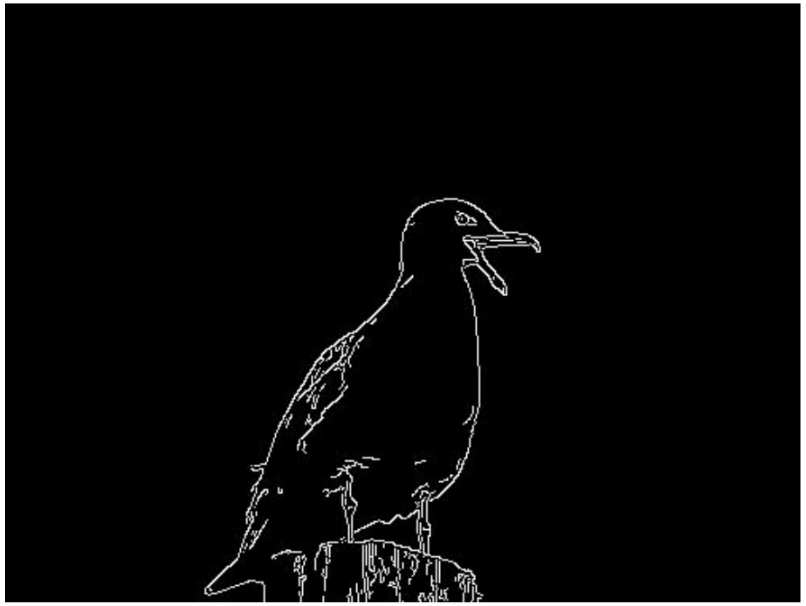
Digital Image Processing ECE 566

Ahmad Ghasemi

Department of Electrical and Computer Engineering

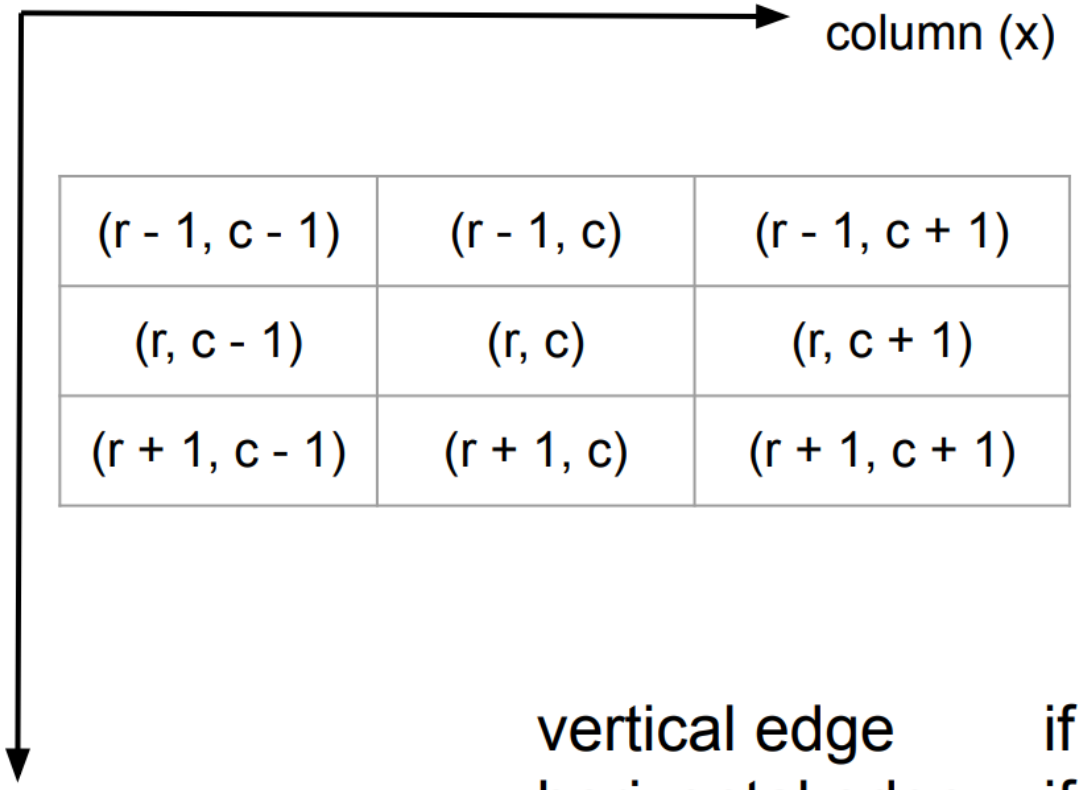
University of Massachusetts Amherst

Edge Detection

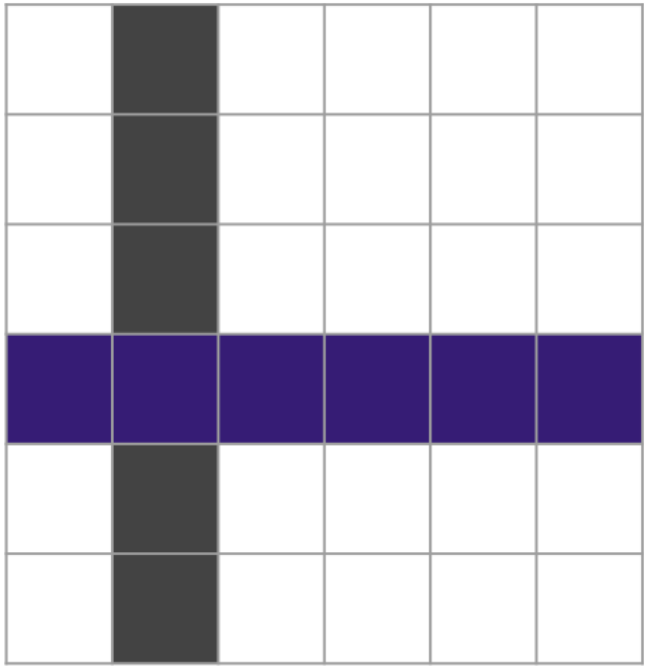


University of Massachusetts Amherst

Define edge



(r - 1, c - 1)	(r - 1, c)	(r - 1, c + 1)
(r, c - 1)	(r, c)	(r, c + 1)
(r + 1, c - 1)	(r + 1, c)	(r + 1, c + 1)



vertical edge if $|i(r,c) - i(r,c-1)| > t$

horizontal edge if $|i(r,c) - i(r-1,c)| > t$

