

**Некоторые вейвлет-преобразования,
находящие применение для сжатия изображений**

Таблица 1

Краткая характеристика преобразований

Условное обозначение	Название преобразования	Описание	Длина фильтров	Число исчезающих моментов
Haar	Хаара	ортогональное, симметричное, интерполирующее	2 / 2	(1,1)
TS	“два-шесть”	биортогональное, симметричное	2 / 6	(3,1)
CDF22	Козна-Добеши-Фово	биортогональное, симметричное, интерполирующее	5 / 3	(2,2)
CDF24	Козна-Добеши-Фово	биортогональное, симметричное, интерполирующее	9 / 3	(2,4)
CDF97	Козна-Добеши-Фово	биортогональное, симметричное	9 / 7	(4,4)
V610	Вилласенора	биортогональное, симметричное	6 / 10	(4,4)
MIT97	MIT9/7	биортогональное, симметричное, интерполирующее	9 / 7	(4,2)
BCW3	Койфмана, биортогональное	биортогональное, симметричное, интерполирующее	13 / 7	(4,4)

Коэффициенты вейвлет-фильтров
(для симметричных фильтров показана только половина коэффициентов)

Haar	$h_0 = 1/\sqrt{2}, 1/\sqrt{2};$ $h_1 = -1/\sqrt{2}, 1/\sqrt{2}.$
TS	$h_0 = 1/\sqrt{2};$ $h_1 = -1/8\sqrt{2}, -1/8\sqrt{2}, 8/8\sqrt{2}.$
CDF22	$h_0 = -1/4\sqrt{2}, 2/4\sqrt{2}, 6/4\sqrt{2};$ $h_1 = 1/2\sqrt{2}, -2/\sqrt{2}.$
CDF24	$h_0 = 3\sqrt{2}/128, -6\sqrt{2}/128, -16\sqrt{2}/128, 38\sqrt{2}/128, 90\sqrt{2}/128;$ $h_1 = \sqrt{2}/4, -2\sqrt{2}/4.$
CDF97	$h_0 = 0.03783, -0.02385, -0.1106, 0.3774, 0.8527;$ $h_1 = 0.06454, -0.04069, -0.4181, 0.7885.$
V610	$h_0 = -0.1291, 0.04770, 0.7885;$ $h_1 = -0.01891, 0.006989, 0.06724, 0.1334, -0.6151.$
MIT97	$h_0 = \sqrt{2}/64, -8\sqrt{2}/64, 16\sqrt{2}/64, 46\sqrt{2}/64;$ $h_1 = -\sqrt{2}/32, 9\sqrt{2}/32, -16\sqrt{2}/32.$
BCW3	$h_0 = -1/256\sqrt{2}, 18/256\sqrt{2}, -1/16\sqrt{2}, -63/256\sqrt{2}, 9/16\sqrt{2}, 348/256\sqrt{2};$ $h_1 = -1/16\sqrt{2}, 9/16\sqrt{2}, -16/16\sqrt{2}.$

Текст функций, выполняющих прямое и обратное вейвлет-преобразование изображения

// Функции для вычисления прямого и обратного вейвлет-преобразования
// изображения с использованием биортогональных 9/7 фильтров. Текст
// программы – модифицированная версия текста G.Davis (см. список
// Интернет-источников). Выполнение вейвлет-преобразования изображения
// размером 512x512 на компьютере P-166 занимает примерно 0.4 с.

```
#include "wavelet.h"
```

```
#define al0      0.852699
#define al1      0.377403
#define al2      -0.110624
#define al3      -0.023849
#define al4      0.037829
```

```
#define ah0      0.788485
#define ah1      -0.418092
#define ah2      -0.040690
#define ah3      0.064539
```

```
#define sl0      -0.852699
#define sl1      -0.418092
#define sl2      0.110624
#define sl3      0.064539
#define sl4      -0.037829
```

```
#define sh0      -0.788485
#define sh1      0.377403
#define sh2      0.040690
#define sh3      -0.023849
```

```
#define AnaLowPass(v)  al0*v[0]+al1*(v[1]+v[-1])+al2*(v[2]+v[-\
2])+al3*(v[3]+v[-3])+al4*(v[4]+v[-4])
```

```

#define AnaHighPass(v)  ah0*v[0]+ah1*(v[1]+v[-1])+ah2*(v[2]+v[-\
2])+ah3*(v[3]+v[-3])
#define SynLowPass(v)   sl0*v[0]+sl1*(v[1]+v[-1])+sl2*(v[2]+v[-\
2])+sl3*(v[3]+v[-3])+sl4*(v[4]+v[-4])
#define SynHighPass(v)  sh0*v[0]+sh1*(v[1]+v[-1])+sh2*(v[2]+v[-\
2])+sh3*(v[3]+v[-3])

