Introduction to Image Processing

Lecture 4
22nd Oct. 2015

Guillaume Lemaître
guillaume.lemaitre@udg.edu

Université de Bourgogne

Image Restoration

Image restoration vs. image enhancement

Image enhancement

- Largely a subjective process
- Prior knowledge about the degradation is not a must (sometimes no degradation is involved)
- Procedures are heuristic and take advantage of the psychophysical aspects of human visual system

Image restoration

- More an objective process
- Images are degraded
- Try to recover the images by using the knowledge/model about the degradation

Image degradation

Degradation types

- Additive noise
  - Spatial domain restoration (denoising) techniques are preferred
- Image blur
  - Frequency domain techniques are preferred

Image degradation

Degradation model

$$g(x,y) = h(x,y)^{*} f(x,y) + \eta(x,y)$$

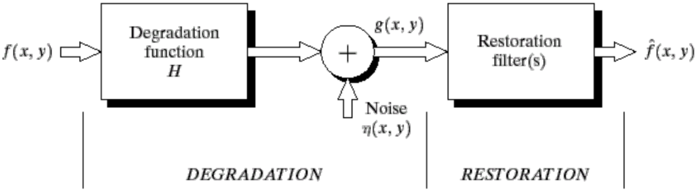
- $f(x,y)$ is the input image free from any degradation
- $g(x,y)$ is the degraded image
- $h(x,y)$ is the degradation function
- $\eta(x,y)$ is additive noise
- $*$ is the convolution operator
- In FD : $G(u,v) = H(u,v)F(u,v) + N(u,v)$

Different cases

$$g(x,y) = f(x,y) + \eta(x,y)$$
$$g(x,y) = f(x,y)^{*} h(x,y)$$
$$g(x,y) = f(x,y)^{*} h(x,y) + \eta(x,y)$$

Image Degradation / Restoration

Degradation/restoration model



Degradation Due to Noise
Noise models

- White noise
- Gaussian noise
- Rayleigh noise
- Erlang (gamma) noise
- Exponential noise
- Uniform noise
- Impulse (salt-and-pepper) noise

Degradation Due to noise
Noise models

White noise

- Autocorrelation function is an impulse function multiplied by a constant

$$a(x,y) = \sum_{s=0}^{N-1} \sum_{t=0}^{M-1} \eta(s,t).\eta(s-x,t-y) = N_0 \delta(x,y)$$

- There is no correlation between any two pixels in the noise image
- There is no way to predict the next noise value
- The spectrum of the autocorrelation function is a constant (White)

Degradation Due to noise
Noise models

Gaussian noise

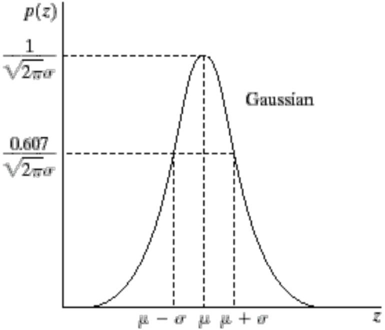
- Noise can be classified according the distribution of the pixel values of the noise image or its normalized histogram
- Gaussian noise is characterized by two parameters, mean (μ) and variance (σ^2)

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

- 68 % of z values fall in the range $[(\mu - \sigma), (\mu + \sigma)]$
- 95 % of z values fall in the range $[(\mu - 2\sigma), (\mu + 2\sigma)]$

Degradation Due to noise
Noise models

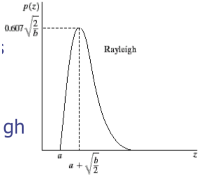
Gaussian noise



Degradation Due to noise
Noise models

Rayleigh noise

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$



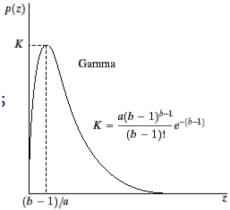
- The mean and variance of this density are given by $\mu = a + \sqrt{(\pi b)/4}$, $\sigma^2 = \frac{b(\pi - 4)}{4}$
- a and b can obtained through mean and variance

