

RBE549 - Final Exam

Keith Chester

Due date: December 12, 2022

Problem 1

In this problem, we are tasked with solving an iterative optical flow problem. To compute optical flow, we learned an iterative method to update $u(x, y)$, $v(x, y)$ at each iteration, according to:

$$\begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix}^2 = \begin{bmatrix} \lambda I_x^2 + 4 & \lambda I_x I_y \\ \lambda I_x I_y & \lambda I_y^2 + 4 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{n \in \text{neighbors}(x, y)} u^{\text{old}}(n) - \lambda I_x I_t \\ \sum_{n \in \text{neighbors}(x, y)} v^{\text{old}}(n) - \lambda I_y I_t \end{bmatrix} \quad (1)$$

We are asked to consider a local coordinate frame (x', y') where x' is aligned with the image gradient and y' is perpendicular to the image gradient. Likewise, $(u', v') = (\frac{dx'}{dt}, \frac{dy'}{dt})$ are the image velocities in this frame. In this coordinate frame,

$$I_{x'} = \sqrt{I_x^2 + I_y^2} \text{ and } I_{y'} = 0 \quad (2)$$

We wish to show that the update equations:

$$\begin{bmatrix} u'(x, y) \\ v'(x, y) \end{bmatrix}^2 = \begin{bmatrix} \lambda I_{x'}^2 + 4 & \lambda I_{x'} I_{y'} \\ \lambda I_{x'} I_{y'} & \lambda I_{y'}^2 + 4 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{n \in \text{neighbors}(x, y)} u'^{\text{old}}(n) - \lambda I_{x'} I_t \\ \sum_{n \in \text{neighbors}(x, y)} v'^{\text{old}}(n) - \lambda I_{y'} I_t \end{bmatrix} \quad (3)$$

...can reduce to

$$u'^{\text{new}}(x, y) = \bar{u}'^{\text{old}} - \frac{I_{x'}^2 \bar{u}'^{\text{old}} + I_{x'} I_t}{I_{x'}^2 + \frac{4}{\lambda}} \quad (4)$$

$$v'^{\text{new}}(x, y) = \bar{v}'^{\text{old}} \quad (5)$$

. To do this, we first expand our second term to an equivalent form to make matters easier for us to work with. Specifically, the inverse of a 2×2 matrix is:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad (6)$$

...and thus...

$$\begin{bmatrix} \lambda I_{x'}^2 + 4 & \lambda I_{x'} I_{y'} \\ \lambda I_{x'} I_{y'} & \lambda I_{y'}^2 + 4 \end{bmatrix}^{-1} = \frac{1}{(\lambda I_{x'}^2 + 4)(\lambda I_{y'}^2 + 4) - \lambda I_{x'} I_{y'} \lambda I_{x'} I_{y'}} \begin{bmatrix} \lambda I_{y'}^2 + 4 & -\lambda I_{x'} I_{y'} \\ -\lambda I_{x'} I_{y'} & \lambda I_{x'}^2 + 4 \end{bmatrix} \quad (7)$$

...which simplifies to:

$$\frac{1}{4\lambda I_{x'}^2 + 4\lambda I_{y'}^2 + 16} \begin{bmatrix} \lambda I_{y'}^2 + 4 & -\lambda I_{x'} I_{y'} \\ -\lambda I_{x'} I_{y'} & \lambda I_{x'}^2 + 4 \end{bmatrix} \quad (8)$$

This in turn leads us to an expanded view from earlier:

$$u'^{\text{new}}(x, y) = \frac{1}{4(\lambda I_{x'}^2 + \lambda I_{y'}^2 + 4)} \begin{bmatrix} \lambda I_{y'}^2 + 4 & -\lambda I_{x'} I_{y'} \end{bmatrix} \begin{bmatrix} 4\bar{u}'^{\text{old}} - \lambda I_{x'} I_t \\ 4\bar{v}'^{\text{old}} - \lambda I_{y'} I_t \end{bmatrix} \quad (9)$$