i is the intensity

t is the threshold

Credit: Yung-Hsiang Lu, Purdue course

Define edge

$|i(r,c) - i(r,c-1)| > t$
$|i(r,c) - i(r-1,c)| > t$

define $\Delta_r = 1$ and $\Delta_c = 1$

$$|\frac{i(r,c) - i(r-\Delta_r,c)}{\Delta_r}| > t \qquad |\frac{i(r,c) - i(r,c-\Delta_c)}{\Delta_c}| > t$$

$$|\frac{\partial i}{\partial r}| > t \qquad |\frac{\partial i}{\partial c}| > t$$

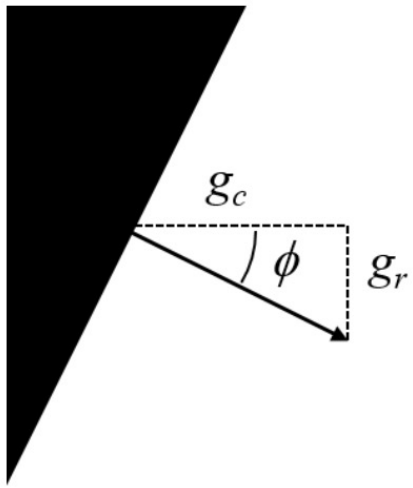
$$\frac{\partial i}{\partial r}\vec{r} + \frac{\partial i}{\partial c}\vec{c} = \nabla i$$

