Multimed Tools Appl (2016) 75:1043–1078
DOI 10.1007/s11042-014-2345-z

A survey on compressed domain video analysis techniques

R. Venkatesh Babu · Manu Tom · Paras Wadekar

Received: 9 October 2013 / Revised: 11 September 2014 / Accepted: 30 October 2014 /
Published online:18 November 2014

Abstract Image and video analysis requires rich features that can characterize various aspects of visual information. These rich features are typically extracted from the pixel values of the images and videos, which require huge amount of computation and seldom useful for real-time analysis. On the contrary, the compressed domain analysis offers relevant information pertaining to the visual content in the form of transform coefficients, motion vectors, quantization steps, coded block patterns with minimal computational burden. The quantum of work done in compressed domain is relatively much less compared to pixel domain. This paper aims to survey various video analysis efforts published during the last decade across the spectrum of video compression standards. In this survey, we have included only the analysis part, excluding the processing aspect of compressed domain. This analysis spans through various computer vision applications such as moving object segmentation, human action recognition, indexing, retrieval, face detection, video classification and object tracking in compressed videos.

Keywords Video object segmentation · Human action recognition · Indexing · Retrieval · Face detection · Video classification · Object tracking · Object localization · Moving object detection · H.264/AVC · HEVC · MPEG · Compressed domain · Quantization parameter · Motion vectors · Transform coefficients · Video analysis

1 Introduction

Video analysis is one of the most important tasks in computer vision for applications such as video surveillance, indexing, retrieval and scene understanding. Most of the video analysis is performed in pixel domain due to the requirement of extracting robust and meaningful features from the visual data. On the contrary, the pixel domain approaches are computationally expensive due to the huge amount of data involved in processing. Large amount

R. Venkatesh Babu (✉) · M. Tom · P. Wadekar
Video Analytics Laboratory, SERC, Indian Institute of Science, Bangalore, India
e-mail: venky@serc.iisc.ernet.in

of video data generated everyday is stored in compressed form for archiving, distribution, and streaming purposes. The increase in popularity of cameras with resolutions ranging in higher definitions has necessitated faster video analysis tools that offer less computational overhead, for real-time applications. Conventional video analysis systems relying on decoding of numerous video streams and processing at pixel levels involves higher computational complexity, which is a bottleneck for real-time performance, even though they offer robust analysis. On the contrary, video analysis in compressed domain requires reduced computing power resulting in reduced bandwidth and storage requirements. Bit rate of uncompressed video is huge compared to its compressed counterpart which inculcates several compression methodologies thus reducing the memory requirement. The computational overhead required for complete decoding is also high for pixel level approaches. Compressed domain analysis, on the other hand, requires only partial decoding of the sparse cues such as motion vectors (MVs), transform coefficients, quantization parameters (QP), macroblock (MB) partition modes etc. The availability of hardware codecs even for the latest video compression standards such as H.264/AVC [89] and HEVC (High Efficiency Video Coding) [73] has made it possible to analyze their performances. Approaches involving compressed-domain processing are amenable to hardware implementation, which can further reduce the computational time.

Video compression is performed to reduce the spatio-temporal redundancy via image transforms and motion compensation. These transform coefficients and the motion vectors generated during this compression process contain useful information about the content of the video. The compressed domain information can be easily extracted from the bit-stream with partial decoding. The bit rate of these compressed domain parameters such as transform coefficients, motion vectors, quantization steps, coded block patterns etc. is very low compared to the pixel domain, leading to rapid analysis.

This paper provides a detailed survey of various state-of-the-art video-analysis research works performed on compressed videos. As in Fig. 1, a video analysis task based taxonomy is chosen, comparing the advantages and limitations of each approach. Various compressed domain approaches utilizing information from parameters such as motion vectors, DCT coefficients, MB partitions, luminance and chrominance values, color, MB sizes etc. are discussed in the paper. The analysis spans through various computer vision applications such as moving object segmentation, human action recognition, indexing, retrieval, face detection, video classification and object tracking in compressed videos. To our knowledge, this is the first detailed survey to review the published works, in compressed domain video analysis. We have covered the works done on open-source video formats such as MPEG-1/2, MPEG-4 Part 2, H.263, MPEG-4 Part 10 (H.264), excluding the proprietary video standards. The paper is organized as follows. Section 2 provides a brief description of the various video compression standards. A comparison between the most popular video standards viz., H.264/AVC and MPEG-2 is given in Section 2.3. The state-of-the-art research works in compressed video platforms available till date, are reviewed in detail in Section 3. Concluding remarks are made in Section 4.

2 Video compression standards

The International Telecommunications Union – Telecommunication Standardization sector (ITU-T) [35] is one of the organizations responsible for development of standards for use on global telecommunication networks. Another working group of experts known as the Motion Picture Experts Group (MPEG) [76] was formed by the International Organization

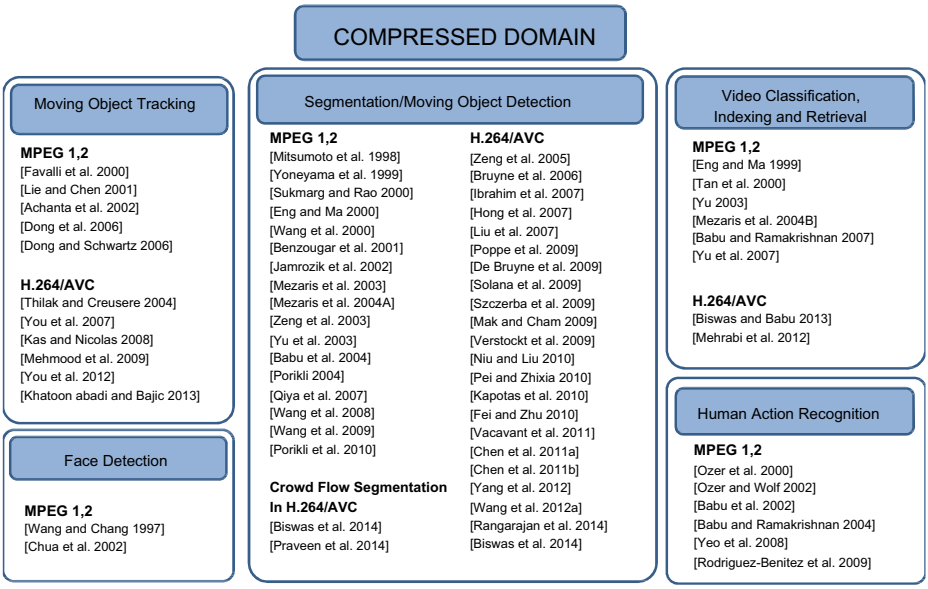


Fig. 1 Survey taxonomy

for Standardization (ISO) [34] and International Electrotechnical Commission (IEC) [88]. Video compression standards aim to minimize the spatio-temporal redundancies by exploiting the characteristics of human visual system along with source coding techniques from information theory.

Here the overview of the procedure is given with respect to H.264 standard [65]. Other standards follow a similar trend. There are mainly three types of pictures (frames) in compressed videos viz., Intra (I), Predicted (P) and Bi-predictive (B) frames. Details of the picture types are shown in Table 1. The arrangement of inter and intra frames in a video stream is specified by the Group of Pictures (GOP). Each coded video stream consists of sequence of GOPs. A GOP always begins with an intra frame, followed by P and B frames. A typical GOP structure *IBBPBBP* is shown in Fig. 2c. Only I and P frames are used as reference frames. The P frame is predicted from the previous I/P frame and B frames are predicted from previous and next I/P frames. The frames are divided into slices and each slice is subdivided into non-overlapping entities called macroblocks (MBs). The size of a MB is typical of the codec and is usually a multiple of 4 (typically 16×16).

Table 1 Picture types in compressed videos

Frame	Complexity	Compression	Motion Compensation	MB Type
I	Low	Low	No compensation	I
P	Moderate	Moderate	Forward	I , P
B	High	High	Forward & backward	I, P , B

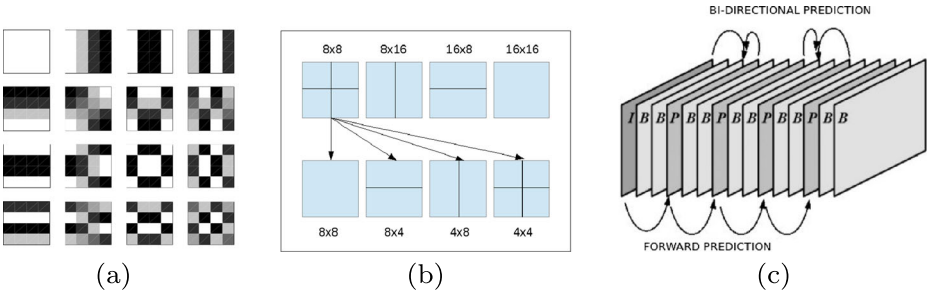


Fig. 2 **a** 4×4 DCT Basis Patterns (image courtesy : Iain Richardson [65]), **b** MB Partition Types (in H.264) and **c** Typical GOP structure

2.1 The compression algorithm

Compression of video data is based on reduction of spatial and temporal redundancies [8, 69] . Block diagram of a typical CODEC (encoder + decoder) is shown in Fig. 3 (The blocks colored in red are present from H.264/AVC standard onwards only). Initially, prediction of the current frame is performed by inter and intra prediction techniques. Residuals are then obtained by subtracting the predicted frame from the reference frame(s). The reconstructed residual is added with the predicted frame, obtained from the prediction module, to completely decode the current frame. In order to make the paper self-contained we briefly describe the main modules of video compression and various open compression standards. More details of these modules can be found in [65].

2.1.1 Motion compensation

Motion estimation is used for reducing temporal redundancy by identifying the match for a macroblock in the current frame with another in the reference frame(s). Motion vectors, as shown in Fig. 4b, indicate the location of matching MBs in the reference frame. Hence,

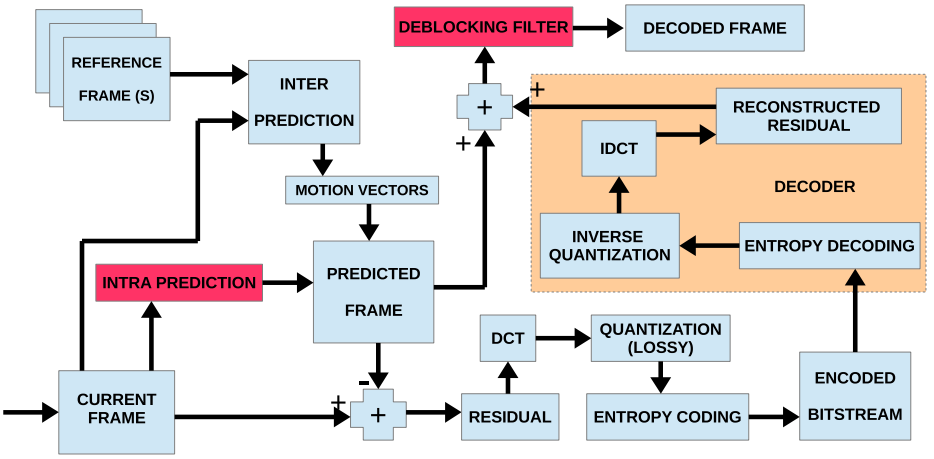
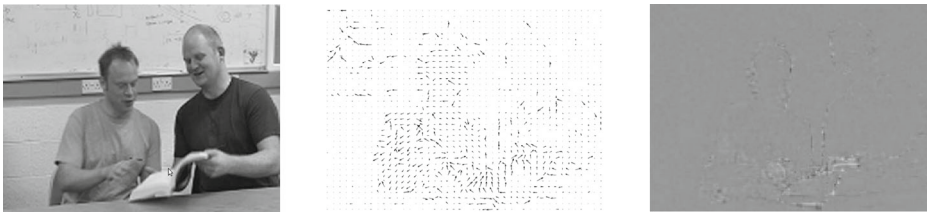


Fig. 3 A typical video CODEC



(a) Frame (b) Motion Vectors (c) Residual

Fig. 4 Motion compensation process (image courtesy: Iain Richardson [65])

motion vector of a block can be considered as a vector pointer, with magnitude as well as orientation. The MV indicates the best match of that particular MB of the current frame with the reference frame(s) that yields the minimum residual, within a stipulated search range. Encoding is done only for the residual or *error* signal, as shown in Fig. 4c, obtained by block matching and differencing, in order to minimize the amount of bits used. The reference frame(s) can be past or future frame(s) that are previously coded. Smaller the residual implies fewer are the bits to be entropy coded. Entropy coding (variable length coding, binary arithmetic coding) reduces statistical redundancy from the bitstream.