```

```
WaveletTransform::WaveletTransform ()
```

```

{
    // Это необходимо для продолжения изображения при симметричном
    отражении
    npad = 4;
}

```

```
WaveletTransform::~~WaveletTransform ()
```

```

{
}

```

```
void WaveletTransform::Reflection(float * h, float * t)
```

```

{
    for (int i=1; i <= npad; i++)
    {
        h[-i] = h[i];
        t[i] = t[-i];
    }
}

```

```
// -----
```

```
// FWT (int ** input, float ** output, int hSize, int vSize,
//      int trSteps, int symExtension)
//
```

```
// Функция прямого вейвлет-преобразования. Выполняется разделимое
// преобразование, как описано в главе 9.
```

```
// Входные параметры:
```

```

// float *input: входной буфер данных,
// float *output: выходной буфер данных,
// int hSize: горизонтальный размер изображения,
// int vSize: вертикальный размер изображения,
// int trSteps: число шагов декомпозиции,
// int symExtension: вид продолжения на границе, по умолчанию -1.
// -----

```

```
BOOL WaveletTransform::FWT (float *coeff, int trSteps, int hSize, int vSize)
```

```

{

```

```

int i, j, lowSize, highSize;
int hTransSize = hSize;
int vTransSize = vSize;

float ** output = new float * [vTransSize];
output[0] = coeff;
for ( i=1; i<vSize; i++ )
    output[i] = output[i-1] + hTransSize;
int *intIn, *intOut;
float *floatIn, *workingVector, *bufferPtr, *bufferEnd;
floatIn = new float [2*npad + Max(hTransSize, vTransSize)];
workingVector = floatIn+npad;
    while (trSteps--)
{
    // Выполняем свертку каждой строки с НЧ фильтром
    for ( i=0; i<vTransSize; i++ )
    {
        lowSize = (hTransSize+1) >> 1;
        highSize = hTransSize - lowSize;
        // Копируем строку i в массив данных
        memcpy( workingVector, output[i], hTransSize*sizeof(float) );
        Reflection( workingVector, workingVector+hTransSize-1);
        for ( bufferPtr = workingVector, j = 0; j < highSize; j++)
        {
            output[i][j] = AnaLowPass(bufferPtr);
            bufferPtr++;
            output[i][j+lowSize] = AnaHighPass(bufferPtr); bufferPtr++;
        }
        if ( lowSize > highSize )
            output[i][highSize] = AnaLowPass(bufferPtr);
    }
    // Выполнение свертки НЧ фильтра со столбцами
    for ( j=0; j<hTransSize; j++ )
    {
        lowSize = (vTransSize+1) >> 1;
        highSize = vTransSize - lowSize;
        // Копируем столбец j в массив данных
        for ( i=0; i<vTransSize; i++ )
            workingVector[i] = output [i][j];
        Reflection( workingVector, workingVector+vTransSize-1);

        for ( bufferPtr = workingVector, i = 0; i < highSize; i++)
        {

```

```

        output[i][j] = AnaLowPass(bufferPtr);
bufferPtr++;
        output[i+lowSize][j] = AnaHighPass(bufferPtr); bufferPtr++;
    }
    if ( lowSize > highSize )
        output[highSize][j] = AnaLowPass(bufferPtr);
    }

    hTransSize = (hTransSize + 1) / 2;
    vTransSize = (vTransSize + 1) / 2;
}
delete output;
delete floatIn;
return TRUE;
}

```

```

// -----
// IWT (float *input, float *output, int hSize, int vSize,
//      int trSteps, int symExtension)
//
// Обратное вейвлет-преобразование
// Входные параметры:
// float *input: входной буфер данных,
// float *output: выходной буфер данных,
// int hSize: размер изображения по горизонтали,
// int vSize: размер по вертикали,
// int maxOrig: максимальное значение пиксела в исх. изображении,
// int minOrig: минимальное значение пиксела в исх. изображении,
// int trSteps: число шагов декомпозиции,
// int symExtension: продолжение на границе, по умолчанию -1.
// -----

```

```

BOOL WaveletTransform::IWT( float *coeff, int trSteps, int hSize,
                           int vSize, int maxOrig, int minOrig)
{
    int i, j;
    float ** input = new float * [vSize];
    input[0] = coeff;
    for ( i=1; i<vSize; i++ )
        input[i] = input[i-1] + hSize;
    int * hTransSize = new int [trSteps];
    int * vTransSize = new int [trSteps];
    int hCrtSize, vCrtSize, normalSize, lowSize, highSize;

