IT 10 061

Examensarbete 30 hp
November 2010

Analysis Application for H.264 Video Encoding

Ying Wang

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Analysis Application for H.264 Video Encoding

Ying Wang

A video analysis application ERANA264(Ericsson Research h.264 video ANalysis Application) is developed in this project. Erana264 is a tool that analyzes H.264 encoded video bitstreams, extracts the encoding information and parameters, analyzes them in different stages and displays the results in a user friendly way. The intention is that such an application would be used during development and testing of video codecs. The work is implemented on top of existing H.264 encoder/decoder source code (C/C++) developed at Ericsson Research.

Erana264 consists of three layers.
The first layer is the H.264 decoder previously developed in Ericsson Research. By using the decoder APIs, the information is extracted from the bitstream and is sent to the higher layers.
The second layer visualizes the different decoding stages, uses overlay to display some macro block and picture level information and provides a set of play back functions.
The third layer analyzes and presents the statistics of prominent parameters in video compression process, such as video quality measurements, motion vector distribution, picture bit distribution etc.

Key words: H.264, Video compression, Bitstream analysis, Video encoding

Handledare: Zhuangfei Wu and Clinton Priddle
Ämnesgranskare: Cris Luengo
Examinator: Anders Jansson
IT10061
Tryckt av: Reprocentralen ITC

Acknowledgements

Fist of all, I am heartily thankful to my supervisors, Fred Wu and Clinton Priddle, whose encouragement, supervision and support from the preliminary to the concluding level enabled me to develop an understanding of the subject. I am grateful to Rickard Sjöberg, not only for his previous delicated work on Ericsson Research's internal H.264 CODEC which is a essential basis of this project, but also for his continuous guidance and support, throughout the whole period of the thesis.

It is an honor for me to have prof. Cris Luengo at Uppsala University as my reviewer, many thanks for his trust and suggestions on the thesis report.

I owe my deepest gratitude to Per Fröjdh, the manager of the visual group for offering me the work opportunity and kindly help. I would also like to thank all the other members in the multimedia technology group, specially Per Boussard and Torbjörn Einarsson, for their valuable suggestions and contributing to a nice working environment.

And finally never enough thanks to my parents and friends for their encouragement and support throughout all my studies at University.

Ying
June, 2010

Contents

Chapter 1

Introduction

Despite the rapidly improving computing and communications capabilities, the demand of efficient and high quality digital video is still drastically increasing in video conferencing, high definition television, online stream video and many other areas. Since the transmission or storage of each single bit is an increase in cost, researchers from companies and universities put a lot of efforts on developing more efficient video compression technology, to get high quality video while keeping coded bitstream size small. Video compression has played an important role in the areas of telecommunication and multimedia systems. The primary goal of video compression is to reduce the amount of information that has to be transmitted or stored without reducing its subjective quality. With the intention of helping to develop and optimize the latest video compression CODECs, a video analysis application is developed in this project.

1.1 Purpose

The new video coding standard H.264 (Advanced Video Coding, AVC) has become the leading standard for new video services such as HDTV, Bluray and mobile TV. Although decoding is standardized, the encoding process is fully open as long as the coded bitstream conforms to the decoder. How a video sequence is coded varies among different encoders. A state-of-the art HDTV encoder, for example, will use that freedom to spend fewer bits on areas that tolerate coding distortion and spend more bits on critical parts.

H.264 standard includes many processes and there can be huge performance difference among standards-compliant encoders and decoders. In order to achieve good performance, careful design and careful choice of coding parameters are required. A video analysis application ERANA264(Ericsson Research h.264 video ANalysis Application) is developed in this project,to help to optimize the CODEC and choose the proper parameters. Erana264 is a tool that analyzes H.264 encoded video bitstreams, extracts the encoding information and parameters, analyzes this information and displays the results in a visual and user friendly way. The intention is that such an application would be used during development and testing of video codecs. The work is implemented on top of existing H.264 encoder/decoder source code (C/C++) developed at Ericsson Research.

1.2 The structure of this thesis

Chapter 2 introduces the basic concepts about video compression, H.264 standard and how video analysis application works.
Chapter 3 discusses about the system requirements, development process and give an overview of this video analyze application.
Chapter 4 introduces the main features and most important functions in this application.
Chapter 5 analyzes some representive video sequences in different perspectives.
Chapter 6 concludes this thesis and provides some ideas about possible future works.

Chapter 2

Background

2.1 Video Compression

A digital video consists of a series of frames, each frame is represented by a two dimensional array. A video sequence contains spatial and temporal redundancy. The redundancy between frames is temporal redundancy, and the redundancy inside a frame is spatial redundancy. An uncompressed raw digital video is extremely inefficient since it contains both spatial and temporal redundancy. It costs a huge amount of space to store and time to transmit. In most situations, videos are compressed by means of removing redundancy mentioned above. Video compression is the process of reducing the quantity of data used to represent a video sequence by using the combination methods of spatial image compression and temporal motion compensation. Most video compression is lossy, and there is a trade off between video quality, bit rates and complexity. Video compression consists of a pair of systems, an encoder and a decoder. Video sources are compressed(encoded) in the source and the bitstream is sent to the destination. At the receiver side, the bitstream is uncompressed(decoded) to get reconstructed video output. The encoder/decoder pair is usually called CODEC together(Figure 2.1).

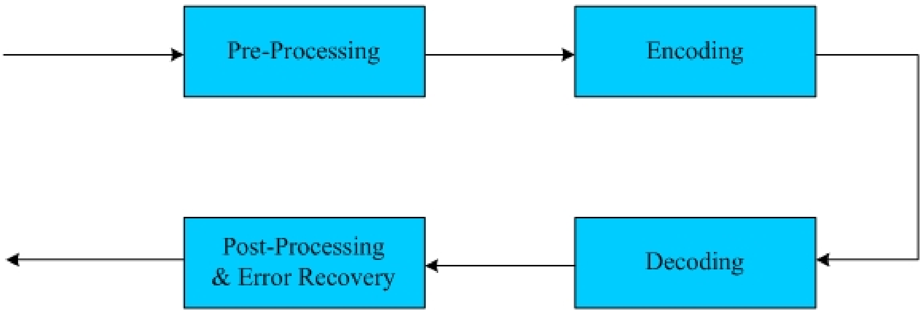


Figure 2.1: CODEC

2.1.1 Temporal Model

In an uncompressed video sequence, there is always a lot of redundancy information between frames. A temporal model is typically used to reduce the redundancy by predicting the difference between the coded frames and the frame being coded. In this way, by sending the residual(difference) frames instead of the whole frame, temporal redundancy can be greatly reduced. The better the prediction is, the more redundancy information can be eliminated.

Motion compensation is introduced for the purpose of reducing temporal redundancy. H.264 standard uses a block-based motion estimation and compensation algorithm. Every frame in the sequence is divided to 16×16 pixel blocks called macro blocks. Macro blocks are the basic unit in the motion compensation algorithm in H.264 standard and many other standards. Motion estimation algorithm typically tries to find the best matching 16×16 region inside a reference frame to the current macro block. Once a region is found, the algorithm calculates the distance from the current macro block to the reference region and gets a motion vector that represents the distance. The selected "Best Match" region will be subtracted from the current macro block to get a residual macro block, the residual will be encoded and transmitted together with the motion vector. The receiver uses the residual and motion vector to reconstruct the original macro block.

2.1.2 Image Model

Image Model is used to decorrelate image data and convert it in a form that can be efficiently compressed. Image models usually have three parts: transformation, quantization and reordering. The purpose of transformation is to decorrelate and compact the data, quantization is used to reduce the precision of the data and through reordering we can put all significant values together.

Predictive Image Coding

Motion compensation is an example of predictive image coding. The encoder makes a prediction based on the previous frame and subtracts the prediction from the current image to get a residual image. The residual image contains less information and can be compressed in less bits. Another type of prediction is prediction based on previous transmitted samples in the same frame.

Transform Coding

The purpose of transform coding is to convert the residual data from the spatial domain to frequency domain. There are many transforms that can be used in image and video compression, and all of them can be put into two types: block-based and image-based. The Karhunen-Loeve Transform(KLT)[3], Singular Value Decomposition(SVD)[4] and the Discrete Cosine Transform(DCT)[5] are block based. The Discrete Wavelet Transform (DWT) is image based. Image-based transformation has better performance but meanwhile has higher requirements for memory(it transforms the whole image as a single unit) and does not cooperate well with block based motion compensation. In the H.264 standard, block-based transforms are used and it will be discussed in the later chapters.

Quantization

Quantization is the process of mapping a continuous range of values to a reduced range of values. The input to a quantizer is the original data, and the output is a number among a finite range of values. Obviously, this process is lossy and can not be reversed. A good quantizer is a quantizer which represents the original data with minimum loss and distortion. There are two kinds of quantization: Scalar Quantization and Vector Quantization.

Scalar quantization A scalar quantization maps an input signal with a range of values X to a quantized signal with a reduced range of values Y, every input sample is processed separately. Scalar Quantization is used in H.264 standard. A general example of a uniform quantizer:

$$\begin{aligned} FQ &= round(\frac{X}{QP}) && (2.1) \\ Y &= FQ \cdot QP && (2.2) \end{aligned}$$

QP (Quantization Parameter) is the step size. [1] Quantizers with higher QP values have lower scale of the output, it can reduce the size of data in the cost of lower video quality. Figure 2.2 and Figure 2.3 are compressed from the same frame in bus sequence but quantized by different QP value.

