TELKOMNIKA, Vol. 11, No. 12, December 2013, pp. 7415 $\sim$ 7422
ISSN: 2302-4046

Implementation And Optimization of A Fast Inter Predication Algorithm

Xiaoyu Li*, **Guowu Yang , and Desheng Zheng**
College of Computer Science and Engineering
University of Electronic Science and Technology of China, Chengdu, 60054
*erin.xiaoyu.li@gmail.com

Abstract

Audio Video coding Standard is the second generation Source Coding-Decoding standards of China, especially for embedded audio/video platform. This paper proposes an efficient and fast inter prediction algorithm, which is one of the key techniques of Audio Video coding Standard. Reducing of the redundancy in source sequence inter prediction, it could improve the picture quality. The optimization schemes include two aspects, which are the algorithm framework, variables and data structure. Experimental results demonstrate that our algorithm has improved significantly $11.09 - 16.97\times$ *efficiency than original algorithm. Furthermore, this research also gives a valuable insight of the combination with quantum information.*

Keywords: *Audio Video coding Standard, inter prediction, reference sample, quantum information, Source Coding-Decoding*

1. Introduction

Recently, following the technology of networks and wireless, the demands of video transmission's high quality are increased. As a result, a lot of techniques are devoted to coping with different transmission cases. For various transmission environments, if you want to keep a distinct transmission quality, it would be an imperative issue that how to develop an efficient decoding technique. Sequentially, the H.264/AVC [1, 2], Audio Video coding Standard (AVS) [3–7], VC-1 and DIRAC are presented. These standards are adopted by ITU-T/ISO-IEC, China standards organization, SMPTE and BBC [8], respectively. AVS is the second generation Source Coding-Decoding standards with independent Intellectual Property rights of China. It includes four main technical standards: system, video, audio and digital copyright management, and support standard: consistency testing [9]. AVS coding efficiency is more than 2-3 time of MPEG-2 [9–12], and similar with AVC [13, 14]. Moreover, the AVS technical plan is succinct, less complex. Compared to H.264 [13, 14], it has the advantages of higher compression, image quality and wide application, low patent fees and low complexity [14–17]. As a second-generation video codec standard, the framework of AVS adopts the hybrid coding system, and combines the spatial domain prediction with the temporal prediction, that is, using intra-prediction and inter-prediction to coding the image. A complete AVS codec in theory is composed by several modules, such as the entropy decoding, inverse-scanning, de-quantization, inverse transformation, intra-prediction, the motion compensation and loop filter. The attendant of AVS is a huge amount of computation, despite it has achieved and abstained higher picture quality and compression efficiency comparing to the previous video coding standard. As extension , the raise of computational complexity limits the application of AVS, especially some of the relatively high real-time applications.

The main principle of inter prediction algorithm is that using the decoding picture of the encoded image as a reference image of the current coded image. Then it will be chosen as a reference sample from a reference picture. Inter prediction technology can be used to enhance the encoding compression efficiency. Therefore, how to implement the inter prediction algorithm efficiently and optimization of it are necessary. Optimization of the program usually refers to optimize the code or the speed of implementation of program. It is also the big challenge to

Received April 22, 2013; Revised August 19, 2013; Accepted August 31, 2013

balance them. While improving the speed of codec, it is inevitable to decline the quality of video picture. With the situation of processor being limited to resources, it is the main direction of the video information processing software that to seek the best compromise between complexity and codec performance [18].

In recent years, the quantum computation and quantum information [19] has been a hot research topic. Base on the characteristics of quantum states including quantum "wave-particle duality", superposition, quantum entanglement and quantum no-clone, the classical source coding has made full use of quantum coherence unique properties to explore a new way for information calculation, coding and transmission. How to use the way of quantum information coding for the traditional source coding and transmission, it will be the next research focus on. For example, it is important for the source decoding to improve decoding rate according to the quantum error correction coding [20–24], etc.

Based on the above research, this paper proposes an efficient and fast inter prediction algorithm, which could reduce the redundancy in source sequence inter prediction and improve the picture quality. Optimized results demonstrate that our algorithm has a notable improvement of the clock cycle efficiency. Furthermore, this research also gives a valuable insight of the combination with quantum information. The structure of this paper is organized as follows: Section 2 is the research method, including the basic introduction of inter prediction algorithm, the implementation of inter prediction algorithm and the optimization of the inter prediction algorithm. The detailed description of implementation of the inter prediction algorithm will be given in Section 2.2. Section 2.3 will optimize the inter prediction algorithm from two aspects: algorithm framework's optimization, the variables and data structure optimization. In Section 3, experimental results will be demonstrated to compare the work and discussed. The conclusions will be given in Section 4.

