An efficient adaptive arithmetic coding image compression technology*

Wang Xing-Yuan(王兴元)†, Yun Jiao-Jiao(云娇娇),‡ and Zhang Yong-Lei(张永雷)

Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, *China*

(Received 23 February 2011; revised manuscript received 21 April 2011)

This paper proposes an efficient lossless image compression scheme for still images based on an adaptive arithmetic coding compression algorithm. The algorithm increases the image coding compression rate and ensures the quality of the decoded image combined with the adaptive probability model and predictive coding. The use of adaptive models for each encoded image block dynamically estimates the probability of the relevant image block. The decoded image block can accurately recover the encoded image according to the code book information. We adopt an adaptive arithmetic coding algorithm for image compression that greatly improves the image compression rate. The results show that it is an effective compression technology.

Keywords: arithmetic coding, adaptive, image compression

PACS: 42.30.Va, 42.30.Wb, 87.57.C– **DOI:** 10.1088/1674-1056/20/10/104203

1. Introduction

With the rapid development of digital media technology and its wide application, the question of how to reduce digital storage and the amount of data required in communications is becoming a hot topic, and this important issue needs to be resolved. As digital images take up increasingly larger volumes of data, image compression technology has become the key technology in related applications.

The concept of arithmetic coding (AC) was proposed by Elias in the early 1960s, however, it only began to enter the practical stage in the 1980s. Langdo and Rissanen[1,2] undertook a great deal of work on the Joint Photographic Experts Group (JPEG) and JPEG-LS image data compression standard. AC is a lossless compression technology which provides an effective mechanism to remove redundant information to aid data compression. Huffman coding is another well-known coding algorithm, but the efficiency of AC is 10% higher than Huffman coding, so it is the most efficient statistical entropy coding method so far. However, due to the coding complexity and some implementation technique restrictions, and the limitations of technological realization and patent rights, its application is not very extensive. Currenly, some countries have made a vast investigation of AC and obtained some important applications. For example, Q coder, the patent of IBM, has been applied in the Joint Photographic Experts Group (JPEG), JPEG2000 and the Joint Bi-level Image Experts Group (JBIG). AC, was studied by Witten *et al.*[3] and also featured in H.263 and H.264, but most have patent protection. In contrast, research on AC is rather limited.

AC needs to work with a modeler. The modeler[4] estimates the probability of each pixel at each iteration of the encoding and decoding processes. It also needs to find the probability distribution of those symbols, which maximizes compression efficiency, and sends this information to the decoder as header information ensured decidability. The models can be static or adaptive. When AC works with an adaptive model, it is usually referred to as adaptive AC[5] (AAC).

Recently, many AAC-oriented coding schemes have been proposed in the literature. Golchin and Paliwal[6] proposed an adaptive lossless image encoder technology that combines a context classification scheme with adaptive prediction and entropy

*Project supported by the National Natural Science Foundation of China (Grant Nos. 60573172 and 60973152), the Superior University Doctor Subject Special Scientific Research Foundation of China (Grant No. 20070141014), and the Natural Science Foundation of Liaoning Province of China (Grant No. 20082165).

†Corresponding author. E-mail: wangxy@dlut.edu.cn

‡E-mail: yunjiaojiao999@126.com

© 2011 Chinese Physical Society and IOP Publishing Ltd

coding. Matsuda *et al.*[7] proposed a lossless coding scheme using a block-adaptive prediction technique to remove redundancy. In 2004, Kuroki *et al.*[8] presented an AAC with prediction errors in lossless image compression. They proposed a model which estimated the probability density of each error pixel by Laplacian distribution with zero mean. In 2007, Veeraswamy *et al.*[9] predicted five AC coefficients. In the next year, he described an image AAC technology based on AC coefficient DHT (discrete hadamard transform) predictions.[10] In recent years, more and more research[11−13] on AC coefficient prediction, modeler application and encryption schemes based on AC have been made. In addition, the C has also been applied to nonlinear dynamics[14] and the famous chaotic system.[15]

Thus, AC has important applications in image compression.[16,17] How to further improve the compression rate has become a problem that requires urgent answers.[18,19] So we propose an efficient lossless image compression scheme for still images based on the AAC compression algorithm. The algorithm increases the image coding compression rate and ensures the quality of the decoded image.

