4F12: Computer Vision
Examples Paper 1 – Answers

1. *Images*

Storage requirement for a stereo pair of HD video cameras:

$$2 \times \frac{1\text{ byte}}{\text{pixel}} \times \frac{(1920 \times 1080)\text{pixels}}{\text{frame}} \times \frac{25\text{ frames}}{\text{second}} \approx 1.04 \times 10^8\text{ bytes/second}$$

A page roughly contains 4200 characters, each requiring a byte. Therefore:

$$1.04 \times 10^8\text{ bytes} \times \left(\frac{4.2 \times 10^3\text{ bytes}}{\text{page}}\right)^{-1} \approx 2.5 \times 10^4\text{ pages}$$

2. *Smoothing by convolution with a Gaussian*

Consider two gaussian filters with different filter size – σ_1 and σ_2.

$$g_{\sigma_1}(x) = \frac{1}{\sigma_1\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma_1^2}\right) \quad \textit{and} \quad g_{\sigma_2}(x) = \frac{1}{\sigma_2\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma_2^2}\right)$$

Then, our goal is to show that their convolution is equal to a single Gaussian filter of variance $\sigma_1^2 + \sigma_2^2$. Once this relation is proven, it can be generalized to arbitrary number of Gaussians.

$$g_{\sigma_1}(x) * g_{\sigma_2}(x) = g_{\sqrt{\sigma_1^2+\sigma_2^2}}(x)$$

Using the convolution theorem: $\mathcal{F}\big(f(x) * g(x)\big) = \mathcal{F}\big(f(x)\big)\mathcal{F}\big(g(x)\big)$,

$$g_{\sigma_1}(x) * g_{\sigma_2}(x) = \mathcal{F}^{-1}\left[\mathcal{F}\left(g_{\sigma_1}(x)\right)\mathcal{F}\left(g_{\sigma_2}(x)\right)\right]$$

The Fourier transform of a Gaussian is:

$$\mathcal{F}\left(g_{\sigma_1}(x)\right) = \int_{-\infty}^{+\infty} g_{\sigma_i}(x)\exp(-2\pi ikx)\,dx$$

Substituting $g_{\sigma_1}(x)$ and using $\exp(-i\theta) = \cos\theta - i\sin\theta$,

$$\mathcal{F}\left(g_{\sigma_1}(x)\right) = \frac{1}{\sigma_1\sqrt{2\pi}}\int_{-\infty}^{+\infty}\exp\left(-\frac{x^2}{2\sigma_1^2}\right)\cos(2\pi kx)\,dx$$

The sine term is discarded since it is an odd function. Now, Recall the following integral rule.

$$\int_{-\infty}^{+\infty}\exp(-at^2)\cos(2xt)\,dt = \sqrt{\frac{\pi}{a}}\exp\left(-\frac{x^2}{a}\right)$$

Apply this to obtain the Fourier transform.

$$\mathcal{F}\left(g_{\sigma_1}(x)\right) = \exp(-2\sigma_1^2\pi^2k^2)$$

Apply this to the convolution theorem to obtain the designated result.

$$g_{\sigma_1}(x) * g_{\sigma_2}(x) = \mathcal{F}^{-1}\left[\mathcal{F}\left(g_{\sigma_1}(x)\right)\mathcal{F}\left(g_{\sigma_2}(x)\right)\right]$$

$$= \mathcal{F}^{-1}[\exp(-2(\sigma_1^2 + \sigma_2^2)\pi^2 k^2)]$$
$$= \frac{1}{\sqrt{2\pi(\sigma_1^2 + \sigma_2^2)}} \exp\left(-\frac{x^2}{2(\sigma_1^2 + \sigma_2^2)}\right)$$

3. *Generating the Gaussian filter kernel*

(a) Following is a 1D Gaussian filter of $\sigma = 1$.

$$g(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

The following table shows the coefficients obtained for different values of x. Since the value of $g(x)$ becomes less than 1/1000 of the peak value from $|x| \geq 4$, the filter size should be set to 7 pixels (from $x = -3$ to $x = +3$). The symmetry

x	0	± 1	± 2	± 3	± 4
$g(x)$	0.399	0.242	0.054	0.004	0.0001

The regions of uniform intensity is unaffected by the Gaussian filter since the coefficients add up to 1 (approximately).