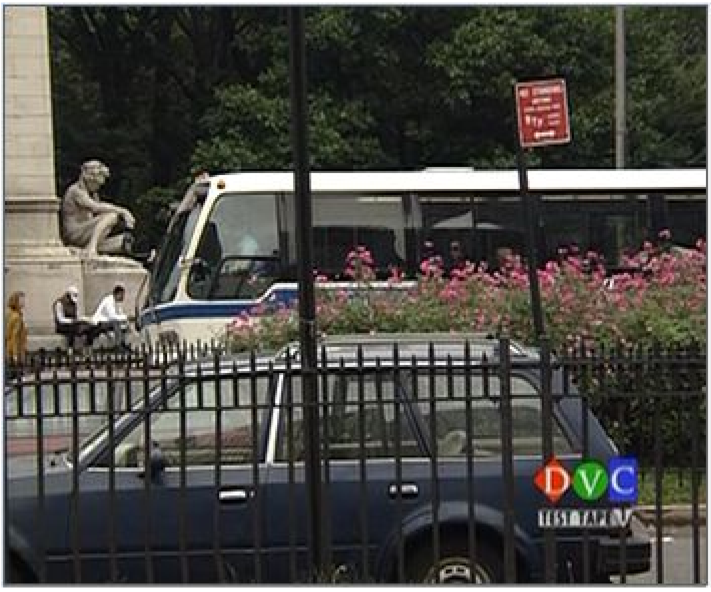


Figure 2.2: A frame in bus sequence with QP=20

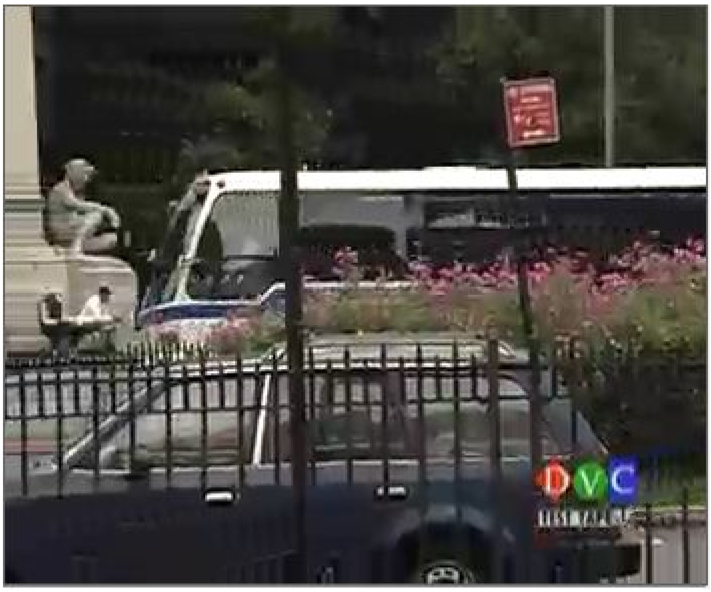


Figure 2.3: A frame in bus sequence with QP=40

Reordering and Zero Encoding

After quantization, the quantized coefficients need to be encoded with as few bits as possible. The coefficients often consist of many zero values and few non zero values. Reordering the coefficients to put the non zero values together can significantly make the entropy encoding more efficient. In DCT(Discrete Cosine Transform), the significant DCT coefficients of a block are usually located close to the "low frequency" positions close to the DC(0,0) coefficient. Based on this characteristic on distribution, the DCT coefficients can be reordered to group together non zero coefficients. A Zig-zag scan is a suitable scan order, starting from the top left DC coefficient and ending at the bottom right coefficient(Figure 2.4).

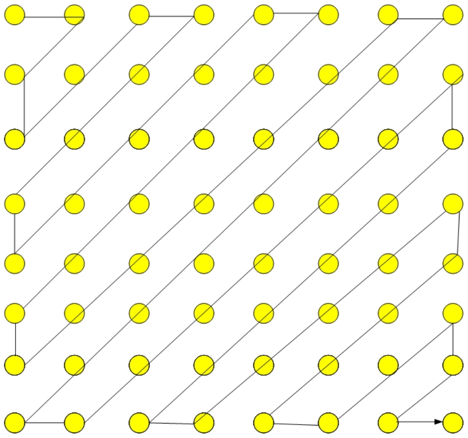


Figure 2.4: Zigzag scan order

2.1.3 Entropy Coder

The entropy coder converts a series of symbols to a compressed bit stream and is a lossless data compression. Entropy here means the amount of information included in the data, an entropy coder encodes the data with minimum necessary bits.

2.2 H.264 Standard

The H.264 standard (also called "Advanced Video Coding"(AVC) and MPEG-4 Part 10) is a standard for video compression. It is used to provide higher quality video sequences with fewer bits compares to H.263+ and previous standards. H.264 contains several profiles that can be applied to different kinds of applications. In this paper, only the baseline profile will be discussed. Baseline profile supports intra and inter coding and entropy coding(CAVLC: Context-adaptive variable-length coding). It is the most basic profile in H.264 standard, widely used in videoconferencing, video telephony, mobile devices and other places where a fast, efficient and low complexity CODEC is needed.

2.2.1 H.264 CODEC

The H.264 standard does not define a CODEC, but it provides the syntax of bit streams that a H.264 decoder should comply to. H.264 CODEC consists of two parts, the encoder and the decoder.

Encoder

H.264 Encoder(Figure 2.5) has two data flow paths, a forward path and a reconstruction path.

Forward path The forward path of H.264 encoder starts with an uncompressed frame F_n that has 16*16 macro blocks as the processing unit. For each macro block, depending on its characteristics, a prediction PRED can be made

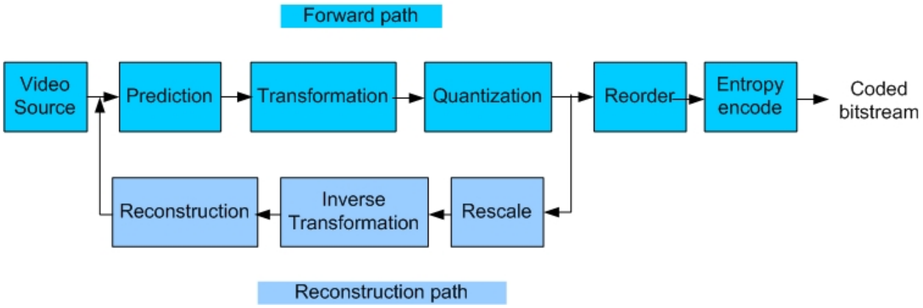


Figure 2.5: Video Encoder

either as inter or intra. The prediction is subtracted from the original block, giving a difference block D_n. Then the difference is transformed and quantized to give the compressed residual X(X is used in reconstruct a frame). The quantized coefficients are then reordered and entropy encoded. Afterwards, the entropy encoded coefficients, prediction modes, QP, and motion vector are ready to be transmitted or stored.

Reconstruction path The purpose of the reconstruction path is to reconstruct the previous encoded frames in the video sequence so that the later frames can reference these frames for inter-frame prediction. The difference block $D_n^{'}$ can be got from rescaled and reverse transformed X. The difference block $D_n^{'}$ together with the prediction PRED form the reconstructed block.

Decoder

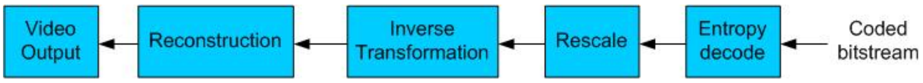


Figure 2.6: Video Decoder

The decoder(Figure 2.6) is pretty much the same as the reverse of the encoder. It receives bit stream as input. Then entropy decoding and reordering are performed to get X. Similar to the reconstruction path in the encoder, the difference block $D_n^{'}$ can be obtained from rescaled and reverse transformed X. The decoder use the header information that extracted from the bit stream to create a prediction block PRED. The difference block $D_n^{'}$ together with the prediction PRED form the reconstructed block.

2.3 Video Analysis

As in previous standards, the H.264 standard does not provide a standard CODEC but a bit stream syntax that all H.264 CODEC must comply to. This gives a lot of freedom when it comes to creating a CODEC for H.264, and the performance of different CODECs varies. Our video analysis here focuses on analyzing the prominent parameters in video encoding, studying how the different types of video sequences are encoded, in order to help during the processes of designing and optimizing the video compression algorithms.

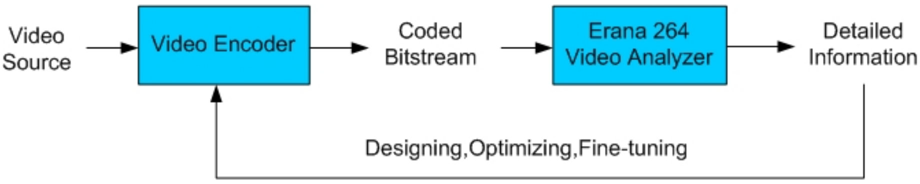


Figure 2.7: Video Analyzer

2.3.1 Important coding parameters

Video compression is a complex process which contains many stages. When thinking about video analysis, there are many factors that should be taken into consideration. Some of the encoding parameters may significantly affect the reconstructed video quality and video compression ratio. The most prominent and interesting parameters will be discussed in this section.

Macro block mode

In the H.264 standard, a picture is processed in the unit of macro block. Each macro block covers a rectangular picture area of 16 × 16 samples of the luma component and 8 × 8 samples of the chroma components. Macro blocks are numbered in raster scan order within a frame. Every coded macro block in H.264 video frame is predicted from previously-encoded data. Samples in an intra macro block are predicted from the previous coded samples in the same frame. Samples in an inter macro block are predicted from the samples in the previous coded frames. To get a better prediction, the macro block can be divided to smaller blocks, for example: 4*4,8*8,8*16 etc. There are also many kinds of partition mode in both Intra and Inter macro blocks. Besides these particular partition modes, there are 2 other special modes. In motion compensation, when the difference block does not contain any energy, no information shall be transmitted, the block is called Skipped block. For intra prediction, there is a I_PCM mode that transmit the values of the image samples directly (skip prediction and transformation).

Motion Vector and Motion Vector Difference

In an inter prediction macro block, the prediction is made by an algorithm that typically searches for the most similar area in the previous coded frames and then sets that area as the “reference”. And then the difference block is formed by subtract the prediction from the block. The motion vector is a vector that pointing from the current block to the reference area. To get better motion compensation, sub macro block is introduced and each sub block can have its own motion vector. A significant amount of bits can be spent to encode motion vectors, especially when using sub block motion vectors. To solve this problem, motion vector prediction (MVp) is introduced. The prediction of motion vectors

Macro block mode	Description
Skipped	Inter prediction block, does not have any difference block after subtract prediction.
Inter16*16	Inter prediction block, with partition 16*16.
Inter16*8	Inter prediction block, with partition 16*8.
Inter8*16	Inter prediction block, with partition 8*16.
Inter8*8	Inter prediction block, with partition 8*8.
Inter8*4	Inter prediction block, with partition 8*4.
Inter4*8	Inter prediction block, with partition 4*8.
Inter4*4	Inter prediction block, with partition 4*4.
Intra16*16	Intra prediction block, with partition 16*16.
Intra4*4	Intra prediction block, with partition 4*4.
I_PCM	Intra block, but transmitted directly without prediction and transformation.