University of Massachusetts Amherst

Define edge

$$\frac{\partial i}{\partial r}\vec{r} + \frac{\partial i}{\partial c}\vec{c} = \nabla i = g_r\vec{r} + g_c\vec{c}$$

$$|\nabla i| = |g_r\vec{r} + g_c\vec{c}| = \sqrt{g_r^2 + g_c^2} > t$$

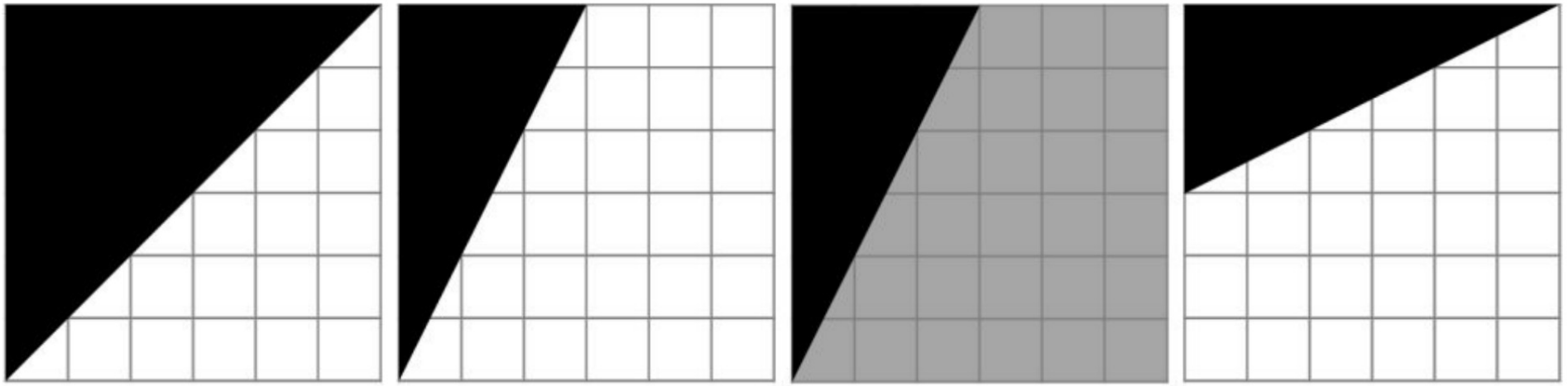


