

SP ++ 3.0 User Guide

By: Zhang
2011-02

With
Heart
Association
Force
Altogether
Create
Open
Source

Translated by: Javad Taghia

0 SP ++.....	4
0.1 SP ++ Overview	4
0.2 SP ++ installation (CodeBlocks)	5
0.3 SP ++ installation (VS2010)	5
0.4 SP ++ and Matlab programming	6
1 Vector class template.....	8
1.1 Basic vector class.....	8
1.2 common mathematical functions.....	11
1.3 common helper.....	12
Simple Timer 1.4	13
2Matrix class template.....	14
2.1 Basic Matrix class	14
2.2 common mathematical functions.....	18
2.3 real matrix and complex matrix decomposition Cholesky.....	19
2.4 real matrix and complex matrix decomposition LU	20
2.5 real matrix and complex matrix decomposition QR	21
2.6 real matrix and complex matrix decomposition SVD.....	22
2.7 real matrix and complex matrix decomposition EVD	23
2.8 Inverse and Generalized Inverse Matrix.....	25
3 Linear Equations	26
3.1 General Linear Equations	26
3.2 overdetermined and underdetermined linear equations	27
3.3 Morbid Linear Equations.....	28
4.1 Roots of Nonlinear Equations	28
4.2 Roots of Nonlinear Equations	29
4.3 Romberg numerical integration.....	29
5.1 Newton interpolation	30
5.2 cubic spline interpolation	30
5.3 Least Squares Fitting	31
6 Optimization	32
6.1 one-dimensional line search	32
6.2 steepest descent method	32

6.3 Conjugate Gradient Method	33
6.4 Quasi Newton method	34
7 Fourier Analysis	35
7.1 2 whole power of the FFT algorithm	35
7.2 any length FFT algorithm	36
7.3 Use common signal FFT	36
7.4 FFTW C ++ interface	37
8 Digital Filter Design	38
8.1 Common window function	38
8.2 base class filter design	38
8.3 FIR digital filter design	39
8.4 IIR digital filter design	40
9 Random Signal Processing	40
9.1 Random Number Generator	40
9.2 Probability and Statistics of commonly used functions	41
9.3 associated with the fast algorithm	42
10 Power spectrum estimation	43
10.1 classical spectral estimation method	43
10.2 parametric spectral estimation method	44
10.3 Characteristics of spectral estimation method	45
11 Adaptive filter	46
11.1 Wiener filter	46
11.2 Kalman Filter	47
11.3 LMS adaptive filter	47
11.4 RLS adaptive filter	48
12 Time-frequency analysis	49
12.1 windowed Fourier transform	49
12.3 Wigner-Wille distribution	52
13 Wavelet Transform	53
13.1 continuous wavelet transform	53
13.2 dyadic wavelet transform	55
13.3 DWT	56

14 Find and sort	59
14.1 binary search tree	59
14.2 balanced binary tree	60
14.3 The basic sorting algorithm.....	60
14.4 Huffman coding.....	61
15 References	62
16 Feeling	64
16.1 whim	64
16.4 matter of opinion.....	67
Figure 1 setup in Global Compiler.....	5
Figure 2 setting in VS2010	5
Figure 3 Matlab setting.....	6
Figure 4 Matlab setting.....	6
Figure 5 Matlab Setting.....	7
Figure 6 Matlab Setting.....	7
Figure 7 Matlab Setting.....	7
Figure 8 Matlab Test	8
Figure 9 Welch power spectrum estimation	44
Figure 10 positive and negative predictive least squares linear power spectrum estimation.....	45
Figure 11 ESPRIT power spectrum estimation.....	46
Figure 12 windowed Fourier transform.....	50
Figure 13 dual function.....	51
Figure 14 Discrete Gabor Transform	51
Figure 15 Wigner-Wille distribution Wigner-Wille distribution	53
Figure 16 continuous wavelet transform.....	55
Figure 17 dyadic wavelet transform, dyadic wavelet transform error analysis	56
Figure 18 DWT, discrete wavelet reconstruction.....	58

0 SP ++

Guide

0.1 SP ++ Overview

SP ++ (Signal Processing in C ++) is a signal processing and numerical calculations on open source C ++ Cheng Sequence library that provides the signal processing and numerical algorithm commonly used C ++ implementation.

All algorithms SP ++ C ++ class template method to achieve, organized together with the header file, so no need to compile local users, as long as the associated header file contains can be used in the project. "XXX.h" said the eclaration file, "XXX-impl.h" indicates that the corresponding implementation file. All functions and classes are located in the name space between "splab", **so when you want to use SP ++ namespace declaration: "using namespace splab".**

SP ++ was first published in the open-source Chinese community, blog address is: <http://my.oschina.net/zmjerry> . After that publish to Google Code, the following address: <http://code.google.com/p/tspl/> , because the name of the conflict, so Project Natori for TSPL (Template Signal Processing Library).

0.2 SP ++ installation (CodeBlocks)

1. The SP ++ 3.0 Unzip the package to a path, such as D: \ Program Files \ SP ++ 3.0;
2. Open CodeBlocks, in Settings-> Compiler and Debugger / Search directories / Compiler added to D: \ Program Files \ SP ++ 3.0 \ include, as below;

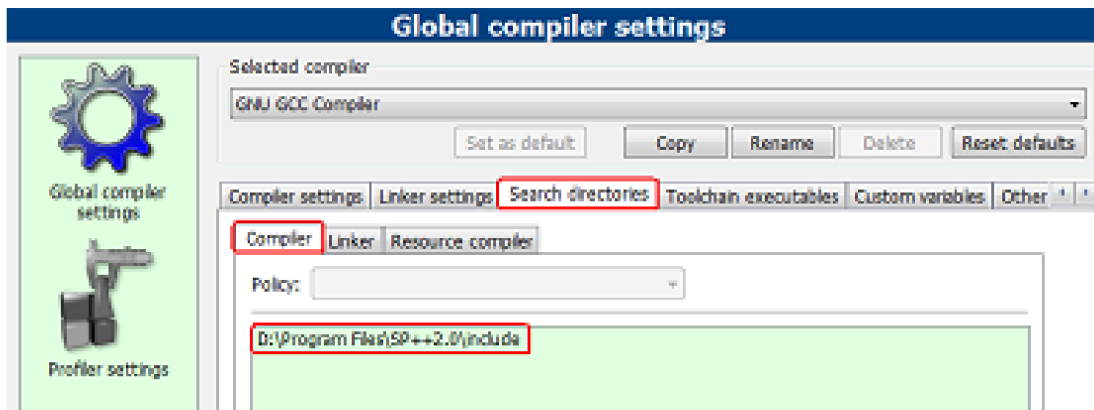


Figure 1 setup in Global Compiler

3. Build C ++ project test programs; e.g. Fir_test.cpp

0.3 SP ++ installation (VS2010)

1. The SP ++ 3.0 Unzip the package to a path, such as D: \ Program Files \ SP ++ 3.0;
2. VS2010 project established in Project-> Propertiesr of VC ++ Directories-> Include Directories

Add D: \ Program Files \ SP ++ 3.0 \ include, as below;

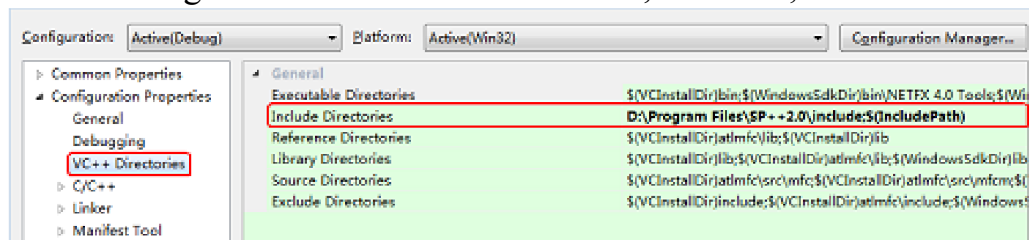


Figure 2 setting in VS2010

3. Established in VS2010 C ++ projects and compile Inverse_test.cpp.

0.4 SP ++ and Matlab programming

1. Build C ++ project, the project name in the Project Management window, right-click, select Properties lowermost Options, such as below.

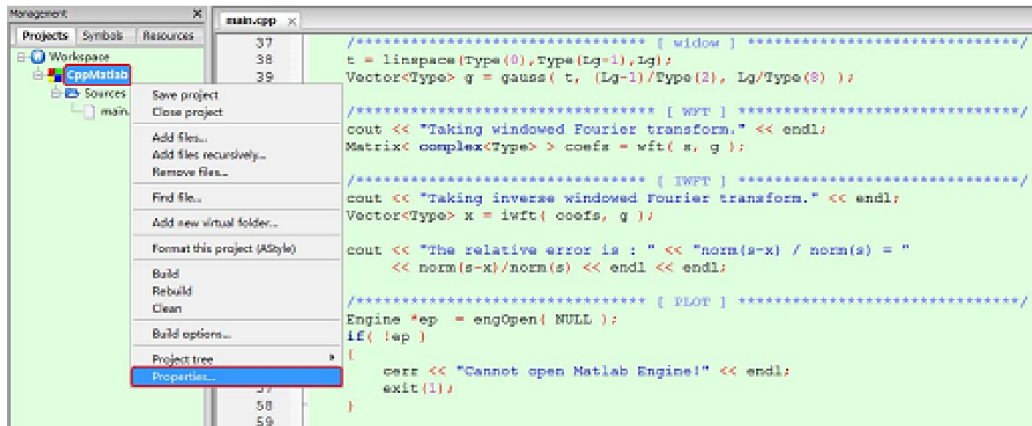


Figure 3 Matlab setting

2. Select the lower right corner of the window that opens Project's build options such as below;

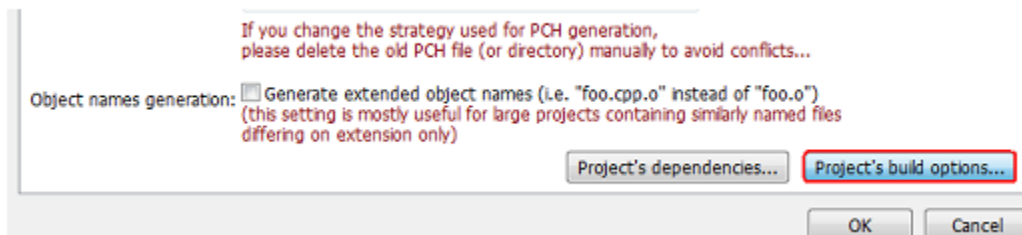


Figure 4 Matlab setting

3. Select Search directories options Compiler options in the window that opens, in which you join SP ++ 3.0 and Matlab \ extern in the include directory, such as shows.

Note: This must be added and upload SP ++ 3.0 \ include, and then load the Matlab \ extern \ include, because SP ++ 3.0 and Matlab has matrix.h headers, if loaded in reverse order, you cannot compile;

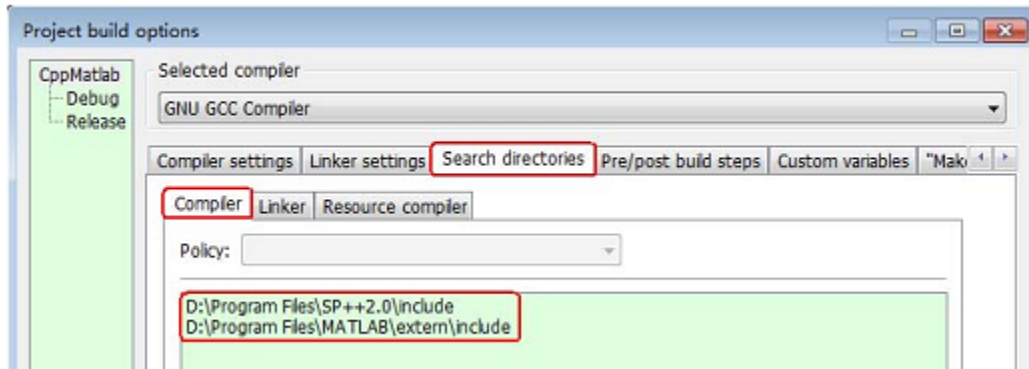


Figure 5 Matlab Setting

4 This step, selects search directories options Linker options in the window that opens, in which Canada Lib directory into Matlab-related, as below;

