International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 06, ISSN No. 2455-2143, Pages 129-139
Published Online October 2024 in IJEAST (http://www.ijeast.com)

A COMPREHENSIVE REVIEW OF ENERGY-EFFICIENT COMPRESSION ALGORITHMS FOR ENHANCING PERFORMANCE AND REDUCING POWER CONSUMPTION IN HANDHELD DEVICES

M.Jyothirmai
Department of ECE
JNTUK Kakinada, AP, India

Dr.Kesavan Gopal
Department of ECE
LPU Punjab, India

Dr.M.Sailaja
Department of ECE
JNTUK Kakinada, AP, India

Abstract— **The energy efficiency of handheld devices is of prime importance for sustainable use of the device. Watermarking, extraction, compression, decompression, encryption, and decryption are some of the tasks which can be performed on handheld devices. Earlier, a study on the effect of watermarking of digital content and its effect on energy efficiency was carried out. In this paper, a comparative analysis of energy efficiency in handheld devices is presented. The comparison, based on hardware and software platforms, focuses on computing systems and handheld devices. Multimedia content such as digital images, videos, and audio files are also considered in the following study. Energy efficiency in handheld devices can be improved by offloading compute-intensive tasks to the remote server which forms the research work by the author in the future. The tasks which are less compute-intense can be performed in handheld devices which can result in energy savings in the devices.**

Keywords— **Water Compute-intense, Handheld, PDA, MAC operations, Energy efficiency, Compression algorithms.**

I. INTRODUCTION

The primary requirement of any handheld device is to sustain the prolonged use of the battery after recharging. Examples of many activities carried out in the devices include compression, watermarking, encryption, etc. When these operations are performed, it consumes huge amounts of energy and hence leads to lesser battery life. More specifically, these activities involve various compute intense operations and are repetitive and consume reasonable energy.

When these compressions are performed on handheld devices, the energy requirements are also high due to the high computational complexity of the algorithms and are very highly compute intensely. A study of the energy efficiency of handheld devices shows that when such algorithms are executed on handheld devices it leads to draining out of the battery quickly.

Compression is of two types – lossy and lossless. In lossy image compression, there is a loss of information in the compressed image whereas there is no loss of information in lossless image compression.

Lossless compression is used in medical applications as there should be no loss of information after compression. In lossless compression, the compression ratio is small whereas lossy compression gives a higher compression ratio. The compressed image is not a replica of the original image but is a close approximation.

In this paper, an overview of the energy efficiency of compression algorithms is discussed. In image compression JPEG, JPEG 2000, in video compression H.264, HEVC and in audio compression MPEG3 and MPEG4 are discussed.

We provide an analysis of these algorithms in terms of computational complexity and energy consumption. We also present a discussion on improving the energy efficiency of

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 06, ISSN No. 2455-2143, Pages 129-139
Published Online October 2024 in IJEAST (http://www.ijeast.com)

image, video, and audio compression algorithms. A summary of the techniques and results is also presented.

The paper is organized as given. Section 2 presents the image compression algorithms and section 3 gives details about audio compression. In section 4 video compression is discussed. Section 5 gives the performance evaluation of compression, section 6 gives the analysis of the mac (multiply-accumulate) operations. Sections 7 and 8 give the energy efficiencies in hardware-based computing systems and handheld devices respectively. The conclusion and future work are presented in section 9.

II. COMPRESSION ALGORITHMS AND THEIR ENERGY EFFICIENCY FOR HANDHELD DEVICES

The classification of compression algorithms in handheld devices is shown in figure 1.

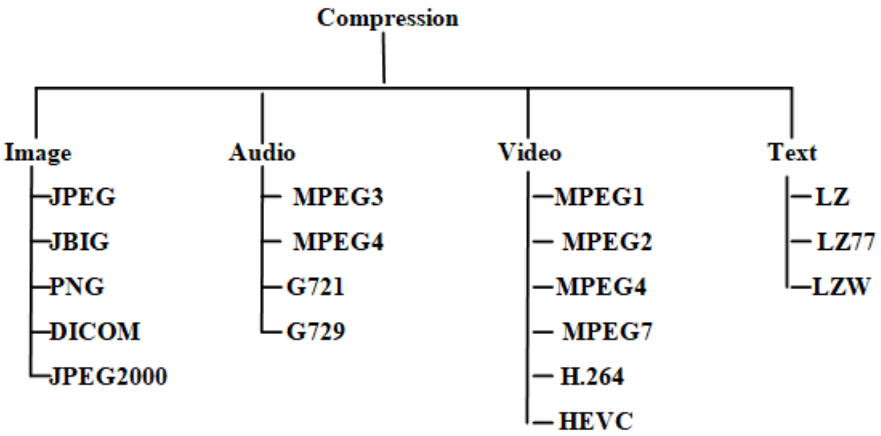

Figure 1. Classification of compression algorithms

2.1 Image Compression

Image compression reduces the size of the image by removing irrelevant or redundant information. Redundant information is that which is unnoticed by the human eye. When the images are compressed the storage space is reduced, they can be transmitted efficiently as the bandwidth is reduced and there can be a reduction in the time required for transmission.

Some of the image compression algorithms are JPEG, JPEG2000, lossless JPEG, MPEG 4-Visual Texture writing, PNG, and SPIHT.

