CS 512 HW 2

Li Xu
A20300318

1. Noise and filtering.

(a) $SNR = \frac{E_S}{E_R} = \frac{\sigma_s^2}{\sigma_N^2} = \frac{\text{Variance of pixels in a sequence of images}}{\text{Variance in uniform area.}}$

(b) • Gaussian noise is a kind of noise having a probability density function equal to that of normal distribution.

Impulsive noise is a kind of noise that ~~takes~~ contains impulse-like content.

Median filter handles impulsive noise better, because in impulsive noise, maximum and minimum sometimes vary a lot, which caused the average shifts too much.

(c) The value of the pixels after convolution filter is 18.

(d) Because of $\frac{d}{dx}(f*g) = (\frac{d}{dx}f) * g = f * (\frac{d}{dx}g)$,

we can first derive the filter and then use the deriative filter to convolve to the image.

(e) ① Add 0 around the images: Simply add 0's outside the boundaries of the image.

② replicate the pixels value on the boundaries.

③ Ignore the boundaries pixels of the image.

(f) $\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$

the sum of all entries is 1.
Because this convolution filter is doing the average.

(g) Gaussian filter is separatable:

$$(I*G)(x,y) = \sum_x \sum_y I(x,y)\, e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$= \sum_x I(x,y)\, e^{-\frac{x^2}{2\sigma^2}} \sum_y I(x,y)\, e^{-\frac{y^2}{2\sigma^2}}$$

$$I * G = (\underbrace{I * G_x}_{\text{rows}}) * \underbrace{G_y}_{\text{columns}}$$

1D. convolution.

(h) $\sigma = 2$. $\quad m \geq 5\sigma = 10$.

5σ captures the most of gaussion. (99.99%).

(i) image at level J $\boxed{I} \xrightarrow{\text{smooth}} \boxed{G} \xrightarrow{\text{down-sample}} \boxed{\downarrow 2} \longrightarrow \boxed{I'}$ image at level J-1

Gaussian Pyramid.

Gaussian Pyramid is to do the multi-layer analyze. Compared to a single image, we need to do down-sample to a smaller image.

(j) $\boxed{I} \xrightarrow{\text{up-sample}} \boxed{\uparrow 2} \xrightarrow{\text{smooth}} \boxed{G} \longrightarrow \boxed{I'}$ laplacian Pyramid.

Use laplacian pyramid to up-sample to a larger image.

2. Edge detection.

(a) Use edge detection can found the location change in the image, which has a better view of how the image constructed

Edge detection properties: depth discontinuity
dis continuity of normal
texture discontinuity
illumination discontinuity.

(b) ① Smooth: To get rid of the noise. (without affecting edges).
Use convolution filter to smooth.

② edge enhancement: to find image derivatives.

③ localization: to find 2nd derivatives.

(c) sobel and Laplacian filters to compute image gradient. Gradient points to the maximum change direction, which indicates the edges.

(d) $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$ $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Smooth derivation //

$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$

e) $f'[x] \hat{=} f[x] * G'[x]$

$\Rightarrow I_x = I * G' * G_y$

$I_y = I * G_x * G_y'$

$G(x) = e^{-\frac{x^2}{2\sigma^2}}$ $G'(x) = \frac{-2x}{2\sigma^2} e^{-\frac{x^2}{2\sigma^2}} = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$

0.04	0.14	0.32	0.61	0.88	1.0	0.88	0.61	0.32	0.14	0.04	$G(x)$
-5	-4	-3	-2	-1	0	1	2	3	4	5	

$\sigma = 2$

0.05	0.14	0.24	0.30	0.22	1	-0.22	-0.30	-0.24	.014	-0.05	$G'(x)$
-5	-4	-3	-2	-1	0	1	2	3	4	5	

(f) To computed the first derivative to find gradient max to localization the edge.

Find zero-crossings in 2nd derivative to localize the edge.

$(\nabla^2 G * I)$

(g) $LOG = \Delta G \frac{r^2 - 2}{1} e^{\frac{-r^2}{2}}$

$= (r^2 - 2) e^{\frac{-r^2}{2}}$, $r = \sqrt{x^2 + y^2}$

First compute convolution with LoG, then determine : $\begin{cases} = 1 & I * LoG > 0, B \\ = 0 & I * LoG \leq 0, W \end{cases}$

Find edge and color change in transition.

(h) Canny edge detection uses direction derivative.

Condition: ① non-maximum Suppression

② hysteresis thresholding

(i) non-maximum is to find local max gradient magnitute in direction of gradient by comparing with their neighbours.

hysteresis thresholding is to find $D(I*G)_{ij} > T_H$ ~~where~~.

Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong ones.