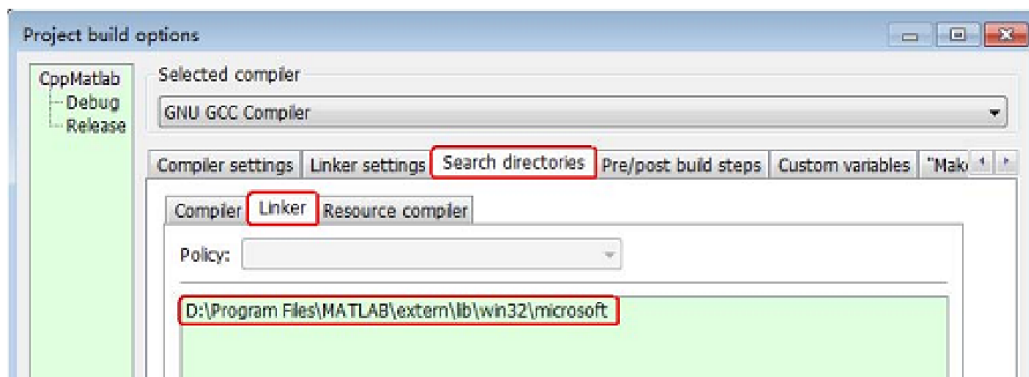


Figure 6 Matlab Setting

5 This step, selects Linker setting options in the window that opens, to which the relevant lib Matlab File, as shown;

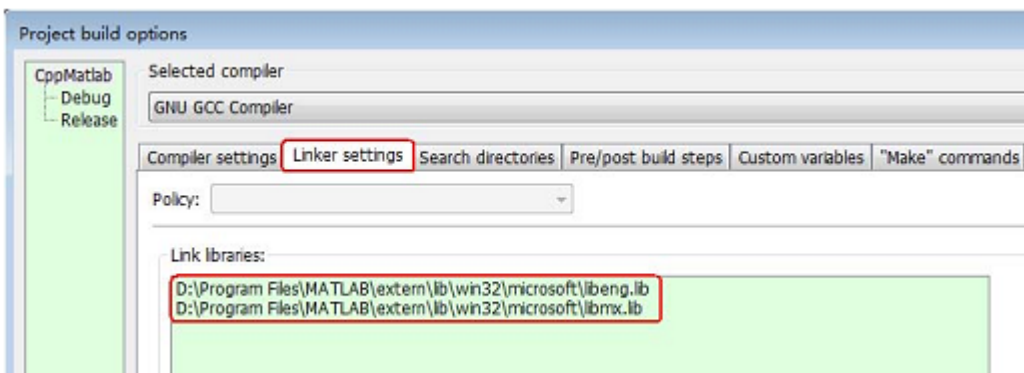


Figure 7 Matlab Setting

6. In the project, run the following test code;
The result:

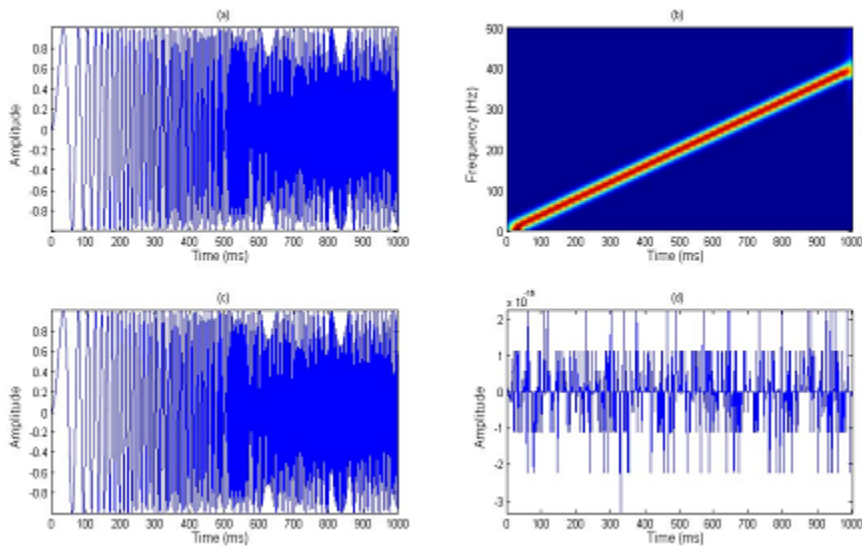


Figure 8 Matlab Test

1 Vector class template

1.1 Basic vector class

Vector class template `Vector <Type>` is a template class designed for linear algebra vector operations and design, supports a variety of computing real and complex vector of vectors. Contains a vector construct and destructor, see [Table 1-1](#) ; Vector The basic properties of the class extraction, see [Table 1-2](#) ; vector used in the calculation of operator overloading, see [Table 1-3](#) ; and some other commonly used functions, see [Table 1-4](#) .

To facilitate the said variable table are involved in the end indicate the type, can be roughly judged according to the variable name

Its type, such as the practice of using the vector `v` is represented by `a` to represent the array is represented by `n` number of elements, and so on. Tool

Function declarations and definitions can be found in the body "`vector.h`" and "`vector-impl.h`".

Table 1-1 constructor and destructor function vector class

Operation
Effect
Vector <Type> v
Create an empty vector
Vector <Type> v1 (v2)
Create a copy of the vector v2 v1
Vector <Type> v (n, x)

Create a constant vector**Vector <Type> v (n, a)****Create a vector through an array****v.Vectro <Type> ()****Destruction and free up space vector**

Table 1-2 vector class property acquisition

Operation**Effect****v.Type * ()****Vector pointer to an array type conversion****v.begin ()****Get the first element of the iterator****v.end****Get under the last element of an iterator****v.dim ()****Get dimension vector****v.size ()****Get a vector of size****v.resize (n)****Reset vector magnitude**

Table 1-3 vector class overloaded operator

Operation**Effect****v1 = v2****Vector for vector assignment****v = x****Constant for vector assignment****v [i]****0 offset index access****v (i)****An offset index access****-v****All the elements negated****v += x****Plus constant vector itself****v -= x****Less constant vector itself****v *= x****Vector multiplication constant itself****v /= x****Vector itself divided by a constant**

v1 += v2
Vector itself plus vector
v1 -= v2
Less vector vector itself
v1 *= v2
Vector vector multiply itself (by element)
v1 /= v2
Vector itself is divided by the vector (element by element)
v + x
Vector and constants and
x + v
Constant and vector sum
v1 + v2
Vector and vector sum
v - x
The difference between vector and constants
x - v
The difference between constant and Vector
v1 - v2
The difference between vector and vector
v * x
Vector product of the constant
x * v
And the product of a constant vector
v1 * v2
Vector and vector product (element by element)
v / x
Vector with constant quotient
x / v
Constant and vector quotient
v1 / v2
Vector and vector quotient (element by element)
>> V
Input vector
<< V
Output vector
Other Functions Table 1-4 vector class
Operation
Effect
sum (v)
Elements of the vector and

min (v)
Find the smallest element vector
max (v)
Seeking maximum element vector
swap (v1, v2)
Two elements of the vector exchange
norm (v)
Vector L2 norm
dotProd (v1, v2)
Inner product of vectors
linspace (a, b, n)
Generate arithmetic sequence
abs (cv)
Modulus values of complex vector
arg (cv)
Seeking phase angle phasor
real (cv)
Seeking the real part of the complex vector
imag (cv)
Seeking the imaginary part of the complex vector
complexVector (vr)
The real vector into a complex vector
complxVector (vr, vi)
Through real vector construct complex vector

Test code:

* Vector_test.cpp

1.2 common mathematical functions

In order to facilitate the numerical calculation, the number of mathematical functions commonly used in signal processing were overloaded, making it Vector objects can be used, such as [in Table 1-5](#) .

Table 1-5 Common Functions of vector version (element by element)

Operation
Effect
cos (v)
Vector cosine
sin (v)
Vector sine
tan (v)
Tangent vectors

acos (v)
Vector inverse cosine
asin (v)
Vector arcsine
atan (v)
Vector arctangent
exp (v)
Vector power function
log (v)
Vector natural logarithm function
log10 (v)
Vector base 10 logarithmic function
sqrt (v)
Vector square root
pow (v, v)
Vector Vector power function
pow (v, p)
Vector constant power function
pow (b, v)
Vector constant power function
gauss (v, u, r)
Vector Gauss function

Test code:

* Vectormath_test.cpp

1.3 common helper

Matlab toolbox provides a very rich, containing numerous studies in the field of common functions, [Table 1-6](#) Column

Out of some C functions in Matlab Signal Processing Toolbox frequently used ++ implementation.

Table 1-6 Common Functions in Matlab

Operation
Effect
mod (m, n)
Take m for n non-negative modulus values
ceil (m, n)
For m / n rounding up
reverse (v)
Vector Reverse
flip (v)
Vector Reverse

shift (v, n)**Zeros shift****cirshift (v, n)****Cyclic shift****fftshift (v, n)****FFT shift****dyadUp (v, oe)****Dyadic upsampling****dyadDown (v, oe)****Dyadic downsampling****fftInterp (v, factor)****Signal in the frequency domain interpolation****wkeep (v, n, first)****Extracting some elements of the vector****wkeep (v, n, direct)****Extracting some elements of the vector****wextend (v, n, direct, mode)****Vector Extension**

Test code:*** Utilities_test.cpp**

Simple Timer 1.4

To facilitate the running time of the test program, SP ++ provides a simple timer class, which provides

Interfaces such as [Table 1-7](#)
Table 1-7 timer

Operation**Effect****Timing time****Create a timer****time.start ()****Start time****time.stop ()****Stop the clock****time.read ()****Read Time**

Test code:*** Time_test.cpp**

2Matrix class template

2.1 Basic Matrix class

Matrix class template Matrix <Type> is a template class designed for linear algebra and matrix design, support

Various operations with complex matrix of real numbers. Contains a matrix structure and destructor, see [Table 2-1](#) ; basically belong to the class of matrices

Of extraction, see [Table 2-2](#) ; matrix used in the calculation of operator overloading, see [Table 2-3](#) ; And other commonly used matrix

Function, see [Table 2-4](#) .Specific function declarations and definitions can be found in "matrix.h" and "matrix -impl.h".

Note: The matrix is a row-store, the first dimension is the number of rows, the second dimension is the number of columns.

Table 2-1 constructor and destructor function matrix classes

Operation
Effect
Matrix <Type> m
Create an empty matrix
Matrix <Type> m2 (m1)
Create a copy of the matrix m1 m2
Matrix <Type> m (r, c, x)
Create a constant matrix
Matrix <Type> m (r, c, a)
Create a matrix by an array
m.Matrix <Type> ()
The destruction of the matrix and release space

Table 2-2 attribute matrices acquisition

Operation
Effect
m.Type * ()
Matrix pointer to an array type conversion
m.size ()
The total number of matrix elements
m.dim (d)
The first dimension of the matrix dimension d
m.rows ()
The number of rows of the matrix
m.cols ()
The number of columns of the matrix
m.resize (r, c)
Reallocation matrix size

m.getRow (r)

The first row of the matrix r extract

m.getColumn (c)

The first column of the matrix c extract

m.setRow (v)

Sets the first r rows of the matrix

m.setColumn (v)

The first column of the matrix set c

Table 2-3 Matrices overloaded operator

Operation

Effect

A1 = A2

Matrix matrix assignment

A = x

Constant matrix assignment

A [i] [j]

0 offset index access

A (i, j)

An offset index access

-A

All the elements negated

Matrix class template

A += x

Plus constant matrix itself

A -= x

Less constant matrix itself

A *= x

Matrix multiplication constant itself

A /= x

Matrix itself divided by a constant

A1 += A2

Matrix Matrix itself plus

A1 -= A2

Less matrix matrix itself

A1 *= A2

Matrix multiplication matrix itself (by element)

A1 /= A2

Matrix itself divided matrix (element by element)

x + A

Constant and matrices and

$A + x$
Matrix and constants and
$A1 + A2$
Matrix and Matrix sum
$A - x$
Difference matrix and constants
$x - A$
The difference between constant and matrices
$A1 - A2$
The difference between the matrix and the matrix
$A * x$
The product of the constant matrix
$x * A$
Product constant and matrices
$A * v$
Matrix by vector
$A1 * A2$
Matrix-matrix of
A / x
Matrix and constant quotient
x / A
Quotient constant and matrices
$A1 / A2$
Matrix providers and Matrices (element by element)
$>> A$
Enter the Matrix
$<< A$
Output matrix

Table 2-4 Other function matrix classes

Operation
Effect
$\text{optMult}(A, B, C)$
Optimized version of the matrix multiplication
$\text{optMult}(A, b, c)$
Optimized version of the matrix is multiplied by a vector
$\text{elemMult}(A, B)$
Multiplied by the matrix elements of the matrix
$\text{elemMultEq}(A, B)$
Matrix and the matrix elements of the assignment by multiplying
$\text{elemDiv}(A, B)$
Matrix and the matrix element by element division

elemDivEq (A, B)

Matrix and the matrix elements of the assignment by division

trMult (A, B)

Multiplying the matrix transpose matrix

trMult (A, b)

Multiplied by the vector matrix transpose

multTr (A, B)

Multiplied by the matrix transpose matrix

multTr (A, b)

Multiplied by the vector matrix transpose

multTr (b, c)

Vector by vector transpose

trT (A)

Matrix transpose

trH (A)

Matrix conjugate transpose

eye (n, x)

Generating unit matrix of order n

diag (A)

Diagonal matrix extract

diag (v)

Vector generated by the diagonal matrix

norm (A)

Frobenius matrix norm

swap (A, B)

Two elements of the matrix switching

sum (A)

By column matrix vector summation

min (A)

Matrix by column vector for the minimum

max (A)

Seeking maximum vector matrix by columns

mean (A)

Averaging by column matrix vector

abs (cA)

Modulus values of complex matrices

arg (cA)

Seeking the complex matrix of angle

real (cA)

Find the real part of a complex matrix

imag (cA)

Seeking the imaginary part of a complex matrix

complexMatrix (rM)

The real matrix designed for complex matrix

complexMatrix (rM, iM)

Real matrix generated by a complex matrix

Test code:

*** Matrix_test.cpp**

2.2 common mathematical functions

In order to facilitate the numerical calculation, the number of mathematical functions commonly used in signal processing were overloaded, making it

Matrix can be applied to objects, such as [Tables 2-5](#)

Table 2-5 Common Functions of matrix version (element by element)

Operation
Effect
cos (m)
Cosine matrix
sin (m)
Sine matrix
tan (m)
Tangent matrix
acos (m)
Matrix inverse cosine
asin (m)
Matrix arcsine
atan (m)
Matrix arctangent
exp (m)
Power function matrix
log (m)
Natural logarithm function matrix
log10 (m)
Matrix-10 logarithm function
sqrt (m)
Square root of a matrix
pow (m, m)
Vector matrix exponential function
pow (m, p)
Constant power function matrix
pow (b, m)
Constant matrix exponential function

Test code:

* `Matrixmath_test.cpp`

2.3 real matrix and complex matrix decomposition Cholesky

For a symmetric positive definite matrix A , can be expressed as $A = L * L'$, where L is a lower triangular matrix, which is

Cholesky decomposition of a symmetric positive definite matrix. Using this decomposition can solve linear equations, you can solve the matrix side Cheng, the specific function Cholesky class see [Table 2-6](#).

Table 2-6 Real Matrix Cholesky decomposition

Operation
Effect
Cholesky <Real> cho
Establish Cholesky class
cho.Cholesky <Real> ()
Cholesky destructor
cho.isSpd ()
Determine whether the symmetric positive definite matrix
cho.dec (A)
For real matrix A Cholesky decomposition
cho.getL ()
Get lower triangular matrix L
cho.solve (b)
Solution of equations $Ax = b$
cho.solve (B)
Solution of the matrix equation $AX = B$

Test code:

* `Cholesky_test.cpp`

Cholesky decomposition of a complex matrix such as [Table 2-7](#)

Table 2-7 Cholesky decomposition of a complex matrix

Operation
Effect
CCholesky <Real> cho
Establish CCholesky class
cho.CCholesky <Real> ()
CCholesky destructor
cho.isSpd ()
Determine whether the symmetric positive definite matrix
cho.dec (cA)
For complex matrix cA were Cholesky decomposition

cho.getL ()
Get lower triangular matrix L
cho.solve (b)
Solution of equations $Ax = b$
cho.solve (B)
Solution of the matrix equation $AX = B$

Test code:

* Ccholesky_test.cpp

2.4 real matrix and complex matrix decomposition LU

For an n-order matrix A, LU decomposition will be divided into a lower triangular matrix L and an upper triangular matrix U product,

Can make use of the matrix LU decomposition to solve the matrix determinant, linear equations and matrix equations, etc., see [Table 2-8](#), the function table provides for a real matrix and complex matrix can be applied. For the number of rows and columns of a matrix ranging,

LU decomposition can be equally successful.