2.1.1 JPEG

JPEG compression [1] removes most of the original information hence it is a lossy compression technique. The basic building blocks of JPEG are tiling, color space conversion, level offset, downsampling, organizing in groups, discrete cosine transform, quantization, encoding, adding a header, and marker insertion. The block diagram is shown in figure 2.

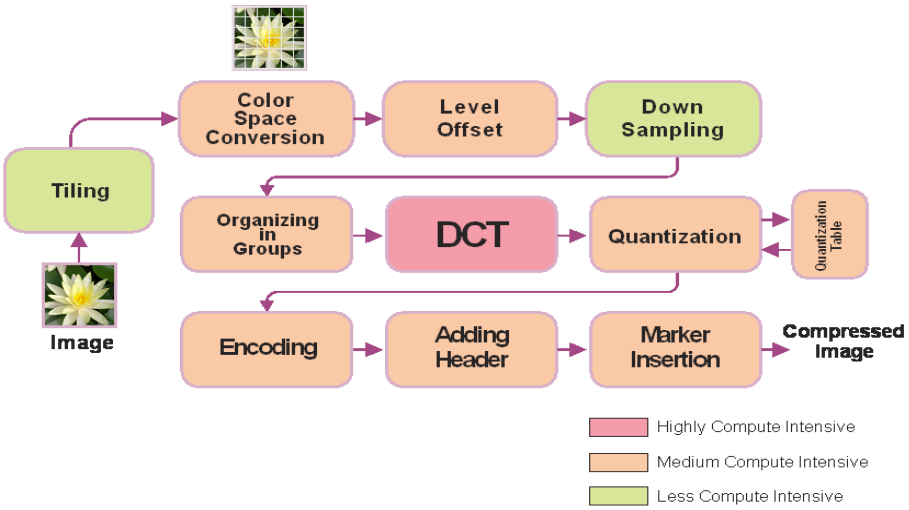

Figure 2. Building blocks of JPEG Image Compression

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 06, ISSN No. 2455-2143, Pages 129-139
Published Online October 2024 in IJEAST (http://www.ijeast.com)

1. Tiling
Tiling subdivides an image into blocks of 8x8 size. Simple tiling divides an image into two or more tiles which are coded from left to right and from top to bottom. It offers flexibility in memory as the decoder requires less memory space. The tiling operation requires no multiplication and accumulation operations.

2. Color Space conversion
Images are to be transformed from RGB color space to another color space leading to three components. This transformation can be reversible or irreversible.

Irreversible Color transform
This transformation uses the YCbCr color space. It is irreversible as it is implemented using a floating-point which causes rounding off errors.

YCbCr values can be calculated directly from RGB as shown in the below equations.

$$Y = 0.299\ R + 0.587\ G + 0.114\ B$$
$$Cb = -0.1687\ R - 0.3313\ G + 0.5\ B + 128$$
$$Cr = 0.5\ R - 0.4187\ G - 0.0813\ B + 128$$

Reversible color transform
This uses a YUV color space and does not have rounding errors, hence it is a reversible transform.

$$Y = \frac{R + 2G + B}{4}$$
$$U = R - G$$
$$V = B - G$$

Color Space conversion operation requires 13 additions and one multiplication.

3. Level Offset
Each 8x8 block is converted to a frequency domain representation using discrete cosine transform. Before computing the DCT coefficients, the pixel values in the 8x8 block are shifted from a positive range to zero centers. For an 8-bit image, the pixel values occur in the range [0,255]. The midpoint of this range 128 is subtracted from each entry and we get a value which is centered around zero, now the modified range is [-128,127]. Level offset operation requires only one addition.

4. Down Sampling
Down sampling resizes an image, the size remains the same as the original at the specified target resolution. In YCbCr color space, Y represents the luminance component and Cb Cr represents chrominance components which give blue and red differences. Since the chrominance component is not of much importance, down sampling can be done to reduce the color components. Y is sampled at each pixel, Cb and Cr are sampled at every block of 2x2 pixels, that is for every 4 Y pixels only one CbCr pixel is retained. This operation requires no multiplication and one accumulation.

5. Discrete cosine transform
Each 8x8 block of each YCbCr component is represented in the frequency domain using DCT. DCT is applied on the level-shifted and down the sampled block. When DCT is applied on an 8x8 block the resultant matrix is rounded off to the nearest two digits after the decimal point. 1D DCT operation requires 13 multiplications and 29 additions whereas 2D operation requires 192 multiplications and 512 additions.

6. Quantization
The process of quantization preserves the low-frequency information and discards the high-frequency components which correspond to noise. Each DCT term is divided by the corresponding number in the quantization table and then rounded off to the nearest integer. Quantization operation requires 64 divisions.

7. Organizing in groups
The modified spectrum is converted from an 8x8 array into a linear sequence. The zeros from the eliminated components are grouped into long runs. The quantized block is rearranged into a zig-zag manner and the run-length encoding is applied.

8. Encoding
The long runs of zeros are compressed by run-length coding. The sequence can be encoded using Huffman or Arithmetic encoding to obtain the compressed file. The encoded sequence is represented as a binary string. Huffman encoding requires many numbers of multiplications, subtractions, shift left, shift right operations, and increments.

9. Adding header
All image formats contain a header followed by the image data. The size of the header can be a few bytes to several hundreds of bytes. The header contains information about the format of the data stored in the file like resolution, color, and details of the file as when and how it was created.

