

# Lab. Image Restoration

# Adding Noise with Function **imnoise**

ฟังก์ชันที่ใช้ใส่สัญญาณรบกวนให้กับรูปภาพคือ **imnoise** ซึ่งมีรูปแบบการใช้ดังนี้

$g = \text{imnoise}(f, \text{type}, \text{parameters})$

- **f** is the input image,
- **type** เป็นชนิดของสัญญาณรบกวน
- **parameters** เป็นค่าตัวแปรของแต่ละชนิดของสัญญาณรบกวน

ฟังก์ชัน **imnoise** จะแปรข้อมูลภาพ **f** เป็นชนิด **double** ที่มีค่าอยู่ในช่วง **[0, 1]** ก่อนที่จะเพิ่มสัญญาณรบกวน ดังนั้นค่าพารามิเตอร์จะต้องสอดคล้องกับข้อมูล เช่น **Gaussian noise** มีค่าเฉลี่ย **64** และความแปรปรวน **400** ในคลาส **uint8** ก็ต้องสเกลค่านี้เป็น **64/255** และความแปรปรวนเท่ากับ **400/(255)<sup>2</sup>** เป็นต้น

## การใช้ `imnoise`

- **`g = imnoise (f, 'gaussian', m, var)`** adds Gaussian noise of mean `m` and variance `var` to image `f`. The default is zero mean noise with 0.01 variance.
- **`g = imnoise (f, 'localvar' , V)`** adds zero-mean, Gaussian noise of local variance, `V`, to image `f`, where `V` is an array of the same size as `f` containing desired variance values at each point.
- **`g = imnoise(f, 'localvar', image_intensity, var)`** adds zero-mean, Gaussian noise to image `f`, where the local variance of the noise, `var`, is a function of the image intensity values in `f`. The `image_intensity` and `var` arguments are vectors of the same size, and `plot (image_intensity, var)` plots the functional relationship between noise variance and image intensity. The `image_intensity` vector must contain normalized intensity values in the range `[0, 1]`.

## การใช้ `imnoise`

- `g = imnoise (f, 'salt & pepper', d)` corrupts image `f` with salt and pepper noise, where `d` is the noise density (i.e., the percent of the image area containing noise values). Thus, approximately  $d \cdot \text{numel}(f)$  pixels are affected. The default is 0.05 noise density.
- `g = imnoise (f, 'speckle', var)` adds multiplicative noise to image `f`, using the equation  $g = f + n \cdot f$ , where `n` is uniformly distributed random noise with mean 0 and variance `var`. The default value of `var` is 0.04.

## การใช้ imnoise

- **g=imnoise (f, 'poisson')** สร้างสัญญาณรบกวนที่มีการกระจายแบบ **Poisson** จากข้อมูลโดยตรง เพื่อให้ค่าความเข้มของจุดภาพเป็นไปตามตัวสถิติของ ซึ่งรองรับคลาส **uint8** และ **uint16** ที่สอดคล้องกับโอกาสการเกิดขึ้นของแต่ละค่าความเข้ม ในกรณีที่จำนวนค่าความเข้มซ้ำกันมากกว่า **65535** (แต่น้อยกว่า  **$10^{12}$** ) ก็จะใช้คลาส **Double** โดยค่าความเข้มจะอยู่ระหว่าง **0** และ **1** เนื่องจากค่าความถี่สะสมจะถูกหารด้วย  **$10^{12}$**

## การสร้างสัญญาณรบกวนแบบสุ่มในสเปาเทียบโดเมน

- สัญญาณรบกวนในสเปาเทียบโดเมนเป็นตัวแปรสุ่มที่กำกับด้วยฟังก์ชันความหนาแน่นความน่าจะเป็น (**Probability Density Function: PDF**) หรืออาจจะใช้ฟังก์ชันการแจกแจงแบบสะสม (**Cumulative Distribution Function: CDF**)
- มีตัวสร้างเลขสุ่มที่สอดคล้องกับรูปแบบการกระจายของสัญญาณรบกวน โดยใช้ **CDF** แบบยูนิฟอร์มที่มีค่าอยู่ในช่วง (**0,1, rand**) แต่ก็มีบางกรณีที่มีการสร้างเลขสุ่มจากเกาส์เซียนที่มีค่าเฉลี่ยเท่ากับศูนย์และความแปรปรวนเท่ากับหนึ่ง (การแจกแจงปกติมาตรฐาน, **randn**)