Table 2-8 LU decomposition

Operation
Effect
LUD <Real> lu
Establish LUD category
lu.LUD <Real> ()
LUD destructor
lu.dec (cA)
For complex matrix cA triangular decomposition
lu.getL ()
Get lower triangular matrix L
lu.getU ()
Get the upper triangular matrix U
lu.det ()
Determinant calculation matrix
lu.isNonsingular ()
To determine whether the non-singular matrix
lu.solve (b)
Solution of equations $Ax = b$
lu.solve (B)
Matrix equation $AX = B$

Test code:

* Lud_test.cpp

2.5 real matrix and complex matrix decomposition QR

For a matrix A n row m column, QR decomposition of order m of a sub-master orthogonal matrix Q and an m rows and n columns

The upper triangular matrix R, satisfies $A = Q * R$, QR decomposition is always there, even for rank-deficient matrix. Using matrix

The QR decomposition can be solved with the least-squares solution, such as a minimum norm solution of overdetermined linear equations, see below for details

A chapter. QR decomposition of C ++ classes, see [Table 2-9](#).

Table 2-9 real matrix QR decomposition

Operation
Effect
QRD <Real> qr
Establish QRD class
qr.QRD <Real> ()
QRD destructor
qr.dec (A)
For real orthogonal matrix A triangular decomposition
qr.getQ ()
Get orthogonal matrix Q
qr.getR ()
Get the triangular matrix R
qr.isFullRank ()
To determine whether the full rank matrix
qr.solve (b)
Least-squares solution of linear equations $Ax = b$
qr.solve (B)
Least Squares Solution of Matrix Equation $AX = B$
Test code:
* Qrd_test.cpp

QR decomposition of complex matrix such as [Table 2-10](#)

Table 2-10 Complex QR Decomposition

Operation
Effect
CQRD <Real> qr
Establish CQRD class
qr.CQRD <Real> ()
CQRD destructor
qr.dec (cA)

For complex orthogonal matrix cA triangular decomposition

qr.getQ ()

Get orthogonal matrix Q

qr.getR ()

Get the triangular matrix R

qr.isFullRank ()

To determine whether the full rank matrix

qr.solve (b)

Least-squares solution of linear equations $Ax = b$

qr.solve (B)

Least Squares Solution of Matrix Equation $AX = B$

Test code:

*** Cqrd_test.cpp**

2.6 real matrix and complex matrix decomposition SVD

For a matrix A m rows and n columns, SVD decomposition of the main points of a m orthogonal matrix U of order, an m-row n

Columns and a diagonal matrix S orthogonal matrix of order n V, satisfy $A = U * S * V'$, SVD matrix decomposition always exists

A. Generalized SVD matrix decomposition can use Matrix rank, matrix norm and condition number 2 and matrix

Inverse, etc., see the specific [Table 2-11](#) .

Table 2-11 real matrix SVD decomposition

Operation

Effect

SVD <Real> sv

Establish SVD class

sv.SVD <Real> ()

SVD destructor

sv.dec (A)

For real matrix A singular value decomposition

sv.getU ()

Get left singular vectors U

sv.getV ()

Get the right singular vectors of V

sv.getSM ()

Get the singular value matrix

sv.getSV ()

Get singular vector

sv.norm2 ()

2 norm (maximum singular value) to calculate the matrix

sv.cond (b)

Condition number of a matrix is calculated

sv.rank (B)

Rank matrix calculation

Test code:

* Svd_test.cpp

SVD decomposition of complex matrices such as [Table 2-12](#)

Table 2-12 complex matrix SVD decomposition

Operation
Effect
CSVD <Real> sv
Establish CSVD class
sv.CSVD <Real> ()
CSVD destructor
sv.dec (cA)
CA for complex matrix singular value decomposition
sv.getU ()
Get left singular vectors U
sv.getV ()
Get the right singular vectors of V
sv.getSM ()
Get the singular value matrix
sv.getSV ()
Get singular vector
sv.norm2 ()
2 norm (maximum singular value) to calculate the matrix
sv.cond (b)
Condition number of a matrix is calculated
sv.rank (B)
Rank matrix calculation

Test code:

* Csvd_test.cpp

2.7 real matrix and complex matrix decomposition EVD

Matrix (or operator) is characterized by the decomposition is a very important concept for a matrix A, there is a positive

Orthogonal matrix V and a diagonal matrix D, meet $AV = V * D$, where the column vector V is the feature vector A and the D diagonal

Elements corresponding eigenvectors eigenvalues (may be plural), further explanation see "evd.h".

Some commonly used functions EVD <Type> class, see [Table 2-13](#) .

Table 2-13 EVD real matrix decomposition

Operation
Effect
EVD <Real> ev
Establish EVD class
ev.SVD <Real> ()
EVD destructor
ev.dec (A)
For real matrix A characteristic decomposition
isSymmetric ()
To determine whether the matrix symmetry
isComplex (tol)
Determine whether the complex eigenvalues of matrix
ev.getV ()
Get real eigenvectors V
ev.getCV ()
Get the complex feature vector CV
ev.getD ()
Get real eigenvalues D
ev.getCD ()
Get the complex eigenvalues CD

Test code:

* [Evd_test.cpp](#)

EVD complex matrix decomposition as [Table 2-14](#).

Table 2-14 EVD complex matrix decomposition

Operation
Effect
CEVD <Real> ev
Establish CEVD class
ev.CSVD <Real> ()
CEVD destructor
ev.dec (cA)
The feature on the complex matrix decomposition cA
isHertimian ()
Symmetric matrix to determine whether Hertimian
ev.getV ()
Get the feature vector V
ev.getD ()

Get eigenvalues D**ev.getRD ()****Get real eigenvalues RD**

Test code:

* Cevd_test.cpp

2.8 Inverse and Generalized Inverse Matrix

Matrix inversion algorithm is extremely important, has been widely used, but its computation is very large, a

Like a cube of N level. SP ++ matrix inversion algorithms provided include: out PCA Gauss elimination method, all the main element Gauss

Elimination method, LUD decomposition and Cholesky decomposition, which

Cholesky decomposition for symmetric positive definite matrix, which can

To reduce the amount of calculation, the specific function, see [Table 2-15](#) below.

Note: Many little derivation matrix inversion can be transformed into linear equations problem, so you can

Reduce the computational methods for solving linear equations related to refer to the relevant content of the next chapter.

Table 2-15 matrix inversion algorithm

Operation**Effect****inv (A, type)****General real matrix inversion algorithm****cinv (cA, type)****General complex matrix inversion algorithm****colPivInv (A)****Real matrix and complex matrix out PCA Gauss elimination method****cmpPivInv (A)****Real matrix and complex matrix full PCA Gauss elimination method**

Test code:

* Inverse_test.cpp

When the matrix is not square or non-full rank matrix, we need to take advantage of the generalized inverse matrix to solve some linear equations,

Generalized inverse matrix can be calculated by SVD decomposition, SP ++ provides a generalized inverse matrix and seeking real Complex Matrix

Solution algorithms, such as [Table 2-16](#) below.

Note: The test file "pseudoinverse_test.cpp" in the GCC compiler (CodeBlocks environment)

Under no mistake, there is an error in VS2010 environment, suggesting that the function template overloading ambiguity occurs, can the complex

Generalized inverse matrix function `pinv (cA, tol)` a change of name can be, for example, to `cpinv`.

Table 2-16 generalized inverse matrix

Operation
Effect
<code>pinv (A, tol)</code>
Real generalized inverse matrix
<code>pinv (cA, tol)</code>
Generalized inverse complex matrix
Test code:
* <code>Pseudoinverse_test.cpp</code>

3 Linear Equations

3.1 General Linear Equations

There are many conventional solution of linear equations, if the coefficient matrix for the general matrix of full rank, you can use out PCA

Gauss elimination method to solve or LU decomposition; if the coefficient matrix is a symmetric positive definite matrix, you can use Cholesky

Decomposition method to solve; if the coefficient matrix is tridiagonal matrix, you can catch up quickly solved by law. Matrix

In fact, the equation is solved multiple linear equations, it is consistent solution, specific functions, see [Table 3-1](#), the table provided

Function with real coefficients of linear equations and complex coefficients of linear equations can be applied.

Table 3-1 conventional method for solving linear equations

Operation
Effect
<code>gaussSolver (A, B)</code>
Gauss elimination method for solving matrix equation
<code>gaussSolver (A, b)</code>
Gauss elimination method for solving linear equations
<code>luSolver (A, B)</code>
LU decomposition method for solving the matrix equation
<code>luSolver (A, b)</code>
LU decomposition method for solving linear equations
<code>choleskySolver (A, B)</code>
Cholesky decomposition method for solving the matrix equation
<code>choleskySolver (A, b)</code>

Cholesky decomposition method for solving linear equations**utSolver (A, b)****Solving the triangular coefficient matrix of linear equations****ltSolver (A, b)****Lower triangular coefficient matrix solving linear equations****febsSolver (A, b)****Catch Solving tridiagonal equations****Test code:***** Linequs1_test.cpp**

.

3.2 overdetermined and underdetermined linear equations

If the number of rows in the coefficient matrix is greater than the number of columns, called the equation for the overdetermined system of linear equations, such equations

There is no exact solution, but the error can be obtained with a constant vector

between the minimum energy solution, namely least-squares solution, the main

To have a least squares method for solving generalized inverse matrix method and QR

decomposition. If the number of rows is less than the number of columns of the

coefficient matrix,

Overdetermined and underdetermined linear equations

This equation is called underdetermined linear equations, there are infinitely many solutions of this equation, but you can find it

Among these minimum energy a solution that minimal norm solution, the main

method for solving minimum norm generalized inverse matrix

France and the QR decomposition. Overdetermined and underdetermined linear

equations in the form of detailed specific function calls [Table 3-2](#) The lift table

Function for real coefficients of linear equations and complex coefficients of linear

equations can be applied.

Table 3-2 overdetermined and underdetermined linear equations method

Operation**Effect****lsSolver (A, b)****Generalized least squares solution of overdetermined inverse linear equations****qrLsSolver (A, b)****QR decomposition method for solving overdetermined linear equations****svdLsSolver (A, b)****SVD decomposition method for solving overdetermined linear equations****lnSolver (A, b)****Minimum norm solution of the generalized inverse underdetermined linear equations****qrLnSolver (A, b)**

QR decomposition method for solving linear equations underdetermined
svdLnSolver (A, b)

QR decomposition method for solving linear equations underdetermined

Test code:

* Linequs2_test.cpp

3.3 Morbid Linear Equations

If the coefficient matrix is rank deficient matrix, called the equation is ill equations, solving such equations party

Method is unstable, it cannot be solved by a conventional method. To mention solutions of high stability, must sacrifice accurate solution

Sex, the more common solution has truncated SVD method, damped SVD method and Tikhonov regularization method and so on. SP ++

Provides a method for solving these three specific function call format, see [Table 3-3](#) ,

The function of the real coefficients provided in table

Number of linear equations and complex coefficients of linear equations can be applied.

Note: The test file "linequs3_test.cpp" in the GCC compiler (CodeBlocks environment)

No error, error in VS2010 environment, suggesting that the function template overloading ambiguity occurs, the function can be changed

Names can be, for example, before the function will solve equations with complex coefficients plus letters 'c' and so on.

Table 3-3 Morbid linear approach solving method

Operation

Effect

tsvd (A, b, tol)

Truncated SVD method for solving linear equations rank deficiency

dsvd (A, b, sigma)

Damping SVD method for solving linear equations rank deficiency

tikhonov (A, b, alpha)

Tikhonov regularization method for solving linear equations rank deficiency

Test code:

* Linequs3_test.cpp

4.1 Roots of Nonlinear Equations

Engineering applications often have to calculate the roots of nonlinear equations, under normal circumstances to get the analytical solution is very difficult,

Therefore, it is important to numerical solution. Three common methods Roots of nonlinear equations SP ++ implemented, namely two

Points method, Newton method and secant method, as described in [Table 4-1](#) .Where the parameter "f" is the function object, representing the non-linear side Process, defined in the file "nlfunc.h" in.

Table 4-1 Solving Nonlinear Equation

Operation
Effect
bisection (f, a, b, tol)
Roots of nonlinear equations dichotomy
newton (f, x0, tol, maxItr)
Newton Method for Solving Nonlinear Equation
secant (f, x1, x2, tol, maxItr)
Secant Method for Solving Nonlinear Equation

Test code:

* Nle_test.cpp

4.2 Roots of Nonlinear Equations

Solving nonlinear equations of nonlinear equations than more complex, usually by iterative method,

For example Seidel iterative method and Newton iteration method, see [Table 4-2](#) .Where the parameter "G" and "F" for function

Like, on behalf of nonlinear equations, defined in the file "nlfuncs.h" in.

Table 4-2 Roots of Nonlinear Equations

Operation
Effect
seidel (G, X0, tol, maxItr)
Seidel method Roots of Nonlinear Equations
newton (F, x0, tol, eps, maxItr)
Newton iterative method for finding roots of nonlinear equations

Test file:

* Nle_test.cpp

4.3 Romberg numerical integration

SP ++ provides a Romberg numerical integration algorithms, such as [Table 4-3](#) below. Where the parameter "f" for the function

Like, on behalf of the integrand function, defined in the file "integrand" in.

Table 4-3 Romberg numerical integration

Operation
Effect

romberg (f, a, b, tol)

Romberg numerical integration algorithm

Test code:

* [Integral_test.cpp](#)

5.1 Newton interpolation

For a given $N + 1$ points, these points can strike through the N -order polynomial interpolation by Newton.

But mainly for the end of the interpolation polynomial interpolation order, because higher-order polynomial usually sharp oscillations in

Will introduce a very large error between the interpolated points.

SP ++ class in Newton interpolation `NewtonInterp <Type>` The use of methods, see [Table 5-1](#) And with three

Splines and least-squares fit of the common heritage of the general class template interpolation `Interpolation <Type>`, the relevant class

Statement can be found in "interpolation.h".