The objective of motion compensation is to attain maximum compression (minimum bits). Hence the motion vectors need not always account for true motion within the video. The MV field, hence may be noisy and requires pre-processing steps prior to analysis. The MVs and residuals contain information of complimentary nature. The P frames have one MV per block (16×16 in MPEG-1 and adaptive block from 16×16 through 4×4 in H.264/AVC) while B frames have two MVs, one forward and one backward.

2.1.2 Transform coding and quantization

Discrete Cosine Transform (DCT) is the most widely used block-based transform coding to convert the motion compensated residual data into frequency domain. Transform coding spatially decorrelates the data and retains most of the energy in few coefficients. The basis patterns for a 4×4 DCT is shown in Fig. 2a. These transformed coefficients are subjected to quantization which removes the less significant coefficients leading to lossy compression.

The extent of omission of data during quantization, which may bring about blockiness in the video for attaining more compression, is denoted by quantization parameter (QP). It has a major role in regulating the bit rate during the encoding process. QP values can be adjusted to maintain a constant bit rate within the allowed channel bandwidth. Hence, real-time encoders heavily depend on varying the QP values to control the trade-off between video quality and compression. Higher QP means greater step size resulting in higher compression at the cost of reduced quality and vice versa. After the quantization step, the DCT coefficients for a block are reordered to group nonzero coefficients. A zigzag scan order starting from the DC (top-left) coefficient is adopted. Each quantized coefficient is copied into a one-dimensional array. Nonzero coefficients tend to be grouped together at the start of the reordered array, followed by long sequences of zeros (high frequency DCT coefficients).

2.1.3 Entropy coding

Entropy coding is then performed to further reduce the data size. Each data symbol is replaced by an appropriate variable length code (VLC) or binary arithmetic code (BAC).

Context adaptiveness (CABAC [46], CAVLC [13]) is allowed in entropy coding from H.264/MPEG-4 AVC standard onwards. The whole entropy coding-decoding process is lossless.

2.2 Video standards

Examples of video compression formats are H.261, MPEG-1, MPEG-2, MPEG-4 Part 2, H.264 (MPEG-4 Part 10), Theora, Dirac, RealVideo RV40, VP8, HEVC etc. Here we have considered only the block-based standards which use DCT as the transform as these set of standards are most widely used open source video compression techniques. A brief overview of these video standards is given in the following sub-sections in chronological order.

2.2.1 H.261

This standard was one of the oldest digital video coding standards designed primarily for video bitrates in the range 40 kbit/s to 2 Mbit/s. H.261 [36] was developed for video over telephone, video conferencing, and other audio-visual services over telephone lines. The concept of macroblock was first adopted in H.261 standard. H.261 supports two video resolutions viz., QCIF (176×144) and CIF (352×288). H.261 standard operates on images represented in YUV color space (Y and Cb and Cr). YUV format represents images with 24 bits per pixel viz., 8 bits each for the luminance and two chrominance components. Subsampling is performed in which all the luma information is retained and chroma information is reduced by a factor 2 in both horizontal and vertical directions ($4 : 2 : 0$ subsampling). The whole subsampling process is lossy but does not affect the perceived quality since the human eye is more sensitive to luminance than to chrominance information. Only I and P frames are supported by H.261.

2.2.2 MPEG-1

This is a lossy compression standard extended from H.261 and Joint Photographic Experts Group (JPEG). Part 2 of the MPEG-1 [31] standard describes video and supports I, P and B frames. Resolutions are supported upto 4095×4095 (12 bits). MPEG-1 is designed typically for coding of moving pictures and associated audio for digital storage media upto about 1.5 Mbit/s, but can even go upto 100 Mbit/s. Common digital storage media include *compact discs (CDs)* and *video compact discs (VCDs)*. 1.2 Mbps out of the allocated 1.5 Mbps is intended for coded video, and 256 kbps can be utilized for stereo audio. Run Length Encoding (RLE) and Huffman coding are the two main types of entropy coding techniques adopted in MPEG-1 to compress the bit-stream. However, MPEG-1 supports only non-interlaced (progressive) video.

2.2.3 MPEG-2

Unlike MPEG-1, which is primarily for playing and storing videos on the CDs at 1.5 Mbps, MPEG-2 [32] was developed for higher quality video at bit rates more than 4 Mbps. This standard was typically developed for digital broadcast applications. Hence, NTSC (720×480) and PAL (720×576) video resolutions are easily supported with frame rates 29.97 fps and 25 fps respectively. MPEG-2 has gained wide acceptance beyond broadcasting digital TV over terrestrial, satellite, or cable networks. It is also adopted for *digital video/versatile discs (DVDs)*. The concept of video interlacing (Picture Adaptive Frame-Field Coding or

PicAFF) was first introduced in the part 2 of MPEG-2 compression standard. One of the major advantages of MPEG-2 is the backward compatibility with its predecessor MPEG-1. Chroma subsampling (4 : 2 : 2 and 4 : 2 : 0) methodologies are also supported. The standard also defines various profiles (simple profile, main profile, multi-view profile etc.) and levels (low level, high level, high 1440 etc.) for application specific usage. MPEG-2 also supports I, P, and B frames.

2.2.4 H.263 / H.263+

H.263 [66] was designed mainly for video conferencing and other audio-visual services transmitted on Public Switched Telephone Networks (PSTN). The standard is also a potential candidate for internet-based video applications like flash videos, aiming at low bit-rate applications less than 64 kbps. Unlike MPEG-2, unrestricted motion vector search range is permitted in H.263. The prediction module was improved to a large extent and forward error correction for the coded video signal was also introduced. An enhanced version known as H.263v2 or H.263+ was later proposed with advanced features like reference picture resampling, new intra coding and quantization modes etc.

2.2.5 MPEG-4 and AVC (H.264)

MPEG-4 Part 2 MPEG-4 Part 2 [33] adapts object based video coding concept where objects can be coded as elementary bit streams and composed into a scene by the author. All the previous standards code the video at frame level only, in which each rectangular frame is treated as a single unit for compression. Object based video coding not only offers higher compression, but is also flexible and useful for video manipulation, composition, indexing, and retrieval applications. MPEG-4 was originally designed for mobile applications in the range of 4.8 to 64 kbps and upto 2 Mbps for other applications like broadcasting. Also, the standard is backward compatible with H.263. While MPEG-1 doesn't support interlaced video at all and MPEG-2 requires entire streams to be either interlaced or progressive, AVC allows individual frames, or even MBs to be encoded as interlaced or progressive. It supports 21 different profiles like simple profile, advanced simple profile, simple studio profile etc. Global motion compensation (GMC) with quarter pel accuracy is also supported. However, MPEG-4 part 2 lacks in-loop deblocking filter and also does not offer much compression performance over MPEG-2 part 2.

Part 10 AVC (H.264) Compared to the predecessors, H.264/AVC [89] offers better video quality at lower bit-rates. More than 50 % bit rate savings can be achieved by replacing MPEG-2 videos with H.264 compression and hence can be employed for applications ranging from internet streaming to digital broadcasting. This standard also offers over 40 % compression performance than H.263+ and MPEG-4 advanced simple profile, and hence is currently a viable option to carry HDTV video content for many potential applications. Various advanced features adopted in H.264/AVC include context adaptive entropy coding, multi-picture inter prediction, flexible MB ordering, intra coding prediction, addition of switching slices, quantization optimization, lossless MB coding etc. H.264/AVC (MPEG-4 Part 10) even allows MB partitions, for variable size block matching, such as $4 \times 4, 8 \times 4, 4 \times 8, 8 \times 8, 16 \times 8, 8 \times 16$ and 16×16 blocks as in Fig. 2b. The standard supports 21 different profiles and quite a few levels for application-oriented usage. The blocking artifacts that may incur after DCT-based compression, are then suppressed via a deblocking filter in H.264/AVC for better visual appearance as in Fig. 5.

Fig. 5 Without (*left*) and with (*right*) de-blocking filter

2.2.6 HEVC (H.265)

HEVC [73] or High Efficiency Video Coding, the successor of H.264/AVC, is the latest video standard which reduces the data rate required for high quality video coding by 50 % compared to the existing counterpart, H.264/AVC, at the expense of increased computational overhead. Detailed bit rate saving compared to other standards is shown in Table 2. The HEVC standard continues the block-based structure found in all video coding standards since H.261. HEVC can support video resolutions upto 8192 × 4320. The concept of MBs are superseded by coded tree blocks (CTB) using block structures varying from 16 × 16 to 64 × 64 instead of the fixed 16 × 16 blocks as in other coding standards. CTBs are further divided into Coding Units (CU) and Prediction Units (PU). The size of a prediction unit can vary from 4 × 4 to 64 × 64. Luma and chroma CTBs form a coding tree unit (CTU). High throughput oriented CABAC is the only entropy encoding scheme allowed in HEVC. It also allows four transform units (TUs) of sizes 4 × 4, 8 × 8, 16 × 16, and 32 × 32 to code the prediction residual. 33 intra prediction modes are permitted in HEVC compared to 9 in H.264/AVC. Maximum frame rate allowed is 300 fps.

2.3 MPEG-2 vs H.264

H.264 / MPEG-4 AVC and MPEG-2 are the most explored video standards for compressed domain analysis. Availability of improved coding techniques in H.264/AVC such as quarter-pel motion accuracy, new prediction modes for intra coded blocks, MB partitioning, integer transform, use of upto 16 reference frames, flexibility in entropy coding, unrestricted MV search range etc. make H.264/AVC superior to MPEG-2. Detailed differences are shown in Table 3.

Table 2 Average Bitrate savings based on equal PSNR [54]

Standard	H.264 HP	MPEG-4 ASP	H.263 HLP	MPEG-2 MP
HEVC MP	35.4 %	63.7 %	65.1 %	70.8 %
H.264 HP	–	44.5 %	46.6 %	55.4 %
MPEG-4 ASP	–	–	3.9 %	19.7 %
H.263 HLP	–	–	–	16.2 %

Table 3 MPEG-2 vs H.264/AVC

Features	MPEG-2	H.264/AVC
Motion vectors	Restricted to frame boundaries	No restriction
Motion estimation accuracy	Half-pel	Quarter-pel
Number of reference frames	1 for P frames and upto 2 for B-frames	Upto 16
Motion compensation block size	16×16, 16×8 and 8×16	16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4
Types of transform	Fixed point DCT	Simple integer transform
Entropy coding	Fixed (VLC)	Context adaptive (CAVLC/CABAC)
Spatial prediction types	None	Nine intra prediction modes
Deblocking filter	None	In-loop filter
Transform size	8×8	Adaptive (4×4, 8×8)
Picture coding type	Frame, Field, PicAFF	Frame, Field, PicAFF, MBAFF

3 Compressed domain video analysis

In this section, we will review the various compressed domain video analysis techniques developed in various compression standards. The topics covered include human action recognition, face detection, video classification, indexing, retrieval and object tracking.

Various parameters extracted from the compressed videos like motion vectors, MB size etc. are encoded with an aim of reducing the residual information. These are, therefore, very noisy and cannot be directly used for processing. Hence these are preprocessed before any further analysis, though not mentioned explicitly for every methodology in this section.

This paper covers various video analysis techniques proposed in the past 10–15 years across various compression standards. Most of these analyses were benchmarked with the authors' own datasets, which are not publicly available. This makes it very difficult to compare the performance of these algorithms on the basis of size, complexity of datasets and the evaluation methodology. Also, since the authors of various algorithms have not made their implementations available, it is not possible to directly compare them. In case we try to implement them, they may not be implemented in an optimized way as the authors would have done, which might lead to incorrect measurements.

3.1 Human action recognition

3.1.1 MPEG (MPEG-1, MPEG-2, MPEG-4 part 2)

Recognizing human actions in videos is one of the challenging areas of computer vision research. Applications of human action recognition include video surveillance, analysis of sports events, and patient monitoring. Day by day, the length and breadth of the problem statement is expanding. Challenges involved in recognition of actions are scale, appearance, illumination and orientation variations, occlusions, background clutter and camera motions. Researchers have come up with robust algorithms to tackle most of these challenges to a large extent. Still there is much left to explore in this field. The most popular publicly available databases for reporting recognition accuracy are Weizman [14], KTH [68], HMDB51 [41] and UCF101 [71]. The former two consists of relatively simpler actions while the

other two have complex actions with large intra-class variations and inter-class similarities, making them tough to crack, even in pixel domain analysis.

All of the notable activity recognition works in MPEG compressed domain have utilized cues only from motion vectors. Ozer et al. [56] proposed a hierarchical approach for human activity detection and recognition in MPEG compressed video sequences. Body parts were first segmented out and Principal Component Analysis (PCA) [58] was performed on the segmented MVs, prior to classification. However, the performance of the algorithm solely depends on the temporal duration of the activities. Later in [57], action recognition was done by creating eigenspace representation of human silhouettes obtained from AC-DCT coefficients. However, the method used compressed and uncompressed domain parameters. The low-resolution compressed domain data was connected with high level semantics in spatial domain to achieve real-time performance. Frames with specific postures were stored and global activity of the human body was estimated. This information was then used as an input in the pixel domain for gesture/action recognition. The first step retrieved possible frames in the compressed domain where people are present. The system then analyzed the extracted region for posture recognition. If a suspicious movement or human posture is detected, the next step is a more detailed investigation of the activity/gesture of the person in the uncompressed domain. First part of the algorithm was invariant to changes in intensity, color and textures.