2. Research Method

2.1. The principle of inter prediction algorithm

Inter prediction technology could be used to enhance and improve the encoding compression efficiency and picture's quality. In order to remove the temporal redundancy in the video streams and sequences, inter-prediction needs to predict from the previously decoded frames or fields. AVS inter prediction algorithm has a maximum of two reference frames, including the case of two forward frames and two backward frames, or the case of one forward and one opposite of the total two reference frames. In the inter-prediction coding, there are two kinds of frame: forward prediction of P-frame and bidirectional prediction of B-frame. Where P-frame uses two forward frames as the reference sample. Meanwhile, B-frame chooses one forward frame and one backward as the reference sample. Inter prediction algorithm takes the macro block as the basic unit and has been adopted the sub-pixel motion vector (MV) in the motion estimation.

Every partitioning or sub-block of the macro block of inter coding is obtained by predicting in the same region of the reference picture. The difference between the two locations is luma component with $1/4$ pixel precision and chroma component with $1/8$ pixel precision. The luma and chroma pixel which is on the location of the sub-pixel does not exist in reference picture actually. Instead, it is got by using the nearby pixel of reference samples to interpolate pixel. If the MV is an integer, it means that the corresponding pixel of reference picture block already exists in practice. Or else, it needs to be predicted by pixel interpolation. The luma pixel interpolation of AVS is divided into $1/2$ precision and $1/4$ precision. The location relationships of the pixels are shown as Figure 1.

1. Half position sample calculation:
   Half-pixel interpolation uses the four-tap filter $F_1(-1, 5, 5, -1)$ to filter at first.

   - half samples b: Use F_1 to filter the nearby four integral samples of horizontal direction at first, then obtain the intermediate result $b' = (-C + 5D + 5E - F)$, then, $b = Clip1((b' + 4) >> 3)$.
   - half samples h: Use F_1 to filter the most similar four integral-pixel samples of vertical

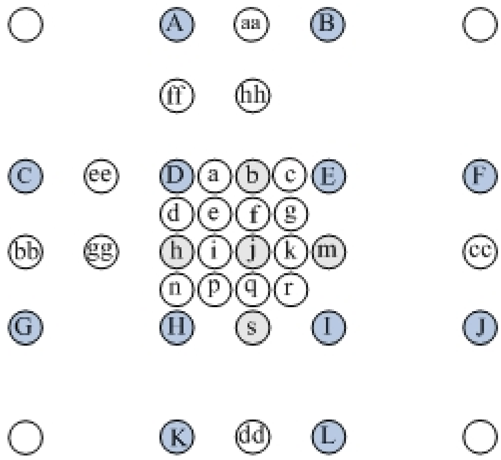

Figure 1. The Location Relationships of The Pixels

direction at first, then obtain the intermediate result $h' = (-A + 5D + 5H - K)$, then, $h = Clip1((h' + 4) >> 3)$.

- half samples j: Use F_1 to filter the most similar four half-pixel samples' intermediate value of vertical direction at first, then obtain the intermediate result $j' = (-bb' + h' + m' - cc')$ or $j' = (-aa' + b' + s' - dd')$, which bb', h', m', c', aa', B', s' and dd' are the half sample median values of the corresponding location.

The final result of interpolation: $j = Clip((j'+32) >> 6)$. Several other half location samples with the same calculation method as above described.

2. Quarter-position sample calculation:
   Quarter-pixel sample interpolation uses the filter $F_2(1, 7, 7, 1)$ to filter at first.