Table 5-1 Newton interpolation

Operation

Effect

`NewtonInterp <Type> intp (xi, yi)`

Creating Newton interpolation classes

`intp. NewtonInterp <Type>`

Newton interpolation class destructor

`intp.calcCoefs ()`

Calculating the coefficients of the polynomial interpolation

`intp.evaluate (x)`

Calculate the coordinates of the given function values

`intp.getCoefs ()`

Get interpolation polynomial coefficients

Test code:

* [Newtoninterp_test.cpp](#)

5.2 cubic spline interpolation

Cubic spline interpolation method for each interpolation interval cubic polynomial approximation, and to ensure the interval endpoints plug

The value of its first derivative of the polynomial are continuous. Therefore, the resulting interpolation polynomial good nature, and for more

Point interpolation has a high degree of approximation. Use cubic spline with the same Newton, see [Table 5-2](#) .

Table 5-2 cubic spline interpolation method

Operation
Effect
Spline3Interp <Type> intp (xi, yi, d2l, dwr)
Create a cubic spline interpolation class
intp. Spline3Interp <Type>
Destructor cubic spline interpolation class
intp.calcCoefs ()
Calculating the coefficients of the polynomial interpolation
intp.evaluate (x)
Calculate the coordinates of the given function values
intp.getCoefs ()
Get interpolation polynomial coefficients
Test code:
* Spline3interp_test.cpp

5.3 Least Squares Fitting

Given the observations they seek to comply with the required function curve fitting, the most common type of fitting square

France is the least squares fit to the observed data and the method to be proposed as a function of the mean square error between the proposed guidelines strike

Bonding parameters. The more common practice is a linear least squares fit (many non-linear least squares fitting can be transferred

Into a linear least squares fit), which is known to be a function of fitting a linear combination of function clusters.

SP ++ provides a linear least squares fit class LSFitting <Type>, where the constructor "f" is

Known function clusters, which defined "fitcurves.h", specific methods, see [Table 5-3](#).

Table 5-3 least-squares fit

Operation
Effect
LSFitting <Type> lsf (xi, yi, f)
Create a least-squares fit classes
lsf. LSFitting <Type>
Least-squares fit class destructor
lsf.calcCoefs ()
Fit function coefficient calculation
lsf.evaluate (x)
Calculate the coordinates of the given function values

lsf.getCoefs ()
Get fit function coefficients

Test code:

* Lsfit_test.cpp

6 Optimization

6.1 one-dimensional line search

All optimization methods are based on the decline in the direction required by the one-dimensional search to determine the step length, step length decreases directly affect the convergence rate optimization method, so one-dimensional search algorithm is a very critical step optimization.

SP ++ provides an inexact one-dimensional search algorithm class LineSearch

<DType, Ftype>, and can be obtained to determine the

Count the number of steps when the objective function. Specific function, see [Table 6-1](#) . Wherein "func" is the objective function, defined

See "objfunc.h" headers.

Table 6-1 one-dimensional line search

Operation
Effect
LineSearch <DType, Ftype> ols
Create a one-dimensional search for classes
ols. LineSearch <DType, Ftype> ()
One-dimensional search class destructor
ols. getStep (func, x0, dk, maxItr)
Strike one dimensional search step
ols. getFuncNum ()
Get the number of calculations of the objective function
ols. isSuccess ()
Determine the success of a one-dimensional search

6.2 steepest descent method

Steepest descent method (ie, gradient method) is one of the oldest and most simple optimization algorithm, the advantage of stability

Relatively high, the disadvantage is the convergence rate is very slow. So the convergence rate is not very high and high dimensional stability problems

Relatively poor problem, which is still a very useful algorithm.

SP ++ class in the steepest descent SteepDesc <DType, Ftype> provides seeking optimal value, the minimum function value ladder Modulus values and the objective function of the number of calculations and other functions, as described in [Table 6-2](#) .

Table 6-2 steepest descent method

Operation
Effect
SteepDesc <DType, Ftype> fmin
Creating the steepest descent method class
fmin.SteepDesc <DType, Ftype> ()
Destructor steepest descent method class
fmin.optimize (func, x0, tol, maxItr)
Specify the parameters for the minimum function
fmin.getOptValue ()
Obtain the optimal value of the argument
fmin.getGradNorm ()
Get the modulus of the gradient vector of the iterative process
fmin.getFuncMin ()
Get the minimum function
fmin.getItrNum ()
Get iterations

Test code:

* Steepdesc_test.cpp

6.3 Conjugate Gradient Method

Conjugate gradient method to determine the direction of each iteration of the search by generating conjugate direction, with the second and termination

A higher rate of convergence, and very little storage space required, so for some very high-dimensional optimization problem

Apply. However, when the objective function is not a quadratic function (in practice often not quadratic objective function) is not

N iteration can get the optimal value, so the need to re-start technology to determine the descent direction. And when the target

When the function is not well approximated by a quadratic function, the convergence rate is relatively steepest descent method does not have obvious advantages

Potential.

SP ++ conjugated gradient type ConjGrad <DType, Ftype> functions provided with the steepest descent method as Phase

Like, see [Table 6-3](#) , Where the default re-start times for the objective function of dimension n.

Table 6-3 Conjugate Gradient Method

Operation
Effect
ConjGrad <DType, Ftype> fmin
Create a conjugate gradient method class
fmin. ConjGrad <DType, Ftype> ()
Destructor Conjugate Gradient Method class
fmin. optimize (func, x0, tol, maxItr)
Specify the parameters for the minimum function
fmin.getOptValue ()
Obtain the optimal value of the argument
fmin.getGradNorm ()
Get the modulus of the gradient vector of the iterative process
fmin.getFuncMin ()
Get the minimum function
fmin.getItrNum ()
Get iterations
Test code:
* Conjgrad_test.cpp

6.4 Quasi Newton method

Newton method proposed by the gradient vector and matrix correction formula to approximate the objective function Hess, also has

The second termination, and the relative steepest descent method and conjugate gradient method has a high rate of convergence. The more common two Class of Quasi Newton algorithms DFP and FBGS, is now recognized for solving nonlinear unconstrained optimization problems most

Good algorithms. But for large-scale optimization problems, the proposed Newton requires a large storage space, and matrix operation

Count the time it takes the shortcomings highlighted.

SP ++ provides a BFGS quasi Newton algorithm, which call format and the steepest descent method and the conjugate gradient method

The same as described in [Table 6-4](#) .

Table 6-4 quasi Newton method

Operation
Effect
BFGS <DType, Ftype> fmin
Create a quasi Newton method class
fmin. BFGS <DType, Ftype> ()
Newton method proposed class destructor

fmin.optimize (func, x0, tol, maxItr)
Specify the parameters for the minimum function
fmin.getOptValue ()
Obtain the optimal value of the argument
fmin.getGradNorm ()
Get the modulus of the gradient vector of the iterative process
fmin.getFuncMin ()
Get the minimum function
fmin.getItrNum ()
Get iterations

Test code:

* Bfgs_test.cpp

7 Fourier Analysis

7.1 2 whole power of the FFT algorithm

SP ++ to achieve a power of two for the whole length of the FFT algorithm, FFTMR <Type> class, using a specific group

8, 4, and mixed base algorithm based group 2, and the use of economic storage mode, see [Table 7-1](#) .Length is an integer power of two

FFT computation efficiency signal is very high, so many practical applications have taken this approach.

Table 7-1 length is a power of two FFT algorithm

Operation
Effect
FFTMR <Type> ft
Creating FFTMR class
ft. FFTMR <Type> ()
FFTMR class destructor
ft.fft (cxn)
Fourier transform complex signal
ft.fft (rxn, Xk)
Fourier transform of a real signal
ft.ifft (cXk)
Inverse Fourier transform complex signal
ft.ifft (Xk, rxn)
Inverse Fourier transform real signal

Test code:

* Fftmr_test.cpp

7.2 any length FFT algorithm

SP ++ FFT algorithm is implemented in any length, FFTPF <Type> class, the algorithm uses the prime factor scores

Method to calculate any length DFT, see specific function [Table 7-2](#) . If you need a lot of computing FFT, recommends calling

FTTW, in order to improve the computational efficiency.

Table 7-2 FFT algorithm of arbitrary length

Operation
Effect
FFTPF <Type> ft
Creating FFTPF class
ft. FFTPF <Type> ()
FFTPF class destructor
ft.fft (rxn, Xk)
Fourier transform of a real signal
ft.ifft (Xk, rxn)
Inverse Fourier transform real signal
ft. fft (cxn, Xk)
Fourier transform complex signal
ft.ifft (Xk, cxn)
Inverse Fourier transform complex signal
Test code:
* Fftpf_test.cpp

7.3 Use common signal FFT

For ease of use, the whole power of the FFT algorithm with arbitrary length two packages together, when the signal length

2 The whole power of the call FFTMR class function, improve computational efficiency; otherwise, when the signal length is not equal to 2

When the power in the whole, the call function FFTPF class. Packaged functions see [Table 7-3](#) .

Table 7-3 FFT call ordinary signals

Operation
Effect
fft (s)
Discrete Fourier Transform to calculate the real signal
fft (cs)
Discrete Fourier transform calculation of complex signals
ifft (cS)
Calculate the inverse discrete Fourier transform complex signal

fftr2c (s)
Discrete Fourier Transform to calculate the real signal
fftc2c (cs)
Discrete Fourier transform calculation of complex signals
ifftc2r (cS)
Real inverse discrete Fourier transform calculation of complex signals
ifftc2c (cS)
Calculate the inverse discrete Fourier transform complex signal

Test code:

* Fft_test.cpp

7.4 FFTW C ++ interface

MIT scholar FFTW is a C language used to calculate the length of any one or more dimensions of discrete

Fourier transform library, the computational efficiency of the library is very high, has been widely used in signal processing is necessary

Essential for a C language library functions. SP ++ for FFTW latest version (3.2.2) in a one-dimensional FFT provides a C ++

Interface, e.g. [Table 7-4](#) below. The table functions are template function, support float, double, and long double

Three types of data.

Table 7-4 FFTW C ++ interface

Operation
Effect
fft (rxn, Xk)
Fourier transform of a real signal
fft (cxn, Xk)
Fourier transform complex signal
ifft (Xk, rxn)
Inverse Fourier transform real signal
ifft (Xk, cxn)
Inverse Fourier transform complex signal

Test code:

* Fftw_test.cpp

8 Digital Filter Design

8.1 Common window function

[Table 8-1](#) A list of commonly used form of window function call, where the parameter "amp" as a function of the magnitude of the window, with When you specify the return type of the window function, the results with the same window functions in Matlab.

Table 8-1 commonly used window function

Operation
Effect
<code>window (wnName, N, amp)</code>
Generate a window function type "wnName" the
<code>window (wnName, N, alpha, amp)</code>
Generate a window function type "wnName" the
<code>rectangle (n, amp)</code>
Rectangular window
<code>bartlett (n, amp)</code>
Bartlett window
<code>hanning (n, amp)</code>
Hanning window
<code>hamming (n, amp)</code>
Hamming window
<code>blackman (n, amp)</code>
Blackman window
<code>kaiser (n, alpha, amp)</code>
Kaiser window
<code>gauss (n, alpha, amp)</code>
Gauss window

Test code:

* `Window_test.cpp`

.

8.2 base class filter design

All filters have in common, such as frequency selection characteristic frequency passband cutoff frequency, gain, etc.,

Therefore, these commonalities can be extracted as a base class for filter design, such as [Table 8-2](#) Fig.

Table 8-2 Digital Filter Design base class

Operation

Effect
DFD f (select)
Create a DFD class
f. DFD <Type> ()
DFD class destructor
f. setParams (fs, f1, a1, f2, a2)
Filter design based on given parameters
f. setParams (fs, f1, a1, f2, f3, a2, f4, a3)
Filter design based on given parameters
f. dispInfo (fs, f1, a1, f2, f3, a2, f4, a3)
Display filter design results

8.3 FIR digital filter design

Finite impulse response (FIR) filter having a linear phase characteristic, and therefore in some of the more sensitive to phase

The occasion has been widely used. And FIR no feedback, so stability is better, of course, its frequency selection

Optional relatively poor. SP ++ implements the FIR design method based on window functions, specifically, see [Table 8-3](#) .

Table 8-3 FIR digital filter design

Operation
Effect
FIR <Type> f (select, win)
Creating FIR class
FIR <Type> f (select, win, a)
Creating FIR class (for Kaiser and Gauss window)
f.FIR <Type> ()
FIR class destructor
f.design ()
Filter design based on given parameters
f.dispInfo ()
Display filter design results
f.getCoefs ()
Get the filter coefficients

Test code:

* Fir_test.cpp

8.4 IIR digital filter design

Infinite impulse response (IIR) filter uses feedback, so it has a high frequency selectivity, but at the same

Also lost a linear phase characteristics and stability of the FIR is not as good. IIR filters can be set via analog

Meter, while the analog filter has a very mature design method, which gives IIR design brings great benefits. SP ++ solid

Now based bilinear transform IIR design method, as described in [Table 8-4](#) .

Table 8-4 IIR digital filter design

Operation
Effect
IIR <Type> f (select, method)
Creating IIR class
f.IIR <Type> ()
IIR class destructor
f.design ()
Filter design based on given parameters
f.dispInfo ()
Display filter design results
f.getNumCoefs ()
Get molecular filter coefficients
f.getDenCoefs ()
Get denominator filter coefficients

Test code:

* Iir_test.cpp

9 Random Signal Processing

9.1 Random Number Generator

Numerical computation and signal processing often produce some given distribution of random sequence to simulate the actual letter

No noise, Gauss white noise, such as frequently used. Random Number SP ++ provides a common distribution with random

Generate a sequence of functions, specifically including uniform, normal, exponential distribution, Rayleigh distribution, Poisson

Distribution, Bernoulli distribution, see [Table 9-1](#) .

Table 9-1 random number generator with a random sequence

Operation
Effect

randu (s, a, b)
Uniformly distributed random numbers

randn (s, u, sigma)
Normally distributed random numbers

rande (s, beta)
Exponentially distributed random numbers

randr (s, sigma)
Rayleigh distributed random numbers

randp (s, lambda)
Poisson distributed random numbers

randb (s, p)
Bernoulli distributed random numbers

randu (s, a, b, N)
Uniformly distributed random sequence

randn (s, u, sigma, N)
Normally distributed random sequence

rande (s, beta, N)
Random sequence exponential distribution

randr (s, sigma, N)
Rayleigh distributed random sequence

randp (s, lambda, N)
Poisson distribution random sequence

randb (s, p, N)
Bernoulli distributed random sequence

Test code:
*** Random_test.cpp**

9.2 Probability and Statistics of commonly used functions

"Statistics.h" header file provides a random signal processing of some commonly used algorithms such as random variables

Mean, median, variance, skewness, kurtosis, and the standardization of random variables and probability density function estimation algorithm,

See [Table 9-2](#) .

