Classification-Based Adaptive Search Algorithm for Video Motion Estimation

by

Mahdi Asefi

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2006

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Mahdi Asefi

Abstract

A video sequence consists of a series of frames. In order to compress the video for efficient storage and transmission, the temporal redundancy among adjacent frames must be exploited. A frame is selected as reference frame and subsequent frames are predicted from the reference frame using a technique known as motion estimation. Real videos contain a mixture of motions with slow and fast contents. Among block matching motion estimation algorithms, the full search algorithm is known for its superiority in the performance over other matching techniques. However, this method is computationally very extensive. Several fast block matching algorithms (FBMAs) have been proposed in the literature with the aim to reduce computational costs while maintaining desired quality performance, but all these methods are considered to be sub-optimal. No fixed fast block matching algorithm can efficiently remove temporal redundancy of video sequences with wide motion contents. Adaptive fast block matching algorithm, called classification based adaptive search (CBAS) has been proposed. A Bayes classifier is applied to classify the motions into slow and fast categories. Accordingly, appropriate search strategy is applied for each class. The algorithm switches between different search patterns according to the content of motions within video frames. The proposed technique outperforms conventional stand-alone fast block matching methods in terms of both peak signal to noise ratio (PSNR) and computational complexity. In addition, a new hierarchical method for detecting and classifying shot boundaries in video sequences is proposed which is based on information theoretic classification (ITC). ITC relies on likelihood of class label transmission of a data point to the data points in its vicinity. ITC focuses on maximizing the global transmission of true class labels and classify the frames into classes of cuts and non-cuts. Applying the same rule, the non-cut frames are also classified into two categories of arbitrary shot frames and

gradual transition frames. CBAS is applied on the proposed shot detection method to handle camera or object motions. Experimental evidence demonstrates that our method can detect shot breaks with high accuracy.

Acknowledgments

First and foremost, I would like to express my gratitude and appreciation to my supervisors, Prof. Sherman Shen and Dr. Mohamed-Yahia Dabbagh, for encouragement during my studies, careful reading and correction of my thesis and provision of constructive remarks.

My deepest gratitude, love and affection belong to my parents for their constant love and support. I owe all that I am and all I have ever accomplished.

to my parents

and

my lovely brother
Mahan

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Video has huge redundant information which must be exploited to be stored and transmitted efficiently. The common technique to achieve this goal is known as *motion estimation*. In this technique, the current frame is predicted from a previous frame known as reference frame by using *motion vectors*. With the increasing demand of multimedia applications, considerable efforts are needed for efficient video compressing and encoding algorithms. Motion estimation has proven to be an effective technique for exploiting the temporal redundancy in video sequences and is therefore an essential part of MPEG and H.263 compression standards. Since motion estimation is the most computationally intensive portion of video encoding, efficient fast motion estimation algorithms are highly desired for video compressors subject to diverse requirement on bit rate, video sequence characteristics and delay. Knowledge of the motion is not available from a video data and must be deduced using computationally intensive algorithms. For efficient handling of motions with variety of contents, the need for adaptive motion estimation methods is inevitable.

Generally, the optimal full search (FS) block matching algorithm results in the

best performance with respect to the quality of decoded video sequences, however, it is computationally very intensive. Due to the huge demand of the computational requirement several fast search algorithms have been developed and introduced in recent years including the three step search (TSS), the new three step search (NTSS), the four step search (FSS), the block based gradient descent search (BBGDS), the diamond search(DS), and the hexagon-based search (HEXBS). Since videos exist with variety of contents, no stand-alone fast block matching algorithm (FBMA) can efficiently remove the redundant data.

In this thesis, a new adaptive method based on bayesian classification technique is proposed. This method classifies predicted motion of each block within each image frame either in slow (C_{slow}) or fast category (C_{fast}). In order to apply bayesian classifier, conditional probability distribution functions (PDFs), $P(x|C_{slow})$ and $P(x|C_{fast})$, are estimated where x is the length of blocks motion vector. Parzen window method with gaussian kernel is used to estimate required PDFs. After estimating the motion class, appropriate search pattern is employed to find the best matching block within the frame.

Our main contribution has the following aspects, namely: 1) Applying Bayes Classifier and nonparametric Parzen window probability density function estimation of length of the motion vectors contained in each frame for development of an adaptive video motion estimation algorithm. However, to the best of our knowledge, there is no adaptive motion estimation method which applies Bayes classifier for classifying the content of motions in video frames and accordingly developing adaptive motion estimation algorithm. 2) The simulation results reveal that the proposed algorithm is able to maintain high and constant quality of performance in terms of peak signal to noise ratio (PSNR) and computational complexity. It outperforms conventional stand-alone fast block matching algorithms and shows

competitive results compared to other adaptive motion estimation algorithms.

In addition to compensation of motions in video encoder, motion estimation has important role in other video processing algorithms. For example, video shot detection algorithms can benefit from motion compensation to make themselves more robust for handling camera and object motions. As a study case, we apply our adaptive motion estimation method on a hierarchical classification based method for video shot detection. This application shows an improved performance of the shot detection, as illustrated by several simulations.

1.1 Thesis Organization

Chapter 2 provides some background on video motion compensation. It introduces the problem of video motion estimation for general problems in computer vision and video compression. This is followed by discussion on different fast block matching motion estimation methods and their advantages and drawbacks.

Chapter 3 presents a classification-based approach for designing adaptive video motion estimation. It starts by a review of recent proposed adaptive methods. A set of preliminary concepts in pattern classification are introduced. This is followed by a detailed discussion on design criteria and reasoning for selection of specific classifier and search schemes for the proposed motion estimation algorithm.

Chapter 4 presents detailed experimental results of the proposed algorithm in comparison to standard algorithms on various videos with different characteristics. We discuss the results with respect to the observations from experiments.

Chapter 5 presents a new hieratical classification based video shot segmentation algorithm. The temporal segmentation problem is transformed to multi-class cate-

gorization problem. We applied classification based adaptive motion compensation method, introduced in chapter 3, to cope with camera and object movements. The proposed shot detection method is based on information theoretic classification (ITC) rule. K-mean clustering is applied to cluster different types of gradual shot transitions. Finally, experimental results on different videos are provided to illustrate the performance of the proposed algorithm in terms of precision and recall.

Chapter 6 concludes the thesis and suggests some future research directions.

Chapter 2

Background on Video Motion Compensation

2.1 Motion Estimation

Image compression techniques rely on two principles: reduction of statistical redundancies in data and characteristics of human visual perception [1]. In video coding framework, statistical redundancies are grouped into spatial and temporal categories. Compression techniques which reduce temporal redundancies, are referred to as *interframe* techniques while those reducing spatial redundancies are referred to as *intraframe* techniques. Motion estimation (ME) algorithms have been applied for the reduction of temporal redundancies [1, 2, 3].

ME algorithms are originally developed for applications such as computer vision, image sequence analysis and video coding [1]. They can be categorized in the following main groups: gradient techniques [4, 5], pel-recursive techniques [9, 10], block matching techniques [19]–[31], and frequency-domain techniques [5]. From the

perspective of video coding, ME methods are used to reduce bandwidth corresponding to motion difference information and motion overhead. These requirements can be in contradiction to each other since on the one hand, motion estimation algorithms should provide suitable prediction information, while on the other hand, they should have low overhead information.

We first introduce the notation used in the following sections. Let $I(x, y, t)$ be the image intensity at time instant t at location $r = (x, y)$ and $d = (d_x, d_y)$ is displacement during time interval Δt. All techniques rely on the assumption that change in image intensity is only due to the displacement d [4], i.e.

$$I(r, t) = I(r - d, t - \Delta t) \tag{2.1}$$

1. Gradient Techniques

   The first assumption in gradient techniques is that image luminance is invariant during motions. Taylor's series expansion of right hand side of (2.1) would give

$$I(r-d, t-\Delta t) = I(r, t) - d \cdot \nabla I(r, t) - \Delta t \frac{\partial I(r, t)}{\partial t} + \text{higher order terms} \tag{2.2}$$

   where $\nabla = [(\partial/\partial x), (\partial/\partial y)]$ is the gradient operator and by assuming $\Delta t \to 0$, neglecting higher order terms, and defining the motion vector as $v = (v_x, v_y) = d/\Delta t$ we obtain [4]

$$v \cdot \nabla I(r, t) + \frac{\partial I(r, t)}{\partial t} = 0 \tag{2.3}$$

which is known as *spatio temporal constraint*. Since the motion vector has two components, the motion field can be solved only by introducing an additional constraint. Additional constraint known as *smoothing constraint* is introduced in [4] that minimizes optical flow gradient magnitude. The motion field is obtained by minimizing the following error term defined as [1]

$$\int\int\{(v\cdot\nabla I+\frac{\partial I}{\partial t})^2+ \alpha^2\left[\left(\frac{\partial v_x}{\partial x}\right)^2+\left(\frac{\partial v_x}{\partial y}\right)^2+\left(\frac{\partial v_y}{\partial x}\right)^2+\left(\frac{\partial v_y}{\partial y}\right)^2\right]\} \tag{2.4}$$

where α^2 is a minimization factor. This optimization problem can be solved by variational calculus. Many variations of the above algorithm are proposed in literature [6, 7, 8]. From coding perspective, these motion estimation methods suffer from two main drawbacks. First, the prediction error has high energy due to smoothness constraint, and second, the motion field requires high motion overhead.

2. Pel-Recursive Techniques

These methods rely on recursive reduction of predictive error or DFD defined in (2.5). The displacement frame difference (DFD) or frame dissimilarity measure is denoted by

$$DFD(r,t,d)=I(r,t)-I(r-d,t-\Delta t) \tag{2.5}$$

These methods are among the very first algorithms designed for video coding

with the goal of having low hardware complexity. The first pel-recursive algorithm was proposed in [2] , which minimizes DFD^2 by applying steepest descent technique. The displacement d at $(k+1)$ iteration is given by [1]:

$$d^{(k+1)} = d^{(k)} - \frac{\epsilon}{2}\nabla_d DFD^2(r, t, d^{(k)}) \tag{2.6}$$

where ϵ is a constant gain, k is the iteration index, and ∇_d is the gradient vector with respect to the displacement d. Substituting DFD, i.e. (2.5), in above formula, we obtain

$$\nabla_d DFD^2(r, t, d^{(k)}) = 2DFD(r, t, d^{(k)}) \cdot \nabla_d I(r - d^{(k)}, t - \Delta t) \tag{2.7}$$

By substituting (2.7) into (2.6), the displacement field update is obtained as follows

$$d^{(k+1)} = d^{(k)} - \epsilon DFD(r, t, d^{(k)}) \cdot \nabla_d I(r - d^{(k)}, t - \Delta t) \tag{2.8}$$

The performance of pel-recursive algorithms strongly depends on the way for computing update term in (2.8). Various research works in literature have focused on proposing efficient methods for computation of above formula [9, 10]. Casuality constraints reduces the predictive capability of these algorithms comparing to non-casual methods. High computational complexity is another drawback of pel-recursive algorithms. Furthermore, the error function to be minimized has generally many local minima. These algorithms are also very sensitive to noise and large displacements and discontinuities in the motion field which can not be efficiently handled.

3. Block Matching Techniques

Block matching is widely used for stereo vision, vision tracking, and video compression. Video coding standards such as MPEG-1, MPEG-2, MPEG-4, H.261, H.263 and H.264 use block based motion estimation algorithms due to their effectiveness and simplicity for hardware implementation. The main idea behind block matching estimation is the partitioning of the target (predicted) frame into square blocks of pixels and finding the best match for these blocks in a current (anchor) frame. To find the best match, a search inside a previously coded frame is performed and the matching criterion is utilized on the candidate matching blocks. The displacement between the block in the predictor frame and the best match in the anchor frame defines a motion vector. In the encoder, it is only necessary to send the motion vector and a residue block, defined as the difference between the current block and the predictor block.

The matching criterion is typically the *mean of absolute errors* (MAE) or the *mean of square errors*(MSE), given respectively by:

$$MAE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \tag{2.9}$$

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \tag{2.10}$$

where $N \times N$ is the size of each block, C_{ij} and R_{ij} are respectively the pixel values in the current block and the reference block. *Peak signal to noise ratio* (PSNR) characterizes the motion compensated image created by predicted

motion vectors and blocks from the reference frame.

$$\begin{aligned} PSNR &= 10\log_{10}\left[\frac{\left(\text{peak to peak value of the original signal}\right)^2}{\text{MSE}}\right] \\ &= 10\log_{10}\left[\frac{255\times 255}{\text{MSE}}\right] \end{aligned} \tag{2.11}$$

Block matching algorithms have been originally designed for prediction of displacements with one pixel accuracy, however, it is possible to achieve sub-pixel accuracy by interpolating the image intensity at factorial pixel locations. This is done in practice by means of post processing after one pixel accuracy motion vectors have been computed. However, the post processing increases computational complexity. To reduce the computational complexity and to consider multi scale characteristic of motion in a scene, hierarchical methods [11, 12] have been proposed.

In standard block matching algorithms, motions were restricted to translational displacements. However, block matching algorithms that investigate affine translations have been investigated to reduce this limitation [12]. In [13], a generalized block matching algorithm is proposed by considering complex motion models such as rotation and nonlinear deformation.

In spite of their intensive applications, block matching algorithms also have some serious drawbacks such as unreliable motion fields in the scene of true motion, block artifacts, and poor motion prediction along block edges.

2.1.1 Comparison of Motion Estimation Techniques

Experiments haven been performed in [1] to asses the performance of several motion estimation techniques. Best known algorithms are selected for comparison, namely

Horn-Schunck gradient technique [4], and Netravali-Robbins pel-recursive technique [2], and full-search block matching technique. In general smooth motion fields are more desired in coding in order to prevent artificial discontinuity. It should be emphasized that performance of pel-recursive algorithm is highly dependent on the way the recursive term is computed. Results clearly show that the pel-recursive algorithm is significantly less efficient than the other two techniques especially when dealing with large displacements and discontinuities. As gradient methods provide more dense motion field compared to block matching technique (one vector per pixel compared to one vector per block), the gradient method is expected to outperform block matching technique. However, the performance results are very similar to each other. Hence, the Horn-Schunck gradient technique does not result in enhanced capability, while it highly increases the overhead information. The method is more interesting from analysis point of view rather than application in coding.

The block matching technique relies on simple motion model which leads to precise motion estimation with low overhead information. Therefore, it achieves appropriate allocation of bandwidth between DFD and motion parameters. Due to this considerations, block matching methods are the most widely used techniques in video coding applications.

2.2 Fast Block Matching Algorithms

In the following sections, an overview of several standard fast block matching motion estimation algorithms (FBMAs) is presented. Among them, the diamond search (DS) algorithm is accepted by MPEG-4 verification model [30], and it is considered the state-of-the-art search scheme.

2.2.1 Full Search

The best predicted representative of the current block is searched by computing the matching criterion between the current block and all blocks in the search area. If the algorithm applies MSE as matching criterion, then for checking each point with block size of 16×16, it requires 256 subtractions, 256 multiplications and 255 additions to calculate the MSE. The size of the search area is given by

$$\text{Search Area} = (2p + 1) \times (2p + 1) \tag{2.12}$$

where p is the search parameter . The illustration of search area is shown in Figure 2.1 When $p = 7$, the size of the search area will be 225 and hence 225 points must be checked in full search (FS) algorithm which is very intensive from computational standpoint. Given a frame of size $M \times M$ and a block size of $N \times N$, the number of operations for each block is

$$N_b = (2p + 1)^2 N^2 \tag{2.13}$$

and the number of operations for each frame is

$$N_f = (2p + 1)^2 M^2 \tag{2.14}$$

For $N = 16$, $p = 7$ and $M = 256$, we have

$$N_b = 57600 \tag{2.15}$$

$$N_f = 14745600 \tag{2.16}$$

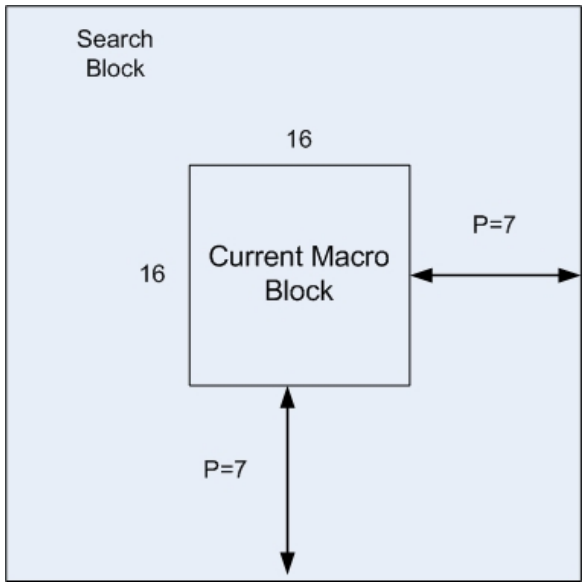

Figure 2.1: Block Matching a macro block of size 16×16 pixels and a search parameter p of size 7 pixels.

which illustrates that FS is computationally very intensive.

In [19], an alternative one dimensional full search (1DFS) algorithm for 2-D full search has been proposed. The 1DFS is hardware-oriented algorithm. Instead of direct searching for 2-D motion vector, 1FDS first utilizes full search method to find the location with minimum distortion along horizontal direction. The algorithm then tries to find the minimum distortion along the vertical direction.

2.2.2 Three Step Search

Three step search (TSS) starts with a search location at the center of the search area and searches in search window with sides of 4 for a usual search area ($p = 7$) as shown in Figure 2.2. Nine points are checked including one point at the center and eight points on the borders of the search window. The search center is moved next to the place of the best match found. In the second step, it searches in the search

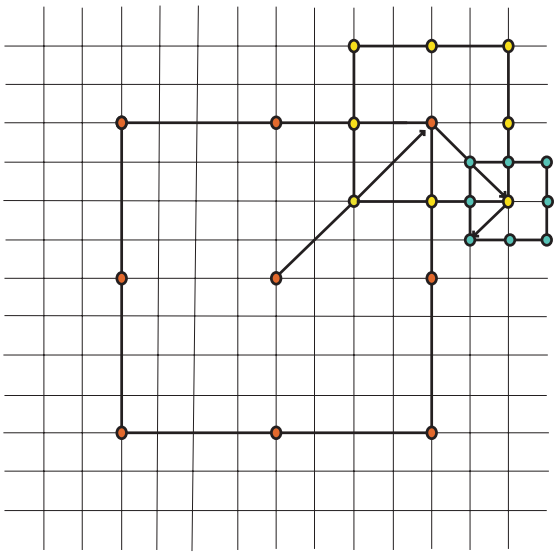

Figure 2.2: Three step search

window with sides equal to one half of the sides of the window in the first step. Eight points on the borders of the search window are checked. In the third step, the window with sides equal to one half of the size of the second step is considered around the best matched point found in pervious step and again eight points on the border of the window will be checked to find the best match. Consequently, the total number of the checking points for TSS [20] is 25.

2.2.3 New Three Step Search

The new three step search (NTSS) algorithm [21] utilizes additional checking points and two half stop conditions to improve the performance of TSS. In the first step, additional eight neighbors of the center are checked. If the best match is found on this small window, then additional three or five points are checked and the algorithm will stop. This is the second stop condition as shown in Figure 2.3. If

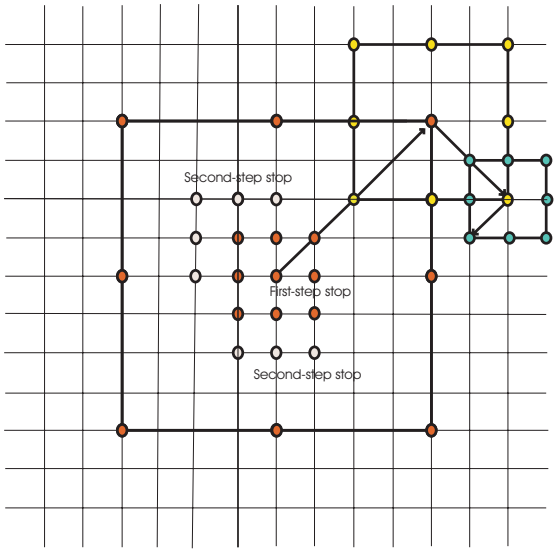

Figure 2.3: New three step search

the best matching point in the first step found on the boundary of the outside window (similar to three TSS), then the second and third steps of the algorithm are the same as those of TSS. The main difference between TSS and NTSS is that TSS utilized uniformly distributed search points in its first step. NTSS employs center based checking pattern in first step and half way stop technique is applied to reduce computational costs. Compared to TSS, NTSS is much more robust and produces smaller compensation error.