- Quarter sample interpolation a: Use F_2 to filter the nearby samples at first, and then obtain the intermediate result $a' = (ee' + 7D' + 7b' + E')$, which ee', b' are the half sample median values of the corresponding location, D' and E' are the corresponding location of the integer sample eight times. The final results: $a = Clip1((a' + 64) >> 7)$.
- Quarter sample interpolation d: Use F_2 to filter the nearby pixels at first, and then obtain the intermediate result $d' = (ff' + 7D' + 7h' + H')$, which ff', h' are the half sample median values of the corresponding location, D' and H' are the corresponding location of the integer sample eight times. Final results: $d = Clip1((d' + 64) >> 7)$. The interpolation calculation method of the quarter-sample is similar to d.
- Quarter sample interpolation i: Use F_2 to filter the most nearby four half-pixel samples' intermediate value of horizontal direction at first, and then obtain the intermediate result $i' = (gg' + 7h'' + 7j' + m'')$, which gg', j' are the half sample median values of the corresponding location, h'' and m'' are the corresponding location of the half sample eight times. Final results: $i = Clip1((i' + 512) >> 10)$.
- Quarter sample interpolation f: Use F_2 to filter the most nearby four half-pixel sample hh', b'', j', s'' intermediate values of vertical direction at first, and then obtain the intermediate result $f' = (hh' + 7b'' + 7j' + s'')$, which hh' and j' are the half sample median values of the corresponding location, b'' and s'' are the corresponding location of the half sample eight times. The final results: $f = Clip1((512 + f') >> 10)$.
- Quarter sample interpolation e, g, p, r: The four quarter-sample's calculation is relatively simple and has the same method, as follows:

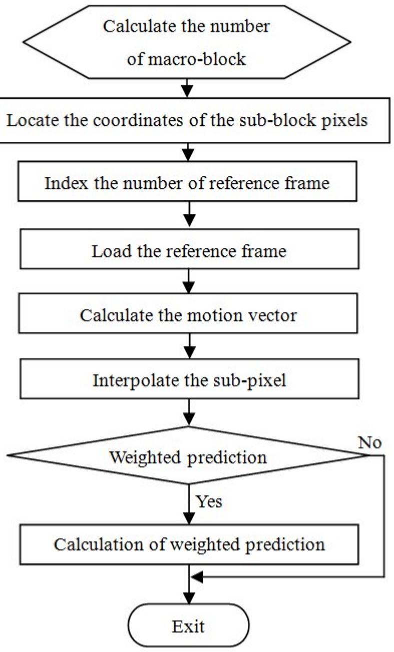

Figure 2. The Flow Chart of Inter Prediction Algorithm

$$e = Clip1((D'' + j' + 64) >> 7),$$
$$g = Clip1((E'' + j' + 64) >> 7),$$
$$p = Clip1((H'' + j' + 64) >> 7),$$
$$r = Clip1((I'' + j' + 64) >> 7).$$

D'', E'', H'', I'' are the corresponding location of the integral sample 64 times, j' is the half of the sample j' value [13]. The sample interpolation of AVS can be precised to $1/4$ pixel. The pixel-interpolation is simple to achieve. It is useful to hardware implementation, and also has provided a theoretical basis for the development of multimedia devices based on the AVS standards.

2.2. Implementation of the inter prediction algorithm

The module of inter-prediction is a key core module of AVS codec. Decoding frame has been used as a reference of the current frame. Meanwhile, combination the motion vector interpolation from reference samples, then form a sample block. This process is the so-called motion compensation. AVS inter prediction has a maximum of two reference frames. This can reduce the complexity of computation and improve the speed of implementation.

Here for the inter prediction algorithm, a framework of it has been done. Following this flow chart (Figure 2), the program of inter-prediction has been finished by C language. Figure 2 is the inter-prediction algorithm to achieve the framework of this process. In this paper, we use the official AVS RM50d decoder modeled to analyze the key algorithm, make TI's TMS320DM642 DSP development board as the development platform, and make fully rational use of DM642's hardware resources. Code Composer Studio is the supporting software development kit. The paper focuses on the inter-prediction module and optimizes it from two aspects, not only taking into account the code efficiency, but also improving the picture quality. The algorithm is shown as blow.

1. Calculate the specific location of the current macro-block's first 8×8 sub-block in the frame.

2. Locate the coordinates of the pixels in sub-block. The coordinate makes pixels as units.

Table 1. The Time Consuming of Each Module in Decoder

Model Name	**Time Ratio($\%$)**
$Read\_golomb\_grad_bits$	17.7
$readCoeff\_B8$	15.1
$Reconstruct$	6.0
$Inv\_Transform\_B8$	5.5
$Inter\_Pred$	28.2
$Intra\_Pred$	2.3
$Others$	25.2

3. Calculate the reference frame number of current frame from the reference frame number array. Because each decoding frame should be provided with two reference frames in AVS, so it is necessary to index the reference frame array to ensure which frame can be used as a reference at a final confirmation.