10. Marker insertion
A JPEG image consists of a sequence of segments each beginning with a marker. Each marker begins with a 0xFF byte followed by a byte indicating the marker. In the entropy encoded data, 0x00 byte is inserted by the encoder after the 0xFF byte.

2.1.2 JPEG 2000
JPEG 2000 [3] [23] gives a good compression ratio when compared to JPEG with a new set of features. The compression blocks in JPEG 2000 pre-processing, Discrete wavelet transform, quantization, encoding, and bitstream organization.

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 06, ISSN No. 2455-2143, Pages 129-139
Published Online October 2024 in IJEAST (http://www.ijeast.com)

A color image can be taken as the input with three color components red, green, blue, or YCbCr. The sample values for every color may be either signed or unsigned integers with a bit-depth in the range of 1–38 bits. If the bit-depth is B bits, the unsigned illustration would correspond to (0, 2B+1), whereas the signed illustration would correspond to (2B-1, 2B+1,1). The bit-depth, resolution, and signed versus unsigned specification will vary for every element for various bit-depths, the foremost vital bits of the parts should be aligned to facilitate distortion estimation at the encoder.
Pre-processing partitions the input image into rectangular and non-overlapping tiles of equal size. The tile can be of any size. If the file size is the same as the tile it represents only one tile or the tile may be as small as a single pixel. In the next step, unsigned sample values in each component are level shifted (DC offset) by subtracting a fixed value of 2B-1 from each sample to make its value symmetric around zero.
Signed sample values are not level-shifted. Similar to the level shifting performed in the JPEG standard, this operation simplifies certain implementation issues such as numerical overflow, arithmetic coding context specification, etc., but does not affect the coding efficiency. Finally, the level-shifted values are subjected to a forward point-wise inter-component transformation to de-correlate the color data.
Two transforms operate on the first three components (RGB) of an image tile. One such transform is the irreversible color transform (ICT), which can only be used for lossy encoding and is equivalent to the conventional RGB to YCbCr color transformation and can only be used for lossy encoding. The other transform is the reversible color transform (RCT), which is a reversible integer-to-integer transform that approximates the ICT for color de-correlation and can be used for both lossless and lossy coding.
The Y component has the same bit-depth as the RGB components while the U and V components have one extra bit of precision. The inverse RCT is capable of exactly recovering the original RGB data. At the decoder, the decompressed image is subjected to the corresponding inverse color transform and the removal of the DC level shift. Since each component of each tile is compressed independently, the basic compression method in JPEG2000 is about a single tile of a monochrome image.

III AUDIO COMPRESSION

Audio compression is a lossy or lossless compression in which the amount of data in audio can be reduced without losing the quality of the audio. A few of the audio compression algorithms are MPEG, MPEG-3.

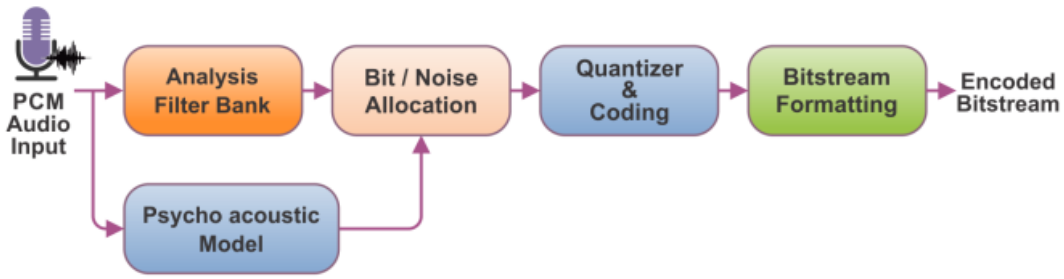

Figure 3. Audio Compression

1. Filter bank: The filter bank decomposes the input signal into sub-sampled spectral components that are to convert from the time domain to the frequency domain. It forms an analysis/synthesis system along with the corresponding filter bank in the decoder.
2. Psychoacoustic model: The time-domain input signal or the output of the analysis filter bank is taken and an estimate of the masking threshold is obtained using psychoacoustics.
3. Quantization and encoding: The spectral components are quantized and encoded. The noise, which was introduced by quantization is maintained below the masking threshold.
Thus a compressed audio file is obtained.

3.1 MPEG-3

MP3 makes use of lossy facts to encode data by the use of inexact approximations and the partial discarding of information. The result is a big reduction in audio length when compared to uncompressed audio.
MP3 compression works with the aid of decreasing or approximating the accuracy of sure additives of sound which are taken into consideration to be beyond the listening to talents of most people. This method is generally referred to as perceptual coding, or psychoacoustic modeling[8]. The last audio data is then recorded efficiently. In comparison to CD-great digital audio, MP3 compression can normally obtain a seventy-five to 95% discount in size. For example, an MP3 encoded at a consistent bitrate of 128 kbit/s might bring about a report of about 9% the scale of the unique CD audio.
An MP3 file is made from MP3 frames, which include a header and a records block. This collection of frames is called a standard circulation. Because of the "byte reservoir", frames aren't independent objects and cannot generally be extracted on arbitrary frame boundaries. The MP3 information blocks include compressed audio facts in terms of frequencies and amplitudes. The diagram shows that the MP3 Header consists of a sync word, that's used to pick out the start of a valid body.