Table 2.1: Macro Block Mode

is calculated from previous encoded neighboring blocks' motion vector. Both the encoder and the decoder use the same prediction algorithm to get MVp, in this way, only the motion vector difference (MVD) is transmitted.

Peak Signal to Noise Ratio

Peak Signal to Noise Ratio (PSNR) is an objective measurement of the compressed video frame quality. PSNR is calculated based on Mean Squared Error (MSE) between the original video frame and the compressed video frame.

$$MSE = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(C_{ij}-R_{ij})^2 \tag{2.3}$$

$$PSNR_{dB} = 10\log_{10}\frac{(2^n-1)^2}{MSE} \tag{2.4}$$

The motion compensation block size is $N \times N$ samples, C_{ij} is the current area and R_{ij} is the reference area samples. Given an original frame and the encoded frame, PSNR can be conveniently calculated. A high PSNR value means high video quality and a low PSNR value means low video quality. PSNR is very commonly used in measuring the encoded video quality, but the drawback is PSNR is not totally correlated to the subjective quality of the video. This means a human being may feel a lower PSNR video has better quality than a higher PSNR video which is compressed from the same video sequence. However, since it is the most widely accepted objective measure of visual distortion, we choose average PSNR (of luminance component and chrominance components) as our primary measure of video quality. A compressed video with acceptable quality usually has an average PSNR value from 20dB to 50dB.

Bitrate

Bitrate is the number of bits that is produced by an encoder per second in terms of video coding. When it comes to video streaming on network, though the network bit rates is increasing rapidly, it is still necessary to compress the video to a reasonable bit rates and quality. Lower bit rate and higher quality (PSNR) are the goal of video compression development.

Chapter 3

System Design

In this chapter, we discuss about the system requirements, development process and give an overview of this video analyze application.

3.1 Requirements

Implementing a high performance CODEC requires careful choice of coding parameters and algorithms. The choice of coding parameters and algorithms can have a significant impact on computational efficiency and compression performance. In order to help optimization of a CODEC and choose the proper parameters, Erana264 extracts encoding information and parameters, analyzes them in different stage/aspects and displays the results in a visual and user friendly way. The H.264 standard includes many processes and thus there are huge amount of parameters that affect the CODEC performance. Among these parameters and processes, some of them has more influence in CODEC performance and thus are more interesting to be investigated on. Based on this principal, we narrowed the list of parameters that would be analyzed. Then we further developed our requirements with potential users' feedbacks and prioritized the tasks together with the potential users. After this, we get a list of parameters which can be divided into three categories:

1. Runtime information: The information that can be retrieved in real time during the video sequence decoding. It consists mostly of picture parameters, macro block syntax elements, picture & MB insight information and other encoding parameters. The purpose of presenting these data is to get a straightforward view about how the encoder works with this current picture.

2. Summary statistics: Summary statistics includes basic information about the whole sequence.

3. Analytical information: Analytical information consists of some information that is extracted from the decoder during the decoding processes. It can help us to get in-depth knowledge about every stage in the encoding / decoding process, and might be useful for the encoder enhancement. This part needs to be improved, completed and analyzed in greater detail.

3.2 Development environment and programming languages

C/C++ is used in the parts that are within H.264 Decoder and interfacing with H.264 Decoder. The upper layers and the user interface are written in C#. This application is developed under Windows Vista Enterprise Version. Following tools are used during development:

- Microsoft Visual Studio 2008 (Compiler)
- TortoiseSVN 1.6.5 (Version Control)

3.3 Development Process

The development process of Erana264 consists three stages, at the first we collected user requirements and prioritized the tasks; then we implemented this analysis application; the last part was to run simulation and fine tuning.

3.3.1 Collect user requirements

In the first stage of this project, we ran initial simulation to find out which types of information is important in video bit stream for different QP, different sizes of sequences etc. A detailed plan is made about what information is going to be extracted and tested ,e.g motion vector, modes, encoding time for each part. Then we discussed with the potential users and prioritized the tasks. A detailed analysis plan is made during that process. Three kinds of information was decided to be extracted and tested, includes runtime information, summary statistics and analytical information as shown in Table 3.1, Table 3.2 and Table 3.3. Runtime information provides realtime, frame-based, detailed and extensive information to the users. Summary statistics gives an overview of the sequence level parameters. Analytical information offers some analyzed and visualized view of some most important parameters.

Name	Description
PPS	Picture parameter sets
Picture bits	Integer Value
MB type	Inter/Intra/Skip/etc
Total Bits	Bit
QP	Quantity Parameter
MV	Motion Vector
Coefficients	Value
MVp	Prediction of Motion Vector
MVD	Motion Vector Difference
MB searching	Search for MB according to specified request
MB Grid	Display a grid in the edge of each macro block
Zoom In/Out	Zoom in/out the frames
Mouse Hover	Show corresponding MB information at real time

Table 3.1: Runtime Information

Name	Description
Profile	Baseline
Video Format	4:2:0 / 4:2:2
Level	Value
Resolution	Width and height in pixels
Pictures Decoded	The total number of pictures that have been decoded
QP	Min / Max / Avg / Bits spent on QP
Mode/Partition Summary Info	Counting, Percentage
Time Cost	Time cost for decoding the video sequence
Bit rate	Kbps

Table 3.2: Summary Statistics

Name	Description
Predicted Picture	Display the predicted picture
Residual Picture	Display the residual picture
Distribution of MV length	Chart
PSNR	Chart
Luma/Chroma	Display Luma/Chroma component separately

Table 3.3: Analytical Information

3.3.2 Implementation

The second step was to implement analyzer code to collect all the data that is planed to be investigated and present it in a user friendly and visual format. Based on the parameters' property, they are displayed as pictures, overlay on top of the frame, lists or charts. Pictures are used to visualize the residual, predicted and reconstructed frames. Overlay display provides a more straightforward view to the users, it suits the parameters that differs for each macroblock in the same frame very well. Lists gives clear and general information for parameters in picture level, sequence level and some less important parameters in macroblock level. Chart is the best choice when we want to do comparison for some parameters between different frames and see how this parameter varies during the whole sequence.

3.3.3 Simulation utilizing and Fine-tuning

The last step is to run simulation of the developed analysis tool on different sequences, using it to find out interesting information that could be the source for help generating new ideas of performance improvement. Also some fine-tuning work is done based on these experiments. We also talked to the potential users and got valuable feedbacks and integrated some new features at this stage.

3.4 System Structure

Figure 3.1 shows the structure of Erana264 application. The first layer is the H.264 decoder previously developed in Ericsson Research. On the top of this we have a H.264 video viewer that displays the frames in different decoding stages, the overlay display of some macroblock level and picture level parameters on the top of the picture and an overview of the whole video sequence. Beside these, we retrieve some macroblock level parameters and sequence level parameters, list them in different tables. We also use some charts to visualize the PSNR values against frames, picture bits distribution and motion vector distribution etc.

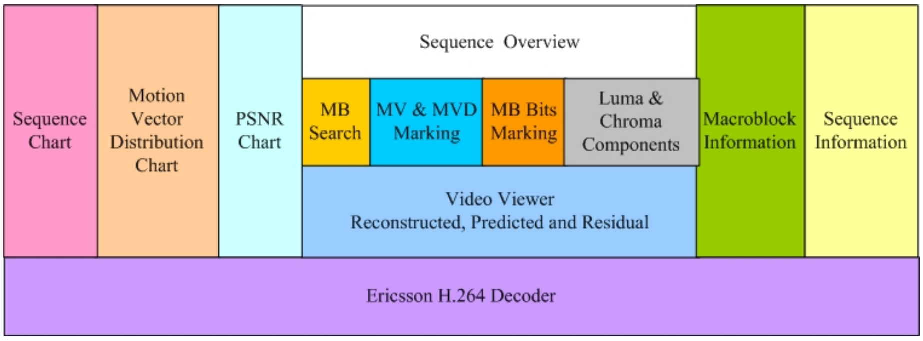
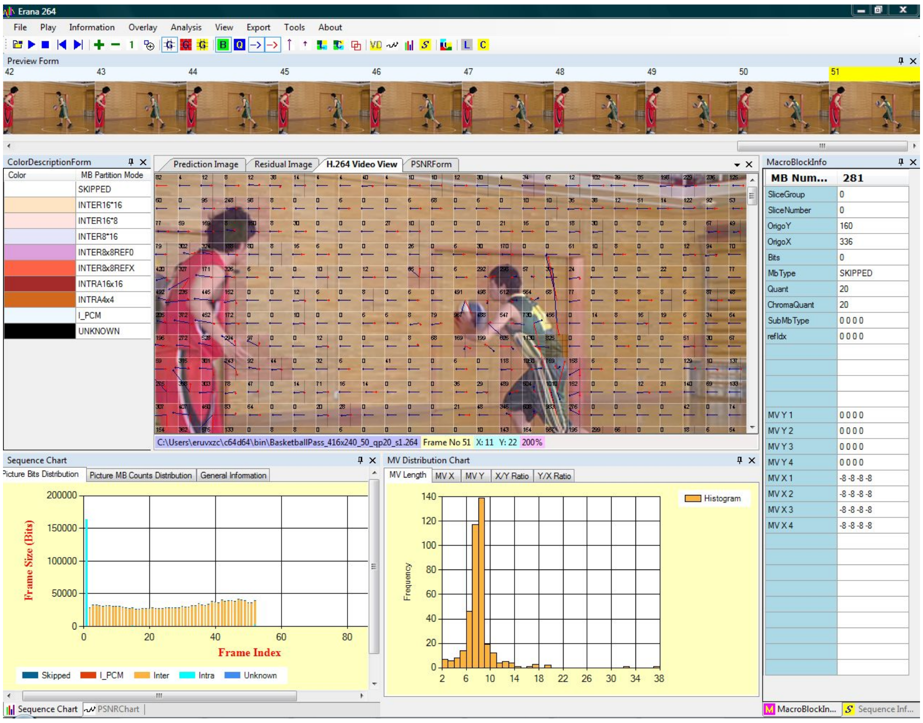


Figure 3.1: Erana264 System Structure

Chapter 4

Erana264 Functionality

Erana264(Figure 4.1) is an extensive H.264 video bitstream analyzer which extracts and analysis encoding parameters in different video encoding stages. We discuss about the main functions of Erana264 in this chapter. As a H.264 video real time analyzer, instead of decoding the whole sequence as soon as opening the sequence, Erana264 decodes in a frame by frame fashion, one frame is decoded right before it is displayed. This helps to reduce the waiting time for decoding a long sequence or a high resolution sequence and also accelerate response time. However, users can specify a buffer size for storing the previous decoded frames.