2.2.4 Four Step Search

Four step search (FSS) employs the center biased property of the motion vectors (MVs) similar to NTSS. First, the search center is located at MV $(0, 0)$ and the search step size is set to 2 as shown in Figure 2.4. Nine points are checked in the search window. If the best match occurs at the center of the widow, the neighbor

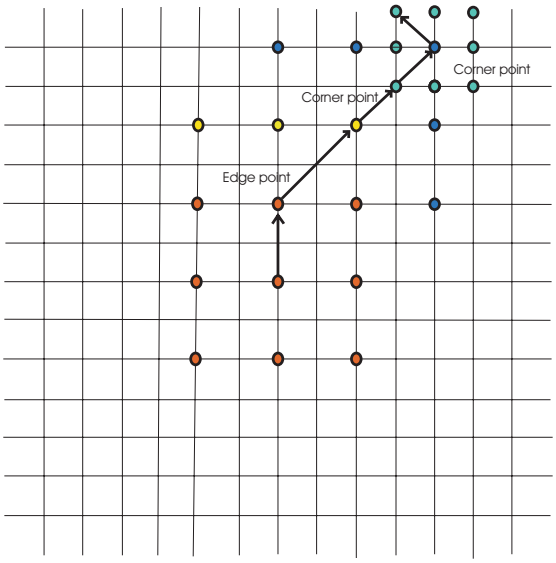

Figure 2.4: Four step search

search window with step size reduced to one with eight checking points on the sides will be checked and the best match is the best predicted motion vector. If the best match in the first step occurs on the edges or corners of the search window, additional three or five points will be checked in the second step, respectively. If the current minimum occurs on the center of the search window, the step size will reduce to one. the algorithm stops while all the neighboring points are checked. With respect to the computations, the best case requires 17 checking points and the worst case requires 32 checking points [22]. Simulation results have shown that FSS generally performs better than TSS in terms of motion compensation error. Compared to NTSS, FSS reduces worst case computations from 33 to 27 search points and average computations from 21 to 19 search points.

2.2.5 Diamond Search

This algorithm is Based on the observation that 50% to 80% of the MVs are located in the circular area of radius 2 and centered on the position of zero motion vector [23]. The diamond search (DS) is introduced and accepted by MPEG-4 verification model. As shown in Figure 2.5, the algorithm employs two search patterns: small diamond search pattern (SDSP) and large diamond search pattern (LDSP). The LDSP pattern will be employed repeatedly until the best match occurs at the center of the LDSP. Based on the location of the best match in each step, the additional three or five points will be checked respectively if the minimum occurs on the sides or corners of the diamond. Next, the algorithm starts searching SDSP pattern centered at the bast matched point found in pervious LDSP search. Simulation results show that DS outperforms TSS and has close performance to NTSS from compensation error point of view while reducing computational cost by approximately 20 to 25 percent.

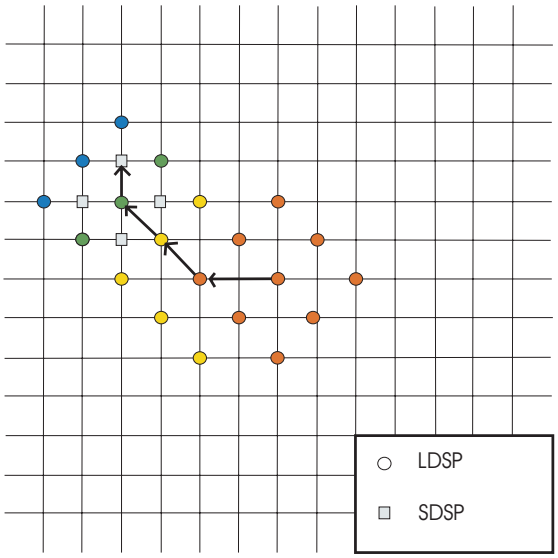

Figure 2.5: Diamond search

2.2.6 Cross Search

The basic idea is to use logarithmic step search with some alternations which results in fewer search point computations [26]. The main difference between cross search (CR) and other methods that use logarithmic step search is that at each iteration 4 search points which are end of a cross (X). At the final stage, the search points can be either the end points of (X) or (+) crosses. For maximum motion displacement of w pel/frame, this algorithm only needs $(5 + 4log_2 w)$ computations to find the best match. For maximum motion displacement of 8 pel/frame, this algorithm is 0.25 to 0.27 bit/pel inferior to the full search method, however, its computational complexity is lower by 17 times.

2.2.7 Block-Based Gradient Descent Search

Square blocks of size 3×3 are checked in this method [27]. The search starts by initializing the checking block so that its center point is at origin. The dissimilarity measure is computed for all 9 points within the block. If the minimum occurs at center then search terminates, otherwise updates the checking block so that its center is the winning pixel. Block-based gradient descent search (BBGDS) moves the search in the direction of optimal gradient descent and this is the direction where we expect the dissimilarity measure acquires its minimum value. Mean square error (MSE) and computational complexity performance of several block matching algorithms are illustrated in Table 2.1 and Table 2.2.

FBMA	Foreman	Salesman	Miss-America	Car Phone
FS	26.82	6.52	4.99	21.17
BBGDS	28.80	6.66	5.05	22.34
TSS	34.23	6.97	5.50	24.80
NTSS	28.63	6.62	5.04	22.11

Table 2.1: MSE performance for BBGDS compared to some other FBMAs with search range ± 7 pixels for different video sequences [27]

FBMA	Foreman	Salesman	Miss-America	Car Phone	average complexity
FS	100	100	100	100	100
BBGDS	5.4	4.22	5.08	5.40	5.03
TSS	11.43	11.36	11.39	11.40	11.40
NTSS	3.75	2.96	3.71	3.75	3.54

Table 2.2: Computational complexity performance (%) for BBGDS compared to other FBMAs with search range ± 7 pixels for different video sequences [27]

Although FS introduces best MSE performance, it is computationally very intensive. For sequences with wide variety of motions like *foreman*, condensed patterns such as BBGDS or NTSS show lower MSE performance. In sequences with small motions, these algorithms, have close performance to FS. In general, fast block matching algorithms, are based on the assumption that motion estimation matching error decreases monotonically as the search moves toward the point of global minimum distortion [30]. The uni-modal error surface with global minimum error point is shown in Figure 2.6. The optimum motion vector is searched among the points in a fixed search pattern.

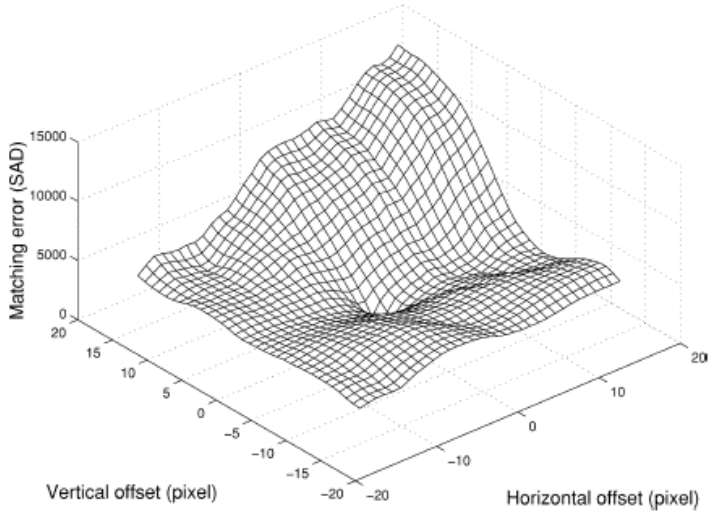

Figure 2.6: Uni-modal error surface with global minimum error point [30]

2.3 Computation Reduction

Motion estimation is one of the most important parts for computational complexity reduction in video encoders. Computation reduction techniques for low bit rate video coders have been developed [15, 16]. In addition to motion estimation, video encoder requires Discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT), motion compensation and entropy encoder. Many fast algorithms for computation of DCT and IDCT have been emerged. In [15], by jointly considering DCT and quantization, the computational reduction techniques for DCT have been proposed with prediction of zero block [17, 18]. All zero and zero motion detection are designed to terminate unnecessary search points in motion estimation algorithms. The average search points and PSNR of different FBMAs in comparison to the case where all-zero detection is applied on the algorithms are provided in Table 2.3 and Table 2.4.

PPMB	Jrene	Miss-am	Akiyo	Claire	Carphone
FS	787.88	787.88	787.88	787.88	787.88
FSZ*	282.72	196.82	355.82	255.81	448.35
TMN8	17.07	18.25	13.91	18.78	21.11
TMN8Z	7.89	5.10	6.05	5.05	13.36
TSS	28.91	28.91	28.89	28.90	29.16
TSSZ	11.25	9.33	13.24	10.15	18.77
DS	13.98	13.18	12.14	12.50	16.42
DSZ	6.72	4.79	5.97	4.83	10.88
BBGDS	14.93	13.92	12.55	12.92	17.86
BBGDSZ	7.41	5.26	6.03	4.99	12.31
DS1	7.01	6.10	5.01	5.29	9.27
DS1Z	4.37	3.00	2.80	2.63	7.29
$\star$ Z denotes the algorithm using all-zero block detection					

Table 2.3: Average search points for various search algorithms at 9600 bit/sec [17].

PPMB	Jrene	Miss-am	Akiyo	Claire	Carphone
FS	29.13	35.50	33.33	34.24	27.88
FSZ*	29.05	35.06	33.39	34.23	27.73
TMN8	29.18	35.56	33.32	34.22	27.78
TMN8Z	29.06	35.25	33.27	34.22	27.68
TSS	29.07	35.09	33.31	34.16	27.75
TSSZ	29.07	34.81	33.33	34.15	27.65
DS	29.17	35.60	33.31	34.28	27.80
DSZ	29.06	34.72	33.28	34.17	27.66
BBGDS	29.12	35.44	33.32	34.28	27.81
BBGDSZ	29.10	34.83	33.32	34.18	27.77
DS1	29.15	35.44	33.32	34.24	27.77
DS1Z	29.09	34.76	33.32	34.15	27.67
$\star$ Z denotes the algorithm using all-zero block detection					

Table 2.4: Average PSNR achieved for various search algorithms at 9600 bit/sec [17].

2.4 Motion Estimation for MPEG-4

In the standardized video coding schemes [14], the distortion criteria is computed for all pels regardless of being in foreground or background. This causes the resulted motion vector not truly reflecting the movement of object pels. In MPEG-4, the object shape description is called α plane. This α plane of a video object can be represented by semi-automatic segmentation of video sequences. The α plane refers to the pel of the current video object at time instance k and contains information that the pixels form the object ($\alpha > 0$) and which of the pixels are not inside the object. The binary α plane is restricted to 0 and 1. At the encoder the shape information helps reduce the ME error by restricting the error to pixels inside the object.

In VLSI design of MPEG-4 encoder, a suitable block matching algorithm related to particular application must be employed. Silicon area, I/O requirement, image

quality arrays structure and effect of different sizes of the on-chip memory are among the most important parameters. VLSI architecture for the high speed full search ME is shown is Figure 2.7.

Relative performance in chip area and I/O bandwidth between various algorithms are strongly dependent on picture size and search range [28]. For small pictures and slow motion (small search range), all BMAs are almost equivalent. However, for larger picture sizes (CCIR-601) and fast motion, certain fast search algorithms have the advantage of a significantly smaller chip area. For a specific algorithm, a designer may alter the implementation due to economical considerations. Comprehensive study on estimating the complexity of various motion estimation algorithms, their chip area, data bandwidth, and image quality has been provided in [28].

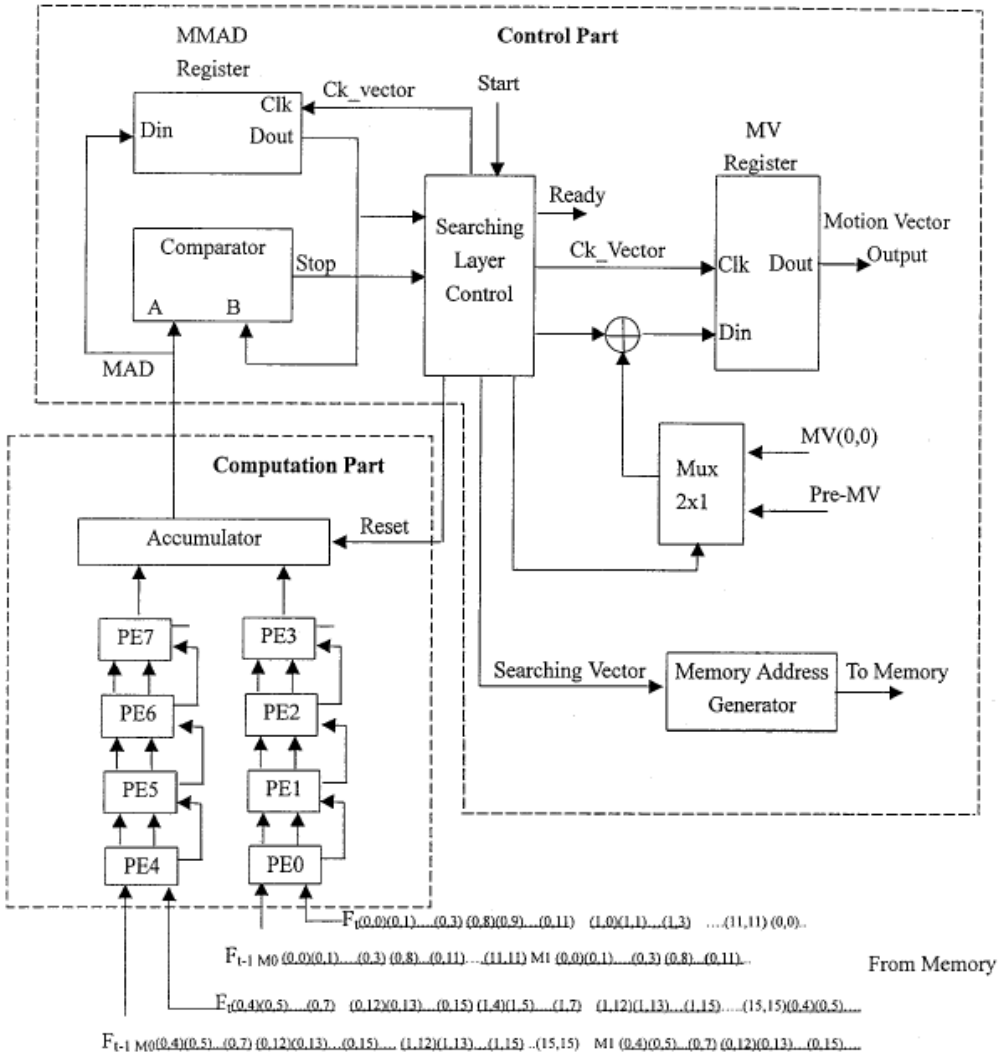

Figure 2.7: VLSI architecture for the high speed full search ME [28]

Chapter 3

Classification-Based Adaptive Motion Estimation

3.1 Related Work

A novel and simple fast block matching algorithm, called adaptive rood pattern search (ARPS) consists of two sequential search steps [30]. The initial search is utilized to locate a good starting point. In this way, the chance for being trapped by local minima is highly reduced and unnecessary intermediate search points can be skipped. For initial search, as shown by Figure 3.1, rood pattern has been utilized while size of the rood is dependent on the motion vectors of neighbor blocks which are called region of support (ROS).

The speed and accuracy of the rood pattern based search algorithm is highly related to the size of the pattern. First step of the proposed method permits the algorithm to adapt itself to the content of motion. In most cases, adjacent blocks belong to the same moving object have similar motions. Therefore, it is reasonable

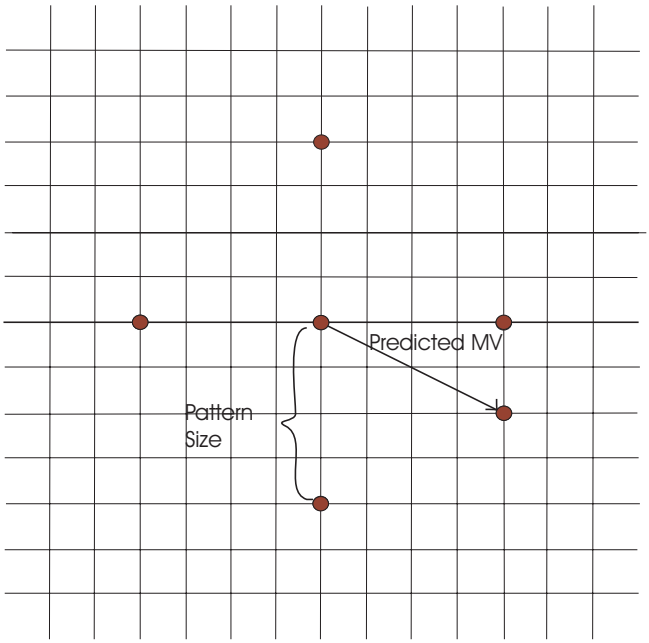

Figure 3.1: Adaptive rood pattern search

to predict the motion of current blocks from motion vectors of adjacent blocks. In order to obtain accurate prediction for MV of current block, the choice of ROS is of importance. In spatial domain, since blocks of each video frame are processed in a raster-scan order, the candidate blocks for prediction of motion in current block are immediate left, above, above left and above right to the current block as shown in Figure 3.2. Calculating the statistical average of MVs in the ROS is a common approach for prediction of motion vector in current block. In [30], the median criterion has been tested in addition to the mean. Experimental results show that four possible choice of ROS shown in Figure 3.2 and two types of prediction criteria, the mean and median, fairly yield similar results in terms of PSNR. Therefore, the simplest ROS, i.e. immediate left block has been adapted in ARPS and since only one block is used for prediction there would be no need to utilize any prediction criteria which also reduces hardware complexity.

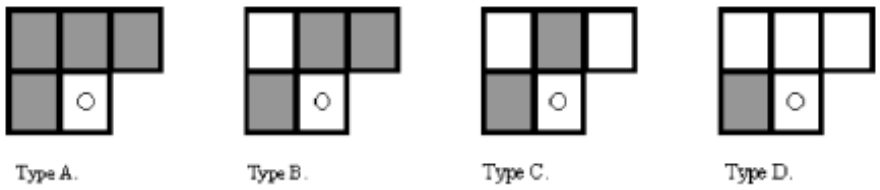

Figure 3.2: Four types of ROS indicated by shaded blocks, the block with ∘ inside is current block [30]

The shape of adaptive rood in initial search is symmetrical, with four points at the end points of the rood. The choice of rood shape, is based on observation of motion feature in real-world sequences. It has been noticed that most of the motions occur in horizontal and vertical directions, since most of the camera movements are in these directions. Finally, ARPS symmetry in shape not only benefits hardware implementation, but also increases robustness.

The adaptive rood search pattern leads to the new search center directly to the most promising area which is around the global minimum. Hence, instead of performing full search, a compact and small search pattern can be utilized to locate the global minimum. When a minimum point is located, this point can be the center for next iteration until the minimum occurs at the center of search pattern.

The number of static motion blocks per frame could be as high as 70% for most video sequences . Therefore, zero-motion prejudgement [15] can be also employed to reduce the computations. Zero motion pre-judgment reduces the the number of searches by predicting that if the block motion in next frame is zero and therefore it can skip the search for that block.

