

Politechnika Świętokrzyska

Laboratorium

Cyfrowe przetwarzanie sygnałów

Ćwiczenie 7

Transformata falkowa.

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studentów z funkcjami pozwalającymi na realizację dyskretnej transformaty falkowej.

dr inż Robert Kazała

Funkcje pakietu pywavelets do realizacji transformaty falkowej

Funkcje do realizacji prostej i odwrotnej transformaty falkowej dostępne są w bibliotece PyWavelets.

families()	Wypisanie listy dostępnych rodzin falek
wavelist(rodzina)	Wypisanie falek dostępnych w wybranej rodzinie.
dwt(x)	Dyskretna transformata falkowa.
dwt2()	Dwuwymiarowa dyskretna transformata falkowa.
wavedec()	Wielopoziomowa dyskretna transformata falkowa
wavedec2()	Dwuwymiarowa wielopoziomowa dyskretna transformata falkowa
idwt(x)	Odwrotna dyskretna transformata falkowa.
idwt2()	Odwrotna dwuwymiarowa dyskretna transformata falkowa.
waverec()	Wielopoziomowa rekonstrukcja sygnału.
waverec2()	Dwuwymiarowa wielopoziomowa rekonstrukcja sygnału.

Metody rozszerzania sygnału wejściowego

Wiele z funkcji modułu PyWavelets pozwala określić za pomocą parametru *mode* metody rozszerzania sygnału wejściowego:

- zpd - zero-padding – sygnał jest rozszerzany za pomocą zerowych próbek:

... 0 0 | x1 x2 ... xn | 0 0 ...

- cpd - constant-padding – próbki graniczne są powielane:

... x1 x1 | x1 x2 ... xn | xn xn ...

- sym - symmetric-padding – sygnał jest rozszerzany poprzez odbicie próbek:

... x2 x1 | x1 x2 ... xn | xn xn-1 ...

- ppd - periodic-padding – sygnał traktowany jest jako powtarzający się:

... xn-1 xn | x1 x2 ... xn | x1 x2 ...

- sp1 - smooth-padding – sygnał jest rozszerzany poprzez obliczenie pierwszej pochodnej na krawędziach

Przykład 1

Wyświetlenie dostępnych rodzin falek i falek z rodziny 'coif'.

```
import pywt
print pywt.families()
print pywt.wavelist('coif')
```

Przykład 2

Wyznaczenie prostej i odwrotnej transformaty falkowej dla przebiegu jednowymiarowego.

```
import pywt
x = [3, 7, 1, 1, -2, 5, 4, 6]
cA, cD = pywt.dwt(x, 'db2')

print pywt.idwt(cA, cD, 'db2')
```

Przykład 3

Wyznaczenie prostej i odwrotnej wielopoziomowej transformaty falkowej

```
import pywt
coeffs = pywt.wavedec([1,2,3,4,5,6,7,8], 'db1', level=2)
cA2, cD2, cD1 = coeffs
print cD1
print cD2
print cA2

print pywt.waverec(coeffs, 'db1')
```

Przykład 4

Wyznaczenie współczynników transformaty falkowej dla sygnału dwuwymiarowego.

```
import pywt, numpy
data = numpy.ones((4,4), dtype=numpy.float64)
coeffs = pywt.dwt2(data, 'haar')
cA, (cH, cV, cD) = coeffs
print cA
print cV
```

```
(cA, (cH, cV, cD)) <---->
-----
|      |      |
| cA(LL) | cH(LH) |
|      |      |
-----
|      |      |
| cV(HL) | cD(HH) |
|      |      |
-----
```

Przykład 5

Wyznaczenie prostej i odwrotnej transformaty falkowej dla sygnału dwuwymiarowego.

```
import pywt, numpy
data = numpy.array([[1,2], [3,4]], dtype=numpy.float64)
coeffs = pywt.dwt2(data, 'haar')
print pywt.idwt2(coeffs, 'haar')
```

Przykład 6

Wyznaczenie wielopoziomowej transformaty falkowej dla sygnału dwuwymiarowego.

```
import pywt, numpy
coeffs = pywt.wavedec2(numpy.ones((8,8)), 'db1', level=2)
cA2, (cH2, cV2, cD2), (cH1, cV1, cD1) = coeffs
print cA2
```

Przykład 7

Wyznaczenie prostej i odwrotnej wielopoziomowej transformaty falkowej dla sygnału dwuwymiarowego.

```
import pywt, numpy
coeffs = pywt.wavedec2(numpy.ones((4,4)), 'db1')
print "levels:", len(coeffs)-1
print pywt.waverec2(coeffs, 'db1')
```

Literatura

Zadania

1. Przeanalizować dokumentację funkcji z pakietu pywavelets.
2. Zapoznać się z falkami dostępnymi w module pywavelets przeglądając stronę <http://wavelets.pybytes.com/>.
3. Wykonać przykłady z instrukcji.
4. Wykonać i przeanalizować przykłady ze strony <http://www.pybytes.com/pywavelets/regression/index.html>.
5. Wyznaczyć transformaty falkowe dla własnych sygnałów jednowymiarowych i linii obrazu, dla różnych rodzin falek. Porównać uzyskane wyniki, analizując ilość współczynników o dużych wartościach.
6. Wyznaczyć transformaty falkowe dla własnych obrazów, z wykorzystaniem różnych rodzin falek. Porównać uzyskane wyniki, analizując ilość współczynników o dużych wartościach.
7. Dla uzyskanych transformat falkowych dokonać kwantyzacji współczynników zawierających detale z dokładnością 8, 6 i 4 bity, a następnie wykonać transformatę odwrotną. Przeanalizować uzyskane rezultaty.