Benson G (1992) Efficient two-dimensional compressed matching. In: IEEE proceedings of data compression conference, pp 279–288

Amir A, Calinescu G (1996) Alphabet independent and dictionary scaled matching. In: Proceedings of combinatorial pattern matching (LNCS 1075), pp 320–334

Amir A, Landau G, Vishkin U (1992) Efficient pattern matching with scaling. J Algorithms 13:2–32

Amir A, Bensonb G, Farach M (1996) Let sleeping files lie: pattern matching in z-compressed files. J Comput Syst Sci 52(2):299–307

Amir A, Kapah O, Tsur D (2006) Faster two-dimensional pattern matching with rotations. Theor Comput Sci 368(3):196–204
Anantharaman B (2001) Compressed domain processing of MPEG audio. PhD thesis, Indian Institute of Science, Bangalore
Andrews H (1970) Computer techniques in image processing. Academic Press, New York
Angadi SA (2007) An intelligent integrated automation system for efficient processing of postal mail. PhD thesis, Department of Studies in Computer Science, University of Mysore
Antonacopoulos A, Bridson D, Papadopoulos C, Pletschacher S (2009) A realistic dataset for performance evaluation of document layout analysis. In: Proceedings of the 10th international conference on document analysis and recognition, (ICDAR2009). Barcelona, pp 296–300
Apostolico A, Landau GM, Skiena S (1997) Matching for run-length encoded strings. In: Proceedings of complexity and compression of sequences
Ascher R, Nagy G (1974) A means for achieving a high degree of compaction on scan-digitized printed text. IEEE Trans Comput 23:1174–1179
Asghari E, KeyvanPour M (2015) Xml document clustering: techniques and challenges. Artif Intell Rev 43(3):417–436
Avcibas I, Kharrazi M, Memon ND, Sankur B (2005) Image steganalysis with binary similarity measures. EURASIP J Appl Signal Process 17:2749–2757
Avrithis YS, Doulamis AD, Doulamis ND, Kollias SD (1999) Astochastic framework for optimal key frame extraction from mpeg video databases. Comput Vis Image Underst 75(1/2):3–24
Baird H (1987) Skew angle of printed documents. In: Proceedings of SPSE's 40th annual conference and symposium on hhybrid imaging systems, pp 21–24
Baird HS, Bunke H, Yamamoto K (eds) (1992) Structured document image analysis. Springer, New York
Baird HS, Nagy G (1994) Self-correcting 100-font classifier. Doc Recognit 2181:106–115
Baird HS, Tombre K (2014) The evolution of document image analysis. In: Doermann D, Tombre K (eds) Handbook of document image processing and recognition, pp 63–71
Bell T, Powell M, Mukherjee A, Adjeroh DA (2002) Searching bwt compressed text with the boyer-moore algorithm and binary search. In: IEEE proceedings of data compression conference, pp 112–121
Berry M W (2013) Survey of text mining: clustering, classification, and retrieval. Springer, New York
Bhaskaran V, Konstantinides K, Beretta G (1997) Text and image sharpening of scanned images in the jpeg domain. In: Proceedings of international conference on image processing, vol 2, pp 326–329
Bolan S (2012) Document image enhancement. PhD thesis, National University of Singapore
Breuel TM (2003) High performance document layout analysis. In: Proceedings of symposium on document image understanding technology
Breuel TM (2008) Binary morphology and related operations on run-length representations. In: International conference on computer vision theory and applications - VISAPP, pp 159–166
Bunke H, Csirik J (1993) An algorithm for matching run-length coded strings. Computing 50:297–314
Bunke H, Csirik J (1995) An improved algorithm for computing the edit distance of run-length coded strings. Inf Process Lett 54:93–96
Ceci M, Berardi M, Malerba D (2005) Relational learning techniques for document image understanding: comparing statistical and logical approaches. In: Proceedings of the eighth international conference on document analysis and recognition, pp 473–477
