ATLANTIS PRESS
Advances in Computer Science Research (ACSR), volume 76
7th International Conference on Education, Management, Information and Mechanical Engineering (EMIM 2017)

Lifting Scheme Combined with SPIHT Color Image Compression Algorithm

Xinwen Bi
Institute of Information Technology and Media, Beihua University, Jilin City, China
52600613@qq.com

Keywords: MRCT; SPIHT; Lossless Compression

Abstract. The embedded zero tree wavelet coding algorithm in wavelet transform domain is the efficient image compression coding algorithm, the wavelet transform coefficients according to their order of importance to, realize the progressive image coding, based on aggregation and S transform and MRCT transform in the lossless compression algorithm for image, through the simulation experiment, the JP2 respectively, RAR, ZIP, PNG, TGA, PCX, TIF lossless compression results improved an average of 2%, 15%, 51%, 51%, 34%, 52%, 31%.

Introduction

The SPIHT algorithm[1] is based on the embedded zerotree wavelet algorithm[2] it inherits the EZW method using the correlation between the corresponding position coefficients[4] in different frequency bands[3] after wavelet transform, Easy to control the compression ratio[5] and achieve scalable coding[6-7],Greatly optimizing the location information[8] encoding. However, the singularity of the compression coding can not be completely redundant, under this condition, a variety of algorithms combined with coding technology to achieve the redundancy of the compression technology, in this paper, we study the image compression algorithm of S-transform and SPIHT algorithm, compression results have been improved.

SPIHT Algorithms

SPIHT Algorithm Specific Symbols. $O(i,j)$ indicates the node (i,j) a collection of all child coordinates.

$$O(i,j) = \{(2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\} \text{。} \tag{1}$$

$D(i,j)$ represents the set of all the descendant coordinates of node (i,j).

H denotes the set of transform coefficients of the largest scale of wavelet transform, both LL_J, HL_J, LH_J and HH_J.

$$L(i,j) \text{ means } L(i,j) = D(i,j) - O(i,j) \text{。} \tag{2}$$

Three lists:

(LIS), important pixel list (LIP), important pixel list (LSP), in LSP,LIP, (i,j) represents a single pixel, in LIS, (i,j) represents set $L(i,j)$ or $D(i,j)$. In order to distinguish between the two types of collections, if it is $D(i,j)$ said LIS table value for the A type, if it is $L(i,j)$ said LIS table value for the B type.

SPIHT Specific Implementation Process. Ⅰ Initialization:

Output $n = \lfloor \log_2(\max_{(i,j)}\{|C_{i,j}|\}) \rfloor$, set the LSP is empty, the coordinate $(i,j) \in H$ into the LIP, and H in the descendants (high frequency part: HLJ,LHJ,HHJ) into the LIS, as the A value.

Ⅱ Sorting process:

(1) for each $(i,j)\in$ LIP

1) Output $S_n(i,j)$;

2) If $S_n(i,j)=1$, move $S_n(i,j)=1$ into the LSP and output the sign of $c(i,j)$;

(2) for each $(i,j)\in$ LIS

1) If the value of A, then

① output $S_n(D(i,j))$;

② if $S_n(D(i,j))=1$, then for each $(k,l)\in O(i,j)$, as:

• Output $S_n(k,l)$;

• If $S_n(k,l)=1$, send (k, l) to the LSP and output its symbol;

• If $S_n(k,l)=0$, send (k, l) to the end of the LIP;

③ If $L(i,j) \neq \phi$, move (k, l) to the end of the LIS as the B value; otherwise, remove (i, j) from the LIS.

2) If the value of B, then

① output $S_n(L(i,j))$;

② if $S_n(L(i,j))=1$, then

• for each $(k,l)\in O(i,j)$ added to the end of the LIS as an A value;

• Delete (i, j) from LIS.

(3) refinement process: for each $(i,j)\in LSP$ (not including the last splitting process), output the nth most significant bit of $|c_{i,j}|$;

(4) quantization step size refresh: n = n-1; return (2).

S Transform

The simplest, most generally reversible integer to integer mapping is the S transform, the S transform is well known as a reversible integer to integer transform, and its basic component is the Haar wavelet transform.

It can be expressed as:

$$\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} \left\lfloor \frac{1}{2}(x_0 + x_1) \right\rfloor \\ x_0 - x_1 \end{bmatrix} \tag{3}$$

When $\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} t \\ t - y_1 \end{bmatrix}$ is: t $\cong y_0 + \left\lfloor \frac{1}{2}(y_1 + 1) \right\rfloor$

Of course, the above formula is the most basic form of S transform, we often use more complex three-dimensional form, S-transform can be extended to the generalized S-transform generalized S-transform (GST).

Here we set x, y are:

$$x \cong \lfloor X_0 \quad X_1 \cdots \quad X_{N-1} \rfloor^T \tag{4}$$

$$y \cong \left[y_0 \quad y_1 \cdots \quad y_{N-1} \right]^T \tag{5}$$

There are y = Cx + Q ((B-I) Cx) where B and C are what? It can be expressed as:

$$B = \begin{pmatrix} 1 & b_1 & b_2 \ldots b_{N-1} \\ 0 & 1 & 0 \ldots 0 \\ 0 & 0 & 1 \ldots 0 \\ \ldots & \ldots & \ldots \\ 0 & 0 & 0 \ldots 1 \end{pmatrix} \tag{6}$$

$$C \cong \begin{pmatrix} C_{0,0} & C_{0,0} & C_{0,0} & \ldots & C_{0,N-1} \\ C_{1,0} & C_{1,1} & C_{1,2} & \ldots & C_{1,N-1} \\ C_{2,0} & C_{2,1} & C_{2,2} & \ldots & C_{2,N-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ C_{N-1,0} & C_{N-1,1} & C_{N-1,2} & \ldots & C_{N-1,N-1} \end{pmatrix} \tag{7}$$

Here, if A ≅ BC then x = C-1 (y-Q ((B-I) y)). It should be noted that C-1 is also an integer matrix. Then we look at the matrix B, or it can be replaced by any two unknown variables u, v: V = u + Q ((B-I) u) or u = v-Q ((B-I) v). Equivalent to us into u = Cx and v = y.

