行政院國家科學委員會專題研究計畫　成果報告

以可調式內插法實作漸進式影像傳輸

計畫類別：個別型計畫
計畫編號：NSC92-2213-E-032-025-
執行期間：92 年 08 月 01 日至 93 年 07 月 31 日
執行單位：淡江大學資訊工程學系

計畫主持人：葛煥昭
共同主持人：施國琛

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 93 年 8 月 26 日

行政院國家科學委員會專題研究計畫成果報告

以可調式內插法實作漸進式影像傳輸

計劃編號：NSC 92-2213-E-032 -025
執行期限：2003 年 8 月 1 日至 2004 年 7 月 31 日
主持人:葛煥昭　共同主持人：施國琛　淡江大學資訊工程學系

中文摘要

漸進式影像傳輸(Progressive Image Transmission)是一種圖片傳輸及顯示的機制，它可以一邊傳輸、一邊顯示，將已經傳輸的結果顯示，並且以近似原圖的方式顯示，使得接收端的使用者能在圖片仍未完全傳輸完成的情形下即可預覽圖片。我們以可調式內插法為基礎，透過對影像色彩分佈的特徵分析，只要傳輸少量的影像資訊，就可以使得網路接收端，快速瀏覽整張圖片的大概內容，隨著傳輸回合數的增加，將會有更好的影像品質。這個影像傳輸的技術，是運用影像色彩分佈的特徵與影像中相鄰像素間的關係，使用已經接收到的影像資訊，猜測下一個傳輸目標像素的內容，其實際上這個目標像素是不必傳輸的；此技術大大減少影像的傳輸量，且針對像素有可能的猜測錯誤的問題，提出一個有效的修正方法。

關鍵詞： 漸進式影像傳輸，影像編碼，像素內插法，BPM

ABSTRACT

Progressive image transmission is a mechanism that transmits the most significant portion of an image, followed by its less important parts. Applications of such a mechanism include browsing large image files on the Internet. We propose an adaptive mechanism, based on the characteristics of images. The mechanism use neighbor pixels to guess a target pixel value, without actually transmitting the target pixel. An error correction scheme is also designed to cope with a failure guessing. The prototype is tested on 1500 bit-mapped pictures of different categories. Preliminary results should that the transmission rate is lower than others, with reasonable PSNR values of the transmitted images. Interested readers can find the prototype tool and our evaluations at *http://www.mine.tku.edu.tw/demos/ProgTransmission*.

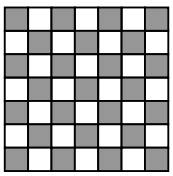
Key words: Progressive Image Transmission, BPM, Pixel Interpolation, Image Coding, Network Applications

1. INTRODUCTION

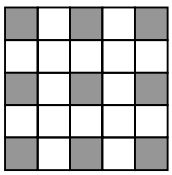
Progressive Image Transmission (PIT) is a mechanism which transmits the most significant potion of an image, followed by the relatively less significant potions. Available mechanisms and systems can be categorized into transform domain [1], spatial domain [2], and pyramid-structured progressive transmission [4]. A germinal and instinctive method for progressive image transmission in the spatial domain is the bit plane method (BPM) [5]. The BPM is a simplest and faster method, which does not have any encoding processes and complex transmission algorithms. In the method, the transmitter transmits one bit for each pixel in each round and the transmitted bits are coursed from the most significant bit (MSB) to the least significant bit (LSB). The receiver receives either a '0' or a '1' for each pixel and reconstructs each pixel using the mean as the predicted value. In our earlier approach [3], we improved the BPM method by color guessing, to provide a fast PIT scheme. In this paper, we propose a new method, and experiment with a complete test on different types of images, the multiple sizes of pixel blocks, and the combination of transmission orders. We have tested 1500 pictures. Conclusion shows that, with different strategies, the method can be used as an adaptive mechanism, which has a very low bit rate and good PSNR values.