Another fast block matching algorithm [29] has been proposed recently which switches between different search patterns according to the content of data. The

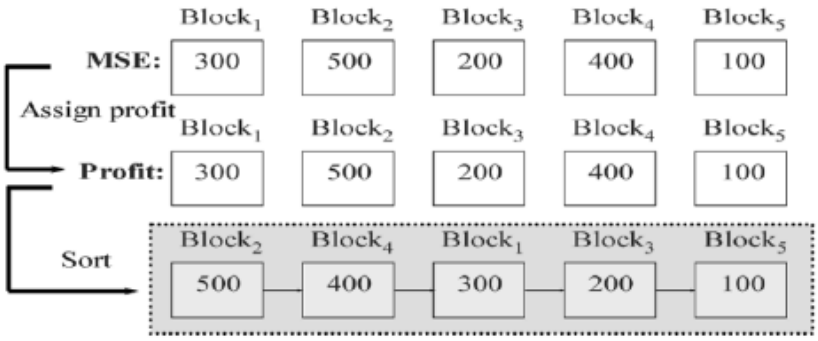

Figure 3.3: Example to illustrate predicted profit list [29]

algorithm uses a predicted profit list to switch between the search patterns. Based on this concept, it proposes an adaptive ME with 3 zones: initial zone, adaptive zone and cleanup zone. We first describe the idea of predicted profit list: The estimation of MV is a searching procedure that locate the point with minimum distortion. The initial point is typically selected at the beginning and its MSE is taken as initial value. Then, the searching algorithm aims at minimizing the MSE as much as possible by considering another candidate block. This improvement in reducing the distortion is referred to as profit. As example to illustrate the predicted profit list is shown in Figure 3.3. The cumulative distribution of profits in third frame of football sequence is shown in Figure 3.4. The so-called predicted profit list is a stored list of these blocks in descending order. The acquired profit list has some characteristics. First, the distribution of profits is not uniform. Second, the blocks in the predicted profit list usually include several various motion contents which no stand-alone FBMA can solve them perfectly. Third, the MVs in neighbor blocks are highly correlated. Fourth, the MV is very likely to be zero near the end of the list. The size of the MV of the arbitrary frame (34th frame) of football sequence is shown in Figure 3.5

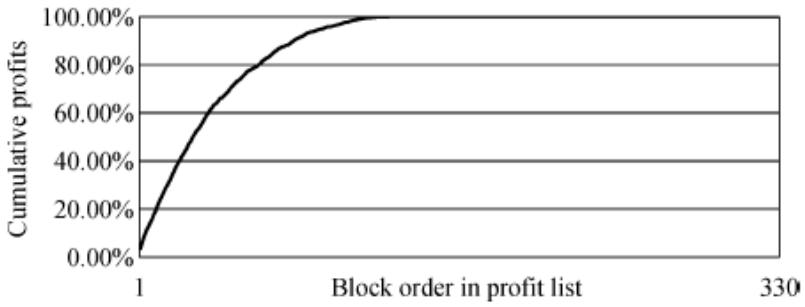

Figure 3.4: Cumulative distribution of profits in the third frame of football sequence [29]

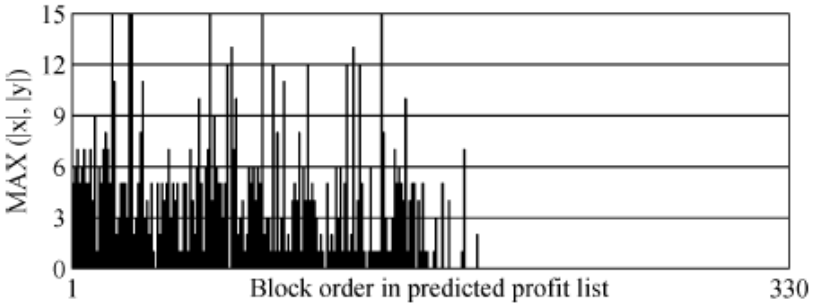

Figure 3.5: MV of 34th frame in football sequence [29]

Based on above observations, the adaptive TSS/DS/BBGDS algorithm or in short (A-TDB) algorithm has been proposed [29]. The A-TDB algorithm adaptively utilizes fast block matching techniques among TSS, DS and BBGDS respectively for slow, moderate and fast motions. After the predicted profit list is created, candidate blocks corresponding to the selected FBMAs can be determined. The list can be divided into three zones. The top of the list is initial zone. This zone is critical to final MSE results. The second and third characteristics of the predicted profit list imply that it may be assembled by multiple successive motion content. Hence, adaptively three fast block matching motion estimation algorithms are employed in adaptive zone, namely TSS, DS and BBGDS. The cleanup zone is refereing to the forth characteristic of the list. The blocks in cleanup zone must be skipped to reduce computations. The simulation results show that A-TDB significantly outperforms single FBMAs.

Adaptivity for motion estimation have been studied from other perspectives. In H.264 encoder, the most time consuming part is variable block size. To reduce this complexity, an early termination method has been proposed [31]. In this technique, the best motion vector is predicted by examining only one search point and some of the search points can be skipped early.

The technique is composed of seven inter prediction modes with different block sizes varying from 16×16 to 4×4. Zero motion detection (ZMD) is applied to skip unnecessary search points. In ZMD, if distortion or a cost function, J, of a block is less than a predefined threshold, the block can be regarded as zero motion block (ZMB). An example of a cost function is defined in [31]. The threshold THZ_i for $i = 1, 2, ...7$ for seven modes are defined. During ME, the $MV(0, 0)$ is first examined. If the cost function satisfies

$$J_i < THZ_i \quad for\ i = 1, 2...7 \tag{3.1}$$

then $MV(0,0)$ is the best prediction and the remaining blocks can be skipped. It is obvious that if we choose larger threshold, more ZMBs are detected which could result in quality loss. Therefore, there is a trade off between quality of video frames and computational complexity. In practice, improving the video quality is more important than minor increase in computations. Detection accuracy is selected as a guide for identifying the thresholds. Further research directions may include making threshold adaptive to quantization level and motion level and employing early termination methods for sub-pixel ME [31].

Several adaptive motion estimation methods have been also proposed for MPEG-4 standard. An algorithm is proposed in [32] which first estimates initial motion vector using motion vector information in the previous frame. Based on SAE of initial MV, an appropriate motion estimation scheme is selected. A novel ME algorithm called adaptive motion estimation algorithm is proposed in [33] based on statistical sum of absolute differences (AMESSAD). The algorithm adaptively finds motion search widow size based on short-term and long-term statistical distribution of motions.

Finally, based on evolution strategies (ESs) with correlated mutations, an adaptively correlated ES motion estimation (ACESME) is proposed in [34]. In this algorithm, the $(\mu, \lambda) - ES$ algorithm with correlated mutations is adopted to block motion estimation.

3.2 Preliminaries

3.2.1 Image and Video Features

A feature is defined as a descriptive parameter in image or video which can be used for processing tasks such as classification, segmentation, etc. Features in visual data can be basically divided into the following categories [42, 43].

1. Statistical features: These are extracted from video without concern about content and driven from algorithms such as camera motion flow, video structure, image difference or scene change.

2. Compressed domain features: A feature that is extracted from a compressed image or video without regard to the content of visual data.

3. Content-based features: A feature that is extracted for purpose of describing the actual content of data.

Image difference measures amount of similarity between pair of images. There are different fundamental image difference methods. Absolute difference, color histogram difference, difference in information theoretic frame work can be cited among other methods.

Patterns are random vectors in an n dimensional space usually called feature space. To classify an object, we make measurements and then extract features which are desired to reflect the defining attributes [39, 40]. Given a set of features, we design a classifier based on distance or probability measures of similarity or discriminant functions.

3.2.2 The Bayesian Classifier

For development of classifiers, we have to consider two main aspects: The basic assumptions that a classifier makes about the data and the optimization procedure to fit the model to the sample data. It is possible to design a very complex classifier, but without sufficient data, this classifier is not useful [38]. The Bayesian classifier is introduced here since it is applied in our method for classifying the motion vectors.

Bayesian decision theory is a fundamental statistical tool in pattern classification problems. The approach is based on probability of events and cost functions that will accompany decisions on selection of each class. In short, the decision is based on tradeoff between cost function and likelihood of events [41].

To illustrate the method, consider a two-category classification problem. We have two class, ω_1 and ω_2 and we are interested in determining weather a new sample x belongs to either class ω_1 or ω_2. A hypothetical class conditional probability density functions for two classes is shown in Figure 3.6. Suppose that we know both the priori probabilities $p(\omega_j)$ and the conditional probabilities $p(x|\omega_j)$ for $j = 1, 2$. Using the famous Bayes formula from probability theory

$$p(\omega_j|x) = \frac{p(x|\omega_j)p(\omega_j)}{p(x)} \tag{3.2}$$

where in the case of two-category problem

$$p(x) = \sum_{j=1}^{2} p(x|\omega_j)p(\omega_j) \tag{3.3}$$

Bayes formula shows that by observing value of x we can convert priori probability $p(\omega_j)$ to a posteriori probability $p(\omega_j|x)$ which is the probability of being in

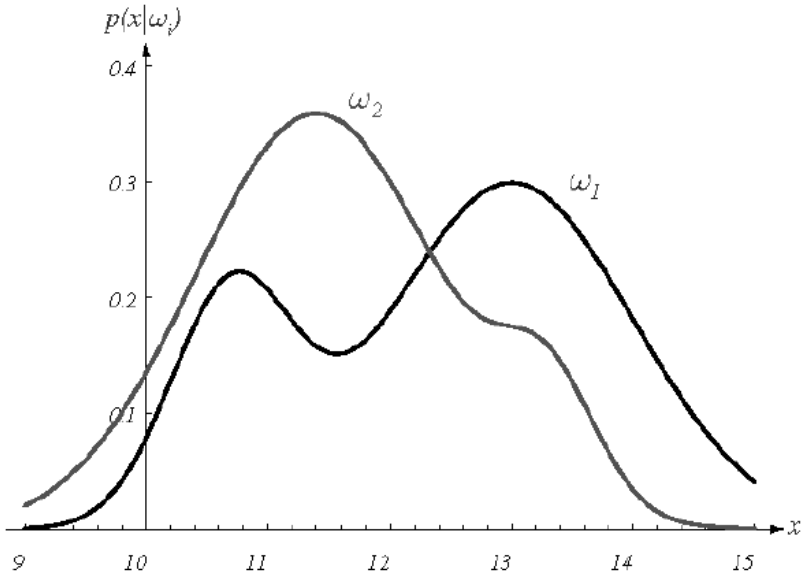

Figure 3.6: Hypothetical class conditional probability density functions for two classes [79]

class ω_j given that value x is observed.

Naturally, if we have observation from x for which $p(\omega_1|x)$ is greater than $p(\omega_2|x)$, then we can decide on class ω_1 and vice versa if $p(\omega_2|x)$ is greater than $p(\omega_1|x)$. Let us consider the probability of error if a decision on a specific class is made. The probability of error is [79]

$$p(e|x) = \begin{cases} p(\omega_1|x), & \text{if we decide } \omega_2 \\ p(\omega_2|x), & \text{if we decide } \omega_1 \end{cases} \tag{3.4}$$

and consequently the average probability of error is given by

$$P(e) = \int_{-\infty}^{+\infty} p(e,x)dx = \int_{-\infty}^{+\infty} p(e|x)p(x)dx \tag{3.5}$$

If the probability of error is small for each decision that is made, then the

integral must be small as well. Thus we have justified the following Bayes decision rule for minimizing the probability of error:

$$\text{Decide } \omega_1 \text{ if } p(x|\omega_1)p(\omega_1) > p(x|\omega_2)p(\omega_2); \text{ otherwise decide } \omega_2 \tag{3.6}$$

3.2.3 Non-Parametric Density Estimation

Generally, there are two approaches to probability density estimation, parametric and non-parametric. In parametric approach, specific form for the distribution function is assumed and next the required parameters of the selected distribution must be estimated by means of some techniques such as Maximum Likelihood (ML), Bayes estimation, etc. The potential problems with parametric approaches are that in practice we usually are not able to determine a specific form of the distribution. Therefore, nonparametric approaches are more useful for our purposes.

The probability of vector x falling in region R is given by

$$p(x \in R) = \int_R p(x')dx' \approx P \tag{3.7}$$

Thus, P represents an averaged or smoothed version of the density $p(x)$. Assuming reasonably that $p(x)$ is constant over R, to estimate $p(x)$ we could first estimate P.

The probability of k out of n samples fall into R is given by binomial distribution.

$$p(k \text{ out of } n \text{ vectors} \in R) = \binom{n}{k} p^k (1-p)^{n-k} \tag{3.8}$$

This analysis indicates that $p(k$ out of n vectors $\in R)$ is large when $k \equiv np$ and

it is small otherwise. On this basis, it can be assumed that it is likely that the number of observed vectors falling in R is the mean which is $P = k_{obs}/n$. Assuming $p(x)$ is constant over R, we can estimate (3.7) as

$$p(x \in R) = \int_R p(x')dx' \approx p(x)V \tag{3.9}$$

where $x \in R$ and define density volume

$$V = \int_R dx' \tag{3.10}$$

Using pervious results, we obtain

$$p(x) = \frac{k/n}{V} \tag{3.11}$$

The estimates converges to true value as $n \to \infty$ [36].

Parzen Windowing

As mentioned previously, nonparametric approaches estimate the pdf without any priori assumption on the form of distribution. The most fundamental techniques rely on the fact that the probability P that a vector x will fall in region R is given by

$$P = \int_R p(x')dx' \tag{3.12}$$

where P is smoothed version of $p(x)$ and we can estimate this smoothed probability by estimating the probability P.

Parzen-window approach can be introduced by assuming that region R is a d-dimensional hypercube. Let h_n be the length of an edge of the nth hypercube. Also define a window function:

$$\varphi(u_j) = \begin{cases} 1, & |u_j| \leq 1/2 \ ; \ j = 1, ..., d \\ 0, & \text{otherwise.} \end{cases} \tag{3.13}$$

where d is the total number of dimensions. Hence, $\varphi((x - x_i)/h_n)$ is equal to unity if x_i falls within the hypercube. It follows that the number of samples falling in the nth hypercube is given by

$$k_n = \sum_{i=1}^{n} \varphi \left(\frac{x - x_i}{h_n} \right) \tag{3.14}$$

With respect to the fact that integrative sum of probabilities must equal to one we can deduce

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi \left(\frac{x - x_i}{h_n} \right) \tag{3.15}$$

where V_n is the volume of each region [79, 35, 36]. An illustration of Parzen window method is illustrated in Figure 3.7.

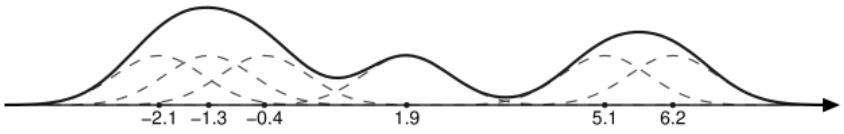

Figure 3.7: Parzen window illustration (taken from Wikipedia) [54]

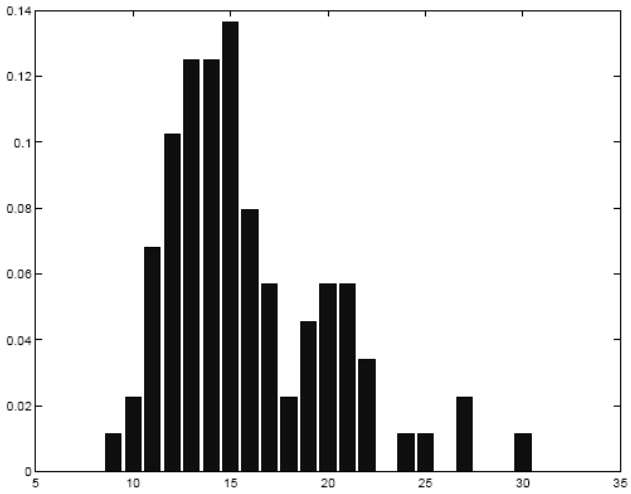

Figure 3.8: Histogram density estimation

Histogram Estimation

Consider an interval $R = [a, b]$; if $p(x)$ is constant over this interval, we can write

$$p(x \in R) = \int_a^b p(x)dx = p(a) \cdot (b - a) \tag{3.16}$$

Applying binomial distribution as described before, we see that the maximum likelihood estimation of the PDF is given by

$$p(x) = \frac{k/n}{b - a} = \frac{k}{n \cdot |R|} \tag{3.17}$$

where $|R|$ is the size of the region R, and this is exactly what is called a histogram. Given a set of bins R_i, we sort the M_i samples in each bin, and plot the approximating PDF as

$$\hat{p}(x) = \frac{M_i}{n \cdot |R_i|} \tag{3.18}$$

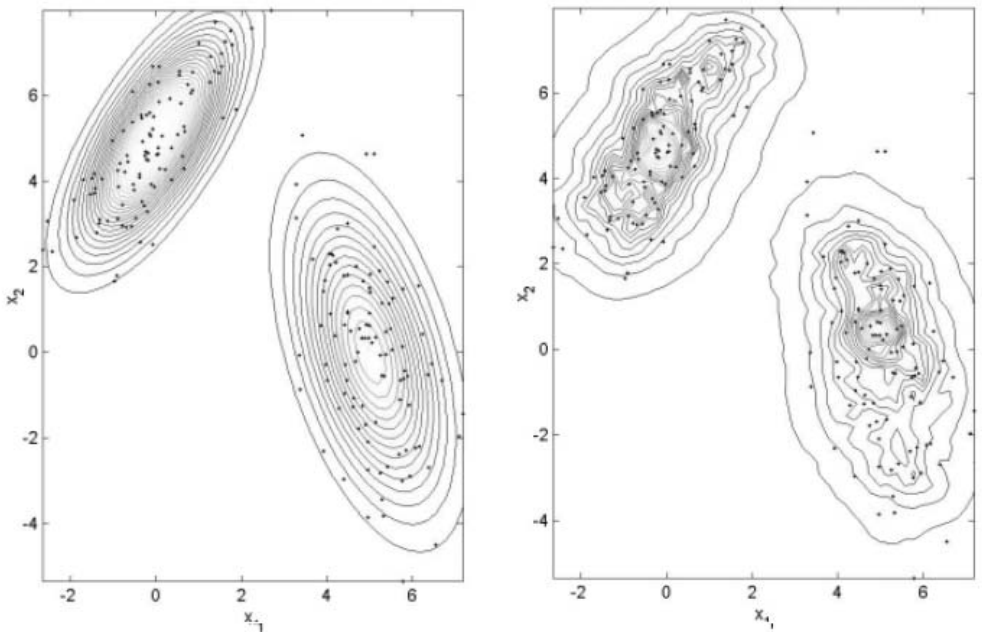

Figure 3.9: True density contours (left) vs KNN density estimate contours (right)[53]

It is important to note that these derivations are based on the assumption that the PDF is constant over each bin which may not be the case in most problems. However, if the size of each region can be taken infinitely small, then the approximation could converge to true PDF [79]. Histogram density estimation is shown in Figure 3.8.

K_N-Nearest-Neighbor Estimation

In Parzen window and Histogram estimation method, respectively, the width of the window function and region size are fixed. This explicitly controls the resolution along x axis and PDF resolution is data dependent.

In KNN method [36, 79, 37], we fix the number of samples k in each region and then determine the required size for each region to enclose this many samples. This

means that in this case we explicitly control the PDF resolution and resolution along x axis becomes data dependent. An example of KNN density estimation contours compared to true density contours are shown in Figure 3.9

To compute $p(x)$ at each point x, an interval $[x-\alpha, x+\alpha]$, centered around x is considered. We increase the factor α until k samples lie within the interval. The density estimate at x is

$$p(x) = \frac{k}{n \cdot |R(x)|} = \frac{k}{n \cdot 2\alpha} \tag{3.19}$$

where $R(x)$ is the smallest possible region, centered at x and contains k samples. $k = \sqrt{n}$ is conventionally selected. If the sample density is high, $|R(x)|$ and the estimate has high resolution where is needed and if the sample density is low, the region size $|R(x)|$ is large and density resolution will be lower which is acceptable in sparsely populated regions. The method avoids to have zero value for the n regions that no sample appears and in this way it estimates a realistic non-zero estimation proportional to $1/|R(x)|$. The main drawback of this method is that it is highly peaked and non normalized.

3.3 Overview of the Proposed Method

Real videos contain a mixture of motions with slow and fast contents. No fixed fast block matching algorithm can efficiently remove temporal redundancy of video sequences with wide motion contents. In this thesis, an adaptive fast block matching algorithm, called classification based adaptive search (CBAS) has been proposed. A Bayes classifier is applied to classify the motions into slow and fast categories. Accordingly, appropriate search strategy is applied for each class. The algorithm switches between different search patterns according to the content of motions within video frames. Experimental results show the proposed technique outperforms conventional stand-alone fast block matching methods in terms of both peak signal to noise ratio (PSNR) and computational complexity.