Another notable work was put forward by Babu et al. [3] in MPEG compressed domain. The work proposed three feature extraction techniques for person independent action classification viz, *Projected 1-D feature* corresponding to the horizontal and vertical components of the MVs, *2-D polar feature* corresponding to a polar tiling of the (horizontal and vertical components) MVs and *2-D Cartesian feature* corresponding to a Cartesian tiling of the (horizontal and vertical components) MVs. The feature vectors were then fed to Hidden Markov Model (HMM) for classification of actions. Totally seven actions were trained with distinct HMM for classification. The performances of all the feature vectors were compared and the overall discriminating property of 2-D polar feature was found to be better than the other features. Recognition results of more than 90 % have been achieved.

Later, the well known motion history image concept was adapted to compressed domain action recognition by Babu et al. [5]. In this method, authors analyzed motion flow history (MFH) and motion history image (MHI) [18] constructed using MVs in MPEG compressed domain. *Projection profile* and *centroid-shift-based* features were extracted from the static MHI and *affine features* from MFH. Figure 6 illustrates MHI and MFH for *bend-down* action. Histogram of the horizontal and vertical components of the MVs were utilized to form the *Projected-1D feature*. Also, a *2D Polar feature* was developed using histogram of magnitude and orientation of MVs. The extracted features were used to train the KNN, Neural network, SVM and the Bayes classifiers to recognize a set of seven human actions and achieved more than 90 % recognition rate on their dataset with 7 actions.

After a long gap, DCT coefficients were employed for the first time along with MVs for action recognition and localization, in [92]. In this paper, the authors developed a high-speed algorithm in MPEG streams based on computing motion correlation measure, by utilizing differences in motion direction and magnitudes. The approach is based on computing a coarse estimate and a confidence map of the optical flow using MVs and DCT coefficients. However, the algorithm cannot handle scale variations. After the formation of optical flow, the approach is equivalent to any pixel-based algorithm. Hence the computational complexity is equivalent to pixel domain approaches. The results were reported on KTH dataset [68], but the videos with scale variations were excluded during the analysis.

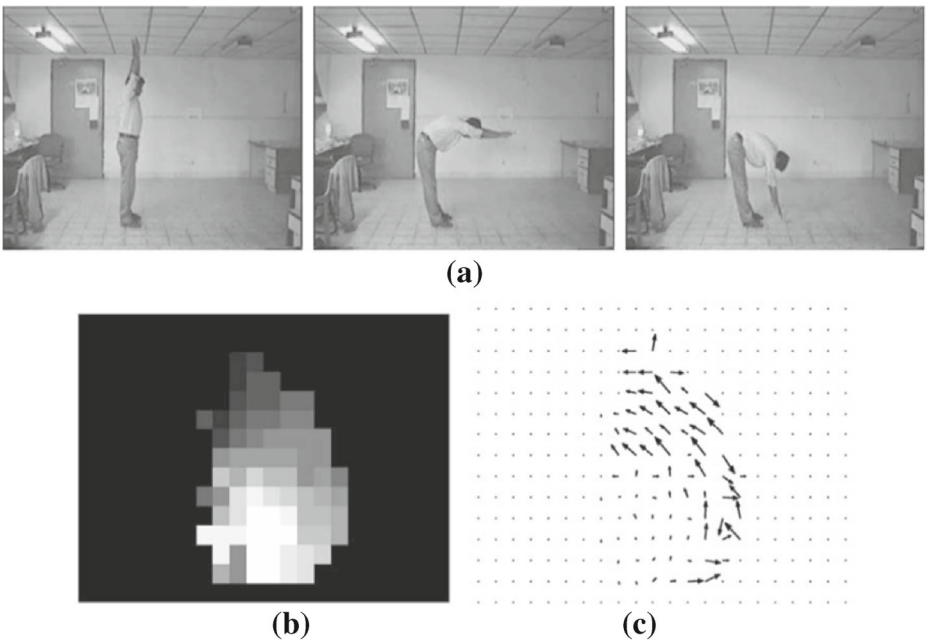


Fig. 6 **a** Key-frames of bend-down sequence and corresponding coarse **b** MHI **c** MFH [5]

Even though not a pure action recognition approach, a linguistic description of the motion of objects, which describes the object movement in the scene in a natural way, was proposed for MPEG streams in [67]. The algorithm utilized fuzzy logic. The inclusion of fuzzy logic helped in managing the noise inherent in the compressed domain input to a large extent. Initially, intra frame segmentation was performed by clustering of valid linguistic MVs based on a distance measure. Linguistic blobs in adjacent frames were then tracked temporally. Motion of these blobs was then modeled and their behavior was then recognized. The experimentation was performed on traffic videos with QVGA resolution. The objects were the vehicles performed different actions like stopping, turning etc.

3.1.2 Summary

Since all the MPEG compressed domain action recognition approaches depend on the block level features, these algorithms are naturally robust to appearance variations. However, almost all these works rely on MV information only. Other compressed domain parameters such as MB sizes (in bits), MB partition information, residuals, color information, quantization parameters etc. are still to be explored for action recognition. Almost all the reported research, in compressed domain, recognized only simple actions like running, jumping, waving hands, etc., performed by single subject at a time.

3.1.3 H.264/AVC (MPEG-4 part 10)

A few approaches were implemented on H.264 videos for action recognition. Tom et al. [79] proposed a fast algorithm for human action recognition. The algorithm utilizes cues from quantization parameters and motion vectors extracted from the compressed video sequence

for feature extraction and further classification using Support Vector Machines (SVM). Also, it can handle illumination changes, scale, and appearance variations, and is robust in outdoor as well as indoor testing scenarios. It extracts the Quantization Parameter Gradient Image (QGI) and the motion vectors (MV) and uses them to form a feature vector. The feature vector is then compared with the feature vector of other video sequences and the action is determined using the SVM classifier. The experimentation on various standard datasets on QCIF videos gives 85 % accuracy at 2000 fps for the classification of 7 actions like walking, running etc.

Rangarajan et al. [64] had used a similar approach using QGI and motion vectors. They proposed a new classifier, Projection Based Learning of the Meta-cognitive Radial Basis Function Network (PBL-McRBFN). The performance was shown to be improved by using the same features with this new classifier as compared to the conventional SVM based classifier.

Biswas et al. [10] proposed an algorithm to detect anomaly by utilizing cues from the motion vectors in H.264/AVC compressed videos. It is principally motivated by the observation that motion vector magnitude exhibits different characteristics during anomaly. The approach proposes hierarchical processing where detection starts at coarsest level upto the final one. Gaussian Mixture Model (GMM) is used classify the anomalous behavior from the usual one. This approach is further improved in [11] by adding orientation information for the motion vectors. They also used non-parametric modeling against the earlier parametric one, which helped in improving the accuracy of detection. The experimental results show that the algorithm processes at 70 fps.

3.1.4 Summary

Tom et al. [79] used QGI, attaining a very high speed as quantization parameter can easily be processed to detect the motion region in the image sequence. Further MVs are used to recognize the action performed in the sequence. Similar features were used by Rangarajan et al. [64], achieving better performance owing to their proposed PBL-McRBFN classifier instead of SVM.

Biswas et al. [10] have tried to detect anomalies in the motion pattern using motion vector magnitudes. Biswas and Babu [11] improved the algorithm by using orientation of motion vectors. The hierarchical approach in these methods aims at improving the processing speed.

Table 4 summarizes the technical details of the compressed domain works discussed above. The accuracy values are obtained from the graphs and results based on the experiments done by the respective authors. Since the experiments are not performed on a uniform database, the values cannot be directly compared.

3.2 Video classification, indexing and retrieval

3.2.1 MPEG (MPEG-1, MPEG-2, MPEG-4 part 2)

While handling visual media, there is a demand for indexing and retrieval of visual information from a huge multimedia databases, due to large storage space and processing power requirements. Video indexing and retrieval are closely related to summarization. If summary pertains to a single video, annotation and indexing are for a whole digital video database or library, containing large number of videos. An ideal video-retrieval system should provide with an abstract of the video content with minimal computation overhead.

Table 4 Summary of works done for human action recognition in compressed domain

Author	Std.	Approach	Features	Databases	Performance
Ozer et al. [56]	MPEG 1/2	Graph Matching, Super ellipse fitting	MV, DCT coeff.	MPEG-7	70 %
Ozer et al. [57]	MPEG 1/2	Partial spatial domain processing	MV, DCT coeff.	Own	95 %
Babu et al. [3]	MPEG 1/2	Hidden Markov Model (HMM)	MV based 2D cartesian, polar features	Own	90 %
Babu et al. [5]	MPEG 1/2	1D, 2D polar projection	MFH, MHI	Own	90 %
Yeo et al. [92]	MPEG 1/2	Confidence map of Optical Flow	MV, DCT coeff.	Own	90 %
Albusac et al. [67]	MPEG 1/2	Fuzzy logic	MV	–	–
Tom et al. [79]	H.264	SVM classification	QGI, MV	Weizman, KTH	85 %
Biswas et al. [10]	H.264	Hierarchical computation, GMM	MV	UCSD, Ped1, Ped2, UMN	–
Rangarajan et al. [64]	H.264	PBL-McRBFN	QGI, MV	Weizman, KTH	90 %

Eng et al. [22] introduced a video indexing technique in MPEG platform by extracting motion trajectory based on the motion vectors, for unsupervised object segmentation followed by tracking of each segmented moving object. Noise Adaptive Soft-Switching Median (NASM) filter was employed to remove the noisy MVs from the MV field without removing fine motion details. The filtered MVs were then clustered into separate homogeneous MV groups. They proposed an unbiased fuzzy clustering (UFC) technique for the same. UFC automatically identified the actual number of clusters by locating the position of each motion vector of the obtained MV clusters in the spatial domain. They also proposed backward projection and bidirectional motion tracking methodologies for different complexity levels of motion behavior.

Tan et al. [75] developed an algorithm based on estimating camera motion for determining certain characteristics of the content using the low-level information extracted from MPEG-compressed video. Pan, tilt and zoom type camera motions were estimated from the P-frames without feature selection and feature matching. The six-unknown parameters of the projective transformation from the previous anchor frame to the current P-frame were estimated. The algorithm was then applied on four basketball test sequences for testing. Full court advances (FCAs), Fast breaks (FBs) etc. were detected and used for detecting the event of shots at the basket by coupling the information from the estimated camera motion. Temporal segmentation was then done and the basketball video was classified into wide-angle and close-up video shots. Thus, an annotation file of basketball video content was generated by gathering data such as video shot boundaries, wide-angle and close-up shots, camera motion, the locations of full court pan, FBs, rebounds, shots at the basket, etc.

Yu [96] developed a video analysis and indexing system, in MPEG compressed domain, for Air Traffic Service (ATS) surveillance (a ground-based system that enables the identification of aircraft) and sports videos, by integrating the domain specific knowledge and pattern recognition techniques. By combining the transform domain as well as pixel domain features, the approach was able to exhibit robustness against illumination changes and distractors, while estimating the face poses. DC image, MVs, texture, color and edge information were utilized for feature formation. A coarse-to-fine strategy was also introduced for mining the semantic concept of clear face within a priori context. In addition, the author has analyzed and reorganized the collected mid-level features within each shot for robust video shot classification. This was then used for developing a semantic indexing hierarchy, which expressed the general understanding of semantics in sports videos made within a production environment.

Later, Mezaris et al. [49] presented a real-time, unsupervised segmentation of image sequences applied to video indexing and retrieval using motion as well as color information in MPEG-2 compressed domain. Spatio-temporal moving objects were first segmented out and tracked using iterative MB rejection technique. After foreground spatio-temporal objects had been extracted, background segmentation was performed based on classifying the remaining MBs to one of a number of background spatio-temporal objects. Ontology was used to facilitate the mapping of low-level descriptor values to higher level semantics. An object ontology and a shot ontology were employed to enable the user to form a simple qualitative description of the desired objects and their relationships in the shot respectively. After narrowing down the search to a set of potentially relevant spatio-temporal objects, relevance feedback [28] was employed to produce a qualitative evaluation of the degree of relevance of each spatio-temporal object.

For video retrieval, Babu et al. [4] presented a system to extract the object-based and global features from compressed MPEG video using the MV information. Initially, the MVs were accumulated over a few frames from the reliable MBs. The temporally accumulated

motion vectors were then subjected to median filtering and further spatially interpolated to get the dense motion field. Subsequently, the global parameters like motion activity and camera motion were extracted. The object features such as speed, area and trajectory were then obtained after a segmentation process. The number of objects in a given video shot was determined by K-means clustering procedure and the object segmentation was done by applying expectation maximization algorithm. The global and object features with the user given weights were used for retrieval.

