



Perbaikan Citra (Image Restoration)

IMAGE RESTORATION

- ▮ Seperti halnya Image Enhancement, **tujuan utama teknik restorasi** adalah untuk meningkatkan kualitas suatu citra
- ▮ Restorasi berupaya untuk **merekonstruksi (reconstruct)** atau mendapatkan kembali (recover) suatu citra yang telah mengalami penurunan kualitas (degraded) dengan menggunakan pengetahuan mengenai fenomena degradasi
- ▮ **Teknik restorasi :**
 - ▮ memodelkan degradasi dan menerapkan proses inverse yang bertujuan untuk memulihkan citra asli

IMAGE RESTORATION

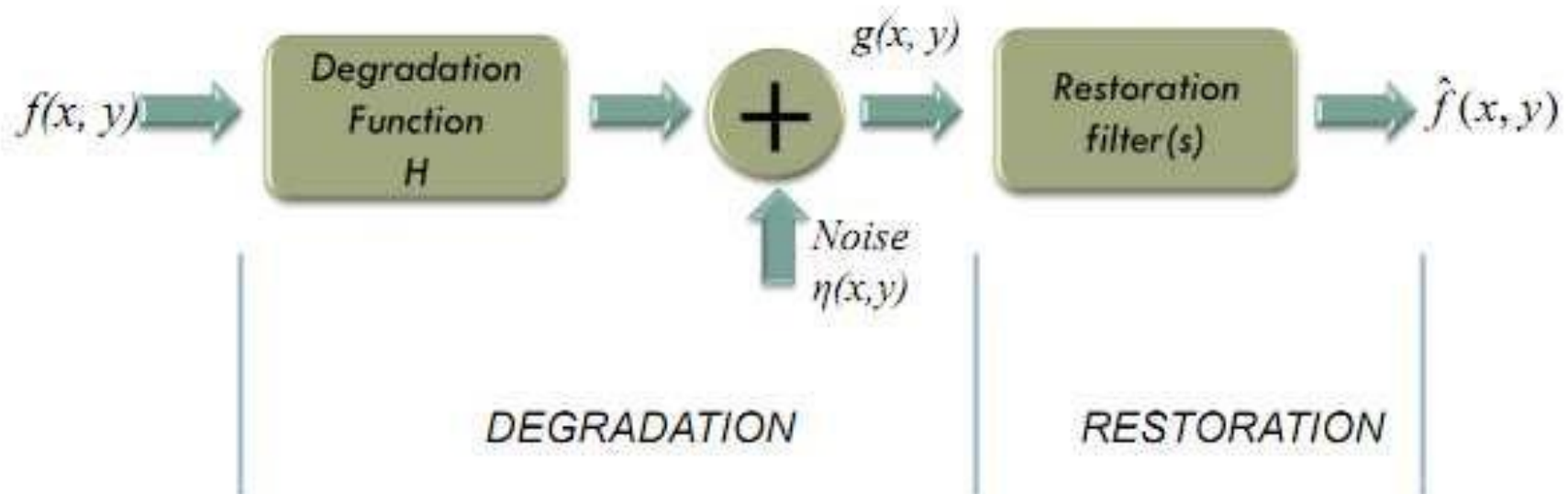
▮ Image Enhancement :

- ▮ memperbaiki kualitas citra untuk tujuan tertentu atau bahkan memberi efek berlebih pada citra

▮ Image Restoration :

- ▮ memperbaiki suatu citra yang terkena noise (model noise sudah diketahui atau diduga sebelumnya)

MODEL DEGRADASI CITRA/PROSES RESTORASI



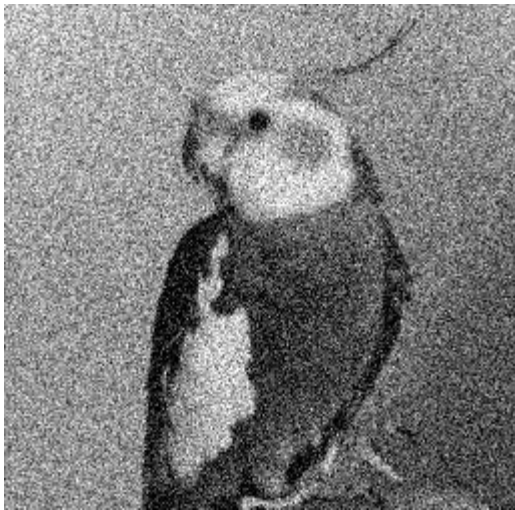
SUMBER NOISE

- Setiap gangguan pada citra dinamakan dengan **noise**
- Noise bisa terjadi :
 - Pada saat proses capture (pengambilan gambar), ada beberapa gangguan yang mungkin terjadi, seperti :
 - Kamera tidak fokus
 - Munculnya bintik-bintik yang bisa jadi disebabkan oleh proses capture yg tdk sempurna
 - Adanya kotoran-kotoran yang terjadi pada citra