# การสร้างสัญญาณรบกวนแบบสุ่มในสเปาเทียบโดเมน

- สมมุติฐานในการสร้างสัญญาณรบกวน จะกำหนดให้  $w$  เป็นตัวแปรสุ่มที่มีการแจกแจงแบบยูนิฟอร์ม ที่มีค่าอยู่ในช่วง  $(0,1)$  ดังนั้นตัวแปรสุ่ม  $z$  ของสัญญาณรบกวนของการแจกแจงแบบต่างๆ ใน CDF สามารถกำหนดได้โดย

$$z = F_z^{-1}(w)$$

นั่นก็คือ  $F_z(z) = w$

- เช่นเมื่อต้องการสร้างสัญญาณรบกวนแบบเรเล

$$F_z(z) = \begin{cases} 1 - e^{-(z-a)^2/b} & \text{เมื่อ } z \geq a \\ 0 & \text{เมื่อ } z < a \end{cases}$$

ดังนั้นเราสร้างหาสัญญาณรบกวน  $z$  ได้จาก

$$w = 1 - e^{-(z-a)^2/b}$$

หรือ  $z = a + \sqrt{b \ln(1 - w)}$

# การสร้างสัญญาณรบกวนแบบสุ่มในสเปาเทียบลโดเมน

- จะเห็นว่าค่า  $z = a + \sqrt{b \ln(1 - w)}$  บางครั้งก็จะเรียกว่า “Random number generator equation” หรือ สมการสร้างเลขสุ่ม เนื่องจากสมการนี้ได้บอกว่าจะสร้างเลขสุ่มได้ อย่างไรจากตัวแปร  $w$
- ตารางที่จะแสดงต่อไปได้สรุป ตัวแปรสุ่มที่กำลังสนใจที่ประกอบด้วย **PDF, CDF,** และสมการสร้างเลขสุ่ม จากตารางจะเห็นว่าตัวแปรสุ่ม จาก **PDF** บางตัวสามารถหาสมการสร้างเลขสุ่มได้เช่น เรเลและเอ็กโพเนนเชียล แต่บาง **PDF** ก็ไม่สามารถกำหนดได้ เช่น **Gaussian** และ **lognormal**



# ตารางตัวแปรสุ่มที่ใช้สร้างสัญญาณรบกวน Z

Name	PDF	Mean and Variance	CDF	Generator <sup>†</sup>
Uniform	$p_z(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$	$m = \frac{a+b}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12}$	$F_z(z) = \begin{cases} 0 & z < a \\ \frac{z-a}{b-a} & a \leq z \leq b \\ 1 & z > b \end{cases}$	MATLAB function rand
Gaussian	$p_z(z) = \frac{1}{\sqrt{2\pi b}} e^{- z-a ^2/2b^2}$ $-\infty < z < \infty$	$m = a, \quad \sigma^2 = b^2$	$F_z(z) = \int_{-\infty}^z p_z(v) dv$	MATLAB function randn
Salt & Pepper	$p_z(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$ $b > a$	$m = aP_a + bP_b$ $\sigma^2 = (a-m)^2P_a + (b-m)^2P_b$	$F_z(z) = \begin{cases} 0 & \text{for } z < a \\ P_a & \text{for } a \leq z < b \\ P_a + P_b & \text{for } b \leq z \end{cases}$	MATLAB function rand with some additional logic
Lognormal	$p_z(z) = \frac{1}{\sqrt{2\pi b z}} e^{-[\ln(z)-a]^2/2b^2}$ $z > 0$	$m = e^{a+(b^2/2)}, \quad \sigma^2 = [e^{b^2} - 1]e^{2a+b^2}$	$F_z(z) = \int_0^z p_z(v) dv$	$z = ae^{bN(0,1)}$
Rayleigh	$p_z(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & z \geq a \\ 0 & z < a \end{cases}$	$m = a + \sqrt{\pi b/4}, \quad \sigma^2 = \frac{b(4-\pi)}{4}$	$F_z(z) = \begin{cases} 1 - e^{-(z-a)^2/b} & z \geq a \\ 0 & z < a \end{cases}$	$z = a + \sqrt{b \ln[1 - U(0,1)]}$
Exponential	$p_z(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$	$m = \frac{1}{a}, \quad \sigma^2 = \frac{1}{a^2}$	$F_z(z) = \begin{cases} 1 - e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$	$z = -\frac{1}{a} \ln[1 - U(0,1)]$
Erlang	$p_z(z) = \frac{a^b z^{b-1}}{(b-1)!} e^{-az}$ $z \geq 0$	$m = \frac{b}{a}, \quad \sigma^2 = \frac{b}{a^2}$	$F_z(z) = \left[ 1 - e^{-az} \sum_{n=0}^{b-1} \frac{(az)^n}{n!} \right]$ $z \geq 0$	$z = E_1 + E_2 + \dots + E_b$ (The $E$ 's are exponential random numbers with parameter $a$ )

<sup>†</sup>  $N(0,1)$  denotes normal (Gaussian) random numbers with mean 0 and a variance of 1.  $U(0,1)$  denotes uniform random numbers in the range (0,1).