We have formulated the design of adaptive scheme as a two-category classification problem. The motion length of each macro block is predicted from neighbor blocks. Then, a Bayes classifier is applied to label the motion as slow or fast. Finally, appropriate search pattern is applied with respect to the label of motion to find the best matching block within the image frame. Adaptive rood pattern, proposed in [30], is selected for fast motion estimation and diamond pattern is selected for slow motions due to reasons given in subsection 3.4.1.

3.4 Estimation and Learning

In classification problems, Bayes classifier achieves the minimum probability of error [79]. Therefore, it is a suitable classifier for problems with known class conditional probability density function (PDF), $p(x|c_i)$. If the density functions are not known apriori, it is still possible to estimate an approximation for density functions from

labeled sample data, as discussed before . Since the functional form of class probability density functions are not known, the non-parametric estimation is applied. Different non-parametric estimation approaches are available: Histogram Estimation, k-Nearest-Neighbor (KNN) Estimation and Parzen widowing which employs Kernel smoothing functions to estimate the PDF. The Parzen windowing method can be summarized as follows: Given a sample $X_1, .., X_n$ with a continues, univariate density f, the Parzen density estimation is

$$\hat{f}(x,h) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - X_i}{h}) \tag{3.20}$$

where K is the Kernel and h is the bandwidth. Under mild conditions (h must decrease when n increases), the probability estimation converges to true probability. The histogram method introduces a tradeoff: for good resolution along x, small-sized regions are required. In KNN methods, the resolution along the PDF axis is data dependent, i.e. the resolution along the x-axis is explicitly controlled. The principal virtue of the KNN scheme is that it avoids setting $p(x)$ identically to zero in regions which happen not to have any samples, rather it results in a more realistic non-zero probability. The principal drawback of the KNN method is that the estimated PDF is highly peaked and non-normalized. In addition, KNN methods are usually time-consuming and complex which might be undesirable in practical implementations with online access and limited storage space. We apply Parzen windowing with gaussian smoothing function to estimate the conditional PDF from sample labeled data. The main advantages of Parzen window method are its simplicity and fast implementation.

The selected feature for classification is length of motion vectors (MVs). For the first image frame, since there is no pervious data to be used for PDF estimation,

the algorithm follows a rigid thresholding approach which is comparing the length of motions to a predefined threshold and classify the vectors in two groups of slow and fast motions. After motion classification for each macroblock, appropriate searching scheme is employed for that macroblock. After computation of all the motion vectors and their class label, they can be utilized by parzen window method to estimate class conditional PDFs. Starting from second frame, motion vector of each macroblock, x, within the image is predicted by motion vector of immediate left macroblock. Then using class conditional PDFs that are estimated from pervious frame, Baysian classifier is applied to classify x as either slow or fast and accordingly apply relevant search scheme.

Knowing the class probability density functions, $p(x|C_{fast})$ and $P(x|C_{slow})$, the bayes classifier can be expressed as follows:

The motion is classified as fast motion, if:

$$P(C_{\text{Fast}}|x) > P(C_{\text{Slow}}|x) \tag{3.21}$$

From the bayes formula we have,

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)} \tag{3.22}$$

substituting (3.22) into (3.21) we obtain

$$\frac{P(x|C_{\text{Fast}})P(C_{\text{Fast}})}{P(x)} > \frac{P(x|C_{\text{Slow}})P(C_{\text{Slow}})}{P(x)} \tag{3.23}$$

or

$$P(x|C_{\text{Fast}})P(C_{\text{Fast}}) > P(x|C_{\text{Slow}})P(C_{\text{Slow}}) \tag{3.24}$$

Moreover, if the probabilities of having image frames with rapid or slow motion content are assumed to be equal, the classification criterion can be simplified as follows:

$$P(x|C_{\text{Fast}}) > P(x|C_{\text{Slow}}) \tag{3.25}$$

An important point for consideration is an appropriate selection of region of support (ROS) which is defined as neighboring blocks whose MVs are used to predict the motion vectors in the current block and the algorithm used to compute the predicted motion vectors for each class. Exhaustive experiments on considering different sets of immediate left, above-left, above and above-right to the current block and two types of prediction criteria–mean and median operation on lengths of motions in ROS, have been performed in [30]. The experiments show that the results have fairly similar performance in terms of PSNR. Hence, we apply the least complex choice, i.e., using the immediate left block for predicting the motion vector of current block. After computing all the motion vectors of current frame, we can update the PDFs to be used for motion classification in next frame. The procedure is repeated for subsequent frames. This technique is able to adapt itself to the contents of motions and establish higher performance quality compared to stand-alone fast block matching algorithms. Simulation results provided in section 4.2 are provided in support of our proposed algorithm.

3.4.1 Selection of Search Patterns

A series of experiments on standard block matching techniques have been conducted on selected video sequences containing variety of motion contents. The performance parameters for each of the algorithms in each video sequences are recorded and compared to each other. The peak signal to noise ratio (PSNR) and

computational complexity have been employed to evaluate performance of the algorithms on sequences with different motion contents. Observations show that for small motions (less than 3 blocks), the algorithms with compactly spaced points result in more accurate approximations of motion vectors. Among the tested search patterns, DS shows superior results for sequences with small motions. Prohibitive nature of DS, i.e. prevention from being trapped into local minima, in addition to appropriate accuracy, led us to select this algorithm for block matching search when the motion is classified as slow. In addition, DS is successful for prediction of motions with moderate lengths (3 to 4 blocks). As we mentioned before, both small and moderate motion vectors have been classified under Slow category and same searching algorithm, i.e. DS, is employed for prediction of motions in this class.

For fast motion, the proposed method uses a rood pattern method suitable for this class. In [30], a rood pattern with one point at center and four search points located at the four vertices has been proposed. The main structure has a symmetrical rood shape, and its size refers to distance between vertices and the center point of the rood. The choice of rood shape is based on the observations on real-world video sequences. The MV distribution in the vertical and horizontal directions are higher than that in other directions since most of the camera movements occur in these directions [30]. The size of the rood is adaptive with respect to predicted length of the current block's motion vector. The prediction of target MV, is obtained through MVs of neighbor blocks' vectors. The flexible size of the rood, prevents the search to be trapped in local minima which is of importance when searching for blocks with fast motions. The rood search will be followed by small diamond search pattern (SDSP) steps until the best match occurs at the center of the pattern.

Simulation results, given in Chapter 4, demonstrate that the proposed technique

outperforms conventional fast block matching methods in terms of higher PSNR and less computational complexity. In summary, an intelligent encoder should apply adaptive motion estimation techniques instead of relying on fixed patterns. The ideas of machine learning and pattern recognition can be applied for the design of adaptive intelligent motion estimation techniques.

Chapter 4

Simulation Results

4.1 Definitions and Assumptions

Simulations are based on the encoding platform under MPEG-4 test conditions where each sequence contains 100 frames (except claire sequence with 30 frames) and has QCIF (Quarter Common Intermediate Format) or CIF (Common Intermediate Format) formats. For comparison, the average peak to peak signal to noise ratio of our classification based adaptive search (CBAS) has been computed for various video sequences and compared to other standard block matching motion estimation methods including FS, DS, TSS, NTSS and FSS. Computational complexity is measured by computing average number of checking points per MV generation which is also related to speed of match finding where computational gain is defined as the ratio of the search speed of full search (FS) or exhaustive search (ES) to that of our algorithm.

4.2 Characteristics of Video Sequences and Simulation Results

Claire sequence (Figure 4.1), contains relatively slow motions within the frames. The PSNR performance results (Figure 4.2) show that all algorithms have close performance. The similarity of PSNR performance is that this video sequence contains very small motions between any consecutive frames. FBMAs introduce much less computations compared to FS (Figure 4.3). Our algorithm CBAS has the best performance in terms of computations and also best performance in PSNR among FBMAs.

Diskus sequence (Figure 4.4) contains high movement of camera during the frame sequence. Panning, zooming, and change of shot are characteristics of this sequence. Around frame number 20 to 40 in Figure. 4.5, there is gradual decrease in PSNR performance for all the tested algorithms. This is because camera zooming during these frames results in less accurate motion vectors and as a result, lower quality of compensated frames. Although this sequence is a mixture of wide variety of motions from slow to fast, our method stands in second order after FS and on the top of all FBMAs, while it also introduces the least computations (Figure 4.6).

Flower garden (Figure 4.7) is an example of video with rapid movement of camera in one direction while new objects appear as camera moves forward and some objects disappear. This results in rapid increase and decrease in PSNR performance (Figure 4.8). CBAS is the most efficient methods for handseling this sequence and shows best performance in terms of computations (Figure 4.9).

Mother and daughter (Figure 4.10) is a video sequence similar to *claire* sequence with slightly faster movements. Results of this sequence are very similar to *clair*

sequence, since both contain very slow motions. PSNR and computation results have been shown respectively in Figure 4.11 and Figure 4.12.

Osu sequence (Figure 4.13) is a selected sequence because it has a wide range of motions, starting with small movements to gradual but rapid change of shot continuing with fast displacement of camera. Although it is composed of small motions and all the algorithms have very similar performance in terms of PSNR (Figure 4.14), there is a shot change from frame 55 to 60 which results in decrease in PSNR values. Again, CBAS is the most efficient in terms of computations (Figure 4.15).

Table tennis sequence (Figure 4.16) has been used in verification of MPEG video standard and contains slow, moderate and fast movement and abrupt change of shot during the sequence. Compared to our algorithm, NTSS and TSS have better performance in terms of PSNR (Figure 4.17), while in terms of computations CBAS has the best performance (Figure 4.18). There is a shot change around frame 88 which results in abrupt decrease of PSNR value for all the algorithms.

All comparison results for Diskus and flower garden sequences in terms of PSNR performance and computational complexity are provided in Table 4.1 and Table 4.2. The results shows that CBAS, introduces the best performance in terms of computations for all sequences. In terms of PSNR, it stands the second after FS and on top of all FBMAs, expect for *Table tennis* sequence which TSS and NTSS have better performance. The reason could be related to the nature of video in this sequence.

We also provided the comparison results of our algorithm with A-TDB which is another adaptive motion estimation algorithm. Table 4.3 and Table 4.4 show the PSNR and computation performance, respectively. A-TDB shows better PSNR

performance for different sequences, while in average CBAS has better computation performance.

In all the figures in simulation results, SESTSS is equivalent to three step search (TSS) and SS4 is equivalent to four step search (FSS).

(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.1: *Claire* sequence of frames

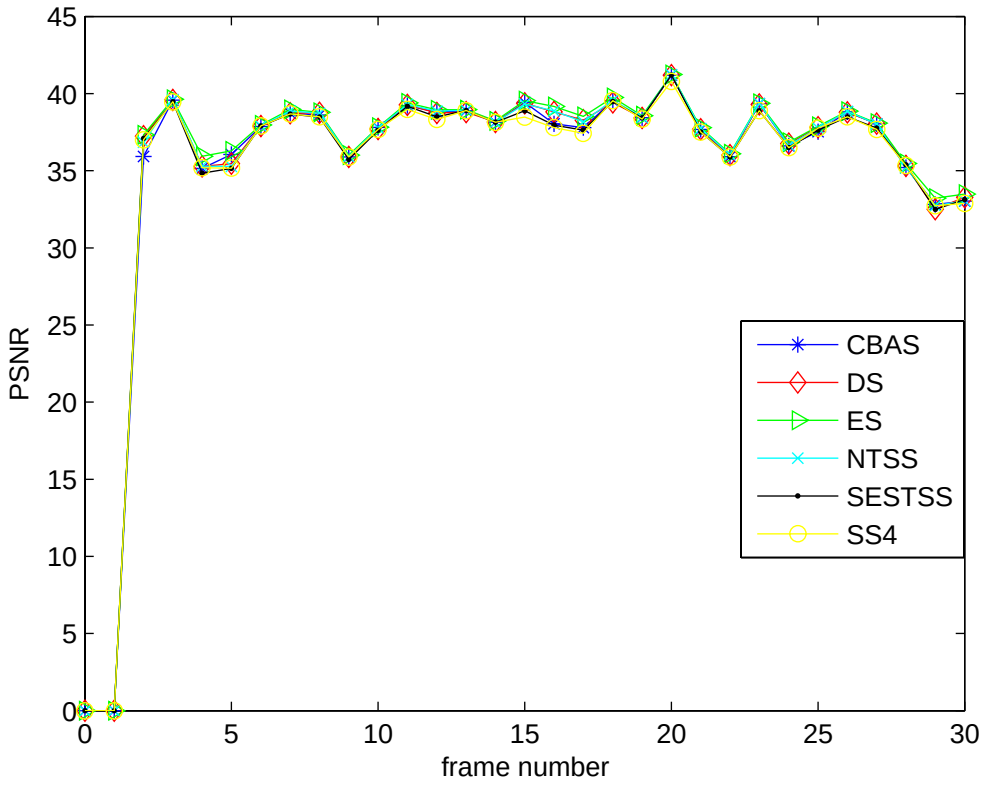

Figure 4.2: PSNR performance over sequence of frames (PSNR performance of standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparision to classification-based adaptive search (CBAS))

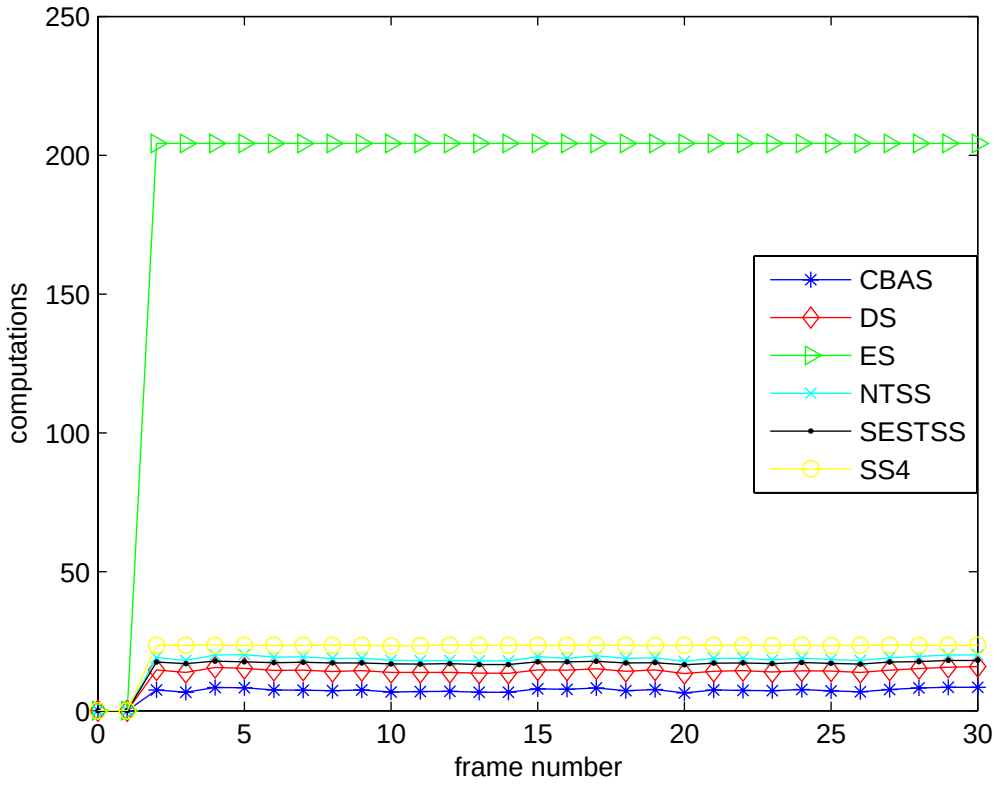

Figure 4.3: Computational complexity over sequence of frames (number of computations per block for standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparison to classification- based adaptive search (CBAS))

(a) (b)

(c) (d)

(e) (f)

Figure 4.4: *Diskus* sequence of frames

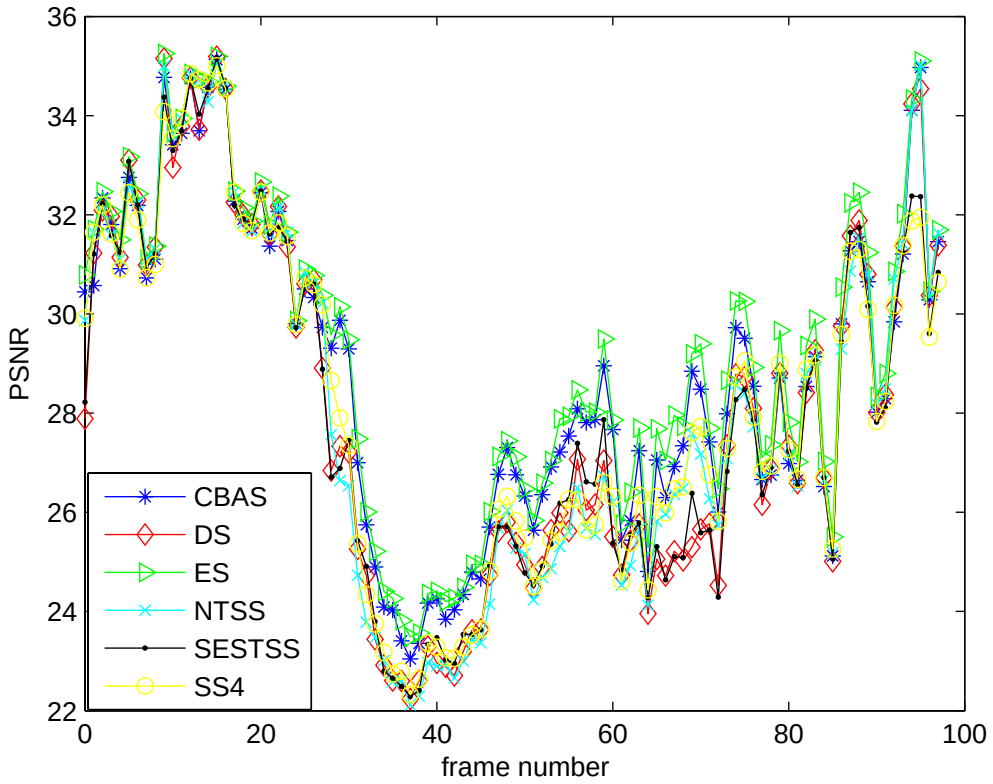

Figure 4.5: PSNR performance over sequence of frames (PSNR performance of standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparison to classification-based adaptive search (CBAS))

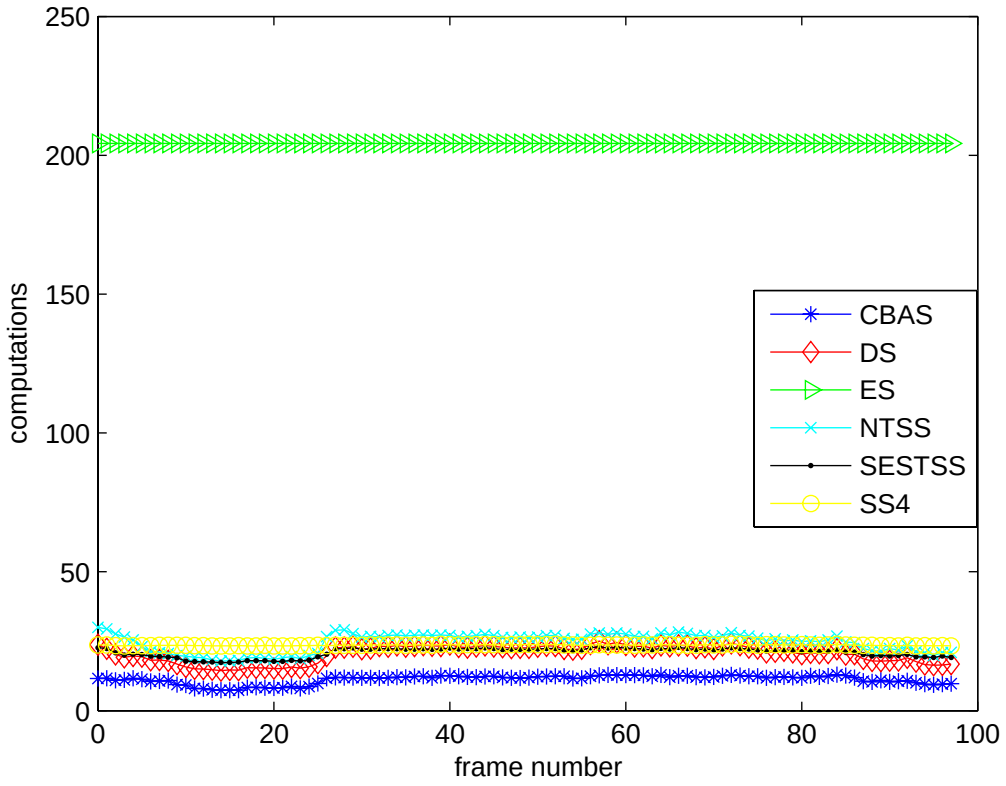

Figure 4.6: Computational complexity over sequence of frames (number of computations per block for standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown compared to classification- based adaptive search (CBAS))

