CSc 8830: Computer Vision

Assignment 3: Question - 2

Teja Krishna Cherukuri, 002763290

April 17, 2024

Deriving Motion Tracking Equation

Let $I(x, y, t)$ represent the intensity of a pixel at position (x, y) in frame t. The goal of motion tracking is to estimate the motion vector $\vec{v} = (v_x, v_y)$ that represents how each pixel in the current frame $I(x, y, t)$ moves from its position (x, y) in the previous frame $I(x - v_x, y - v_y, t - 1)$.
We can formulate this problem by considering the relationship between the intensities of corresponding pixels in consecutive frames. Let $I(x, y, t)$ and $I(x - v_x, y - v_y, t - 1)$ be the intensity values of two consecutive frames at positions (x, y) and $(x - v_x, y - v_y)$ respectively. We assume that the intensity values remain constant within a local neighborhood over small time intervals. Then, the brightness constancy assumption can be expressed as:

$$I(x, y, t) \approx I(x - v_x, y - v_y, t - 1)$$

Mathematically, this can be written as:

$$I(x, y, t) - I(x - v_x, y - v_y, t - 1) \approx 0$$

Expanding this equation using Taylor series expansion up to the first-order terms, we get:

$$I(x, y, t) - I(x, y, t - 1) + v_x \frac{\partial I}{\partial x} + v_y \frac{\partial I}{\partial y} \approx 0$$

Where $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ represent the spatial gradients of the image intensity.
Rearranging terms, we obtain the motion tracking equation:

$$v_x \frac{\partial I}{\partial x} + v_y \frac{\partial I}{\partial y} = I(x, y, t - 1) - I(x, y, t)$$

This equation represents the relationship between the motion vector $\vec{v}$ and the spatial gradients of image intensity. By solving this equation, we can estimate the motion vector $\vec{v}$ for each pixel.

Derivation of Lucas-Kanade Algorithm for Affine Motion

We start by considering the motion model for an affine transformation:

$$u(x, y) = a_1 x + b_1 y + c_1$$
$$v(x, y) = a_2 x + b_2 y + c_2$$

where $u(x, y)$ and $v(x, y)$ are the horizontal and vertical motion components respectively, and a_1, b_1, c_1, a_2, b_2, c_2 are the affine motion parameters.

We aim to estimate these parameters to track motion between consecutive frames.
Given a pixel at location (x, y) in the current frame and assuming small motion, we can write the intensity at that pixel in the current frame as:

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

where $I(x, y, t)$ is the intensity at position (x, y) in frame t, $I(x - u(x, y), y - v(x, y), t - 1)$ is the intensity at the corresponding position in the previous frame, and δI represents the change in intensity due to motion.
Applying a first-order Taylor expansion to the intensity function around the point (x, y) yields:

$$I(x - u(x, y), y - v(x, y), t - 1) \approx I(x, y, t - 1) - u(x, y)\frac{\partial I}{\partial x} - v(x, y)\frac{\partial I}{\partial y}$$

where $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ are the spatial gradients of the intensity.
Substituting this approximation into the previous equation, we get:

$$I(x, y, t) = I(x, y, t - 1) - \left(a_1\frac{\partial I}{\partial x} + b_1\frac{\partial I}{\partial y} + c_1\right)x - \left(a_2\frac{\partial I}{\partial x} + b_2\frac{\partial I}{\partial y} + c_2\right)y$$

We aim to minimize the squared error between the left and right sides of this equation over a local neighborhood of the pixel (x, y). We define a neighborhood window centered at (x, y) and consider all pixels within this window.
Let $\mathbf{p} = [a_1, b_1, c_1, a_2, b_2, c_2]^T$ be the parameter vector. Then, the squared error for the i-th pixel within the neighborhood window is:

$$E_i(\mathbf{p}) = \left[I(x_i, y_i, t - 1) - I(x_i, y_i, t) + \left(a_1\frac{\partial I}{\partial x} + b_1\frac{\partial I}{\partial y} + c_1\right)x_i + \left(a_2\frac{\partial I}{\partial x} + b_2\frac{\partial I}{\partial y} + c_2\right)y_i\right]^2$$

We want to find the parameter vector $\mathbf{p}$ that minimizes the total squared error over all pixels in the neighborhood window. This can be expressed as the minimization problem:

$$\min_{\mathbf{p}} \sum_i E_i(\mathbf{p})$$

To solve this optimization problem, we differentiate the error function $E(\mathbf{p})$ with respect to $\mathbf{p}$, set the derivative to zero, and solve for $\mathbf{p}$. This leads to a system of linear equations, which can be solved using techniques like least squares.
Once the optimal parameters $\mathbf{p}$ are found, they represent the affine motion model parameters that best describe the motion between the consecutive frames. These parameters can then be used to compute the motion vectors for each pixel and perform motion tracking under the affine motion model.