Chang S (1995a) Compressed domain techniques of image/ video indexing and manipulation. In: IEEE international conference on image processing (ICIP95), special session on digital library and video on demand
Chang S (1995b) Some new algorithms for processing images in the transform compressed domain. In: SPIE symposium on visual communications and image processing
Chang S, Messerschmitt D (1995) Manipulation and compositing mc-dct compressed video. IEEE J Sel Areas Commun 13(1):1–11
Chang S, Chen W, Messerschmitt D (1992) Video compositing in the dct domain. In: IEEE workshop on visual signal processing and communications
Chen B, Wornell GW (2001) Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Trans Inf Theory 47(4):1423–1443
Chen N, Blostein D (2007) A survey of document image classification: problem statement, classifier architecture and performance evaluation. IJDAR 10(1):1–16
Chen B, Latifi S, Kanai J (1999) Edge enhancement of remote sensing image data in the dct domain. Image Vis Comput Elsevier 17:913–921
Chen K, Yin F, Liu C-L (2013) Page segmentation with efficient whitespace rectangles extraction and grouping. In: 12th international conference on document analysis and recognition, pp 958–962
Chiptrasert B, Rao K (1990) Discrete cosine transform filtering. Signal Process 19(3):233–245

Chua TS, Zhao Y, Kankanhalli MS (2002) Detection of human faces in a compressed domain for video stratification. Vis Comput 18:121–133
Chung K-L, Huang H-L, Lu H-I (2004) Efficient region segmentation on compressed gray images using quadtree and shading representation. Pattern Recognit 37:1591–1605
Cleary JG, Teahan WJ (1997) Unbounded length contexts for ppm. Comput J 40(2/3):67–75
Crochemore M, Hancart C, Lecroq T (2007) Algorithms on strings. Cambridge University Press, Cambridge
Cvision Technologies (2015). Reduce tiff file size (http://www.cvisiontech.com/file-formats/tiff/reduce-tiff-file-size.html)
Dash KS, Puhan NB, Panda G (2016) Odia character recognition: a directional review. Artif Intell Rev, pp 1–25
de Queiroz RL (1998) Processing jpeg-compressed images and documents. IEEE Trans Image Process 7(12):1661–1672
de Queiroz RL, Eschbach R (1997) Segmentation of compressed documents. In: Proceedings of international conference on image processing, vol 3, pp 70–73
de Queiroz RL, Eschbach R (1998) Fast segmentation of the jpeg compressed documents. J Electron Imaging 7(2):367–377
Deng S, Latifi S, Kanai J (1998) Manipulation of text documents in the modified group 4 domain. In: Multimedia signal processing, IEEE second workshop, pp 438–443
Deng S, Latifi S, Kanai J (1999) Document image analysis using a new compression algorithm. In: Document analysis systems: theory and practice (Lecture notes in computer science), vol 1655, pp 32–41
Dhandra BV, Nagabhushan P, Hangarge M, Hegadi R, Malemath VS (2006) Script identification based on morphological reconstruction in document images. In: Proceedings of the 18th international conference on pattern recognition, vol 2, pp 950–953
Ding S, Zhu H, Jia W, Su C (2012) A survey on feature extraction for pattern recognition. Artif Intell Rev 37:169–180
Doermann D (1998) The indexing and retrieval of document images: a survey. Comput Vis Image Underst 70(3):287–298
Doermann D, Li H, Kia O (1998) The detection of duplicates in document image database. Image Vis Comput 16:907–920
Doermann D, Tombre K (eds) (2014) Handbook of document image processing and recognition. Springer, London
Dong Y, Tao D, Li X (2015a) Nonnegative multiresolution representation-based texture image classification. ACM Trans Intell Syst Technol 7(1):4:1–4:21
Dong Y, Tao D, Li X, Ma J, Pu J (2015b) Texture classification and retrieval using shearlets and linear regression. IEEE Trans Cybern 45(3):358–369
Dugad R, Ahuja N (2001) A fast scheme for image size change in the compressed domain. IEEE Trans Circuits Syst Video Technol 11(4):461–474