(a) (b)

(c) (d)

(e) (f)

Figure 4.7: *Flower garden* sequence of frames

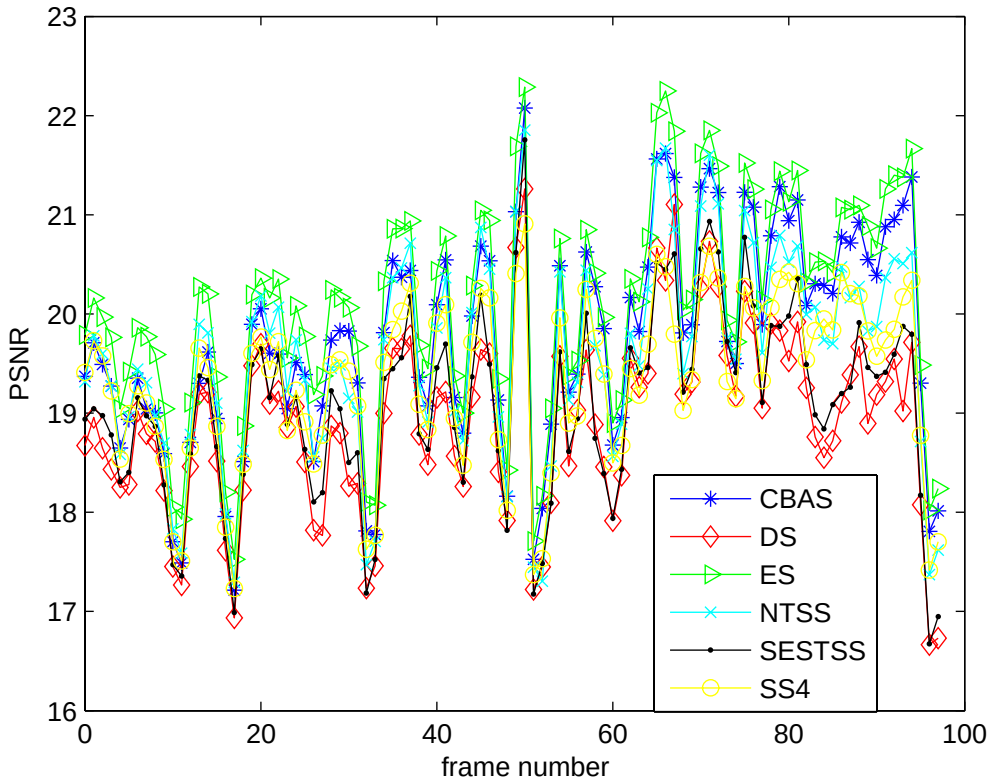

Figure 4.8: PSNR performance over sequence of frames (PSNR performance of standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparison to classification-based adaptive search (CBAS))

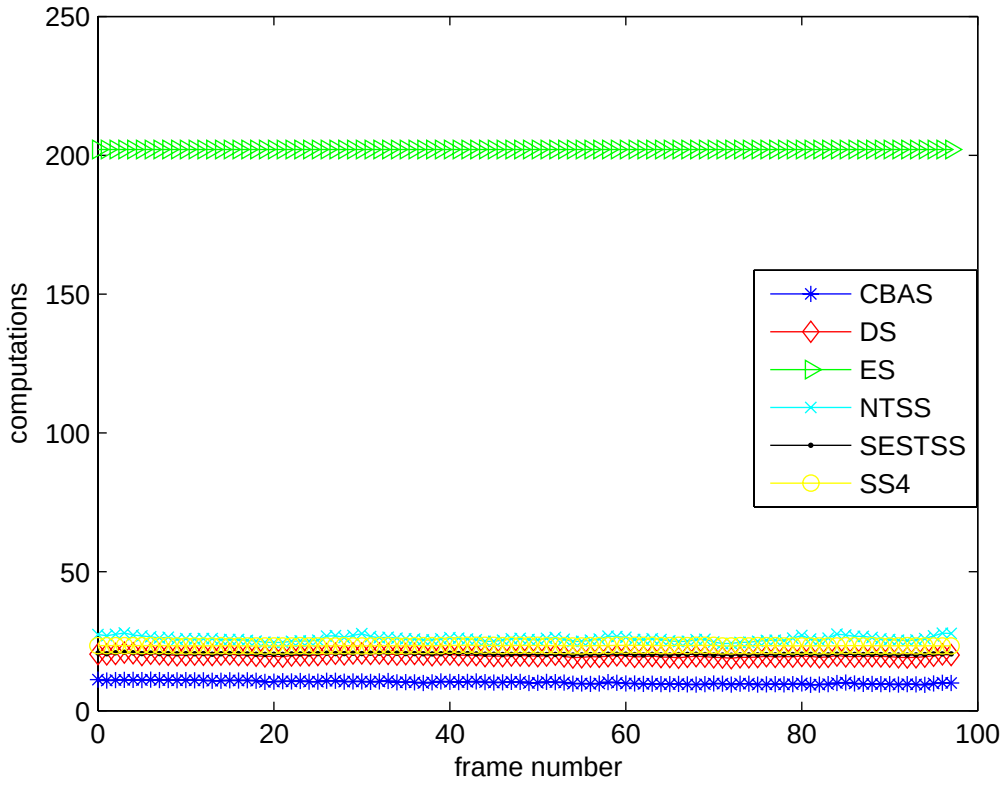

Figure 4.9: Computational complexity over sequence of frames (number of computations per block for standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparison to classification- based adaptive search (CBAS))

(a) (b)

(c) (d)

(e) (f)

Figure 4.10: *Mom & Daughter* sequence of frames

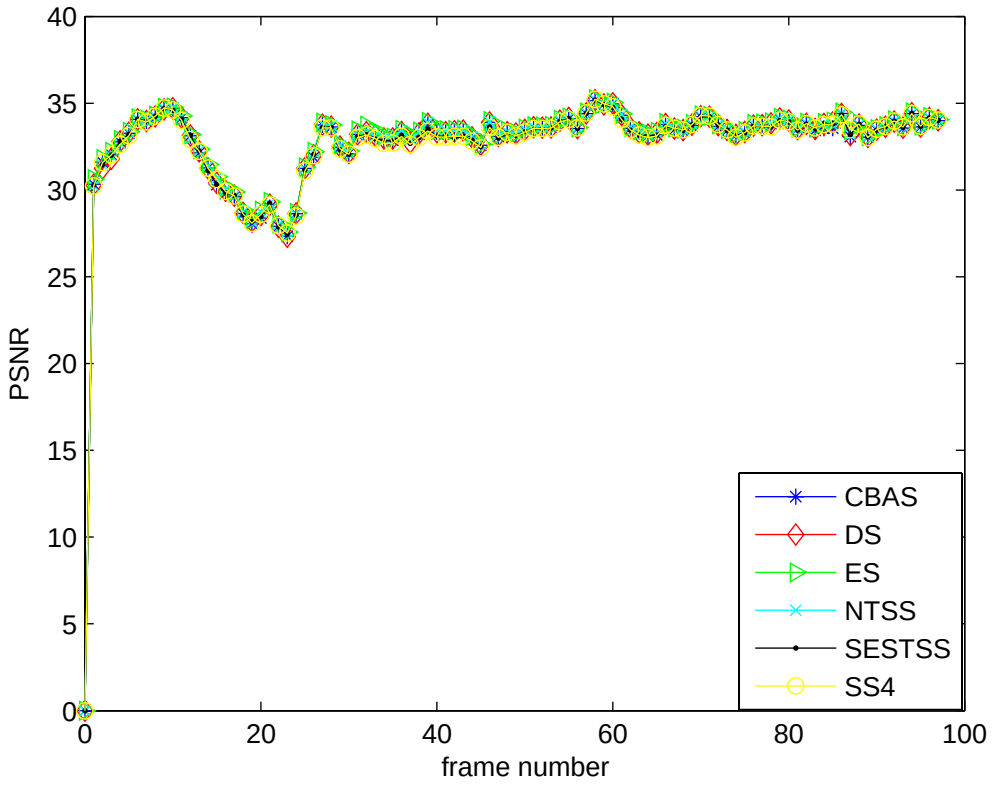

Figure 4.11: PSNR performance over sequence of frames (PSNR performance of standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparison to classification- based adaptive search (CBAS))

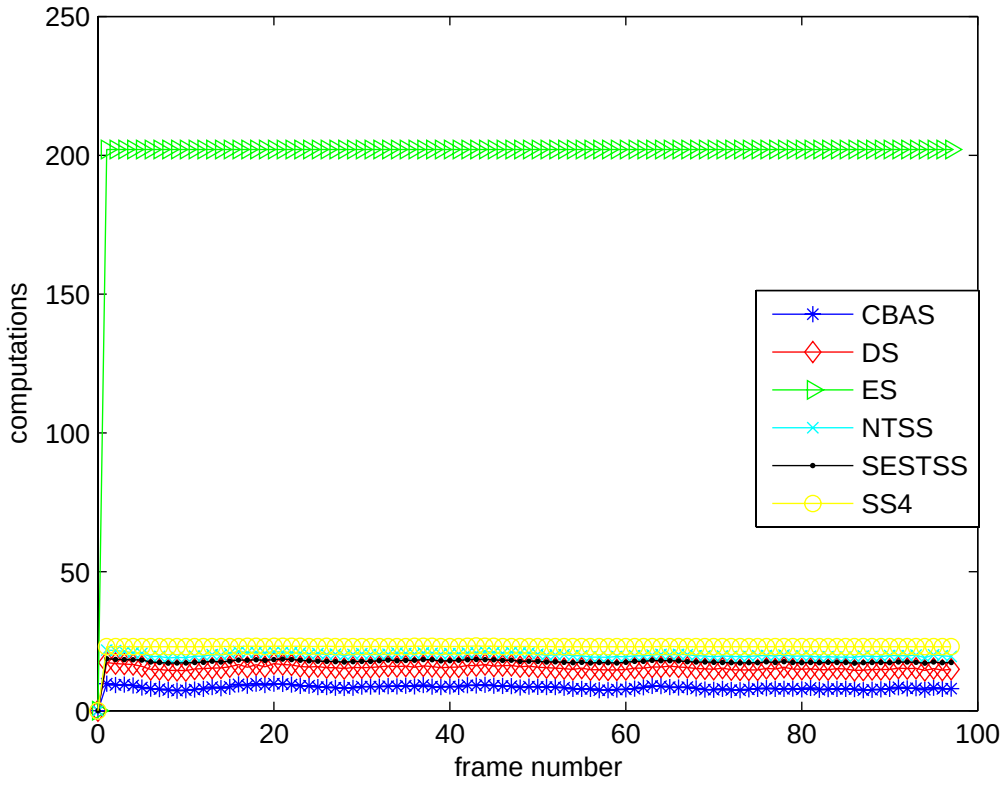

Figure 4.12: Computational complexity over sequence of frames (number of computations per block for standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparison to classification- based adaptive search (CBAS))

(a)

(b)

(c)

(d)

(e) PSNR

(f)

Figure 4.13: *Osu* sequence of Frames

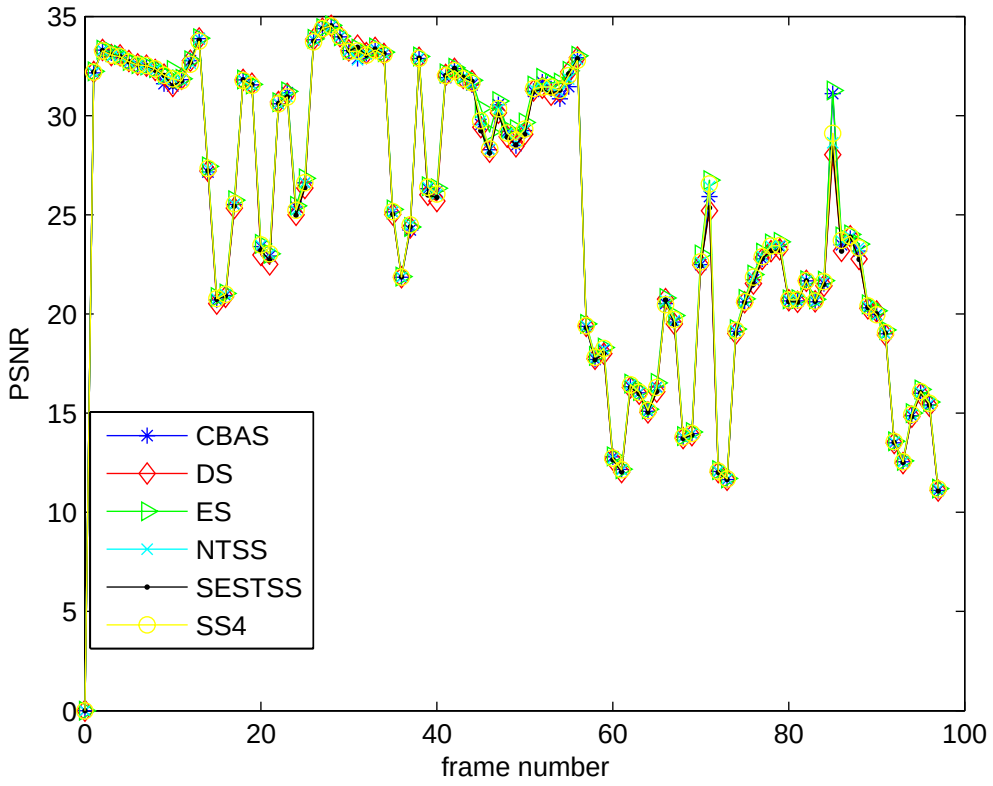

Figure 4.14: PSNR performance over sequence of frames (PSNR performance of standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparison to classification- based adaptive search (CBAS))

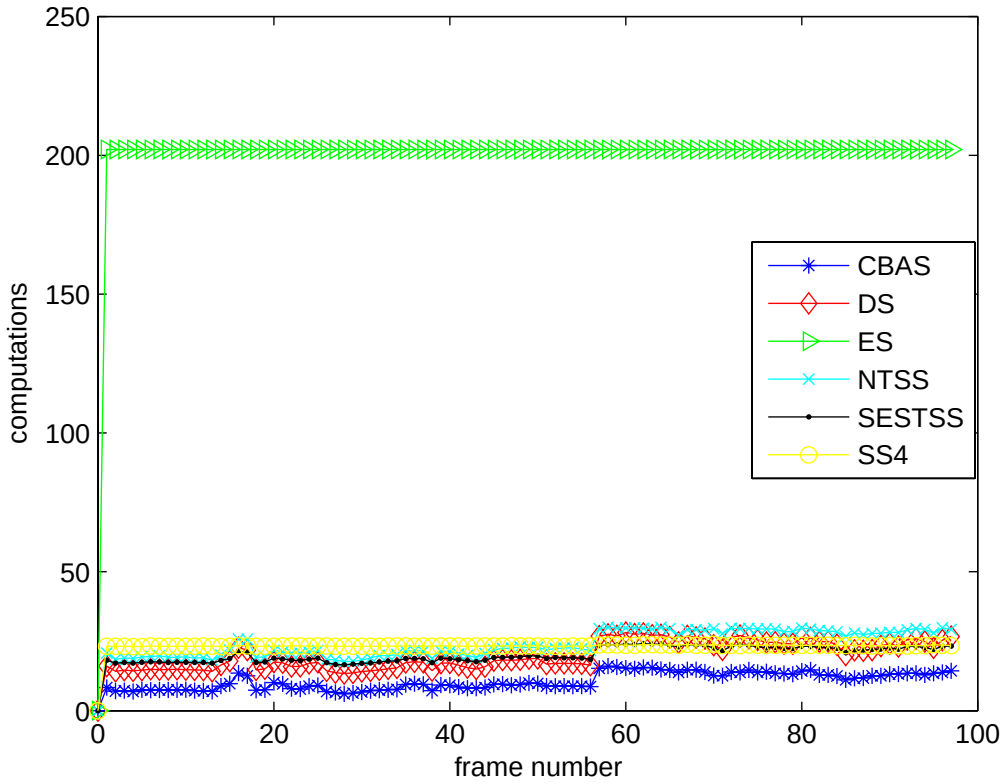

Figure 4.15: Computational complexity over sequence of frames (number of computations per block for standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparison to classification- based adaptive search (CBAS))

(a) (b)

(c) (d)

(e) (f)

Figure 4.16: *Table tennis* sequence of frames

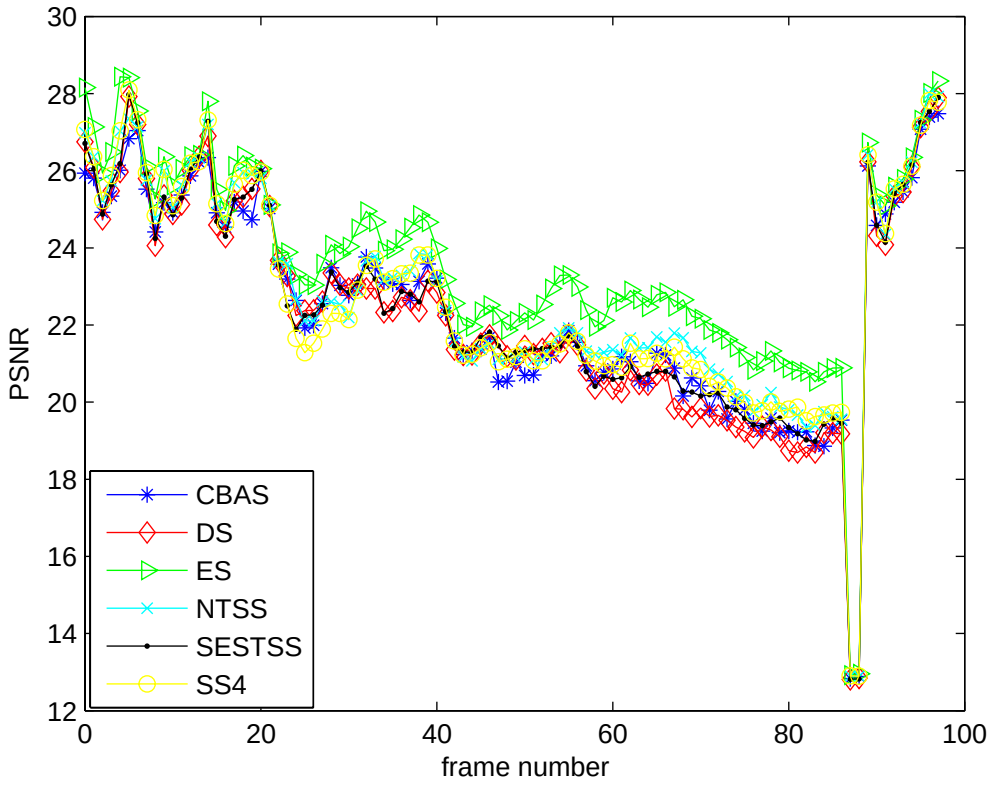

Figure 4.17: PSNR performance over sequence of frames (PSNR performance of standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparison to classification- based adaptive search (CBAS))