...We can then expand it into this:

$$u'^{new}(x, y) = \frac{1}{4(\lambda I_{x'}^2 + \lambda I_{y'}^2 + 4)} ((\lambda I_{y'}^2 + 4)(4\bar{u}'^{old} - \lambda I_{x'} I_t) - (\lambda I_{x'} I_{y'})(4\bar{v}'^{old} + \lambda I_{x'} I_{y'} \lambda I_{y'} I_t)) \quad (10)$$

$$u'^{new}(x, y) = \frac{\lambda I_{y'}^2 4\bar{u}'^{old} - \lambda I_{y'}^2 \lambda I_{x'} I_t + 16\bar{u}'^{old} - 4\lambda I_{x'} I_t - (\lambda I_{x'} I_{y'}) 4\bar{v}'^{old} + \lambda^2 I_{y'}^2 I_{x'} I_t}{4(\lambda I_{x'}^2 + \lambda I_{y'}^2 + 4)} \quad (11)$$

$$u'^{new}(x, y) = \frac{4(\lambda I_{y'}^2 \bar{u}'^{old} + 4\bar{u}'^{old} - \lambda I_{x'} I_t - \lambda I_{x'} I_{y'} \bar{v}'^{old})}{4(\lambda I_{x'}^2 + \lambda I_{y'}^2 + 4)} \quad (12)$$

$$u'^{new}(x, y) = \frac{(\lambda I_{y'}^2 \bar{u}'^{old} + 4\bar{u}'^{old} - \lambda I_{x'} I_t - \lambda I_{x'} I_{y'} \bar{v}'^{old})}{(\lambda I_{x'}^2 + \lambda I_{y'}^2 + 4)} \quad (13)$$

$$u'^{new}(x, y) = \frac{\bar{u}'^{old}(\lambda I_{y'}^2 + 4) - \lambda I_{x'}(I_t + I_{y'} \bar{v}'^{old})}{\lambda(I_{x'}^2 + I_{y'}^2 + \frac{4}{\lambda})} \quad (14)$$

We established earlier per the problem specifications that $I_{y'} = 0$, so if we plug that in here:

$$u'^{new}(x, y) = \frac{\bar{u}'^{old}(0^2 + 4) - \lambda I_{x'}(I_t + 0\bar{v}'^{old})}{\lambda(I_{x'}^2 + 0^2 + \frac{4}{\lambda})} = 4\bar{u}'^{old} - \frac{\lambda I_{x'} I_t}{\lambda(I_{x'}^2 + \frac{4}{\lambda})} \quad (15)$$

$$u'^{new}(x, y) = 4\bar{u}'^{old} - \frac{I_{x'} I_t}{I_{x'}^2 + \frac{4}{\lambda}} = \bar{u}'^{old} - \frac{I_{x'}^2 \bar{u}'^{old} + I_{x'} I_t}{I_{x'}^2 + \frac{4}{\lambda}} \quad (16)$$

...which matches what we were seeking to reduce to for $u'^{new}(x, y)$. Doing this for $v'^{new}(x, y)$ we would follow a similar path - we would find that:

$$v'^{new}(x, y) = \bar{v}'^{old} - \frac{I_{y'}}{I_{x'}^2 + I_{y'}^2 + \frac{4}{\lambda}} (I_{x'} \bar{u}'^{old} + I_{y'} \bar{v}'^{old} + I_t) \quad (17)$$

...and again, since we were told that $I_{y'} = 0$:

$$v'^{new}(x, y) = \bar{v}'^{old} - \frac{0}{I_{x'}^2 + 0^2 + \frac{4}{\lambda}} (I_{x'} \bar{u}'^{old} + 0\bar{v}'^{old} + I_t) = \bar{v}'^{old} - 0 = \bar{v}'^{old} \quad (18)$$