Eilam-Tzoreff T, Vishkin U (1988) Matching patterns in strings subject to multi-linear transformations. Theor Comput Sci 60:231–254
Farach M, Thorup M (1995) String matching in lempel-ziv compressed strings. In: Proceedings of annual ACM symposium on the theory of computing, pp 703–712
Farahmand A, Sarrafzadeh A, Shanbehzadeh J (2013) Document image noises and removal methods. In: Proceedings of the international multiconference of engineers and computer scientists, vol 1, pp 436–440
Faro S, Lecroq T (2013) The exact online string matching problem: a review of the most recent results. ACM Comput Surv 45(2):13:1–13:42
Fredriksson K, Mozgovoy M (2006) Efficient parameterized string matching. Inf Process Lett 100(3):91–96
Gambhir M, Gupta V (2016) Recent automatic text summarization techniques: a survey. Artif Intell Rev 47:1–66
Garain U, Chakraborty MP, Chanda B (2006a) Lossless compression of textual images: a study on indic script documents. In: ICPR, vol 3, pp 806–809
Garain U, Datta AK, Bhattacharya U, Parui SK (2006b) Summarization of jbig2 compressed indian language textual images. In: ICPR, vol 3, pp 344–347
Gargi U, Antani S, Kasturi R (1998) Indexing text events in digital video databases. In: IEEE proceedings of ICPR, pp 916–918
Gasieniec L, Rytter W (1999) Almost optimal fully lzw-compressed pattern matching. In: IEEE proceedings of data compression conference, pp 316–325
Gawrychowski P (2011) Optimal pattern matching in lzw compressed strings. In: Proceedings of symposium on discrete algorithms, pp 362–372

Gawrychowski P (2012) Tying up the loose ends in fully lzw-compressed pattern matching. In: Proceedings of symposium on theoretical aspects of computer sciences, vol 14, pp 624–635
Ghosh D, Dube T, Shivaprasad A (2010) Script recognition-a review. IEEE Trans Pattern Anal Mach Intell 32(12):2142–2161
Giancarlo R, Gross R (1997) Multi-dimensional pattern matching with dimensional wildcards: data structures and optimal on-line search algorithm. J Algorithms 24:223–265
Gonzalez RC, Woods RE (2009) Digital Image Processing, 3rd edn. Pearson, New Delhi
Habibi A (1977) Survey of adaptive image coding techniques. IEEE Trans Commun 25:1275–1284
Hendahewa A (2010) 8 Image enhancement techniques in document capture. EIM BLOG (http://www.docudude.com/2010/04/8-image-enhancement-techniques-in.html)
Hernndez JR, Amado M, Gonzlez FP (2000) Dct-domain watermarking techniques for still images: detector performance analysis and a new structure. IEEE Trans Image Process 9(1):55–68
Hinds S, Fisher J, D'Amato D (1990) A document skew detection method using run-length encoding and the hough transform. In: Proceedings of 10th international conference on pattern recognition, vol 1, pp 464–468
Hull JJ (1997) Document matching on ccitt group 4 compressed images. In: SPIE conference on document recognition IV, pp 8–14
Hull JJ, Cullen J (1997) Document image similarity and equivalence detection. In: IEEE proceedings of ICDAR, vol 1, pp 308–312
Hull JJ (1998) Document image similarity and equivalence detection. Int J Doc Anal Recognit 1:37–42
Inglis S, Witten I (1994) Compression based template matching. In: IEEE proceedings of data compression conference, pp 106–115
Ito I, Kiya H (2007) Dct sign-only correlation with application to image matching and the relationship with phase-only correlation. In: IEEE proceedings of international conference on speech, acoustic and signal processing, pp 1237–1240
Iwamura M, Shafait F (2013) Camera-based document analysis and recognition. In: 5th international workshop on camera-based document analysis and recognition
Jain A (1989) Fundamentals of digital image processing. Prentice Hall, New Jersey
Jathanna VE, Nagabhushan P (2015) Microcontroller based mechanised videographing of text and auto-generation of voice text in real time. IJCSIT 6(3):2419–2425