The rest of this paper is organized as follows. Conventional AC (CAC) is introduced in Section 2. The improved AC, AAC is presented in Section 3. Section 4 shows the experimental results and future work. Finally, the conclusion and the application prospects of AC are given in Section 5.

2. The CAC

In this section, we present CAC. AC is a lossless compression coding method, it can compress the source redundancy effectively and make its encoding rate tend to the entropy of the source. The main idea of AC is to map the encoded information to a real number between 0 and 1, then divide [0, 1) into several intervals, each value on this interval can uniquely identify the encoded data. This interval is closed on the left, denoted by "[", and open on the right, denoted by ")". Each sub-interval can be represented as a real number, but the precondition for distinguishing the sub-interval is that the real number must be precise enough, that is, there must be a sufficient number of bits. A symbolic sequence series composed by real numbers is the corresponding code word. Obviously, the greater probability of the symbolic sequence occurring is and the wider the sub-interval is, the fewer number of bits there are and the shorter the corresponding code word is.

A classic AC is Elias coding. In the Elias coding, for binary value symbol sequence "0" or "1", the current probability space is divided into two sub-intervals in terms of the respective ratio of probability, the set of lower limit generated by the probability interval makes up code values. The code values are revised one by one according to the input of binary symbols. Figure 1 shows the representation of Elias code:

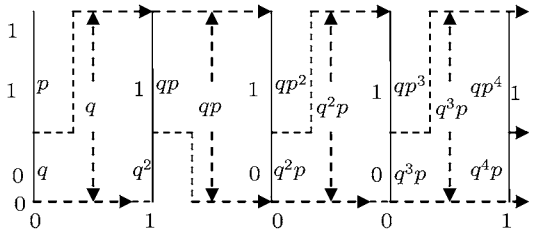


Fig. 1. Elias code.

Suppose the input is P 1, Q 0, and the probability of 0 and 1 is p, q, respectively, then the average code length is

$$L = [-\log_2 p^P q^Q] = [-P\log_2 p - Q\log_2 q]. \quad (1)$$

If the times of p or q are greater than a specified number, then we let $A = A - A \times Qe$, $q = Q/N$, so

$$L = N \times [-p\log_2 p - q\log_2 q], \quad (2)$$

where A is the interval length register, which is used to store the subinterval length. So we can find that the average code length L/N is almost equal to the entropy of a symbol. According to Shannon's noiseless source coding theorem: the algorithm is close to an ideal entropy coding algorithm.[4]

For conventional data compression, the most important benefits of AC are its optimality and independent internal encoding. That is, it is strictly optimal for a random data source whose probability is determinate. It does not require the probability of source symbols, in another words, it does not depend on the prior probability of each symbol. It can adaptively adjust its probability estimated value based on an appropriate probability model and the current frequency of each symbol. So it is more flexible in comparison with Huffman coding. Another advantage of AC is that it is easy to maintain the dictionary order under the condition that it does not affect the compression

efficiency, so the encoded code has the same order as the non-encoded data. In image compression, that is, encoding image information in the original image sub-block sequence strictly avoids the scrambling of sub-blocks.

However, AC also has its disadvantages. Since AC is a finite-precision encoding algorithm, with an increase in the input number, an increase in the precision of decimal places arises. In addition, the lack of a prefix code means that the error resilience of the bit stream decreases (the reason for error: the last bits of a bit stream will be truncated in the encoding process where an error code will appear in the decoding process). We propose an AAC compression algorithm in order to resolve this problem.

3. Adaptive arithmetic coding

3.1. AAC

CAC such as the Elias code considers only the total statistical property of the original file, that is the probability of the occurrence of each symbol in L/N non-encoded symbols s_j, without considering the correlation between each symbol in the source. However, in practice, many information sources are highly correlated, such as in image compression, image sub-block pixels demonstrate very strong correlation both in the contrast and the brightness, which reduces the performance of AC. However, AAC makes up for the weaknesses of the Elias code in this area. At the same time AAC is an algorithm with finite precision, the accuracy of the algorithm does not increase with the augment of number of encoded symbols, which is easy to be realized by a computer's hardware. In addition, the algorithm does not distinguish one source from the another by "0" and "1", but divides the interval by the MPS (more probable symbol) and the LPS (less probable symbol). Moreover, the precondition of dividing the current probability interval into two sub-intervals is that the MPS subinterval can be configured to the LPS subinterval. Therefore, if the symbol is an MPS, when we attach the LPS to the code, let the width of the probability interval be that of the MPS; if the symbol is an LPS, the width of the probability interval is that of the LPS. Repeat recursively as judgment for input.