2. INTERPOLATION OF PIXELS

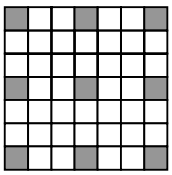
In our earlier approach [3], we interpolate pixel colors by guessing average values of neighborhood pixels. The approach has a better result compared to the BPM approach. However, the average bit rate of our method after 4 steps (from the total of 8) of transmission is about 85.75% compared to the BPM. The average PSNR value at the 4^{th} step is the same to BPM due to our error correction scheme. In this paper, we propose a new pixel interpolation scheme, which adapts different characteristics of pictures, and allows an adaptive color interpolation. For instance, a cartoon picture has very high pixel continuity, which is different from a photo. Using our earlier approach, the transmission rate is still high. Using the newly proposed scheme, the transmission rate is much lower.



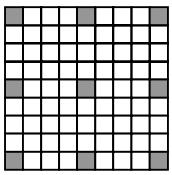
(a) GBN



(b) D = 2



(c) D = 3



(d) D = 4

Figure 1: Interpolation matrices of different schemes

Figure 1(a) shows the interpolation matrix of our earlier approach (Guessing by Neighbors, or GBN). The average transmission rate is about 50% (transmitted pixels are shown in dark boxes). However, overheads are necessary in the error correction process. In figure 1(b), 1(c), and 1(d), we illustrate the positions of the transmitted pixels. The transmission rates are lower (but the overheads are higher). We define the *distance index*, D, to be the number of pixels in between two consecutive pixels transmitted, plus one. In general, a larger D should be used for a picture with higher pixel coherence. We briefly discuss our GBN method. An image is divided into two streams, the even stream and the odd stream. Even streams are transmitted, and odd streams are "guessed." In case of a miss-guessing, we have an error connection syndrome followed by the actual odd stream elements. The concept relies on the same guessing function is applied to both the transmission and the receiving sides. Thus, error recovery can be made on the receiving side based on an error syndrome [3]. The guessing function has two levels. The first level guesses the pixel value based on an average of neighbor pixels (the pixels above, below, to the left, and to the right). In case of a tie situation (i.e., the average value is exactly between two integers), a second level guess function is used. The second level function looks at two additional pixels to the upper-left and upper-right direction w. r. t. the pixel being guessed. Arbitrary selection is used in case of a second level tie. The algorithm runs reasonably well.

In the newly proposed methods, we use the same error recovery procedure. But, the guess function uses a multiple resolution pixel box. Table 1 illustrates a pixel box of the size $\boldsymbol{n}$ by $\boldsymbol{m}$. In a normal situation, $\boldsymbol{n}$ is equal to $\boldsymbol{m}$. The pixels transmitted are $\boldsymbol{P}_{1,1}$, $\boldsymbol{P}_{1,m}$, $\boldsymbol{P}_{n,1}$, and $\boldsymbol{P}_{n,m}$ (i.e., the pixels on the four corners).

Table 1: A pixel box of size *(n, m)*

$\boldsymbol{P}_{1,1}$	$\boldsymbol{P}_{1,2}$	…	$\boldsymbol{P}_{1,j}$	…	$\boldsymbol{P}_{1,m}$
$\boldsymbol{P}_{2,1}$	…	…	…	…	$\boldsymbol{P}_{2,m}$
…	…	…	…	…	…
$\boldsymbol{P}_{i,1}$	…	…	$\boldsymbol{P}_{i,j}$	…	$\boldsymbol{P}_{i,m}$
…	…	…	…	…	…
$\boldsymbol{P}_{n,1}$	$\boldsymbol{P}_{n,2}$	…	$\boldsymbol{P}_{n,j}$	…	$\boldsymbol{P}_{n,m}$

There are two steps to perform a guess function. In addition to the 4 corner pixels, a pixel box has *surrounding pixels* and *interior pixels*. A surrounding pixel is located on the four boundaries of a pixel box. Other pixels are all interior pixels. To compute the guess values of surrounding pixels, we use:

$$\forall 1 < i < n,\ P_{i,1} = \alpha_1 * P_{1,1} + \beta_1 * P_{n,1}$$
$$\forall 1 < i < n,\ P_{i,m} = \alpha_1 * P_{1,m} + \beta_1 * P_{n,m}$$
$$\forall 1 < j < m,\ P_{1,j} = \alpha_2 * P_{1,1} + \beta_2 * P_{1,m}$$
$$\forall 1 < j < m,\ P_{n,j} = \alpha_2 * P_{n,1} + \beta_2 * P_{n,m}$$

where $\alpha_1 = (D - i + 1) / D$, $\beta_1 = (i - 1) / D$,
$\alpha_2 = (D - j + 1) / D$, $\beta_2 = (j - 1) / D$, and
D is the distance index

