Computer Vision
Homework 04: Optical flow, Image modeling
CS 670, Fall 2019

Name: Kunjal Panchal
Student ID: 32126469
Email: kpanchal@umass.edu

Oct 06, 2019

Contents

1 Problem 1: Why are corners good for estimating optical flow

1.1 Optical Flow and Motion Perception

Let's look at some basic definitions first:

- OPTICAL FLOW is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene. [1]
- MOTION PERCEPTION is the process of inferring the speed and direction of elements in a scene based on visual inputs.
- MOTION FIELD is the projection of the 3D scene motion into the image [2]

Key assumptions in estimating the optical flow from two subsequent images are [2]:

- BRIGHTNESS CONSTANCY projection of the same point looks the same in every frame
- SMALL MOTION points do not move very far
- SPATIAL COHERENCE points move like their neighbors

1.2 First Order Approximation of Optical Flow

If given two subsequent frames as shown in Figure 1, estimate the apparent motion field $u(x, y)$ and $v(x, y)$ as follows;

According to Brightness Constancy Equation:

$$I(x, y, t-1) = I(x + u(x, y), y + v(x, y), t)$$

Assuming the movement to be small, the image constraint at $I(x, y, t)$ with Taylor series can be developed to get:

$$I(x, y, t-1) \approx I(x, y, t) + I_x u(x, y) + I_y v(x, y)$$

Hence,

$$\boxed{I_x u + I_y v + I_t \approx 0} \tag{1}$$

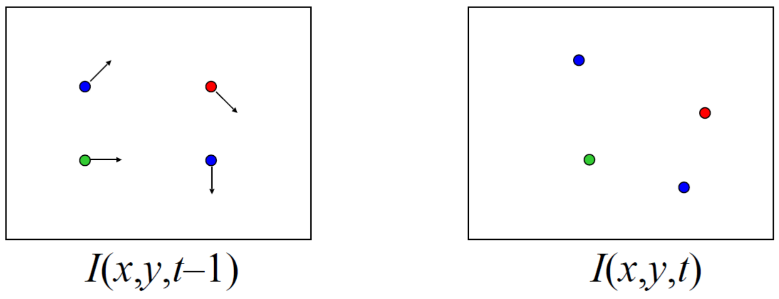

Figure 1: Left: position of 4 points at time $t-1$; Right: position of those same 4 points after displacement at time t

Equation 1 says that there are 2 unknowns and 1 equation per pixel.

This constraint means,
1. The component of the flow in the gradient direction is determined
2. The component of the **flow parallel to an edge is unknown**

1.3 Aperture Problem

Let's look at an example:
Suppose we have an image where $I(x, y) = y$, i.e., image will look like this:
111111111111111111
222222222222222222
333333333333333333
and suppose there is optical flow of (1, 1).

The new image will look like:
-111111111111111111
-222222222222222222

So, $I_t(3, 3) = -1$ and $\bigtriangledown I(3, 3) = (0, 1)$.

Our constraint equation will be

$$0 \approx -1 + < (0, 1), (u, v) >$$

which is $1 = v$. **We recover the v component of the optical flow, but not the u component.**

THIS IS THE APERTURE PROBLEM.

Formal Definition

The motion direction of a contour is ambiguous, because the motion component parallel to the line cannot be inferred based on the visual input.

This means that a variety of contours of different orientations moving at different speeds can cause identical responses in a motion sensitive neuron in the visual system. See Figure 2

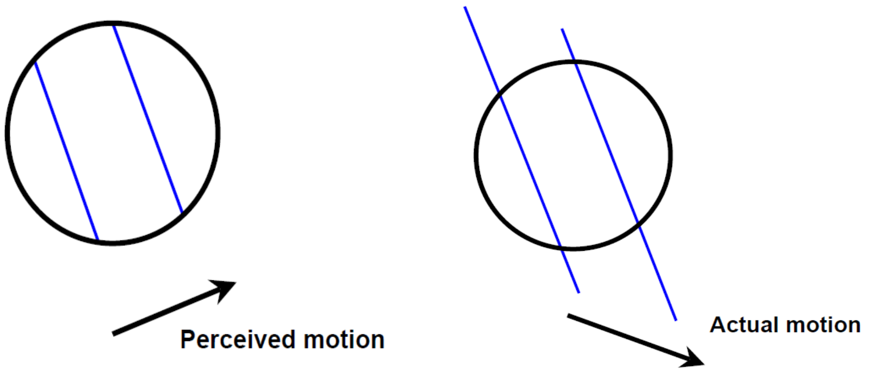

Figure 2: The Aperture Problem where flow perpendicular to the gradient is unknown [2]

1.4 Lucas-Kanade Flow

To solve the aperture problem discussed above, we can pretend that the pixel's neighbors have the same (u, v), we can create a system of equations as shown below:

$$\begin{bmatrix} I_x(p1) & I_y(p1) \\ I_x(p2) & I_y(p2) \\ \vdots & \vdots \\ I_x(pn) & I_y(pn) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p1) \\ I_t(p2) \\ \vdots \\ I_t(pn) \end{bmatrix}$$

We can represent the above equation in terms: $Ad = b$ and can solve d by

$$(A^T A)d = A^T b$$

Therefore,

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

This is solvable when, $\mathbf{A^T A}$ is invertible, is not too small die to noise (its eigenvalues λ_1 and λ_2 should not be too small) and λ_1/λ_2 should not be too large. It is called **second moment matrix**.