```

```

hTransSize[0] = hSize;
vTransSize[0] = vSize;
for (i = 1; i < trSteps; i++)
{
    hTransSize[i] = (hTransSize[i-1] + 1) / 2;
    vTransSize[i] = (vTransSize[i-1] + 1) / 2;
}

int *intIn, *intOut;
float *floatIn, *workingVector, *bufferPtr, *bufferEnd;
floatIn = new float [2*npad + Max(hSize, vSize)];
workingVector = floatIn+npad;

while (trSteps--)
{
    hCrtSize = hTransSize[trSteps];
    vCrtSize = vTransSize[trSteps];
    // Выполняем реконструкцию для столбцов
    for (j = 0; j < hCrtSize; j++)
    {
        lowSize      = (vCrtSize + 1) >> 1;
        highSize = vCrtSize - lowSize;
        normalSize   = highSize*2;
        bufferPtr    = workingVector;
        for ( i = 0; i < highSize; i ++ )
        {
            *bufferPtr = input[i][j];
            bufferPtr++;
            *bufferPtr = input[i+lowSize][j];
            bufferPtr++;
        }
        if ( lowSize > highSize )
            workingVector[normalSize] = input[highSize][j];
        Reflection(workingVector, workingVector + vCrtSize - 1);
        bufferPtr = workingVector;
        for ( i = 0; i < normalSize; )
        {
            input[i++][j] = SynHighPass(bufferPtr);
            bufferPtr++;
            input[i++][j] = SynLowPass(bufferPtr);
            bufferPtr++;
        }
    }
}

```

```

        if ( lowSize > highSize )
            input[normalSize][j] = SynHighPass(bufferPtr);
    }
    // Выполняем реконструкцию для строк
    for ( i = 0; i < vCrtSize; i ++ )
    {
        lowSize          = (hCrtSize + 1) >> 1;
        highSize = hCrtSize - lowSize;
        normalSize       = highSize*2;
        bufferPtr        = workingVector;
        for ( j = 0; j < highSize; j ++ )
        {
            *bufferPtr = input[i][j];
            bufferPtr++;
            *bufferPtr = input[i][j+lowSize];
            bufferPtr++;
        }
        if ( lowSize > highSize )
            workingVector[normalSize] = input[i][highSize];
        Reflection(workingVector, workingVector + hCrtSize - 1);
        bufferPtr        = workingVector;
        for ( j = 0; j < normalSize; )
        {
            input[i][j++] = SynHighPass(bufferPtr);
            bufferPtr++;
            input[i][j++] = SynLowPass(bufferPtr);
            bufferPtr++;
        }
        if ( lowSize > highSize )
            input[i][normalSize] = SynHighPass(bufferPtr);
    }
}

```

```

int imgSize = vSize*hSize;
bufferPtr=coeff, bufferEnd = coeff+imgSize;

```

```

while ( bufferPtr < bufferEnd )
{
    *bufferPtr = Round(*bufferPtr);
    if (*bufferPtr > maxOrig)
        *bufferPtr = maxOrig;
    else
        if (*bufferPtr < minOrig)

```



```

        *bufferPtr = minOrig;
        bufferPtr ++;
    }
    delete floatIn;

delete hTransSize;
delete vTransSize;
delete input;
return TRUE;
}

void WaveletTransform::SP_TransformStep(int *input, int *output, int size)
{
    int i, k, d1, d2;

    int lowSize = (size+1)/2;
    int highSize = size - lowSize;
    int *l = output;
    int *h = output + lowSize;
    int *in = input;

    for (i = k = 0; i < highSize; i++, k += 2)
    {
        l[i] = (in[k] + in[k+1]) >> 1;
        h[i] = in[k] - in[k+1];
    }
    d2 = l[0] - l[1];
    h[0] -= d2 >> 2;
    for (i = 1; i < highSize-1; i++)
    {
        d1 = d2;
        d2 = l[i] - l[i+1];
        h[i] -= (((d1 + d2 - h[i+1]) << 1) + d2 + 3) >> 3;
    }
    h[i] -= d2 >> 2;

    if ( lowSize > highSize )
        l[highSize] = in[2*highSize];
}

