CS/RBE 549 Computer Vision Name: **Sol U. Tion**
Fall 2016 Wednesday 07 Dec 2016

Exam #2 Solutions

1. **Iterative Optical Flow (30 pts):** An iterative method for computing optical flow updates $u(x,y), v(x,y)$ at each iteration according to

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

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

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

Show that the update equations in this coordinate frame

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

reduce to

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

Hint: This is not as hard as it looks.

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{u}'(\boldsymbol{x},\boldsymbol{y}) \\ \boldsymbol{v}'(\boldsymbol{x},\boldsymbol{y}) \end{bmatrix}^{\text{new}} &= \begin{bmatrix} \boldsymbol{\lambda I_{x'}^2 + 4} & \boldsymbol{\lambda I_{x'} I_{y'}} \\ \boldsymbol{\lambda I_{x'} I_{y'}} & \boldsymbol{\lambda I_{y'}^2 + 4} \end{bmatrix}^{-1} \begin{bmatrix} \sum_{\boldsymbol{n} \in \textbf{neighbors}} \boldsymbol{u}'^{\text{old}}(\boldsymbol{n}) - \boldsymbol{\lambda I_{x'} I_t} \\ \sum_{\boldsymbol{n} \in \textbf{neighbors}} \boldsymbol{v}'^{\text{old}}(\boldsymbol{n}) - \boldsymbol{\lambda I_{y'} I_t} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{\lambda I_{x'}^2 + 4} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{4} \end{bmatrix}^{-1} \begin{bmatrix} \sum_{\boldsymbol{n} \in \textbf{neighbors}} \boldsymbol{u}'^{\text{old}}(\boldsymbol{n}) - \boldsymbol{\lambda I_{x'} I_t} \\ \sum_{\boldsymbol{n} \in \textbf{neighbors}} \boldsymbol{v}'^{\text{old}}(\boldsymbol{n}) \end{bmatrix} \\
&= \begin{bmatrix} \frac{\boldsymbol{1}}{\boldsymbol{\lambda I_{x'}^2 + 4}} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\boldsymbol{1}}{\boldsymbol{4}} \end{bmatrix} \begin{bmatrix} \sum_{\boldsymbol{n} \in \textbf{neighbors}} \boldsymbol{u}'^{\text{old}}(\boldsymbol{n}) - \boldsymbol{\lambda I_{x'} I_t} \\ \sum_{\boldsymbol{n} \in \textbf{neighbors}} \boldsymbol{v}'^{\text{old}}(\boldsymbol{n}) \end{bmatrix}
\end{aligned}$$

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

2. **Object Representation (40 pts):** 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. We can combine *x* and *y* into a single complex function $z(s) = x(s) + jy(s)$. The Discrete Fourier Transform (DFT) of *z* is

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

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

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

a. (10 pts) 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 $Z(k)$?

**Only the DC *k*=0 component is affected.** $\boldsymbol{Z'(0) = Z(0) + S(\Delta x + j\Delta y), Z'(k) = Z(k)}$ **for** $\boldsymbol{k > 0}$

b. (15 pts) Suppose that the object is scaled by integer constant *c*, that is, $z'(s) = cz(s)$. For simplicity, assume that $S' = S$. How is $z'$'s DFT $Z'(k)$ related to $Z(k)$?

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

c. (10 pts) What object has $z(s) = [x_0 + R\cos\frac{2\pi s}{S}] + j[y_0 + R\sin\frac{2\pi s}{S}]$? Sketch it.

**A circle with radius** $\boldsymbol{R}$ **centered at** $\boldsymbol{(x_0, y_0)}$**.**

d. (5 pts) What is $Z(k)$ corresponding to $z(s)$ from Part c? Hint: Most coefficients are 0.

$$\boldsymbol{Z(0) = S(x_0 + jy_0), Z(1) = SR, Z(k) = 0 \text{ for } k > 1}$$

3. **Chain Code (30 pts):** An object boundary is represented by an 8-connected chain code C with values $c_i \in \{0, \ldots, 7\}$. Each $c_i$ represents a segment at 45° = π/4 radians, with $c_i = 0$ representing (1, 0), $c_i = 1$ representing (1, 1), $c_i = 2$ representing (0, 1), etc. Note that segments have length 1 when $c_i$ is even, and length $\sqrt{2}$ when $c_i$ is odd. Computer Vision researcher Ima Robot claims that the chain code for a closed contour of length *N* must obey Eq (1):

$$\sum_{i=0}^{N-1} e^{-j\frac{\pi}{4}c_i} = 0$$

a) (15 pts) Show that Eq (1) holds for a square

*C* = [0, 0, 2, 2, 4, 4, 6, 6]

and octagon

*C* = [0, 1, 2, 3, 4, 5, 6, 7]

**Square:** $\boldsymbol{1 + 1 + j + j - 1 - 1 - j - j = 0}$**. Octagon:** $\boldsymbol{1 + \frac{1+j}{\sqrt{2}} + j + \frac{-1+j}{\sqrt{2}} - 1 - \frac{1+j}{\sqrt{2}} - j - \frac{-1+j}{\sqrt{2}} = 0}$**.**

b) (15 pts) Show that Eq (1) holds for any closed contour or provide a counterexample.

**Counterexample: Triangle = [1, 4, 6].**  $\boldsymbol{\frac{1+j}{\sqrt{2}} - 1 - j \neq 0}$

4. **Teamwork (1 pt):** On a scale of 1 to 5, with 1 being the lowest, 3 is neutral and 5 being the highest, rate how well your project team is working together.
**No right / wrong answer.**