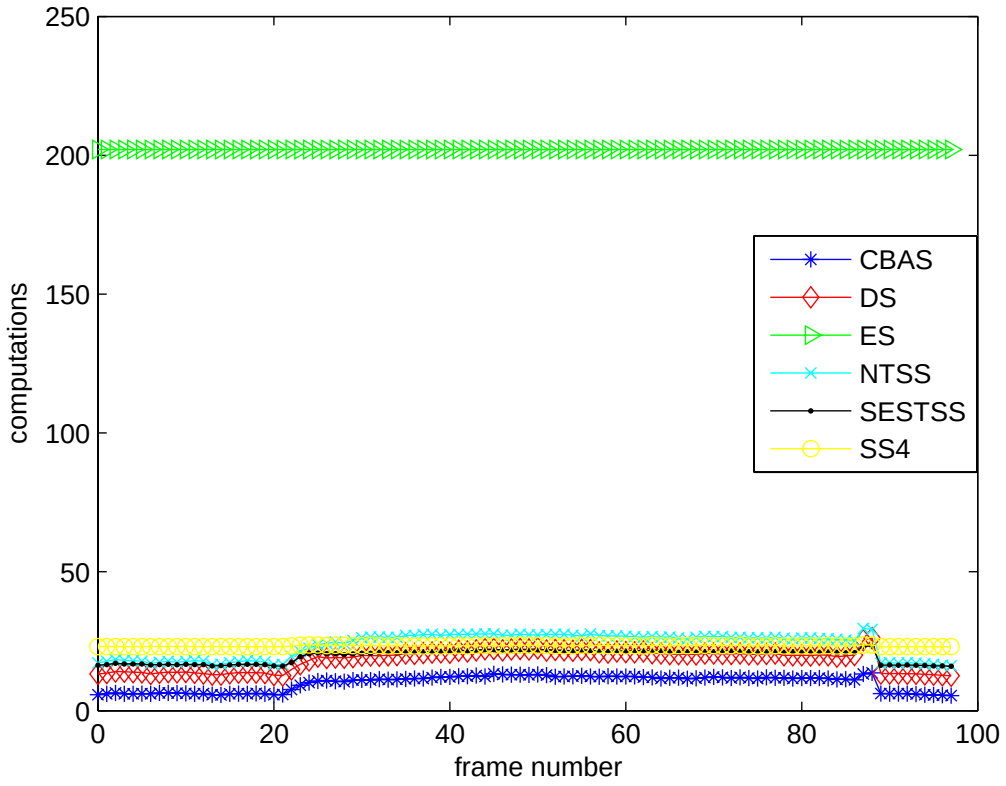

Figure 4.18: Computational complexity over sequence of frames (number of computations per block for standard FBMAs (ES, DS, TSS, NTSS, FSS) are shown in comparison to classification- based adaptive search (CBAS))

Algorithm	Claire	Diskus	Flower-Garden	Mom-daughter	Osu	Table Tennis
ES	35.4233	29.14	20.093	32.7653	25.2609	23.6390
CBAS	35.1604	28.7336	19.7496	32.6472	25.0965	22.4553
DS	35.2414	28.0300	18.9325	32.6567	24.9953	22.3714
TSS	35.0276	28.1621	19.3349	32.5886	25.1002	22.6962
NTSS	35.248	28.0983	19.5930	32.6980	25.0882	22.7974
FSS	35.0631	27.9853	19.0849	32.5896	24.9918	22.496

Table 4.1: Average PSNR for standard FBMAs comparing to classification-based adaptive search (CBAS)

Algorithm	Claire	Diskus	Flower-Garden	Mom-daughter	Osu	Table Tennis
ES	191.1033	204.2828	202.048	199.986	199.988	202.0485
CBAS	6.9177	11.2040	10.1481	8.2159	10.5208	9.9573
DS	13.4506	20.5541	19.7864	15.2746	19.725	18.3302
TSS	21.9751	23.5653	23.383	22.824	23.1156	23.2860
NTSS	17.6404	24.6127	25.818	19.883	23.8546	23.4972
FSS	16.1209	20.9394	20.343	17.512	19.9850	19.7360

Table 4.2: Average number of computations per block for standard FBMAs comparing to classification-based adaptive search (CBAS)

Algorithm	Flower-Garden	Mom-daughter	Table Tennis
A-TDB	23.5995	39.6489	26.7633
CBAS	19.7496	32.6472	22.4553

Table 4.3: Comparison of PSNR between CBAS and another adaptive video motion estimation algorithm (A-TDB)

Algorithm	Flower-Garden	Mom-daughter	Table Tennis
A-TDB	16.51	3.55	10.28
CBAS	10.1481	8.2159	9.9573

Table 4.4: Comparison of computational complexity between CBAS and another adaptive video motion estimation algorithm (A-TDB)

4.3 Summary

The PSNR and computation performance results illustrate that CBAS has better PSNR performance and less computation than other algorithms, including state-of-the-art DS algorithm while also introduces less computations. In comparison with ES, our algorithm greatly improves the search speed. CBAS is almost 12.75 times faster than ES while the PSNR level closely follows that of the ES with slight degradation (less than $0.10 - 0.13$ dB). The algorithm is able to maintain rather consistent PSNR performance. The efficiency of our algorithm largely dependent on the precision of estimated probability functions and selection of suitable search scheme for each class.

Chapter 5

Hierarchical Classification-Based Video Shot Detection

5.1 Introduction

Video shot detection refers to the process of detecting transition occurring between scenes in a digital video stream. It can provide disjoint contiguous video segments that can be utilized as basic units to be indexed, annotated and browsed. The detection of shot cut involves detecting a significant change in visual content between two frames or gradual change within number of frames. The development of video shot boundary detection techniques have the longest and richest history in the area of content-based video analysis and retrieval [60, 62]. The importance of video shot detection algorithms initiates from the necessity for almost all video abstraction and high level video processing. The detection of Scene breaks and partitioning of video into short homogenous temporal segments is the first step toward annotation of digital video sequences. We may be able to replace fast forward button on video

browsers by a button that searches for next scene break [74]. Shot segmentation is also important for other applications such as motion-based compression algorithms such as MPEG in the way that they can achieve higher compression ratio without sacrificing the quality of data if the locations of scene breaks are known [74]. For example, in the process of coloring black and white movies, information about location of shot boundaries can provide time stamps for switching between different gray-to-color look-up tables [60].

A shot is defined as an unbroken sequence of frames taken by one camera. Using motion picture terminology, shot change can belong to one of the following categories [61]:

- Cut: This is an abrupt change between two consecutive frames where one frame belongs to the disappearing shot and the other belongs to an appearing shot.
- Fade: Either the intensity of disappearing shot changes from normal into black frame (fade out), or intensity of the black frame changes into appearing shot (fade in).
- Dissolve: In this case, few frames of disappearing shot overlap with few appearing frames of appearing shot. The intensity of disappearing shot decreases to zero (fade out) while intensity of appearing shot increases from zero (fade in).
- Wipe: Here, the appearing and disappearing shots coexist in different spatial regions of the intermediate video frames, and the region occupied by former grows until it gradually replaces the latter.

The first step in shot detection algorithm is to extract one or more features from video frame or subset of it called region of interest (ROI). The algorithm can then use different techniques to detect shot changes and classify type of changes. Among various existing measurement techniques, information theoretic measures provide better results because it exploits the inter-frame information in a more compact way than frame subtraction. In attempt for designing threshold free shot segmentation algorithms, some existing works in literature consider the problem from a different perspective. For example, transforming the temporal segmentation problem into a multi-class categorization issue. The work in [64], is an example of supervised classification methods for video shot segmentation. Computer vision techniques allow content-based processing of video frames. Various methods for detection and classification of scene breaks have been proposed. Different features of the video data have been used in video shot detection. Some of these features are

1. Average intensity measurement: The average of the intensity values for each component (YUV, RGB, etc.) in each frame is computed and compared to the successive frame.

2. Euclidian distance: The frame is divided into series of blocks and then Discrete Cosine Transform (DCT) is performed on each block. In [56], the Euclidian distance between mean of DC values of blocks has been utilized as a degree of dissimilarity between frames.

3. Histogram comparison: It is based on subtracting the histograms, e.g., gray level histograms of subsequent image frames.

4. Likelihood ratio: Generating the measure of likelihood that two corresponding regions are similar. Each region is represented by a second order statistics

under the assumption that this property is constant over the region. For example we can divide the frames into blocks and then compute the likelihood ratio calculation over the blocks.

5. Motion estimation vectors: We estimate the next frame in a video sequence based on information acquired from estimating the motion vectors. Then the absolute difference between the reconstructed frame and the original frame is calculated and summed.

6. Edges: Edges are very informative. The number of edge pixels and their locations can be computed in any successive frames. If the edges are appearing or disappearing far from the edges in pervious frame, we can recognize that a scene break has occurred.

A typical shot boundary detection system is shown in Figure 5.1. Most of the shot detection methods rely directly on intensity data and have difficulty with dissolves and motions within scenes. In the first step of shot break detection, feature extraction is employed. Then a metric is selected to compute the value of dissimilarity between frames with respect to the selected feature or features. This dissimilarity value serves as input to shot cut detector and it is compared against a *threshold*. If the threshold is exceeded then shot cut between frames is detected.

Despite various techniques proposed by researchers, we can relate the following major criteria addressed in [60] to rank the degree of success of an algorithm.

1. Excellent performance in detection for all types of shot changes (instantaneous and gradual boundaries)

2. Constant quality of performance for all types of videos with minimal need to fine tuning of detection parameters.

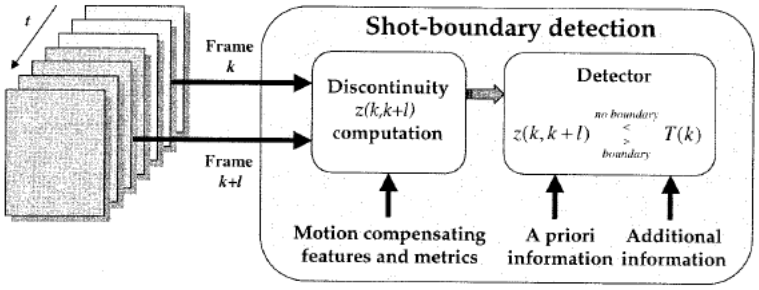

Figure 5.1: A typical shot boundary detection system [60]

If the detection performance is poor, significant involvement of operator is required in order to correct wrong decisions. In addition, if the parameters of shot detection algorithm are highly sequence dependant, it would be hard for the operator to find the optimal parameters for each sequence. Bad detection performance may negatively affect the performance of high level video analysis. In this introductory background, we address the issues that must be taken into consideration when designing the video shot detection algorithm specially in light of the criteria addressed before. It can be realistically assumed that changes in visual contents of frames are mainly caused by camera motions or lighting changes. The best way to tackle the problem that might occur according to these changes, is to select features and metric measures which are least sensitive to these variations. However, the influence of strong and abrupt lighting changes on detection performance can not be easily reduced. In addition, further improvement to the detection performance can be accomplished if *a priori* information about distribution of shot boundaries is available [60].

5.2 Review of Shot Segmentation Methods

Comprehensive overview on existing methods in video shot segmentation both on uncompressed and compressed data can be found in [60], [61], [63], [77].

For uncompressed domain, most of the algorithms are based on suitable thresholding of differences between successive frames. However these thresholds are highly sensitive to the type of video. A supervised classification method by transforming the temporal segmentation problem to a multi-class categorization issue is proposed in [64]. The paper in [65] views the problem as a 2-class clustering problem and uses k-means to cluster frame differences. From computer vision approaches, the paper in [66] applies graph cuts to find the globally optimal segmentation of the N-dimensional image.

Several information theoretic approaches have been proposed by researchers to overcome the problem. The paper in [67] uses mutual information (MI) and affine image registration. The MI measures statistical difference between frames while affine registration compensates for camera movements, panning and zooming. Another paper in [68] introduces information theoretic metrics for detecting cuts, fade-ins and fade-outs which rely on MI and joint entropy (JE).

Application of color information is used in [69] to partition the video into dynamically homogeneous segments using the criterion inspired by compact coding theory. They have performed information-based segmentation using the Minimum Message Length (MML) criterion and minimization by Dynamic Programming Algorithm (DPA). This method is capable of detecting all types of transmission in a generic manner.

A shot cut detection technique is presented in [70] which applies combination of multiple experts by exploiting the complementarity of expert knowledge, i.e. the

fact that various experts calculate different features of the video sequences. This method significantly gives better results compared to those experts alone.

In [58], the detection of scene breaks is improved by the use of threshold that adapts itself to the statistics of the sequence. A single statistic is generated for each pair of frames which quantify the degree of dissimilarity between the two frames. It has been assumed that dissimilarity measures come from two stationary distributions: one for shot boundary and one for a non-shot boundary. We assume that the cost of false positives and true negatives are the same, then simple hypothesis testing approach yields the value of optimal threshold which will result in smallest error probability. Another approach for adaptive shot detection scheme proposed in [59] suggests the use of three different features and then combine the results of shot detection from all three measures to decide for the best location of shot boundary. The proposed algorithm can be divided into two stages. First extracting all the needed features which in their case are fast fourier transform (FFT), YUV and gray histogram. The second part is to decide whether the shot cut has occurred with respect to each of these features independently by considering the difference between two consecutive frames compared to a threshold which can be determined explicitly or adaptively. Then, combining the results to form the final list of shot boundaries. To determine the adaptive threshold, they defined it as a percentage value of the maximum difference. According to the change of maximum dissimilarity, the amount of threshold changes with content of data. Finally, for each feature, shot boundaries belonging to the same shot are merged. Another similar strategy which combines multiple experts knowledge for classifying video sequences has been suggested in [70]. In [60] a statistical detector that is based on minimization of average detector error probability has been proposed. The problem of video shot detection has also been explored in frequency domain. The work reported in [55]

calculates the normalized correlation field in frequency domain instead of spatial domain.

As pioneered by the above methods, classification methods show promising results for this task. However, most existing shot detection algorithms use ad hoc frame classification with arbitrary thresholding rule [64]. To illustrate these algorithms, two approaches are described in the following two subsections. First an information theoretic approach is described. Subsequently a feature based method for scene change detection is described.

5.2.1 Information Theoretic Approaches to Shot Detection

With respect to the use of information theory in our classification based shot detection method, we briefly review some of the existing work in literature that also benefit from information theory.

In [67], mutual information (MI) and affine image registration are used to solve this problem. The advantages of this approach is high robustness to illumination changes within a shot and easy parallelization. To define MI, let us consider X and Y to be two random variables with marginal probability distribution $p(x)$ and $p(y)$ and joint probability distribution $p(x, y)$. The MI between X and Y is

$$I(X;Y) = H(Y) - H(Y|X) = -\sum_{x,y} p(x,y) \cdot log(\frac{p(x,y)}{p(x) \cdot p(y)}) \tag{5.1}$$

While joint entropy is defined as

$$H(X,Y) = -\sum_{x,y} P_{XY}(x,y) \cdot logP_{XY}(x,y) \tag{5.2}$$

where $H(.)$ is Shannon entropy. In other words, MI contains information that one random variable contains about other random variable which bears the context that it can be used as a measure of similarity among two random variables. A large MI between two adjacent images, implies large dependance of image frames with respect to the selected features. Another work which also uses mutual information as a measure of similarity between adjacent frames has been done in [57]. Mutual information is used for detecting the abrupt cutes when a large difference occurs in color content of frames. A large difference in color content, results in small value of mutual information. MI and joint entropy between two successive frames are calculated separately for each of the RGB components. Let us consider that gray levels vary between 0 to $N-1$. Then we obtain three $N \times N$ matrices $C^R_{t,t+1}, C^G_{t,t+1}, C^B_{t,t+1}$ that carry information between successive frames f_t and f_{t+1}.

Following the definition of MI in equation (5.1), we obtain [57]

$$I^R_{t,t+1} = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} C^R_{t,t+1}(i,j)\ log\frac{C^R_{t,t+1}(i,j)}{C^R_{t,t+1}(j)} \tag{5.3}$$

and total mutual information is given by

$$I_{t,t+1} = I^R_{t,t+1} + I^G_{t,t+1} + I^B_{t,t+1} \tag{5.4}$$

Similarly, the joint entropy for each component can be computed as follows

$$H^R_{t,t+1} = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} C^R_{t,t+1}(i,j)\ logC^R_{t,t+1}(i,j) \tag{5.5}$$

and the total joint entropy is obtained by:

$$H_{t,t+1} = H^R_{t,t+1} + H^G_{t,t+1} + H^B_{t,t+1} \tag{5.6}$$

In [57], an adaptive threshold is applied in order to detect the outliers. The average of MI between successive frames over a window W of size N_W is obtained at each time instance and is trimmed at the current window center:

$$\bar{I}_{t_c} = E[I_{t,t+1}],\ t \in W,\ t \neq t_c \tag{5.7}$$

The value $\bar{I}_{tc}/I_{t_c,t_{c+1}}$ is then compared to the threshold ϵ_c. If this value exceeds the threshold, a shot cut is detected.

Fade detection: In order to have high performance in detection of fades and distinguish them from abrupt cuts, the joint entropy can be applied which is the average amount of information carried within a number of frames. Its value decreases during fades where weak amount of information is present. Only values of $H_{t,t+1}$ below a threshold T are examined. The end of fade out, t_e, is the place where joint entropy presents local minima. To search for the start of fade out, t_s, the criteria presented below is applied [57]

$$\frac{H_{t_s,t_s+1} - H_{t_{s-1},t_s}}{H_{t_s-1,t_s} - H_{t_s-2,t_s-1}} \geq \epsilon_f \tag{5.8}$$

where ϵ_f is a predefined threshold. A similar procedure is applied for detecting fade ins.

Another interesting work is performed in [69]. The novel characteristics of the algorithm is that it does not use predefined threshold and it is parameter free. The Jeffery divergence (Appendix C) is used to measure the distance between color

histograms. If H_i and H_j are histograms containing N beams, the Jeffery divergence between two histograms is defined by:

$$D_{col}(i,j) = \sum_{k=1}^{N} \left[(H_i(k) \left(\frac{H_i(k)}{m(k)} \right) + H_j(k) log \left(\frac{H_j(k)}{m(k)} \right)) \right] \tag{5.9}$$

where $m(k) = \frac{H_i(k)+H_j(k)}{2}$. Partitioning of video is defined as finding the partitioning that best describes the data assuming a model $y_m^\theta(t)$ with different parameters $\theta = (a_0, a_1, \sigma)$ in each segment. If we assume that variations from frame to frame is constant and evolution of colors is homogenous then the model can be defined as:

$$y_m^\theta(t) = a_1 t + a_0 + e_t \tag{5.10}$$

where e_t is an additive error term. a_0 and a_1 take into account static and dynamic characteristics of data respectively.

5.2.2 A Feature Based Algorithm for Shot Detection

In this subsection we review a successful approach suggested by R. Zabih et al [74]. This approach applies edges as a selected feature for solving the problem of shot boundary detection.

The algorithm takes two consecutive images I and I' and performs edge detection on images resulting in two binary images E and E'. Let ρ_{in} and ρ_{out} denote the number of edges that appear and disappear respectively in E' more than a fixed distance r from closest edges in E. ρ_{in} should be a high value during fade in and ρ_{out} should be a high value during fade out. The basic measure of dissimilarity which is called fraction of edge changes is defined as

$$\rho = \max(\rho_{in}, \rho_{out}) \tag{5.11}$$

Motion Compensation

We can apply motion compensation methods to handle the false positives that may occur because of motions within shots. The estimation algorithm must be efficient and robust in presence of multiple motions.

Computation of edge change fraction

Basically, edges are referred to the collection of pixels in an image which lie on the boundary between two regions [44].

Canny edge detection (Appendix A) is among the best edge detection algorithms available for images [45]. Therefore, it is employed in our proposed shot detection scheme. The images are first smoothed by a Gaussian filter of width σ. Next the gradient magnitude is computed which indicates that how fast the local intensities are changing. This magnitudes are compared with a predefined threshold of value τ to detect edges. Next step is dilation. Let $\bar{E}$ and $\bar{E}'$ be the dilated copies of E and E' which are created by replacing each edge pixel by a diamond whose height and width are $2r + 1$ pixels in length. To use the Manhattan distance between edges, dilatation with diamond is more suitable [74]. If we want to use Euclidian distance between edges, dilatation by a circle is more favorable.

Consider ρ_{in} which is a fraction of pixels in E' which are farther than distance r away from the edges in E. A black pixel $E'[x, y]$ exists when $E[x, y]$ is not a black pixel(since the black pixels in $\bar{E}$ are exactly those pixels within distance r of an edge in E [74].

$$\rho_{in} = 1 - \frac{\sum_{x,y} \bar{E}[x+\delta x, y+\delta y] E'[x,y]}{\sum_{x,y} E[x+\delta x, y+\delta y]} \tag{5.12}$$

Similarly, ρ_{out} is the fraction of edges in E which are farther than distance r away from edge pixels in E. The equation ρ_{out} is calculated by

$$\rho_{out} = 1 - \frac{\sum_{x,y} E[x+\delta x, y+\delta y] \bar{E}'[x,y]}{\sum_{x,y} E[x,y]} \tag{5.13}$$

The edge change fraction illustrated in equation (5.11) is the maximum of ρ_{in} and ρ_{out}.

