Компьютерное зрение '2014

Who? Александр Вахитов

When? March 6, 2014

План лекции

Восстановление сигнала, не содержащего частот выше Ω_c

Построили X_s
Установили, что $X_s(j\Omega) = X(e^{j\Omega T})$
Как по $x[n]$ восстановить x_c?

$$H_r = rect(-\Omega_c, \Omega_c)$$

$$H_r(j\Omega)X_s(j\Omega) =$$

$$= H_r(j\Omega)\frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s)) = X_c(j\Omega)$$

(если $\Omega_s > 2\Omega_c$)
Обратным п.Ф. получаем $x_c(t)$ из $X_c(j\Omega)$.

Фильтр нижних частот

(рассказ о фильтре)

$$y(t) = \int_{-\infty}^{\infty} H_r(j\Omega) X_s(j\Omega) e^{j\Omega t} d\Omega$$

$$|H_r(j\Omega)| = T$$

$$h_r(t) = \frac{1}{2\pi} \int H_r(j\Omega) e^{j\Omega t} d\Omega = \frac{T}{2\pi jt} (e^{j\Omega_c t} - e^{j(-\Omega_c)t}) =$$

$$= \frac{\sin(\Omega_c t)}{\pi t}.$$

$$y(t) = \sum_n h_r(t-n) x[n]$$

Фильтры с бесконечной импульсной характеристикой

Задача

До сих пор:

- сигнал бесконечной длины.
- непрерывная импульсная характеристика фильтра

Как сделать реализуемый практически дискретный фильтр?
Импульсная инвариантность

$$h[n] = T_d h_c(nT_d)$$

Подход

- сделать фильтр для непрерывной частоты
- дискретизировать его

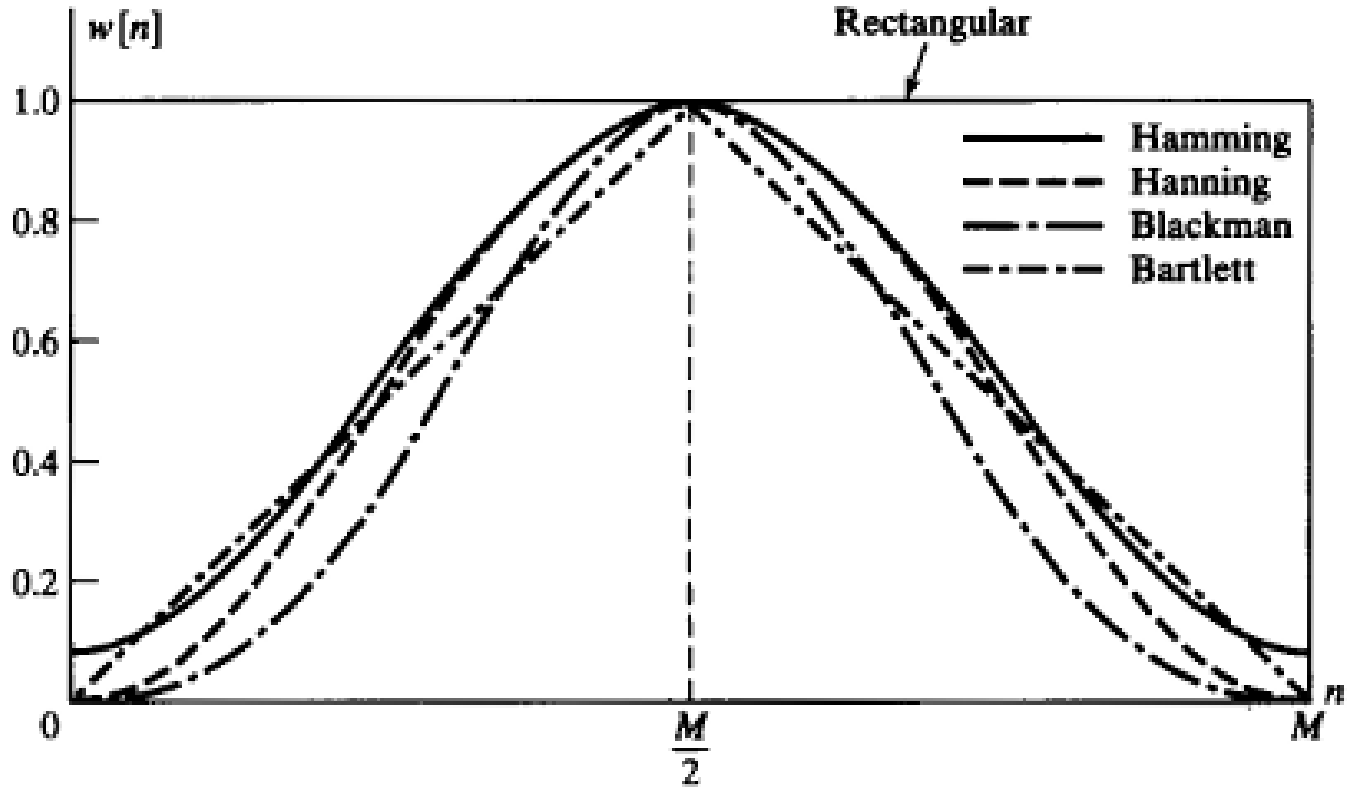
Избежать наложения спектров!
См. пример 7.2 из (Oppenheim et al. Discrete-time DSP)