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix} = C = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Here, C(1, 1) is sum over a small region, the hypothetical corner. C(1, 2) is the gradient with respect to x, times gradient with respect to y.

1.5 General Case, Corners and Conclusion

If either λ_1 or λ_2 is close to 0, then it is **not** a corner but a flat region.

If λ_1 or λ_2 are big, corners are there and **this is where LUCAS-KANADE WORKS**.

Corners are regions with two different directions of gradient (at least).

Aperture problem disappears at corners.

Because at corners, 1^{st} order approximation fails.

Systems will be "well-conditioned" at corners. System is "solvable" there.

2 Problem 2: Optical Flow Vector Dependencies

We will first see some basic geometry about the 3D points in world space and their corresponding 2D points in the image space:

2.1 Definitions and Formulae

- MOTION FIELD is the projection of the 3D relative vectors onto the 2D image plane.
- MOTION FIELD IS WHAT WE WANT TO KNOW, OPTIC FLOW IS WHAT WE CAN ESTIMATE

For the Motion Field, we will derive an equation relating to 3D scene **velocity** induced by camera motion and its 2D **motion flow field**.

Let's say; at time t, a point's 3D position is at P; at time $t+1$, a point's 3D position will be at $R*P+T$, where R is relative rotation of that point compared to the same point at time t and T is the translation or displacement. Thus, the net 3D displacement will be

$$RP+T-P$$

We can make some general assumptions about the velocity and flow of a 3D point by saying that the displacement from time t to time $t+1$ will be infinitesimal and thus, the rotation angles will be small.

Euler Angle Transformation says

$$R = \begin{bmatrix} 1 & -\psi & \theta \\ \psi & 1 & -\phi \\ -\theta & \phi & 1 \end{bmatrix} = I + \begin{bmatrix} 0 & -\psi & \theta \\ \psi & 0 & -\phi \\ -\theta & \phi & 0 \end{bmatrix} = I + S$$

where ψ is rotation about $Z-axis = \theta_z$; θ is rotation about $Y-axis = \theta_y$ and ϕ is rotation about $X-axis = \theta_x$.

$\therefore$ displacement = $(I+S)P+T-P = SP+T$ and $SP = [\theta_x, \theta_y, \theta_z]^T P$

HENCE, in limit, displacement becomes **velocity**.

$$V = T + \omega P \tag{2}$$

For $T = [T_x, T_y, T_z]; \omega = [\omega_x, \omega_y, \omega_z]; P = [X, Y, Z]$

$$V_x = -T_x + -\omega_y Z + \omega_z Y \tag{3}$$

$$V_y = -T_y + -\omega_z X + \omega_x Z \tag{4}$$

$$V_z = -T_z + -\omega_x Y + \omega_y X \tag{5}$$

The equations 3, 4 and 5 are for relative velocity of a 3D point P with respect to camera as shown in Figure 3

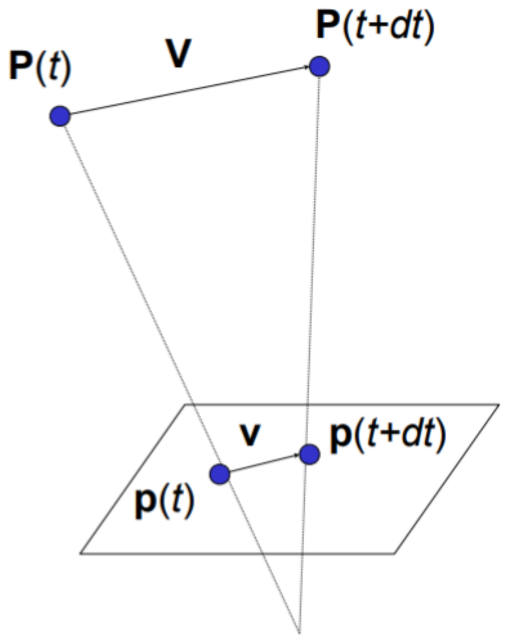

Figure 3: A 2D representation p on an image plane of a 3D point P, while P is under motion in world space[2]

2.2 Flow Vectors and Motion Field

The equations 3, 4 and 5 can be thought as **vectors** of small magnitude of the scene depicted in Figure 3.