First, setup two registers A and C, A denotes the interval length register used to store the subinterval length, C denotes the code register used to store the mantissa of the code stream.

Let Q_{e} be the probability evaluation of the LPS. When the input symbol is judged to be an MPS:

$$C = C + A \times Q_{\mathrm{e}}, \quad A = A - Q_{\mathrm{e}}.$$

When the input symbol is judged to be an LPS:

$$C = C, \quad A = A - A \times Q_{\mathrm{e}}.$$

For simplicity, we make a renormalization and condition exchange about the above formula. The purpose of renormalization is to ensure the fixed accuracy of the sub-interval and to avoid its length being too small, and make it remain at [0.75, 1.5]. If $A < 0.75$, A and C are multiplied by 2 until the interval length register is greater than 0.75. Meanwhile, in order to avoid overflow of C, the C register shifts left one bit periodically on the high-order. Condition exchange is a transition when the length of the MPS interval is less than the LPS interval, its purpose is to make the interval length LPS a constant approximated to 1, then there is the formula below.

When the input symbol is judged to be an MPS,

$$C = C + Q_{\mathrm{e}}, \quad A = A - Q_{\mathrm{e}};$$

when the input symbol is judged to be an LPS,

$$C = C, \quad A = A - Q_{\mathrm{e}}.$$

Currently, the AC has an important application in image compression. We know that there is a strong context correlation when the gray values of the current image in pixels closely resemble each other or are perfectly equivalent. AAC obtains compressed data based on the preprocessing of an inputted image and context information provided by a bit plane encoder (BPE), as shown in Fig. 2.

The JPEG uses the probability estimation state machine to provide a probability evaluation and the conversion relationship under different states according to the encoded input and context information. During the encoding process, each state S corresponds to a probability $Q_{\mathrm{e}}(S)$ of the LPS, a next-index-MPS (S) (i.e. if the input is an MPS, the state machine moves to the index value of new state) and a next-index-LPS (S) (i.e. if the input is an LPS, the state machine moves to the index value of the new state). The JPEG also stipulates that state transition can be carried out only in the case of renormalization.

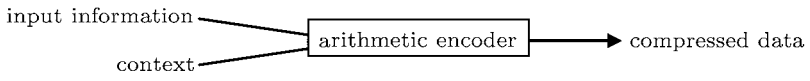


Fig. 2. Adaptive arithmetic coding.

When several great probability symbols appear in a row, $A > 0.75$, the MPS needs to be renormalized. If the previous state is S_1, the state machine moves to a new state S_2. If $Q_e(S_2) < Q_e(S_1)$ at this moment, the adaptive adjustment of probability estimation is achieved. Whereas, when several small probability symbols appear in a row, the state machine moves to the greater state $Q_e(S)$. During the movement process, if the probability of an LPS is greater than that of an MPS, switch the corresponding symbol of the LPS and that of the MPS. Then the probability estimated value Q_e of LPS will change correspondingly. Figure 3 displays subinterval flowchart of an arithmetic encoder.

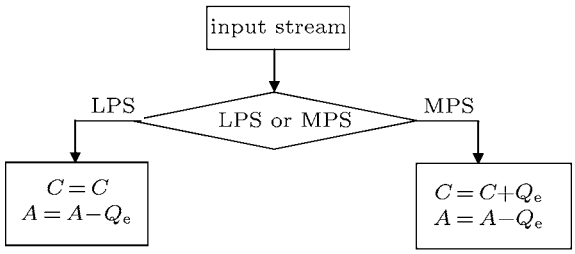


Fig. 3. Subinterval flowchart of arithmetic encoder.

3.2. Proposed approach

Our AAC involves two major processes: establishing the probability model and encoding.

(I) Establishing the probability model: handle the pixels of an image as one-dimensional data $X_i = x_1 x_2 \ldots x_i$ by time series. The purpose of the model is to deduce $X_i + 1$ by putative conditional probability distribution $P(.|X_i)$.