...thus proving that:

$$u'^{new}(x, y) = \bar{u}'^{old} - \frac{I_{x'}^2 \bar{u}'^{old} + I_{x'} I_t}{I_{x'}^2 + \frac{4}{\lambda}} \quad (19)$$

$$v'^{new}(x, y) = \bar{v}'^{old} \quad (20)$$

Problem 2

We can represent an object by its boundary, $(x(s), y(s)), 0 \leq s \leq S$, where S is the length of the object's boundary and s is distance along that boundary from some arbitrary starting point. Combine x and y into a single complex function $z(s) = x(s) + jy(s)$. The Discrete Fourier Transform (DFT) of z is:

$$Z(k) = \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} z(s), 0 \leq k \leq S-1 \quad (21)$$

We can use the coefficients $Z(k)$ to represent the object boundary. The limit on s is $S-1$ because for a closed contour $z(S) = z(0)$. The Inverse Discrete Fourier Transform is:

$$z(s) = \frac{1}{S} \sum_{k=0}^{S-1} e^{+2\pi j \frac{ks}{S}} Z(k), 0 \leq s \leq S-1 \quad (22)$$

A

Suppose that the object is translated by $(\Delta x, \Delta y)$, that is, $z'(s) = z(s) + \Delta x + j\Delta y$. How is z' 's DFT $Z'(k)$ related to $Z(k)$?

$$Z(k) = \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} z(s), 0 \leq k \leq S-1 \quad (23)$$

Which we can define $z'(s)$ as:

$$z'(s) = z(s) + \Delta x + j\Delta y \quad (24)$$

If we plug this into our original equation, we get...

$$Z'(k) = \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} z(s) + \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} (\Delta x + j\Delta y)(1) \quad (25)$$

Here we see that we have a segment that is equivalent to our defined $Z(k)$, so we can simplify by expressing:

$$Z(k) + (1)(\Delta x + j\Delta y) \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} \quad (26)$$

With Δx and $j\Delta y$ isolated, we can use a table of known Fourier Transforms to identify the resulting conversion. Based on our problem's definition ΔX and Δy are both constants, so we can state that:

$$Z'(k) = Z(k) + (\Delta x + j\Delta y) \sigma\left(\frac{2\pi k}{S}\right) \quad (27)$$

...where $\frac{1}{S}$ acts as our scaling factor.

B

In this section, we are asked to suppose that the object is scaled by an integer constant c , that is $z'(s) = cz(s)$. For simplicity, we are to assume that $S' = S$. How is $Z'(k)$ as the DFT of z' related to $Z(k)$? Starting with our definition of $Z(k)$, which is our Fourier Transform of $z(s)$ as defined earlier:

$$Z(k) = \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} z(s), 0 \leq k \leq S-1 \quad (28)$$

If we then define $z'(s)$ as above, $z'(s) = cz(s)$, we can plug it into our Fourier Transform:

$$Z'(k) = \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} cz(s) \quad (29)$$

Since c is a constant, we know we can pull it in front of the fourier Transform like so:

$$Z'(k) = c \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} z(s) \quad (30)$$

...and further simplified:

$$Z'(k) = cZ(k) \quad (31)$$

...which shows that multiplying our time series function by a given scalar simply multiplies our frequency domain by the same scalar.

C

We wish to know what object has $z(s) = [x_0 + R \cos(\frac{2\pi s}{s})] + j [y_0 + R \sin(\frac{2\pi s}{s})]$. Below we created a graph drawing the resulting shape, and include the code utilized to generate it. Arbitrary values were chosen for S , r , x_0 , and y_0 for the sake of plotting.