Table 9-2 Common Functions of probability and statistics

Operation

Effect

mid (v)
The median amount of orientation

mean (v)
Calculating the mean of the random variable

var (v)
Calculate the variance of the random variable
stdVar (v)
Calculating the difference between the standard random variable
standard (v)
Standardized random variable
skew (v)
Calculation of random variables skewness
kurt (v)
Calculation of random variables kurtosis
pdf (v, lambda)
Estimate the probability density function of the random variable

Test code:
*** Statistics_test.cpp**

.

9.3 associated with the fast algorithm

Correlation is a random signal processing operation often used, according to the relationship associated with the convolution may be related transfer

Into the convolution operation, and thus can use the convolution theorem to the relevant time domain transformed to the frequency domain is calculated, using the FFT

The time complexity of the algorithm can be from N^2

Reduced to $N \log_2 N$, greatly improve the computational efficiency. SP ++ provides a Autocorrelation and cross-correlation functions, such as [Table 9-3](#) As shown in the results and Matlab same parameters can "opt"

Select Normal correlation operation, there is a partial correlation estimates and unbiased correlation estimates.

Table 9-3 and fast convolution algorithm related

Operation
Effect
corr (x, opt)
X is calculated autocorrelation
corr (x, y, opt)
Calculating a cross-correlation of x and y
fastCorr (x, opt)
By the FFT calculation of the autocorrelation x
fastCorr (x, y, opt)
Calculating x and y of the cross-correlation by FFT

Test code:

* Correlation_test.cpp

10 Power spectrum estimation

10.1 classical spectral estimation method

By limiting observations to estimate the power of a random signal distribution in the frequency domain power spectrum of a signal is the random Estimates. SP ++ provides a classical spectral estimation algorithms, including correlation function, periodogram and several improved Welch Phase diagram, as [Table 10-1](#) Fig. These types of classical spectral estimation algorithm is characterized by stability is better, but a spectrum Lower resolution.

Table 10-1 classical spectral estimation method

Operation
Effect
correlogramPSE (xn, L)
Correlation function spectrum estimation
periodogramPSE (xn, wn, L)
Periodogram spectral estimation
bartlettPSE (xn, M, L)
Bartlett spectrum estimation method
welchPSE (xn, wn, K, L)
Welch spectral estimation method
btPSE (xn, wn, L)
BT spectral estimation method

Test code:

* Classicalpse_test.cpp

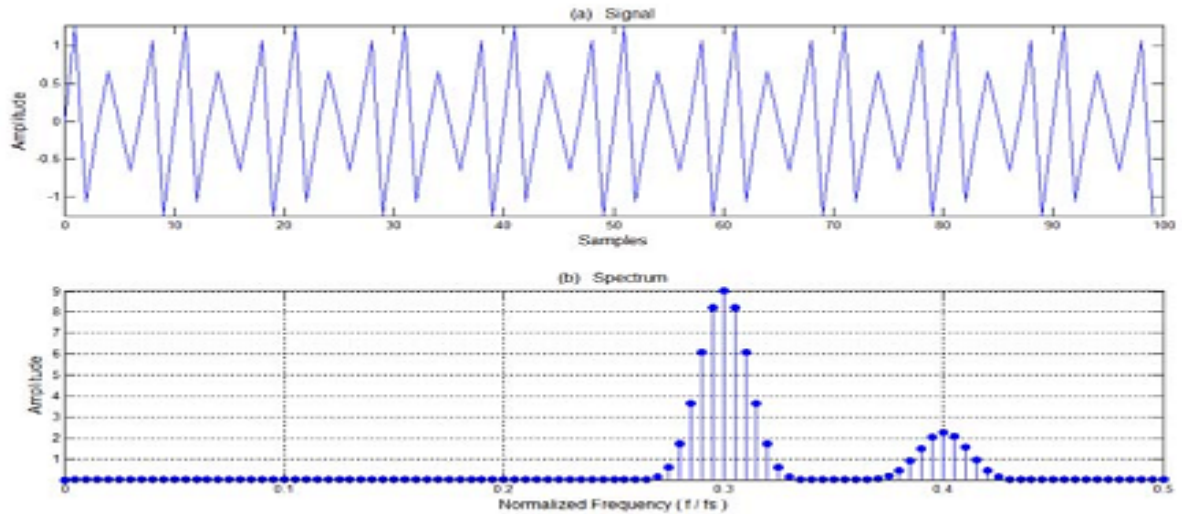


Figure 9 Welch power spectrum estimation

10.2 parametric spectral estimation method

Parametric spectral estimation method assumes that the random sequence to meet certain structures, such as the AR model, and so on, and from the view of The measured data to estimate the parameters of the described structure, and to determine the power spectrum of a random sequence by the estimated parameters. Such methods are specific high frequency resolution, but the model structure have sufficient prior information, otherwise it will produce Great estimation error. SP ++ provides four types of parametric spectral estimation methods have AR models and ARMA models Specifically as [Table 10-2](#) below.

Table 10-2 parametric spectral estimation method

Operation
Effect
yulewalkerPSE (xn, p, sigma)
Yule-Walker spectral estimation method
burgPSE (xn, p, sigma)
Burg spectral estimation method
fbplsPSE (xn, p, sigma)
Positive and negative predictive least squares linear spectral estimation method
armaPSD (ak, bk, sigma, L)
ARMA model spectrum estimation method

Test code:

* Parametricpse_test.cpp

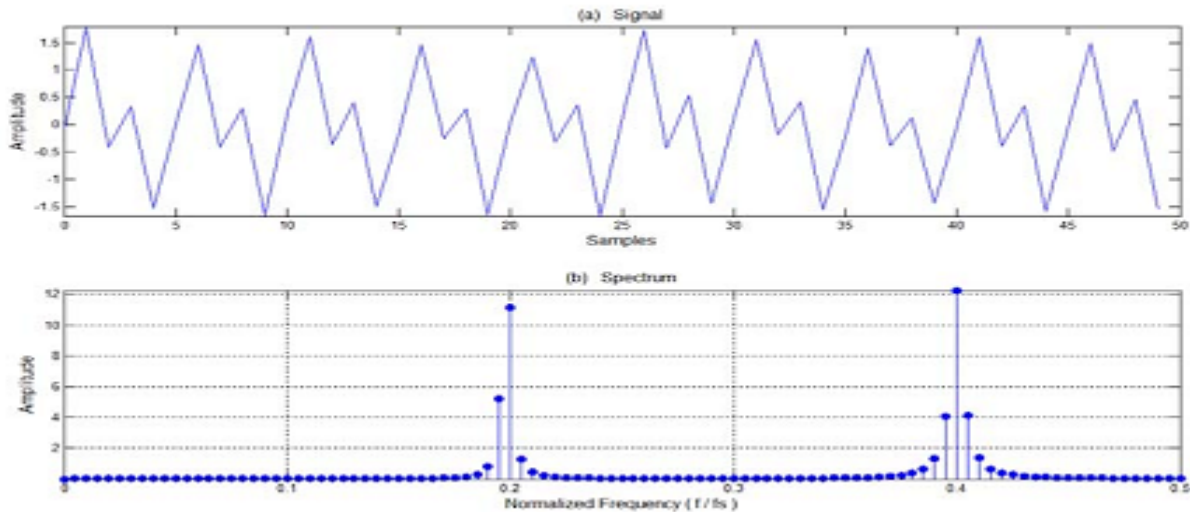


Figure 10 positive and negative predictive least squares linear power spectrum estimation

10.3 Characteristics of spectral estimation method

Spectral characteristics of the self-correlation matrix estimation method of random signal is decomposed into a signal subspace and noise subspace

Room, then use orthogonal frequency components of both estimates of random signals. This method is noise pollution from several

Superposition of a sine wave signal is very effective, SP ++ offers four feature-based analysis of spectral estimator

Method, e.g. [Table 10-3](#) below.

Table 10-3 Characteristics of spectral estimation method

Operation

Effect

caponPSE (xn, M, L)

Capon spectral estimation method

musicPSE (xn, M, p, L)

MUSIC spectral estimation method

pisarenkoPSE (xn, M, p, L)

Pisarenko spectral estimation method

espritPSE (xn, M, p)

ESPRIT spectral estimation method

Test code:

* Eigenanalysispse_test.cpp

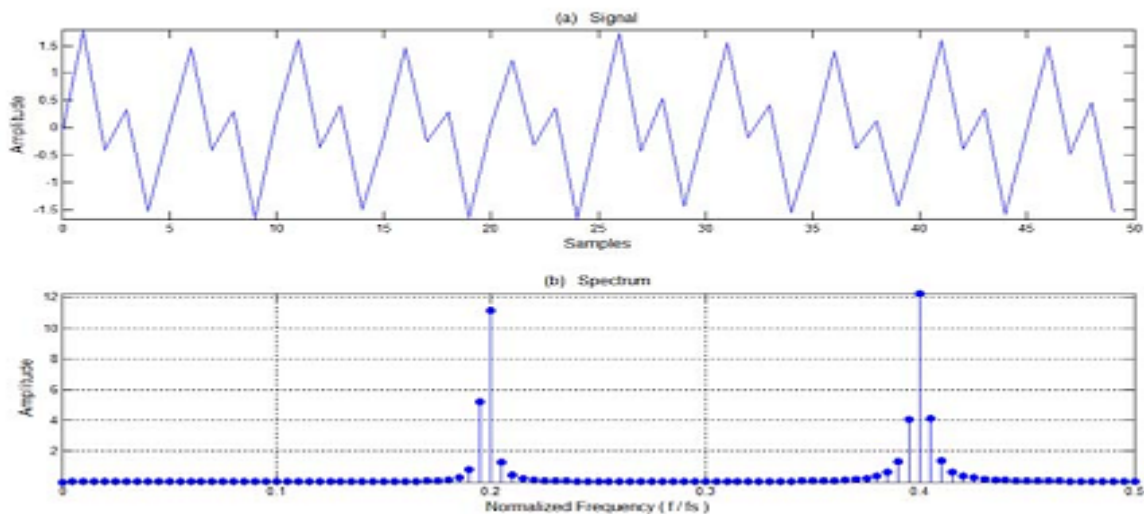


Figure 11 ESPRIT power spectrum estimation

11 Adaptive filter

11.1 Wiener filter

Wiener filter is a minimum mean square error sense optimal linear filter for stationary signals. Can

By rescue Wiener-Hopf equation (using Levinson-Durbin fast recursive algorithm) to get its time domain solution. In

Frequency domain as follows: The smaller band attenuation greater the SNR, the greater the smaller the signal to noise ratio of the attenuation band, and thus the maximum

The extent to filter out the noise, retention signal. SP ++ provides a Wiener filter and Wiener predictor two algorithms,

Also includes solving Toeplitz coefficient matrix of linear equations Levinson recursive algorithms such as [Table 11-1](#) Fig.

Table 11-1 Wiener filter

Operation

Effect

wienerFilter (x, d, p)

P-order Wiener filter

wienerPredictor (x, p)

P-order Wiener predictor

levinson (t, b)

Levinson algorithm for Toeplitz equations

Test code:

* Wiener_test.h

11.2 Kalman Filter

Kalman filter is also optimal linear estimator minimum mean square error sense, for smooth and non-equilibrium

The process can be applied, which can be regarded as an adaptive Wiener filter. The main idea is to use the "new

Interest "and the Kalman gain is calculated correction term, to estimate the internal state of a linear dynamic system variables by recursive methods.

SP ++ provides a conventional Kalman filter algorithm, such as [Table 11-2](#) below.

Table 11-2 Kalman Filter

Operation
Effect
kalman (A, C, Q, R, xp, y, V)
Kalman filter, a detailed description see the code comments

Test code:

* Kalman.h

11.3 LMS adaptive filter

LMS (Least Mean Squares) is the most common adaptive filtering algorithm, it is replaced with the current error

Expected error, adjust the filter coefficients by gradient descent method to track changes in the input signal or system,

Achieve adaptive purpose. Step factor has an important influence on the performance of the algorithm, a big step factor provides fast

Tracking features, while producing a large amount of imbalance and instability, and vice versa versa. LMS There are many types, SP ++

Provides three kinds of commonly used LMS algorithm, namely the general LMS algorithm, LMS-Newton algorithm and normalized LMS count

France, calling the form as [Table 11-3](#) Fig.

Table 11-3 LMS adaptive filtering algorithm

Operation
Effect
lms (x, d, w, mu)
Conventional LMS algorithm
lmsNewton (x, d, w, mu, alpha, delta)
LMS-Newton algorithm
lmsNormalize (x, d, w, rho, gamma)
Normalized LMS algorithm

Test code:
 * Lms_test.cpp

11.4 RLS adaptive filter

RLS (Recursive Least Square) algorithm and LMS algorithms, not under the minimum mean square sense of self
 Adaptive filter, but in order to solve the error signal as a function of the cost issue by recursive weighted least squares method,
 Thereby update the filter coefficients. The biggest advantage is that RLS algorithm convergence speed, good stability, but at the
 Calculate the difference between large and tracking performance for the price (of course, this can be adjusted by forgetting factor, the forgetting factor
 Small, the convergence faster, better tracking ability, but the larger the offset amount, stability worse). SP ++ provides a 5
 Class RLS adaptive filtering algorithm, the traditional RLS algorithms, steady and rapid transverse RLS algorithm, ordinary lattice RLS algorithms,
 Error feedback lattice RLS algorithm and RLS algorithm based on QR decomposition, such as [Table 11-4](#) Specific meaning, the parameters
 Justice can refer to code comments.