```

```

void WaveletTransform::SP_InvertStep(int *input, int *output, int size)
{
    int i, k, d1, d2, t;
    int lowSize = (size+1) >> 1;
    int highSize = size - lowSize;
    int *l = input;
    int *h = input + lowSize;
    int *out = output;

    t = (h[highSize-1] += (d1 = l[highSize-2] - l[highSize-1]) >> 2);
    for (i = highSize - 2; i > 0; i--)
    {
        d2 = d1;
        d1 = l[i-1] - l[i];
        t = (h[i] += (((d1 + d2 - t) << 1) + d2 + 3) >> 3);
    }
    h[0] += d1 >> 2;

    for (i = k = 0; i < highSize; i++, k += 2)
    {
        out[k] = l[i] + ((h[i] + 1) >> 1);
        out[k+1] = out[k] - h[i];
    }

    if ( lowSize > highSize )
        output[2*highSize] = l[highSize];
}

//
// Файл wavelet.h
//
#ifndef WAVELET_H
#define WAVELET_H

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
#include <string.h>

```

```

class WaveletTransform
{
public:

    WaveletTransform ();
    ~WaveletTransform ();

    BOOL FWT (float *coeff, int trSteps, int hSize, int vSize);
    BOOL IWT (float *coeff, int trSteps, int hSize, int vSize,
              int maxOrig, int minOrig);

    int npad;

protected:

    void Reflection(float * h, float * t);
    void SP_TransformStep(int *input, int *output, int size);
    void SP_InvertStep(int *input, int *output, int size);
};
#endif

```

БИБЛИОГРАФИЯ

Глава 1

1. Вайдьянатхан П.П. Цифровые фильтры, блоки фильтров и полифазные цепи с многочастотной дискретизацией: Методический обзор //ТИИЭР, 1990, №3. С.77-120.
2. Харатишвили Н.Н. Пирамидальное кодирование. М.: Мысль, 1997.160с.
3. Burt P., Adelson H.. The Laplacian pyramid as a compact image code //IEEE Transactions on Communications, 1983, №4. P.23-38.
4. Malvar H.S. Lapped transforms for efficient transform /subband coding // IEEE Trans. Acoust. Speech Signal Process., 1990, №2. P.969-978.
5. Simoncelli E., Adelson E.. Subband transforms / Woods J., editor. Subband Image Coding. Kluwer Academic Publishers, Norwell, MA, 1990.
6. Vaidyanathan P.P. Theory of optimal orthonormal subband coders //IEEE Trans. on Signal Processing, 1998, №4. P.1528-1543.

Глава 2

1. Астафьева Н.М. Вейвлет-анализ:основы теории и некоторые приложения //Успехи физических наук, 1996, №11. С.1145-1170.
2. Желудев В.А. О вейвлетах на базе периодических сплайнов //Докл. РАН, 1994, №1. С. 9- 13.
3. Петухов А. П. Периодические дискретные всплески //Алгебра и анализ, 1996, №3. С. 151–183.
4. Кравченко В. Ф., Рвачев В. А., Пустовойт В. И. Ортонормированные системы типа wavelet на основе атомарных функций //Докл. РАН, 1996, №1.С. 16–18.
5. Daubechies I. Orthonormal Bases of Compactly Supported Wavelets //Comm. Pure Appl. Math., 1988, vol.41.P. 906-966.
6. Daubechies I. The wavelet transform, time-frequency localization and signal analysis // IEEE Trans. Inform. Theory, 1990, №5. P.961-1005.
7. Jawerth B., Sweldens W. An overview of wavelet based multiresolution analysis //SIAM Rev., 1994, №3. P.377-412.
8. Mallat S. A theory for multiresolution signal decomposition:the wavelet representation //IEEE Trans. Pattern Analysis and Machine Intelligence, 1989, №7. P.674-693.

9. Rioul O., Vetterli M. Wavelets and signal processing //IEEE Signal Processing Magazine, 1991, №10. P. 14-38.

10. Schumaker L., Webb G., editors. Recent Advances in Wavelet Analysis. Academic Press, New York, 1993.480p.

11. Strang G. Wavelets and dilation equations: A brief introduction //SIAM Rev., 1989, №4. P.614-627.

12. Vetterli M., Herley C. Wavelets and Filter Banks: Theory and Design // IEEE Transactions on Signal Processing, 1992, v.40. P. 2207-2232.

Глава 3

1. Balasingham I. On Optimal Perfect Reconstruction Filter Banks for Image Compression. PhD Thesis. NUST, Trondheim, Norway, 1998r.

2. Johnston J. A filter family designed for use in quadrature mirror filter banks //Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 1980. P.291-294.

3. Vetterli M. Splitting a signal into subband channels allowing perfect reconstruction //Proc. IASTED Conf. Appl. Signal Proc., 1985, Paris.

Глава 4

1. Strang G., Nguyen T. Wavelets and Filter Banks. Wellesley-Cambridge Press, Wellesley, MA, 1996.

2. Majani E. Biorthogonal wavelets for image compression //Proc. SPIE Visual Communications and Image Processing, 1994.

3. Smith M., Barnwell T. Exact reconstruction techniques for tree - structured subband coders // IEEE Transactions On Acoustic, Speech, Signal Processing, 1986, №6.P. 434-441.