KARAKTERISTIK NOISE

- Berdasarkan bentuk dan karakteristiknya, noise pada citra dibedakan menjadi beberapa macam, yakni sebagai berikut :
 - **Gaussian**
 - Merupakan model noise yg mengikuti distribusi normal standard dengan rata-rata nol dan standard deviasi 1
 - **Efek dari noise** ini adalah munculnya titik-titik berwarna yg jumlahnya sama dengan prosentase noise.
 - **Speckle**
 - Merupakan model noise yg membrikan warna hitam pada titik yg terkena noise
 - **Noise salt & pepper**
 - Memberikan noise seperti halnya taburan garam, akan memberikan warna putih pada titik yang terkena noise.

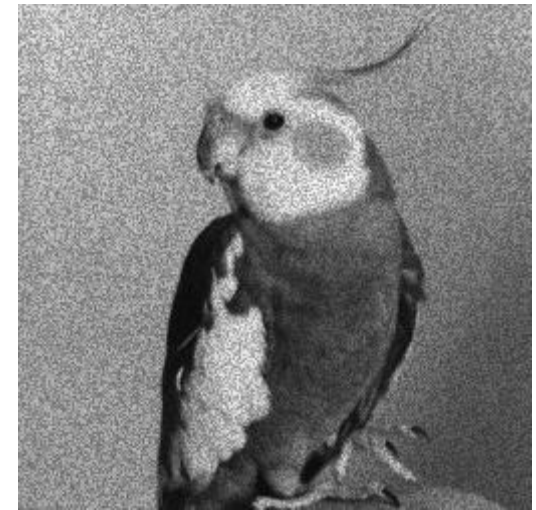
GAMBAR NOISE



A



B



C

Macam-macam Noise (A). Gaussian (B). Salt & Pepper (C) Speckle

NOISE GAUSSIAN

- ▮ Dibuat dengan cara membangkitkan bilangan random $[0,1]$ dengan **distribusi Gaussian**
- ▮ Untuk piksel yang terkena noise, nilai fungsi citra ditambahkan dengan noise yang ada, atau dirumuskan dengan :

$$y(i, j) = x(i, j) + p.a$$

- ▮ Dimana :
 - ▮ a : Bilangan acak berdistribusi Gaussian
 - ▮ p : Prosentase noise
 - ▮ $y(i,j)$: nilai citra yang terkena noise
 - ▮ $x(i,j)$: nilai citra sebelum kena noise

NOISE UNIFORM

- ▮ Noise Uniform seperti halnya noise gaussssian dapat dibangkitkan dengan cara membangkitkan bilangan acak $[0,1]$ dengan distribusi uniform.
- ▮ Kemudian untuk titik-titik yang terkena noise, nilai fungsi citra ditambahkan dengan nilai noise yang ada, atau dirumuskan dengan:

$$y(i, j) = x(i, j) + p.a$$

- ▮ Dimana :

- ▮ a : Bilangan acak berdistribusi Uniform dari noise
- ▮ p : Prosentase noise
- ▮ $y(i,j)$: nilai citra yang terkena noise
- ▮ $x(i,j)$: nilai citra sebelum kena noise

- ▮ Untuk membangkitkan bilangan acak berdistribusi Gaussian, tidak dapat langsung menggunakan fungsi `rnd`, tetapi diperlukan suatu metode yang digunakan untuk mengubah distribusi bilangan acak ke dalam fungsi `f` tertentu