# การสร้างสัญญาณรบกวนแบบสุ่มในสเปาเทียบโดเมน

- ตัวสร้างเลขสุ่มที่มีอยู่ใน **imnoise** และที่แสดงไว้ในตาราง จะมีบทบาทสำคัญในการจำลองพฤติกรรมของสัญญาณรบกวนเชิงสุ่ม ดังที่ได้แสดงการให้การแจกแจงยูนิฟอรั่ม
- สัญญาณรบกวนแบบเกาส์เซียน ถูกใช้ในการจำลองการเกิดขึ้นของสัญญาณรบกวนในตัวรับรู้สัญญาณภาพในที่มีระดับความเข้มของแสงต่ำ
- แบบจำลองการเกิดสัญญาณรบกวนแบบ **salt&pepper** เกิดขึ้นเนื่องจากความผิดพลาดในการสับสวิตช์
- แบบจำลองการเกิดสัญญาณรบกวนเรเลจะเกิดขึ้นใน **range imaging** ในขณะที่ **Erlang noise** ใช้จำลองสัญญาณรบกวนใน **Laser imaging**

## ตัวอย่างการสร้างสัญญาณรบกวนแบบเรเล

- จากสมการที่ใช้สร้างตัวแปรสุ่มเรเล

$$z = a + \sqrt{b \ln(1 - w)}$$

- $w$  เป็นตัวแปรสุ่มแบบยูนิฟอร์มที่สร้างจากฟังก์ชัน **rand** เมื่อต้องการสร้างสัญญาณรบกวนที่มีขนาด  $M \times N$  ก็สามารถกำหนดได้ดังนี้

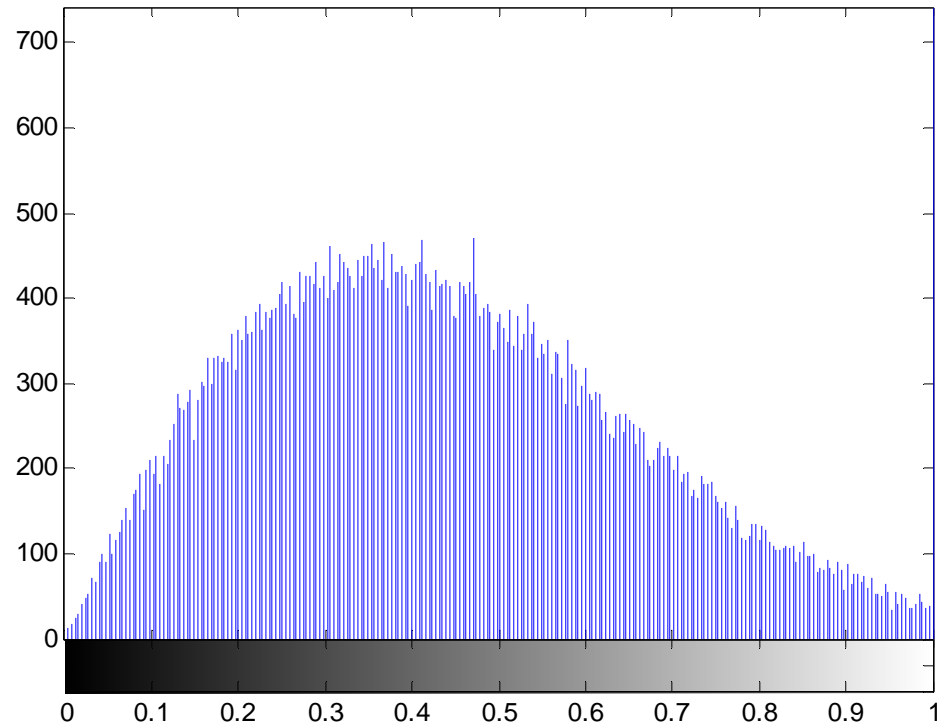
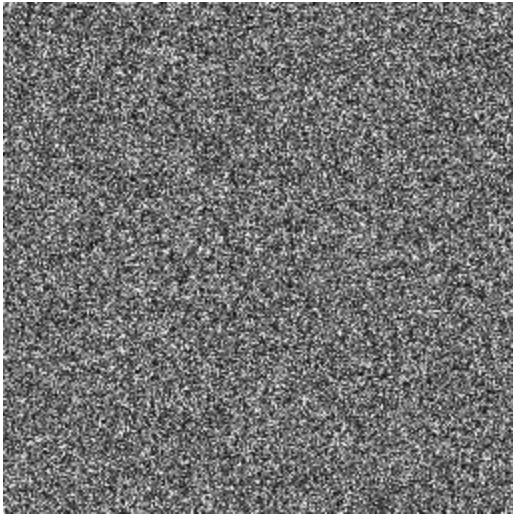
$$w = \text{rand}(M, N);$$

