ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Motion Estimation and Video Compression of Low Power H.264

Kamatam Sateesh
kamatamsateesh@yahoo.com

Abstract

This paper presents a method to reduce the computation and memory access for variable block size motion estimation (ME) using pixel truncation. Previous work has focused on implementing pixel truncation using a fixed-blocksize (16×16 pixels) ME .In recent years, the mobile phone industry has become one of the most dynamic technology sectors. The increasing demands of multimedia services on the cellular networks have accelerated this trend. This paper presents a low power SIMD architecture that has been tailored for efficient implementation of H.264 encoder/decoder kernel algorithms. However, pixel truncation fails to give satisfactory results for smaller block partitions. In this paper, we analyze the effect of truncating pixels for smaller block partitions and propose a method to improve the frame prediction. Our method is able to reduce the total computation and memory access compared to conventional full-search method without significantly degrading picture quality. With unique data arrangement, the proposed architectures are able to save up to 63% energy compared to the conventional full-search architecture. This makes such architectures attractive for H.264 application in future mobile devices.

Keywords: Motion estimation, video coding, VLSI architecture.

Introduction

JPEG, Motion JPEG and MPEG are three well-used acronyms used to describe different types of image compression format. But what do they mean, and why are they so relevant to today's rapidly expanding surveillance market? This White Paper describes the differences, and aims to provide a few answers as to why they are so important and for which surveillance applications they are suitable. When an ordinary analog video sequence is digitized according to the standard CCIR 601, it can consume as much as 165 Mbps, which is 165 million bits every second. With most surveillance applications infrequently having to share the network with other data intensive applications, this is very rarely the bandwidth available. To circumvent this problem, a series of techniques – called picture and video compression techniques – have been derived to reduce this high bit-rate. Their ability to perform this task is quantified by the compression ratio. The higher the compression ratio is, the smaller is the bandwidth consumption. However, there is a price to pay for this compression: increasing compression causes an increasing degradation of the image. This is called artifacts.

Two basic standards: JPE G and MPE G

The two basic compression standards are JPEG and MPEG. In broad terms, JPEG is associated with still digital pictures, whilst MPEG is dedicated to digital video sequences. But the traditional JPEG (and JPEG 2000) image formats also come in flavors that are appropriate for digital video: Motion JPEG and Motion JPEG 2000. The group of MPEG standards that include the MPEG 1, MPEG-2, MPEG-4 and H.264 formats have some similarities, as well as some notable differences. One thing they all have in common is that they are International Standards set by the ISO (International Organization for Standardization) and IEC (International Electro technical Commission) — with contributors from the US, Europe and Japan among others. They are also recommendations proposed by the ITU (International Telecommunication Union), which has further helped to establish them as the globally accepted de facto standards for digital still picture and video coding. Within ITU, the Video Coding Experts Group (VCEG) is the sub group that has developed for example the H.261 and H.263 recommendations for video-conferencing over telephone lines. The foundation of the JPEG and MPEG standards was started in the mid-1980s when a group called the Joint Photographic Experts Group (JPEG) was formed. With a mission to develop a standard for color picture compression, the group's first public contribution was the release of the first part of the JPEG standard, in 1991. Since then the JPEG group has continued to work on both the original JPEG standard and the JPEG 2000 standard. In the late 1980s the Motion Picture Experts Group (MPEG) was formed with the purpose of deriving a standard for the coding of moving pictures and audio. It has since produced the standards for MPEG 1, MPEG-2, and MPEG-4 as well as standards not concerned with the

ISSN: 2277-9655
Impact Factor: 1.852

actual coding of multimedia, such as MPEG-7 and MPEG-21.