`g = imnoise(f, 'gaussian', m, var)`

Default untuk `m = 0` dan `var = 0.01`

NOISE SPECKLE

- ▮ Noise ini dapat dibangkitkan dengan cara membangkitkan bilangan 0 (warna hitam) pada titiktitik yang secara probabilitas lebih kecil dari nilai probabilitas noise, dan dirumuskan dengan

$$f(x, y) = 0 \text{ jika } p(x, y) < \text{ProbNoise}$$

- ▮ **Dimana :**

- ▮ $f(x, y)$ adalah nilai gray-scale pada titik (x, y)
- ▮ $p(x, y)$ adalah probabilitas acak

- ▮ Jika menggunakan Matlab bisa menggunakan perintah :

$g = \text{imnoise}(f, 'speckle', var)$

Default nilai $var = 0.04$

NOISE SALT & PEPPER

- ▮ Noise ini dapat dibangkitkan dengan cara membangkitkan bilangan 255 (warna putih) pada titik-titik yang secara probabilitas lebih kecil dari nilai probabilitas noise, dan dirumuskan dengan:

$$f(x, y) = 255 \text{ jika } p(x, y) < \text{ProbNoise}$$

- ▮ **Dimana :**

- ▮ $f(x, y)$ adalah nilai gray-scale pada titik (x, y)
- ▮ $p(x, y)$ adalah probabilitas acak

- ▮ Bila menggunakan Matlab bisa menggunakan perintah :

$g = \text{imnoise}(f, 'salt \& pepper', d)$

Default nilai $d = 0.05$

MEAN FILTER

- **Arithmetic Mean Filter**

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

- Dapat diimplementasikan dengan menggunakan mask konvolusi yang semua koefisiennya bernilai **$1/mn$**
- Noise berkurang sebagai akibat dari blurring

$$H = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \text{ atau ditulis } H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} / 9$$

$$H = \begin{bmatrix} \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{bmatrix}$$

MEAN FILTER

- ***Geometric Mean Filter***
- Tiap pixel yang telah dipulihkan (*restored pixel*) diperoleh dari hasil perkalian pixel-pixel pada subimage yang kemudian dipangkatkan dengan ***$1/mn$***
- Lebih mengarah ke smoothing, namun cenderung kehilangan detail citra dalam prosesnya

REDUKSI NOISE MENGGUNAKAN FILTER GAUSSIAN

- Filter gaussian ini sebenarnya hampir sama dengan filter rata-rata hanya ada nilai bobot yang tidak rata seperti pada filter rata-rata, tetapi mengikuti fungsi gaussian sebagai berikut:

$$G(x, y) = \frac{1}{s\sqrt{\pi}} e^{-((x-m_x)^2 + (y-m_y)^2) / 2s}$$

- dimana:
 - s adalah sebaran dari fungsi gaussian
 - (m^x, m^y) adalah titik tengah dari fungsi gaussian

KERNEL GAUSSIAN

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix} / 13 \quad \text{atau} \quad H = \begin{bmatrix} 0.077 & 0.077 & 0.077 \\ 0.077 & 0.308 & 0.077 \\ 0.077 & 0.077 & 0.077 \end{bmatrix}$$

Harmonic Mean Filter

- Harmonic mean filter

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

- Baik** digunakan untuk **salt noise**, namun **buruk** digunakan untuk **pepper noise**
- Selain itu baik juga digunakan untuk model noise yang lain, seperti Gaussian noise

Contraharmonic Mean Filter

▮ *Contraharmonic mean filter*

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

Contraharmonic Mean Filter

- ▮ Q adalah order dari filter Filter ini sesuai digunakan untuk mengurangi efek salt-and-pepper noise
- ▮ Q positif, filter mengurangi pepper noise
- ▮ Q negatif, filter mengurangi salt noise
- ▮ $Q = 0$, reduksi noise dengan arithmetic mean filter
- ▮ $Q = -1$, reduksi noise dengan harmonic mean filter

- ▮ **Arithmetic dan geometric mean filters** → random noise seperti Gaussian atau uniform noise
- ▮ **Contraharmonic filter** → impulse noise, seperti salt- and-pepper noise. Namun kekurangannya, harus diketahui apakah noise gelap atau terang agar dapat menentukan nilai Q yang sesuai

MSE (Mean Square Error)

- ▮ **MSE merupakan** salah satu image quality metrics yang digunakan untuk mengevaluasi algoritma noise reduction
- ▮ **Semakin kecil nilai MSE**, maka **semakin baik** algoritma noise reduction yang digunakan untuk merestorasi citra

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [f(x, y) - \hat{f}(x, y)]^2$$