4. Calculate the motion vector. Each macro block or the sub marc block has its own motion vector. And macro block use the motion vectors to determine the sample position in the reference frame.

5. Pixel interpolation. This processing unit is the most key core of the inter prediction. There are 16 different locations of the pixel interpolation in total; different positions have different interpolation calculation. In addition, in order to avoid the potential case of out of boundary (beyond the reference frame border), during the process of pixel interpolation, this paper has expanded the reference frame's boundary by using the code to achieve pixel interpolation. As a result, it will not have the misjudgment problems any longer, due to out of boundary. The pixel luma component interpolation of AVS can be accurate to $1/4$ precision. The theory is relatively simple, and easy to implement.

6. Weighted prediction. The interpolation reference sample is needed to have the scale-shift processing, and have to cut samples to the range of from $0 \to 255$. This part is an optional module. According to the analytical grammar elements decoding from the bit stream, to determine whether use the weighted prediction.

2.3. Optimization of the inter prediction algorithm

The code analyzing tool Profile of VC Compiler is used to dissect the original decoding version, and video sequences CIF format as a test object. See Table 1, there is the eventually consuming time of each module taken up by the resource information in decoder.

From the table we can see that the major time-consuming decoder module are the entropy decoding module, inter prediction module, intra prediction module, inverse-transformation module, frame reconstruction module, which inter prediction module is one of the main reason to cause decoder's speed slow. These above data just reflect why the code inefficient is low in time. Then, with the code level completing the inter prediction algorithm, the optimization of it is also very important. The main topics of optimization focus on the two aspects, algorithm framework, variables and data structure. Furthermore, taking into account the linear assembly language optimization will also be done next.

2.3.1. Algorithm framework's optimization

As a decoder system in real time, our research is aimed at achieving the decoder. Here, it only needs to decode the I and P frames. While the related work of decoding of the type B frame in the whole system, the code could be streamlined out. Reference to dividing the frame into slices, it will reduce the complexity of computation and enhance the decoding speed. Make sure that the decoder no longer has the function of decoding the frame containing multiple slices, that is, the decoder only deals with the picture of a slice frame. In addition, decoder requires encoding video stream having no field-coding. Therefore, the process related with slice and the encoding operation associated with field could be removed from the decoder. As a result, this will streamline out a large number of condition transfer and loop sentences. Sequently, the probability of CPU flow process has been increased, and the speed of implementation of the software also has been improved obviously.

2.3.2. The variables and data structure's optimization

In the decoder system's variable definition list, there are so many useless and redundancy variables and data structures, for instance, some of the enumeration variable: PictureType, IntraInterDecision, BitCountType, etc. Inevitably, this will cause a waste of storage space and increase the size of the code. In the application of C code with the DSP, taking into account the limited resources of DSP memory space, these variables need to streamline out. Some types of the variables also need to re-type settings, for instance: it is unnecessary to define the matrix of depositing residual coefficient as an integer. Because the residual factor in the analysis by dequantization and inverse-transformation process is the greatest scope $\{-215, 215-65\}$. Therefore, the uses of short integer variables could be stored on a residual factor. This can save more memory space. In addition, optimizing data structures will also give more improvement. There are some data structures which set up a lot of variables. However, in fact, these variables are not in the applications of code (or after transplanting to DSP platform unnecessary to retain). If these structure variables continue to stay in the code, it is inevitable wasting the memory space. For example: structure variables of $Struct\_syntax\_element$ and $Struct\_inp\_par$ are useless, if the decoding system is transplanted to DSP platform. Therefore, after the adjustment transformation of the variables, there is no need to retain them, the same as the structure type $Struct\_snr\_par$. That is to say, after the overall framework optimization, there will be no necessary to reserve it.