Degradation Due to noise
Noise models

Gamma (Erlang) noise

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

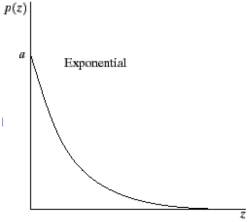
- The mean and variance of this density are given by $\mu = b/a$, $\sigma^2 = b/a^2$
- a and b can obtained through mean and variance



Degradation Due to noise
Noise models

Exponential noise

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$



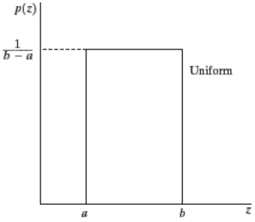
- The mean and variance of this density are given by $\mu = 1/a$, $\sigma^2 = 1/a^2$
- Special case of Erlang *pdf* with $b = 1$

Degradation Due to noise
Noise models

Uniform noise

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \geq b \\ 0 & \text{Otherwie} \end{cases}$$

- The mean and variance of this density are given by $\mu = (a+b)/2$, $\sigma^2 = (b-a)^2/12$

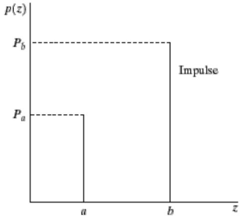


Degradation Due to noise
Noise models

Impulse (salt-and-pepper) noise

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

- If either P_a or P_b is zero, the impulse noise is called unipolar
- a and b usually are extreme values because impulse corruption is usually large compared with the strength of the image signal
- It is the only type of noise that can be distinguished from others visually

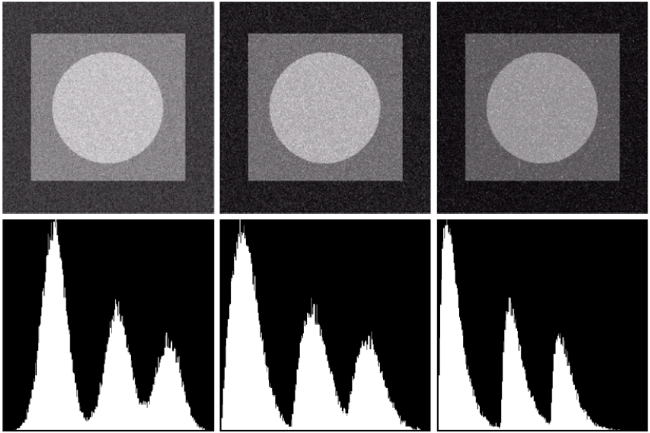


Degradation Due to Noise
Noise models

Noise in practice

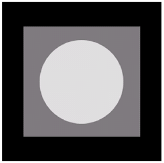
- **Gaussian noise:** electronic circuit noise and sensor noise due to poor illumination or high temperature
- **Rayleigh noise:** range imaging
- **Erlang noise:** laser imaging
- **Impulse noise:** quick transients take place during imaging
- **Uniform noise:** used in simulations