(b) Suppose the coefficient becomes smaller than 1/1000 of the peak value at $x = n + 1$. Then, the following should be satisfied.

$$\exp\left(-\frac{(n+1)^2}{2\sigma^2}\right) < \frac{1}{1000}$$

Simplifying the above inequality yields the following.

$$\frac{(n+1)^2}{2\sigma^2} > \log 1000 \quad \rightarrow \quad n > \sqrt{2 \log 1000}\,\sigma - 1 \approx 3.7\sigma - 1$$

Hence, the nearest integer to $3.7\sigma - 0.5$ satisfies the designated condition. Setting $\sigma = 5$, resulting n is 18, suggesting that the kernel size should be 37 pixels.

(c) The value of σ should be set in accordance with the scale of interest. A filter with low σ can detect edges at fine scale, while that with high σ can detect edges of larger scale.

4. *Discrete Convolution*

$$\begin{aligned} s(x) &= \sum_{u=-n}^{+n} g(u) I(x-u) \\ &= 0.004 \times 134 + 0.054 \times 133 + 0.242 \times 130 + 0.399 \times 118 \\ &\quad + 0.242 \times 99 + 0.054 \times 77 + 0.004 \times 57 \\ &= 115 \end{aligned}$$

5. *Derivative of Convolution Theorem*

(a)

$$\begin{aligned} \frac{d}{dx}[g_\sigma(x) * I(x)] &= \frac{d}{dx}\int_{-\infty}^{+\infty} g_\sigma(x-u) I(u) du \\ &= \int_{-\infty}^{+\infty} \frac{d}{dx} g_\sigma(x-u) I(u) du \\ &= \int_{-\infty}^{+\infty} g'_\sigma(x-u) I(u) du = g'_\sigma(x) * I(x) \end{aligned}$$

(b) Edges occur at local extrema of the derivative of the intensity of the smoothed signal. Applying the abovementioned relation, the condition becomes equivalent to the zero-crossings of $g_\sigma''(x) * I(x)$.

$$\frac{d}{dx}[g_\sigma'(x) * I(x)] = g_\sigma''(x) * I(x) = 0$$

6. *Differentiation and 1D Edge Detection*

The derivative of a function can be defined and approximated as following.

$$\frac{dS}{dx} = \lim_{h \to 0} \frac{S(x+h) - S(x-h)}{2h} \approx \frac{S(x+1) - S(x-1)}{2}$$

Such operation is equivalent to the convolution with the kernel $(1/2, 0, -1/2)$

$$\begin{aligned}\frac{dS}{dx} \approx \sum_{u=-1}^{+1} g(u)S(x-u) &= g(-1)S(x+1) + g(+1)S(x-1) \\ &= \frac{S(x+1) - S(x-1)}{2}\end{aligned}$$

Smoothed	48	50	53	56	64	79	98	115	126	132	133	133	132
Derivative	×	2.5	3.0	5.5	11.5	17	18	14	8.5	3.5	0.5	−0.5	×

The above table shows the first order derivatives of the smoothed row of pixels. The intensity discontinuity occurs at the pixel with intensity 98.

7. *Decomposition of 2D Convolution*

$$\begin{aligned}G_\sigma(x,y) * I(x,y) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{1}{2\pi\sigma^2}\exp\left(-\frac{u^2+v^2}{2\sigma^2}\right) I(x-u, y-v)dudv \\ &= \frac{1}{2\pi\sigma^2}\int_{-\infty}^{+\infty} \exp\left(-\frac{u^2}{2\sigma^2}\right)\left[\int_{-\infty}^{+\infty} \exp\left(-\frac{v^2}{2\sigma^2}\right) I(x-u, y-v)dv\right] du \\ &= g_\sigma(x) * [g_\sigma(y) * I(x,y)]\end{aligned}$$

At each pixel, a 2D convolution requires $(2n+1)^2$ multiplications while two 1D convolutions only require $2(2n+1)$ multiplications.