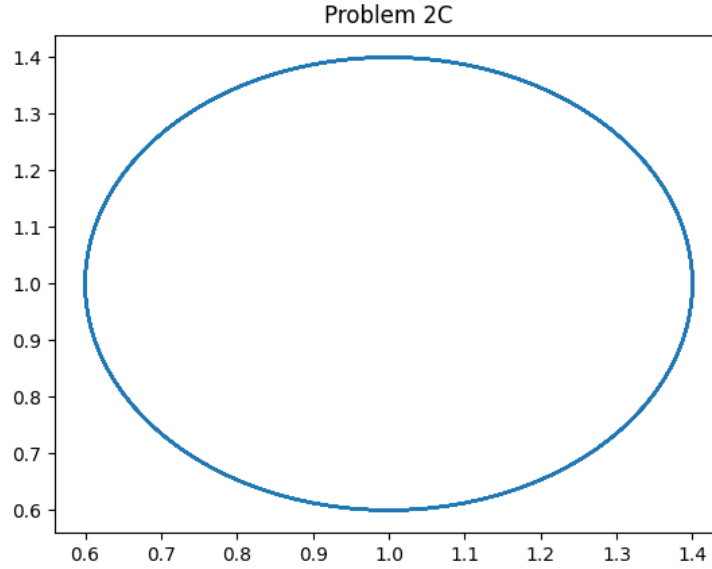


Figure 1: Our resulting shape

```
import numpy as np
from numpy import cos, sin, pi
import matplotlib.pyplot as plt

figure = plt.figure()
plt.title("Problem 2C")

S = 10
r = 4
x0 = 1
y0 = 1

theta = [theta for theta in np.arange(0, S, 0.01)]
X = [
    x0 + r * cos(2*pi*theta)/S
    for theta in theta
]
Y = [
    y0 + r * sin(2*pi*theta)/S
    for theta in theta
]

# Plot the results
plt.plot(X, Y)
plt.savefig("./imgs/prob2_c.png")
```

D

What is $Z(k)$ corresponding to $z(s)$ from Part C? To do this, we begin with our $z(s)$:

$$z(s) = [x_0 + R \cos(\frac{2\pi s}{s})] + j [y_0 + R \sin(\frac{2\pi s}{s})] \quad (32)$$

We can utilize the inverse of Euler's formula; $\cos(x) = \frac{e^{ix} + e^{-ix}}{2}$ and $\sin(t) = \frac{e^{ix} - e^{-ix}}{2ix}$. This allows us to expand our starting equation:

$$z(s) = x_0 + R \frac{e^{j\frac{2\pi s}{S}} + e^{-j\frac{2\pi s}{S}}}{2} + jy_0 + R \frac{e^{j\frac{2\pi s}{S}} - e^{-j\frac{2\pi s}{S}}}{2} \quad (33)$$

$$z(s) = x_0 + jy_0 + \frac{R}{2} (e^{j\frac{2\pi s}{S}} + e^{-j\frac{2\pi s}{S}} + e^{j\frac{2\pi s}{S}} - e^{-j\frac{2\pi s}{S}}) \quad (34)$$

$$z(s) = x_0 + jy_0 + \frac{R}{2} (2e^{j\frac{2\pi s}{S}}) \quad (35)$$

$$z(s) = x_0 + jy_0 + Re^{j\frac{2\pi s}{S}} \quad (36)$$

The Discrete Fourier Transform (DFT) from earlier in our problem we can begin to expand this equation now. Starting with:

$$Z(k) = \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} z(s), 0 \leq k \leq S-1 \quad (37)$$

...which can lead us to:

$$Z(k) = \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} (x_0 + jy_0 + Re^{j\frac{2\pi s}{S}}) \quad (38)$$

...constants can be pulled out, leaving us with:

$$Z(k) = \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} (x_0 + jy_0) + R \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} e^{j\frac{2\pi s}{S}} \quad (39)$$

$$Z(k) = (x_0 + jy_0) \sum_{s=0}^{S-1} e^{-2\pi j \frac{ks}{S}} + R \sum_{s=0}^{S-1} e^{-j2\pi(k-1)\frac{s}{S}} \quad (40)$$

We know from a Fourier Transform lookup table we can then convert this to:

$$Z(k) = (x_0 + jy_0) \sigma\left(\frac{2\pi k}{S}\right) + \sigma\left(2\pi \frac{k-1}{S}\right) \quad (41)$$