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 06, ISSN No. 2455-2143, Pages 129-139
Published Online October 2024 in IJEAST (http://www.ijeast.com)

This is followed via a piece indicating that this is the MPEG preferred and bits that suggest that layer three is used.

3.2 MPEG-4

MPEG-4 SBR, (Spectral Band Replication) is a bandwidth extension device utilized in combination with e.g. the AAC trendy audio codec. When integrated into the MPEG AAC codec, a good improvement of the performance is to be obtained, which can be used to decrease the bitrate or improve the audio. This is carried out with the aid of replicating the high band, i.e. the high frequency a part of the spectrum. A small amount of information representing a parametric description of the high band is encoded and used inside the decoding system.

IV. VIDEO COMPRESSION

Some of the video compression algorithms are MPEG, MPEG-4, H.264, HEVC, and VP8. The steps in video compression are motion estimation, discrete transform, inverse discrete cosine transform, motion compensation, quantization, inverse quantization, and multiplexing.

The main component of the encryption method is a source encoder that compresses the incoming video signal. A video scene captured as a sequence of frames can be efficiently encoded by estimating and compensating for motion between frames before generating an inter-frame difference signal for coding. Each frame of video is uniformly partitioned into smaller units called macro-blocks, where each macro-block consists of a 16 × 16 block of luminance, and corresponding chrominance blocks.

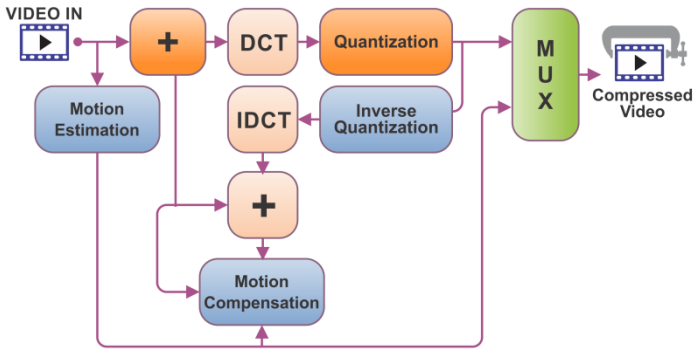

Figure 4. Video Compression

1. Motion estimation

Each block of pixels in the current frame is compared with a set of bocks of the same size in the previous frame to determine the one that predicts the current block. The set of blocks that are compared are from the search region in the previous frame centered on the position of the current block in the current frame.

When the best match is found, a motion vector is determined which is the reference block. Motion estimation is usually considered as the computationally most intensive part of the video compression system, performing up to 50% of the computations executed by the entire system. This operation requires 1 accumulation, 1 subtraction, and 1 multiplication.

2. DCT

Discrete cosine transform on an 8x8 pixel block is performed by combining transform coding and predictive coding in the form of DPCM to reduce storage and computation of the compressed image. A high degree of compression is obtained. This operation requires 13 multiplications and 29 additions.

3. Motion compensation

As it is difficult to perform this operation in the frequency domain, a motion compensation prediction error is created in the spatial domain. For each block of the current frame, a prediction block in the reference frame is obtained using a motion vector which was obtained by motion estimation. The difference in the frames produces the prediction error signal.

This computation requires only a single frame for storage in the encoder and decoder.

The matching process is done by using the mean absolute difference (MAD) or the mean square error (MSE) criteria This operation requires 3 additions, 1 multiplication, 1 subtraction, 3 comparisons, and 1 OR operation.

4. Quantizer

The transformed result is rounded off to the nearest integer using an adaptive quantizer. This operation requires one multiplication.

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 06, ISSN No. 2455-2143, Pages 129-139
Published Online October 2024 in IJEAST (http://www.ijeast.com)

5. Encoder
The quantized result is entropy encoded using a variable-length encoder and is buffered for transmission over a fixed-rate channel.

4.1 H.264
H.264 is a standard encoder used for video compression which is a process of compressing video into a format that takes up less storage space and less transmission bandwidth. Video compression is important in digital TV, DVD-Video, mobile TV, video conferencing, and internet video streaming. By standardizing video compression it is possible to encode and decode and store digital media. The H.264 **encoder** converts video into a compressed format and **the decoder** converts compressed video back into an uncompressed format.
The H.264 video encoder steps are prediction, transform and encoding to produce a compressed H.264 encoded bit-stream. A decoded video sequence is created by an H.264 video decoder by performing the complementary steps of decoding, inverse transform, and reconstruction. The biggest advantage of H.264 over MPEG-2 and MPEG-4 is compression performance, better image quality at the same compressed bit rate, or a lower compressed bit rate for the same image quality.
As an example, a single-layer DVD can be used to store a film of 2 hours' length in MPEG-2 layout. By the usage of H. 264, it should be possible to keep four hours or extra of film-great video at the identical disk (i.e. decrease bitrate for the same first-rate). as a substitute, the H.264 compression format can supply better exceptional on the identical bitrate as compared with MPEG-2 and MPEG-4.
The progressed compression performance of H.264 comes on the price of more computational price. Since H. 264 uses more advanced compression techniques than older ones, compressing and decompressing H.264 video may require significantly more computing power.]