5.3 Hierarchical Classification-Based Video Shot Detection Method

5.3.1 Overview of the Proposed Method

We propose a hierarchical classification scheme for detection and clustering the type of shot boundaries. The fraction of change in edge pixels between any two consecutive frames is computed and treated as a feature vector. Classification Based Adaptive search (CBAS) for motion compensation, described in chapter 3 is applied to handle camera or object motions. First, each frame is classified into two classes of Cut and Non-cut using information theoretic classification rule on computed feature vectors described in section (5.3.3). Second step uses the same classification technique to classify the non-cut frames into two classes of *Normal shot frame* or *Gradual transition frame*. K-nearest neighbor exchange algorithm is recalled to optimize the information theoretic classification rule. In third step

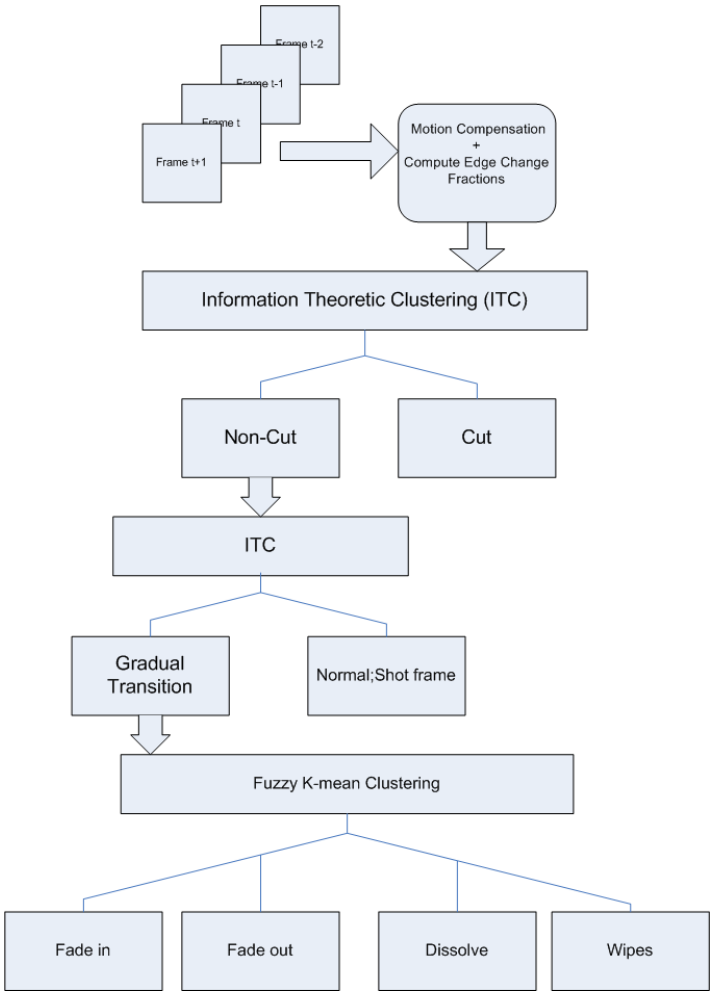

Figure 5.2: Schematic diagram of the proposed shot segmentation algorithm

of hierarchy, Fuzzy k-means clustering is applied on gradual transition frames to categorize them into four groups of *Fade in*, *Fade out*, *Dissolves* and *Wipes*. The schematic diagram of the proposed method is shown in Figure 5.2.

5.3.2 Computing the Edge Change Fraction

Edges are very informative. Amount of edge pixels and relative location of edges in any consecutive frames in video sequence can be related to the change in frame's visual content. As a result, high amount of change in location of edges informs us about significant change in frame's content or shot change. For computing the edge fraction, we follow the method proposed in [74]. The first step is extracting the edges from each image frame. Canny edge detection is applied for this purpose. Let δx and δy, computed by one of the motion estimation methods, be the translation necessary to align image frames I and I'. Let E and E' denote the edge frames. Next copies of E and E' are dilated by replacing each edge pixel by a diamond with radius r. The dilated images are denoted by $\bar{E}$ and $\bar{E}'$.

Consider ρ_{out}, the fraction of edge pixels in E which are farther r away from an edge pixel in E'. A black pixel $E[x, y]$ is an exiting pixel when $\bar{E}'[x, y]$ is not a black pixel. The equation for ρ_{out} is

$$\rho_{out} = 1 - \frac{\sum_{x,y} E[x + \delta x, y + \delta y]\bar{E}'[x, y]}{\sum_{x,y} E[x, y]} \tag{5.14}$$

Similarly, ρ_{in}, the fraction of edge pixels that are entering the frame, can be computed by

$$\rho_{in} = 1 - \frac{\sum_{x,y} \bar{E}[x + \delta x, y + \delta y]E'[x, y]}{\sum_{x,y} E[x + \delta x, y + \delta y]} \tag{5.15}$$

Therefore, the fraction of changed pixels can be represented by ρ which is defined as

$$\rho = \max(\rho_{in}, \rho_{out}) \tag{5.16}$$

Scene breaks or shot cuts are equivalent to the peaks in the edge change fraction ρ.

5.3.3 Classification Learning Rule

In this section we follow the information theoretic classification proposed in [73]. Consider $\rho \in \Omega_\rho$ as a selected feature vector where Ω_ρ is the space of feature vectors. Let us define the likelihood function:

$$L(\rho|C, C') = p(\rho|C) \cdot p(\rho|C') \tag{5.17}$$

where $C \in \Omega_C$ is the true class label and $C' \in \Omega_C$ is estimated class label where Ω_c denotes the space of class labels. In other words, this formula represents the likelihood of having ρ knowing that its true class label is C and estimated label is C'. This approach can be extended by defining the global transmission of class labels as the likelihood of transmitting the class label C to the estimated class label C' over the entire feature domain Ω_ρ [73]:

$$\begin{aligned} T(C, C') = \int_{\Omega_\rho} \sum_{C \in \Omega_C} \sum_{C' \in \Omega_C} D(C, C') \\ \cdot L(\rho|C, C') p(C) p(C') d\rho \end{aligned} \tag{5.18}$$

where $p(C)$ and $p(C')$ are the prior true class probability and the estimated class probability distributions, respectively. The function D is dissimilarity function and can be defined with respect to the application. Our strategy in classification is to minimize the error transmission, or maximize the true class label transmissions. For the sake of simplicity, we can neglect class-dependent similarities, i.e., we assume $D(C, C')$ is equal to 1 when $C = C'$ and 0 otherwise.

Both conditional probability density functions (PDFs) appearing in (5.17), can be estimated using nonparametric parzen window estimator [75]. It consists of placing a kernel function such as the well known Gaussian with width σ on each data sample. We use the Gaussian kernel because it provides simplification in the analysis which will be discussed later in this section. Therefore, the estimated conditional probability density functions can be expressed by:

$$\hat{p}(\rho|C) = \frac{1}{|S_C|} \sum_{\rho_i \in S_C} N(\rho - \rho_i, \sigma_i^2) \tag{5.19}$$

$$\hat{p}(\rho|C') = \frac{1}{|S_{C'}|} \sum_{\rho_j \in S_C} N(\rho - \rho_j, \sigma_j^2) \tag{5.20}$$

where the Gaussian kernels are defined by:

$$N(\rho - m, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left[-\frac{\|\rho - m\|^2}{2\sigma^2}\right] \tag{5.21}$$

and sets S_c and $S_{C'}$ contain data samples with class labels C and C' respectively, and $|S_c|$ and $|S_{C'}|$ are their size. Let M be the number of prototype data samples. Substituting estimated conditional probabilities into (5.18) and replacing $p(C)$ and $p(C')$ by $\frac{|S_C|}{M}$ and $\frac{|S_{C'}|}{M}$ respectively, we have

$$
\begin{aligned}
T(C, C') = \int_{\Omega_\rho} & \sum_{C \in \Omega_C} \sum_{C' \in \Omega_C} D(C, C') \\
& \cdot \frac{1}{M} \sum_{\rho_i \in S_C} N(\rho - \rho_i, \sigma_i^2) \\
& \cdot \frac{1}{M} \sum_{\rho_j \in S_C} N(\rho - \rho_j, \sigma_j^2) d\rho
\end{aligned}
\tag{5.22}
$$

We simplify above formula by using the fact that integration of two Gaussian random variable has still Gaussian distribution with a mean equal to the difference of means and a variance equal to summation of variances of the original Gaussian functions [76], i.e.

$$
\begin{aligned}
\int_{-\infty}^{+\infty} & \frac{1}{M} N(x - x_i, \sigma_i^2) \cdot \frac{1}{M} N(x - x_j, \sigma_j^2) dx \\
& = \frac{1}{M^2} N(x_i - x_j, \sigma_i^2 + \sigma_j^2)
\end{aligned}
\tag{5.23}
$$

Applying this property in (5.22), we have the following information theoretic learning rule

$$
\hat{C}' = arg \max_{C'} \sum_{C \in \Omega_C} \sum_{C' \in \Omega_C} D(C, C') V(S_C, S_{C'}) \tag{5.24}
$$

where $V(S_C, S_{C'})$ is defined as the *information potential* (IP) [71], which is closely related to Renyi's quadratic entropy [76], i.e.

$$
V(S_C, S_{C'}) = \frac{1}{M^2} \sum_{\rho_i \in S_C} \sum_{\rho_j \in S_{C'}} N(\rho_i - \rho_j, \sigma_i^2 + \sigma_j^2) \tag{5.25}
$$

IP is a positive decreasing function of distance between samples ρ_i and ρ_j, similar to the potential energy of physical particles [76].

To illustrate the relevancy of this concept to Renyi's entropy, consider the general Renyi's entropy formula [76]:

$$H_\alpha(y) = \frac{1}{1-\alpha} \log \int f(y)^\alpha dy \tag{5.26}$$

where $f(y)$ is the PDF of an event random variable y. If we set $\alpha = 2$ then Renyi's quadratic entropy is given by

$$H_2(y) = -\log(V) \tag{5.27}$$

where $V = \int f(y)^2 dy$. The expression presented in (5.25) is Parzen window estimation of information potential. Appendix B provides more details about the concept of IP.

5.3.4 Optimization of the Learning Rule

The learning rule (5.24) can be optimized by applying k-nearest neighbor exchange technique. This technique escapes from local minima and saves computations by labeling data groups instead of individual data samples [73]. The optimization method can be summarized by the following steps:

1. Assign a random class label for each of the learning data samples $\rho_{l=1}^{M_l}$ and select the initial group size $K = K_0$.

2. Create M_l groups by searching for the first K data samples in the vicinity of each data samples.

3. Omit repeated groups, resulting in p_l groups.

4. Repeat the following steps until there is no further improvement

   - For each group, change its label and report the improvement if any.
   - If there is any improvement, randomly permute the group indices.

5. If $K > 1$, divide K by 2 and go back to step 2.

6. End

5.4 Categorization of Gradual Transitions

Once a gradual transition is detected, the next problem is to label it as fade ins, fade outs, dissolves and wipes. During the fade in, since the frame is turning from black to a frame with some visual content, the number of edges that are entering the frame, i.e. ρ_{in}, is much higher than exiting edge pixels, ρ_{out}. On the other hand, during the fade out ρ_{out} is much higher than ρ_{in}. While in dissolve, there is an initial peak in ρ_{in} followed by a peak in ρ_{out}, hence, in some frame between these two peak frames, since ρ_{in} is decreasing and ρ_{out} is increasing, these two value cross each other. We define the following fraction as feature to cluster the data.

$$\alpha = \rho_{in}/\rho_{out} \tag{5.28}$$

We use Fuzzy k-mean clustering to cluster different gradual transitions[79], [80]. The Fuzzy k-means clustering seeks the minimum of heuristic global cost function [80].

$$J_{fuz}(M, C) = \sum_{i=1}^{k} \sum_{j=1}^{n} m_{ij}^{\phi} \|\alpha_j - \mu_i\|^2 \tag{5.29}$$

subject to:

$$\sum_{j=1}^{k} m_{ij} = 1 \quad i = 1, 2, ..., n \tag{5.30}$$

and

$$\sum_{i=1}^{n} m_{ij} > 1 \quad j = 1, 2, ..., k \tag{5.31}$$

where $m_{ij} \in [0\ 1]$ denotes the elements of membership matrix, M and μ_i is centroid of set of clusters denoted by C. The known number of patterns is denoted by n and desired number of clusters by k. ϕ is a free parameter chosen to adjust the blending of different clusters. For $\phi > 1$, the criterion allows each pattern to belong to multiple clusters. If ϕ is set to zero J_{fuz} is a sum of square differences criterion. Fuzziness performance index (FPI) estimates the degree of fuzziness generated by a specified number of classes and is defined as [81]

$$FPI = 1 - \frac{kF - 1}{k - 1} \tag{5.32}$$

where F is the partition coefficient:

$$F = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} m_{ij}^2 \tag{5.33}$$

The normalized classification entropy (NCE) which is also called the modified partition entropy (MPE) estimates the degree of disorganization created by a specified number of classes and is defined by [81]

$$NCE = \frac{H}{\log k} \tag{5.34}$$

where H is the entropy function [81]

Figure 5.3: Key frames from NASA documentary video sequence. The first row of images shows a dissolve occurring between two shots.

$$H = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} m_{ij}\log(m_{ij}) \tag{5.35}$$

5.5 Experimental Results

The results of the algorithm on TREC 2001 video archive which includes NASA's documentary video are provided in terms of precision and recall. The results are summarized in Table 5.1 and Table 5.2.

5.5.1 Experimental Data

We use the data in TREC-2001 video track [78], provided by national institute of standards and technology (NIST), which allows consistent comparison and evaluation between results of our proposed method and other systems. The video collections are mostly of documentary style videos and widely varying in age, production style and quality. The sequences contain a large variety of different boundaries. Figure 5.3 shows key frames from the video track.

5.5.2 Performance Evaluation

e use precision/recall score to evaluate our algorithm in our experiments. *Recall*, indicates that among all the transitions (cut or gradual) how many are detected by the system. *Precision* indicates that among all the transitions (cuts or gradual) detected by system, how many are true transitions.

Table 5.1: Performance analysis of proposed method in terms of recall, precision and F1

	Recall	Precision	F1
Cut	0.82	0.94	0.91
Gradual	0.78	0.63	0.69

Table 5.2: Validation results of Fuzzy K-mean Clustering on gradual shot changes

$\phi =$	1.5	2	2.5	3	3.5
FPI	0.141	0.330	0.520	0.633	0.725
NCE	0.134	0.348	0.520	0.647	0.738

A good detector must have both high precision and recall. The commonly used metric, F1, combines both precision and recall to evaluate the performance of the shot detection algorithm [64]. F1 is high when both precision and recall are high.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.36}$$

Chapter 6

Conclusions and Recommendation for Future Research

In this thesis, a statistical pattern classification scheme for the design of adaptive motion estimation algorithm (CBAS) has been proposed. Simulation results demonstrate that the proposed technique outperforms conventional fast block matching methods in terms of higher PSNR and less computational complexity. In summary, an intelligent encoder should apply adaptive motion estimation techniques instead of relying on fixed patterns. The ideas of machine learning and pattern recognition can be applied for the design of adaptive intelligent motion estimation techniques. The PSNR and computation gain performance results illustrate that CBAS has better PSNR performance and less computation than other algorithms, including the state-of-the-art DS algorithm. The CBAS algorithm is almost 12.75 times faster than ES while the PSNR level closely follows that of the ES with slight degradation less than $0.10 - 0.13$ dB. The algorithm is able to maintain rather constant PSNR performance.

In addition, the application of this algorithm was examined on our proposed hierarchical video shot boundary detection method. The method uses classification based adaptive search (CBAS) to cope with camera and object movements within shots in order to reduce the rate of false detection of shot boundaries. Infrastructure of shot detection algorithm is based on information theoretic classification (ITC) rule. ITC was used with the goal of improving the classification results because second order statistics are not sufficient to distinguish nonlinearly separable classes. Fuzzy K-means clustering is applied in order to categorize gradual shot transitions into different groups. Experiments show that the method can be improved by applying CBAS method for compensation of motions. The algorithm has excellent performance in terms of precision and recall on TREC 2001 video track.

The main focus of this thesis which is based on using Bayesian classification for adaptive video motion estimation. It is possible to improve the algorithm. Bayesian classifier applies Parzen window to estimate conditional probability functions. If the kernel widths are not chosen properly, we might easily overfit the data, resulting in a classification scheme not representative of the true classes. The focus can be put on designing adaptive kernel size scheme with respect to a suitable criterion, for example the average energy of error in compensated image compared to a true frame.

Video shot segmentation method suggested in chapter 5 has promising opportunity to be extended. We conclude our investigations by introducing major problems that can be investigated in video shot segmentation. They are

1. Sensitivity to illumination: Significant change in illumination from a frame of similar shot have caused some algorithms to handel poorly these situations. Appropriate selection of feature can fairly solve the problem. Histogram and

information theoretic approaches show high robustness to illumination within shots.

2. Sensitivity to motions within a shot: Large and sudden changes in frames of one shot can also fool the algorithm to erroneously detect a shot break. Motion estimation vectors are suitable features to handel this problem.

3. Fixed Threshold: High fixed threshold may skip many true shot breaks. on the other hand, very low threshold may cause false positives. The solution could be to eliminate the threshold. Clustering of shots with regard to a predefined dissimilarity measure could be a solution.

In fact, the proposed method allows multiple features to be used simulatively to improve the performance of the algorithm. Possible extension of the algorithm can also include audio features in order to improve performance.

Appendix A

Canny Edge Detection

The Canny edge detection is known to be an optimal edge detection technique [45]. First, Canny edge detector smooths the image to eliminate noise. A Gaussian filter is used exclusively in the Canny algorithm for noise filtering because it can be computed using a simple mask. The larger the width of the mask, the less sensitive is the Gaussian mask to noise. The error in localization of edges increases as Gaussian width is increased. A sample Gaussian filter is given below

Then, to find regions that have more probability for existence of edges, it computes the gradient over pixel intensities to find regions with high spatial derivatives. The gradient of on image $f(x, y)$ at location (x, y) is defined as the vector

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{A.1}$$

2-D spatial gradient measurement can be done by Sobel operator [45]. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient

in the x-direction and the other in the y-direction. The approximate absolute gradient magnitude which is called *edge strength* at each point can be found using the following relation

$$|G| = |G_x| + |G_y| \tag{A.2}$$

The next step is to find the direction of edges applying a simple formula

$$\theta = \arctan(\frac{G_y}{G_x}) \tag{A.3}$$

After this stage, the edge direction can be traced on image. Looking at a single pixel in image, there are only four possibilities for directions on the surrounding pixels: 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), and 135 degrees (along the negative diagonal). So, the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to approximate value computed by (A.3). The algorithm then searches along the edge directions and suppresses the pixels that are not at the maximum (non-maximum suppression) which results in thin edges.

In the final step the method uses two thresholds, $T1$ and $T2$, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. Using two thresholds causes the algorithm to be less sensitive to noise since the intensity value of edge pixels can fall bellow threshold due to noise.

Appendix B

Information Potential

Information theory is a powerful tool which provides a basis for designing optimum and reliable communication systems. However, applying information theory, demands a priori knowledge about the distribution of data and the mapping which produces the outputs of the system.

In fact, learning is related to the extraction of information from data [46]. Shannon's entropy is not easy to implement for learning from examples, however, Reneyi's entropy definition can be generally integrated with probability estimation methods such as Parzen windowing to provide learning scheme which can be applied in many applications. The concept of information potential (IP) which is described in this section is related to the degree of interaction between information particles extracted from the data. IP can be very well illustrated by refereing to one of its applications in research works [76], which investigates error-entropy minimization in adaptive system training. The method is described in the following paragraphs.

Mean square error (MSE) is a popular criterion for training adaptive systems including artificial neural networks; mainly because of its analytical tractability

and the fact that real life random phenomena can be modeled sufficiently by second order statistics. It has become evident that linearity and Gaussianity assumptions may not be appropriate when dealing with nonlinear systems [76]. The entropy criterion can serve as an alternative for MSE in supervised adaptation [49] while it is particularly suitable for dynamic modeling [50]. The goal of dynamic modeling is to acquire a nonlinear dynamic system which can produce the given input-output mapping.