Color	MB Partition Mode
⬜	SKIPPED
🟧	INTER16*16
▨	INTER16*8
▨	INTER8*16
🟪	INTER8x8REF0
🟥	INTER8x8REFX
🟥	INTRA16x16
🟧	INTRA4x4
⬜	I_PCM
⬛	UNKNOWN

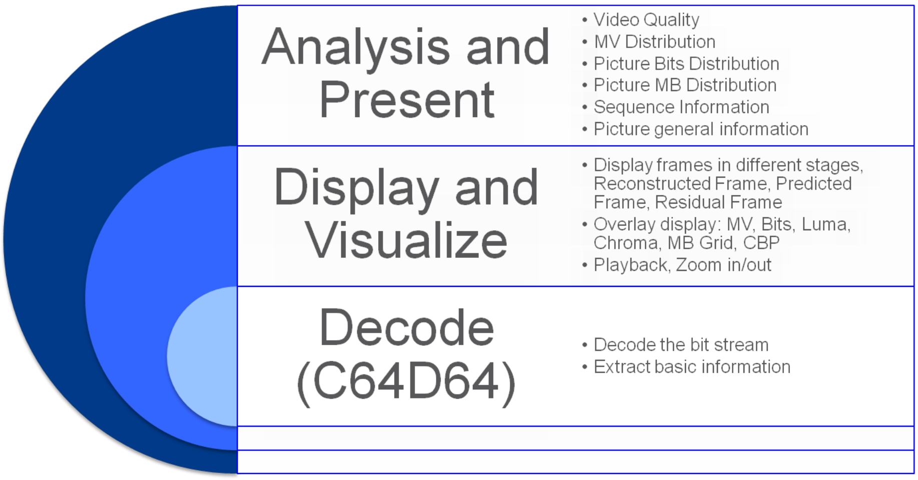
C:\Users\eruvxzc\c64d64\bin\BasketballPass_416x240_50_qp20_s1.264 Frame No 51 X: 11 Y: 22 200%

MB Num...	281
SliceGroup	0
SliceNumber	0
OrigoY	160
OrigoX	336
Bits	0
MbType	SKIPPED
Quant	20
ChromaQuant	20
SubMbType	0 0 0 0
refIdx	0 0 0 0
MV Y 1	0 0 0 0
MV Y 2	0 0 0 0
MV Y 3	0 0 0 0
MV Y 4	0 0 0 0
MV X 1	-8 -8 -8 -8
MV X 2	-8 -8 -8 -8
MV X 3	-8 -8 -8 -8
MV X 4	-8 -8 -8 -8

Figure 4.1: Erana264 Screenshot

4.1 Erana264 Overview

As is shown in the Figure 4.2, Erana264 consists of three layers. The first layer is the H.264 video bit stream decoder previously developed in Ericsson Research. This decoder is used to decode the encoded bit stream and extract the information and parameters, and send them to the higher layers. The second layer displays and visualizes the frames at different decoding stages, overlay display for some macro block and picture level information and supports basic play back functions. The third layer analyzes and presents the prominent parameters in video compression process, such as video quality measurements, mv distributions, picture bits distribution etc.



Analysis and Present

- Video Quality
- MV Distribution
- Picture Bits Distribution
- Picture MB Distribution
- Sequence Information
- Picture general information

Display and Visualize

- Display frames in different stages, Reconstructed Frame, Predicted Frame, Residual Frame
- Overlay display: MV, Bits, Luma, Chroma, MB Grid, CBP
- Playback, Zoom in/out

Decode (C64D64)

- Decode the bit stream
- Extract basic information

Figure 4.2: Erana264 three layers structure

4.2 Main Features

The main features in Erana264(Figure 4.3) can be divided to eight parts: full display mode, mb insight analysis, picture display in different decoding stages, picture general information, summary statistics, overlay display,visualize prominent parameters and import/export.

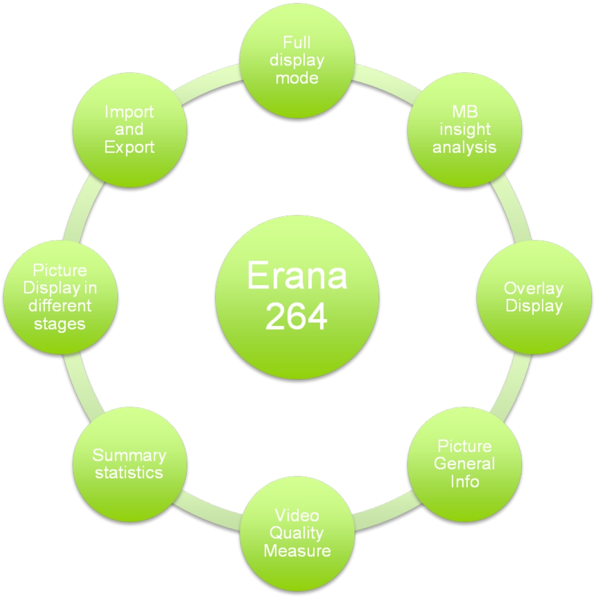


Figure 4.3: Erana264 main features

4.2.1 Full display mode

- Play / Pause / Step Forward / Step Backward: A set of play back functions.
- View Video Navigator: This shows a thumbnail view of the frames, users can navigate a video stream picture by picture.
- Zoom In / Zoom Out / Original Size: The Zooming functions enables you to adjust the magnification level of an active frame.
- Single MB detailed view: Zooms in current active macroblock.
- Luma / Chroma components only: Displays luminance or chrominance component of the image separately.

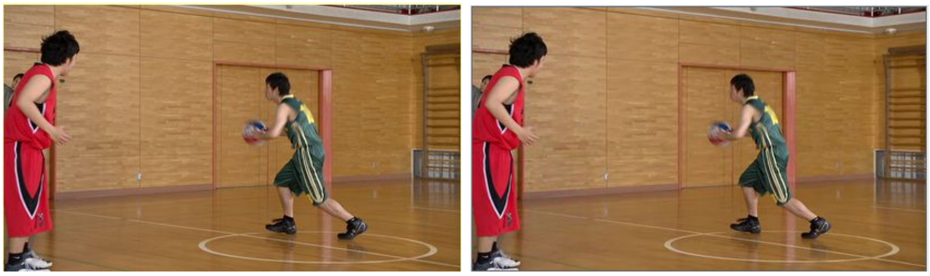


Figure 4.4: Left:The Original Frame Right:The Reconstructed Frame

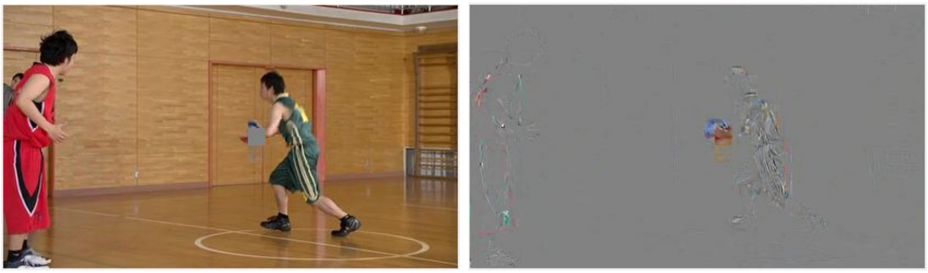


Figure 4.5: Left:The Predicted Frame Right:The Residual Frame

4.2.2 Picture display in different decoding stages

In Erana264, for each frame in the video sequence, there are four pictures visualized corresponding to four different decoding stages. They are original frame in the uncompressed video sequence, prediction frame, residual frame and the reconstructed frame. The reconstructed frame is equivalent to the sum of the prediction frame and the residual frame. Figure 4.4 and Figure 4.5 demonstrates an example of the original frame, reconstructed frame, predicted frame and residual frame that taken from “Basketball Pass” sequence.

4.2.3 Picture general information

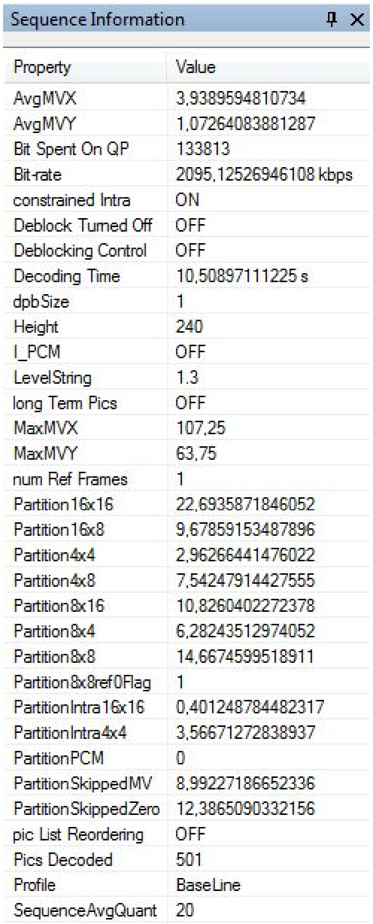
This function displays frame level statistics for the current frame.

- Picture Bits: the amount of bits that are costed to encode this frame.
- Max/Min MB Bits: the macroblocks that cost maximum/minimum bits to encode in current frame.
- MB number: the number of macroblocks that are contained in this frame.
- MB counts for different MB encode mode: the number of macroblocks in different types.