To compute the guess value of interior pixels, we use:

$$\forall 1 < i < n,\ \forall 1 < j < m,$$
$$P_{i,j} = ((\alpha_1 * P_{1,j} + \beta_1 * P_{n,j}) + (\alpha_2 * P_{i,1} + \beta_2 * P_{i,m})) / 2$$

As an example, when D=3, the above equations can be reduced to values in table 2.

Table 2: A pixel box of size *(4, 4)* and D=3

a	ab1= 2/3a+1/3b	ab2= 1/3a+2/3b	b
ac1= 2/3a+1/3c	(2/3ac1 +1/3bd1 +2/3ab1 +1/3cd1)/2	(1/3ac1 +2/3bd1 +2/3ab2 +1/3cd2)/2	bd1= 2/3b+1/3d
ac2= 1/3a+2/3c	(2/3ac2 +1/3bd2 +1/3ab1 +2/3cd1)/2	(1/3ac2 +2/3bd2 +1/3ab2 +2/3cd2)/2	bd2= 1/3b+2/3d
c	cd1= 2/3c+1/3d	cd2= 1/3c+2/3d	d

Thus, the 4 corner pixels are used as the base of the guessing function. When these pixels are transmitted, in each step, only a few bits are transmitted in a pixel. The number of bits depends on the format of the picture (e.g., 8-bit gray level image, or 24-bit true color image). One important concept is, on the receiver side, after a transmission step, the transmitted bits are re-calculated, with a correction procedure. Thus, previous bits of a pixel transmitted will not be changed in current step. In the article [3], we have detailed examples.

It is also important to point out our error correction procedure. We define an *error correction stream*, which consists of two parts: the *error syndrome* and the *recovery data stream*. An error syndrome is a bit stream, of the length the same as the number of pixels in the non-transmitted stream. For instance, for D=3, the length of error syndrome is equal to 12 since only 4 pixels are transmitted. In the syndrome, a '1' represents an error guess and a '0' otherwise. A recovery data stream contains

original bit values, whose values are incorrectly guessed, in a respective order. For instance, if an error connection syndrome = 01100100 111110, the error guesses are in position 2, 3, and 6, with the original data 11, 11, and 10, respectively (we assume that two bits out of 16 are transmitted each time). Note that, the number of bits is 14, which is smaller than the total length 16. In general, if the occurrence of errors is less than 50%, transmission volume is saved. Due to the continuity of most image pixels, it is very likely that the error rate is less than 50 percents. The recovery process simply takes the positions of '1's in the error syndrome, and over write the image with the recovery data stream in the relative position. After an iteration step, the stream bit-pairs in the receiver side should be the same as those in the original image.