$$\tan(\phi) = \frac{g_r}{g_c}$$

Credit: Yung-Hsiang Lu, Purdue course

University of Massachusetts Amherst

Define edge

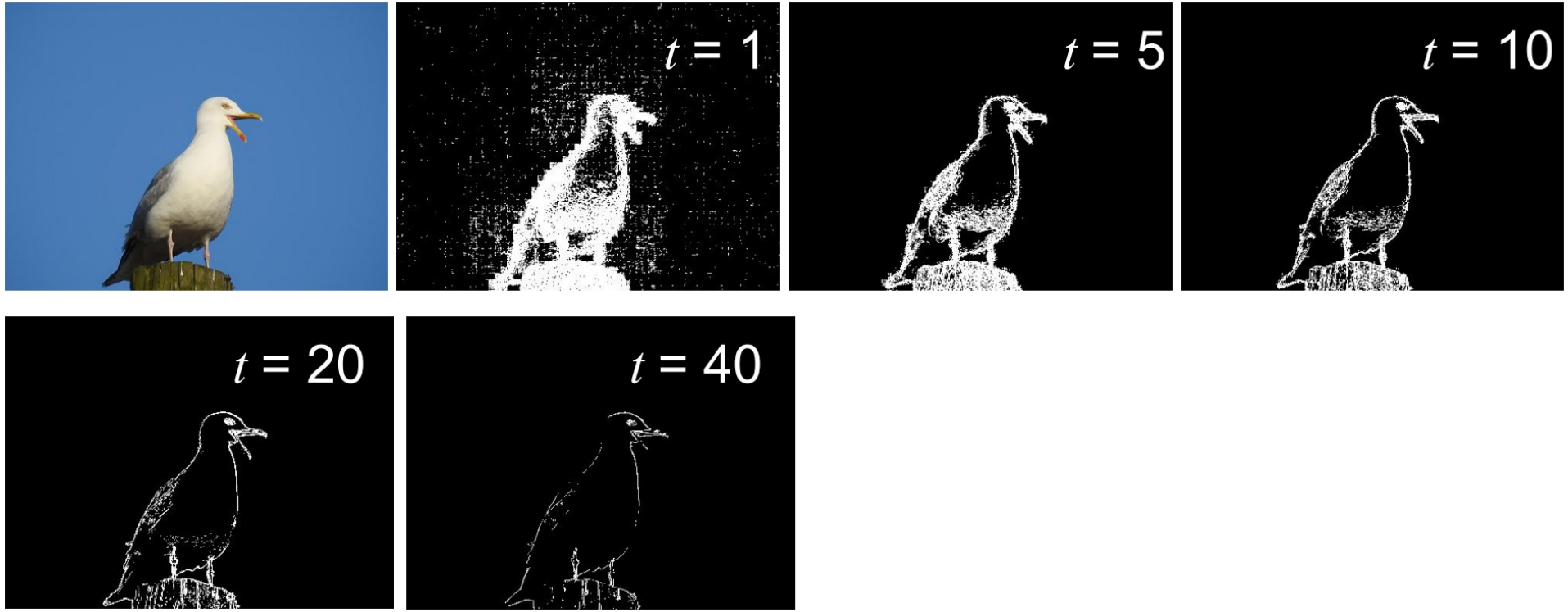


(a) $\alpha\vec{r} + \alpha\vec{c}$ (b) $\alpha\vec{r} + 2\alpha\vec{c}$ (c) $\beta\vec{r} + 2\beta\vec{c}$ (d) $2\alpha\vec{r} + \alpha\vec{c}$

