Math 369: Lab 9

Prof. Schaeffer, TA: Linan Zhang

Background: For Lab 9, you will write a program that takes a 2D image and outputs its edges. The input data (left) and sample output (right) are shown below. In Lab 8, we defined a 1D filter and obtain a smoother approximation via an integral equation given a 1D signal. Similarly, we will define a 2D filter and obtain a smoother approximation via a multi-variable integral equation given a 2D image. For 2D images, we will need to extend the data past the boundaries in order to define the integral. This is referred to as padding.

 

Problem: Let $f(x, y)$ be the noisy input image and let $u(x, y)$ be a smooth approximation. To find $u(x, y)$, you will write an algorithm that approximates the following equation:

$$u(x, y) := \int_{1-r}^{n+r} \int_{1-r}^{m+r} f(s_1, s_2) G(x - s_1, y - s_2)\, \mathrm{d}s_2\, \mathrm{d}s_1\,, \tag{1}$$

where G in the truncated 2D Gaussian filter with parameters $\sigma > 0$ and $r > 0$:

$$G(x, y) := \begin{cases} \dfrac{1}{2\pi\sigma^2} \exp\left(-\dfrac{x^2 + y^2}{2\sigma^2}\right) & \text{if } (x, y) \in [-r, r] \times [-r, r], \\ 0 & \text{otherwise.} \end{cases}$$

Assume that f satisfies the "replicated boundary condition", i.e.,

$$\begin{aligned} f(1-t, y) &= f(1+t, y), \quad & f(n-t, y) &= f(n+t, y), \\ f(x, 1-t) &= f(x, 1+t), \quad & f(x, m-t) &= f(x, m+t), \end{aligned}$$

for all $0 \leq t \leq r$, $1 \leq x \leq n$, and $1 \leq y \leq m$.

Solve Equation (1) using the composite trapezoidal rule with no more than two loops. A pseudo-code including edge detection of u will be provided. You should turn in your code and a plot for the edges.

Brief notes: This is an application of numerical integration.

Theorem 1. *Let $f \in C^2([a,b] \times [c,d])$, $h = (b-a)/n$, $k = (d-c)/m$, $x_i = a + ih$ for each $i = 0, 1, \ldots, n$, and $y_j = c + jk$ for each $j = 0, 1, \ldots, m$. The composite trapezoidal rule is*

$$\int_a^b \int_c^d f(x,y)\,\mathrm{d}y\,\mathrm{d}x \approx \frac{hk}{4}(I_1 + 2I_2 + 4I_3),$$

where

$$\begin{aligned}
I_1 &= f(a,c) + f(b,c) + f(a,d) + f(b,d),\\
I_2 &= \sum_{i=1}^{n-1} f(x_i, c) + \sum_{i=1}^{n-1} f(x_i, d) + \sum_{j=1}^{m-1} f(a, y_j) + \sum_{j=1}^{m-1} f(b, y_j),\\
I_3 &= \sum_{j=1}^{n-1} \sum_{i=1}^{m-1} f(x_i, y_j).
\end{aligned}$$

The error term of the composite trapezoidal rule is $\mathcal{O}(h^2) + \mathcal{O}(k^2)$.

The integral to be approximated is given by

$$u(x,y) := \int_{1-r}^{n+r} \int_{1-r}^{m+r} f(s_1, s_2) G(x - s_1, y - s_2)\,\mathrm{d}s_2\,\mathrm{d}s_1 \tag{2}$$

for $(x,y) \in [1,n] \times [1,m]$, where f is an image defined on $[1,n] \times [1,m]$, and G is the truncated 2D Gaussian filter with parameters $\sigma > 0$ and $r > 0$:

$$G(x,y) := \begin{cases} \dfrac{1}{2\pi\sigma^2} \exp\left(-\dfrac{x^2 + y^2}{2\sigma^2}\right) & \text{if } (x,y) \in [-r,r] \times [-r,r],\\ 0 & \text{otherwise.} \end{cases}$$

Since f is only defined on $[1,n] \times [1,m]$, we extend the domain of f to $[1-r, n+r] \times [1-r, m+r]$ by assuming that f satisfies the "replicated boundary condition", i.e.,

$$\begin{aligned}
f(1-t, y) &= f(1+t, y), \quad & f(n-t, y) &= f(n+t, y),\\
f(x, 1-t) &= f(x, 1+t), \quad & f(x, m-t) &= f(x, m+t),
\end{aligned}$$

for all $0 \leq t \leq r$, $1 \leq x \leq n$, and $1 \leq y \leq m$.

Fix $x \in \{1, 2, \ldots, n\}$ and $y \in \{1, 2, \ldots, m\}$. The composite trapezoidal rule applied to Equation (2) is

$$u(x,y) \approx \frac{1}{4}(I_1 + 2I_2 + 4I_3),$$

where

$$I_1 = H(1-r, 1-r) + H(n+r, 1-r) + H(1-r, m+r) + H(n+r, m+r), \tag{3a}$$

$$I_2 = \sum_{i=2-r}^{n+r-1} H(i, 1-r) + \sum_{i=2-r}^{n+r-1} H(i, n+r) + \sum_{j=2-r}^{m+r-1} H(1-r, j) + \sum_{j=2-r}^{m+r-1} H(n+r, j), \tag{3b}$$

$$I_3 = \sum_{j=2-r}^{n+r-1} \sum_{i=2-r}^{m+r-1} H(i,j), \tag{3c}$$

where

$$H(i,j) = f(i,j)G(x-i, y-j)$$

for $i = 1-r, 2-r, \ldots, n+r$ and $j = 1-r, 2-r, \ldots, m+r$.

Since MATLAB index starts at 1, we convert Equation (3) into "MATLAB syntax" as follows. Define a (discrete) function F such that

$$F(i,j) = f(i-r, j-r)$$

for $i = 1, 2, \ldots, n+2r$ and $j = 1, 2, \ldots, m+2r$. Then Equation (3) is equivalent to:

$$\begin{aligned}
J_1 &= W(1,1) + W(n+2r,1) + W(1,m+2r) + W(n+2r,m+2r), \\
J_2 &= \sum_{i=2}^{n+2r-1} W(i,1) + \sum_{i=2}^{n+2r-1} W(i,n+2r) + \sum_{j=2}^{m+2r-1} W(1,j) + \sum_{j=2}^{m+2r-1} W(n+2r,j), \\
J_3 &= \sum_{j=2}^{n+2r-1} \sum_{i=2}^{m+2r-1} W(i,j),
\end{aligned}$$

where

$$W(i,j) = F(i,j)G(x-i+r, y-j+r)$$

for $i = 1, 2, \ldots, n+2r$ and $j = 1, 2, \ldots, m+2r$.