4.2.4 Summary statistics

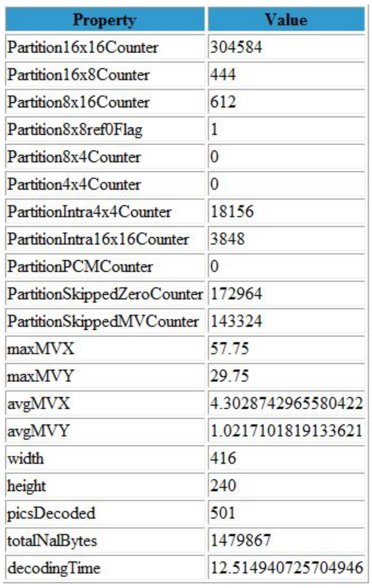
This function displays sequence level statistics for the video bitstream(see Figure 4.6). We need to decode the whole sequence to retrieve these parameters. Since

the decoding process is time and computational consuming, we create a new thread to take care of the decoding operation in the background.
An XML file which contains all these parameters is automatically generated and exported to user's disk after the decoding process(see Figure 4.7).



Property	Value
AvgMVX	3,9389594810734
AvgMVY	1,07264083881287
Bit Spent On QP	133813
Bit-rate	2095,12526946108 kbps
constrained Intra	ON
Deblock Turned Off	OFF
Deblocking Control	OFF
Decoding Time	10,50897111225 s
dpbSize	1
Height	240
I_PCM	OFF
LevelString	1.3
long Term Pics	OFF
MaxMVX	107,25
MaxMVY	63,75
num Ref Frames	1
Partition16x16	22,6935871846052
Partition16x8	9,67859153487896
Partition4x4	2,96266441476022
Partition4x8	7,54247914427555
Partition8x16	10,8260402272378
Partition8x4	6,28243512974052
Partition8x8	14,6674599518911
Partition8x8ref0Flag	1
PartitionIntra16x16	0,401248784482317
PartitionIntra4x4	3,56671272838937
PartitionPCM	0
PartitionSkippedMV	8,99227186652336
PartitionSkippedZero	12,3865090332156
pic List Reordering	OFF
Pics Decoded	501
Profile	BaseLine
SequenceAvgQuant	20

Figure 4.6: Sequence Statistics



Property	Value
Partition16x16Counter	304584
Partition16x8Counter	444
Partition8x16Counter	612
Partition8x8ref0Flag	1
Partition8x4Counter	0
Partition4x4Counter	0
PartitionIntra4x4Counter	18156
PartitionIntra16x16Counter	3848
PartitionPCMCounter	0
PartitionSkippedZeroCounter	172964
PartitionSkippedMVCounter	143324
maxMVX	57.75
maxMVY	29.75
avgMVX	4.3028742965580422
avgMVY	1.0217101819133621
width	416
height	240
picsDecoded	501
totalNalBytes	1479867
decodingTime	12.514940725704946

Figure 4.7: XML file

- Video Format
- Video Resolution
- Pictures Decoded
- Total NAL bytes
- Picture Size
- QP
- Number of Reference frames
- Max/Min/Avg MV
- Mode / Partition Summary Info
- Export to XML

4.2.5 MB insight analysis

This function enables displaying statistics on a macro block by macro block basis for the current frame.

- MB type, Position, MV, MVD: displays these statistics for current active macroblock.

4.2.6 Overlay display

This function helps user to get a straight forward view about how the current frame is encoded by displaying the parameters as overlay on the top of the picture. These parameters include:

- MB type, bits, QP, CBP, coefficients overlay display: the macro block encode mode is visualized in the frames by differentially coloring the different modes(see Figure 4.8 and Figure 4.9).
- MV, MVD overlay display: the macro block mv and mvd is visualized in the frames by painting arrows for mv and mvd on top of the image.(see Figure 4.10).
- MB Search: it allows user to locate and mark MB by specifying certain MB encode mode, MB size(the cost of bits) etc.

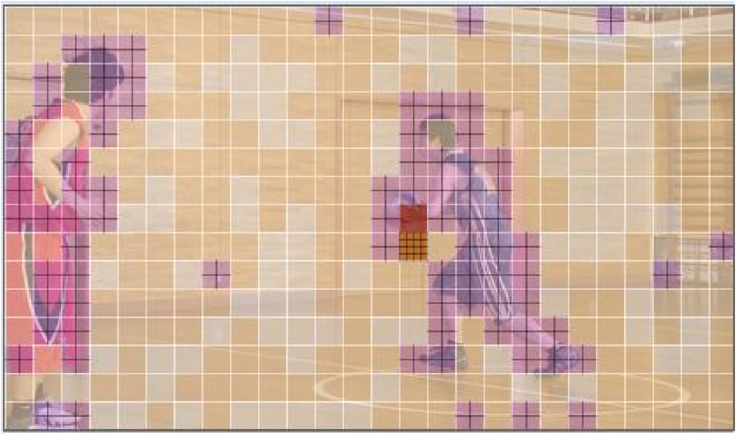
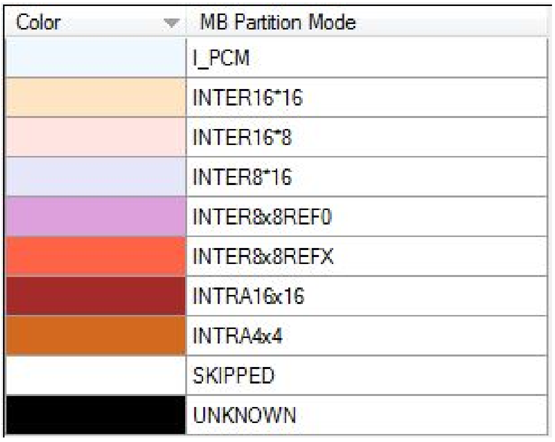


Figure 4.8: Macroblock Encode Mode



Color	MB Partition Mode
🟦	I_PCM
🟧	INTER16*16
🟥	INTER16*8
🟦	INTER8*16
🟪	INTER8x8REF0
🟥	INTER8x8REFX
🟫	INTRA16x16
🟧	INTRA4x4
⬜	SKIPPED
⬛	UNKNOWN

Figure 4.9: Color Description

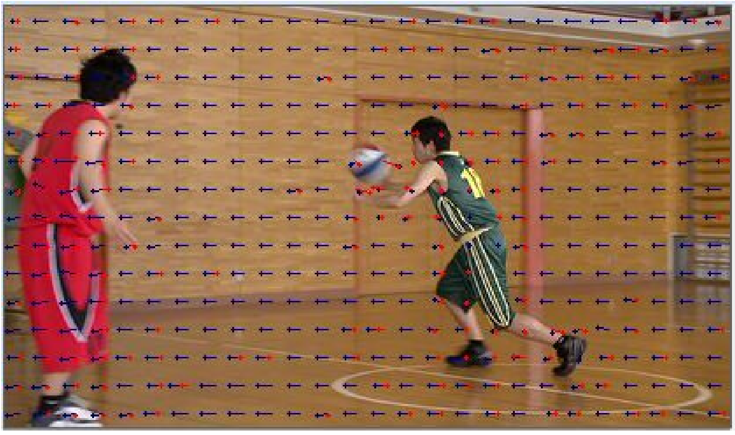


Figure 4.10: MV and MVD

4.2.7 Visualize prominent parameters

- PSNR: Erana264 use a line chart to demonstrate the PSNR variation between frames in a video sequence(see Figure 4.11). This function requires the availability of the original video.

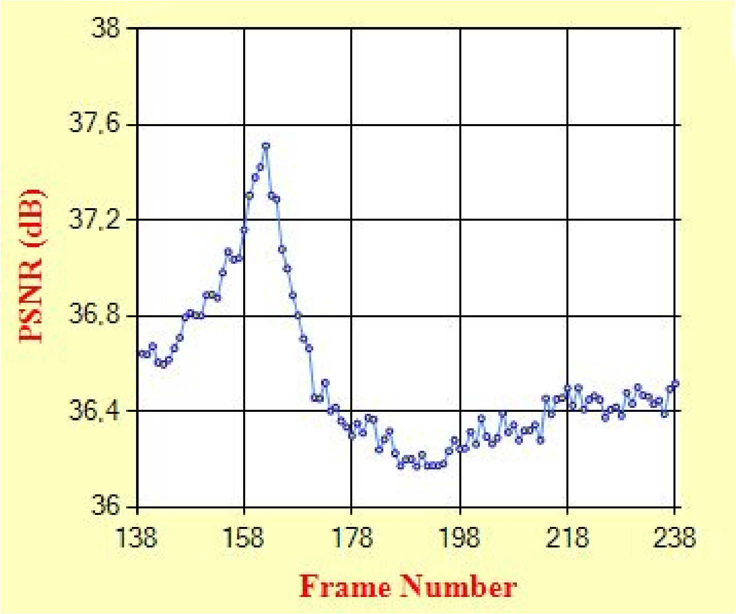


Figure 4.11: PSNR

- MV Distribution: displays the motion vector distribution(of length) for current frame, fives types of charts are provided: MV, MV X,MV Y, MV X/Y ratio, MV Y/X ratio (see Figure 4.12).

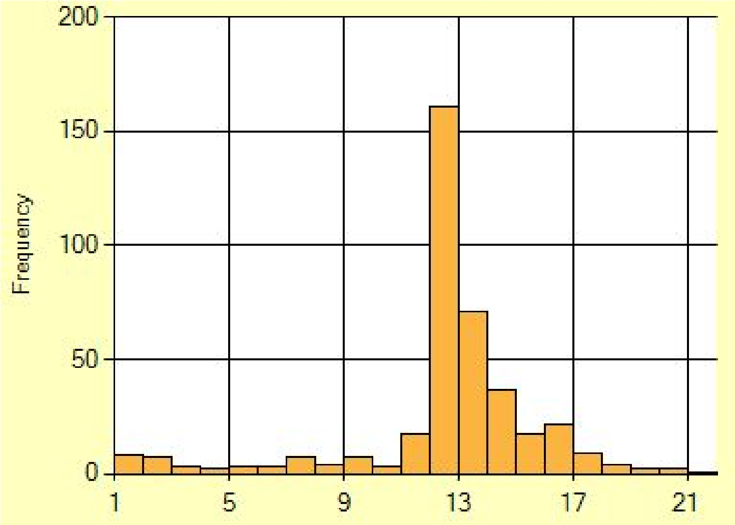


Figure 4.12: MV Distribution

- Picture bits distribution: the bit rates of video frames in a video sequence varies from frame to frame, this chart shows the bitrate variation for the sequence and the picture bits distribution for each frame (see Figure 4.13).