8. *Isotropic and Directional Edge Finders*

(a) Isotropic Edge Finder (Marr-Hildreth)

$$\begin{aligned}\nabla^2 G_\sigma(x,y) &= \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\left[\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)\right] \\ &= \frac{1}{2\pi\sigma^2}\left[\frac{x^2+y^2}{\sigma^4} - \frac{2}{\sigma^2}\right]\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \\ &= -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)\end{aligned}$$

Since the value of $x^2 + y^2$ is invariant under rotation, the Laplacian of Gaussian also possesses rotational symmetry. Suppose the coordinates x and y are placed along an ideal step edge. The intensity would be positive and constant in one direction (namely I_{step}) and zero in the other direction. Then, it can be shown that the operator produces zero. Here, the coordinates u and v are parameterized using r and θ

$$
\begin{aligned}
\nabla^2 G_\sigma(x,y) * I(x,y) &= \int_{-\pi}^{+\pi}\int_0^{+\infty} -\frac{1}{\pi\sigma^4}\left[1-\frac{r^2}{2\sigma^2}\right]\exp\left(-\frac{r^2}{2\sigma^2}\right) I(x-r\cos\theta\,, y-r\sin\theta)\,dr\,d\theta \\
&= \int_{\theta'}^{\theta'+\pi}\int_0^{+\infty} -\frac{1}{\pi\sigma^4}\left[1-\frac{r^2}{2\sigma^2}\right]\exp\left(-\frac{r^2}{2\sigma^2}\right) I_{step}\,dr\,d\theta \\
&= \int_{\theta'}^{\theta'+\pi}\left[-\frac{1}{\sigma^4} r\exp\left(-\frac{r^2}{2\sigma^2}\right)\right]_{r=0}^{r=+\infty} I_{step}\,d\theta = 0
\end{aligned}
$$

This suggests that the operator produces zero if the signal $I(x-r\cos\theta\,, y-r\sin\theta)$ has no dependency on r.

(b) Directional Edge Finder (Canny)
The Canny operator looks for the local maximum of $|\nabla(G_\sigma * I)|$ in the direction of $\hat{\mathbf{n}}$. Such condition can be formally written as following.

$$\frac{\partial|\nabla(G_\sigma * I)|}{\partial s} = 0$$

By definition, the gradient of the smoothed signal $G_\sigma * I$ has magnitude of $|\nabla(G_\sigma * I)|$ in the direction $\hat{\mathbf{n}}$. In other words, differentiating $G_\sigma * I$ along $\hat{\mathbf{n}}$ would yield $|\nabla(G_\sigma * I)|$.

$$\frac{\partial(G_\sigma * I)}{\partial s} = |\nabla(G_\sigma * I)|$$

Therefore, finding the maximum of $|\nabla(G_\sigma * I)|$ in the direction $\hat{\mathbf{n}}$ is equivalent to finding the zero-crossing of the second derivative.

$$\frac{\partial|\nabla(G_\sigma * I)|}{\partial s} = \frac{\partial^2(G_\sigma * I)}{\partial s^2} = 0$$

(c) Advantages & Disadvantages
Isotropic edge detector computes two convolutions, $g''_\sigma(x)$ and $g''_\sigma(y)$, and search for zero-crossings. Directional detector computes two convolutions, $g'_\sigma(x)$ and $g'_\sigma(y)$, calculate the direction $\hat{\mathbf{n}}$ at each pixel, and find the local maximum along that direction. Hence, the isotropic detector has lower computational cost.

9. Auto-correlation and Corner Detection

(a) The Taylor expansion of $S(\mathbf{x}+\mathbf{n})$ can be written as following.

$$S(x+n_x, y+n_y) \approx S(x,y) + \frac{\partial}{\partial x}S(x,y)\cdot n_x + \frac{\partial}{\partial y}S(x,y)\cdot n_y$$

Hence, the difference between two patches $S(\mathbf{x}+\mathbf{n})$ and $S(\mathbf{x})$ can be approximated as $\nabla S(\mathbf{x})\cdot\mathbf{n}$.

$$S(\mathbf{x}+\mathbf{n}) \approx S(\mathbf{x}) + \nabla S(\mathbf{x})\cdot\mathbf{n} \quad \rightarrow \quad \left(S(\mathbf{x}+\mathbf{n}) - S(\mathbf{x})\right)^2 \approx S_n^2$$