Javed M, Nagabhushan P, Chaudhuri BB (2013) Extraction of projection profile, run-histogram and entropy features straight from run-length compressed text-documents. In: Second IAPR Asian conference on pattern recognition (ACPR2013), pp 813–817
Javed M, Nagabhushan P, Chaudhuri BB (2015a) Automatic extraction of correlation-entropy features for text document analysis directly in run-length compressed domain. In: 13th international conference on document analysis and recognition (ICDAR), pp 1–5
Javed M, Nagabhushan P, Chaudhuri BB (2015b) A direct approach for word and character segmentation in run-length compressed documents with an application to word spotting. In: 13th international conference on document analysis and recognition (ICDAR), pp 216–220
Javed M (2016) On the possibility of processing document images in compressed domain. PhD thesis, Department of Studies in Computer Science, University of Mysore
Javed M, Krishnanand SH, Nagabhushan P, Chaudhuri BB (2016a) Visualizing ccitt group 3 and group 4 tiff documents and transforming to run-length compressed format enabling direct processing in compressed domain. Procedia Comput Sci 85:213–221
Javed M, Nagabhushan P, Chaudhuri BB (2016b) Spotting of keyword directly in run-length compressed documents. In: Proceedings of Computer Vision and Image Processing (CVIP), vol 459. Springer, pp 367–376
Jawahar CV, Meshesha M, Balasubramanian A (2004a) Searching in document images. In: Proceedings of the international conference on visualization, graphics and image processing, pp 622–627
Jawahar CV, Million M, Balasubramanian A (2004b) Word level access to document image datasets. In: Proceedings of the workshop on computer vision, graphics and image processing, pp 73–76
Jayadevan R, Kolhe SR, Patil PM, Pal U (2012) Automatic processing of handwritten bank cheque images: a survey. Int J Doc Anal Recognit 15(4):267–296
Jing XY, Zhang D (2004) A face and palmprint recognition approach based on discriminant dct feature extraction. IEEE Trans Syst Man Cybern 34(6):2405–2415
Kanai J, Bagdanov AD (1998) Projection profile based skew estimation algorithm for jbig compressed images. Int J Doc Anal Recognit 1:43–51
Kasturi R, Gorman LO, Govindaraju V (2002) Document image analysis: a primer. Sadhana Part 1(27):3–22
Kia O (1997) Document compression and analysis. PhD thesis, Institute for Advanced Computer Studies, University of Maryland

Kieffer JC, Yang EH (2000) Grammar-based codes: a new class of universal lossless source codes. IEEE Trans Inf Theory 46(3):737–754
Klein B, Agne S, Dengel A (2004) Results of a study on invoice-reading systems in germany. Lecture notes in computer science, vol 3163, pp 451–462
Klein ST, Shapira D (2005) Pattern matching in huffman encoded texts. Inf Process Manag Elsevier 41:829–841
Klein ST, Shapira D (2011) Compressed matching in dictionaries. Algorithms 4(1):61–74
Knight JR, Myers, EW (1999) Super-pattern matching. Technical Report TR-92-29, Department of Computer Science, University of Arizona
Kou W (1995) Digital Image compression: algorithms and standards. Kluwer Academic Publishers, Amsterdam
Kresch R, Merhav N (1999) Fast dct domain filtering using the dct and the dst. IEEE Trans Image Process 8:821–833
Latifi S, Kanai J (1997) Rapid manipulation of images compressed by the ccitt group iii 1-d coding scheme. In: Proceedings of international conference on imaging sciences, systems, and technology (CISST'97), pp 351–354
Lee DS, Hull JJ (2001) Detecting duplicates among symbolically compressed images in a large document database. Pattern Recognit Lett 22:545–550
Lee I, On B-W (2011) An effective web document clustering algorithm based on bisection and merge. Artif Intell Rev 36(1):69–85
Lee J, Lee B (1992) Transform domain filtering based on pipelining structure. IEEE Trans Signal Process 40(8):2061–2064