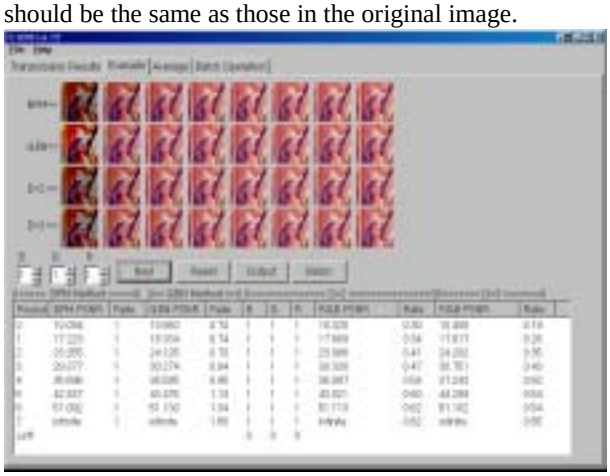


Figure 2: A prototype tool of PIT

Table 3: Simulation Results based on 1500 Images

	BPM		GBN Method		New (D=2)		New (D=3)	
S	PSNR	R	PSNR	R	PSNR	R	PSNR	R
0	10.91	1	11.97	0.74	11.16	0.30	11.26	0.19
1	16.80	1	17.66	0.65	17.23	0.35	17.39	0.26
2	22.84	1	23.97	0.78	23.45	0.40	23.70	0.31
3	29.17	1	30.11	0.83	29.94	0.44	30.24	0.36
4	35.81	1	36.68	1.00	36.65	0.48	36.92	0.40
5	42.82	1	43.34	1.13	43.50	0.50	43.84	0.42
6	51.32	1	51.23	1.35	51.26	0.52	51.26	0.43
7	max	1	max	1.54	max	0.52	max	0.44

We use the guess function and the error recovery procedure, to design a prototype tool. Figure 2 shows the results, with transmission bit rates and PSNR values. Four methods were implemented: the BPM, the GBN method, and our newly proposed method (with D=2 and D=3). The prototype is also an analysis tool to test different combinations of transmission bits. With our analysis, it is possible to design an intelligent transmission scheme, to achieve better PSNR values.

3. EXPERIMENTAL RESULTS AND ANALYSIS

We include different types of images in the evaluation process: 500 photos, 500 cartoon images, 450 painting, and 50 Chinese painting. The simulation result in table 3 shows that, in average of 1500 images, our method has lower transmission rates (i.e., 'R' in the table). The rates are very low, if D=3. The PSNR values are about the same as the BPM method, and even better sometimes. This also concludes that, our guess function performs much better than the GBN method.

Some images have a lot of details (such as painting) while others (i.e., cartoon) are less complicated. Thus, it is necessary to consider the variance of an image. For those images with higher color coherence, we may use a larger pixel box. Otherwise, a smaller pixel box should be used. Larger distance index (larger box) may result in a lower transmission rate. Thus, before a transmission, the variance of an image is calculated. Suppose that an image block has *i*j* pixels. The color variance, var, can be calculated as:

$$\bar{x} = \frac{\sum_{\forall i}\sum_{\forall j} x_{ij}}{i \times j} \quad , \quad \text{var} = \left(\sum_{\forall i}\sum_{\forall j}(x_{ij} - \bar{x})^2\right)/(i \times j - 1) \quad ,$$

where $\bar{x}$ is the average of pixel color. Color variance has a strong indication of the degree of details in a picture. For a color picture, the variance is taken from the average of RGB variances. We further normalize the value of variance according to the size of an image for the ease of comparison.

Table 4: Simulation Results based on 1500 Images

	BPM		GBN Method		New (D=2)		New (D=3)	
G	PSNR	R	PSNR	R	PSNR	R	PSNR	R
1	35.74	1	36.64	1.05	36.96	0.57	37.32	0.51
2	35.78	1	36.66	1.03	36.87	0.54	37.20	0.47
3	35.76	1	36.60	0.96	36.43	0.43	36.64	0.35
4	35.83	1	36.67	0.95	36.36	0.40	36.53	0.31
5	35.87	1	36.72	0.95	36.33	0.39	36.49	0.29

We divided images into 5 groups according to their variances. The lower bounds of each group are 0, 2000, 4000, 6000, and 8000, with 374, 459, 328, 221, and 118 images, respectively. We summarize the average PSNR values and average bit rates at the 4th run of each group in table 4. With the average variances increases, the average bit rates drops (for both D=2 and D=3). The average

PSNR values drops as well (but within a reasonable range, i.e., above 30dB).

Table 5: Transmission Orders

R	G	B
3	0	0
0	3	0
0	0	3
0	1	2
0	2	1
1	0	2
1	2	0
2	0	1
2	1	0
1	1	1

The last issue of our investigation is on the number of bits for each pixel that should be transmitted at each step. Since we use 24-bit RGB images for evaluation, an R, G, or B is represented by 8 bits. To make the comparison easy and to follow the strategy of BPM and GBN methods, we assume that each step will transmit 3 bits. The combinations of bit numbers to be transmitted for each color of a pixel can be found in table 5.