(b) Following shows how the weighted SSD can be expressed as $\mathbf{n}^T A\mathbf{n}$

$$
\begin{aligned}
C(\mathbf{n}) &= \sum_{\mathbf{x}\in W} w(\mathbf{x})\left(\frac{\partial}{\partial x}S(x,y)\cdot n_x + \frac{\partial}{\partial y}S(x,y)\cdot n_y\right)^2 \\
&= n_x^2\sum_{\mathbf{x}\in W} w(\mathbf{x})\left(\frac{\partial S}{\partial x}\right)^2 + 2n_x n_y\sum_{\mathbf{x}\in W} w(\mathbf{x})\frac{\partial S}{\partial x}\frac{\partial S}{\partial y} + n_y^2\sum_{\mathbf{x}\in W} w(\mathbf{x})\left(\frac{\partial S}{\partial y}\right)^2 \\
&= n_x^2\langle S_x\rangle + n_x n_y\langle S_x S_y\rangle + n_y^2\langle S_y\rangle = \mathbf{n}^T A\mathbf{n}
\end{aligned}
$$

(c) The directional derivative can be written as:

$$I_n = \nabla I(x,y) \cdot \hat{\mathbf{n}} = [I_x \quad I_y]^T \cdot \hat{\mathbf{n}}$$

The autocorrelation matrix can be obtained by squaring the derivative:

$$I_n^2 = \frac{\mathbf{n}^T \nabla I \nabla I^T \mathbf{n}}{\mathbf{n}^T \mathbf{n}} = \frac{\mathbf{n}^T \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \mathbf{n}}{\mathbf{n}^T \mathbf{n}}$$

Lastly, the smoothing can be done by convolving the matrix with a 2D Gaussian filter.

$$C_n(x,y) = G_\sigma * I_n^2 = \frac{\mathbf{n}^T \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix} \mathbf{n}}{\mathbf{n}^T \mathbf{n}}$$

(d) The value of $C_n(x,y)$ represents the smoothed intensity change in the direction $\mathbf{n}$. If the coordinates are placed on edge, the directional change would be small along the edge, and large in the normal-to-edge direction. If the coordinates are placed on a corner, the directional change should be large in every direction. In other words, a pixel can be marked as corner if both the lower and upper bound for $C_n(x,y)$ are large. Instead of finding the eigenvalues of the autocorrelation matrix, its determinant (which is equivalent to $\lambda_1\lambda_2$) and trace (which is equivalent to $\lambda_1 + \lambda_2$) are used. In practice, the value of $\lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2$ is used.

10. Feature Detection and Scale Space

(a) The image pyramid makes use of the incremental blurs. In each octave, the Gaussian filter of $G(\sigma_k)$ is applied so that the σ is double at the end of the octave. Suppose an octave consists of s images, then the value of σ should be multiplied by $2^{1/s}$ after each image. From this the value of σ_k can be obtained.

$$\sqrt{\sigma_i^2 + \sigma_k^2} = 2^{\frac{1}{s}}\sigma_i \quad \rightarrow \quad \sigma_k = \sigma_i\sqrt{2^{\frac{2}{s}} - 1}$$

After smoothing the original image s times, the original image is subsampled (e.g. so that its width and height are half the original). Then the subsampled image undergoes same procedure, to fill the next octave. Such process can be repeated to form an image pyramid.

(b) Band-pass filtering can be implemented using the Difference of Gaussians. Subtracting an image in the pyramid from the one directly above it produces the corresponding band-pass filtered image. The DoG can be approximated as the Laplacian of Gaussians, and can thus be used as a blob-detector. Once the DoG images are obtained across different scales, the pixels can be compared to its neighbors – both in the same scale and in the neighboring scales. The position and corresponding scale of such local extrema would specify the position and scale of the blob-like shape in the image.

(c) Firstly, zero-normalised patches of different images can be compared using cross-correlation. However, such method is not robust to the change in orientation. Secondly, the response to the filter bank, consisting of edge and blob detecting filters of different scales, can be used to characterize and compare each pixel.

Lastly, the SIFT descriptor can be obtained at each pixel. The SIFT algorithm finds the local extrema in the DoG images. Then, the points with low contrast, and the points place on edges are discarded. The orientation of each remaining candidate is obtained as the primary direction of the intensity gradient. Each keypoint is then characterized using the intensity gradients, binned and normalised in each cell of the given patch of image. The resulting numbers are again treated as high dimensional vectors, which can be compared using algorithms such as kNN.