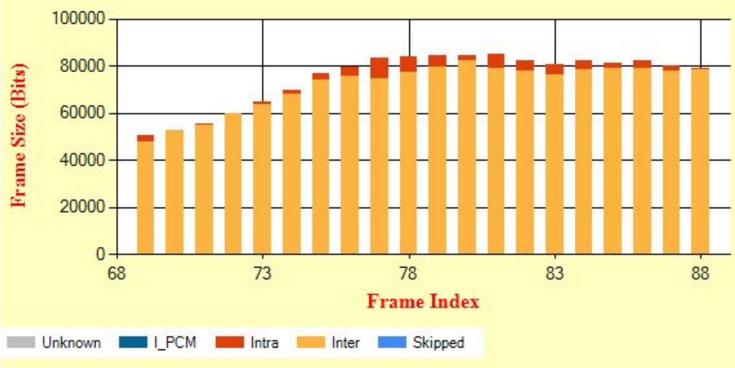


Figure 4.13: Video Frame Bits

- Picture mb distribution: displays the distribution of different mb types for current frame.

4.2.8 Import and Export

- Reconstructed / Prediction / Residual frames: export Reconstructed / Prediction / Residual frames to an external image file.

- Charts: Export / Import the charts to / from an external XML file.

Chapter 5

Experiments

We conducted two different experiments in this chapter in order to demonstrate and validate this application. Each of these two experiments is targeting a particular application area. In the first experiment we compared different encoding parameters with the same video sequence. In the second experiment we made a comparison between different video sequences which are encoded by the same encoding parameters.

5.1 Comparison between different encoding parameter with same video sequence

Erana264 enables user to do comparison between different encoding parameters and algorithms. It provides a straight forward view to see the encoding performance of different configurations. We choose “Basketball Pass” video sequence with resolution 416×240 pixels, frame rate 50fps with the length of 500 frames. This sequence has fast camera and content motion with random movement.
We encode this sequence with four different configurations(an internal H.264 encoder previously developed in Ericsson Research was used to encode these sequences). These configurations are named as “speed 1”,“speed 2”,“speed 3” and “speed 4”. These four configurations use different RDO(Rate Distortion Optimize) algorithms and motion vector searching algorithms.

- Speed 4: Only supports full pixel motion vector. Speed 4 does not support sub macroblock. For Intra prediction, only Intra16×16 with DC prediction is supported.

- Speed 3: Comparing to speed 4, the support to point wise search for sub pixel motion vectors is added.

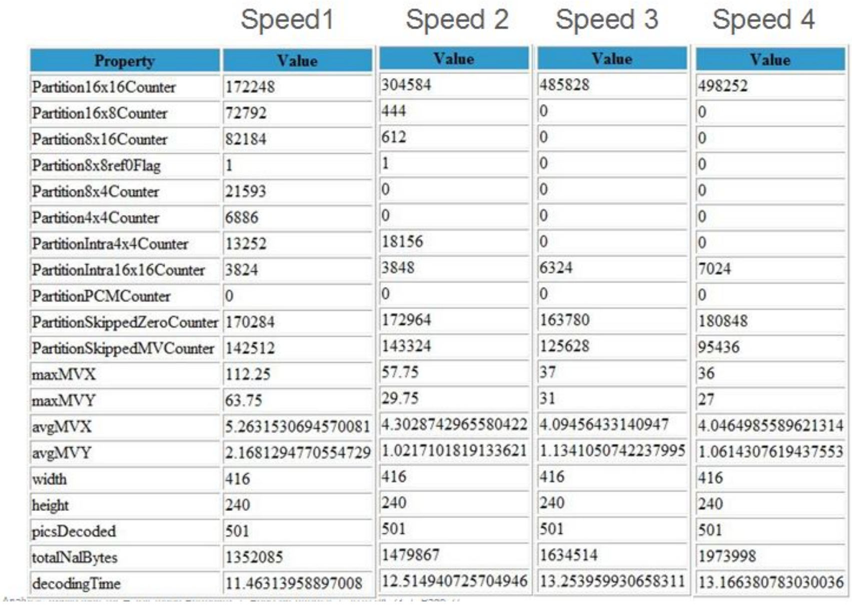
- Speed 2: Comparing to speed 3, it calculates SAD after transformation. Computes rate distortion values for each inter candidate. Tries to find the best intra mode, supports intra4×4 mode.

- Speed 1: In addition to speed 2, using RDO when searching for motion vectors, computes SAD for all block partitions. Chooses the best intra mode based on rate-distortion.

While we keep other parameters as the same(QP,profile etc.) The encoding time for these different configurations are as following: Speed 1: Encoding Time = 329.6168s; Speed 2: Encoding Time = 10.7884s; Speed 3: Encoding Time = 4.3979s; Speed 4: Encoding Time = 2.9490s.

5.1.1 Sequence Statistics

Figure 5.1 shows the summary information of these different settings. The speed 1 provides approximately 35% bit-rate savings over speed 4, closely followed by speed 2. Speed 3 and speed 4 do not support sub macroblock partitions. Speed 1 has larger motion vector searching range comparing to the other three configurations.



	Speed1	Speed 2	Speed 3	Speed 4
Property	Value	Value	Value	Value
Partition16x16Counter	172248	304584	485828	498252
Partition16x8Counter	72792	444	0	0
Partition8x16Counter	82184	612	0	0
Partition8x8ref0Flag	1	1	0	0
Partition8x4Counter	21593	0	0	0
Partition4x4Counter	6886	0	0	0
PartitionIntra4x4Counter	13252	18156	0	0
PartitionIntra16x16Counter	3824	3848	6324	7024
PartitionPCMCounter	0	0	0	0
PartitionSkippedZeroCounter	170284	172964	163780	180848
PartitionSkippedMVCounter	142512	143324	125628	95436
maxMVX	112.25	57.75	37	36
maxMVY	63.75	29.75	31	27
avgMVX	5.2631530694570081	4.3028742965580422	4.09456433140947	4.0464985589621314
avgMVY	2.1681294770554729	1.0217101819133621	1.1341050742237995	1.0614307619437553
width	416	416	416	416
height	240	240	240	240
picsDecoded	501	501	501	501
totalNalBytes	1352085	1479867	1634514	1973998
decodingTime	11.46313958897008	12.514940725704946	13.253959930658311	13.166380783030036

Figure 5.1: Sequence Statistics Comparison

5.1.2 Macro block Prediction Mode

Erana264 can easily get the distribution of different macro block prediction mode and the distribution of picture bits on different prediction mode macro blocks. The figures on the left side show the distribution of different macro blocks in all frames in the video sequence. The figures on the right side illustrate the distribution of bits on different macro blocks in all frames in the video sequence.

From Figure 5.2, Figure 5.3, Figure 5.4 and Figure 5.5 we can find in all these four sequences, most macro blocks are encoded as inter macro blocks, and most bits are spent on inter macro blocks. The video bitstreams that encoded by Speed 3 and speed 4 have much less intra mode macroblocks comparing to the other two configurations, which is resulted by their lacking of support to intra 4×4 mode..