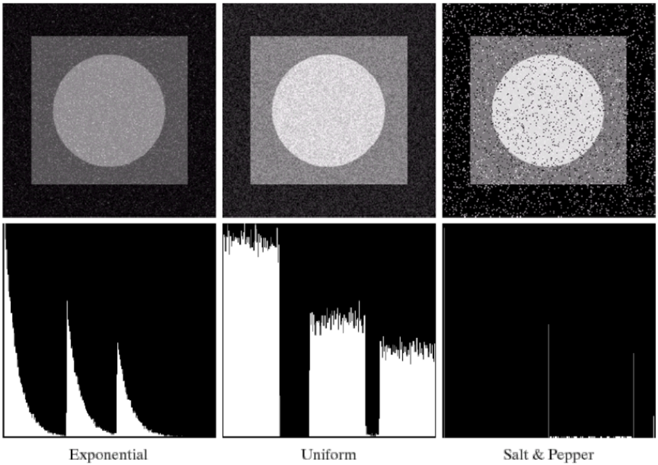
Degradation Due to Noise
Noise example



Gaussian Rayleigh Gamma

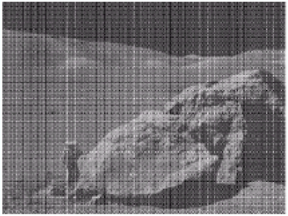
Degradation Due to Noise
Noise example

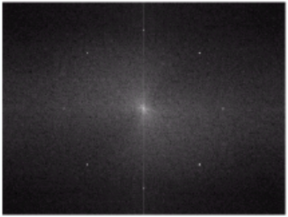




Degradation Due to Noise
Periodic Noise

- Arises typically from electrical or electromechanical interference during image acquisition
- It can be observed by visual inspection both in the spatial and frequency domain





Degradation Due to Noise
Estimation of noise parameters

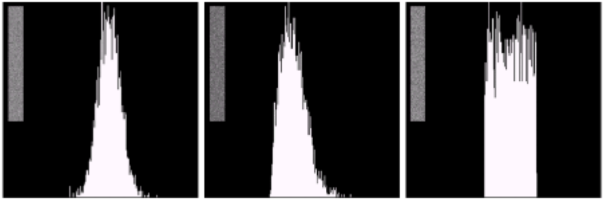
Periodic noise

- Parameters can be estimated by inspection of the spectrum

Noise *pdf*

- Sensor specifications
- Capture a set of images of plain environment, if imaging sensors are available
- If only noisy images are available, *pdf* parameters can be estimated from small patches of constant regions of the noisy image

Degradation Due to Noise
Periodic Noise



Degradation Due to Noise
Estimation of noise parameters

- Commonly, only mean and variance need to be estimated
- Considering a sub-image with plain scene S,

$$\hat{\mu} = \frac{1}{N_s} \sum_{(x_i, y_i) \in S} z(x_i, y_i)$$

$$\hat{\sigma_1^2} = \frac{1}{N_s} \sum_{(x_i, y_i) \in S} (z(x_i, y_i) - \hat{\mu})^2$$

De-Noising

Mean filter

$g(x, y)$ is the corrupted image and $S_{x,y}$ is the mask

- Arithmetic mean filter

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{x,y}} g(s, t)$$

- Geometric mean filter $\rightarrow$ Tends to preserve more details

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$$

- Harmonic mean filter $\rightarrow$ Works well for salt noise but fails for pepper noise

$$\hat{f}(x, y) = \frac{mn}{\sum_{(x_i, y_i) \in S_{xy}} \frac{1}{g(s,t)}}$$

De-Noising

Mean filter

- Contraharmonic mean filter

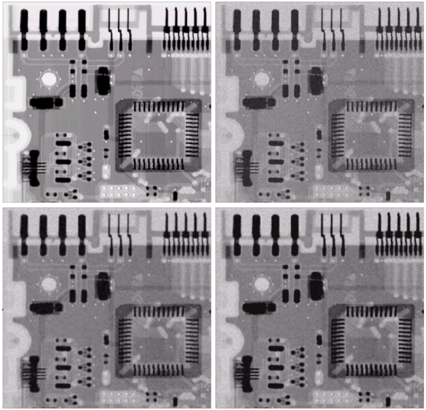
$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

  - Q is order of the filter
  - $Q > 0 \rightarrow$ pepper noise
  - $Q < 0 \rightarrow$ salt noise
  - $Q = 0 \rightarrow$ arithmetic mean filter
  - $Q = -1 \rightarrow$ harmonic mean filter