Table 11-4 RLS adaptive filtering algorithm

Operation
Effect
rls (x, d, w, lambda, delta)
Conventional RLS algorithm
sfttls (x, d, w, lambda, epsilon, training)
Steady and rapid transverse RLS algorithm
lrls (x, d, v, lambda, epsilon, training)
Ordinary lattice RLS algorithm
eflrls (x, d, v, lambda, epsilon, training)
Error feedback lattice RLS algorithm
qrlms (x, d, w, lambdaSqrt, training)
RLS algorithm based on QR decomposition

Test code:
 * Rls_test.cpp

12 Time-frequency analysis

12.1 windowed Fourier transform

Windowed Fourier transform (WFT) or short-time Fourier transform (STFT) due to its intuitive thinking to achieve

Simple, has become the most commonly used method of time-frequency analysis. WFT through the signal with the frequency atom as an inner product, the To the time-frequency domain signal from the time domain signal is transformed to achieve a localized time-frequency analysis features to overcome the Fourier transform

Global disadvantage.

Time-frequency atom is the basic window by the time shift function and modulation income, so the frequency of choice of window function when

Transform results influence is very large, the frequency is usually used when a good window aggregation functions, such as Gauss Hamming window or the like.

WFT transform the pros call format, see [Table 12-1](#) , which with the suffix "FFTW" function that the use FFTW

Library computing DFT, the use of these functions need to install FFTW .

Table 12-1 windowed Fourier transform

Operation
Effect
wft (sn, gn, mod)
Sn computing windowed Fourier transform coefficients
iwft (coefs, gn)
Computing coefs windowed Fourier inverse transform
wftFFTW (sn, gn, mod)
Sn computing windowed Fourier transform coefficients
iwftFFTW (coefs, gn)
Computing coefs windowed Fourier inverse transform

Test code:

* Wft_test.cpp

In order to be more intuitive, below a Matlab simulation results, as [in Figure 12-1](#) below.

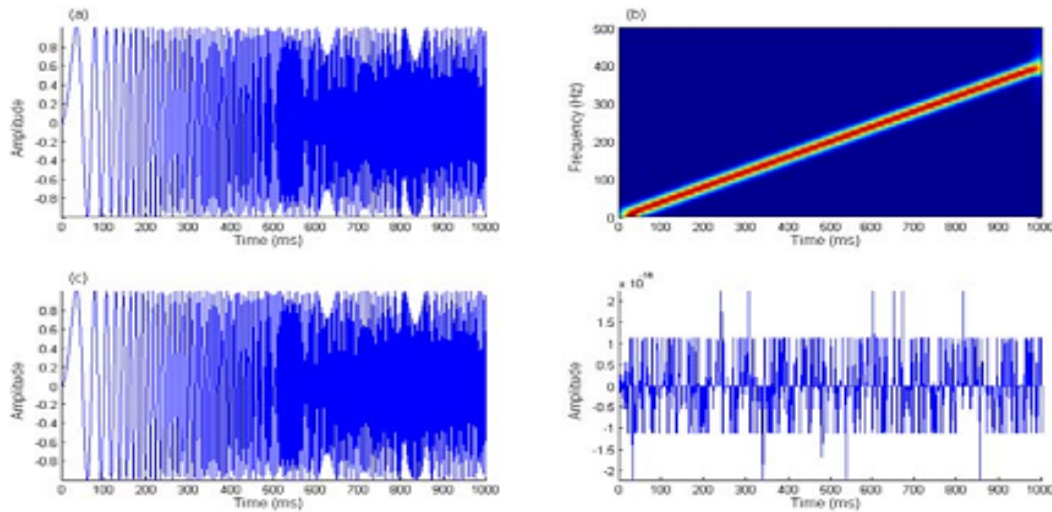


Figure 12 windowed Fourier transform

12.2 Discrete Gabor Transform

Discrete Gabor transform (DGT) is a frequency decomposition analysis by Gabor frame with integrated frame when the signal

And comprehensive, DGT can be seen WFT at a sampling frequency domain. Its redundancy possibly through oversampling rate

Adjustment, and can also aid the FFT algorithm, and therefore the relative computational efficiency is greatly WFT

Increase.

DGT is a frame operation, so the analysis window function and synthesis window function is generally not the same, but to

Dual window function is calculated by. "Dgt.h" header file provides a dual function as well as positive and negative variations of discrete Gabor

Calculating change, specifically to see [Table 12-2](#), which with the suffix "FFTW" **FFTW library functions that the use of computing DFT,**

Using these functions need to install FFTW .

Table 12-2 Discrete Gabor Transform

Operation
Effect
dual (gn, N, dM)
Calculation of the dual window function gn
dgt (sn, gn, N, dM, mode)
Sn computing discrete Gabor transform
idgt (coefs, gn, N, dM)
Discrete Gabor inverse transform of computing coefs
dualFFTW (gn, N, dM)
Calculation of the dual window function gn
dgFFTWt (sn, gn, N, dM, mode)

Sn computing discrete Gabor transform

idgtFFTW (coefs, gn, N, dM)

Discrete Gabor inverse transform of computing coefs

Test code:

* Dgt_test.cpp

In order to be more intuitive, below a Matlab simulation results, such as [Figure 12-2](#) And [Figure 12-3](#) below.

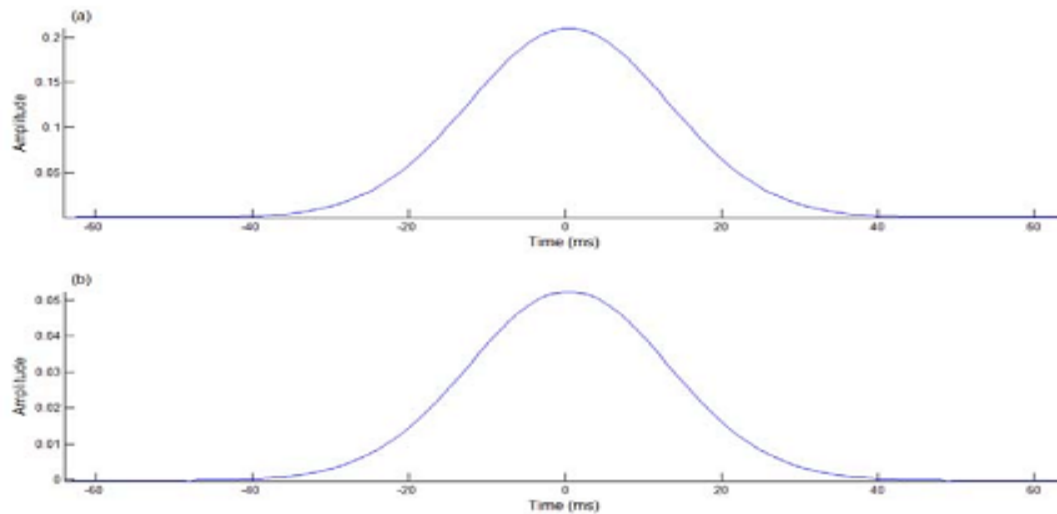


Figure 13 dual function

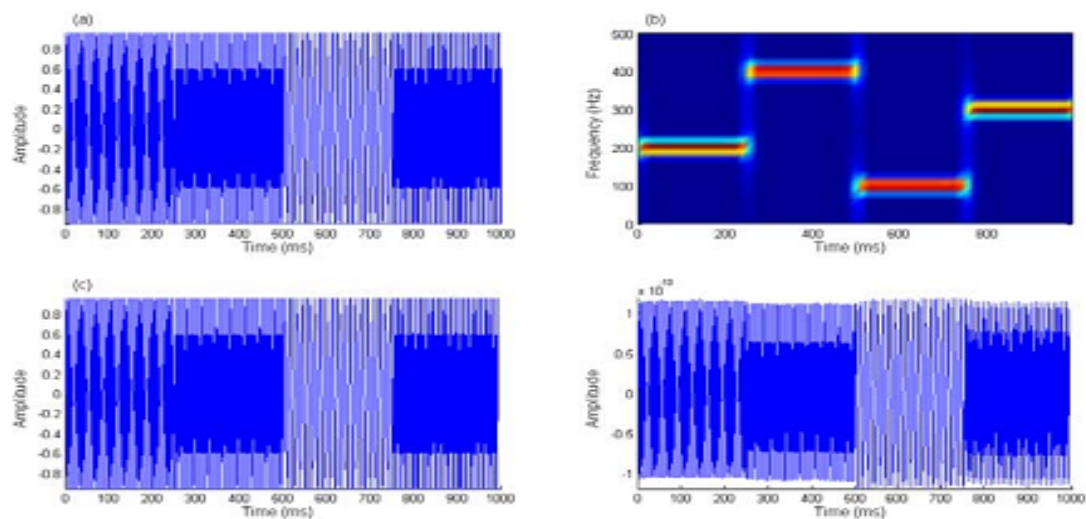


Figure 14 Discrete Gabor Transform

12.3 Wigner-Wille distribution

Windowed Fourier transform and wavelet transform time-frequency signals are used to do with the family of atomic inner product, so their time

Due to the time-frequency atomic frequency resolution is restricted. So people do not attempt to define a time-frequency resolution loss

Parameter distribution, i.e., such a distribution depends only on the signal itself, is not limited by a particular atom of the Group momentary frequency.

Wigner-Ville distribution (WVD) is the kind of time-frequency when there is no loss of resolution of the frequency distribution of the energy of people, it is

By the time the signal itself and the calculated frequency translation.

WVD has many advantages, such as the WVD is a time when the distribution, satisfy the conditions of time and frequency edge of the strip edge

Pieces, with instantaneous frequency and group delay characteristics, time shift and modulation characteristics, etc. But there is a fatal flaw WVD, both

Cross-term interference. SP ++ provides a real signal with a complex signal WVD, call the same format as [in Table 12-3](#) The

Show.

Table 12-3 Wigner-Wille distribution

Operation
Effect
wvd (sn)
Sn calculate the Wigner-Ville distribution

Test code:

* Wvd_test.cpp

In order to be more intuitive, below a Matlab simulation results, as [in Figure 12-4](#) and [Figure 12-5](#) Fig.

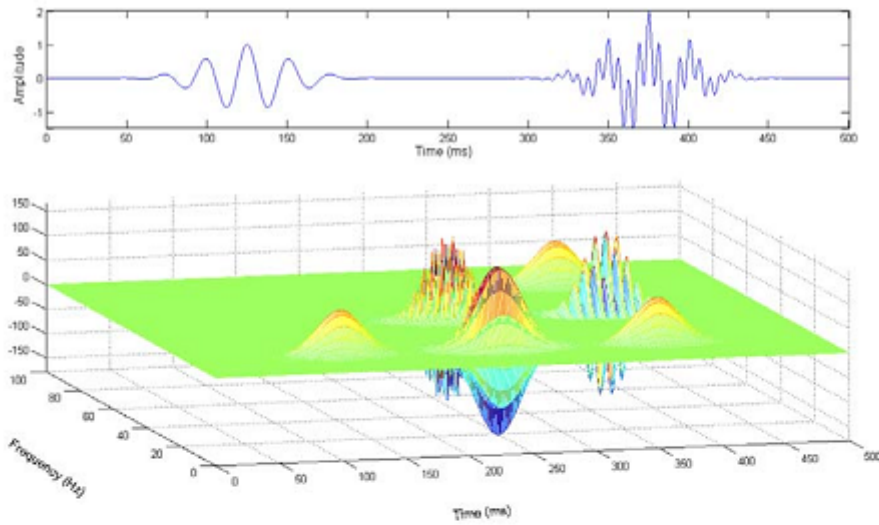


图 12-4 Wigner-Wille 分布

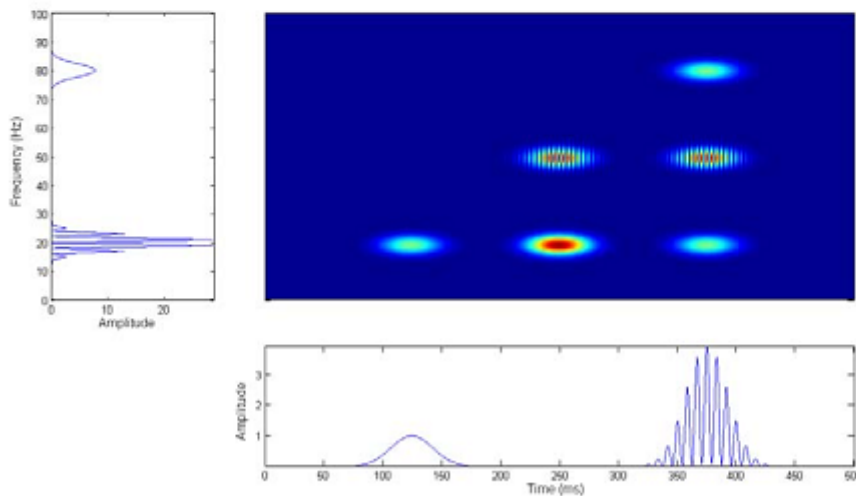


图 12-5 Wigner-Wille 分布

Figure 15 Wigner-Wille distribution Wigner-Wille distribution

13 Wavelet Transform

13.1 continuous wavelet transform

Wavelet analysis since the since the 1980s has been the rapid development, and in seismic signal processing, biomedical

Learn signal processing, image compression and coding and mechanical troubleshooting and other areas has made a very successful application.

Has now become the most common kind of signal analysis tools.

Continuous wavelet transform (CWT) has a high degree of redundancy, but it contains a wealth of information, often used for signal

The attribute extraction; while CWT has good stability, de-noising has also been widely used. SP ++ in mentioning

For a real continuous wavelet transform (in Mexico hat wavelet, for example) and complex continuous wavelet transform (with Morlet wavelet for

Example) pros transformation process, see call format [Table 13-1](#) , which with the suffix "FFTW" function that the use of

FFTW library computing DFT, the use of these functions need to install

FFTW . Accuracy can be reconstructed by the inverse transform scales

Parameters can be adjusted, the higher the accuracy, the greater the amount of calculation, the default parameters of the reconstruction error roughly $10e-4$

To $10e-8$ Between.