3.2.2 Summary

There is a great deal of variation in the approaches for indexing and classification of MPEG videos. Tan et al. [75] used the P-frames to mark the shot boundaries, camera motion etc. The implementation is very specific to sports videos as it doesn't look for specific objects in the scene. Eng et al. [22] used fuzzy clustering to classify MVs into separate homogeneous groups. Object segmentation and ontology were used for mapping different objects and their relationships by Mezaris et al. [49]. All these are purely based on compressed domain analysis, whereas Yu et al. [96] partially exploited the pixel domain to exhibit robustness to the methodology. Utilizing cues from the pixel domain calls for a trade-off between processing speed and reliability.

3.2.3 H.264/AVC (MPEG-4 part 10)

Biswas et al. [9] captured orientation information from the MVs to classify H.264 compressed videos based on the action content. This approach utilized the fact that motions corresponding to similar videos (having similar actions) will follow similar orientation pattern. They have proposed histogram of oriented motion vectors (HOMV) for partially overlapping hierarchical space-time cubes to define the motion characteristics. Bag of Features (BOF) approach was further used to define the video as histogram of HOMV keywords, obtained using k-means clustering. A video is expressed as a combination of key oriented histograms which results in single feature vector per video. This is the first work reported for large scale video classification in compressed domain.

Recently, Mehrabi et al. [48] used color histogram feature of DC-pictures (derived from I frames) for content-based information retrieval in H.264/AVC compressed domain. Initially, the DCT coefficients were extracted and were then utilized to calculate the DC values for each sub-block. The DC values were then used to generate a lower quality DC picture and the respective color components were utilized to form color histogram as the feature vector for retrieval. The approach has higher computational advantage due to the fact that it is independent of non-I frames in a GOP. However, the method can only be applied to I-frames in the H.264 video and the accuracy of the algorithm will come down in case of higher GOP encoding.

Table 5 summarizes the technical details of the compressed domain works discussed above.

3.3 Moving object tracking

3.3.1 MPEG (MPEG-1, MPEG-2, MPEG-4 part 2)

One of the leading research areas of computer vision is moving object tracking with wide applications in surveillance, navigation, transportation monitoring, human computer

Table 5 Summary of works done for video classification, indexing and retrieval in compressed domain

Author	Std.	Approach	Features
Eng et al. [22]	MPEG 1/2	NASM filter, unbiased fuzzy clustering	MV
Tan et al. [75]	MPEG 1/2	Camera motion estimation	MV
Yu [96]	MPEG 1/2	Partial pixel domain computation	MV, DC image
Mezaris et al. [49]	MPEG 1/2	Object ontology	MV, DCT coeff.
Babu et al. [4]	MPEG 1/2	K-means clustering	MV
Yu et al. [97]	MPEG 1/2	Projective camera model	MV
Biswas et al. [9]	H.264	Histogram of oriented MV	MV
Mehrabi et al. [48]	H.264	Color histogram	MV, DCT coeff.

interaction, and robotics. The major trade off in visual tracking is between accuracy and processing speed. The challenges involved are scale changes, sudden illumination variations, partial occlusion, pose changes, and view point changes. Significant research has been done in this area. Still, robust tracking remains a challenge for computer vision researchers. The compressed domain features were explored more in tracking algorithms, unlike action recognition. Color information, residuals, and DCT coefficients were employed apart from MVs for tracking algorithm in MPEG-2 domain.

One of the initial works in tracking was put forward by Favelli et al. [24] in MPEG-2 compressed domain. The object/region to be tracked, at MB level, was first identified manually, called the *marking* process. The MB information in adjacent frames was utilized to find the new position of the tracker. The algorithm tracks the object through the video autonomously using the MVs associated to the MBs. If the visual information of a MB overlaps that of a MB in a neighboring position (in the next frame) by more than 25 %, then the new MB will also be considered part of the object, and both MBs will be tracked. If this quantity exceeds 75 %, only the new one will be considered. The approach was very naive but still was able to track objects for a span of 100 frames. However, block matching errors can cause the object to be truncated or disappeared.

Lie et al. [42] adopted segmentation-free *detection by tracking* strategy in MPEG-2 domain by linking MBs in the temporal domain and then pruning and merging the formed paths by considering spatial adjacency property. They used AC spectrum energy along with information from the MVs for error free MB linking. Pruning and merging steps were performed to take care of the noisy paths generated after the linking step. Tracking of multiple objects was possible but the algorithm cannot handle moving camera scenarios.

Achanta et al. [1] used color information from intra frames to identify the object to be tracked and forward MVs in the P and B frames to track the object. As opposed to the related works till that time, object was tracked in the B frames too. Proceeding only with MVs for tracking introduced cumulative errors. Hence they used DCT values of Cr and Cb (Chrominance) in the intra frames to incorporate color based tracking, which involved in identifying the best image area that matches with the original object marked by the user. The system offered speed, simplicity and robustness against occlusion and camera motion. Camera pans were handled with MVs while zoom required the use of color information.

Dong et al. [21] presented a robust tracking system using MVs and associated residual, spatial and textural confidence measures that are derived from MPEG-2 compressed video. The object to be tracked was initialized in the first frame. The MVs hitting the object, together with their derived residual, spatial and textural confidence measures were then used for deciding the new position and size of the object. Changes of the object properties due to

occlusion and global illumination changes were simultaneously detected by observing the three confidence measures. Later, real-time object tracking in compressed domain was proposed in [20], which was purely dependent on motion and color cues generated from DCT coefficients in I-frames. The temporal motion and spatial color information were then fused by means of a posterior probability framework which allows the information from different measurement sources to be fused in a principled manner. The algorithm also selectively updates the background and reference model, avoiding false updates.

3.3.2 Summary

The approach followed by Favelli et al. [24] based on the MVs is very simple and easy to implement. But the performance is dependent on the block matching and thus highly noise prone. This problem was taken care by Lie et al. [42]. Noise immunity was induced by linking the MBs based on the AC spectrum energy in the MBs. The tracker by Achanta et al. [1] is more suitable for the specific case of home videos, with long shots, few special effects and objects of interest occupied large image regions as it is based on color for tracking. The initial work presented by Dong et al. [21] was robust to various changes like illumination as it was based on three different confidence measures. To make the performance real-time in [20], they modified it to take cues from the I-frames only. This, to some extent, hampered the robustness of the algorithm and made it sensitive to the GOP length.

3.3.3 H.264/AVC (MPEG-4 part 10)

Thilak et al. [77] used binary image information, optimal pixel classification and clustering to segment out the object to be tracked in H.264 platform. The segmentation/ detection algorithm is a simple threshold operation that classifies a pixel either as object or as background. The Probabilistic Data Association Filter (PDAF), employed in the approach, allows the detection algorithm to classify more than one cluster as target, as opposed to the other related works.

You et al. [94] proposed a feature-based dissimilarity energy minimization algorithm utilizing MVs and luminance signals to perform adaptive object tracking in H.264/AVC videos. First, a rough prediction of the position of each feature point was done using MVs. Further, the best position inside the given search region was found out using clues like texture, form, and motion dissimilarity energies. Later, they came up with a tracking algorithm utilizing probabilistic spatio-temporal MB filtering (PSMF) to segment and track multiple objects with real-time speed in H.264/AVC bitstream [95] as shown in Fig. 7. All the skip-MBs in P frames were first filtered out of analysis. Remaining MBs which comprise of several non-skip MBs connected in the horizontal, vertical, or diagonal directions were then clustered as shown in Fig. 7II. Then a spatial filtering was used to discard isolated MBs followed by a temporal filtering to remove erroneous MBs. To accurately refine and recover object trajectory, background subtraction was employed in I frames and motion interpolation in P frames.

Kas et al. [39] proposed an unsupervised MV based trajectory estimation approach for moving objects in H.264 compressed videos. Once the MVs were extracted, global motion estimation (GME) was performed. The outliers were then filtered and object detection was performed on the resulting masks followed by a simple matching algorithm for solving object correspondence. Object History Images (OHIs) were then employed to stabilize the trajectories and the center of gravity-based trajectories were represented by smooth splines. This algorithm can deal with moving camera scenarios also.

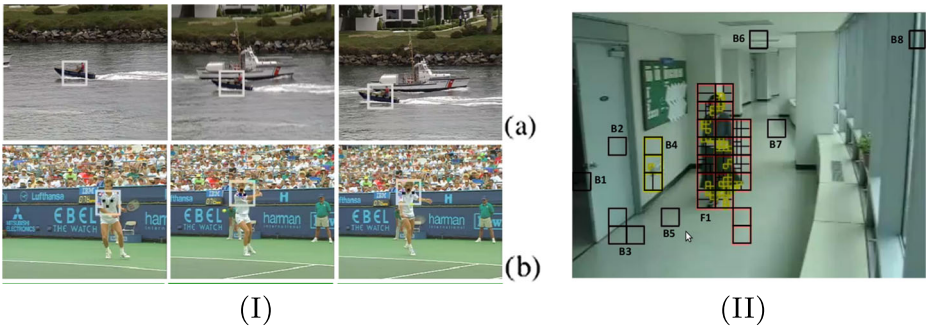


Fig. 7 **I** Object tracking in **a** Coastguard, **b** Stefan with 100 frames [94] and **II** Clustering of non-skip MBs [95]

Mehmood et al. [47] came up with real-time object tracking using motion information in H.264/AVC compressed domain and its SVC extension. First a frame was decoded to select the object to be tracked by defining a model using shape and coordinates. Also, MVs were extracted from the compressed video to update the tracking model. This approach calculates the new position of the target object using initial MVs after processing necessary tracking parameters. The video objects do not always correspond to the motion block shapes. Therefore the MVs that correspond to the blocks inside the target were used for predicting the motion of all the blocks related to that object. Object trajectories of intra coded pictures were derived from the inter coded pictures. To remove the spurious MVs, spatial median filter-based smoothening was done as illustrated in Fig. 8. Isolated motion vectors were smoothened in the areas that mostly correspond to the object boundaries.

Spatio-Temporal Markov Random Field (ST-MRF) model was used by Khatoonabadi et al. [40] for object tracking in H.264/AVC compressed domain, integrating the spatial and temporal aspects of object's motion. To enrich the MVs, preprocessing was done through intracoded block motion approximation and global motion compensation. The concept of polar vector median (PVM) was introduced to assign MVs to the intra coded blocks during the pre-processing step. The magnitude of PVM is the median of magnitudes of all the representative vectors while the orientation is the median of angles of selected representative

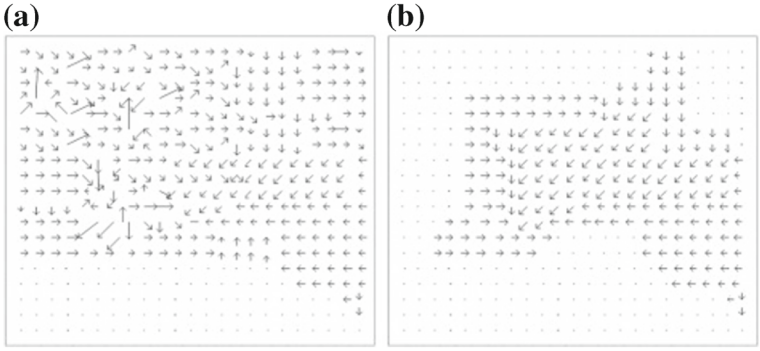


Fig. 8 Adaptive smoothing for noisy motion vectors **a** before processing and **b** after processing [47]

vectors. The ST-MRF model helps to optimize object tracking by referring to motion coherence and spatial compactness, as well as temporal continuity of the object's motion. In each frame, the method first approximates MVs of intra-coded blocks, followed by estimation of the global motion parameters, and then removal of global motion from the MV field.

3.3.4 Summary

Thilak et al. [77] implemented PDAF which detected multiple target clusters. Hence the method could handle the cases in which the target splits into many clusters or multiple clusters are detected (classified) as targets. Quite similar to this, You et al. [94] used the PSMF to mark the MBs as objects and then removed out the isolated noisy ones. The algorithm can track multiple objects with real-time accuracy but can be applied only for fixed camera scenarios. Also, the object should have a minimum size of two MBs to be tracked. Kas et al. [39] overcame this problem of fixed camera constraint. They used GME and OHI to take care of moving background. However, the moving objects should neither be too numerous nor should they occupy the whole viewable image area. Mehmood et al. [47] came up with the implementation that was not specific to any object or environment as they used only MBs that were entirely inside the target to predict and track the object. Khatoonabadi et al. [40] used ST-MRF model to improve the robustness of the algorithm. The approach is stable to a large extent. However, the algorithm fails to track while dealing with two or more nearby objects with comparable MVs.