In our lossless compression scheme, probability assignment is divided into three parts:

i) prediction, given a prediction $X_{X_i}+1$ of X_i+1 through a subset of X_i,

ii) determine the context of $X_i + 1$, which is usually a function of encoded subsequence,

iii) establish the probability model of prediction errors $E_i + 1 = X_i + 1 - X_{X_i} + 1$ of the context of $X_i + 1$.

(II) Encoding: first, the original image with size of $M \times N$ is divided with fixed segmentation, the sizes of the resulting image blocks are $M \times N/k^2$, $k = 1, 2, \ldots, \min(M, N)$. Then we employ a kind of multi-symbol AC (MSAC) called a range coder on AC bases instead of the variable-length code in encoding processing. A probability table used in the range coder is dynamically given to realize the context of the image blocks. The set of thresholds $\{T_1, T_2, \ldots, T_m\}$ are predisposed to limit the range of context information, then take AAC by using prediction errors. Since the range order requires no floating-point, in addition to the high efficiency and simpleness of AAC by using prediction errors, it can work faster than CAC methods.

4. Experimental results

4.1. Evaluation criteria of image compression performance

I) Compression ratio: the essence of image compression coding is to represent an image with the minimum bits under a certain condition. In order to assess the compression efficiency of an algorithm, we adopt compression ratios that are defined as follows:

$$C_r = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} r_b(i,j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} r_c(i,j)}, \tag{3}$$

where r_b is the bits per pixel of original image, r_c is the bits per pixel of the compressed image.

Usually the compression ratio is specified by 8 bit/pixel, in other words, after pulse code modulation (PCM), the quantification bit number $n = 8$ with non-compression is encoded, the average number of bits per pixel is r_c, so formula (1) can be expressed as

$$C_r = \frac{8}{r_c}. \tag{4}$$

II) Mean square error: generally, the criterion to evaluate the quality of a compressed image is the fidelity criterion. There are two kinds of fidelity criterion: objective fidelity criterion and subjective fidelity criterion. The former is measured by the error of non-compressed and compressed images; the latter depends on the subjective feeling. Here, we adopt an often used global objective fidelity criterion, the mean

square error is defined as

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [g(x,y) - f(x,y)]^2, \qquad (5)$$

where $M \times N$ is the total number of pixels and $g(x,y)$ and $f(x,y)$ are the pixel values in the original and reconstructed image.

4.2. Experimental results

We apply our AAC method to 8 bit/pixel gray images with size of 512 × 512. Four images, Goldhill, Lena, Pepper and Zelda in Fig. 4, are suitable for evaluating the test effect of the algorithm to the images with different features since they have much correlation in content. Experiments start at the block with size of 8 × 8 by fixed segment and are compared with the AAC0 method and the AAC method in Ref. [5] to explore the advantages and limitations of our method. The experimental simulation is done on a PC with a Windows XP operation system and an AMD/Athlon 1.6 GHz processor.

As a sample, figure 5 shows the decoded image by using our AAC method. Figures 5(a), 5(b), 5(c) and 5(d) correspond to Figs. 4(a), 4(b), 4(c), and 4(d), respectively.



Fig. 4. Original image.



Fig. 5. Decoded image.

4.3. Analysis of experimental results

Table 1 shows the comparison of results between our AAC and AAC0 and the latest, representative AAC method formulated by French scholors Atef *et al.* in Ref. [5]. From Table 1 we can see that the compression ratios for our AAC are between 0.21% and 1.998%, which are higher than those obtained by AAC in Ref. [5], and it outperforms the AAC in Ref. [5] by a factor of 1.203% on average. Similarly the compression ratios for our AAC are between 2.009% and 13.464%, higher than those obtained by AAC0, and it outperforms AAC0 by a factor of 5.273% on average. Theoretical analyses and the results indicate an improvement of compression performance over two other approaches. However, all the values of MSE are 0, showing the decoded image without any quality loss. So we can say that this algorithm realizes the lossless compression of an image.

We have already improved the AC algorithm based on an adaptive modeler, but it still limits the large quantity of calculation in AC algorithm to some degree, so we will continue to learn about AAC and to reduce its compression time.