Credit: Yung-Hsiang Lu, Purdue course

University of Massachusetts Amherst

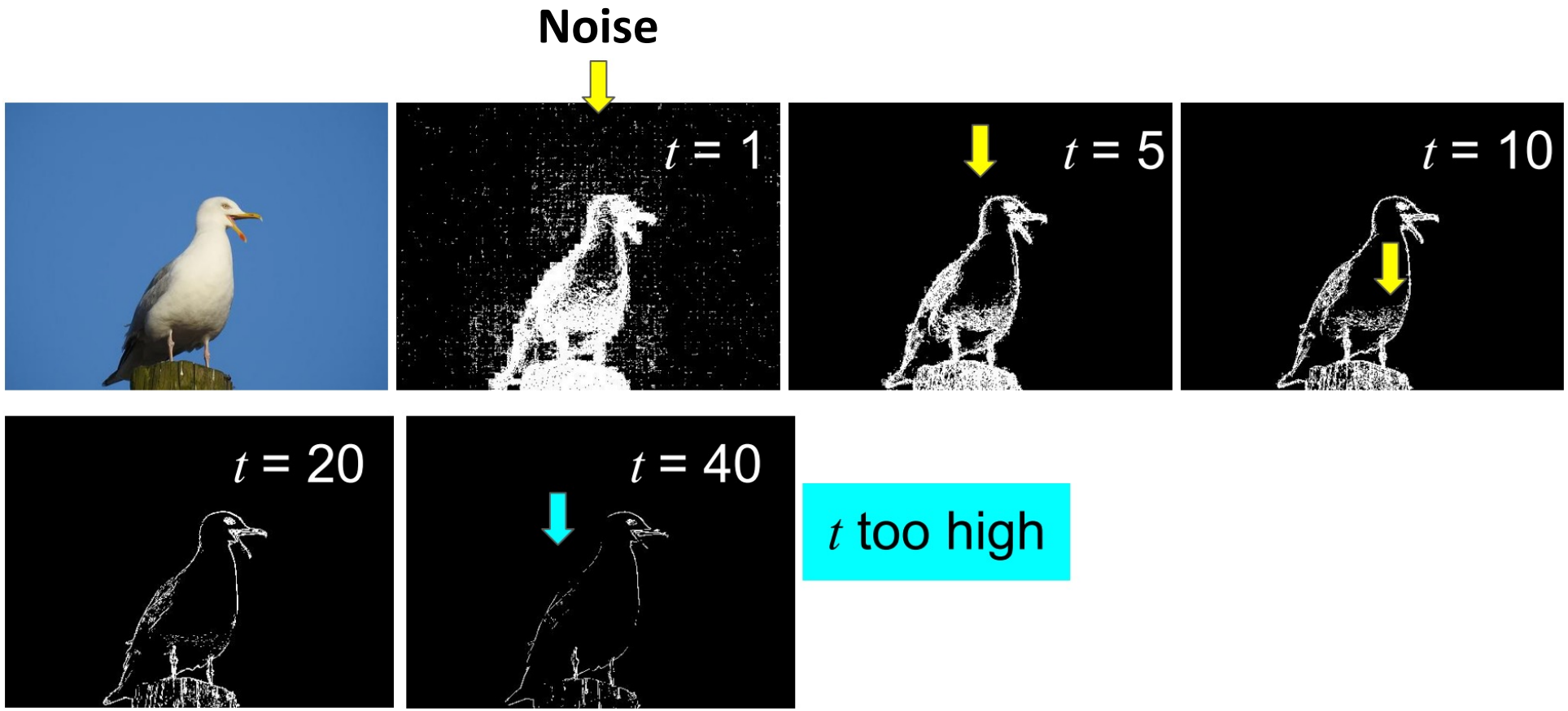
Does it work?