In perspective projection,

$$p = \frac{fP}{Z} \tag{6}$$

Taking derivative with respect to time,

$$\frac{dp}{dt} = v = \frac{d\frac{fP}{Z}}{dt} \tag{7}$$

$$\frac{dp}{dt} = v = \frac{f}{Z^2}\left[\frac{dP}{dt}Z - P\frac{dZ}{dt}\right]$$

$$\frac{dp}{dt} = v = \frac{f}{Z^2}\left[V \cdot Z - P \cdot V_z\right]$$

$$p = \frac{fP}{Z} \& P = \frac{pZ}{f}$$

$$\boxed{v = f\frac{V}{Z} - p\frac{V_z}{Z}} \tag{8}$$

v is for 2D projection so $v_z = 0$

From equations 3, 4, 5 and 8:

$$\boxed{v_x = \frac{T_z x - T_x f}{Z} - \omega_y f + \omega_z y + \frac{\omega_x xy}{f} - \frac{\omega_y x^2}{f}} \tag{9}$$

$$\boxed{v_y = \frac{T_z y - T_y f}{Z} + \omega_x f - \omega_z x - \frac{\omega_y xy}{f} + \frac{\omega_x y^2}{f}} \tag{10}$$

THESE ARE THE FINAL EQUATIONS to convert a 3D motion in a world space to 2D points in a image plane with relative depth.

2.3 Angle and Vector Length of the Flow Vectors

In both equations 9 and 10, the first terms **translational components** and all other are **rotational components**.

$\therefore$ it is apparent that for rotational components or the “angle” is independent of the depth Z of a 3D point.

But, the whole vector length depends on depth Z as the translation part has the depth component in it.

3 Problem 3: Markov Assumption for Demosaicing

3.1 (a) a pixel's value vs its left neighbor's value

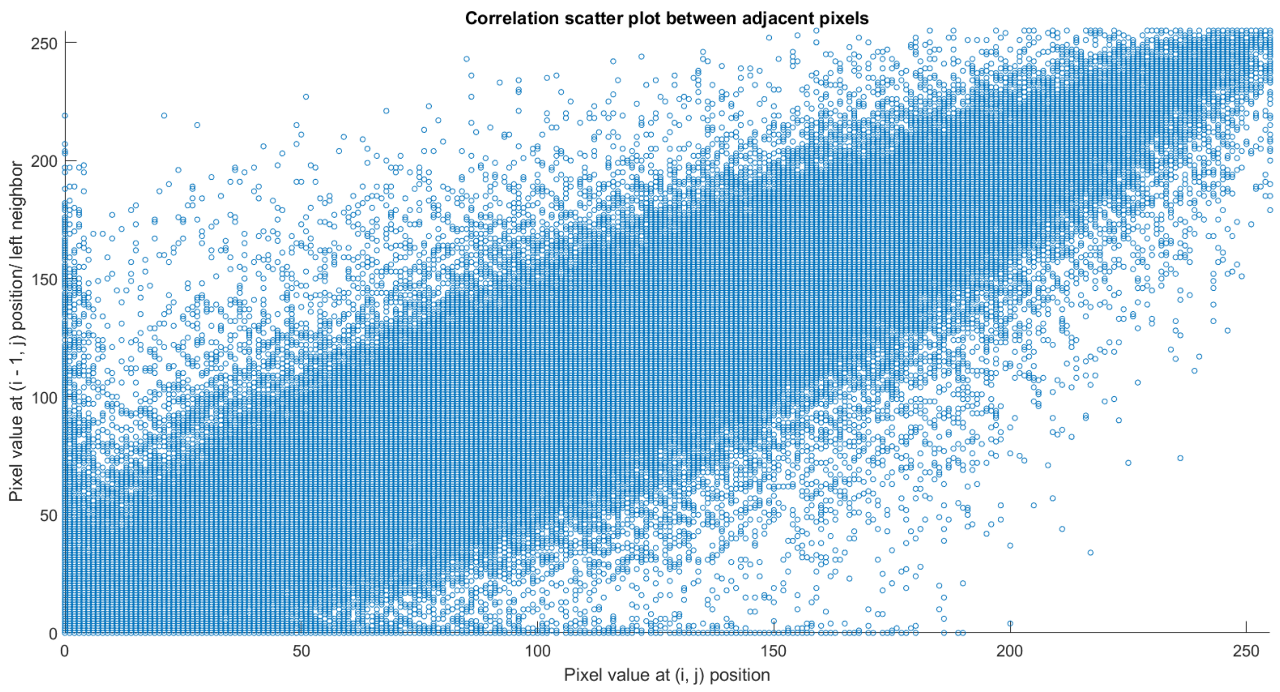

Figure 4: A scatter plot of value of pixel (x, y) versus its left neighbor's value

The higher density along the $y = x$ line proves that a pixel is likely/ probably to have the same or approximately the same value as its neighbor.