Note that, we make an assumption to evaluate the bit rates and PSNR values at the 4th run, which has about 50% of bits transmitted (the total will cost 8 runs). Theoretically, to find the best combination, we should use a brute force search. However, even only for 4 runs, there are 10000 combinations, since there are 10 possible candidates at each run (shown in table 5). We propose an intelligent approach to solve this problem. In the first run, we select the best combination by testing the PSNR values. Since the first run will transmit the most significant bits, it is unlikely that the second run will have a better combination that requests a change of the combination of the first run. Thus, the combination of the first run can be fixed. And, there are also 10 combinations in the second run. The third run follows the same rule. The total good combinations of the 4 runs are essentially equal to 40, instead of 10000. This is due to the special property of the 24-bit RGB images, which has the most significant bits at the beginning of each 8-bit RGB color.

Table 6: Simulation Results of Transmission Orders

	Bit Order				Hits	PSNR	Rate	Variance
A	111	111	111	111	829	30.453	0.2992	4108.2
B	012	210	111	111	85	30.751	0.3007	2317.2
C	021	201	111	111	64	30.926	0.3172	2014.2
D	120	102	111	111	49	30.687	0.2855	2725.9
E	111	012	210	111	35	29.467	0.2252	5667.8
	…	…	…	…				
X	003	030	021	012	1	27.174	0.2325	2114.4

The prototype tool run through the 40 combinations of 1500 images, and shows the combinations according to their efficiency (in terms of the best PSNR values). We briefly discuss the result here. Table 6 shows the transmission order of our method (with D=3). When D=2, a similar result occurs. That means, the size of pixel block does not affect too much of the bit orders. The first bit order (i.e., A) has the highest hit (829 times), with the average PSNR, average transmission rate, and average variance shown in table 6. This means that, for most pictures to hit a good result, each of the R, G, and B colors has one bit transmitted at each run. However, some pictures require a different transmission order to obtain a better performance. To design an intelligent progressive transmission tool, it is possible to use our scheme to calculate the bit order, which is transmitted to the receiver side before the corresponding bits are sent. Using the scheme, a highest PSNR value can be obtained at the 4th run. Another suggestion is, in our mechanism, the results of D=3 transmission is better than those of D=2. However, as the size of D becomes bigger, the frequency of error guessing may increase. An interesting future work is to investigate the relation between D, PSNR, and the bit rate.

4. CONCLUSIONS

We propose a new PIT algorithm, with adaptive size of pixel blocks. The simulation result is based on 1500 images of different categories. Our algorithm has a similar PSNR value but much lower bit rate compared to BPM and GBN methods. We also propose a mechanism to pre-calculate transmission orders, to achieve a best PSNR result on the receiver side. The proposed method is implemented and a demo Web site is available.

REFERENCES

[1] M. Accame and F. Granelli, "Hierarchical Progressive Image Coding Controlled by a Region Based Approach," *IEEE Transactions on Consumer Electronics*, Vol. 45, No. 1, February 1999, pp. 13-20.

[2] C. C. Chang, F. C. Shine, and T. S. Chen, "A New Scheme of Progressive Image Transmission Based on Bit-Plane Method," *Proceedings of the Fifth Asia-Pacific Conference on Communications and Fourth Optoelectronics and Communications Conference (APCC/OECC 99')*, Beijing, China, 1999, Vol. 2, pp. 892-895.

[3] C. C. Chang, Timothy K. Shih and I. C. Lin, "An efficient progressive image transmission method based on guessing by neighbors," in the Visual Computer journal, Springer Verlag, 2002.

[4] J. H. Kim and W. J. Song, "Pyramid-structured progressive image transmission using quantisation error delivery in transform domains," *IEE Vision, Image Signal Processing*, Vol. 143, No. 2, April 1996, pp.132-136.

[5] K. H. Tzou, "Progressive image transmission: a review and comparison of techniques," *Optical Engineering*, Vol. 26, No. 7, 1987, pp. 581-589.