Table 6 summarizes the technical details of the compressed domain works discussed above.

3.4 Moving-object-detection and segmentation

3.4.1 MPEG (MPEG-1, MPEG-2, MPEG-4 part 2)

Detection and segmentation of moving objects of relevance, in videos, are of great interest to researchers due to their wide applications in intelligent visual surveillance, video database browsing, human-computer interaction, and object-based video coding. Motion detection serves as the very first step in video analysis systems, used either for triggering

Table 6 Summary of works done for object tracking in compressed videos

Author	Standard	Approach	Features
Favelli et al. [24]	MPEG 1/2	Tracking macroblocks	MV
Lie et al. [42]	MPEG 1/2	Linking macroblocks	MV, DCT coeff.
Achanta et al. [1]	MPEG 1/2	Chrominance features	MV, DCT coeff.
Dong et al. [21]	MPEG 1/2	Residual, spatial and textural confidence measures	MV
Dong et al. [20]	MPEG 1/2	Process only I-frames	MV, DCT coeff.
Thilak et al. [77]	H.264	Probabilistic data association filter	MV
You et al. [94]	H.264	Dissimilarity energy minimization	MV, DCT coeff.
You et al. [95]	H.264	Probabilistic spatiotemporal macroblock filter	MB size
Kas et al. [39]	H.264	Global motion estimation	MV
Mehmood et al. [47]	H.264	Kalman filter	MV
Khatoonabadi et al. [40]	H.264	Spatio-temporal Markov random field	MV

alarms or to determine which video sequences need to be stored. Foreground segmentation is done by extracting the moving part from the background in a video sequence. Each frame in a video is segmented by means of automatic image analysis techniques. Unconstrained environments, object viewpoint variations, non stationary background, camera motion, and different object motion patterns are the various challenges involved.

The initial work of video segmentation, in MPEG-2 compressed domain, was developed by Mitsumoto et al. [52]. An initial segmentation was done by merging similar MVs utilizing the information from motion magnitude and direction. Noisy motion vectors were discarded by simple magnitude comparison, prior to processing. Motion estimation was then performed to track between adjacent frames. Each target region, a group of MBs, had its own averaged MVs. Simple interpolation was done to compensate for holes in the object after MV merging. Correspondences were determined using the similarity of those averaged MVs of each target region. In case of occlusion, appearance changes, physical collision etc., when the appropriate region could not be found, the tracking was suspended. The normalized distance predefined in the principal component space formed by the available DCT coefficients was used to recover matching of the suspended target. Yoneyama et al. [93] introduced a fast moving-object detection and identification algorithm in MPEG compressed videos by characterizing motion of the object in the coded domain. Motion information and DCT coefficients were employed for detection at MB level. Each moving object was then identified by figuring out the corresponding object in the adjacent frames using cues from object size, motion and position.

Sukmarg et al. [72] proposed one of the first works in MPEG domain for video object segmentation. The algorithm used clustering of the luminance and chrominance color components in the MPEG video followed by region merging based on spatio-temporal similarities. The result of spatio-temporal segmentation was then applied for foreground/background classification based on the average temporal change of regions. Eng et al. [23] designed and implemented an algorithm for unsupervised segmentation in MPEG compressed domain utilizing the homogeneity property of the spatio-temporally localized information from the moving video objects. Initially, the location of moving objects were found out using the MVs to perform temporal segmentation. Finer spatial-segmentation was then done using the DCT coefficients using maximum entropy fuzzy clustering algorithm. The small unidentified homogeneous regions formed around the video object boundaries were then classified as background or foreground using maximum a posteriori (MAP) estimate based on the correlations to their neighboring labeled regions.

Wang et al. [85] developed an algorithm for moving object extraction in MPEG-2 compressed domain using spatial, temporal, and directional confidence measures derived from the incoming stream. A combined confidence measure thus calculated was utilized to slice out the spurious MBs which may not be part of the moving object. Then one or more linear or non-linear motion filtering operations were performed to remove the holes occurred in the motion field. The dominant motion was then separated out by a recursive least square algorithm to get an object mask. K-means and/or EM clustering based on spatial and motion features were then performed to identify multiple objects, if present, followed by tracking of the objects based on their location and motion. One limitation of this work is the trade off between accuracy and false alarm rate. In terms of speed, this work was superseded by a real-time algorithm proposed by Benzougar et al. [7] using MVs and DCT coefficients. Initially, the dominant image motion between each P frame and the next reference frame (P or I) was estimated based on a M-estimator (Tukey's function) using affine model parameters of the forward MVs. A DC image [91], a spatially reduced version of the original image from DCT coefficients, was then formed. Global minimization of the energy function

formed using the DC Image and dominant motion information was then performed for final tracking. Even though the algorithm works in real-time, the accuracy is heavily dependent on the parameters selected for experimentation.

An algorithm for the automatic identification and coarse segmentation of video objects in the MPEG compressed domain has been developed by Jamrozik et al. [37]. First, region merging was done by applying leveled watershed techniques. The merging is based on the results of a motion map created from the MVs. The frames were selectively simplified by leveling, which utilized only the DC and first two AC coefficients, before the watershed transform was applied. Low resolution images thus generated were used as reference to reduce computational overhead in traditional leveled watersheding. Mezaris et al. [50] came up with an approach that utilized the MVs from the P-frame and color information from the DC coefficients of I frames. Further, an iterative rejection scheme based on the bilinear motion model was used to perform segmentation. The bilinear motion model was primarily used for modeling the camera motion. Later, a context-specific real-time unsupervised object detection approach was proposed in [51], which utilized the color and motion information along with a simple object model. Temporal tracking of the MBs was performed utilizing the associated motion information to detect the objects in P frames. On the other hand, detection in intra frames was done by utilizing Dominant Color Descriptor [45], clustering of MBs and model-based cluster selection.

Zeng et al. [100] computed Higher Order Statistics (HOS) on the inter-frame differences of partly decoded picture from the compressed video (DC image) for moving object extraction. After the DC image construction, the inter-frame difference was calculated between the current frame and the buffered DC frame. Then the background was detected by the second order statistic detector and the moving object by fourth-order moment measure. Finally, post-processing was performed for fine removal of noise signals. Yu et al. [98] used complimentary information from MVs and the DCT coefficients for robust moving object segmentation in MPEG compressed domain. First, clustering of the MVs was done to generate a motion mask for the moving regions in the motion clustering module. Also the foreground regions were subtracted from the DC image in I-frames and P-frames to produce a difference mask. The motion mask was used to exclude the DC coefficients from the moving regions and adaptively threshold the DC image. The motion mask and the difference mask were then combined conditionally based on the heuristic rules to generate the final object mask.

Babu et al. [6] introduced the concept of enriching the sparse information in MVs by accumulating over time for object segmentation with pixel accuracy. The number of moving objects present in the video were automatically estimated and then individually extracted. Temporally accumulated MVs are further interpolated spatially to obtain a dense field. Expectation maximization (EM) algorithm was then applied on the dense motion field for final segmentation. The number of appropriate motion models for the EM step were determined using a block-based affine clustering method and the segmented objects were temporally tracked to obtain the video objects. The segmentation results are shown in Fig. 9.

Porikli [61] developed a real-time object segmentation approach, that combined motion and frequency information, in MPEG compressed domain. Each GOP consisted of a layer of vectors that correspond to blocks in a frame. Selected AC-DCT coefficients and accumulated forward-pointing MVs were used to form each vector. A spatial filtering was also performed for the noisy MVs, prior to accumulation. A frequency-temporal data structure for the multiple GOPs, between two scene-cuts, was then constructed. Volume growing was then performed within the 3D data structure. The seeds for growing were chosen among the blocks with minimum local texture and gradient. The volume growing provided with

Fig. 9 Object segmentation results [6] (**a**) Original frame (**b**) Frame without the object (**c**) Segmented object

the connected parts of video that had consistent DCT coefficient and motion properties. Volume descriptors were then determined for each volume, including affine motion parameters, utilizing trajectories and MVs. Similar volumes were then merged pair-wise using their descriptors to obtain an object-partition tree for final segmentation. Recently, Porikli et al. [62] presented an automatic segmentation method that takes advantage of the inter-frame motion and intra-frame spatial frequency information embedded in MPEG compressed video exploiting the block and GOP structure. After parsing MVs and DCT coefficients, a multidimensional frequency temporal data structure was constructed by using multiple GOPs between two scene cuts. Initial segmentation was achieved using volume expansion of the video across each frame and across multiple frames. Clustering of the frequency temporal blocks in multiple kernels of spatial, motion, and frequency domains was performed using mean-shift. Then, an iterative merging of the similar volumes was done using their descriptors to obtain a hierarchical object partition tree. At each iteration, the volume descriptors are updated that consists motion and DCT-based terms. Unlike other approaches prevalent at that time, the algorithm processes a multitude of GOPs at the same time.

Qiya et al. [63] focused on a coarse to fine strategy for real-time moving object extraction in MPEG-2 compressed domain. Initially, to extract the object at a coarse level, fusion of contour-feature (developed by automatic seeding applied on the DC + 2 AC images of I frame) and MV based projection were performed. Then, the blocks in intra frames were partially decoded to refine the segmentation accuracy. One of the prominent works for background subtraction in MPEG 2 compressed domain was put forward by Wang et al. [87]. As opposed to the compressed domain counterparts which primarily depend on MVs, the algorithm used information only from the DC and AC coefficients of the DCT. Three different styles of background modeling approaches were presented viz., the running average (RA) algorithm, the median algorithm and the mixture of Gaussian (MoG) algorithm. The segmentation was done in two steps. First, a new background subtraction technique in the DCT domain was exploited to identify the block regions fully or partially occupied by foreground objects, and then pixels from these foreground blocks were further classified in the spatial domain. Pixel accuracy was achieved with comparable efficiency and less computation overhead.

MVs and DCT coefficients were filtered and manipulated to obtain a dense and reliable motion vector field (MVF) over consecutive frames in [84] to detect object and camera motion in MPEG-2 compressed domain. An iterative segmentation scheme based upon the generalized affine transformation model was then utilized to effect the global camera motion detection. The segmented foreground blocks were then temporally tracked using the dense MVF to ensure the temporal consistency of the segmentation. Iterative motion estimation and temporal tracking were performed at pixel level for refining the segmentation.

3.4.2 Summary

Yoneyama et al. [93] used MVs and DCT coefficients to segment out the objects in the scene. The algorithm is, however, not able to detect when the object moves along the focus axis of the camera. Also, the identification accuracy was heavily dependent on the detection results in the first phase. Sukmarg et al. [72] utilized the luminance and chrominance components for region merging which led to segmentation. Eng et al. [23] implemented a two level segmentation, first based on the MVs and the second on the DCT coefficients. Benzougar et al. [7] formed a DC image from the DCT coefficients for the global minimization of the energy function. Even though this works in real-time, the accuracy is heavily dependent on the parameters selected for experimentation.

Jamrozik et al. [37] used leveled watersheding to merge the various regions based on the MVs. Mezaris et al. initially created an object model for segmentation in [50]. This was an iterative process. Later in [51], they improved the algorithm using color information along with the object model, reducing the number of iterations. This improvement made the approach real-time. Yu et al. [98] used the MVs and the DCT coefficients for segmentation. The approach is however limited to fixed camera applications. Babu et al. [6] approach involved the EM algorithm for segmentation. They were able to detect the number of moving object in the scene.

Contrary to many other approaches, Zeng et al.[100] did not use the MVs. Instead they processed the DC image using DCT coefficients, which made the algorithm immune to the noise in MVs. One more approach which relied primarily on DCT coefficients was the one adopted by Wang et al. [87]. They were able to achieve pixel level accuracy with comparable efficiency and less computation overhead.

3.4.3 H.264/AVC (MPEG-4 part 10)

Zeng et al. [99] utilized MVs for object segmentation in H.264 compressed videos. Moving objects were extracted from the motion field through the Markov Random Field (MRF) classification process. The MVs were first classified into several MV types. Different MV types provided different contributions to the segmentation process. Further, moving blocks were extracted by the MRF classification. For object segmentation in I-frames, an object label projection scheme was implemented to track the segmentation results of the previous P-frame, and the labels were projected to the current I-frame by inverting the MVs of the previous P-frame. However, the algorithm suits only for segmenting videos captured from the fixed cameras.

Unlike the previous works, Hong et al. [29] proposed a moving object segmentation approach in H.264 compressed domain considering moving camera scenarios. They have used cues only from the block partition modes and MVs in the compressed bit stream. Appropriate MVs were chosen according to the partition modes to estimate the global motion, which was then subtracted from all the relevant MVs to cancel out the camera motion. Intermediate segmentation results were thus obtained by just using the MVs. Different weightings were employed later for different partition modes to enhance the segmentation, followed by spatial and temporal filtering and an adaptive thresholding to refine the result.