There are few outliers to this assumption in the Figure 4 but that's because we have only not take into account boundary of thickness 1 pixel. The more we observe the core part of the image, the more our observation about the pixel value correlation is confirmed.

This property can be used in demosaicing an image with the assumption that an unknown pixel's value will be very probable to have its neighboring pixel's value, which is the idea behind **Nearest Neighbor Interpolation**.

3.2 (b) a pixel's value vs the mean value of its left, right, top and bottom neighbors

If we compare a pixel's value to average of its left, right, top and bottom neighbors, we get a plot like one shown in Figure 5.

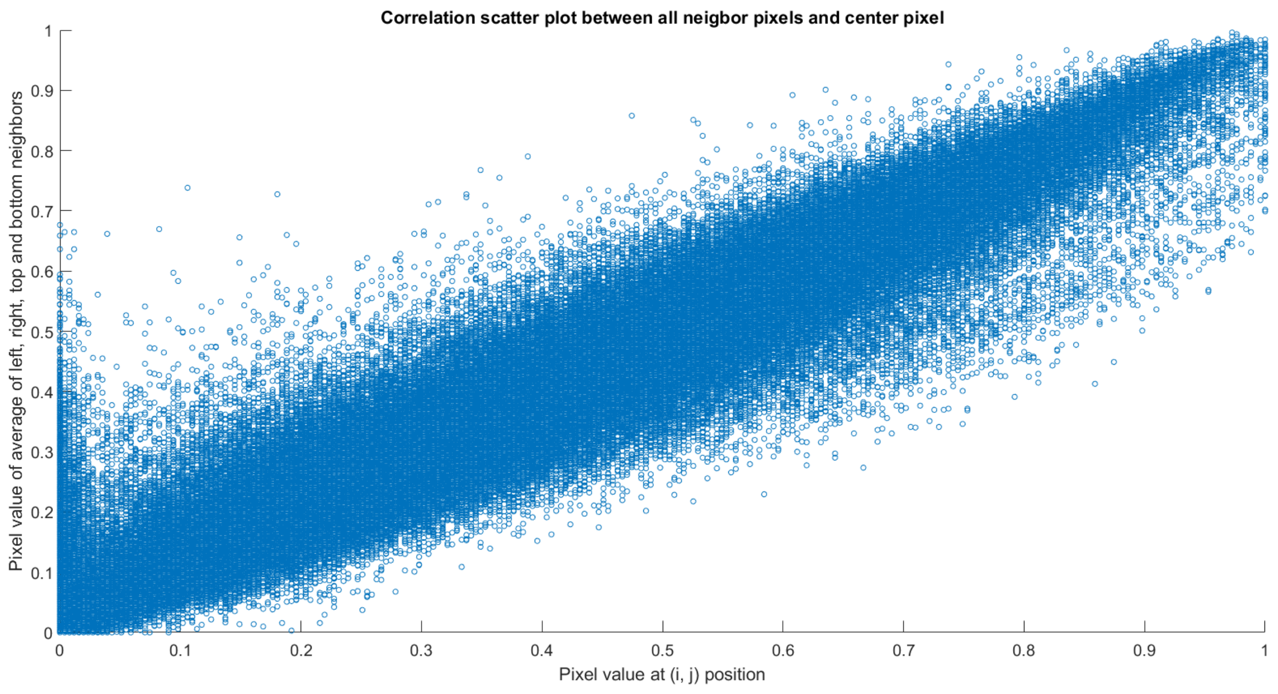

Figure 5: A scatter plot of value of pixel (x, y) versus its left, right, top and bottom neighbor's average value

We can see that in comparison to only comparing a pixel's value to its one neighbor as in Figure 4, the comparison with all neighbors shows the plot points are more congested towards $y = x$ line.

That's because an average of all neighbors will always give better idea about what the center pixel value might be, instead of just looking at a neighbor in one direction.

It might happen that left neighbor is black but all other 3 neighbors are white, making the pixel almost white too.

If we only judge by the black neighbor, we might falsely classify that center pixel to black too.

Therefore, in demosaicing, it is always better to take into account all neighbors whose values are known instead of just picking out the closest one.

The average is more likely to be near the actual value.

Which is the idea behind **Linear Interpolation**.

References

[1] Andrew Burton & John Radford, *Thinking in Perspective: Critical Essays in the Study of Thought Processes* ISBN 978-0-416-85840-2

[2] Subhransu Maji, *Optical Flow* `https://www.dropbox.com/s/9y5gkgycho3iqd7/lec07_optical_flow.pdf?dl=0`