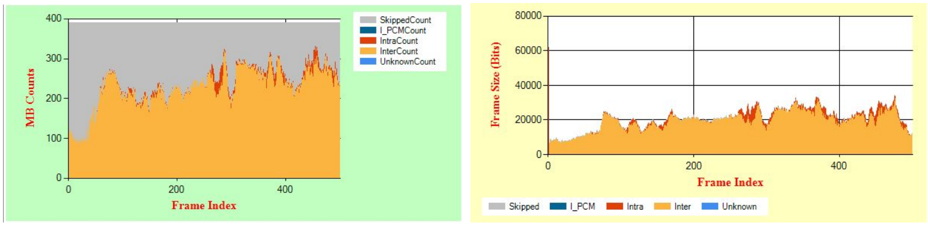


Figure 5.2: Speed 1: MBCounts and PicBits Charts

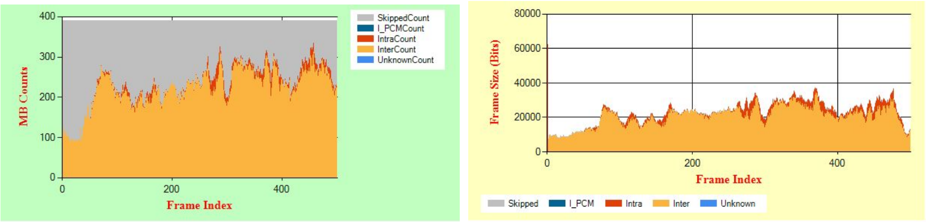


Figure 5.3: Speed 2: MBCounts and PicBits Charts

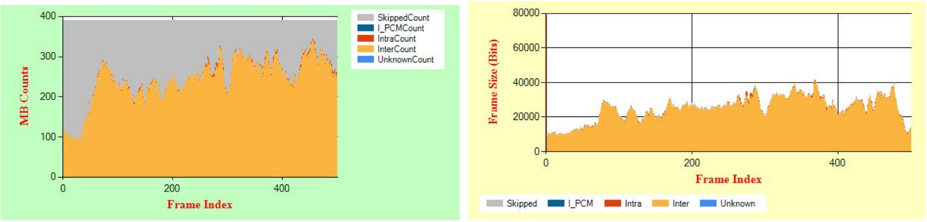


Figure 5.4: Speed 3: MBCounts and PicBits Charts

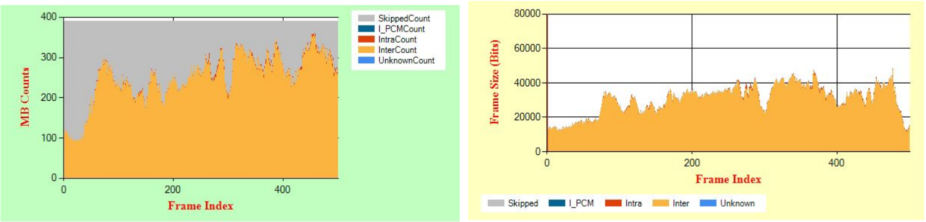


Figure 5.5: Speed 4: MBCounts and PicBits Charts

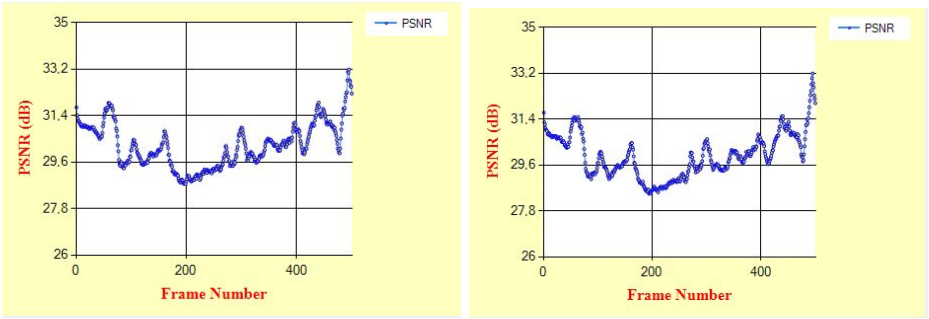


Figure 5.6: PSNR: Speed 1 vs Speed 2

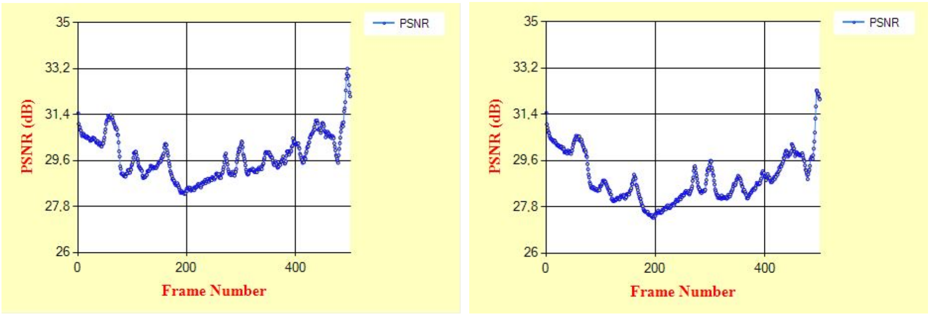


Figure 5.7: PSNR: Speed 3 vs Speed 4

5.1.3 PSNR

The comparison of video quality for these four settings. It is immediately clear from these figures that the speed 1 provides best value quality and closely followed by speed 2. While speed 4 has highest video distortion among these four settings.

5.2 Comparison between different video sequence with same encoding parameters

Four video sequences are chosen to be analyzed in this experiment. They are "Race Horses" sequence,"Basketball Pass" sequence,"Blowing Bubbles" sequence and "BQ Square" sequence.

5.2.1 Overview of the sample video sequence

These four video sequences are in the same resolution, composed of 416 × 240 pixels. Each of these four video sequences has its own distinct characteristic.

RaceHorses

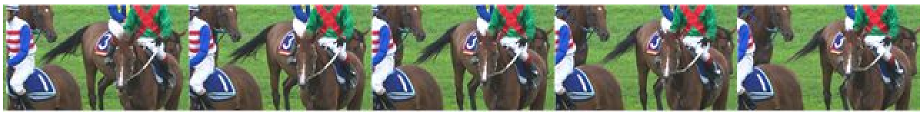


Figure 5.8: RaceHorses

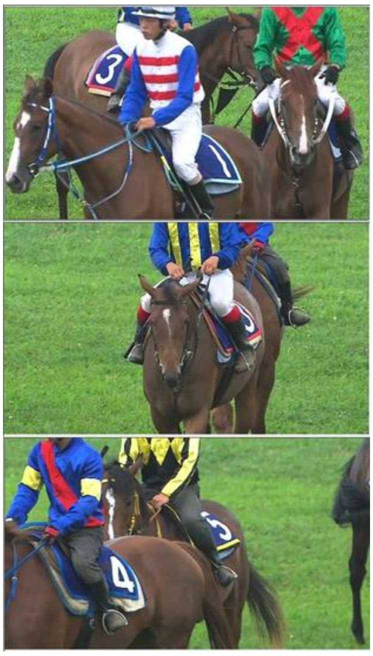


Figure 5.9: RaceHorses Frame 1, 145 and 298

RaceHorses is a video sequence records horse racing. In this video sequence, horse racers and horses consists of the foreground, the grass is the background, both foreground and background are moving. It's a dynamic, motion-filled video. There are lots of high-frequency details in this video sequence as well, the grass texture, horse tail are usually costly to encode.

BasketballPass

Figure 5.10: BasketballPass

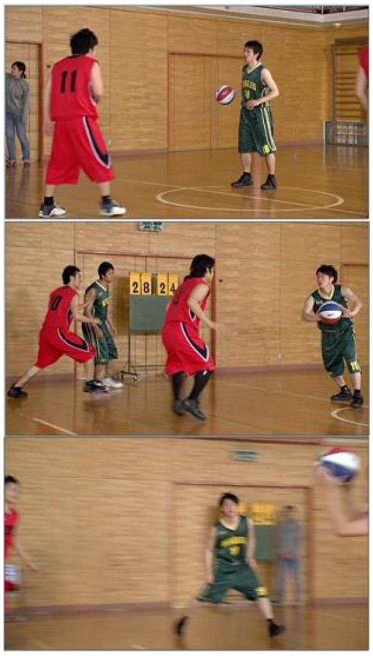


Figure 5.11: BasketballPass Frame 0, 250 and 499

BasketballPass is a video sequence taken during a basketball game. The "Basketball Pass" sequence contains pictures of high motion activity and high contrast. The background (floor and wall)has rather similar texture. The basketball players are moving in random directions make the prediction much more difficult. The camera follows the players' moving direction.

BlowingBubbles

Figure 5.12: BlowingBubbles

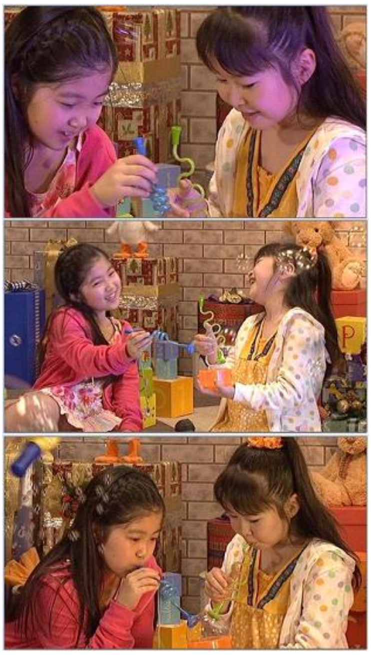


Figure 5.13: BlowingBubbles Frame 0, 250 and 498

BlowingBubbles is a video clip about two girls that blowing bubbles. The background is comparablely static. The bubbles are growing and moving in random directions. The camera zooms out generally from the beginning to the end. The bubble texture is costly to encode because it's dynamic and high frequency.

BQSquare

Figure 5.14: BQSquare

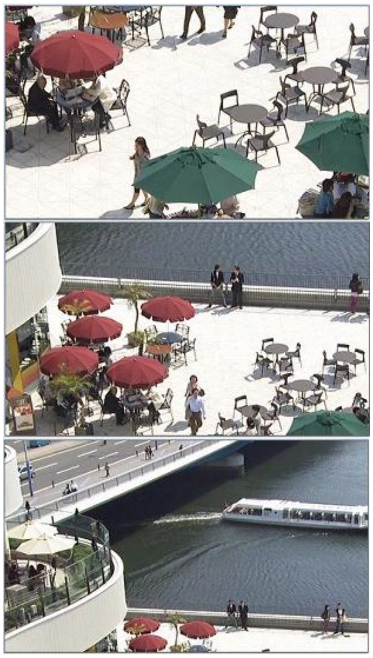


Figure 5.15: BQSquare Frame 0, 300 and 598

BQSquare is a video clip taken in a square. The background is low motion. Some people are moving in predicable directions with low speed. Generally this sequence has lower motion activities. The camera moves from the left to up right slowly.

The basic information about these four sequences are listed below:

Video Clip	Resolution	Frame rate	Length
Race Horses	416 × 240	30	299
Basketball Pass	416 × 240	50	500
Blowing Bubbles	416 × 240	50	500
BQ Square	416 × 240	60	600

Table 5.1: Basic Information

5.2.2 Motion Vector

Table 5.2 illustrates the maximum and average motion vector for different sequences. MVX is the x-axis component of motion vector and MVY is the y-axis component of motion vector. From table 5.2 we can see the “Race Horses” and “Basketball Pass” have much higher average motion vector value than the other two. “Race Horses” and “Basketball Pass” have higher motion activities. When other factors are the same, higher motion sequence usually needs more bits to encode than more static sequence.

Video Clip	Max MVX	MAX MVY	AVG MVX	AVG MVY
Race Horses	21.75	21	2.81	1.20
Basketball Pass	63	31	3.30	0.55
Blowing Bubbles	22.75	32.75	0.22	0.22
BQ Square	87	63	0.26	0.47

Table 5.2: Max and Avg motion vectors in Different Sequences

5.2.3 PSNR

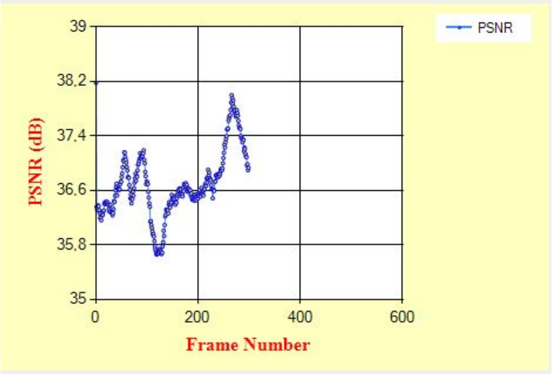


Figure 5.16: “Race Horses” PSNR Chart

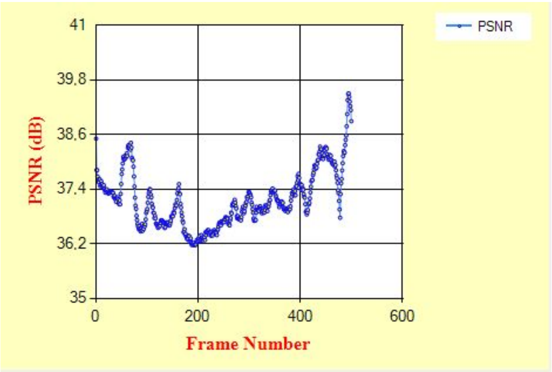


Figure 5.17: “Basketball Pass” PSNR Chart

Figure 5.16, Figure 5.17, Figure 5.18, and Figure 5.19, illustrates the PSNR values of each frame in these four different sequences. It can be clearly seen that in the “Race Horses” sequence and “Basketball Pass” sequence, the PSNR values are more dynamic. “Blowing Bubbles” sequence and “BQ Square” sequence PSNR line chart are quite stable. Consider together the motion vector, we can find that a dynamic, motion-filled video sequence usually has more dynamic PSNR values. The PSNR value increase when the motion vector difference value decrease and vice versa. When encoding a video sequence, the MVD will normally take significantly amount of bits, e.g. designing a better MV prediction scheme in dynamic region will help coding efficiency.