4.2 H.265
High excessive-efficiency Video Coding (HEVC), also called H.265, is a new video compression standard, developed by the Joint Collaborative group on Video Coding. HEVC gives twice the compression performance compared to H.264 / AVC. In HEVC blocks of pixels are encoded by referring to some other region inside the same frame which is known as inter prediction or in any other frame which is known as inter-prediction.
HEVC can describe a far wider range of block sizes, up to 64 x 64 pixels, than H.264/AVC, which only defines macro blocks up to 16 x 16 pixels. HEVC allows predicted blocks to be coded in special block sizes than the residual errors. Each top degree coding unit is first coded as a prediction quad-tree, wherein at every depth the encoder decides whether to encode with merge/pass, inter, or intra coding.
The residual from those predictions is then encoded with a 2nd quad-tree that can optionally have greater depth than the prediction quad-tree. For instance, this permits the residual errors from a 32×32 inter-coded coding unit (CU) to be represented via a mixture of 16×16, 8×8, and 4×4 transforms.

V. PERFORMANCE EVALUATION OF COMPRESSION
Data are the diagrammatic representation of a fewer number of bits than the initial illustration.
A compressed file is a sort of archive that contains one or more files that have been reduced in size. Since these files are smaller they can be stored without taking up much space or can be transferred at higher speeds over the internet. Compressing files is the process of making them smaller in terms of the number of bytes of data that they contain.
Compression reduces the resources needed to store and transmit information. These resources are consumed within the compression and decompression tasks. Lossless compression represents information while not losing any data and therefore the method is reversible.
Lossy compression loses some data and it is irreversible, thus reducing storage space with minimum degradation in the image, video, or audio quality. For example, in audio compression, the data representing sound that is beyond the range of human hearing is removed from the file during the conversion process. Similarly, in images, the pixels which are unnoticed by the human eye are removed. When such data is removed it results in a compressed file thereby occupying less space.
The compressed files use less bandwidth and the transfer speed to upload or download the files is also more. The cost of storing the data is reduced as a result of compression as more files can be stored in less storage space and less storage space means less cost.

VI. COMPARISON OF COMPUTATIONAL COMPLEXITY IN COMPRESSION
When images are compressed they occupy about 12.5% of their original size. Similarly, the audio and video files also occupy much less space when compressed. Some of the compute intense operations in compression are color space conversion, down-sampling, extracting 8x8 block, 1-D DCT for rows and columns, quantization, entropy encoding, video streaming, graphics processing and rendering e.g. games, videos, heavy mathematical computations like matrix multiplication, finding factorials, prime numbers, etc. For one base operation, the required number of cycles is 7.
The number of base operations for addition/accumulation and subtraction is 1, for multiplication, it is 3 and for division, it is 5.
Table 1 below shows the number of multiplications and accumulation operations are done in the various blocks of compression.

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 9, Issue 06, ISSN No. 2455-2143, Pages 129-139
Published Online October 2024 in IJEAST (http://www.ijeast.com)

Compression operation	No. of accumulations	No. of multiplications/ divisions	No. of base operations for a block of 8x8
Color space conversion	13	1	16
Level offset	1	None	1
Discrete cosine transform(2D)	512	192	1088
Quantization	None	64 divisions	320
Motion estimation	2	1	5
Motion compensation	8	1	11
Downsampling	1	None	1
Encoding	Many	Many	2208

Table 1. Analysis of MAC operations

Mobile devices such as cell phones, PDAs, and music players have become quite common these days and therefore the use of compression on these devices is unavoidable. The compression algorithms were designed with little or no consideration for some of the system constraints such as computational complexity and energy availability. Mostly the portable devices which are easy to carry such as mobile phones and personal digital assistants have a limited battery life which drains out quickly if used frequently for applications. The computational burden refers to the number of computations. Digital compression tasks place an additional burden on the available energy in these devices and drain out the battery quickly.

VII. ENERGY EFFICIENCY IN HARDWARE-BASED COMPUTING SYSTEMS

Energy-efficient computing means reducing the amount of energy consumed, whether it is a computing system or a handheld device. As an example, in a battery-operated device, the standard objective is to reduce the whole energy of a computation that is, the integral of power with respect to time over the full computation. In most devices, power is the additional constraint. Some machines, like artificial satellites, mix alternative energy with battery backup, so they are subjected to some constraints only.

In addition, several categories are to be optimized. In any complex application, the batteries need usage to offer the best performance. Also, the system may not be energy efficient to the slightest degree. But the user has to be aware of the application.

The authors in [12] demonstrated the mapping of JPEG onto resource-restricted processors employing a style setting that creates specific use of native word lengths of the target processor. They developed a style framework that analytically determines the optimum number and three-quarter bit-widths for the signal ways within the compression method to ensure preciseness. They used this framework to mechanically generate platform-targeted JPEG C code and performed experiments. They measured the energy consumed by the compression method and compared this with the energy needed to transmit the images in their uncompressed state. It was observed that an overall speed and energy improvement of a factor of two to five was achieved. They concluded that transmission with compression is energy economical than transmission without compression.

Huaming Wu [13] proposed compressing and transmitting images in a multi-hop wireless network. He focused on the design and performance evaluation of distributed image compression algorithms. He demonstrated the benefit of using distributed image compression in sensor networks in two cases. In the first case, nodes have extremely constrained computation power. As a result, a node lacks the processing capacity necessary to completely compress a large raw image. He used a distributed method to share the processing task to overcome the computation power limitation of every single node.