De-Noising

Mean filter

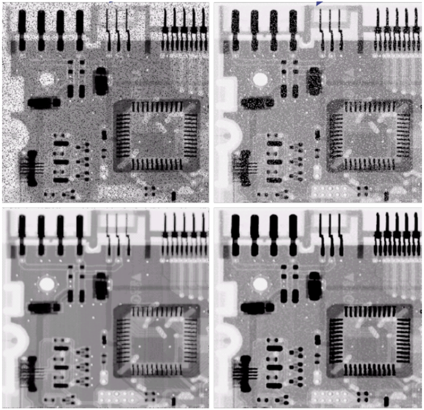
From right to left and up to bottom : original image, corrupted with Gaussian noise, mean filtering, and geometric mean filtering



De-Noising

Mean filter

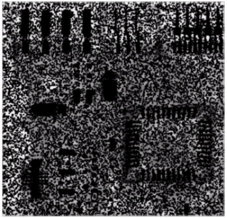
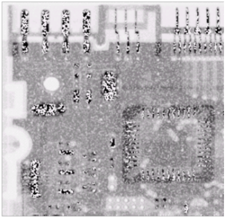
From right to left and up to bottom : corrupted with pepper noise, corrupted with salt noise, Contraharmonic filter $Q = 1.5$, Contraharmonic filter $Q = -1.5$



De-Noising

Mean filter

- Selecting the wrong sign for Contraharmonic filter, from right to left: Contraharmonic filter $Q = -1.5$, Contraharmonic filter $Q = +1.5$

De-Noising

Median filter

$$\hat{f}(x,y) = median_{(s,t)\in S_{xy}} g(s,t)$$

- Median represents the 50^{th} percentile of a ranked set of numbers

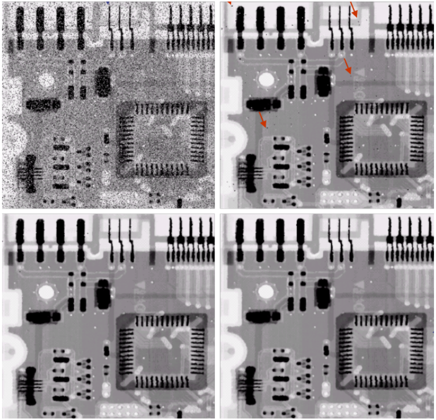
Max and min filter

- Max filter uses the 100^{th} percentile of a ranked set of numbers
  - Good for removing pepper noise
- Min filter uses the 1^{th} percentile of a ranked set of numbers
  - Good for removing salt noise

De-Noising

Median filter example

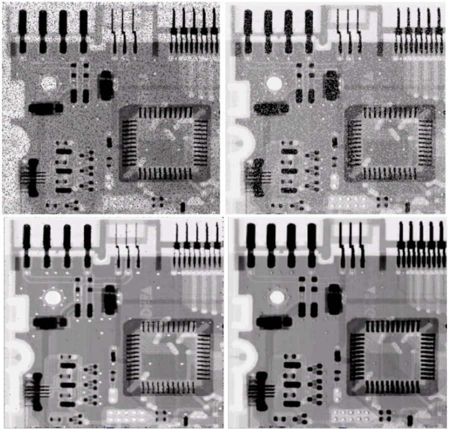
Corrupted image with salt-and-pepper, and one, two and third pass with median filter, respectively



De-Noising

Max and Min filter example

First row: Corrupted images with pepper and salt noise, respectively. Second row: the results of Max and Min filtering, respectively



De-Noising

Midpoint filter

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}} g(s,t) + \min_{(s,t)\in S_{xy}} g(s,t)\right]$$

