**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY**

**FACULTY OF MECHANICAL ENGINEERING**

**DEPARTMENT OF MECHATRONIC ENGINEERING**

**-------o0o-------**



**COMPUTER VISION PROJECT**

**Topic: WIENER FILTER**

**Lecturer: PhD. Lê Thanh Hải**

|  |  |
| --- | --- |
| **Student’s name** | **ID** |
| Đặng Minh Quân | 1752445 |
| Lâm Hùng Minh | 1752343 |
| Trần Tiểu Bình | 1752010 |
| Cao Tri Thức | 1752533 |

Ho Chi Minh city, December 26th, 2020

1. **Digital image definition**

A digital image a[m,n] described in a 2D discrete space is derived from an analog image a(x,y) in a 2D continuous space through a sampling process that is frequently referred to as digitization. The effect of digitization is shown in Fig. 1. The 2D continuous image a(x,y) is divided into N rows and M columns. The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates [m,n] with {m = 0,1,2,. . .,M 1} and n = 0,1,2,. . .,N 1} is a[m,n]. We will consider the case of 2D, monochromatic, static images [1,2]: The image shown in Fig. 1 has been divided into N = 16 rows and M = 16 columns. The value assigned to every pixel is the average brightness in the pixel rounded to the nearest integer value. The process of representing the amplitude of the 2D signal at a given coordinate as an integer value with L different gray levels is usually referred to as amplitude quantization or simply quantization.

1. **Fourier transform**

The Fourier transform produces another representation of a signal, specifically a representation as a weighted sum of complex exponentials. Because of Euler’s formula [5]:

(1)

where j2 = -1, we can say that the Fourier transform produces a representation of a (2D) signal as a weighted sum of sines and cosines. The defining formulas for the forward

A picture containing diagram

Description automatically generated

*Fig. 1. Digitization of a continuous image. The pixel at coordinates [m=10, n=3] has the integer brightness value 110.*

Fourier and the inverse Fourier transforms are as follows. Given an image *a* and its Fourier transform *A*, then the forward transform goes from the spatial domain (either continuous or discrete) to the frequency domain which is always continuous [6,7].

*Forward:*

*A = F{a}* (2)

The inverse Fourier transform goes from the frequency domain back to the spatial domain.

*Inverse:*

*a = F-1{A}* (3)

The Fourier transform is a unique and invertible operation so that

*a = F-1{F{a}} and A = F{F1{A}}* (4)

The specific formulas for transforming back and forth between the spatial domain and the frequency domain are given as

*Forward:*

(5)

*Inverse:*

(6)

In programming, we usually use the discrete Fourier transformation than the continuous one because numerical computations can be conveniently performed with the Discrete  
Fourier Transform (DFT) and we can apply DFT to calculate convolution.

• Fourier Transform of a 2D *a-periodic* signal defined over a 2D discrete grid  
– The grid can be thought of as a 2D brush used for sampling the continuous  
signal with a given spatial resolution (Tx,Ty)

Text, letter

Description automatically generated

Where : normalized frequency

Text, letter

Description automatically generated

1. **Spectral sensitivity**

There are several ways to describe the sensitivity of the human visual system.  
The perceived intensity as a function of k, the spectral sensitivity, for the ‘‘typical observer’’ is shown in Fig. 2. The high sensitivity of silicon in the infra-red means that, for applications where a CCD (or other silicon-based) camera is to be used as a source of images for digital image processing and analysis, consideration should be given to using an IR blocking filter [8,9]. Sensors, such as those found in cameras and film, are  
not equally sensitive to all wavelengths of light.

Chart

Description automatically generated

*Fig. 2. Spectral characteristics of silicon, the sun, and the human visual system. UV = ultraviolet and IR = infra-red.*

1. **Summary of smoothing algorithms**

A variety of smoothing filters have been developed that are not linear. While they cannot, in general, be submitted to Fourier analysis, their properties and domains of application have been studied extensively. We have:

*4.1. Median filter*

A median filter is based upon moving a window over an image (as in a convolution) and computing the output pixel as the median value of the brightnesses within the input window. If the window is *J \* K* in size we can order the *J \* K* pixels in brightness value from smallest to largest. If *J \* K* is odd then the median will be the *J \* K + 1/2* entry in the list of ordered brightnesses.

* 1. *Kuwahara filter*

Edges play an important role in our perception of images as well as in the analysis of images. Although this filter can be implemented for a variety of different window  
shapes, the algorithm will be described for a square window of size *J = K = 4L + 1* where *L* is an integer. The window is partitioned into four regions as shown in Fig. 3.

Diagram

Description automatically generated

*Fig. 3. Four, square regions defined for the Kuwahara filter. In this example L = 1 and thus J = K = 5. Each region is [(J + 1)/2] × [(K + 1)/2].*

Table

Description automatically generated

A collage of a person

Description automatically generated with medium confidence

*Fig. 4. Illustration of various linear and non-linear smoothing filters. (a) Original, (b) uniform 5×5, (c) Gaussian (r = 25), (d) median 5×5, (e) Kuwahara 5×5.*

Table 1 summarizes the various properties of the smoothing algorithms. The filter size is assumed to be bounded by a rectangle of *J \* K* where, without loss of generality, *J ≥ K*. The image size is *N × N*.