He proposed two data exchange methods of distributed wavelet transform and investigated their performance in terms of energy consumption and image quality over a multi-hop wireless network. By using an image tiling technique for the raw source images, the scheme is simple and easy to implement while still satisfying image quality requirements. The suggested techniques additionally make use of node rotation to distribute the workload of computing among nodes. He proved that it prolongs the system lifetime by up to 4 times and has total energy consumption compared to the centralized algorithm.

Sungju Lee[16] used three commercial multi-core processors and analyzed the machine characteristics. He analyzed the energy efficiency by measuring the actual power consumption with a power meter. He also used three compression algorithms, various image/video data, and diverse network conditions. He showed that the proposed approach improved the energy efficiency by a factor of 2 to 5 compared to the transmission of uncompressing / compressing data with equal image/video quality.

Said [21] discussed SPIHT which is an effective method for image compression by the coefficient magnitude and ordered bit plane transmission.

V. Raghunathan [26] analyzed the characteristics of typical detector node architectures and the assorted factors that effect on system period. He then developed a group of techniques

that perform energy optimization of individual nodes. Maximizing network lifetime requires a well-structured style methodology that allows energy-aware style and operation of all aspects of the detector network. Adopting such a holistic approach ensures that energy awareness is incorporated not solely into individual detector nodes however additionally into teams of communication nodes and also the entire detector network. By following the energy-efficient methods the network lifetime can be increased by several orders of magnitude.

Table 2 shows the comparison of energy efficiency in computing systems.

Author	**Algorithm used**	**Results**
Lee Dong U[12]	JPEG with resource-constrained processors	Transmission with compression is more economical.
Huaming Wu[13]	Distributed image compression algorithms	System lifetime is prolonged up to four times.
Sungju Lee[16]	Image, Video, and data compression algorithms	20 to 50%
Said [21]	SPIHT algorithm	Encoding and decoding are very fast.
V. Raghunathan [26]	Energy optimization of nodes	The network lifetime was increased by several orders of magnitude.
R. Trobec[31]	Parellel maximum clique	3%
Albert Y.Zomaya[32]	ECS Algorithm	12%
Kristof Du Bois[33]	SWEEP Algorithm	80-85%
Yu S[34]	Non-Uniform memory access	14.7%
Y Chen[35]	Distributed online Scheduling	Achieved tradeoff between energy efficiency and system performance.

Table 2. Analysis of energy efficiency in computing systems.

VIII. ENERGY EFFICIENCY IN HANDHELD DEVICES

Handheld devices such as mobile, digital camera, portable computer, tablet PC, PDA, Android tablet, and many more are increasingly used in today's world because of the ease of carrying them anywhere. Energy is the most important criterion for handheld devices as they may be used in remote places where it is not possible to recharge them frequently.

Since handheld devices operate with a battery, we need to consider the energy consumption for efficiently compressing image/video content while still satisfying the user's image/video quality requirements. Some computations require large energy which leads to the drain of the available battery, If such computation operations are done on handheld devices it leads to the drain of the battery very quickly. To prolong the battery life, the compute intense operations can be offloaded to a proxy server. The other operations which use minimal battery can be performed on handheld devices. Hence by using proxy-based partitioning, the battery life can be extended.

Sungju Lee [18] proposed a multicore-based solution to compress and protect video data and evaluated the effectiveness of the solution in terms of both execution time constraint and energy efficiency. He measured the power consumption and the execution time of MPEG2 and AES-CCM modules with CIF data(i.e., about 3.0MB per second). He showed that the parallel software on a multicore platform requires higher total power consumption than the sequential software since each core performs some computation. Since the execution time was reduced by 3.4 times, however, the parallel software could reduce the total energy consumption by more than a factor of 2.2.

Antti P. Miettinen[27] discussed the computing to communication ratio, which is an important factor for the decision between local processing and computation offloading. The trade-off point is strongly dependent on the energy efficiency of wireless communication and local processing. He proved that the energy consumed was less by transmitting the data in one burst rather than transmitting a sequence of small data packets.

Kejriwal [15] presented an approach in which he partitioned the watermarking embedding and extraction algorithms and migrated compute-intensive tasks to a proxy server. He demonstrated that there was lower energy consumption on the handheld device as well as security of the process of watermarking. He proved that executing watermarking partitioned between the proxy and the handheld reduced the total energy consumed by 80% overrunning it only on the handheld. The performance was also improved by over two orders of magnitude.

Dong-Gi Lee [14] gave an energy-efficient transform algorithm (EEWITA) for compressing images. He also proved that there was a minimum degradation in the quality of the image, a good reduction in the number of computations, and the energy consumed. He identified wavelet image compression parameters that can be used to achieve a trade-off between the energy savings, quality of the image, and required communication bandwidth.

This was done by a method by selecting the optimum parameters such that the energy is minimized. Finally, he demonstrated the significant energy and air time (service cost) savings possible by using the energy-efficient, adaptive image codec under different cellular access technologies.

Zhiyuan Li [20] constructed a cost graph for a given application, applied a partition scheme, and divided the program into server tasks and client tasks. His results showed that there was a considerable saving in the energy through offloading.

Abdel Hamid [26] used JPEG by exploiting the energy compaction property of DCT. He proved that by partitioning the image into several sub-images and processing each region of interest, there was a considerable saving in processing time and obtained a smaller file size.

Farah Saab[28] showed the energy efficiency of multimedia applications for improved battery life while maintaining an acceptable level of user experience. For JPEG, he created a stochastic processing method to reduce energy usage, especially on mobile devices. Since video streaming and conference applications are some of the most energy-consuming applications, this is a first step in increasing MPEG's energy efficiency. He presented evidence supporting the viability of stochastic processing for multimedia encoders and examined the energy savings that resulted.