Credit: Yung-Hsiang Lu, Purdue course

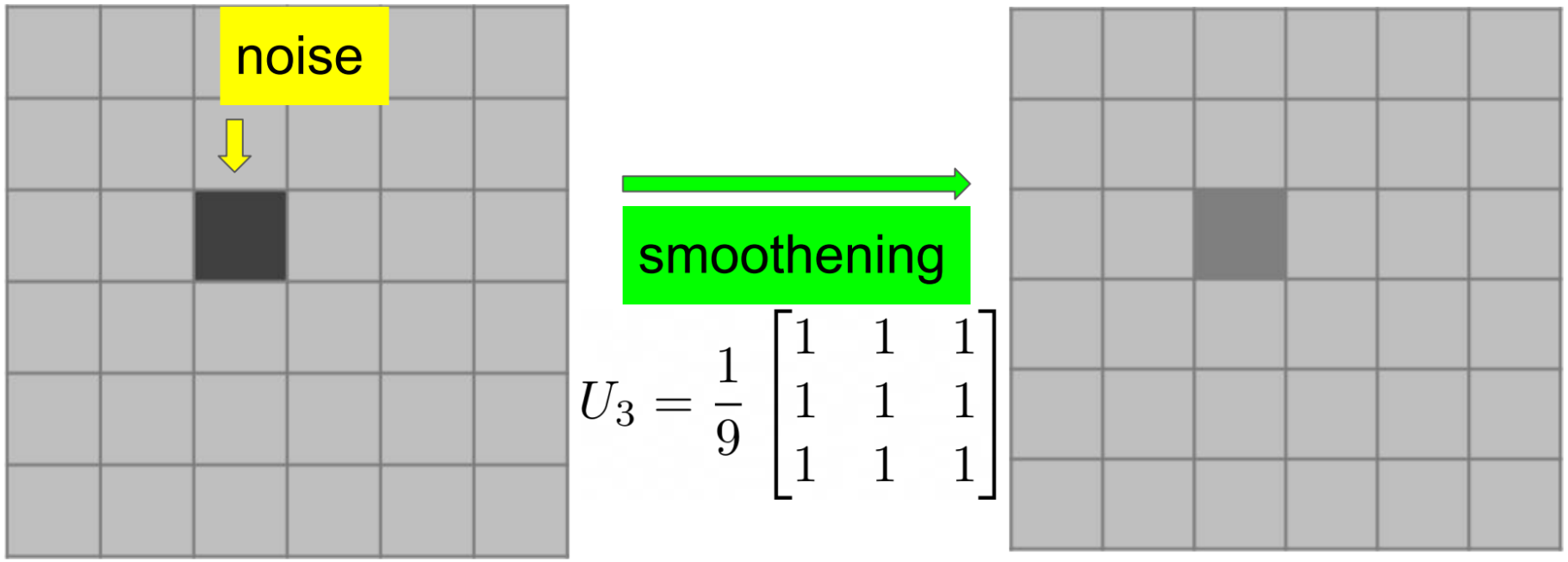
University of Massachusetts Amherst

Does it work?



Credit: Yung-Hsiang Lu, Purdue course

Noise and Smoothing: Uniform Filter

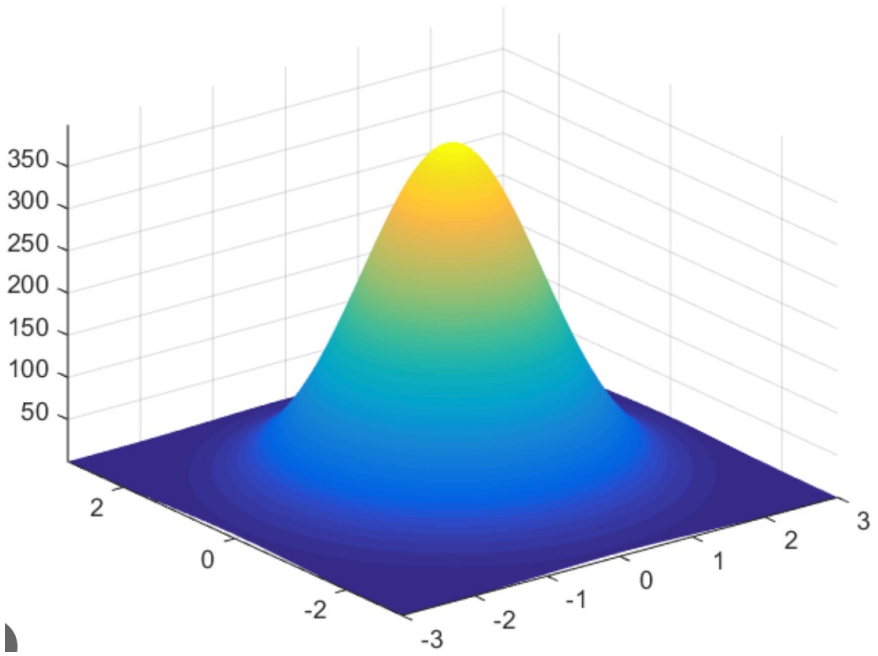


3 x 3 uniform filter

Credit: Yung-Hsiang Lu, Purdue course

University of Massachusetts Amherst

Noise and Smoothing: Gaussian Filter



$$G = \frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

$$G = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 26 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Credit: Yung-Hsiang Lu, Purdue course