- Works best for noise with symmetric *pdf* like Gaussian or uniform noise

De-Noising

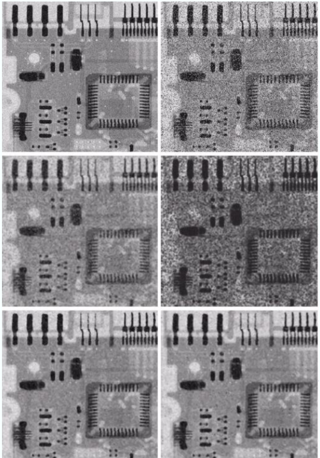
Alpha trimmed mean filter

$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t)\in S_{xy}} g_r(s, t)$$

- Takes the mean value of the pixels enclosed by an $m \times n$ mask after deleting the pixels with the $d/2$ lowest and highest gray-level values
- $g_r(s, t)$ represent the remaining $mn - d$ pixels
- Useful while dealing with multiple noise (e.g salt-and-pepper and Gaussian)

De-Noising - example

First column: Corrupted by uniform noise, mean (5×5), and median (5×5), respectively. Second column: Corrupted by uniform and salt-and-pepper noise, geomean(5×5), and alpha-trimmed mean (5×5)



De-Noising
Adaptive Filter

- Filters whose behavior changes based on statistical characteristics of the image.
- Two types of adaptive filter:
  - Adaptive, local noise reduction filter
  - Adaptive median filter

Adaptive, local noise reduction filter

Parameters:

- mean(μ), average gray level
- variance(σ^2), average contrast

Measurements:

- noisy image $g(x, y)$, the variance of noise σ_η^2, local mean m_L in S_{xy}, local variance σ_L^2

De-Noising
Adaptive Filter

Adaptive, local noise reduction filter

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_L^2}[g(x,y) - m_L]$$

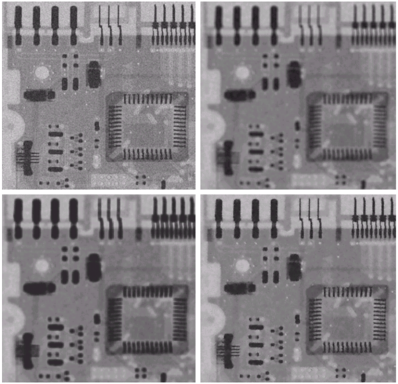
Conditions:

- $\sigma_\eta^2 = 0 \rightarrow$ Zero noise case
  - Return the value of g(x,y)
- $\sigma_L^2 > \sigma_\eta^2$
  - Possible edge and should be preserved
  - Return the value close to g(x,y)
- $\sigma_L^2 = \sigma_\eta^2$
  - When the local area has similar properties with the overall image.
  - Return arithmetic mean value of the pixels in S_{xy}

De-Noising
Adaptive Filter

Adaptive, local noise reduction filter

LRUB: Image corrupted by additive Gaussian nois $(\mu, \sigma^2) = (0, 1000)$, arithmetic mean filter, geometric mean filter, and adaptive noise reduction. Filter size $= 7 \times 7$



De-Noising
Adaptive Filter

Adaptive median filter

- Effective for removing salt and pepper noise
- The density of the impulse noise can not be too high for median filter compare to adaptive median filter
- Notations:
  - Z_{min}, minimum gray value in S_{xy}
  - Z_{max}, maximum gray value in S_{xy}
  - Z_{med}, median gray value in S_{xy}
  - Z_{xy}, gray value of the image at (x, y)
  - S_{max}, maximum allowed size of S_{xy}