Фильтры с конечной импульсной характеристикой

$$h[n] = h_d[n]w[n],$$

$$w[n] = \begin{cases} f(n), & 0 \leq n \leq M \\ 0, & \text{иначе.} \end{cases}$$

$f(n)$ - функция окна (см. ниже), M - ширина окна.



Оконные функции $w[n]$

Примеры фильтров с конечной ИХ

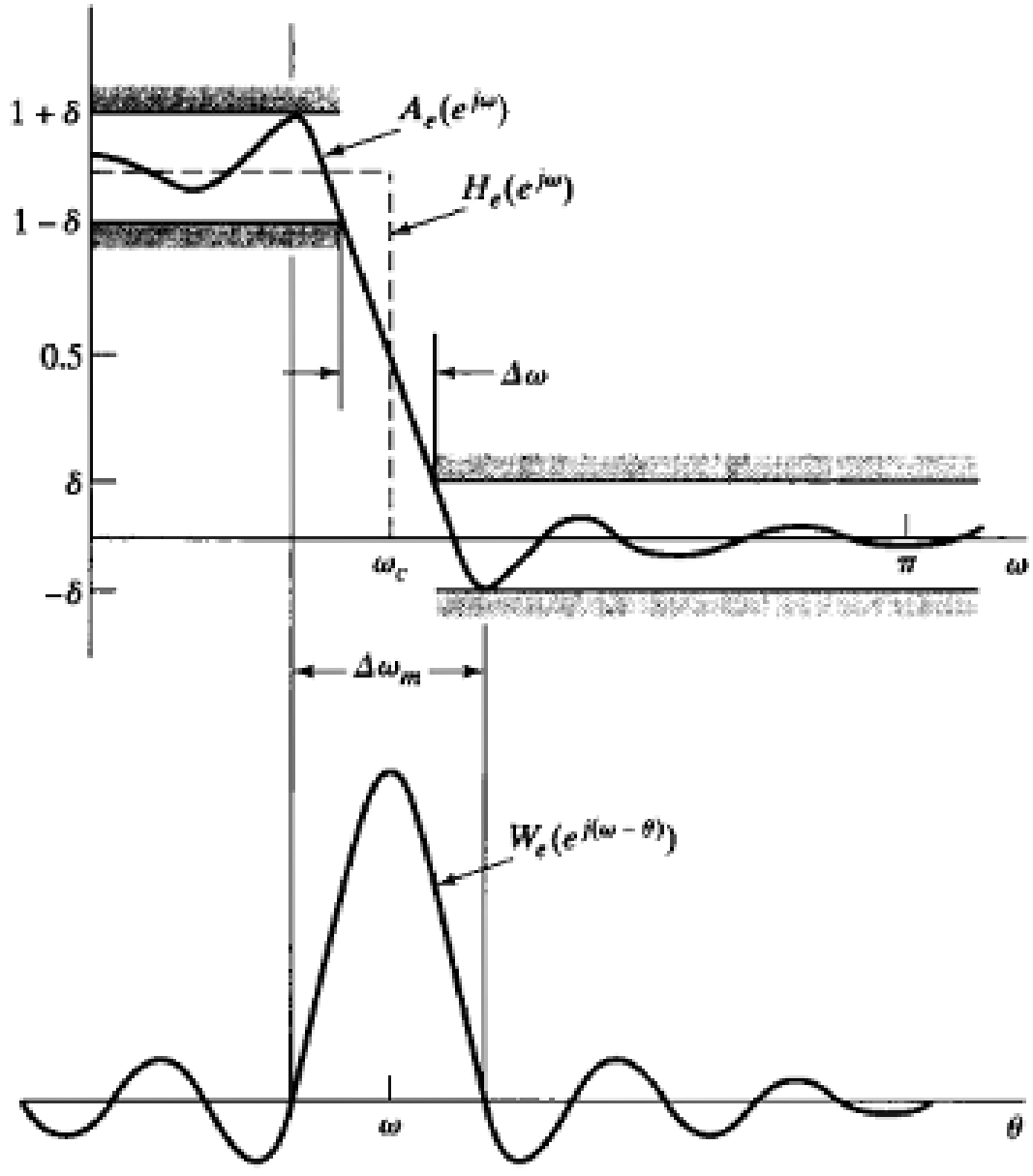
Hanning:

$$w[n] = \begin{cases} 0.5 - 0.5\cos(2\pi n/M), & 0 \le n \le M, \\ 0, & \text{иначе} \end{cases}$$

Идеальный фильтр нижних частот:

$$h[n] = \frac{\sin(\omega_c(n - M/2))}{\pi(n - M/2)} w[n].$$

Фильтр верхних частот?


$1+\delta$
$1-\delta$
0.5
δ
$-\delta$
$A_e(e^{j\omega})$
$H_e(e^{j\omega})$
$\Delta\omega$
ω_c
π
ω
$\Delta\omega_m$
$W_e(e^{j(\omega-\theta)})$
ω
θ

Дискретное преобразование Фурье

$$W_n = e^{j\frac{2\pi n}{N}k}$$

Фильтры для 2D

$$y[n,m] = \sum_{n_0=-\infty}^{\infty} \sum_{m_0=-\infty}^{\infty} x[n,m]h[n-n_0, m-m_0]$$

Обычно:

$$y[n,m] = (x[n,m] * h_x[n]) * h_y[m]$$

$$H(e^{j\omega_x}, e^{j\omega_y}) = H_x(e^{j\omega_x})H_y(e^{j\omega_y})$$

Python и задачи на дом

```
import cv2
import numpy as np

f = open('c:\\sampleprojects\\python\\logbig.txt', 'w');

for fnum in range(1,4):

    cap = cv2.VideoCapture('C:\\svn\\MATLAB\\trunk\\pulse\\w_118_' + str(

    while(1):
        ret ,img = cap.read()
        if ret == True:
            ressum = 0;
            for i in range(521, 720):
                for j in range(701, 950):
                    ressum = ressum + float(img[i,j,1]);
            result = ressum / (720-521+1) / (950-701+1);
            print result

            f.write(str(result));
            f.write(' ');


        else:
            break

f.close();
```

Python и задачи на дом

```
import cv2
import numpy as np

for fnum in range(1,4):
    pref = 'C:\\svn\\MATLAB\\trunk\\pulse\\w_118_';
    cap = cv2.VideoCapture(pref + str(fnum) + '.avi')
    fps = 19
    ret ,img = cap.read()
    frameSize = (1000, 750);
    print frameSize
    dst = cv2.bilateralFilter(img, 9, 15, 15);
    a = pref + '_c_' + str(fnum) + '.avi';
    writer = cv2.VideoWriter(a, -1, fps, frameSize, 1)
    writer.write(dst)
    while(1):
        ret ,img = cap.read()
        if ret == True:
            dst = cv2.bilateralFilter(img, 9, 15, 15);
            writer.write(dst);
        else:
            break

    writer.release();
```

Задачи

Установка Python
Заведение папки
Задачи по частотной фильтрации изображений
Задачи по (нелинейной) фильтрации
Статьи