Lee JS, Kim DK, Park K, Cho Y (1997) Efficient algorithms for approximate string matching with swaps. In: Proceedings of combinatorial pattern matching (LNCS), vol 1264, pp 28–39
Lee MS, Shen M, Yoneyama A, Kuo CCJ (2005) Dct-domain image registration techniques for compressed video. In: IEEE proceedings of international symposium on circuit systems, vol 5, pp 4562–4565
Lee S (2007) An efficient content-based image enhancement in the compressed domain using retinex theory. IEEE Trans Circuits Syst Video Technol 17(2):199–213
Li L, Tong CS, Choy SK (2010) Texture classification using refined histogram. IEEE Trans Image Process 19(5):1371–1378
Li M, Han J (2009) Streaming audio retrieval based on fuzzy classification in mpeg-1 compressed domain. In: International conference on mechatronics and automation, pp 5035–5039
Li X, Cui G, Dong Y (2016) Graph regularized non-negative low-rank matrix factorization for image clustering. IEEE Trans Cybern PP(99):1–14
Lim J (1990) Two dimensional signal and image processing. Prentice Hall, New Jersey
Lloret E, Palomar M (2012) Text summarisation in progress: a literature review. Artif Intell Rev 37(1):1–41
Lu CS (2002) Block dct-based robust watermarking using side information extracted by mean filtering. In: IEEE proceedings of ICPR, vol 2, pp 1001–1004
Lu J, Jiang D (2011) Survey on the technology of image processing based on dct compressed domain. In: ICMT, pp 786–789
Lu S, Su B, Tan CL (2010) Document image binarization using background estimation and stroke edges. IJDAR 13(4):303–314
Lu Y, Tan CL (2003a) Document retrieval from compressed images. Pattern Recognit 36:987–996
Lu Y, Tan CL (2003b) Word searching in ccitt group 4 compressed document images. In: IEEE proceedings of ICDAR, pp 467–471
Lu Y, Tan CL, Huang W, Fan L (2001) An approch to word image matching based on weighted hausdorff distance. In: Proceedings of ICDAR, pp 921–925
Maa CY (1994) Identifying the existence of bar codes in compressed images. CVGIP. Graph Models Image Process 56(4):352–356
Makinen V, Ukkonen E, Navarro G (2003) Approximate matching of run length compressed strings. Algorithmica 35:347–369
Manber U (1997) A text compression scheme that allows fast searching directly in the compressed file. ACM Trans Inf Syst 15(2):124–136
Mandal MK, Idris F, Panchanathan S (1999) A critical evaluation of image and video indexing techniques in the compressed domain. J Image Vis Comput 17:513–529
Marinai S, Gori M, Soda G (2005) Artificial neural network s for document analysis and recognition. IEEE Trans PAMI 27(1):23–35
Marinai S (2008a) Introduction to document analysis and recognition. Stud Comput Intell 90:1–20
Marinai S (2008b) Machine learning in document analysis and recognition. Springer, Heidelberg
Marti UV, Wymann D, Bunke H (2000) Ocr on compressed images using pass modes and hidden markov models. In: Proceedings of IAPR workshop on document analysis systems, pp 77–86

Martucci SA (1995) Image resizing in the discrete cosine transform domain. In: IEEE proceedings of internation conference on image processing, vol 2, pp 224–227
Mazzarri A, Leonardi R (1995) Perceptual embedded image coding using wavelet tranforms. ICIP, pp 586–587
Merhav N, Bhaskaran V (1997) Fast algorithms for dct-domain image down-sampling and for inverse motion compensation. IEEE Trans Circuits Syst Video Technol 7(6):468–476
Meunier JL (2005) Optimized xy-cut for determining a page reading order. In: International conference on document analysis and recognition, vol 1, pp 347–351
Miano J (1999) Compressed image file formats: JPEG, PNG, GIF, XBM, BMP. ACM Press, New York
Moiron S, Faria S, Navarro A, Silva V, Assunc P (2009) Video transcoding from h.264/avc to mpeg-2 with reduced computational complexity. Signal Process Image Commun 24:637–650
Moura ES, Navarro G, Baeza-Yates R (2000) Fast and flexible word searching on compressed text. ACM Trans Inf Syst 18(2):113–139