Table 13-1 continuous wavelet transform

Operation
Effect
CWT <Type> wt (wname)
Creating CWT class
wt.CWT <Type> ()
CWT class destructor
CWTFFTW <Type> wt (wname)
Creating CWTFFTW class
wt. CWTFFTW <Type> ()
CWTFFTW class destructor
wt.setScales (fs, fmin, fmax, dj)
Set the scale parameter
wt.cwrR (sn)
Sn calculate the real wavelet transform
wt.icwrR (coefs)
The actual calculation of the inverse wavelet transform coefs
wt.cwrC (sn)
Sn calculation of complex wavelet transform
wt.icwrC (coefs)
Coefs complex computing inverse wavelet transform

Test code:

* Cwt_test.cpp

In order to be more intuitive, below a Matlab simulation results, such as [Figure 13-1](#)

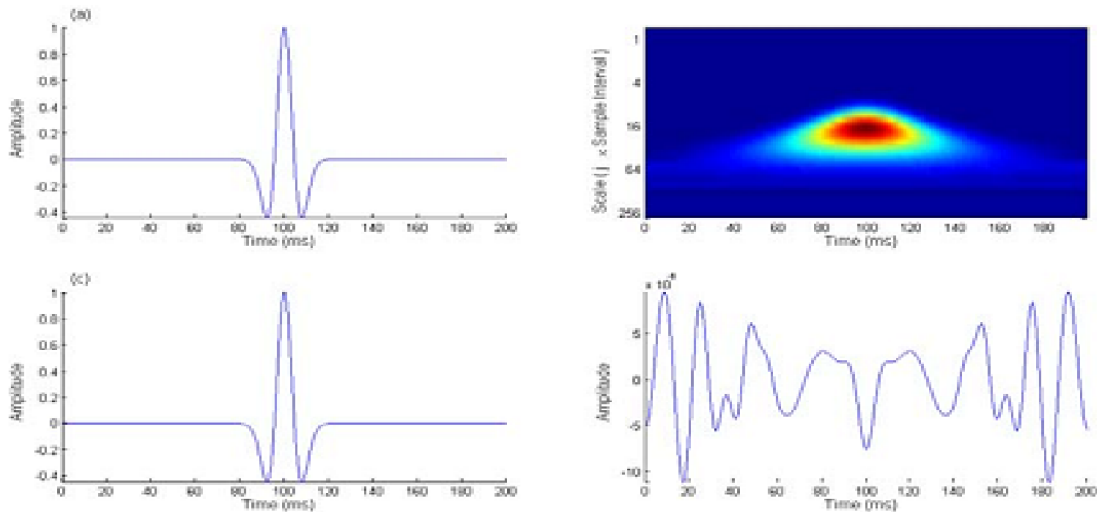


Figure 16 continuous wavelet transform

13.2 dyadic wavelet transform

Scale and translation parameters of the continuous wavelet transform values are continuous, so the high redundancy, if foot

Degree discrete parameters while maintaining continuous translation parameters, we obtain the dyadic wavelet transform. Dyadic wavelet transform was reduced

Less redundancy, but also to ensure the translation invariance wavelet transform, so it has been widely used.

SP ++ is provided based on quadratic spline dyadic wavelet transform, as [in Table 13-2](#) .Signal into

Line J-dyadic wavelet transform to obtain a level of detail coefficients J and J class smoothing factor, if wanted

j-level smoothing factor, the wavelet transform coefficients can Jj level reconstruction can be.

Table 13-2 dyadic wavelet transform

Operation
Effect
bwt (sn, J)
Sn conducted for J level dyadic wavelet transform
ibwt (coefs, j)
Coefs performed on stage j dyadic wavelet reconstruction

Test code:

* Bwt_test.cpp

In order to be more intuitive, below a Matlab simulation results, such as [Figure 13-2](#) And [Figure 13-3](#) below.

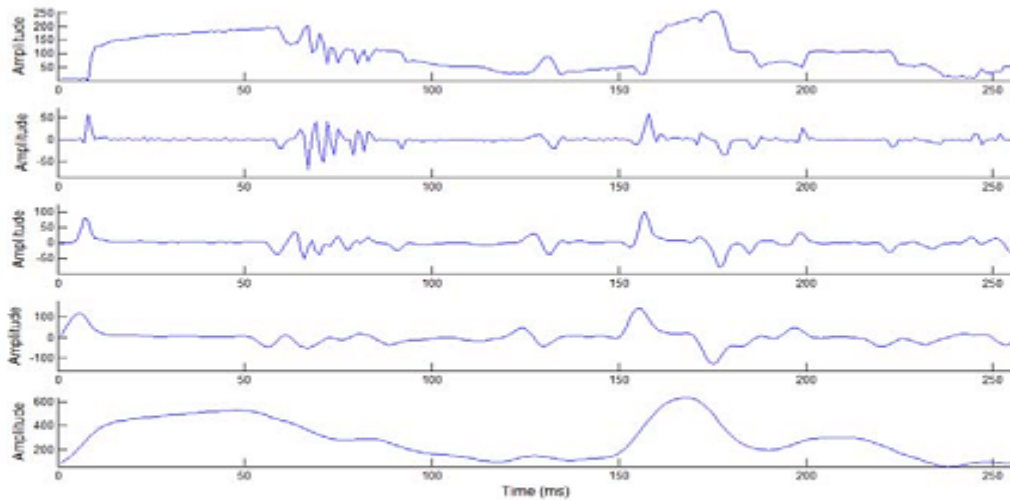


图 13-2 二进小波变换

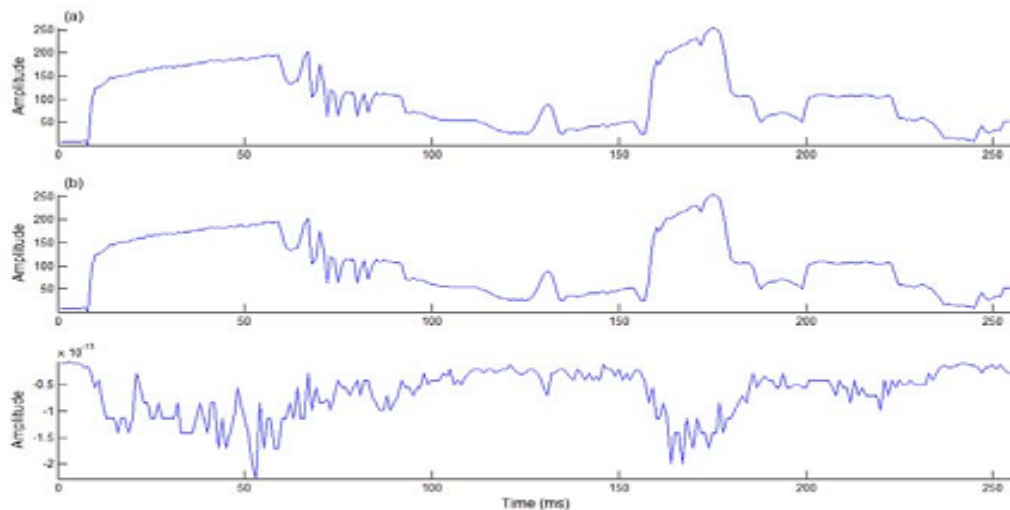


图 13-3 二进小波变换误差分析

Figure 17 dyadic wavelet transform, dyadic wavelet transform error analysis

13.3 DWT

If the scale parameter and the translation parameter underwent continuous wavelet transform discrete binary, namely discrete wavelet Transformation. Discrete wavelet transform signal is actually projected onto orthogonal wavelet basis, so there is no redundancy, and in

Mallat tower fast algorithm to calculate the efficiency is very high. Therefore, the discrete wavelet transform in image compression and encoding was

To a very successful application.

SP ++ provides a general algorithm based on discrete wavelet transform, the default filter set to "db4" wavelet user

Be modified according to their needs. In order to save storage space, the wavelet transform coefficients stored in a one-dimensional array

Can be accessed through an array of different indicators of detail coefficients and approximation coefficients, but these implementation details have been

Internal encapsulated in a class, the user need not be concerned. Specific operations, see Common [Table 13-3](#) .

Table 13-3 DWT

Operation
Effect
DWT <Type> wt (wname)
Creating DWT class
wt.DWT <Type> ()
DWT class destructor
wt.dwt (sn, J)
Class J sn carried on discrete wavelet transform
wt.idwt (coefs, j)
Coefs performed on stage j discrete wavelet reconstruction
wt.getApprox (coefs)
Get smooth discrete wavelet transform coefficients
wt. getDetial (coefs, j)
Get j-level detail coefficients of discrete wavelet transform
wt.setApprox (a, coefs)
Smoothed discrete wavelet transform coefficients
wt. setDetial (d, coefs, j)
Set discrete wavelet transform level detail coefficients j
Test code:
* Dwt_test.cpp

In order to be more intuitive, below a Matlab simulation results, such as [Figure 13-4](#) And [Figure 13-5](#).

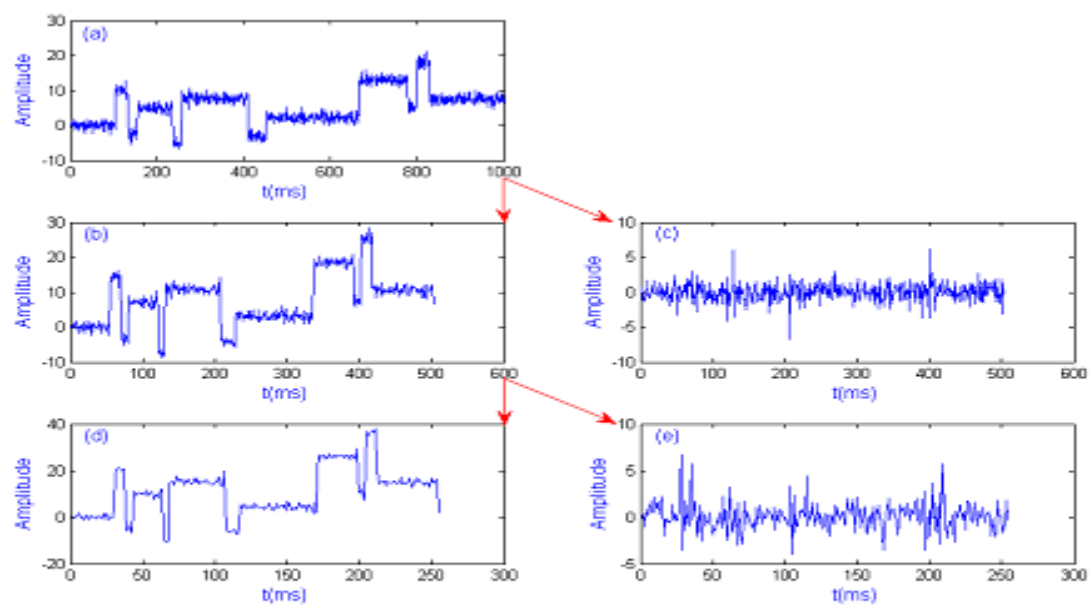


图 13-4 离散小波分解

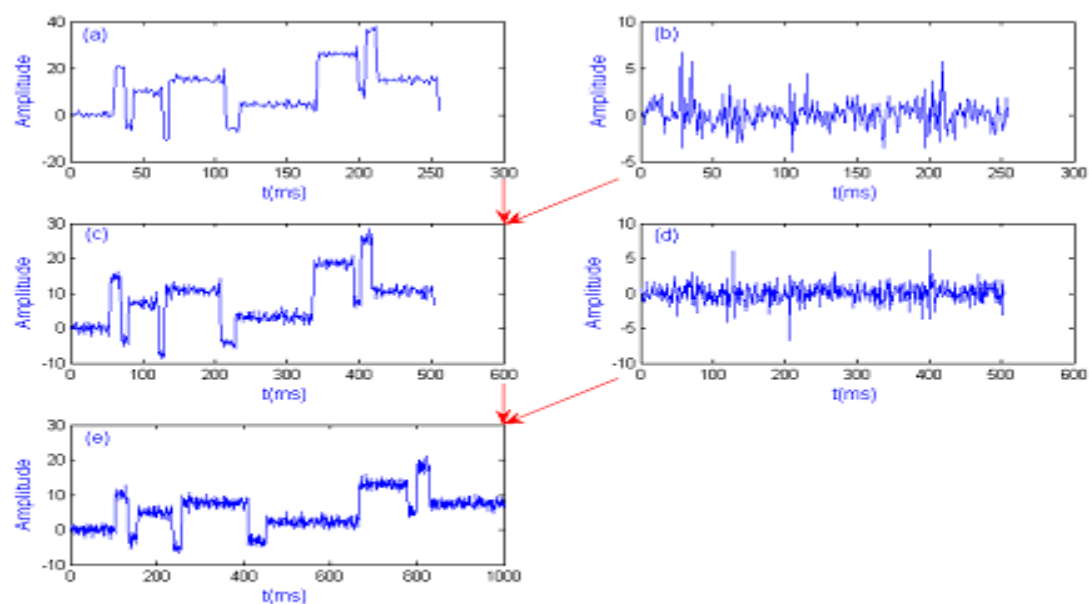


图 13-5 离散小波重构

Figure 18 DWT, discrete wavelet reconstruction

14 Find and sort

14.1 binary search tree

Binary Tree is a very important class of data structures, has a wide range of applications. SP ++ implementation of the binary tree

Common operations, including the preamble, in order and post-order traversal, insert, delete, search, maximum and minimum term items, etc.

All algorithms are implemented in the form of non-recursive, implementation of basic data structures used in the stack, see "stack.h"

File , see the specific function [Table 14-1](#) .

Table 14-1 binary search tree

Operation
Effect
BSTree bst
Create a binary search tree
bst.BSTree ()
Destructor binary search tree
bst.isEmpty ()
To determine whether the tree is empty
bst.makeEmpty ()
The tree blank
bst. preTraverse ()
Preorder traversal
bst. inTraverse ()
Preorder
bst. postTraverse ()
After preorder
bst. insert (x)
Insert elements
bst.remove (k)
Remove elements
bst.search (k)
Find elements
bst.maxItem ()
Find the largest item
bst.minItem ()
Find the smallest item

Test code:

* Bstree_test.cpp

14.2 balanced binary tree

Binary tree can be time complexity is reduced $\log_2 N$ operations

N , which is based on the premise of a more balanced binary tree

If unbalanced tree, in the worst case complexity remains for N , but also to waste extra storage space. One

Kinds of solutions is through the rotation technique enables balanced binary tree, the binary tree is called a balanced binary tree,

Or AVL tree. SP ++ to achieve a common operating AVL tree, including the traverse, search, insert, delete, etc.,

Realization of basic data structures used in the stack, see "stack.h" file , see the specific function [Table 14-2](#) .