Estimation of the probability distribution function (pdf) of a random variable, is necessary for the evaluation of the entropy. Nonparametric methods such as Parzen windowing do not assume any particular format for distribution function and introduce more generality which is desirable characteristic of nonparametric methods. In Parzen windowing, the pdf is approximated by the sum of shifted versions of a kerenl function such as Gaussian or Laplacian among others. In [48], it has been proved that error entropy minimization is equivalent to minimizing the error in pdf matching between the actual and desired output of a system.

B.1 Error Entropy Minimization and Probability Density Matching

Let x be the input of the system, d and y are respectively the desired and actual outputs of the system. Then the error between desired and actual output would be $e = d - y$. Hence the pdf of error can be written [48], [49]

$$f_{\epsilon,\omega}(e) = f_{y|x,\omega}(d - e|x) \tag{B.1}$$

where the subscript ω indicates dependence on the weights of the adaptive system. Minimizing Renyi's error entropy with respect to the parameter ω is given by

$$\begin{aligned} \min_{\omega} & \frac{1}{1-\alpha} \log \int f_{\epsilon,\omega}^{\alpha}(e)de \\ &= \frac{1}{1-\alpha} \log \int f_{y|x,\omega}^{\alpha}(d-e|x)de \\ &= \frac{1}{1-\alpha} \log \int -f_{y|x,\omega}^{\alpha}(y|x)dy \end{aligned} \tag{B.2}$$

Since multiplying the cost function by a factor which is independent of the weights of the adaptive system will not make a change to the optimization problem, the integral of the power-α of the pdf of the input can be introduced to obtain an equivalent optimization problem

$$\begin{aligned} &\equiv \int_{\omega} f_{y|x,\omega}^{\alpha}(y|x)dy \cdot \int f_{x}^{\alpha}(x)dx \\ &= \int \int f_{xy,\omega}^{\alpha}(x,y)dxdy \\ &\equiv \int \int f_{xy,\omega}^{\alpha}(x,y)dxdy \cdot \int \int f_{xd}^{1-\alpha}(x,y)dxdy \\ &= \int_{\omega} \int \int f_{xy,\omega}(x,y) \left(\frac{f_{xd}(x,y)}{f_{xy,\omega}(x,y)} \right)^{1-\alpha} dxdy \end{aligned} \tag{B.3}$$

which can be recognized as the Csiszar distance with convex chosen to be $(.)^{1-\alpha}$. In general the Csiszar distance between two densities $p(x)$ and $q(x)$ is given by [51]

$$D_C(p;q) = \int q(x) f\left(\frac{p(x)}{q(x)} \right) dx \tag{B.4}$$

where f is convex. For Shannon's entropy, the distance measure in (B.3) reduces to the Kullback-Leibler divergence [49]

$$\begin{aligned} \lim_{\alpha\to 1}\frac{1}{\alpha-1}\ log\int\int f_{xy,\omega}(x,y)\left(\frac{f_{xy,\omega}(x,y)}{f_{xd}(x,y)}\right)^{1-\alpha}dxdy \\ = \int\int f_{xy,\omega}(x,y)log\left(\frac{f_{xy}(x,y)}{f_{x,\omega}(x,y)}\right)^{1-\alpha}dxdy \end{aligned} \tag{B.5}$$

In practice the probability distribution function of the random process which describes our input data is usually unknown a priori [46][47]. Parzen windowing estimation with Gaussian kernel provides a number of advantages. The Gaussian function is continuously differentiable [48]. The Gaussian function also provides a computational simplification in the learning algorithm design. In [48], it is proven that the global minimum of the entropy is still a minimum of the nonparametric estimated entropy for both Shannon's and Renyi's definition when Parzen windowing estimation method is applied. The Parzen estimator of the error pdf is given by

$$\hat{f}_e(\xi) = \frac{1}{N}\sum_{i=1}^{N}\kappa(\xi - e_i, \sigma^2) \tag{B.6}$$

where κ denotes the Gaussian kernel and σ^2 is the variance for simplicity. We investigate a much simpler case which is the nonparametric estimation of Renyi's quadratic entropy ($\alpha = 2$). Using Renyi's quadratic definition and Parzen windowing estimator with Gaussian kernels we obtain

$$H_2 = -log\int\left(\frac{1}{N}\sum_{i=1}^{N}\kappa(\xi - e_i, \sigma^2)\right)^2 d\xi = -logV(e) \tag{B.7}$$

where $V(e)$ is called *information potential* [48] which illustrates the relation between information potential of a given set of sample errors for an arbitrary kernel size. For the case of Gaussian kernels, it can be computed as

$$V(e) = \frac{1}{N^2} \sum_{j}^{N} \sum_{i}^{N} \kappa(e_j - e_i, 2\sigma^2) \tag{B.8}$$

Appendix C

Non-Parametric Measures

The kullback-Leibler divergence (KL) is defined as a measure of extent to which two PDFs, $p(x)$ and $\tilde{p}(x)$ agree. It is defined as [52]

$$L = -\int p(x) ln \frac{\tilde{p}(x)}{p(x)} dx \tag{C.1}$$

It can be shown that $L \geq 0$. For two discrete distribution, the integration becomes the summation over all the bins. The Jeffery divergence (JD) is a symmetric version of KL with repected to $p(x)$ and $\tilde{p}(x)$, given by

$$JD = \int p(x) ln \frac{\tilde{p}(x)}{p(x)} dx + \int \tilde{p}(x) ln \frac{p(x)}{\tilde{p}(x)} dx \tag{C.2}$$

Bibliography

[1] F. Dufaux and F. Moscheni,"Motion estimation techniques for digital tv: A review and a new contribution," in Proc. IEEE on Engineering in Medicine and Biology, vol. 83, no. 6, 1995, pp. 858-876.

[2] A. Netravali and JD Robbins,"Motion compensated television coding: Part 1", Bell System Technical Journal, 58:631–670, 1979.

[3] T.Koga, K.Iinuma, A.Hirano, Y.Iijima and T.Ishiguro, "Motion compensated interframe coding for video conferencing", in Proc. NTC81, pp. C.9.6.1-9.6.5, New Orleans, LA, Nov. 1981.

[4] B. K. P. Horn and B. G. Schunck, "Determining optical flow," Artitificial Intelligence, vol. 17, pp. 185-203, 1981.

[5] D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," International Journal of Computation and Vision, vol. 5, pp. 77-104, 1990.

[6] Bruce D. Lucas, Takeo Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", Proceedings of Imaging Understanding Workshop, pp 121-130, 1981.

[7] H.H. Nagel: "Displacement vectors derived from second-order intensity variations in image sequences", Comp. Graph. Image Processing 21, 1983

[8] H.-H. Nagel. "Constraints for the estimation of displacement vector fields from image sequences" In Proc. Eighth International Joint Conference on Artificial Intelligence, colume 2, pages 945-951, Karlsruhe, Germany, August 1983.

[9] C.Cafforio and F.Rocca "The differential method for motion estimation" in Image Sequence Processing and Dynamic Scene Analysis, T.S.Huang, Ed., New York, Springer-Verlag, 1983, pp. 104-124.

[10] D.R. Walker and K.R. Rao, Improved pel-recursive motion compensation, IEEE Trans. Commun. COM-32, 11281134 (1984).

[11] P. Anandan, "A Unified Perspective on Computational Techniques for the Measurement of Visual Motion". In Proceedings of the 1 International Conference on Computer Vision, 1987.

[12] M. Bierling, "Displacement estimation by hierarchical block matching", in: SPIE Visual Communications and Image Processing, Vol. 1001, 1998, pp. 942–951.

[13] V. Seferidis and M. Ghanbari, "General approach to block matching motion estimation," Optical Engineering, No. 7, pp. 1464–1474, Jul. 1993.

[14] P. Kuhn, et al., Complexity and PSNR-comparison of several fast motion estimation algorithms for MPEG-4, in Proc. SPIE Applications of Digital Image Processing XXI, July 1998, pp. 486-499.

[15] Jar-Ferr Yang; Shih-Cheng Chang; Chin-Yun Chen, "Computation reduction for motion search in low rate video coders ", IEEE Trans. Circuits and systems for video tech., Volume 12, Issue 10, Oct. 2002 Page(s):948 - 951.

[16] L. Koskinen, A. Paasio; Halonen, K.A.I., "Motion estimation computational complexity reduction with CNN shape segmentation ", IEEE Trans. Circuits and systems for video tech., Volume 15, Issue 6, June 2005 Page(s):771 - 777.

[17] I-Ming Pao; Ming-Ting Sun, "Computation reduction for discrete cosine transform", IEEE Int. Symposium on Circuite and Systems 1998, Volume 4, 31 May-3 June 1998 Page(s):285 - 288 vol.4.

[18] X. Zhou, Z. H. Yu, and S. Y. Yu, Method for detecting all-zero DCT coefficients ahead of discrete cosine transformation and quantization, Electron. Lett., vol. 34, no. 19, Sept. 1998.

[19] M. J. Chen, L. G. Chen, and T. D. Chiueh, One-dimensional full search motion estimation algorithm for video coding, IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 504-509, Oct. 1994.

[20] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *in Proc. Nat. Telecommun. Conf.,* New Orleans, LA, Nov. 1981, pp. G5.3.1-G5.3.5.

[21] R. Li, B. Zeng, and M. L. Liou, A new three-step search algorithm for block motion estimation, *IEEE Trans. Circuits Syst. Video Technol.,* vol. 4, no. 8, pp. 438442, Aug. 1994. L. M. Po and W. C.

[22] Ma, A novel four-step search algorithm for fast block motion estimation, *IEEE Trans. Circuits Syst. Video Technol.,* vol. 6, no. 6, pp. 313317, Jun. 1996.

[23] S. Zhu and K.-K. Ma, "A new diomond search algorithm for fast block matching motion estimation,"*IEEE Trans. Image Process.,*vol.9, no. 2, pp. 287-290, Feb. 2000

[24] C.-H. Cheung,L.-M. Po," A Novel small-cross-diamond search algorithm for fast video coding and videoconferencing applications,"*in Proc. IEEE Int. Conf. Image Processing*, 2002, pp. 681-684.

[25] S. Zhu and K.-K. Ma, A new star search algorithm for fast blockmatching motion estimation, in Proc. *Workshop on Very Low Bitrate Coding (VLBV),* Oct. 89, 1998, pp. 173176.

[26] M. Ghanbari, The cross-search algorithm for motion estimation, IEEE Trans. Commun., vol. 38, pp. 950-953, 1990.

[27] L. Lurng-Kuo, E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding", IEEE Trans. Circuits Syst. Video Technol. 6 (4) (1996) 419–422.

[28] Sheu-Chih Cheng; Hsueh-Min Hang;"A comparison of block-matching algorithms mapped to systolic-array implementation ", Cir. and Sys for Video Tech., Volume 7, Issue 5, Oct. 1997 Page(s):741 - 757.

[29] S.-Y. Huang, C.-Y Cho,and J.-S. Wang,"Adaptive fast block matching algorithm by switching search patterns for sequences with wide-range Motion Content,"*IEEE Trans. Circuit Syst. Video Technol.*, vol.15, no.11, pp1373-1384, Nov. 2005.

[30] Y. Nie and K.-K. Ma, "Adaptive Rood Pattern Search for Fast Block Matching Motion Estimation,"*IEEE Trans. Image Process.,*vol.11, no. 12, pp. 1442-1449, Dec. 2002

[31] Libo Yang, Keman Yu, Jiang Li, and Shipeng Li. "An Effective Variable Block-Size Early Termination Algorithm for H.264 Video Coding". IEEE Trans. on Circuits and Systems for Video Technology, VOL. 15, NO. 6, JUNE 2005. P784-788.

[32] Seong-Ju Kim, Jong-Hak Ahn, Changhoon Yim, "Adaptive motion estimation algorithm for MPEG-4 video coding", Seventh Int. Symp. on signal Proc. and its applications, Volume 2, 1-4 July 2003 Page(s):141 - 144 vol.2.

[33] Kai Sun, "Adaptive motion estimation based on statistical sum of absolute difference", ICIP 1998, 4-7 Oct. 1998 Page(s):601 - 604 vol.3

[34] Wang Hui; Mao Zhigang, "An adaptive motion estimation algorithm based on evolution strategies with correlated mutations ", Image Processing, 2004. ICIP '04. 2004 International Conference on Volume 3, 24-27 Oct. 2004 Page(s):1469 - 1472.

[35] V. Solo, "Smoothing estimation of stochastic processes", University of Wisconsin-Madison, Mathematics Research Center, 1980.

[36] Schalkoff, Robert. Pattern Recognition. New York: John Wiley, 1992.

[37] M.E. Jernigan, P.W. Fieguth, "Syde 372 course notes: introduction to pattern recognition", University of Waterloo.

[38] Phan, F., and Micheli-Tzanakou, E. Supervised and unsupervised pattern recognition: feature extraction and computational intelligence. CRC Press, Boca Raton, 2000.

[39] W. S. Meisel,"Computer-Oriented Approaches to Pattern Recognition", Academic Press, 1972.

[40] L. F. Pau C. H. Chen and P.S.P.Wang. Handbook of Pattern Recognition and Computer Vision.World Scientific, 1993.

[41] R. Kasturi and R. Jain, editors. Computer Vision: Principles. IEEE Computer Society Press, 1991.

[42] A. Murat Tekalp. Digital Video Processing. Prentice-Hall, 1995.

[43] Al Bovic,"Handbook of Image and Video Processing",San Diego: Academic Press(2000)

[44] R. C. Gonzalez, R. E. Woods, "Digital Image Processing" , second edition, 2002.

[45] B. Green,"Canny Edge Detection Tutorial", from web resource. www.pages.drexel.edu/ weg22/cantut.html

[46] J. C. Principe and D. Xu, "Information-theoretic learning using renyi's quaratic entropy," in First International Workshop on Independent Component Analysis and signal Separation (ICA99),J.F. Cardoso, Ch. Jutten, and Ph. loubaton, Eds., Aussois, France, 11-15 January 1999, pp. 407-412.

[47] Erdogmus, D., Principe, J. C., Comparison of Entropy and Mean Square Error Criteria in Adaptive System Training Using Higher Order Statistics, ICA 2000, June 2000.

[48] Deniz Erdogmus, Jose C. Principe, "An Error-Entropy Minimization Algorithm for Supervised Training of Nonlinear Adaptive Systems " Trans. on Signal Processing, Vol. 50, No. 7, pp. 1780-1786, July 2002.

[49] Deniz Erdogmus, Jose C. Principe, "Generalized Information Potential Criterion for Adaptive System Training," Trans. on Neural Networks, Vol. 13, No. 5, pp. 1035-1044, Sept. 2002.

[50] S. Haykin and J. C. Principe, Dynamic modeling with neural networks, IEEE Signal Processing Mag., vol. 15, p. 66, Mar. 1998.

[51] I. Csiszr and J. Krner, Information Theory: Coding Theorems for Discrete Memoryless Systems. New York: Academic, 1981.

[52] A. Shahrokni, T. Drummond and P. Fua, Nonparametric measures: online resource. http://homepages.inf.ed.ac.uk/rbf/CVonline

[53] Online Tutorial on nonparametric density estimation methods. www.cs.pitt.edu/ milos/courses/cs2750-Spring02/

[54] Wikipedia, the free encyclopedia. www.wipkidia.com

[55] Porter, S.V.; Mirmehdi, M.; Thomas, B.T.;"Video cut detection using frequency domain correlation", 15th International Conference on Pattern Recognition, 2000. Proceedings,Volume 3, 3-7 Sept. 2000 Page(s):409 - 412 vol.3 Digital Object Identifier 10.1109/ICPR.2000.903571

[56] M. J. Swain and D. H. Ballard,"Color indexing",International Journal of Computer Vision, 7(1):11-32, 1991.

[57] Z. Cernekova,, C. Nikou and I. Pitas. Entropy metrics used for video summarization. In Proceedings of the Inernational Spring Conference on Computer Graphics (ISCCG 2002), Budmerice, Slovakia, 2002

[58] Y. Yusoff, W. Christmas and J. Kittler, "Video shot cut detection using adaptive thresholding", Proc. British Machine Vision Conference, 2000.

[59] A. Miene, A. Dammeyer, T. Hermes, and O. Herzog, "Advanced and adaptive shot boundary detection", ECDL WS generalized Documents, 2001.

[60] A. Hanjalic, "Shot-boundary detection: Unraveled and resolved?" IEEE Transactions on Circuits and Systems for Video Technology, 12(2):90–105, February 2002.

[61] C. Cotsaces, N. Nikolaidis and L. Pitas, "Video Shot Detection and Condensed Representation: A Review", Signal Processing Magazine, Vol. 23, Mar 2006.

[62] A. Bovik, "Handbook of Image and Video Processing", Academic Press, 2000.

[63] I. Koprinska, S. Carrato, "Temporal video segmentation: a survey", Signal Processing: Image Communication, Vol. 16, pp. 477–500, 2001.

[64] Y. Qi, A. Hauptmann, T. Liu, "Supervised Classification for Video Shot Segmentation", IEEE International Conference on Multimedia and Expo (ICME),2003.

[65] B. Gunsel, A.Ferman, A.M. Tekalp,"Temporal video segmentation using unsupervised clustering and semantic object tracking", Journal of Electronic Imaging 1998, pp. 592-604.

[66] Y. Y. Boykov, M-P. Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images",Proceedings of International Conference on Computer Vision, Vancouver, Canada, July 2001.

[67] T. Butz, J-P. Thiran, "Shot Coundary Derection with Mutual Information", Image Processing, 2001. Proceedings. 2001 International Conference on Volume 3, 7-10 Oct. 2001 Page(s):422 - 425 vol.3.

[68] Z. Cernekova, C. Nikou, I. Pitas, "Information Theory-Based Shot Cut/Fade Detection and Video Summarization" , IEEE Trans. on Circuit and Systems for Video Technology, Vol. 16, NO. 1, January 2006.

[69] B. Janvier, E. Bruno, T. Pun, "Information Theoretic Temporal Segmentation of Video and Applications: Multiscale Keyframes Selection and Shot Boundaries Detection", 2005 Kluwer Academic Publishers.

[70] Y. Yusoff, J. Kittler, W. J. Christmas, "Combining Multiple Experts for Classifying Shot Changes in Video Sequences",International Conference on Multimedia Computing and Systems, 1999.

[71] E. Gokcay and J. C. Principe. Information theoretic clustering. IEEE Patt. Anal. Mach. Int., 24(2):158–171, 2002.

[72] T. Butz, J-P. Thiran, "From Error Probability to Information Theoretic Signal Processing", 2003.

[73] C. Archambeau , T. Butz, V. Popovici, M. Verleysen, J. P. Thiran, "Supervised Nonparametric Information Theoretic Classification", International Conference on Pattern Recognition (ICPR) 2004.

[74] R. Zabih, J. Miller, K. Mai,"A Feature-Based Algorithm for Detecting and Classifying Scene Breaks"(1995), Multimedia Systems.

[75] E. Parzen. "On measures of entropy and information" In Proc. of 4th berkeley symp. on prob. Math. Stat., 33:1065-1076, 1962.

[76] D. Erdogmus, J.C. Principe, "An Error-Entropy Minimization Algorithm for Supervised Training of Nonlinear Adaptive Systems", IEEE Transactions On Signal Processing, Vol. 50, NO. 7. July 2002.

[77] H. Zhang, A. Kankanhalli, and S. Smoliar,"Automatic Partitioning of Full Motion Video", Multimedia Systems, 1:10-28, 1993.

[78] A.F. Smeaton, P. Over, R. Taban, "The TREC 2001 Video Track Framework", Proceedings of the Tenth Text Retrieval Conference (TREC-2001), Gaithersburg, Maryland, Nov. 13-16, 2001.

[79] R. O. Duda, P. E. Hart, D. G. Stork, "Pattern Classification", Second Edition, Wiley Interscience Publication, NewYork, 2001.

[80] J.C. Bezdek (1981) Pattern recognition with fuzzy objective function algorithms, Plenum Press, New York.

[81] Roubens, M., 1982. Fuzzy clustering algorithms and their cluster validity. European Journal of Operational Research 10, 294-301.