- จากคุณสมบัติของการแจกแจงเรเล เมื่อกำหนดค่าของตัวแปร  $a$  และ  $b$  ก็จะได้ตัวแปรสุ่ม  $z$  ที่แทนสัญญาณรบกวนขนาด  $M \times N$  ได้ดังนี้

$$z = a + \text{sqrt}(b * \log(1 - w));$$

## ตัวอย่างการสร้างสัญญาณรบกวนแบบเวเล

- เมื่อให้  $a=0$ ,  $b=0.25$ ,  $M=N=256$



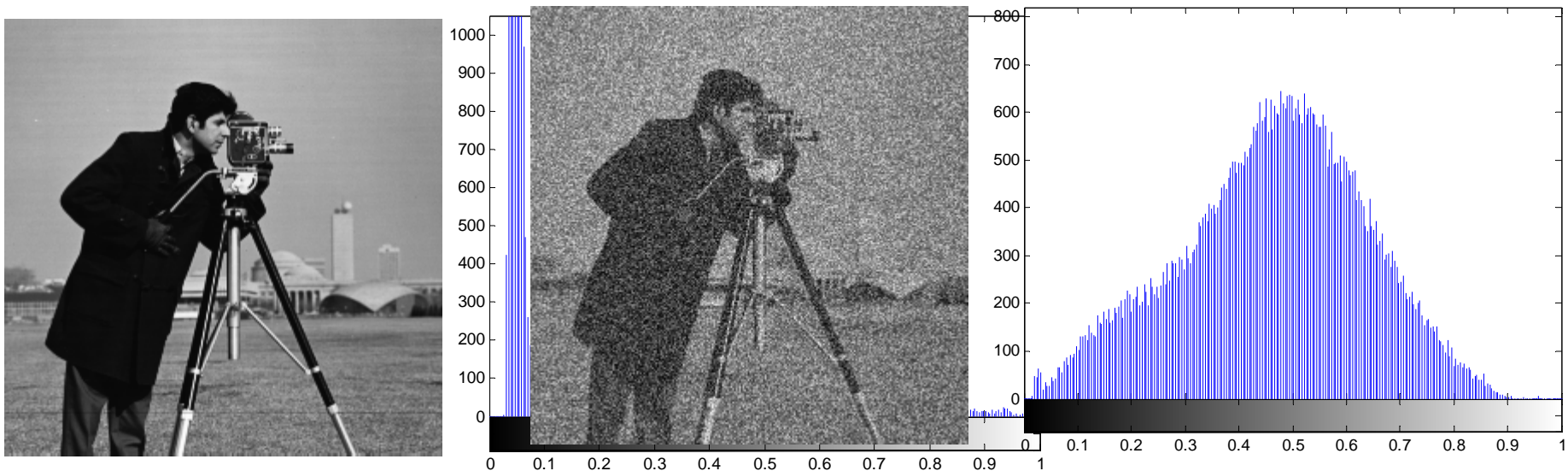
## ตัวอย่างการสร้างสัญญาณรบกวนแบบเวเล

- เมื่อนำสัญญาณรบกวน  $z$  ไปรวมกับข้อมูลภาพตามแบบจำลองของ **additive noise** ก็จะได้ ภาพที่มีสัญญาณรบกวนแบบเวเลดังนี้

```
>> f=imread('cameraman.tif');
```

```
>>g = im2double(f);
```

```
>>gn = g+z;
```



# Restoration(Spatial domain)