De-Noising
Adaptive Filter

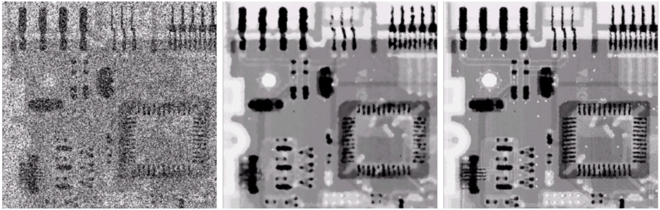
Adaptive median filter

- Two levels of operations:
- Level A, Used to test whether Z_{med} is part of s-and-p noise, if yes, window size is increased
  - $A_1 = Z_{med} - Z_{min}$
  - $A_2 = Z_{med} - Z_{max}$
  - if $A_1 > 0$ & $A_2 < 0 \rightarrow$ Level B
  - else $\rightarrow$ increase S_{xy} size by 2
  - if window size $\leq S_{max}$ repeat Level A else return Z_{xy}
- Level B, Used to test whether Z_{xy} is part of s-and-p noise, if yes, apply regular median filtering
  - $B_1 = Z_{xy} - Z_{min}$
  - $B_2 = Z_{xy} - Z_{max}$
  - if $B_1 > 0$ & $B_2 < 0$, return Z_{xy}
  - else return Z_{med}

De-Noising
Adaptive Filter

Adaptive median filter

Image corrupted with s-and-p noise, median filter (7×7), and adaptive median filtering ($S_{max} = 7$), respectively



Periodic noise reduction
Frequency domain Filtering

- Lowpass and highpass filters for image enhancement
- Bandreject, bandpass and notch filters for periodic noise reduction or removal

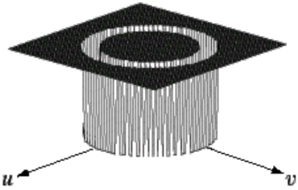
Bandreject filters

- Bandreject filters remove or attenuate a band of frequencies about the origin of the Fourier transform
- Ideal, Butterworth, Gaussian bandreject filters

Periodic noise reduction

Bandreject Ideal filters

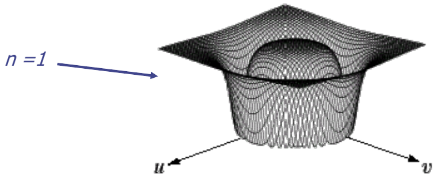
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) < D_0 - \frac{W}{2} \\ 0 & \text{if } D_0 - \frac{W}{2} \leq D(u,v) \leq D_0 + \frac{W}{2} \\ 1 & \text{if } D(u,v) > D_0 + \frac{W}{2} \end{cases}$$



Periodic noise reduction

Bandreject Butterworth filters

$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)W}{D^2(u,v) - D_0^2}\right]^{2n}}$$



Periodic noise reduction

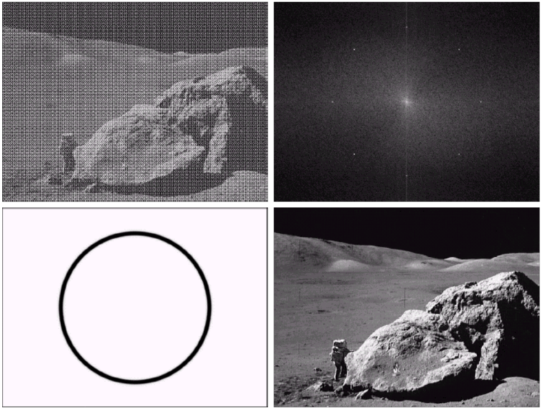
Bandreject Guassian filters

$$H(u,v) = 1 - e^{\frac{-1}{2}\left[\frac{D^2(u,v) - D_0^2}{D(u,v)W}\right]}$$

Periodic noise reduction

Bandreject filter

Corrupted image by sinusoidal noise and filtered image by Butterworth bandrejected filter



Periodic noise reduction

Bandpass filter

Performs opposite of bandreject filter

$$H_{bp}(u, v) = 1 - H_{br}(u, v)$$

Notch filter

- Notch filter ejects frequencies in predefined neighborhoods about a center frequency
- It appears in symmetric pairs about the origin because the Fourier transform of a real valued image is symmetric