A = BC, then A can be defined as

$$A \cong \begin{pmatrix} a_{0,0} & a_{0,0} & a_{0,0} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & a_{1,2} & \cdots & a_{1,N-1} \\ a_{2,0} & a_{2,1} & a_{2,2} & \cdots & a_{2,N-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{N-1,0} & a_{N-1,1} & a_{N-1,2} & \cdots & a_{N-1,N-1} \end{pmatrix} \tag{8}$$

Break down the steps:

Step1. The integer term of A for N-1 rows is:

Ai,j∈Z (i = 1, 2, ..., N-1, j = 0,1, ..., N-1)

Step 2. The modulus of A is 1.

Step 3. {Det minor (A, 0, i)}N-1is a quality.

MRCT Transformation. When compressing a color image, the color is expressed as multiple components often in order to improve the coding efficiency. This transformation can be expressed by brightness. In this way, people can easily extract the gray scale of the image. For RGB color images, the brightness is the color of the element, it can usually be expressed as y = 0.299r +0.587g +0.114b We use this formula to express the brightness and gray value when you can use this formula: $y = \frac{1}{4}r + \frac{1}{2}g + \frac{1}{4}b$

In this way the calculation is rough, there will be a lot of data loss in practical applications, if the structure of such changes can improve the approximate brightness, we call this method for the MRCT. Extended to GST as follows:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 + \left\lfloor \frac{1}{128}(90\,t_0 + 15\,t_1) \right\rfloor \\ t_0 \\ t_1 \end{bmatrix}$$ then t0=x2-x1,t1=x0-x1；

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \\ s_1 + y_2 \end{bmatrix}$$ then s=y0- $\left\lfloor \frac{1}{128}(90\,y_1 + 15\,y_2) \right\rfloor$,s0=s2-y1

Experimental Results and Conclusions

In order to illustrate the effectiveness of the algorithm, this paper compares the lossless image compression algorithms of JP2, RAR, ZIP, PNG, TGA, PCX and TIF in 12 color international standard test images, and on average, no compression ratio Than the above algorithm were increased by 2%, 15%, 51%, 51%, 34%, 52%, 31%, as shown in Table 1.

Table 1 12 color international standard test image compression experiment comparison results

Compression scheme / InternaTional Standard Test Images	JP2	RAR	PCX	TGA	ZIP	TIF	PNG
Kodak1	511631	598789	1161410	1157891	796111	1065614	783060
Kodak2	451678	508771	1142233	1150300	665774	865916	621478
Kodak3	399015	458212	1096862	1085783	569796	890712	549676
Kodak4	461292	512279	1172752	1153733	747996	108740	640877
Kodak5	533178	684773	1185005	1162343	915984	1238244	810363
Kodak6	472736	532397	1188493	1148689	718077	1001630	673733
Kodak7	419224	488048	1113169	1109121	650774	976216	573980
Kodak8	548967	804423	1309333	1163564	951692	1255784	791697
Kodak9	446250	488367	1190110	1152950	619045	838012	587850
Kodak10	454440	495213	1171219	1154093	686399	958320	598508
Kodak11	458044	553611	1139853	1129959	699897	930316	643047
Kodak12	426781	482724	1247133	1130272	598262	881932	575313
SPIHT+MRCT	2%	15%	51%	51%	34%	52%	31%

Acknowledgements

This work was financially supported by National Ministry of Education Chunhui Plan(104900071, 1049000150), Exploring Practice of C Language Program Design" Case Teaching for Professional Application (XJQN2016035).

References

[1] Tasai M J, Villasenor T D, Chen F. Stack-run image coding[J].IEEE Transactions on Circuits System Video Technology, 1996,6(5): 519-521.

[2] Mallat S. Multifrequency channel decompositions of images and wavelet models[J].IEEE

Transactions on ASSP,1989,37(12): 2091-2109.
[3] Moffat A. Linear time adaptive coding [J].IEEE Transactions on Info Theory,1990,36(2): 401-406.
[4] Shapiro J M. Embedded image coding using zerotree of wavelet coefficients [J].IEEE Transactions on Signal Processing, 1993, 41 (12): 3445-3462.
[5] Ordentlich E, Weinberger M, Seroussi G. On modeling and ordering for embedded image coding[C]//Proc IEEE Int Symp Information Theory,2000: 297-299.
[6] Taubman D. High performance scalable image compression with EBCOT [J].IEEE Transactions on Image Processing, 2000, 9 (6): 1158-1170.
[7] Said A, Pearlman W A.A new fast and efficient image codec based on set partitioning in hierarchical trees[J].IEEE Transactions on Circuit and Systems for Video Technology, 1996, 6(3): 243-249.
[8] Witten I H, Neal R M,Clew J G. Arithmetic coding for data compression[J]. Commun ACM, 1987, 30(6): 520-540.