Liu et al. [43] segmented video objects in the H.264 compressed domain exploiting cue only from MV field. Initially, they performed temporal and spatial normalization of the MV field. Normalization is done according to the temporal distance and the direction indicated by the reference frame index. Then the motion saliency was enhanced by MV accumulation

using an iterative backward projection scheme followed by global motion compensation on the accumulated MV field. For each current frame, the normalized MV fields of subsequent inter coded frames were backward accumulated to obtain a salient MV field. The hypothesis testing using the residuals of global motion compensation was employed for intra-frame classification of segmented regions, and the projection was exploited for inter-frame tracking of previous video objects. A correspondence matrix based spatiotemporal segmentation approach was then performed to segment video objects in real-time under different situations such as appearing and disappearing objects, splitting and merging objects, stopping moving objects, multiple object tracking and scene change.

One of the prominent works on moving-object detection in H.264/AVC compressed domain was reported by Poppe et al. [60]. Almost all the related works till that time heavily depended on MVs. On the contrary, Poppe et al. proposed segmentation using size of MBs (in bits) after compression as the main cue. To achieve SubMB-level (4×4) precision, the information from transform coefficients was also utilized. The system achieved high execution speeds, upto 20 times faster than the MV-based related works. Analysis was restricted to P frames and a simple interpolation technique was employed to handle intra (I) frames. The whole algorithm was based on the assumption that the MBs that contain an edge of a moving object are more difficult to compress since it is hard to find a good match for this MB in the reference frame(s). During the training phase, the number of bits that MBs use within a frame, were used to create an effective background model. They followed a hierarchical approach in which an initial segmentation was performed at MB level (16×16) using spatial and temporal filtering operations. Then, the segmentation accuracy at the boundary foreground MBs were improved to SubMB level (4×4) using transform coefficients.

MVs are generated during the motion estimation process aiming highest achievable compression. These noisy MVs, if directly used for segmentation, will degrade the overall performance of the system. Bruyne et al. [19] estimated the reliability of MVs by comparing them with projected MVs to generate motion similarity measures in H.264/AVC domain from neighboring frames to filter out and localize the noisy MVs. This information along with the MV magnitude was used to segment out the foreground objects from background.

Cipres et al. [70] proposed an approach in H.264 compressed domain based on fuzzy logic to detect the moving objects. The MVs were converted into linguistic MVs in the fuzzification step. The fuzzy sets suppress the noise inherent to the encoding process and obtain conceptual representations that describe the regions detected in a comprehensive way. The valid MVs were then grouped into linguistic blobs, each of them could be identified as a moving object in the video scene. Finally the linguistic blobs were filtered to delete noisy MVs. By using approximate reasoning and a clustering algorithm, the segmentation method obtained the moving regions of each frame and described them with common terms like shape, size, position and velocity.

Szczerba et al. [74] used temporal and spatial relations of the MVs to generate a Bayesian probability based confidence measure for each motion vector. This measure represented the likeliness of the MV to represent real motion. Temporal confidence was based on correlation of temporally adjacent MVs and spatial confidence was based on spatial clustering of (high) confidence MVs. Based on the spatio-temporal confidence array the final binary motion mask was extracted, representing the final segmentation.

Mak et al. [44] proposed an algorithm that utilized the motion information from the H.264 bit stream form background motion model and segment moving objects. Markov random field (MRF) was employed to model the foreground field so that the spatial and

temporal continuity of objects was preserved. Quantized transform coefficients of the residual frame were then used to improve the segmentation result.

Verstockt et al. [81] constructed a real-time object localization technique in H.264/AVC domain. The algorithm consisted of three steps: MB-based foreground segmentation, object (group) extraction and multi-view object localization. MBs were segmented as foreground and background MBs by comparing the current MB partition modes with the partition mode of the previous and next frames. Only MBs with a stable foreground partition mode were classified as foreground. Object extraction was done by blob merging and convex hull fitting. Finally, each object was located on a ground plane by exploiting the homography constraint. The algorithm was able to fuse four different views and correctly localize objects in real-time.

Niu et al. [53] utilized temporal and spatial correlation of motion to refine MVs and initial segmentation was produced by MV differencing. Then the segmentation was further improved by using intra prediction information from the I-frame. The improved segmentation was projected on to the subsequent frame followed by expansion and contraction operations for refinement. But this method can handle only fixed camera scenarios.

Ant colony clustering algorithm was employed by Pei et al. [59] for moving object segmentation in the H.264/AVC compressed domain. At first, they used coarse grouping of MVs based on the similarity in magnitude and orientation. MV field was then classified into fine clusters by ant colony algorithm in which each MV was defined as an ant with features, magnitude and orientation. Moving objects were finally segmented out by exploiting the orientation histogram of the MV field and the final cluster centers.

Kapotas et al. [38] proposed a static-camera-based moving object detection algorithm for H.264 videos using MVs and variable block sizes used in the inter mode decision. Initially, the pixels in each block were classified as static or moving based on the magnitude of the MV. Then the successive inter frames were accumulated and their moving pixels were merged to obtain the complete contour of the moving object.

Fei et al. [25] presented a mean shift clustering-based moving object segmentation approach at subMB level (4×4) accuracy in the H.264 compressed domain. The reliability and saliency of the MVs were first improved by normalization, weighted 3D median filter and motion compensation. The partitioned block size was used as a measure of motion texture of the MV field. Mean shift-based mode seeking in spatial, temporal and range domain was then employed for compact representation of the MV field. The MV field was further segmented into different motion-homogeneous regions by clustering the modes with small spatial and range distance, and each object was represented by some dominant modes.

Vacavant et al. [80] presented a multi-modal background subtraction technique using the size of MBs for fast moving object segmentation in H.264/AVC domain. They integrated and compared the Gaussian Mixture Model (GMM) and the VuMeter (VUM) [27] in order to build adaptive background models based on MB sizes. The core idea behind the approach was that a MB representing a moving object should be more voluminous than if it contains only background.

Chen et al. [16] put forward an unsupervised segmentation algorithm using global motion estimation and Markov random field (MRF) classification. First, MVs were compensated from global motion and quantized into several representative classes to remove camera motion, from which MRF priors were estimated. Then, a coarse segmentation map of the MV field was obtained using a maximum a posteriori estimate of the MRF label process. Finally, the boundaries of segmented moving regions were refined using color and edge information. Chen et al. [15] analyzed the shape and MV homogeneity of the segmented objects for car and human identification in H.264 framework. After segmenting the moving

object based on clustering MVs and Markov Random Field (MRF) iteration, features were extracted based on motion analysis to obtain the difference of MVs direction (dMVD) and shape analysis to find the number of MBs (nMB). Object classification was then performed using Bayesian classifier. The distinction between cars and humans was based on the number of MBs and the motion similarity of the moving objects. The dMVD was also utilized to distinguish human and car objects based on shape difference and MV homogeneity between human and car.

Yang et al. [90] came up with a coarse to fine segmentation methodology in H.264 compressed domain using the spatial and temporal correlations among adjacent blocks and spatio-temporal Local Binary Pattern (LBP) features of MVs followed by a boundary modification based on DCT coefficients. Initial motion accumulation and filtering was done to achieve reliable MVs.

Wang et al. [86] proposed a segmentation scheme based on intra coding features such as DC values in luma and chroma color space and low-frequency residuals in luma color space. An advantage of this approach is that it allows to vary the quantization parameters unlike the other related works, essentially helping the system to adapt for varying network bandwidth. The intra coding features were then utilized to construct a background codebook model for extracting foreground energy frame. By subtracting the background codebook models, the foreground energy frame was filtered and normalized for determining the existence of moving objects. Thresholds were then obtained automatically to enable unsupervised searching and to overcome the over-segmentation problem. Noisy objects detected were then discarded by connected component labeling and morphological filtering process. However, the approach can be employed only to fixed camera scenarios with very short GOP length. Wang et al. [82] proposed a background modeling technique in H.264 using MVs and DCT coefficients. First, a simple median filtering step based on MV amplitude was performed to correct the noisy motion vectors. Then, initial coarse segmentation was done utilizing the Local Binary Pattern (LBP) [55] of the pre processed MVs. As illustrated in Fig. 10, the MV-LBP values thus generated were introduced into background modeling to extract primary moving objects. The LBP feature was extracted based on the block partitions. The block will be set to 1 if its MV amplitude is higher than that of the current block and 0, if lower. DCT coefficients were then accumulated in the temporal domain to modify the segmentation.

Most of the algorithms use the motion vectors to detect a moving object in the video. Instead, Tom et al. [78] proposed an algorithm based on the MB size and the Quantization Parameter in the H.264 videos. In this approach Quantization Parameter Gradient Image (QGI) is used to detect the foreground MBs while the MBs on the boundary of foreground and background are detected using the QGI and Macro Block (MB) size. A two-tap filter is used to interpolate the MB size information to get the finer boundary of the foreground object.

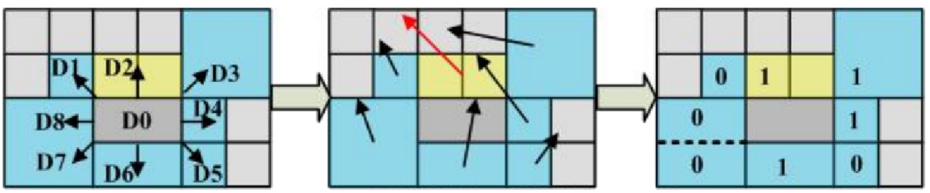


Fig. 10 LBP formation. Extracted LBP value is 01110100 = 116 [86]

Hybrid approaches utilizing pixel as well as compressed domain information were also proposed. MVs and DCT coefficients from H.264 compressed videos were first denoised by Ibrahim et al. [30] and then combined via simple thresholding for segmentation. Localized spatial processing was then performed in the motion region by applying inverse DCT. To obtain pixel level information, a simple frame differencing was done using the IDCT information from the reference I frame. A preset threshold on this difference was then used for extracting moving object, maintaining its shape and contour.

Zeng et al. [99] used MRF to classify the MVs. The algorithm suits only for segmenting videos captured from the fixed cameras. Hong et al. [29] processed the MVs to estimate the global motion. Due to this, their approach was capable of handling the case of moving camera as well. Poppe et al. [60] used the size of MBs to detect the moving objects. This led to very fast processing compared to other techniques. However, the work was restricted to static camera scenarios only as it was primarily designed for video surveillance applications. Cipres et al. [70] followed fuzzy logic approach to process the MVs and formed linguistic blobs. This led to the reduction of noisy MVs and better segmentation. Mak et al. [44] used MRF to model the foreground field. However, the approach works well for larger objects only and the quantization parameters needed to be fixed during the analysis.

Kapotas et al. [38] processed the MVs to gradually detect the moving object's contour. However, their approach has two major disadvantages. The accuracy of the method, especially the detection of an object's contour, heavily depends on the number of the sub-blocks during the motion estimation. Also, the lack of sufficient number of sub-blocks occupied by the object, either due to high QP or slow motion may lead to crude object detection. Moreover, the method cannot handle complex motions like the overlapping motions of two or more moving objects.

Unlike most algorithms, Tom et al. [78] processed the QGI and MB size information to segment the foreground objects. The algorithm achieved a computational speed of 500 fps on CIF videos. Other approaches included hybrid processing i.e. in pixel as well as compressed domain. One such method used by Ibrahim et al. [30] where they processed MVs and DCT coefficients to improve the speed, and then calculate the IDCT of the I frame to conserve the contour of the segmented objects. The algorithm was able to detect the objects with sizes less than a MB.

3.4.4 Crowd flow segmentation

One of the applications similar to object segmentation is dominant flow detection. Here, the aim is to segment the dominant crowd flows present in the scene. This helps us to model the flow at each location and detect abnormal flows. Ali et al. [2] proposed an approach, in pixel-domain, to segment the flows by observing the particle flows over optical flow field. Praveen et al. [26] tackle this problem in compressed domain by clustering the dominant pattern of motion vectors using Expectation-Maximization algorithm. The clusters which converge to a single flow are merged together based on the Bhattacharya distance measure between the histogram of the orientation of the motion vectors at the boundaries of the clusters.

Biswas et al. [12] use the collective representation of the motion vectors of the compressed video sequence, transform it to a color map and perform super-pixel segmentation at various scales for clustering the coherent motion vectors. The major contribution of this paper involves obtaining the flow segmentation by clustering the motion vectors and determination of number of flow segments using only motion super-pixels without any prior

Fig. 11 Crowd flow segmentation by Biswas et al. [12]

assumption of the number of flow segments. The segmentation result for a specific video is shown in Fig. 11.

Though compressed domain approaches perform better than pixel domain approaches for various videos, they fail to capture the flow segments when the crowd flow magnitude is very small and was not captured by motion vectors.