4. Soman A., Vaidyanathan P. On orthonormal wavelets and paraunitary filter banks. // IEEE Trans. Signal Processing, 1993,№3. P. 1170-1183.

5. Vetterli M. Filter banks allowing perfect reconstruction //IEEE Transactions On Signal Processing. 1986, №2. P. 219-244.

6. Vetterli M. Wavelet and filter banks for discrete-time signal processing /Ruskai M. , ed. Wavelets and Their Applications, 1992, Boston. P.17-52.

Глава 5

1. Herley C., Kovacevic J., Ramchandran K., Vetterli M. Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms //IEEE Transactions on Signal Processing, 1993, №12. P. 3341-3359.

2. Herley C., Xiong Z., Ramchandran K., Orchard M. Joint space-frequency segmentation using balanced wavelet packet trees for least-cost image representation //IEEE Transactions on Image Processing, 1997, №9.

3. Ramchandran K., Vetterli M. Best wavelet packet bases in a rate-distortion sense //IEEE Transactions on Image Processing, 1992, №2.P.160-175.
4. Saito N. Local Feature Extraction and Its Applications Using a Library of Bases. Ph.D Dissertation, Yale Univ., 1994.
5. Xiong Z. Image Coding Using Wavelet Transform. PhD Thesis, Illinois Univ., 1996.
6. Wickerhauser M. Acoustic signal compression with wavelet packets. In Chui C. ed. Wavelets :A Tutorial in Theory and Applications, 1992, Academic Press, Boston. P. 679-700.

Глава 6

1. Daubechies I., Sweldens W. Factoring wavelet transforms into lifting steps. Technical report, Bell Laboratories, Lucent Technologies, 1996.
2. Sweldens W., Schreder P. Building your own wavelets at home. Technical Report, Industrial Mathematics Initiative, Mathematics Department, University of South Carolina, 1995.
3. Sweldens W. The lifting scheme: A construction of second generation wavelets. Technical Report, Industrial Mathematics Initiative, Mathematics Department, University of South Carolina, 1995.
4. Sweldens W. The lifting scheme: A custom-design construction of biorthogonal wavelets //Applied and Computational Harmonic Analysis, 1996, №2. P.186-200.

Глава 7

1. Adams M., Antoniou A. Design of reversible subband transforms using lifting //Proc. of IEEE Pacific Rim Conference, 1997, v.1. P. 489-492.
2. Adams M. Reversible Wavelet Transforms and Their Application to Embedded Image Compression. Master Thesis. University of Victoria, 1998.
3. Balasingham I., Lervik J., Ramstad T. Lossless subband image compression using integer coefficient filter banks and classwise arithmetic coding //Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Seattle, 1998.
4. Calderbank A., Daubechies I., Sweldens W., Yeo B. Wavelet transforms that map integers to integers. Tech. report, Department of Mathematics, Princeton University, 1996.
5. Jung H., Prost R. Rounding transform based approach for lossless subband coding //Proc. IEEE Int. Conf. on Image Processing, 1997, v.2. P.274-277.
6. Zandi A., Allen J., Schwartz E., Boliek M. CREW: Compression with reversible wavelets //Proc. of IEEE Data Compression Conf., 1995.P.212-221.

Глава 8

1. Lebrun J., Vetterli M. Balanced multiwavelets: theory and design. IEEE Trans. Signal Proc., №4, 1998.
2. Geronimo J., Hardin D., Massopust P.. Fractal functions and wavelet expansions based on several scaling functions //Journal Approx.Theory, 1993, №3.
3. Strang G., Strela V. Short wavelets and matrix dilation equations //IEEE Trans.Signal Proc., 1995, v.3. P.108-115.
4. Strela V. Multiwavelets: Theory and applications. PhD Thesis. Massachusetts Institute Technology, 1996.
5. Park H. A computational theory of Laurent polynomial rings and multidimensional FIR systems. PhD Thesis. UC Berkeley, 1995.
6. Strela V., Heller P., Strang G., Topiwala G., Heil C. The application of multiwavelet filter banks to image processing //IEEE Trans. Sig.Proc., 1996.
7. Plonka G., Strela V. From wavelets to multiwavelets / Mathematical Methods for Curves and Surfaces M.Dahlen, T.Lyche, L.Schumaker (eds.), Vanderbilt University Press, Nashville, 1998. P. 375-399.