Mukherjee A, Acharya T (1994) Compressed pattern-matching. In: IEEE proceedings of data compression conference, p 468
Mukherjee J, Mitra SK (2006) Image filtering in the compressed domain. In: Proceedings of the 5th Indian conference on computer vision, graphics and image processing (ICVGIP'06), LNCS, vol 4338, pp 194–205
Mukherjee J, Mitra SK (2008) Enhancement of color images by scaling the dct coefficients. IEEE Trans Image Process 17(10):1783–1794
Mukhopadhyay J, Mitra SK (2009) Color constancy in the compressed domain. In: IEEE proceedings of internation conference on image processing, pp 705–708
Mukhopadhyay J (2011) Image and video processing in compressed domain. Chapman and Hall/CRC, Boca Raton
Murugappan A, Ramachandran B, Dhavachelvan P (2011) A survey of keyword spotting techniques for printed document images. Artif Intell Rev 35(2):119–136
Na S, Jinxiao P (2011) Fast and robust skew detection for scanned documents. In: International conference on electronic and mechanical engineering and information technology (EMEIT), vol 8, pp 4170–4173
Nagy G (2000) Twenty years of document image analysis in pami. IEEE Trans PAMI 22(1):38–62
Nagy G, Seth S, Viswanathan M (1992) A prototype document image analysis system for technical journals. Computer 25(7):10–22
Namboodiri AM, Jain AK (2007) Document structure and layout anal. Digit Doc Process, pp 29–48
Navarro G, Raffinot M (1999) A general practical approach to pattern matching over ziv-lempel compressed text. In: Proceedings of combinatorial pattern matching (LNCS 1645), pp 14–36
Navarro G (2001) A guided tour to approximate string matching. ACM Comput Surv 33(1):31–88
Navarro G, Raffinot M (2004) Practical and flexible pattern matching over ziv-lempel compressed text. J Discrete Algorithms 2(3):347–371
Ngan K, Clarke R (1980) Lowpass filtering in the cosine transform domain. In: International conference on communication
Nixon MS, Aguado AS (2012) Feature extraction and image processing. Elsevier, Oxford
Ogier JM, Liu W, Llados J (2009) Graphics recognition: achievements, challenges and evolution. In: ICDAR 2009
Pirsch S (1982) Adaptive intra/interframe dpcm coder. Bell Syst Tech J 61:747–764
Provos N (2001) Defending against statistical steganalysis. In: Proceedings of 10th USENIX security symposium, vol 10, pp 323–335
Ramanathan R, Soman KP, Thaneshwaran L, Viknesh V, Arunkumar T, Yuvaraj P (2009) A novel technique for english font recognition using support vector machines. In: International conference on advances in recent technologies in communication and computing, pp 766–769
Rath T, Manmatha R (2003) Features for word spotting in historical manuscripts. In: International conference on document analysis and recognition, pp 218–222
Rath TM, Manmatha R (2007) Word spotting for historical documents. IJDAR 9(2–4):139–152
Reeves R, Kubik K, Osberger W (1997) Texture characterization of compressed aerial images using dct coefficients. In: Proceedings of SPIE: storage and retrieval for image and video databases, vol 3022, pp 398–407
Regentova E, Latifi S, Deng S, Yao D (2002) An algorithm with reduced operations for connected components detection in itu-t group 3/4 coded images. IEEE Trans Pattern Anal Mach Intell 24(8):1039–1047
Regentova E, Latifi S, Chen D, Taghva K, Yao D (2005) Document analysis by processing jbig-encoded images. IJDAR 7:260–272
Rehman A, Saba T (2012) Off-line cursive script recognition: current advances, comparisons and remaining problems. Artif Intell Rev 37:261–288

Rehman A, Saba T (2014) Neural networks for document image preprocessing: state of the art. Artif Intell Rev 42(2):253–273
Rizzi A, Buccino M, Panella M, Uncini A (2006) Optimal short-time features for music/speech classification of compressed audio data. In: International conference on computational intelligence for modelling, control and automation, p 210