3.4.5 Summary

Tables 7 and 8 summarize the technical details of the compressed domain works discussed above in the MPEG 1/2 and H.264 standards respectively. The performance values are obtained from the results of experiments done by the respective authors. These experiments are performed on different machines and on different databases. Hence these values cannot be directly compared, but are useful to get a performance estimate with respect to the parameters and the methodology used by the authors. For example, it can be observed from

Table 7 Summary of works done for object segmentation in MPEG 1/2 compressed videos

Author	Approach	Features	Speed
Mitsumoto et al. [52]	MB merging, DCT clustering	MV, DCT coeff.	5–10 fps
Yoneyama et al. [93]	Track multiple moving objects	MV, DCT coeff.	–
Sukmarg et al. [72]	Region merging graph	MV, DCT coeff.	–
Eng et al. [23]	Maximum entropy fuzzy clustering	MV, DCT coeff.	–
Wang et al. [85]	3 confidence measures	MV, DCT coeff.	2 fps
Benzougar et al. [7]	Markovian labeling	MV, DCT coeff.	5–10 fps
Jamrozik et al. [37]	Watershed leveling	MV	–
Mezaris et al. [50]	Bilinear motion model	MV, DCT coeff.	600 fps
Mezaris et al. [51]	K-means, color descriptor	MV, DCT coeff.	60 fps
Zeng et al. [100]	High order statistics	DCT DC coeff.	–
Yu et al. [98]	Motion clustering	MV, DCT DC coeff.	–
Babu et al. [6]	EM, Affine clustering	MV	–
Porikli et al. [62]	Region expansion, Mean shift	MV, DCT coeff.	666 fps
Qiya et al. [63]	Partial Decoding (DC+2AC coefficients)	MV, DCT coeff.	42 fps
Patel et al. [84]	Motion vector field for tracking	MV, DCT coeff.	–
Wang et al. [87]	Running Avg., Median filtering, MoG	DCT coeff.	–
Porikli [61]	Hierarchical clustering	MV, DCT coeff.	666 fps

Table 8 Summary of works done for object segmentation in H.264 compressed videos

Author	Approach	Features	Speed	Performance
Zeng et al. [99]	Block based MRF	MV, MB size	2-20 fps	85 %
Cipres et al. [70]	Fuzzy logic	MV, Decision modes	25 fps	77 %
Mak et al. [44]	Markov Random Field	MV	53 fps	84 %
Chen et al. [15]	MRF, Bayesian classifier	MV	14 fps	94 %
Chen et al. [16]	MRF, Global motion estimation	MV	9 fps	74 %
Yang et al. [90]	LBP, coarse-to-fine segmentation	MV, DCT coeff.	15 fps	79 %
Poppe et al. [60]	MB size and transform coefficients	MB size, DCT coeff.	570 fps (At 50 % recall)	50 – 80 %
Ibrahim et al. [30]	Compressed and spatial domain	MV, DCT coeff.	–	–
Hong et al. [29]	Global motion vector	MV	39 fps	–
Liu et al. [43]	Global motion compensation	MV	25 fps	–
Verstockt et al. [81]	MB partition information	MB size	–	–
Szczerba et al. [74]	Temporal and spatial analysis	MV	–	–
Pei et al. [59]	Ant colony clustering	MV	–	–
Kapotas et al. [38]	Classification and refinement	MV	–	–
Fei et al. [25]	Mean shift clustering	MV, MB size	37 fps	–
Vacavant et al. [80]	Mixture of Gaussians	MB size	–	–
Wang et al. [82]	Analyze only intra frames	MV, DCT coeff.	–	–
Wang et al. [86]	LBP, segmentation	MV, DCT coeff.	–	–
Tom et al. [78]	QGI, Temporal Accumulation	QGI, MB size	508 fps	–
Praveen et al. [26]	Expectation-maximization algorithm	MV	–	–
Biswas et al. [12]	Superpixel based clustering	MV	–	–

Table 8 that MRF based approaches are able to achieve better accuracy at reasonable speeds as compared to other methods.

3.5 Face detection

3.5.1 MPEG (MPEG-1, MPEG-2, MPEG-4 part 2)

The goal of face-detection is to determine whether or not there are any faces in the video, and if present, return the location and size of each face. The challenges associated with robust face-detection are face-camera relative pose changes, facial expressions, occlusion, illumination changes, imaging conditions, and presence or absence of facial features such as beards, mustaches, glasses etc. Face detection is one of the least explored computer vision areas, in compressed domain. Only two works were reported, that too on MPEG platform. The rich and varied features of the latest video standards such as H.264 and HEVC, with wider scope, are yet to be explored.

Unlike other vision areas, the possibility of utilizing motion vectors has not yet been tried out for face detection. Wang et al. [83] proposed an algorithm using inverse quantized DCT coefficients of MPEG video for detection of face regions. First, each MB was classified as a face MB or not using chrominance with the Bayesian minimum risk decision rule. The key to this classification is the uniqueness of human skin-tone colors. In the MB mask images, a

Table 9 Summary of works done for face detection in the videos in compressed domain

Author	Std.	Approach	Features	Detection
Chua et al. [17]	MPEG 1/2	Luminance and chrominance	DCT coeff.	85 %
Wang et al. [83]	MPEG 1/2	Skin color	DCT coeff.	85–92 %

Fig. 12 Face region detection results [83]

pixel with value *one* corresponds to a MB whose average color is a skin-tone color. Second, the face regions in MB mask images were detected using binary template matching. As the final step, verification of the detection was performed based on energy distribution of the DCT coefficients. The performance of the algorithm is relatively independent of lighting conditions, as shown in Fig. 12. However, the algorithm can be applied only for color images and videos because of the use of chrominance information.

Later, Chua et al. [17] put forward a frontal face detection method using the gradient energy representation extracted directly from the MPEG video, applicable for color as well as gray-scale images. The gradient energy allows highlighting of facial features of high contrast, such as the eyes, nose and mouth. Initially the gradient energy face model was used to locate potential face regions at multiple scales and locations. Then skin-color verification was done to eliminate falsely detected regions. A rule-based classifier was employed to perform detailed search for faces in the transformed contrast domain (gradient energy representation). In order to justify and optimize the intuitive rules, a neural network-based classifier was also designed. Both the classifiers together classified a gradient energy pattern as face or non-face and the parameters were learned from face and non-face samples.

Table 9 summarizes the technical details of the compressed domain works discussed above.

4 Conclusion

This paper provides a detailed overview of various state-of-the-art research works reported in compressed domain video analysis. Video analysis based taxonomy is chosen in order to present the work lucidly. The paper conveys the idea behind each approach and discusses its advantages and limitations. Various compressed domain approaches utilizing information from parameters such as motion vectors, transform coefficients, macroblock partitions, luminance and chrominance values, color, macroblock sizes etc. are discussed in detail. This

analysis spans through various computer vision applications such as moving object segmentation, human action recognition, indexing, retrieval, face detection, video classification, object tracking, video summarization, scene change detection in compressed videos.

There are many applications yet to be explored in the compressed domain. Video tampering detection is one such application where the aim is to detect whether the video is purposefully distorted after it has been shot. One other application in which compressed domain analysis can be used is moving object identification where different moving objects like car, person can be identified based on their motion. This paper contains approaches implemented only upto the H.264/AVC standard. In future, we expect to see the algorithms to be implemented on H.265/HEVC compressed videos as this standard has better compression capabilities than the currently prevalent H.264 standard.

Acknowledgments This work was supported by CARS (CARS-25) project from Centre for Artificial Intelligence and Robotics, Defence Research and Development Organization (DRDO), Govt. of India. The authors wish to express grateful thanks to the referees for their useful comments and suggestions to improve the presentation of this paper.

References

1. Achanta R, Kankanhalli M, Mulhem P (2002) Compressed domain object tracking for automatic indexing of objects in MPEG home video. In: IEEE international conference on multimedia and expo, vol 2, pp 61–64
2. Ali S, Shah M (2007) A Lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In: IEEE conference on computer vision and pattern recognition (CVPR), 2007, pp 1–6. doi:10.1109/CVPR.2007.382977
3. Babu RV, Anantharaman B, Ramakrishnan KR, Srinivasan SH (2002) Compressed domain action classification using HMM. Pattern Recog Lett 23(10):1203–1213
4. Babu RV, Ramakrishnan K (2007) Compressed domain video retrieval using object and global motion descriptors. Multimed Tools Appl 32(1):93–113
5. Babu RV, Ramakrishnan KR (2004) Recognition of human actions using motion history information extracted from the compressed video. Image Vis Comput 22(8):597–607
6. Babu RV, Ramakrishnan KR, Srinivasan SH (2004) Video object segmentation: a compressed domain approach. IEEE Trans Circ Syst Video Technol 14(4):462–474
7. Benzougar A, Bouthemy P, Fablet R (2001) MRF-based moving object detection from MPEG coded video. In: IEEE international conference on image processing, vol 3, pp 402–405
8. Bhaskaran V, Konstantinides K (1995) Image and video compression standards: algorithms and architectures. Kluwer Academic Publishers
9. Biswas S, Babu RV (2013) H.264 compressed video classification using Histogram of Oriented Motion Vectors (HOMV). In: IEEE international conference on acoustics, speech, and signal processing (ICASSP), pp. 2040–2044
10. Biswas S, Babu RV (2013) Real-time anomaly detection in H.264 compressed videos. In: National conference on computer vision, pattern recognition, image processing and graphics (NCVPRIPG), pp 1–4. doi:10.1109/NCVPRIPG.2013.6776164
11. Biswas S, Babu RV (2014) Anomaly detection in compressed H.264/AVC video. Multimed Tools Appl:1–17. doi:10.1007/s11042-014-2219-4
12. Biswas S, Praveen RG, Babu RV (2014) Super-pixel based crowd flow segmentation in H.264 compressed videos. In: International conference on image processing
13. Bjontegaard G, Lillevold K (2002) Context adaptive VLC coding of coefficients. ISO/IEC Joint Video Team C028
14. Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as space-time shapes. In: IEEE international conference on computer vision, pp 1395–1402
15. Chen W, Yang QX, Lin KW, Wang SY, Huang CL (2011) Human and car identification using motion vector in H.264 compressed video. In: Visual communications and image processing, pp 1–4. doi:10.1109/VCIP.2011.6115985

16. Chen YM, Bajic I, Saeedi P (2011) Moving region segmentation from compressed video using global motion estimation and Markov random fields. IEEE Trans Multimed 13(3):421–431
17. Chua TS, Zhao Y, Kankanhalli MS (2002) Detection of human faces in compressed domain for video stratification. Vis Comput 18(2):121–133
18. Davis J, Bobick A (2001) The recognition of human movement using temporal templates. IEEE Trans Pattern Anal Mach Intell 23(3):257–267
19. De Bruyne S, Poppe C, Verstockt S, Lambert P, Van De Walle R (2009) Estimating motion reliability to improve moving object detection in the H.264/AVC domain. In: IEEE international conference on multimedia and expo, pp 330–333
20. Dong L, Schwartz S (2006) DCT-based object tracking in compressed video. In: IEEE international conference on acoustics, speech and signal processing, vol 2, pp II–II. doi:10.1109/ICASSP.2006.1660430
21. Dong L, Zoghlami I, Schwartz S (2006) Object tracking in compressed video with confidence measures. In: IEEE international conference on multimedia and expo, pp 753–756
22. Eng HL, Ma KK (1999) Motion trajectory extraction based on macroblock motion vectors for video indexing. In: International conference on image processing, vol 3, pp 284–288
23. Eng HL, Ma KK (2000) Spatiotemporal segmentation of moving video objects over MPEG compressed domain. In: IEEE international conference on multimedia and expo, vol 3, pp 1531–1534
24. Favalli L, Mecocci A, Moschetti F (2000) Object tracking for retrieval applications in MPEG-2. IEEE Trans Circ Syst Video Technol 10(3):427–432
25. Fei W, Zhu S (2010) Mean shift clustering-based moving object segmentation in the H.264 compressed domain. IET Image Process 4(1):11–18
26. Gnana Praveen R, Babu RV (2014) Crowd flow segmentation based on motion vectors in H.264 compressed domain. In: 2014 IEEE international conference on electronics, computing and communication technologies (IEEE CONECCT), pp 1–5. doi:10.1109/CONECCT.2014.6740330
27. Goyat Y, Chateau T, Malaterre L, Trassoudaine L (2006) Vehicle trajectories evaluation by static video sensors. In: Intelligent transportation systems conference, pp 864–869
28. Guo GD, Jain AK, Ma WY, Zhang HJ (2002) Learning similarity measure for natural image retrieval with relevance feedback. IEEE Trans Neural Netw 13(4):811–820
29. Hong WD, Lee TH, Chang PC (2007) Real-time foreground segmentation for the moving camera based on H.264 video coding information. In: Future generation communication and networking, vol 1, pp 385–390
30. Ibrahim M, Rao S (2007) Motion analysis in compressed video - a hybrid approach. In: IEEE international workshop on motion and video computing, pp 17–17
31. ISO/IEC JTC1 11172-2: Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video (MPEG-1) (1993)
32. ISO/IEC JTC1 13818-2Generic coding of moving pictures and associated audio information – Part 2: Video (MPEG-2) (1994)
33. ISO/IEC JTC1 14496-2: Coding of audio-visual objects – Part 2: Visual (MPEG-4 visual version 1) (1999)
34. ISO - International Organization for Standardization. http://www.iso.org/iso/home.html
35. ITU Telecommunication Standardization Sector. http://www.itu.int/en/ITU-T/Pages/default.aspx
36. ITU-T: Recommendation H.261, Video Codec for Audiovisual Services at p×64 kbit/s, version 1 (Dec 1990), version 2 (March 1993)
37. Jamrozik M, Hayes M (2002) A compressed domain video object segmentation system. In: International conference on image processing, vol 1, pp 113–116
38. Kapotas S, Skodras A (2010) Moving object detection in the H.264 compressed domain. In: IEEE international conference on imaging systems and techniques, pp 325–328
39. Käs C, Nicolas H (2008) An Approach to trajectory estimation of moving objects in the H.264 compressed domain. In: Proceedings of the 3rd pacific rim symposium on advances in image and video technology, pp 318–329
40. Khatoonabadi S, Bajic I (2013) Video object tracking in the compressed domain using spatio-temporal Markov random fields. IEEE Trans Image Process 22(1):300–313
41. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) HMDB: a large video database for human motion recognition. In: Proceedings of the international conference on computer vision, pp 2556–2563
42. Lie WN, Chen RL (2001) Tracking moving objects in MPEG-compressed videos. In: IEEE international conference on multimedia and expo, pp 965–968
43. Liu Z, Lu Y, Zhang Z (2007) Real-time spatiotemporal segmentation of video objects in the H.264 compressed domain. J Vis Commun Image Represent 18(3):275–290