Analysis of the code by CCS providing the analysis tools Profile, it is found that before the optimization the $Inter\_Luma\_Pred()$ function and $Inter\_Chroma\_Pred()$ function spend a lot of clock cycles in the inter prediction module. Total of the two algorithms occupy the whole clock cycles more than 90%. The main function of $Inter\_Luma\_Pred()$ function is to achieve the $1/2$ luma precision and $1/4$ luma precision of pixel interpolation. $Inter\_Chroma\_Pred()$ focuses on the operation of chroma component interpolation. The main reason of $Inter\_Luma\_Pred()$ function having much time-consuming is the conditional sentence and loop sentences in its sub-function $get\_block()$ too much. When calling this function, it needs to judge what the types of the pixel interpolation. Then according to the determined interpolation type of recycle for pixel sampling, filtering and cutting processing, the final results put into the appropriate block buffer of reference samples. This method of inter interpolation is very inefficient and enhances the speed of the system. With the phenomenon of a reference sample read many times, this proposal uses the two aspects to optimization the system.

Actually, in our research work, we just take into account two aspects to optimize the decoder system: one is algorithm framework, and the other is variables and data structure optimization. Besides, using linear assembly language to complete the original program of C language, this can be more flexible to arrange hardware resources, increase the utilization of system-wide computing unit and the data access, and efficiently achieve the goal of real-time decoder of decoding system.

3. Results and Discussion

This section will implement the optimized inter prediction algorithm and compare the results with the original inter prediction algorithm. We use C programming language to describe this optimized routine and CIF format as our benchmarks. Regarding to the video streaming of CIF format, decoder system could decode one frame with approximately $1211ms$ before the optimiza-

Table 2. Comparison Before And After Optimization

Module Name	Clock Number		Efficiency Ratio
	Former	Later	
$Inter\_prediction$	16790160	983444	$16.97\times$
I_one_frame	79006359	6531531	$11.09\times$
P_one_frame	72646329	4975450	$13.60\times$

tion. However, it just needs $99ms$ after the optimization. Obviously, between the former system and the optimized one, the efficiency has been improved by 91.2%. In addition, the comparison of the code efficiency between the optimized decoder and the former one is another symbol to discuss. That is, the consumption of CLOCK cycle number is different with the same decoding video stream. Here, we take the $Inter\_prediction$, I frame and P frame as the examples. Table 2 give the CLOCK number and efficiency ratio. Specifically, compared with the original algorithm, our algorithm improves the CLOCK number of $Inter\_prediction$, I_one_frame and P_one_frame, by 16.97, 11.09 and 13.60 respectively. Eq. 1 lists how to compute the efficiency ratio. The optimized decoder could decode about 10 frames of CIF format per second. With regard to the video streaming of QCIF format, it can basically meet the demand of real-time decoding. Meanwhile, the image quality of optimized decoder is as clear as the former decoder.

$$ER = \frac{T_1 - T_2}{T_2}; \quad (1)$$

In summary, the inter prediction algorithm as one of the key module in decoder, its implementation and optimization give a good insight of efficient improvement. This paper proposes an efficient and fast inter prediction algorithm. Meanwhile, the improvement of the whole algorithm framework and the optimization of variables and data structures have given a dramatically enhance of the decoding ratio and CLOCK cycle number. Reducing of the redundancy in source sequence inter prediction, it could also improve the picture quality.

4. Conclusion

This paper proposes an efficient and fast inter prediction algorithm, which could reduce the redundancy in source sequence inter prediction and improve the picture quality. Experimental results show that our optimized algorithm significantly $11.09 - 16.97\times$ efficient than the original inter prediction algorithm. In a word, there are many optimizations we could to do further, such as, writing parallel code, data processing package, the software pipelining, iteration unroll, eliminating redundant code and avoiding memory-access conflict.

5. Acknowledgments

The authors also would like to express their sincere thanks to the Digital Multimedia Technology Institute of UESTC. This work was supported by the Fundamental Research Funds for the Central Universities under Grant No. $ZYGX2011YB022$, in part by the National Science Foundation of China under Grant No. 60773205 and 973 Foundation $2010CB328004$.

References

[1] Yueqiang, LI, Breaking the Digital Video Steganography, TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol.11, No.3 (2013).

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Lu-thra, "Overview of the H.264/AVC Video Coding Standard, " IEEE Trans. Circuits and Systems for Video Technology , vol. 13, pp.560-576, (2003).