Table 1. Comparison result between our AAC and AAC0 and AAC in Ref. [5].

Original images	AAC0	AAC in Ref. [5]	Our AAC			
	Compression ratio	Compression ratio	Compression ratio	Gain (AAC in Ref. [5])/%	Gain (AAC0) /%	MSE
Barbara	1.07	1.096	**1.102**	0.588	2.903	0
Boat	1.122	1.133	**1.145**	1.048	2.009	0
France	1.234	1.423	**1.426**	0.21	13.464	0
Goldhill	1.069	1.102	**1.121**	1.69	4.639	0
Lena	1.073	1.117	**1.134**	1.499	5.379	0
Peppers	1.056	1.079	**1.101**	1.998	4.087	0
Zelda	1.1	1.135	**1.151**	1.39	4.431	0
Average	**1.104**	**1.155**	**1.169**	**1.203**	**5.273**	**0**

Table 2 lists the PSNR results of all test images when the compression ratio is 1.100. As shown from the results, the rate-distortion performance of the real wavelet transform (RWT)-version coders, regardless of the set partitioning in hierarchical trees (SPIHT) or EBCOT algorithm, is always superior to their integer wavelet transform (IWT)-version counterparts, due to the excellent energy clustering ability provided by the RWT. For our AAC algorithm, the adaptive encoding makes it exhibit the best performance on average among all test coders. The average PSNR increasing ratio of the AAC over the SPIHT and embedded block coding (EBCOT) are 5.52% and 4.52% for all images, respectively. Compared with the F-Biliner and F-Bicubic method in Ref. [20], the PSNR improvements of the AAC Lena image is higher 1.80 dB and 1.84 dB, respectively. The bar graph, Fig. 6, is a visualized reflection of the comparison of results. The number of x labeled in Fig. 6 corresponds to the number in Table 2, respectively.

Table 2. Comparison result of PSNR between our AAC and SPITH and EBCOT.

Number	Test image	RWT-SPITH	IWT-SPITH	RWT-EBCOT	IWT-EBCOT	AAC	Average increasing ratio (SPITH, EBCOT)
1	Barbara.tif	31.40	30.93	32.29	31.48	33.32	6.92%, 4.51%
2	Boat.tif	33.01	32.87	33.20	32.97	33.89	2.88%, 2.43%
3	France.tif	31.68	31.57	32.68	32.57	33.56	5.82%, 2.86%
4	Goldhill.tif	30.56	30.47	30.59	30.38	32.33	5.95%, 6.05%
5	Lena.tif	31.11	31.02	31.05	30.97	33.57	8.06%, 8.25%
6	Zelda.tif	33.59	33.21	33.70	33.40	34.56	3.48%, 3.01%

The MQ arithmetic coder has been adopted to achieve entropy coding in the latest image compression standard JPEG2000, which is a bit-level operation with an intensive branch and feedback, and thus becomes a serious bottleneck of high speed JPEG2000. In this paper, a comparison the run time results between our AAC and MQ and proposed MQ in Ref. [21] is shown in Table 3. Experimental results as shown in Table 3 demonstrate the efficiency of the proposed method in reducing the computation time of our AAC method. It also indicates that an average 5.75% reduction ratio of run time can be obtained. The bar graph, Fig. 7, is a visualized reflection of the comparison of results. The number of x labeled in Fig. 7 corresponds to the number in Table 3, respectively.

Table 3. Comparison of results of run time between our AAC and MQ and proposed MQ in Ref. [21].

Number	Test image	MQ	Proposed MQ	Our AAC	Reduction ratio
1	Barbara.tif	0.335	0.308	0.291	5.52%
2	Boat.tif	0.322	0.296	0.26	12.16%
3	France.tif	0.321	0.297	0.282	5.05%
4	Goldhill.tif	0.343	0.315	0.296	6.03%
5	Lena.tif	0.279	0.252	0.242	3.97%
6	Zelda.tif	0.235	0.208	0.206	1.96%

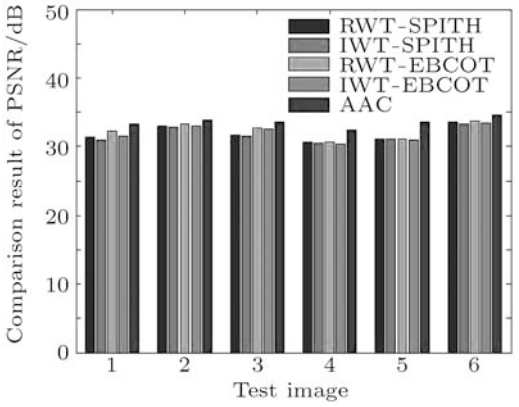


Fig. 6. Comparison result of PSNR.