44. Mak CM, Cham WK (2009) Real-time video object segmentation in H.264 compressed domain. IET Image Process 3(5):272–285
45. Manjunath B, Ohm JR, Vasudevan V, Yamada A (2001) Color and texture descriptors. IEEE Transa Circ Syst Video Technol 11(6):703–715
46. Marpe D, Schwarz H, Wiegand T (2003) Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. IEEE Trans Circ Syst Video Technol 13(7):620–636
47. Mehmood K, Mrak M, Calic J, Kondoz A (2009) Object tracking in surveillance videos using compressed domain features from scalable bit-streams. Signal Process Image Commun 24(10):814–824
48. Mehrabi M, Zargari F, Ghanbari M (2012) Compressed domain content based retrieval using H.264 DC-pictures. MultimedTools Appl 60(2):443–453
49. Mezaris V, Kompatsiaris I, Boulgouris N, Strintzis M (2004) Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval. IEEE Trans Circ Syst Video Technol 14(5):606–621
50. Mezaris V, Kompatsiaris I, Kokkinou E, Strintzis MG (2003) Real-time compressed-domain spatiotemporal video segmentation. IEEE Trans Circ Syst Video Technol 14(5):606–621
51. Mezaris V, Kompatsiaris I, Strintzis MG (2004) Compressed-domain object detection for video understanding. In: Workshop on image analysis for multimedia interactive services (WIAMIS)
52. Mitsumoto S, Yuasa H, Zen H (1998) Moving object detection from MPEG coded picture. In: MVA, pp 422–425
53. Niu C, Liu Y (2010) Moving object segmentation in the H.264 compressed domain. In: Zha H, Taniguchi Ri, Maybank S (eds) Asian conference on computer vision, pp 645–654
54. Ohm J, Sullivan G, Schwarz H, Tan TK, Wiegand T (2012) Comparison of the coding efficiency of video coding standards; including high efficiency video coding (HEVC). IEEE Trans Circ Syst Video Technol 22(12):1669–1684
55. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
56. Ozer B, Wolf W, Akansu A (2000) Human activity detection in MPEG sequences. In: Proceedings workshop on human motion, pp 61–66
57. Ozer I, Wolf W (2002) Real-time posture and activity recognition. In: Workshop on motion and video computing, pp 133–138
58. Pearson K (1901) On lines and planes of closest fit to systems of points in space. Philos Mag 2(6):559–572
59. Pei W, Zhixia W (2010) Moving object segmentation in H.264/AVC compressed domain using ant colony algorithm. In: International conference on signal processing systems (ICSPS), vol 2, pp 716–719
60. Poppe C, Bruyne SD, Paridaens T, Lambert P, de Walle RV (2009) Moving object detection in the H.264/AVC compressed domain for video surveillance applications. J Vis Commun Image Represent 20(6):428–437
61. Porikli F (2004) Real-time video object segmentation for MPEG encoded video sequences. SPIE conference on Real-Time Imaging, vol 5297, pp 195–203
62. Porikli F, Bashir F, Sun H (2010) Compressed domain video object segmentation. IEEE Trans Circ Syst Video Technol 20(1):2–14
63. Qiya Z, Gaobo Y, Weiwei C, Zhaoyang Z (2007) A fast and accurate moving object extraction scheme in the MPEG compressed domain. In: International conference on image and graphics, pp 592–597
64. Rangarajan B, Babu RV (2014) Human action recognition in compressed domain using PBL-McRBFN approach. In: 2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP), pp 1–6. doi:10.1109/ISSNIP.2014.6827622
65. Richardson IEG (2003) H.264 and MPEG-4 video compression: video coding for next-generation multimedia. Wiley
66. Rijkse K (1996) H.263: Video coding for low-bit-rate communication. IEEE Commun Mag 34(12):42–45
67. Rodriguez-Benitez L, Moreno-Garcia J, Castro-Schez J, Albusac J, Jimenez-Linares L (2009) Automatic objects behaviour recognition from compressed video domain. Image Vis Comput 27(6):648–657
68. Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local SVM approach. In: International conference on pattern recognition, pp 32–36
69. Shi YQ, Sun H (2008) Image and video compression for multimedia engineering: fundamentals, algorithms, and standards, 2nd edn. CRC Press, Inc., Boca Raton
70. Solana-Cipres C, Fernandez-Escribano G, Rodriguez-Benitez L, Moreno-Garcia J, Jimenez-Linares L (2009) Real-time moving object segmentation in H.264 compressed domain based on approximate reasoning. Int J Approx Reas 51(1):99–114

71. Soomro K, Zamir AR, Shah M (2012) UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv:abs/1212.0402
72. Sukmarg O, Rao KR (2000) Fast Object Detection and Segmentation in MPEG Compressed Domain. TENCON. Proceedings 3:364–368
73. Sullivan G, Ohm J, Han WJ, Wiegand T (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circ Syst Video Technol 22(12):1649–1668
74. Szczerba K, Forchhammer S, Stttrup-Andersen J, Eybye P (2009) Fast compressed domain motion detection in H.264 video streams for video surveillance applications. In: Proceedings, AVSS, pp 478–483
75. Tan YP, Saur D, Kulkarni S, Ramadge P (2000) Rapid estimation of camera motion from compressed video with application to video annotation. IEEE Trans Circ Syst Video Technol 10(1):133–146
76. The Moving Picture Experts Group website. http://mpeg.chiariglione.org/
77. Thilak V, Creusere CD (2004) Tracking of extended size targets in H.264 compressed video using the probabilistic data association filter. In: EUSIPCO, pp 281–284
78. Tom M, Babu RV (2013) Fast moving-object detection in H.264/AVC compressed domain for video surveillance. In: National conference on computer vision, pattern recognition, image processing and graphics (NCVPRIPG). doi:10.1109/NCVPRIPG.2013.6776202
79. Tom M, Babu RV, Praveen R (2014) Tom M, Babu RV, Praveen R (2014) Compressed domain human action recognition in H.264/AVC video streams. Multimed Tools Appl. doi:10.1007/s11042-014-2083-2
80. Vacavant A, Robinault L, Miguet S, Poppe C, de Walle RV (2011) Adaptive background subtraction in H.264/AVC bitstreams based on macroblock sizes. In: VISAPP, pp 51–58
81. Verstockt S, De Bruyne S, Poppe C, Lambert P, Van De Walle R (2009) Multi-view object localization in H.264/AVC compressed domain. In: IEEE international conference on advanced video and signal based surveillance, pp 370–374
82. Wang FP, Chung WH, Ni GK, Chen IY, Kuo SY (2012) Moving object extraction using compressed domain features of H.264 INTRA frames. In: IEEE international conference on advanced video and signal-based surveillance, pp 258–263
83. Wang H, Chang SF (1997) A highly efficient system for automatic face region detection in MPEG video. IEEE Trans Circ Syst Video Technol 7(4):615–628
84. Wang J, Patel N, Grosky WI, Fotouhi F (2009) Moving camera moving object segmentation in compressed video sequences. Int J Image Graph 9(4):609–627
85. Wang R, Zhang H, Zhang Y (2000) A confidence measure based moving object extraction system built for compressed domain. In: Proceedings of the IEEE international symposium on circuits and systems, p 21–24
86. Wang T, Liang J, Wang X, Wang S (2012) Background modeling using local binary patterns of motion vector. In: IEEE conference on visual communications and image processing, pp 1–5. doi:10.1109/VCIP.2012.6410784
87. Wang W, Yang L, Gao W (2008) Modeling background and segmenting moving objects from compressed video. IEEE Trans Circ Syst Video Technol 18(5):670–681
88. Welcome to the IEC - International Electrotechnical Commission. http://www.iec.ch/
89. Wiegand T, Sullivan G, Bjontegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. IEEE Trans Circ Syst Video Technol 13(7):560–576
90. Yang J, Wang S, Lei Z, Zhao Y, Li S (2012) Spatio-temporal LBP based moving object segmentation in compressed domain. In: IEEE international conference on advanced video and signal-based surveillance (AVSS), pp 252–257
91. Yeo BL, Liu B (1995) Rapid scene analysis on compressed video. IEEE transactions on circuits and systems for video technology 5(6):533–544
92. Yeo C, Ahammad P, Ramchandran K, Sastry S (2008) High-speed action recognition and localization in compressed domain videos. IEEE Trans Circ Syst Video Technol 18(8):1006–1015
93. Yoneyama A, Nakajima Y, Yanagihara H, Sugano M (1999) Moving object detection and identification from MPEG coded data. In: International conference on image processing, vol 2, pp 934–938
94. You W, Sabirin MSH, Kim M (2007) Moving object tracking in H.264/AVC bitstream. In: MCAM, pp 483–492
95. You W, Sabirin MSH, Kim M (2012) Real-time detection and tracking of multiple objects with partial decoding in H.264/AVC bitstream domain. arXiv:abs/1202.4743
96. Yu DL (2003) Video analysis and indexing in compressed domain. Master Of Science Thesis, Institute for Infocomm Research, National University of Singapore
97. Yu X, Xue P, Duan L, Tian Q (2007) An algorithm to estimate mean vehicle speed from MPEG Skycam video. Multimed Tools Appl 34(1):85–105

98. Yu XD, Duan LY, Tian Q (2003) Robust moving video object segmentation in the MPEG compressed domain. In: IEEE international conference on image processing, vol 3. doi:10.1109/ICIP.2003.1247399
99. Zeng W, Du J, Gao W, Huang Q (2005) Robust moving object segmentation on H.264/AVC compressed video using the block-based MRF model. Real-Time Imaging 11(4):290–299
100. Zeng W, Gao W, Zhao D (2003) Automatic moving object extraction in MPEG video. In: Proceedings of the international symposium on circuits and systems, vol 2, pp 524–527

Dr. R. Venkatesh Babu received his Ph.D. degree in electrical engineering from the Indian Institute of Science, Bangalore, India, in 2003. He held post-doctoral positions with the Norwegian University of Science and Technology, Norway, and with IRISA/INRIA, Rennes, France, through ERCIM fellowship. Subsequently, he was a Research Fellow with Nanyang Technological University, Singapore. He spent couple of years working in the industry. He is currently an Assistant Professor and convener of Video Analytics Laboratory at Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore, India. His research interests include video analytics, human-computer interaction, computer vision, and compressed domain video processing. He is a senior member of IEEE.

Manu Tom received his B.Tech. degree in Electronics and Communication Engineering from College of Engineering Thiruvananthapuram, Kerala University. After graduation, he worked in industry for two years. Later, he joined Video Analytics Laboratory, Indian Institute of Science, Bangalore, India. At present, he pursues Masters degree in Electrical Engineering at the RWTH Aachen University, Germany. His research interests include multimedia signal processing, computer vision and medical imaging.

Paras Wadekar received his B. Tech. degree in Electronics and Telecommunication from College of Engineering, Pune in 2011. He finished his M. Tech. from Centre for Electronics Design and Technology, Indian Institute of Science in 2013. He is currently working as a project associate in Video Analytics Lab, SERC, Indian Institute of Science, Bangalore, India.