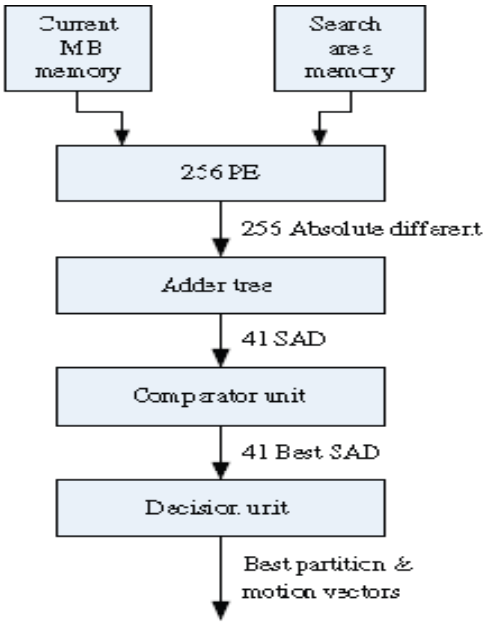
H.264

At the end of the 1990s a new group was formed, the Joint Video Team (JVT), this consisted of both VCEG and MPEG. The purpose was to define a standard for the next generation of video coding. When this work was completed in May 2003, the result was simultaneously launched as a recommendation by ITU ("ITU-T Recommendation H.264 Advanced video coding for generic audiovisual services") and as a standard by ISO/IEC ("ISO/IEC 14496-10 Advanced Video Coding"). Sometimes the term "MPEG-4 part 10" is used. This refers to the fact that ISO/IEC standard that is MPEG-4 actually consists of many parts, the current one being MPEG-4 part 2. The new standard developed by JVT was added to MPEG-4 as a somewhat separate part, part 10, called "Advanced Video Coding". This is also where the commonly used abbreviation AVC stems from. H.264 is the latest generation standard for video encoding. This initiative has many goals. It should provide good video quality at substantially lower bit rates than previous standards and with better error robustness – or better video quality at an unchanged but rate. The standard is further designed to give lower latency as well as better quality for higher latency. In addition, all these improvements compared to previous standards were to come without increasing the complexity of design so much that it would be impractical or expensive to build applications and systems. An additional goal was to provide enough flexibility to allow the standard to be applied to a wide variety of applications: for both low and high bit rates, for low and high resolution video, and with high and low demands on latency. Indeed, a number of applications with different requirements have been identified for H.264:

> Entertainment video including broadcast, satellite, cable, DVD, etc (1-10 Mbps, high latency)
> Telecom services (<1Mbps, low latency)
> Streaming services (low bit-rate, high latency)
> And others

As a note, DVD players for high-definition DVD formats such as HD-DVD and Blu-ray support movies encoded with H.264.

Block Diagram of Motion Estimation



Two-Step Algorithm

In this paper, we propose a method of pixel truncation for VBSME. This method is based on the following observations.

// T = 8'b1100_0000
// Truncating the search window pixel, Y.
$Y_t = RITAND(Y, T)$
// Truncating the current MB pixel, X.
$X_t = RITAND(X, T)$
// Initialize mv and min_cost
$mv_x = 0,\ mv_y = 0,\ cost_{min} = cost_{max}$
// Scanning the search windows and find the best match using block size $N = 8$
For $i_1 = -p_1, p_1$
  For $j_1 = -p_1, p_1$
    $cost = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1}[MATCH_1(X_t(m,n), Y_t(i_1+m, j_1+n))]$
    If($cost < cost_{min}$)
      $cost_{min} = cost,\ mv_x = i_1,\ mv_y = j_1$
  End of j
End of i
// Refining the search result using full pixel for variable block size
$cost_{min} = cost_{max}$
For $i_2 = -p_2, p_2$
  For $j_2 = -p_2, p_2$
    $cost = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1}[MATCH_2(X_t(m,n), Y_t(i_2+m, j_2+n))]$
    If($cost < cost_{min}$)
      $cost_{min} = cost,\ mv_x = i_2,\ mv_y = j_2$
  End of j
End of i

Fig. 1. Pixel truncation algorithm using two-step approach.

1) Truncating pixels for larger block sizes can result in better motion prediction compared to smaller block sizes.

ISSN: 2277-9655
Impact Factor: 1.852

2) At higher pixel resolutions, smaller block sizes can result in better prediction compared to the larger block sizes. To avoid having large motion vector errors with smaller blocks, we have implemented motion prediction in two steps. In the first search, the prediction is performed using pixels with *NTB* = 6 at 8 × 8 block size. Then, the result of the first search is refined using full pixel resolution (8-bit) in a smaller search area. The algorithm is summarized in Fig. 1. Fig. 2 shows the simulation results using truncated pixels with several matching criteria. Two error-based matching criteria and two boolean-based matching criteria are compared against SAD, namely MinMax [11], mean removed MAD (MRMAD) [12], binary XOR (BXOR) [13], and difference pixel count (DPC) [8], respectively. From the figure, at high NTB, error-based matching criteria gives a poor result compared to the boolean-based matching criteria. The combination of *NTB* = 6 and DPC gives a good tradeoff between PSNR and the computational load. At highly truncated bits, 16×16 block size is more reliable since it has more data compared to the smaller block size. However, for complex motion, the motion vector for a smaller block size, especially a 4×4 block, is not necessarily close to that of a 16 × 16 block. Since the block with smaller size difference tends to move in a similar direction, the 8 × 8 block is used in the first search. This allows us to get better predictions for either the smaller block (8×4, 4×8, and 4×4) or the larger block (16 × 8, 8 × 16, 16 × 16) from the 8 × 8 motion vector.