Глава 9

1. Галлагер Р. Теория информации и надежная связь. М.:Мир, 1974.
2. Макхоул Д., Рукос С., Гиш Г. Векторное квантование при кодировании речи //ТИИЭР, т.73, №11, 1985. С.19-61.
3. Шеннон К. Математическая теория связи / Шеннон К. Работы по теории информации и кибернетике.М.:1963. С.243-333.
4. Mallat S. Analysis of low bit rate image transform coding //IEEE Trans. Signal Proc., 1998, №4.P.1127-1142.

Глава 10

1. Antonini M., Barlaud M., Mathieu P. Image Coding Using Wavelet Transform //IEEE Trans. Image Proc., 1992, №2. P.205-220.
2. Criminal Justice Information Services. WSQ Grayscale Fingerprint Image Compression Specification. Federal Bureau of Investigation, Feb.1993.
3. Davis G., Chawla S. Image coding using optimized significance tree quantization //Proc. Data Compression Conference, 1997. P. 387-396.
4. Lewis A., Knowles G. Image compression using the 2-d wavelet transform //IEEE Transactions on Image Processing, 1992, №2. P.244-250.
5. LoPresto S., Ramchandran K., Orchard M.. Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework //Proc. Data Compression Conference, Snowbird, 1997. P. 221-230.
6. Mallat S., Falzon F. Understanding image transform codes //Proc. SPIE Aerospace Conf., Orlando, 1997.

7. Said A., Pearlman W. A new, fast, and efficient image codec based on set partitioning in hierarchical trees //IEEE Trans. on Circuits and Systems for Video Technology, 1996, №3. P.243-250.

8. Shapiro J. Embedded image coding using zerotrees of wavelet coefficients //IEEE Trans. on Signal Processing, 1993, №12. P.3445-3462.

9. Shoham Y., Gersho A. Efficient bit allocation for an arbitrary set of quantizers //IEEE Trans. Acoustics, Speech, and Signal Processing, 1988, №9. P.1445-1453.

10. Villasenor J., Belzer B., Liao J. Wavelet filter evaluation for image compression //IEEE Trans. on image processing, 1995, №8. P.1053-1060.

11. Yoo Y., Ortega A., Yu B.. Progressive classification and adaptive quantization of image subbands. Preprint, 1997.

Глава 11

Техническая информация фирмы Analog Devices, которую можно получить на сайте: <http://www.analog.com>

Интернет-адреса

Адрес сервера фирмы Mathsoft, на котором имеется много ссылок на статьи по вейвлетам, сгруппированных по категориям:

<http://www.mathsoft.com/wavelet.html>

Очень хороший список групп, исследующих вейвлеты, находится на сервере корейского университета:

<http://dali.korea.ac.kr/research/Wavelet/wavelets.html>

В.Свелденс поддерживает в Сети общедоступный вейвлет-дайджест – ежемесячный реферативный журнал. Адрес:

<http://www.wavelet.org>

Адрес сервера В.Свелденса, на котором размещены его статьи и диссертация «Вейвлеты и численный анализ»:

<http://cm.bell-labs.com/who/wim/papers/papers.html>

Статьи посвящены различным математическим аспектам теории вейвлетов.

Адрес Е.Ковачевич, руководителя подразделения лаборатории Bell, занимающегося вейвлетами:

<http://cm.bell-labs.com/who/jelena/papers/papers.html>

Здесь вы найдете много интересных статей, посвященных локальным базисам, произвольному разбиению частотно-временной плоскости.

На сервере Дортмутского колледжа имеется ряд работ Д.Дэвиса, в том числе и его диссертация, посвященная адаптивной нелинейной аппроксимации (алгоритм Matching Pursuit). Здесь же находится общедоступный C++ код вейвлет-кодера изображений.

<http://www.cs.dartmouth.edu/~Davis>

Большое количество работ ученых лаборатории вычислительной математики Райсовского университета размещено на сервере

<http://cml.rice.edu>

Особый интерес этот адрес должен вызвать у людей, занимающихся частотно-временным анализом сигналов, обнаружением локальных особенностей и т.д. Советуем обратить внимание на многочисленные работы Р.Баранюка, которые он любезно сделал доступными в он-лайн-овом режиме.

Читателям, интересующимся вопросом распараллеливания алгоритма вейвлет-преобразования, рекомендуем ознакомиться с работами А. Ухла:

<http://www.mat.sbg.ac.at/~uhl/pubs.html>

Те, кого в первую очередь интересуют вопросы применения многоскоростной цифровой фильтрации для кодирования изображений, не должны пройти мимо работ норвежских ученых:

анализ артефактов, возникающих при кодировании, выполнен в диссертации А.Аасе:

<http://www.hsr.no/~aase/publicatons.html>

оптимизация блоков фильтров, в т.ч. и вейвлет-фильтров, проведена И.Баласингамом:

<http://www.tele.ntnu.no/~ilangko>

обработка ЭКГ:

<http://www.hsr.no/SIGPROC/www/programs.html>

(здесь же имеется программный код).