Examples of the effect of various smoothing algorithms are shown in Fig. 4.

*4.3 Inverse filter*

Start from the generative model

*g(x,y) = h(x,y)\* f(x,y) + n(x,y)* ↔ *G(u,v) = H(u,v) F(u,v) + N(u,v)*

and for the moment ignore *n(x,y),* then an estimate of *f(x,y)* is obtained from

Restoration with an inverse filter:

Text, letter

Description automatically generatedShape

Description automatically generated with medium confidence

Apply fourier trick:

**Text

Description automatically generated**For example: Deblurring (Deconvolution) with inverse filter

A picture containing text

Description automatically generated

1. **Noise suppression**

The techniques available to suppress noise can be divided into those techniques that are based on temporal information and those that are based on spatial information. By temporal information we mean that a sequence of images *{ap[m,n]p = 1,2,. . .,P}* are available that contain exactly the same objects and that differ only in the sense of independent noise realizations. If this is the case and if the noise is additive, then simple averaging of the sequence [10,11]:

*Temporal averaging:*

(7)

1. **Results**

Within the class of linear filters, the optimal filter for restoration in the presence of noise is given by the *Wiener filter*. The word ‘‘optimal’’ is used here in the sense of minimum mean-square error (mse). Because the square root operation is monotonic increasing, the optimal filter also minimizes the root mean-square error (rms). The Wiener filter is characterized in the Fourier domain and for additive noise that is independent of the signal it is given by

(8)

where *Saa(u,v)* is the power spectral density of an ensemble of random images *{a[m,n]}* and *Snn(u,v)* is the power spectral density of the random noise. If we have a single image then *Saa(u,v) = |A(u,v)|2*.

If a two-dimensional signal *a(x,y)* has Fourier spectrum *A(u,v)* then

A collage of a person

Description automatically generated with medium confidence

*Fig. 5. Noise suppression using various filtering techniques. (a) Noisy image (SNR = 20 dB), (b) Wiener filter, (c) Gauss filter (σ = 1.0), rms = 25.7, rms = 20.2, rms = 21.1, (d) Kuwahara (5×5), (e) median filter (3×3), (f) Morph. smoothing (3×3), rms = 22.4, rms = 22.6, rms = 26.2*

The Wiener filter was constructed directly from Eq. (8), because the image spectrum and the noise spectrum were known. The parameters for the other filters were determined choosing that value (either or window size) that led to the minimum rms.

For example: Focus deblurring with a Wiener filter

A picture containing text

Description automatically generated

Graphical user interface

Description automatically generated

1. **Conclusion**

The root mean-square errors (rms) associated with the various filters are shown in Fig. 5. For this specific comparison, the Wiener filter generates a lower error than any of the other procedures that are examined here. The two linear procedures, Wiener filtering and Gaussian filtering, performed slightly better than the three non-linear alternatives.

1. **Implementation (using MATLAB program) and explanation**

**8.1 Inverse filter**

% ex = inverseFilter(y,h,gamma);

% Generalized inverse filtering using threshold gamma:

% inv\_g(H) = gamma\*abs(fft(h))/fft(h), if abs(fft(h)) <= 1/gamma

% inv\_g(H) = inv(H), otherwise

N = size(y,1);

Yf = fft2(y);

Hf = fft2(h,N,N);

% handle singular case (zero case)

sHf = Hf.\*(abs(Hf)>0)+1/gamma\*(abs(Hf)==0);

iHf = 1./sHf;

%lengthzero = length(abs(Hf)==0)

% invert Hf using threshold gamma

iHf = iHf.\*(abs(Hf)\*gamma>1)+gamma\*abs(sHf).\*iHf.\*(abs(sHf)\*gamma<=1);

ex = real(ifft2(iHf.\*Yf));

**8.2 Wiener filter**

% ex = wienerFilter(y,h,sigma,gamma,alpha);

% Generalized Wiener filter using parameter alpha. When

% alpha = 1, it is the Wiener filter. It is also called

% Regularized inverse filter.

N = size(y,1);

Yf = fft2(y);

Hf = fft2(h,N,N);

Pyf = abs(Yf).^2/N^2;

% direct implementation of the regularized inverse filter,

% when alpha = 1, it is the Wiener filter

% Gf = conj(Hf).\*Pxf./(abs(Hf.^2).\*Pxf+alpha\*sigma^2);

% Since we don't know Pxf, the following

% handle singular case (zero case)

sHf = Hf.\*(abs(Hf)>0)+1/gamma\*(abs(Hf)==0);

iHf = 1./sHf;

iHf = iHf.\*(abs(Hf)\*gamma>1)+gamma\*abs(sHf).\*iHf.\*(abs(sHf)\*gamma<=1);

Pyf = Pyf.\*(Pyf>sigma^2)+sigma^2\*(Pyf<=sigma^2);

Gf = iHf.\*(Pyf-sigma^2)./(Pyf-(1-alpha)\*sigma^2);

% max(max(abs(Gf).^2)) % should be equal to gamma^2

% Restorated image without denoising

eXf = Gf.\*Yf;

ex = real(ifft2(eXf));