Motion Estimation

The motion estimation unit, shown in figure 1.2, is the first stage. The uncompressed video sequence input undergoes tem- poral redundancy reduction by exploiting similarities between neighbouring video frames. Temporal redundancy arises since the difference between two successive frames are usually simi- lar, especially for high frame rates, because the objects in the scene can only make small displacements. With motion esti- mation, the difference between successive frames can be made smaller since they are more similar. Compression is achieved by predicting the next frame relative to the original frame. The predicted data are the residue between the current and refer- ence pictures, and a set of motion vectors which represent the predicted motion direction. The process of finding the motion vector is optimal or suboptimal depending on the block match- ing algorithm chosen. Since the correlation between successive frames is inherently very high, the compression in this stage has large impact on the overall performance of the whole sys- tem. The motion predicted frames are usually called P-frames (Predicted frames). The other type of predicted frame is called B-frames (Bi-predicted frames). In this case the frame is predicted from two or more reference frames previously decoded.

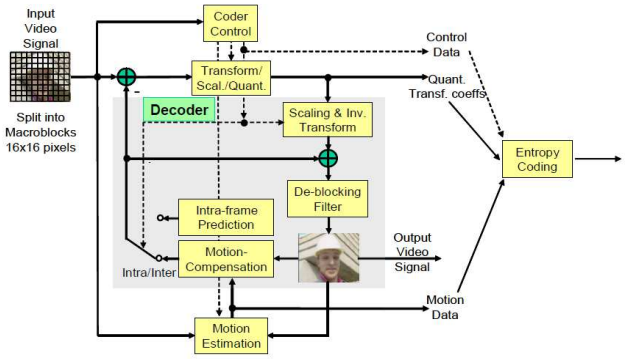


Fig 1.2 First stage of Motion estimation

Simulation and Implementation Results

A. Performance of the Proposed Two-Step Algorithm

PSNR difference using the proposed method against the conventional full-search ME (FS). The comparison is done for the frames predicted using 16 × 16, 8 × 8, and 4 × 4 partitions. Other block sizes are not included for simplicity. The difference is calculated on the basis of the average PSNR of 85 frames. Different frame sequences that represent various types of motion from low to high are used in this experiment: *Akiyo*, *Mobile*, *Foreman*, and *Stefan*. Both QCIF and CIF frame resolutions are considered, which represent the typical frame size for mobile devices. The search range, $p1 = [-8, 7]$ and $p1 = [-16, 15]$ is defined for QCIF and CIF, respectively. 2step8 represents the proposed two-step search using the 8×8 block partition. For comparison, we include the result for the two-step search where the first search is done using 16×16 partitions (2step16). The result of the first search is used as the center for the second search. fs_p4 and fs_p8 represent the conventional full-search ME with a search range equivalent to $(1/2)p1$ for QCIF and CIF, respectively. From the table, our method is able to achieve a good prediction with a smaller PSNR drop compared to the other method. For a low-motion sequence such as *Akiyo*, the PSNR drop for QCIF is below 0.05 dB. The PSNR drop increases slightly for a high-motion sequence such as *Stefan*. This is due to the prediction error and search range limitation during the first and second searches, respectively. The smaller PSNR drop for 2step8 compared to 2step16 shows that the first search using 8 × 8 partition gives a good approximation compared to 16 × 16 block size. In the 8 × 8 partitions, we have more information for the MB motion, which is important when determining the second search range for the high-motion sequence

Conclusion

ISSN: 2277-9655
Impact Factor: 1.852

This paper has presented a method to reduce the computational cost and memory access for VBSME using pixel truncation. Previous work has shown that pixel truncation provides an acceptable performance for motion prediction using a 16 × 16 block size. However, for motion prediction using smaller block sizes, pixel truncation reduces the motion prediction accuracy. In this paper, we have proposed a two-step search to improve the frame prediction using pixel truncation. Our method reduces the total computation and memory access compared to the conventional method without significantly degrading the picture quality. The results show that the proposed architectures are able to save up to 53% energy compared to the conventional full-search ME architecture, which is equivalent to 40% energy saving over the conventional H.264 system. This makes such architecture attractive for H.264 application in future mobile devices.

References

[1] Y. Lin et.al, "Soda: A low-power architecture for software radio," Proc. of the 33rd Annual International Symposium on Computer Architecture, pp. 89–100, June 2005.

[2] I. Richardson, "H.264 and MPEG-4 video compression," WILEY, 2003.

[3] N. Goel, A. Kumar, and P. Panda, "Power reduction in VLIW processor with compiler driven bypass network," Proc. of the 20th International Conference on VLSI Design held jointly with 6th International Conference on Embedded Systems, pp. 233–238, Jan. 2007.

[4] K. Fan et.al, "Systematic register bypass customization for application specific processors," Proc. of IEEE 14th International Conference on Application-Specific Systems, Architectures, and Processors, pp. 64–74, June 2003

[5] ISO/IEC14496-2. Amendment 1, Information technology - coding of audio-visual objects - Part 2: Visual. 2001.

[6] ISO/IEC15444. Information technology - JPEG2000 image coding system. 2000.

[7] V. Iverson, J. MacVeigh, and B. Reese. Real-time H.24- AVC codec on Intel architectures. In Image Processing, 2004. ICIP '04. 2004 International Conference on, vol- ume 2, pages 757–760, Oct. 2004