Table 14-2 balanced binary tree

Operation
Effect
AVLTree avlt
Create AVL tree
avlt. AVLTree ()
Destructor AVL tree
avlt.isEmpty ()
To determine whether the tree is empty
avlt.makeEmpty ()
The tree blank
avlt. print (mode)
Traversal of the tree, including the preamble, in order, after the order
avlt. height ()
Calculate the height of the tree
avlt. search (k)
Find keyword k
avlt. insert (x)
Insert elements
avlt. remove (k, x)
Remove elements

Test code:

* Avltree_test.cpp

14.3 The basic sorting algorithm

Sorting is very important and there are a class of widely used algorithms, SP ++ provides a common class sorting count 6

France, the time complexity and space complexity analysis books refer to the data structure, not repeat them here. Specific letter

See number [Table 14-3](#) .

Table 14-3 sorting algorithm

Operation
Effect
bubbleSort (v, left, right)
Bubble sort algorithm
selectSort (v, left, right)
Select sorting algorithm
insertSort (v, left, right)
Insertion Sort Algorithm
quickSort (v, left, right)
Fast Sorting Algorithm
mergSort (v, left, right)
Merge sort algorithm
heapSort (v, left, right)
Heap sort algorithm

Test code:

* [Sort_test.cpp](#)

14.4 Huffman coding

Huffman coding is based on the statistical data of the law to achieve a lossless compression coding, you can

Huffman tree to achieve encoding and decoding process. See specific function [Table 14-4](#) . **Used in the process to achieve the smallest binary heap**

The basic data structure, see "binaryheap.h" file and node pointer classes designed to achieve by means

Compare the size of the needle points to the contents of the pointer size.

Table 14-4 Huffman coding

Operation
Effect
HuffmanTree ht
Create a Huffman tree
ht. HuffmanTree ()
Destructor Huffman tree
ht.code (codeArray, lenght)
Huffman coding
ht.decode (bits, length, decodeword)
Huffman decoding
ht.printCode (bits, length)

Print codeword

ht.printCodeTable ()

Print code table

Test code:

* Huffmancode_test.cpp

15 References

- [1] John H. Mathews, Kurtis D. Fink. Numerical Methods Using MATLAB, Fourier Edition. Beijing, Publishing House of Electronics Industry.
- [2] Shie Qian. Introduction to Time-Frequency Analysis and Wavelet Transforms. Beijing, China Machine Press, 2005.
- [3] Stephane Mallat. A Wavelet Tour of Signal Processing. Second Edition. Beijing, China Machine Press, 2006.
- [4] B. Widrow, SD Stearns. Adaptive Signal Processing. Beijing, China Machine Press, 2008.
- [5] Paulo SR Diniz. Adaptive Filtering Algorithms and Practical Implementation, Third Edition. Springer Science + Business Media, LLC, 2008.
- [6] Richard C. Aster, Brian Borchers, and Clifford Thurber. Parameter Estimation and Inverse Problems. 2004.
- [7] TM Cover, JA Thomas. Elements of Information Theory, Second Edition. Ruan Ji Shou, Zhang translation. Beijing, China Machine Press, 2008.
- [8] Ingrid Daubechies. Wavelets Ten Lectures. Jianping, Yang years translation. Beijing, National Defense Industry Press, 2004.
- [9] Gezhe school, Javert. MATLABR2007 wavelet analysis theory and implementation. Beijing, Electronic Industry Press, 2007.
- [10] Zou combined digital signal processing (the book). Xi'an, textbooks Keyin Xi'an Jiaotong University, 2006.
- [11] Alen V. Oppenheim, Alen S. Willsky and With S. Hamid Nawab. Signals and Systems, second Edition, Liu shu tang translation. Xi'an, Xi'an Jiaotong University Press, 2005.
- [12] Yaoze Qing, Su ice Applied functional analysis, Beijing, Science Press, 2007.
- [13] Guo Mao positive. Real Variable and functional analysis. Beijing, Peking University Press, 2007.
- [14] Wei fronts. Linear algebra and analytic geometry. Higher Education Press, 2006.
- [15] Deng Jianzhong, Liu trip Calculation method. Xi'an Jiaotong University, 2001.
- [16] Hu Maolin Matrix computing and applications. Beijing, Science Press, 2007.
- [17] High West whole, Ding Yumei digital signal processing, the third edition of Xi'an, Xi'an University of Electronic Science and Technology Press, 2008.

- [18] leaves of pay. Statistical Signal Processing, Anhui, China University of Technology Press, 2009.
- [19] The new solution can, Han Jian, Lin Union. Optimization method, revised edition. Tianjin, Tianjin University Press, 2008.
- [20] Mark Allen Weiss. Data Structures and Algorithm Analysis in C ++. Third Edition. Beijing, Posts & Telecom Press, 2007.
- [21] Stanley B. Lippman, Josée Lajoie, Barbara E. Moo. C ++ Primer, Fourth Edition. Addison Wesley Professional, 2005.
- [22] Scott Meyers. Effective C ++ 3th, Hou Jie translation. Beijing, Electronic Industry Press, 2006.
- [23] Nicolai M. Josuttis. C ++ Standard Library A Tutorial and Reference. Addison Wesley Longman, 1999.
- [24] Kernighan, Ritchie. The C Programming Language, 2nd. Baowen Xu, Li translation. Beijing, Electronic Industry Press, 2009.
- [25] Hemopurification. C ++ object-oriented programming. Beijing, Tsinghua University Press, 2006.
- [26] Yan Wei-Min, Wu Weimin data structure (C language version). Tsinghua University Press, 2006.
- [27] Yan Renkun data structure (object-oriented methods and C ++ language description), second edition. Beijing, Tsinghua out Press, 2007.
- [28] Lin Rui. High quality C / C ++ Programming Guide 2001. Article 15.2
- [29] Christopher Torrence, Gilbert P. Compo. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society, 1998,79 (1): 6178.
- [30] Shie Qian, Dapang Chen. Discrete Gabor Transform. IEEE Transactions on Signal Processing, 1993, 41 (7): 24292438.
- [31] Stephane Mallat A Theory for Multiresolution Signal Decomposition:. The Wavelet Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1989, 11 (7): 674693.
- [32] Stephane Mallat, Sifen Zhong. Characterization of Signals from Multiscale Edges.IEEE Transactions on Pattern Analysis and Machine Intelligence. 1992, 14 (7): 710 732.
- [33] Ingrid Daubechies.The Wavelet Transform, Time-Frequency Localization and Signal . Analysis IEEE Transactions on Information Theory 1990, 36 (5):. 9611005.
- [34] RG Stockwell, L. Mansinha, and RP Lowe Localization of the Complex

Spectrum: The S Transform, IEEE Transactions on Signal Processing, 1996, 44 (4): 998-1001.

[35] Pham DT. Fast algorithms for mutual information based independent component analysis IEEE Transactions on Signal Processing, 2004, 52 (10): 2690 ~ 2700.

15.3 Website

[36] FFTW h <http://www.fftw.org/>

[37] Blitz ++ <http://www.oonumerics.org/blitz/>

[38] GSL h <http://www.gnu.org/software/gsl/>

[39] Numerical Recipes <http://www.nr.com/>

[40] Template Numerical Toolkit ht [TP: //math.nist.gov/tnt/](http://math.nist.gov/tnt/)

[41] IT ++ h <http://sourceforge.net/apps/wordpress/itpp/>

[42] SGI STL h <http://www.sgi.com/tech/stl/>

[43] Boost <http://www.boost.org/>

16 Feeling

SP ++

16.1 whim

SP ++ earliest work began in 2008, but the motivation is to in 2009.

2008 coincided with the first half of 2009 full-year graduate and research and design a class, even in this period is not

Chat, so wanted to write a program in C ++. I have engaged in professional signal processing, signal processing, and therefore often put

Using algorithms written in C ++ again, but these programs for their own array, not a system.

When the second half of 2009 a project to achieve a very complex algorithm, Matlab program called many within

Department function, turn the C language when encountered some difficulties, because the algorithm frequency domain, so a variety of functions are

Based on the plural form. For this purpose Internet search some open source C and C ++ libraries, resulting in two startling discovery, a

Happily, while the other is disappointing: happy that there are thousands of online and on numerical signal

Library handle (I eventually chose GSL); disappointed that so many of the library is almost no Chinese people

Write (alas, in fact, the reason is very simple: in this impetuous and utilitarian academic atmosphere, who would want to do this

"Voluntary labor" thing?).

After a year of "tobacco Monk" life, I have had some thoughts: how students should spend it for three years?

I do not like Pavel Korchagin heroic spirit, but still think we should do something meaningful, so the initiation of

One idea: write a C++ library on signal processing. However, due to the signal processing algorithms to be used frequently

The matrix factorization, content solution of linear and nonlinear equations, interpolation and curve fitting, etc. numerical calculation, therefore

SP++ can actually be divided into two parts: the numerical computation and signal processing.

16.2 arduous

Thinking about doing easily said than done!

In time, energy, Money are limited, one person alone, I talk about how to complete such a work capacity

Easy. I have two problems: are not willing to do anything, since the hands do, we will certainly do everything I can do it well!

So from learning theory, algorithm design, coding, testing, validation, to final documentation writers, maintenance and more

The new step by step to do so.

The first is a matter of time, usually to busy with projects and master's thesis, so this work in the evenings and weekends only

Under the circumstances there is time to do, or to Lu said: "Time is like the water in the sea, trying to squeeze still there." Its

Ci is the energy problem, and now energy watered down, like the original pack the night the day before the exam the next day, and now a boil

Overnight days are slow, however. Finally, the problem is money, a student is difficult, more difficult for students to do in China, too poor

Chink, hundreds of dollars to buy a book should not hesitate a long time, really cup ah!

The most difficult way to transfer programs - Push Formula - Scheduler - Push to write correct and efficient formulas

Code, you must have a thorough understanding of theoretical algorithm. Tasted, plausible is not enough; in already

Some based on rote, Zhaomaohuahu, then there may be a laughing stock.

I firmly believe that "there is hard work is rewarded," After three years of hard work, SP++ version has been updated three times, currently

Algorithm has :(numerical part) vector and matrix basic operations, matrix decomposition, conventional, underdetermined, super

Set of linear equations, finding roots of nonlinear equations, interpolation and curve fitting, nonlinear unconstrained optimization algorithm

Etc; (signal processing section) Fourier transform, digital filter design, random signal processing, adaptive signal

Processing, time-frequency analysis, wavelet transform and so on.

A person's strength is limited, if only by a person to a perfect thing to do, then this person is

Genius! However, I do not believe that the presence of genius in this world, we are all born with talent, the difference is through China

Type of education and training, some people maintain the characteristics of talent (okay, I fall into this category), while others were taught

Bred genius. SP ++ so inevitably there are many deficiencies have been discovered and undiscovered, has been found

Strive to promptly corrected, undiscovered had to do to correct for early detection.

16.3 gains and losses

My people a bit silly, the fact that I did some stupid, not only without paying for almost all free time

Spent on this "voluntary labor", but he still lose six or seven hundred books.

"Gains and losses, gains and losses," I feel this is the greatest wisdom of our ancestors left us (at least "the

One "). Because SP ++ lost something and missed too much, this in no mood to talk, or talk about it proceeds,

To encourage others. Sometimes I want to pay so much in the end get what? One is grateful users. Second

Improve the horizontal? The third is to help me find a job, Fourth, alas, it is not it!

The most immediate rewards when the number of users of thanks, praise and suggestions, criticism. Thanks and praise is good, let

People have to get high; however suggestions and criticism is also essential that you correct mistakes, to go further.

As the level of it, would not say how much improvement can only be strengthened in some ways in the process. Level, or can

Surface force is too broad, and knowledgeable is an ability, people skills is an ability, but in different fields,

Various aspects of the right capabilities at different occasions just not the same weight. And one aspect of capacity strengthening, bound

On the other hand lead to a reduction, let's call it "the law of conservation capacity," it, to see what your options are up.

Our group undertake all project countries, and what the "863 Project", "Science and Technology major projects", etc.

The stuff is very iffy, all the emphasis on the concept of research, usually sounds cattle, but to find a job on cups

The! The only place it is probably Institute (colleges and universities are not small master drops), but unfortunately I have not and will not consider state-owned enterprises (this

People did a background, two did patron, go how mixed ah?). All types of companies are more focused on technology, concern is
 Some practical things to make, rather than theoretical inscrutable, because the company, after all, is to make money, or else
 However, would not be able to live up. Fortunately, from SP ++ developers learned a little technology to meet a written test and interview
 Okay, it seems "to pay returns" This sentence is justified.
 Read a book to learn to speak four state: Society - will learn - will be used - is used. Really have a
 Species found confidant feel great minds think alike ah (the audacity to call themselves what a hero, had fun, ha ha)!
 It is easy to learn, from primary school (even kindergarten) to university, through tempered, experienced and described
 Is over five hurdles warriors, the school also have learned, at least learn about it. Will learn can be difficult,
 Participants learn to learn like the relationship between fish and fishing, but unfortunately according to my observation, many people will learn. Will use is relatively easier,
 But there are good and bad points, used well, and must be appropriately can achieve a multiplier effect. The hardest is to the number
 Used, it is not easy to do that, to be recognized by others must be superior to others. Pull away, and quickly
 Come back!
 All in all, the biggest gain, or for my greatest encouragement than "being used", and this is my one
 Straight SP ++ will be in the end the power lies.

16.4 matter of opinion

SP ++ in the end is what level?

A: Compared to foreigners who wrote mature libraries far, compared to the online pass "stragglers" type of program you want

So a little bit higher. After SP ++ has its own framework, many algorithms in the project through the practice of inspection

Experience, and some algorithms based on expert advice and online friends recommended to do a lot of improvement, to express loyalty thanks!

But for some advice I was stubborn, do not modify (also said loyal thanks), such as some

Significant efficiency. There are two reasons: First, since it is clear that the compiler will help us optimize, so do not

Worry; Second, part of the defect is due to the development of efficiency and operational efficiency of compromise, speed is efficiency, development Speed is efficient, easy to use if you can sacrifice some operating efficiency is worth it.

SP ++ for me in the end what is the point?

A: profound, can not say is more important than a master's degree (of course afraid, have to rely on it after dinner),

At least meaningful than the books. A master's certificate in the end how much gold? Have not been on the do not know,

Had surprised - I did not much gold, pour a lot of moisture. Imagine how many people could be a master's thesis cited

With? After graduation, how many people had been able to continue to focus on the things he studied? Not self-evident, very few,

Because we are in a diploma with no relation to age level. So SP ++ over a million visits for me really

Real ratio goes diplomas more meaningful.

Is a sense of

Zhang, in February 2011 in Xi'an