Ronse C, Devijver P (1984) Connected components in binary images: the detection problem. Research Studies Press, Letchworth
Rosenbaum R, Taubman D (2003) Merging images in jpeg domain. In: ICIP, vol 1, pp 249–252
Saini K, Kaur S (2016) Forensic examination of computer-manipulated documents using image processing techniques. Egypt J Forensic Sci 6(3):317–322
Salomon D, Motta G, Bryant D (2010) Handbook of data compression. Springer, London
Salton G (1988) Automatic text processing. Addison-Wesley Longman Publishing Co, Boston
Saragiotis P, Papamarkos N (2008) Local skew correction in documents. IJPRAI 22(4):691–710
Sayood K (2012) Introduction to data compression, 4th edn. Morgan Kaufmann, Burlington
Schaefer G (2010) Content-based retrieval of compressed images. In: International workshop on databases, texts, specifications and objects (DATESO2010), pp 175–185
Schuller G, Gruhne M, Friedrich T (2011) Fast audio feature extraction from compressed audio data. IEEE J Sel Top Signal Process 5:1262–1271
Scotney BW, Coleman S, Herron M (2005) Direct feature detection on compressed images. Pattern Recogn Lett 26:2336–2345
Shahnaz F, Berry MW, Pauca VP, Plemmons RJ (2006) Document clustering using nonnegative matrix factorization. Inf Process Manag 42(2):373–386
Shao X, Xu C, Wang Y, Kankanhall MS (2004) Automatic music summarization in compressed domain. In: IEEE proceedings of acoustics, speech, and signal processing, vol 4, pp 261–264
Shen B, Sethi I (1995) Inner-block operations on compressed images. In: Proceedings of ACM multimedia'95 San Francisco, pp 490–499
Shen B, Sethi I (1996) Direct feature extraction from compressed images. In: Proceedings of SPIE, storage & retrieval for image and video databases IV, vol 2670, pp 404–414
Shen K, Delp E (1995) A fast algorithm for video parsing using mpeg compressed sequences. In: IEEE proceedings of internation conference on image processing, vol 2, pp 252–255
Shibata Y, Takeda M, Shinohara A, Arikawa S (1999) Pattern matching in text compressed by using antidictionaries. In: Proceedings, combinatorial pattern matching, vol 1645, pp 37–49
Shima Y, Kashioka S, Higashino J (1989) A high-speed rotation method for binary images based on coordinate operation of run data. Syst Comput Jpn 20(6):91–102
Shima Y, Kashioka S, Higashino J (1990) A high-speed algorithm for propagation-type labeling based on block sorting of runs in binary images. In: Proceedings of 10th international conference on pattern recognition (ICPR), vol 1, pp 655–658
Shiraishi S, Feng Y, Uchida S (2013) Skew estimation by parts. IEICE Trans Inf Syst 96:1503–1512
Shneier M, Mottaleb MA (1996) Exploiting the jpeg compression scheme for image retrieval. IEEE Trans Pattern Anal Mach Intell 18(8):849–853
Slimane F, Kanoun S, Hennebert J, Alimi AM, Ingold R (2013) A study on font-family and font-size recognition applied to arabic word images at ultra-low resolution. Pattern Recognit Lett 34(2):209–218
Smith B, Rowe L (1993) Algorithms for manipulating compressed images. IEEE Comput Graph Appl 13:34–42
Smith JR, Chang SF (1994) Transform features for texture classification and discrimination in large image databases. In: IEEE proceedings of ICPR, pp 407–411
Spitz AL (1998) Analysis of compressed document images for dominant skew, multiple skew, and logotype detection. Comput Vis Image Underst 70(3):321–334
T.4-Recommedation (1985) Standardization of group 3 facsimile apparatus for document transmission, terminal equipments and protocols for telematic services, vol. vii, fascicle, vii. 3, Geneva. Technical report
T.6-Recommendation (1985) Standardization of group 4 facsimile apparatus for document transmission, terminal equipments and protocols for telematic services, vol. vii, fascicle, vii. 3, Geneva. Technical report