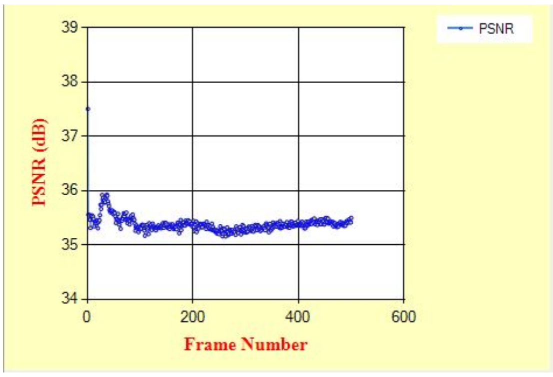


Figure 5.18: “Blowing Bubbles” PSNR Chart

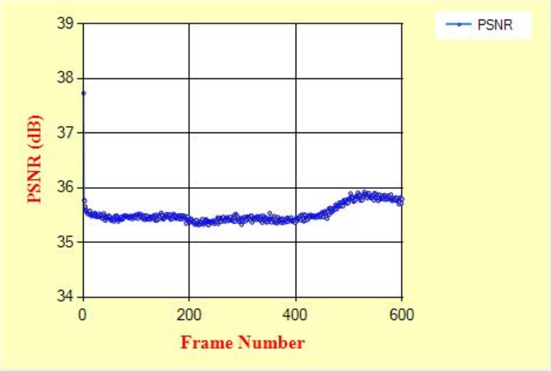


Figure 5.19: “BQSquare” PSNR Chart

Video Clip	Highest PSNR	Lowest PSNR
Race Horses	38.0	35.6
Basketball Pass	39.5	36.1
Blowing Bubbles	35.9	35.2
BQ Square	35.9	35.3

Table 5.3: PSNR Comparison

5.2.4 Macro block Prediction Mode

Erana264 can easily get the distribution of different macro block mode and the distribution of picture bits on different prediction macro block modes. The left figures show the distribution of different macro blocks in all frames in the video sequence. The right figures illustrate the distribution of bits on different macro blocks in all frames in the video sequence.

From Figure 5.20,Figure 5.21,Figure 5.22 and Figure 5.23 we can find in all these four sequences, most macro blocks are encoded as inter macro blocks, and most bits are spent on inter macro blocks.

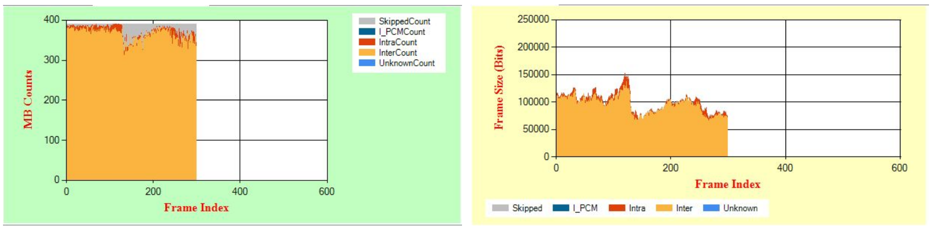


Figure 5.20: RaceHorses MBCounts and RaceHorses PicBits Charts

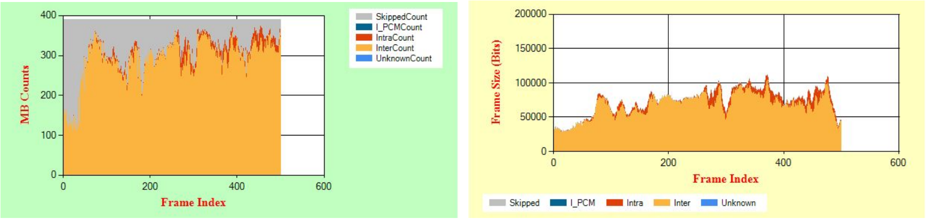


Figure 5.21: BasketballPass MBCounts and BasketballPass PicBits Charts

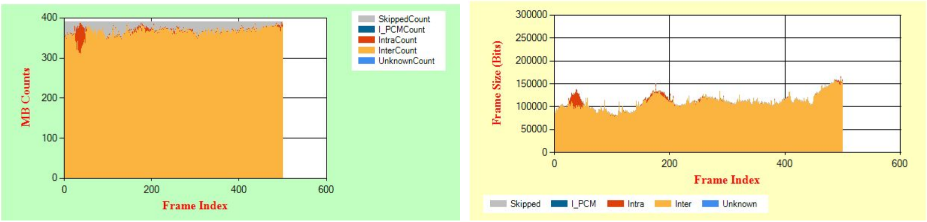


Figure 5.22: BlowingBubbles MBCounts and BlowingBubbles PicBits Charts

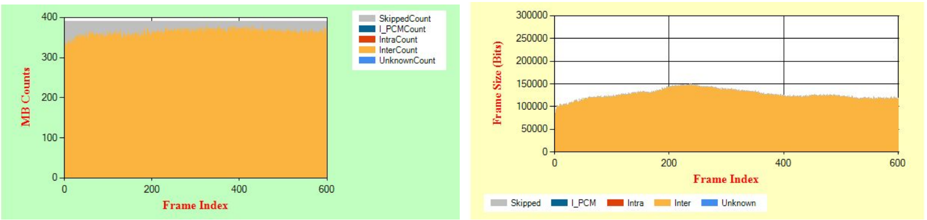


Figure 5.23: BQSquare MBCounts and BQSquare PicBits Charts

Video Clip	Intra(%)	Inter(%)	Skipped(%)
Race Horses	2.60	93.14	4.26
Basketball Pass	3.19	74.15	22.66
Blowing Bubbles	1.35	93.42	5.23
BQ Square	0.32	93.05	6.63

Table 5.4: Macro block Prediction Mode

Table5.4 illustrates the percentage of different partition mode for different sequence. Intra mode includes intra16 × 16 and intra4 × 4, Inter mode includes inter16 × 16,inter16 × 8,inter8 × 16,inter8 × 8,inter4 × 8,inter8 × 4 and inter4 × 4, Skipped mode includes SkippedMV and SkippedZero. From 5.2 we can see the "Race Horses","Blowing Bubbles","BQ Square" have very similar macro block mode distributions, among these three, "BQ Square" has fewest Intra macro blocks, that is because this sequence does not have so much new information come up and has more static background. Another reason is that it has 60 fps where difference from frame to frame is rather small compare to e.g 30 fps sequence, this would cause much higher efficiency if inter mode is chosen. "Basketball pass" has more Skipped macro block, which is due to its high similarity background, the floor and the wall have rather similar textures without any complex details. Figure 5.24 shows the distribution of macro block prediction mode in a frame from "Basketball Pass" sequence. We can see that most background macroblocks are coded in skipped mode, and a few macroblocks for moving basketball players are coded as inter macroblocks.

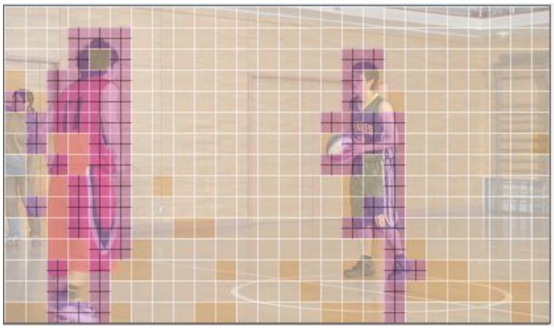


Figure 5.24: "BasketballPass" MB Prediction Mode

Chapter 6

Conclusions and Further development

6.1 Conclusions

The video analysis application ERANA264 is developed in this project. We introduced the concepts of video compression, described the system design, went through the main features and performed two experiments in this paper. Erana264 provides extensive functions to help with the development and optimization of video CODEC. A numerous video encoding parameters are extracted and analyzed in this application, which can be divided into three categories:

1. Runtime information
2. Summary statistics
3. Analytical information

The main features in Erana264 can be divided to eight parts: full display mode, mb insight analysis, picture display in different decoding stages, picture general information, summary statistics, overlay display, visualize prominent parameters and import/export.

6.2 Further development

The flat and extensive structure of this application opens up many possibilities for future work. Currently, Erana264 only processes one sequence at a time, while sometimes users want to do some comparison between different encoding configurations, then the support for multiple sequence comparison would be a nice functionality to be included.

Moreover, currently Erana264 only supports pure bitstream analysis, some extension could be made to support bitstream that contained in a container file format. For example, when the bitstream is encapsulated into RTP payload format, MPEG4 file format or 3GP file format etc.

It would also be interesting to generate XML format analytical report automatically, it saves users' time and user can easily get access to these reports in the disk afterwards.

PSNR is very commonly used in measuring the encoded video quality, but the drawback is PSNR is not totally correlated to the subjective quality of the video. This means a human being may feel a lower PSNR video has better quality than a higher PSNR video which is compressed from the same video sequence. I would also like to introduce some other video quality measures that better correlated to the subjective quality measures.

Bibliography

[1] Richardson, Iain E.G., *H.264 and MPEG-4 Video Compression*, John Wiley & Sons Ltd,, The Robert Gordon University Aberdeen,UK, 1st edition, 2003

[2] Telecommunication Standardization Sector of ITU, *H.264 Standard*, Nov 2007

[3] R. D. Dony, *The Transform and Data Compression Handbook*, CRC Press LLC,(2001), Chapter 1.

[4] Todd Will, *Singular Value Decomposition*, http://www.uwlax.edu/faculty/will/svd/svd/index.html, Retrieved 2010-05-26.

[5] Syed Ali Khayam, *The Discrete Cosine Transform(DCT): Theory and Application*, Department of Electrical & Computer Engineering, Michigan State University, Mar 2003.