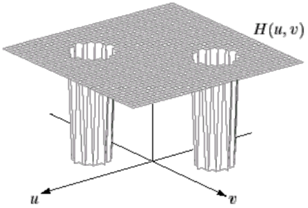
Periodic noise reduction

Ideal notch filter

$$H(u,v) = \begin{cases} 0 & \text{if } D_1(u,v) \le D_0 \text{or } D_2(u,v) \le D_0 \\ 1 & \text{otherwise} \end{cases}$$

$$D_1(u,v) = \left[(u - M/2 - u_0)^2 + (u - N/2 - v_0)^2\right]^{1/2}$$

$$D_2(u,v) = \left[(u - M/2 + u_0)^2 + (u - N/2 + v_0)^2\right]^{1/2}$$

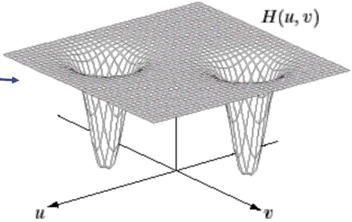


Periodic noise reduction

Butterworth notch filter

$$H(u,v) = \frac{1}{1 + \left[\frac{D_0^2}{D_1(u,v)D_2(u,v)}\right]^{2n}}$$

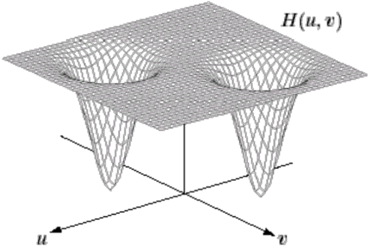
n=2



Periodic noise reduction

Gaussian notch filter

$$H(u,v) = 1 - e^{\frac{-1}{2}\left[\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right]}$$



Periodic noise reduction

Notch filter

- Notch filter that pass, rather than suppress:

$$H_{np}(u, v) = 1 - H_{nr}(u, v)$$

- If $u_0 = v_0 = 0$, NR filters become highpass filters
- If $u_0 = v_0 = 0$, NP filters become lowpass filters

Periodic noise reduction

Optimum Notch filter

- Minimizing the effects of components not presented in estimation of $\eta(x,y)$, by :

$$\hat{f}(u,v) = g(x,y) - w(x,y)\eta(x,y)$$

  $w(x,y)$ is weighted or modulation function
- Here the modulation function is a constant within a neighborhood of size $(2a+1)$ by $(2b+1)$ about a point (x,y)
- We optimize its performance by minimizing the local variance of the restored image at the position (x,y)

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} [\hat{f}(x+s,y+t) - \bar{\hat{f}}(x,y)]^2$$

$$\bar{\hat{f}}(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \hat{f}(x+s,y+t)$$

Periodic noise reduction

Optimum Notch filter

- Points on or near Edge of the image can be treated by considering partial neighborhoods or by zero padding the borders

$$\sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \{[g(x+s,y+t) - w(x+s,y+t)$$

$$\eta(x+s,y+t)] - [\bar{g}(x,y) - \overline{w(x,y)\eta(x,y)}]\}^2$$

- Assumption: $w(x+s,y+t) = w(x,y)$ for $-a \leq s \leq a$ and $-b \leq t \leq b$
  $\Rightarrow \overline{w(x,y)\eta(x,y)} = w(x,y)\bar{\eta}(x,y)$

Periodic noise reduction

Optimum Notch filter

$$\Rightarrow \sigma^2(x,y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^{a} \sum_{t=-b}^{b} \{[g(x+s,y+t)-$$

$$w(x,y)\eta(x+s,y+t)] - [\bar{g}(x,y) - w(x,y)\bar{\eta}(x,y)]\}^2$$

Periodic noise reduction

Optimum Notch filter

- To minimize $\sigma^2(x,y)$

$$\frac{\partial \sigma^2(x,y)}{\partial w(x,y)} = 0$$

$$\Rightarrow w(x,y) = \frac{\overline{g(x,y)\eta(x,y)} - \bar{g}(x,y)\bar{\eta}(x,y)}{\overline{\eta^2}(x,y) - \bar{\eta}^2(x,y)}$$