Tamakoshi Y, Tomohiro I, Inenaga S, Bannai H, Takeda M (2013) From run length encoding to lz78 and back again. In: IEEE proceedings of data compression conference, pp 143–152
Tang J, Peli E, Acton S (2003) Image enhancement using a contrast measure in the compressed domain. IEEE Signal Process Lett 10:289–292
Tang YY, Lee S-W, Suen CY (1996) Automatic document processing: a survey. Pattern Recognit 29(12):1931–1952
Tao T, Mukherjee A (2005) Pattern matching in lzw compressed file. IEEE Trans Comput 54(8):929–938
TIFF (1992) (tagged image file format) revision 6.0 specification. Technical report

Tzanetakis G, Cook P (2000) Sound analysis using mpeg compressed audio. In: IEEE proceedings of acoustics, speech, and signal processing, vol 2, pp 761–764
Vasudev T (2007) Automatic data extraction from pre-printed input data forms: some new approaches. PhD thesis, University of Mysore
Venter F, Stein A (2012) Images & videos: really big data. Anal Mag, pp 15–20
Vetterli M (1984) Multi-dimensional sub-band coding: some theory and algorithms. Signal Process 6(2):97–112
Viswanath K (2009) Image transcoding in transform domain. PhD thesis, Dept. of Computer Science and Engineering, Indian Institute of Technology, Kharagpur
Viswanath K, Mukherjee J, Biswas PK, Pal RN (2010) Wavelet to dct transcoding in transform domain. Signal Image Video Process Springer 4(2):129–144
Wang H, Chang SF (1997) A highly efficient system for automatic face region detection in mpeg video. IEEE Trans Circuits Syst Video Technol 7(4):615–628
Woods J, O'Niel S (1986) Subband coding of images. IEEE Trans Acoust Speech Signal Process 34:1278–1288
Wshah S, Kumar G, Govindaraju V (2012a). Multilingual word spotting in offline handwritten documents. In: ICPR, pp 310–313
Wshah S, Kumar G, Govindaraju V (2012b) Script independent word spotting in offline handwritten documents based on hidden markov models. In: ICFHR, pp 14–19
Xu W, Liu X, Gong Y (2003) Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval, pp 267–273
Yang EH, Kaltchenko A, Kieffer JC (2001) Universal lossless data compression with side information by using a conditional mpm grammar transform. IEEE Trans Inf Theory 47(6):2130–2150
Ye Q, Gao W, Zeng W, Zhang T, Wang W, Liu Y (2003) Objectionable image recognition system in compression domain. In: 4th international conference on intelligent data engineering and automated learning (IDEAL 2003), LNCS, vol 2690, pp 1131–1135
Yeo BL, Liu B (1995a) Rapid scene analysis on compressed video. IEEE Trans Circuits Syst Video Technol 5(6):533–544
Yeo BL, Liu B (1995b) Visual content highlighting via automatic extraction of embedded captions on mpeg compressed video. In: Proceedings of SPIE digital video compression, algorithms and technologies, pp 142–149
Yim C (2004) An efficient method for dct-domain separable symmetric 2-d linear filtering. IEEE Trans Circuits Syst Video Technol 14(4):517–521
Yong X, Guangri Q, Yongdong X, Yushan S (2010) Keyword spotting in degraded document using mixed ocr and word shape coding. In: IEEE international conference on intelligent computing and intelligent systems, pp 411–414
Yucun P, Qunfei Z, kamata S (2010) Document layout analysis and reading order determination for a reading robot. In: IEEE proceedings of TENCON, pp 1607–1612
Zeng K, Yu J, Li C, You J, Jin T (2014) Image clustering by hyper-graph regularized non-negative matrix factorization. Neurocomputing 138:209–217
Zhang HJ, Low CY, Smolia SW (1995) Video parsing and browsing using compressed data. Multimed Tools Appl 1:89–111
Zirari F, Ennaji A, Nicolas S, Mammass D (2013) A document image segmentation system using analysis of connected components. In: 12th international conference on document analysis and recognition, pp 753–757
Ziviani N, Moura ES, Navarro G, Baeza-Yates R (2000) Compression: a key for next generation text retrieval systems. IEEE Comput 33(11):37–44