Вопросы целочисленного вейвлет-преобразования исследовал в своей диссертации М.Адамс:

<http://www.ece.ubc.ca/~mdadams/papers>

Наверное, наиболее популярным алгоритмом вейвлет-кодирования изображения является SPIHT. Полное описание алгоритма, а также EXE-модули можете найти по адресу:

<http://ipl.rpi.edu/RESEARCH/SPIHT>

Много интересных статей, а также С-код алгоритма сжатия EPWIC (не рассмотренного на страницах нашей книги) предлагает Е.Симонселли по адресу:

<http://www.cns.nyu.edu/~eero/publications.html>

С работами Д.Вилласенора можете ознакомиться по адресу

<http://www.icsl.ucla.edu>

Особо рекомендуем прочесть статью “Wavelet filter evaluation for image compression”.

Труды М.Веттерли можно найти по адресу

<http://www.eecs.berkeley.edu>.

К сожалению, они недоступны в он-лайновом режиме.

Вопросам кодирования изображений, видео и одновременного обеспечения их передачи по каналам связи посвящено много трудов. Проблемы передачи сжатой видеоинформации по сетям исследуют по адресу

<http://sipi.usc.edu/~ORTEGA/>

Здесь вы найдете диссертацию и много работ А.Ортега, а также работы его учеников.

С методами вейвлет-компрессии изображений и передачи их по радиоканалу можно ознакомиться, например, в диссертации Х.Джавархани по адресу:

http://www.isr.umd.edu/TechReports/ISR/1997/Ph_D_97_2

В настоящей книге не были рассмотрены вопросы векторного квантования коэффициентов вейвлет-преобразования. Однако это вовсе не означает, что данные методы являются неэффективными. Многие исследователи занимаются этими вопросами. Для первоначального знакомства с проблемой мы бы советовали прочесть обзорную статью

Cosman P., Gray R., and Vetterli M. Vector quantization of image subbands: a survey //IEEE Transactions on Image Processing, 1996, №2. P.202-225. (166 ссылок на источники (!)).

Интересные тексты кодов вейвлет-преобразования можно получить на сервере черноголовского отделения РАН:

<http://netserv2.chg.ru/ftp/prog/wavelets/programming>

Вообще говоря, список Интернет-адресов можно было бы продолжать до бесконечности. Но все-таки поиск информации в Сети – личное дело

каждого. Для осуществления поиска мы, например, использовали такие серверы, как Yahoo, Altavista, Hotbot, Infoseek и другие. Для получения качественного результата поиска необходимо предельно точно набирать запрос. Например, мы набрали: “image compression”. Ответом будут сотни тысяч ссылок, среди которых преобладает рекламная информация. Если же наберем нечто вроде “wavelet zerotree” или “optimal quantization”, ссылок будет значительно меньше, и почти все они будут информативными.

Так что ЖЕЛАЕМ УДАЧИ!

ОГЛАВЛЕНИЕ

Предисловие	3
Введение	5
Глава 1. Субполосное кодирование	7
1.1. Требования, предъявляемые к преобразованиям	8
1.2. Линейные преобразования конечных сигналов	9
1.2.1. Система фильтров анализа-синтеза	9
1.2.2. Каскадное соединение систем А-С	11
1.2.3. Представление субполосного кодирования при помощи аппарата матриц	13
1.2.4. Обратное преобразование	15
1.2.5. Ортогональное преобразование	16
1.3. Некоторые примеры преобразований	17
1.3.1. Преобразование Габора	17
1.3.2. Дискретное косинусное и перекрывающееся ортогональное преобразования	18
1.3.3. Пирамида Лапласа	19
1.4. Квадратурно-зеркальные фильтры	21
1.4.1. Построение КЗФ	23
1.4.2. Асимметричная система	23
1.5. О преимуществе преобразования при помощи блоков фильтров перед преобразованием Фурье	24
Глава 2. Основы теории вейвлет-преобразования	28
2.1. Непрерывное вейвлет-преобразование	28
2.2. Кратномасштабное представление функций	31
2.2.1. Представление функций при помощи вейвлетов	36
2.3. Вейвлет-ряды дискретного времени	39
2.4. Дискретное вейвлет-преобразование	42
2.4.1. Матричное описание DWT	42
2.4.2. Описание DWT посредством блоков фильтров	44
2.5. Гладкость базисных функций	47
Глава 3. Вейвлет-декомпозиция сигналов произвольной длины	50
3.1. Условия полного восстановления сигнала	50