A software-based tool called PETRA was proposed by Dario Di Nucci [30] and compared to the hardware-based MONSOON toolkit on 54 Android apps. The results demonstrate that, despite not utilising any advanced hardware components, PETRA performs similarly to MONSOON. The average relative inaccuracy in relation to the MONSOON is never greater than 0.05. Moreover, 95% of the estimation errors are within 5% of the actual values measured using the hardware-based toolkit.

Table 3 gives the analysis of energy efficiency in handheld devices.

Author	**Algorithm used**	**Results**
Dong G Lee[14]	Energy-efficient transform	Achieved tradeoff between energy-saving, image quality, and transmission bandwidth.
Kejriwal[15]	Partitioning watermarking algorithms	Reduction of energy consumption by 80%
Sungju Lee [18]	Multicore-based solution	Execution time was reduced by 3.4 times.
Zhiyuan Li [20]	Partition scheme to divide the program into server tasks and client tasks.	Considerable savings in energy.
Abdel Hamid[25]	Compaction property of DCT	Processing time was reduced and energy-saving when the image was divided into sub-images and processed.
Antti P. Miettinen[26]	Local processing and computation offloading.	Energy consumed is less when data is transmitted in a single burst.
Farah Saab[27]	JPEG stochastic processing	Considerable energy savings.
Dario Di Nucci[29]	Software tool PETRA	Reduction of estimation errors.
Xiaohu Ge[35]	EEOPA Algorithm	The energy efficiency and effective capacity of MIMO-OFDM was improved.
Zhenyun Zhuang[36]	Adaptive location-sensing framework	75% increase in energy efficiency
Mayo[37]	Energy scale down model	Energy consumption reduced by a factor of 2 to 10

Table 3. Analysis of energy efficiency in handheld devices.

IX. CONCLUSION AND FUTURE WORK

We conclude this paper with a quick discussion of the ideas observed by the authors. The data transfer can be more efficient if the bit rate is made higher. The energy consumption in sensor nodes is proportional to the quantity of information being transmitted.

The latest energy compression standard JPEG 2000 has more advantages than JPEG such as better low bit rate performance and random code stream access. It was also observed that video compression is useful in reducing data transmission costs if energy cost is low.

If a combination of tiling of images, and load balancing by nodes rotation was used, it achieved a much longer system lifetime compared to the centralized image compression. This work provides an important proof of concept that shows the benefits and feasibility of distributed image compression.

In this paper, a comparison of the work in the energy efficiency of compression algorithms was done. By observing the work done by different authors, it can be concluded that, if

the entire compression operations are performed on the device itself, it leads to a drain of the available battery quickly. If the compute intense operations are offloaded to a proxy server and executed there, there can be considerable savings of the battery life of the handheld devices. Hence by using the proxy-based partitioning technique, the battery life of the devices can be extended up to several orders of magnitude.

In future work, we develop a proxy-based partitioning of operations in compression which can be done by offloading the compute intense tasks to a proxy server.

REFERENCES

[1]. Lin, Pao-Yen. (2009) "Basic image compression algorithm and introduction to jpeg standard." National Taiwan University, Taipei, Taiwan, ROC, pp.128-140.

[2]. C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, and M. Miller, (2001) "Rotation, scale, and translation resilient public watermarking for images," IEEE Trans. Image Process., vol. 10, no. 5, pp. 767-782.

[3]. Karthikeyan, A., et al. (2013) "Energy-efficient distributed image compression using jpeg2000 in wireless sensor networks (WSNS)." Journal of Theoretical & Applied Information Technology 47(2) pp. 864-879.

[4]. Chiasserini, C-F., and Enrico Magli. (2002) "Energy consumption and image quality in wireless video-surveillance networks." Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on. Vol. 5. IEEE, pp. 2357-2361.

[5]. Nayak, B. Bharath, and Ch Madhuri Devi. (2013) "Energy Saving Of Images During Transmission Using Spiht Algorithm Combined With Huffman Encoding Over OFDM Channel." Energy 3.2 pp. 1210-1213.

[6]. Rashid, Mohammad, Luca Ardito, and Marco Torchiano. (2015) "Energy consumption analysis of image encoding and decoding algorithms." Green and Sustainable Software (GREENS), 2015 IEEE/ACM 4th International Workshop on. IEEE, pp.

[7]. Ma, Tao, et al. (2013) "A survey of energy-efficient compression and communication techniques for multimedia in resource-constrained systems. "IEEE Communications Surveys & Tutorials 15.3 pp. 963-972.

[8]. http://www.cipr.rpi.edu/research/SPIHT/spiht5.html

[9]. http://scikit-learn.org/0.15 /auto_examples/ cluster/plot _lena_compress.html

[10]. https://www.researchgate.net/figure/263746148

[11]. C. Reams, (2012) "Modelling energy efficiency for computation," Ph.D. dissertation, University of Cambridge,

[12]. Lee, Dong-U., et al. (2009) "Energy-efficient image compression for resource-constrained platforms." IEEE Transactions on Image Processing 18.9 pp. 2100-2113.