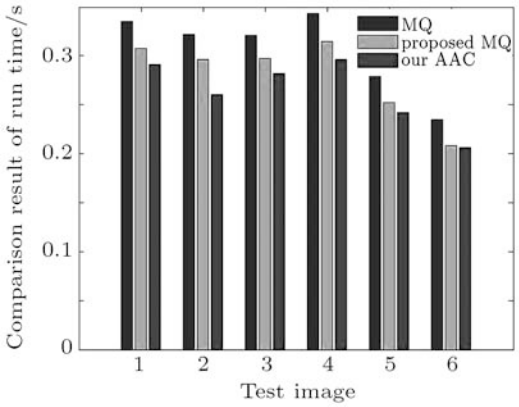


Fig. 7. Comparison result of run time.

5. Conclusion

In this paper, we proposed an efficient lossless image compression scheme—the AAC compression algorithm. The algorithm increases the image coding compression rate and ensures the quality of the decoded image by combining the adaptive probability model with predictive coding. The use of adaptive models for each encoded image block dynamically estimates the probability of the relevant image block. The decoded image block can accurately recover the information according to the code book of the encoded image.

AC is an efficient lossless compression scheme, as a result of AAC for image compression, it can greatly improve the image compression rate. The experimental results show that it is an effective compression technology. With continuous improvement of AAC technology, it will break the shackles of patent permissions and gain much wider applications.

References

[1] Langdon G G and Jorma B 1981 *IEEE Trans. Commun.* **29** 858

[2] Langdon G G and Jorma B 1983 *IEEE Trans. Commun.* **31** 1253

[3] Witten I H, Neal R M and Cleary J G 1987 *Commun. ACM* **30** 520

[4] Zhang H Z and Hu Y Z 2007 *Chin. J. Sci. Instrum.* **28** 385 (in Chinese)

[5] Atef M, William P and Mohamed S B 2010 *J. Electron. Imaging* **19** 1

[6] Golchin F and Paliwal K. K 1998 *IEEE ICASSP* **5** 2545

[7] Matsuda I, Shirai N and Itoh S 2003 *Lect. Notes Comput. Sci.* **2849** 199

[8] Kuroki N, Manabe T and Numa M 2004 *IEEE ISCAS* **3** 961

[9] Veeraswamy K, Mohan B C and Yothirmayi T J 2007 *IEEE CS* **3** 25

[10] Veeraswamy K and Srinivas K S 2008 *J. Multimedia* **3** 16

[11] Gopal L 2008 *IEEE Trans. Circ. Syst. Vid.* **18** 791

[12] Duan L L, Liao X F and Xiang T 2010 *Commun. Nonlinear Sci. Numer. Simul.* Accepted

[13] Li H J and Zhang J S 2009 *Commun. Nonlinear Sci. Numer. Simul.* **14** 4304

[14] Nithin N, Prabhakar G V and Kishor G B 2009 *Commun. Nonlinear Sci. Numer. Simul.* **14** 1013

[15] Mi B, Liao X F and Chen Y 2008 *Chaos, Solitons and Fractals* **38** 1523

[16] Pang C Y, Zhou Z W and Guo G C 2006 *Chin. Phys.* **15** 3039

[17] Pang C Y, Zhou Z W, Chen P X and Guo G C 2006 *Chin. Phys.* **15** 618

[18] Long G L and Xiao L 2004 *Phys. Rev.* A **69** 052303

[19] Ren H P, Ping Z L, Bo W R G, Sheng Y L, Chen S Z and Wu W K 2003 *Chin. Phys.* **12** 610

[20] Xu Y, Dong J T and Wang S H 2010 *Acta Phys. Sin.* **59** 7535 (in Chinese)

[21] Gao Z R, Xiong C Y and Hou J H 2007 *Proc. SPIE* **6787** 1