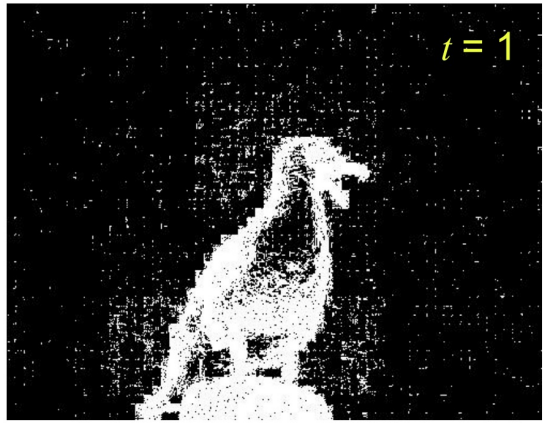
Noise and Smoothing: Gaussian Filter


```
blurimage = cv2.GaussianBlur(grayimage, (5, 5), 0)
```

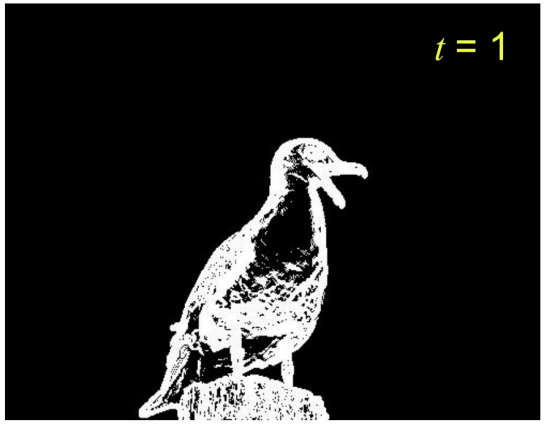

Credit: Yung-Hsiang Lu, Purdue course

University of Massachusetts Amherst

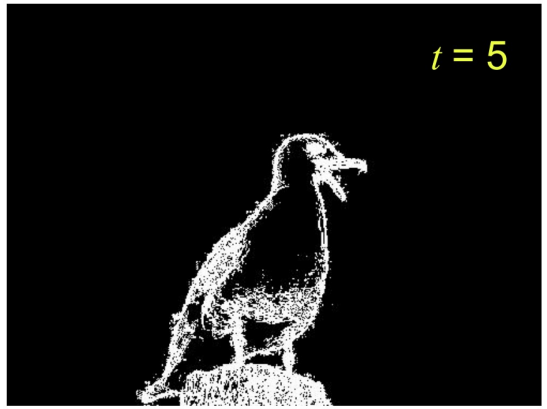
Noise and Smoothing: Gaussian Filter



Gray Level



Blurred



Gray Level

Blurred

Credit: Yung-Hsiang Lu, Purdue course

University of Massachusetts Amherst

Canny Edge Detector

1. 5 x 5 Gaussian filter to reduce noise

Credit: Yung-Hsiang Lu, Purdue course

University of Massachusetts Amherst

Canny Edge Detector

1. 5 x 5 Gaussian filter to reduce noise
2. Sobel filters

$$G_r = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} G_c = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Credit: Yung-Hsiang Lu, Purdue course

University of Massachusetts Amherst

Canny Edge Detector

1. 5 x 5 Gaussian filter to reduce noise
2. Sobel filters

$$G_r = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad G_c = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Convolution outputs: I_x and I_y → $\left\{ \begin{array}{l} \text{Intensity} = \sqrt{I_x^2 + I_y^2} \\ \text{Edge Direction} = \arctan(\frac{I_y}{I_x}) \end{array} \right\}$

Credit: Yung-Hsiang Lu, Purdue course

University of Massachusetts Amherst

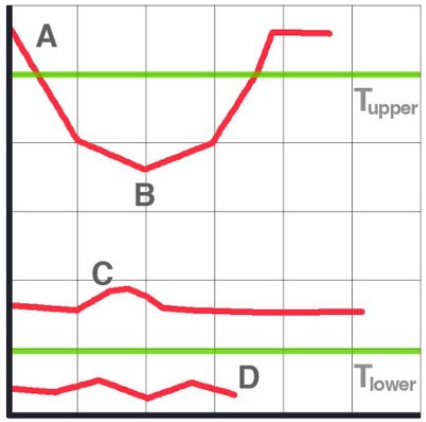
Canny Edge Detector

1. 5 x 5 Gaussian filter to reduce noise
2. Sobel filters

$$G_r = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad G_c = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Convolution outputs: I_x and I_y → $\begin{cases} \text{Intensity} = \sqrt{I_x^2 + I_y^2} \\ \text{Edge Direction} = \arctan(\frac{I_y}{I_x}) \end{cases}$