[13]. Wu, Huaming, and Alhussein A. Abouzeid. (2004) "Energy-efficient distributed JPEG2000 image compression in multihop wireless networks." Proc. of IEEE Workshop on Applications and Services in Wireless Networks. pp. 152-160.

[14]. Lee, Dong-Gi, and Sujit Dey. (2002) "Adaptive and energy efficient wavelet image compression for mobile multimedia data services." Communications, ICC 2002. IEEE International Conference on. Vol. 4. IEEE, pp. 2484-2490.

[15]. Kejariwal Arun, et al. (2006) "Energy-efficient watermarking on mobile devices using proxy-based partitioning." IEEE transactions on very large scale integration (VLSI) systems 14.6, pp. 625-636.

[16]. Lee, Sungju, et al. (2012)"Energy-efficient image/video data transmission on commercial multi-core processors." Sensors 12.11 pp. 14647- 14670.

[17]. Raghunathan, Vijay, et al. (2002) "Energy-aware wireless microsensor networks." IEEE Signal processing magazine 19.2 pp. 40-50.

[18]. Lee, Sungju, Yongwha Chung, and Heegon Kim. (2012) "Secure and energy-efficient video compression on multicore-based handheld devices." International Journal of Advancements in Computing Technology 4.23 pp. 250-257.

[19]. Lee, S. Kim., H. Chung., Y., & Park., (2012) "Energy Efficient Image/Video data transmission on commercial Multicore Processors Sensors" 12(11). pp. 14647-14670.

[20]. Said, Amir, and William A. Pearlman. (1996) "A new, fast, and efficient image codec based on set partitioning in hierarchical trees." IEEE Transactions on circuits and systems for video technology 6.3 pp. 243-250.

[21]. Taylor, Clark N., Sujit Dey, and Debashis Panigrahi. (2001) "Energy/latency/image quality tradeoffs in enabling mobile multimedia communication." Software Radio. Springer London, pp. 55-66.

[22]. A Skodras, C Christopoulos The JPEG 2000 still image compression standard, IEEE Signal Processing Magazine (Volume: 18, Issue: 5, Sep 2001)Page(s) pp. 36 – 58.

[23]. Sungju Lee, Yongwha Chung, Heegon Kim Secure and Energy-Efficient Video Compression on Multicore-based Handheld Devices, International Journal of Advancements in Computing Technology(IJACT) Volume4, Number23, December 2012

[24]. Tao Ma, Michael Hempel, A Survey of Energy-Efficient Compression and Communication Techniques for Multimedia in Resource-Constrained Systems, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 15, NO. 3, THIRD QUARTER 2013 963.

[25]. Abdel Hamid Mammeri Energy-Efficient Transmission Scheme of JPEG Images over Visual Sensor Networks Conference Paper November 2008 DOI: 10.1109/LCN.2008.4664259 Source:IEEE Xplore

[26]. Antii P. Miettinen Energy efficiency of mobile clients in cloud computing ACM Digital Library. HotCloud, 2010 - usenix.org

[27]. Saab, Farah, et al. (2012) "Energy-efficient JPEG using stochastic processing." Energy-Aware Computing, 2012 International Conference on. IEEE, pp.

[28]. Erturk, Sarp. (2007 "Multiplication-free one-bit transform for low-complexity block-based motion estimation." IEEE Signal Processing Letters 14.2 pp. 109-112.

[29]. Di Nucci, Dario, et al. (2017) "Software-based energy profiling of android apps: Simple, efficient and reliable?." Software Analysis, Evolution, and Reengineering (SANER), 2017 IEEE 24th International Conference on. IEEE, pp.

[30]. R. Trobec, (2013) Energy Efficiency in Large-Scale Distributed Computing Systems, Conference: Information & Communication Technology Electronics & Microelectronics (MIPRO), 36th International Convention, pp.

[31]. Albert Y.Zomaya, (2012) Energy-efficient Distributed Computing Systems, ACM Digital Library, Wiley – IEEE Computer Society, pp..

[32]. Kristof Du Bois (2011) SWEEP: Evaluating Computer System Energy Efficiency using Synthetic Workloads, HiPEAC 'Proceedings of the 6th International Conference on High Performance and Embedded Architectures, pp

[33]. Yu S, Yang H, Wang R, Luan Z, Qian D (2017) Evaluating architecture impact on system energy efficiency. PLoS ONE 12(11): e0188428. https://doi.org/10.1371/journal.pone.0188428, pp.

[34]. Y. Chen, C. Lin, J. Huang, X. Xiang, and X. S. Shen, (2017) "Energy Efficient Scheduling and Management for Large-Scale Services Computing Systems," in IEEE Transactions on Services Computing, vol. 10, no. 2, pp. 217-230.

[35]. [35] Ge, Xiaohu, et al. (2014) "Energy-efficiency optimization for MIMO-OFDM mobile multimedia communication systems with QoS constraints." IEEE Transactions on Vehicular Technology 63.5 pp 2127-2138.

[36]. [36] Zhuang, Zhenyun, Kyu-Han Kim, and Jatinder Pal Singh. (2010) "Improving the energy efficiency of location sensing on smartphones." Proceedings of the 8th international conference on Mobile systems, applications, and services. ACM, pp.

[37]. [37] Mayo, Robert N., and Parthasarathy Ranganathan. (2003) "Energy consumption in mobile devices: why future systems need requirements–aware energy scale-down." International Workshop on Power-Aware Computer Systems. Springer, Berlin, Heidelberg, pp.