[3] Lu Yu, Sijia Chen, Jianpeng Wang, ”Overview of AVS-video coding standards,” Signal Processing: Image Communication 24 247-262 (2009).
[4] L. Yu et al, ”An Overview of AVS-Video: tools, performance and complexity”, Visual Communications and Image Processing 2005, Proc. of SPIE, vol. 5960, pp.596021, (2006).
[5] L. Yu et al, ”An area-efficient VLSI architecture for AVS intra frame encoder” Visual Communications and Image Processing 2007, Proc. of SPIE-IS & T Electronic Imaging, SPIE vol. 6508, pp. 650822, Jan. 29, (2007).
[6] W. Gao et al, ”AVS - The Chinese Next-Generation Video Coding Standard” NAB, Las Vegas, (2004).
[7] B. Tang et al, ”AVS Encoder Performance and Complexity Analysis Based on Mobile Video Communication”, WRI International conference on Communications and Mobile Computing, CMC ’09, vol. 3, pp. 102-107, 6-8 Jan. (2009).
[8] K. R. Rao, IEEE Fellow, and Do Nyeon Kim, ”Current Video Coding Standards: H.264/AVC, Dirac, AVS China and VC-1,” 42nd South Eastern Symposium on System Theory University of Texas at Tyler Tyler, TX, USA, March 7-9, (2010).
[9] L. Yu et al, ”An Overview of AVS-Video: tools, performance and complexity”, Visual Communications and Image Processing 2005, Proc. of SPIE, vol. 5960, pp.596021, July 31, (2006).
[10] Jia X. and Sun Y., A Kind of Visual Speech Feature with the Geometric and Local Inner Texture Description, TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol.11, No.2, 877-889, 2013.
[11] L. Yu et al, ”An area-efficient VLSI architecture for AVS intra frame encoder” Visual Communications and Image Processing 2007, Proc. of SPIE-IS & T Electronic Imaging, SPIE vol. 6508, pp. 650822, Jan. 29, (2007).
[12] W. Gao et al, ”AVS - The Chinese Next-Generation Video Coding Standard” NAB, Las Vegas, 2004.
[13] B. Tang et al, ”AVS Encoder Performance and Complexity Analysis Based on Mobile Video Communication”, WRI International conference on Communications and Mobile Computing, CMC ’09, vol. 3, pp. 102-107, 6-8 Jan. (2009).
[14] Haskell, B.G., Puri, A. and Netravali, A.N., Digital video: an introduction to MPEG-2, Springer, (1996).
[15] Bosi, M., Brandenburg, K., Quackenbush, S. and Fielder, L. et al, ISO/IEC MPEG-2 advanced audio coding, Proc. 101 AES Conv, (2012).
[16] Assuncao, P.A.A. and Ghanbari, M., A frequency-domain video transcoder for dynamic bitrate reduction of MPEG-2 bit streams, Circuits and Systems for Video Technology, IEEE Transactions on, 09, vol.8, pp. 953-967, (1998).
[17] Wiegand, T., Sullivan, G.J., Bjontegaard, G. and Luthra, A., Overview of the H. 264/AVC video coding standard, Circuits and Systems for Video Technology, IEEE Transactions on, 07, vol. 13, pp. 560-576, (2003).
[18] Schwarz, H., Marpe, D. and Wiegand, T., Overview of the scalable video coding extension of the H. 264/AVC standard, Circuits and Systems for Video Technology, IEEE Transactions on, 17, vol.9, pp.1103-1120, (2007).
[19] Michael A. Nielsen,Isaac L. Chuang, Quantum computation and quantum information, (2010).
[20] Schumacher B. Quantum coding[J]. Physical Review A, 51(4): 2738, (1995).
[21] Ambainis A, Nayak A, Ta-Shma A, et al. Dense quantum coding and quantum finite automata[J]. Journal of the ACM (JACM), 49(4): 496-511, (2002).
[22] Ambainis, A., Nayak, A., Ta-Shma, A., and Vazirani, U. Dense quantum coding and quantum finite automata. Journal of the ACM (JACM),49(4), 496-511, (2002).
[23] Bin Ji, The Application of Normalized Covariance in Video Mosaicing, TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol.10, NO.7, 1567-1572, (2012).
[24] Zhang T. and Tu, X. and Wang X. and Zha X., Quantum State Sharing of An Arbitrary Three-qubit State Using Two Four-qubit Cluster States, TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol.11, No.5, 1978-1983, 2013.