3.2. Методика расчета фильтров, позволяющих осуществить полное восстановление сигнала	52
3.3. Продолжения сигналов, сохраняющие свойство полного восстановления	54
3.3.1. Периодическое продолжение	55
3.3.2. Симметричное продолжение	56
3.4. Эффективный метод продолжения для декомпозиции сигнала произвольной длины	58
3.5. Симметрично-периодическое продолжение сигнала	59
Глава 4. Сравнение вейвлет-фильтров с фильтрами, применяемыми при субполосном кодировании	61
4.1. Критерии для расчета фильтров	61
4.2. Построение обычных фильтров: фильтры Джонстона	64
4.3. Расчет вейвлет-фильтров	66
4.3.1. Расчет фильтров Добеши	66
4.3.2. Расчет пары биортогональных фильтров	67
4.4. Критерий оптимизации блоков фильтров, используемых при кодировании изображений	71
4.4.1. Выигрыш от субполосного кодирования	72
4.4.2. Оптимальное распределение бит	73
4.5. Сравнение характеристик обычных и вейвлет-фильтров	76
Глава 5. Адаптивные ортогональные преобразования	79
5.1. Пакеты вейвлетов (алгоритм одиночного дерева)	79
5.2. Алгоритм двойного дерева	82
5.3. Частотно-временное дерево	85
5.4. Сравнение обсуждаемых алгоритмов	86
5.4.1. Размерность библиотеки базисов	86
5.4.2. Вычислительная сложность алгоритмов	88
5.4.3. Эффективность кодирования изображений	88
Глава 6. Лифтинговая схема	90
6.1. Этап разбиения	91
6.2. Этап предсказания	91
6.3. Различные операторы предсказания	93
6.4. Этап обновления	98

Глава 7.	Целочисленное вейвлет-преобразование	105
7.1.	Целочисленные вейвлет-преобразования	105
7.2.	Лифтинговая схема и целочисленная биортогональная фильтрация	110
7.3.	Метод коррекции ошибок для получения целочисленного вейвлет-преобразования.	113
Глава 8.	Мультивейвлеты	117
8.1.	Блоки мультифильтров	117
8.1.1.	Основы теории блоков фильтров, изменяющихся во времени	117
8.1.2.	Построение блоков мультифильтров	121
8.1.3.	Итерирование блоков мультифильтров	121
8.2.	Мультивейвлеты	122
8.3.	Обработка сигналов в базисе мультивейвлетов	125
8.4.	Сбалансированные мультивейвлеты	128
Глава 9.	Потенциальные характеристики кодирования изображения с применением вейвлет-преобразования.	130
9.1.	Основные формулы и теоремы теории связи, относящиеся к кодированию с преобразованием при высоких скоростях (>1 бит/пиксел)	130
9.1.1.	Скалярное квантование с ограниченной энтропией	131
9.1.2.	Зависимость искажения от скорости	132
9.2.	Сжатие изображения при низких скоростях кодирования	134
9.2.1.	Функция искажение-скорость	135
9.2.2.	Оптимальный относительный размер интервала квантования	142
9.2.3.	Практическая проверка точности аналитических выражений	142
Глава 10.	Применение вейвлет-преобразования для сжатия изображения	147
10.1.	Базовый вейвлет-кодер изображения	148
10.1.1.	Выбор вейвлетов для сжатия изображения	148
10.1.2.	Осуществление преобразования на границах изображения	150
10.1.3.	Квантование	150
10.1.4.	Энтропийное кодирование	151
10.1.5.	Распределение бит	152
10.1.6.	Меры искажения, взвешенные с учетом восприятия человеком	153
10.2.	Новые идеи в области сжатия изображений, связанные с вейвлет-преобразованием.	153
10.3.	Кодирование посредством нульдерев	156

10.3.1.	Алгоритм Льюиса и Ноулеса	157
10.3.2.	Алгоритмы Шапиро и Саида-Перельмана	159
10.3.3.	Оптимизация нульдереьев по критерию скорость-искажение ...	161
10.4.	Частотно, пространственно-частотно адаптивные кодеры	162
10.5.	Использование зависимостей между вейвлет-коэффициентами внутри субполос	163
10.5.1.	Решетчатое квантование	163
10.5.2.	Субполосные кодеры с РК	165
10.5.3.	Моделирование и оценивание смеси распределений	167
10.6.	Современные направления исследований	169
Глава 11.	Видеокодеки семейства ADV6XX производства фирмы Analog Devices	172
11.1.	Принципы работы ADV601	173
11.2.	Использование микросхемы ADV601	175
Заключение		179
Приложение 1		181
Приложение 2		183
Библиография		192