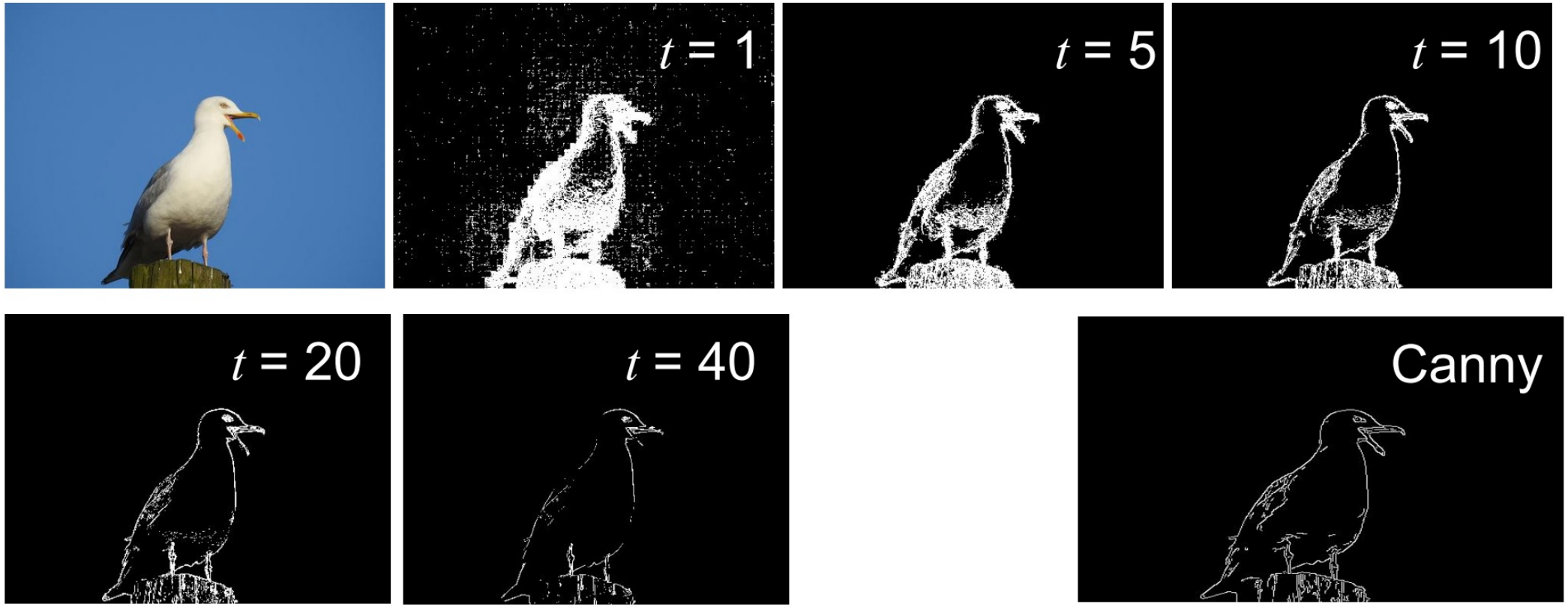
3. adaptive thresholds
   a. A is an edge pixel
   b. B is an edge pixel because of A
   c. C, D are not edge pixels



Credit: Yung-Hsiang Lu, Purdue course

University of Massachusetts Amherst

Does Canny Edge Detector work?



Credit: Yung-Hsiang Lu, Purdue course