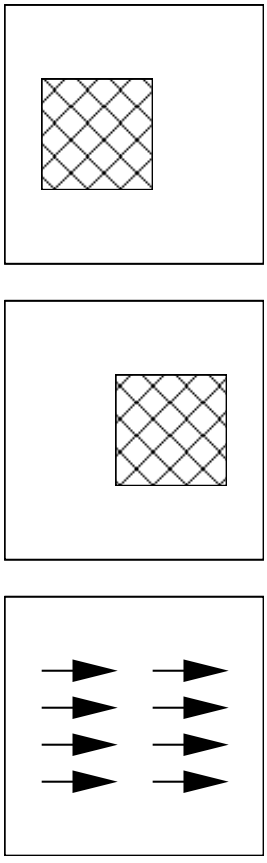
Master of Information and Data Science
DATASCI 281: Computer Vision
Spring 2022

Professor Hany Farid
University of California, Berkeley

Motion Estimation

Our visual world is inherently dynamic. People, cars, dogs, etc. are (usually) moving. These may be large motions, walking across the room, or smaller motions, scratching behind your ear. Our task is to estimate such motions from two or more images taken at different instances in time.



With respect to notation, an image is denoted as $f(x, y)$ and an image sequence is denoted as $f(x(t), y(t), t)$, where $x(t)$ and $y(t)$ are the spatial parameters and t is the temporal parameter. For example, a sequence of N images taken in rapid succession may be represented as $f(x(t), y(t), t+i\Delta t)$ with $i \in [0, N-1]$, and Δt representing the amount of time between image capture (typically on the order of 1/30th of a second). Given such an image sequence, our task is to estimate the amount of motion at each point in the image. For a given instant in space and time, we require an estimate of motion (*velocity*) $\vec{v} = \begin{pmatrix} v_x & v_y \end{pmatrix}$, where v_x and v_y denote the horizontal and vertical components of the velocity vector $\vec{v}$. Shown to the right are a pair of images taken at two moments in time as a textured square is translating uniformly across the image. Also shown

is the corresponding estimate of motion often referred
to as a *flow field*. The flow field consists of a velocity vector at each point in the image (shown of course are only a subset of these vectors).

In order to estimate motion, an assumption of *brightness constancy* is made. That is, it is assumed that as a small surface patch is moving, its brightness value remains unchanged. This constraint can be expressed with the following differential equation:

$$\frac{df(x(t), y(t), t)}{dt} = 0. \tag{1}$$

This constraint holds for each point in space and time. Expanding this constraint according to the chain rule yields:

$$\frac{\partial f}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial f}{\partial t} = 0, \tag{2}$$

where the partials of the spatial parameters x and y with respect to time correspond to the velocity components:

$$f_x v_x + f_y v_y + f_t = 0. \tag{3}$$

The subscripts on the function f denote partial derivatives. Note again that this constraint holds for each point in space and time but that for notational simplicity the spatial/temporal parameters are dropped. This transformed brightness constancy constraint is rewritten by packing together the partial derivatives and velocity components into row and column vectors.

$$\begin{pmatrix} f_x & f_y \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix} + f_t = 0 \tag{4}$$

The space/time derivatives f_x, f_y, and f_t are measured quantities, leaving us with a single constraint in two unknowns (the two components of the velocity vector, $\vec{v}$). The constraint can be solved by assuming that the motion in a small pixel neighborhood is the same. Consider, for example, the nine constraints for a 3×3 pixel neighborhood, yielding the following over-constrained system of linear equations:

$$\begin{pmatrix} f_x(x_1, y_1) & f_y(x_1, y_1) \\ f_x(x_2, y_2) & f_y(x_2, y_2) \\ \vdots & \vdots \\ f_x(x_9, y_9) & f_y(x_9, y_9) \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix} + \begin{pmatrix} f_t(x_1, y_1) \\ f_t(x_2, y_2) \\ \vdots \\ f_t(x_9, y_9) \end{pmatrix} = 0$$

$$A\vec{v} + \vec{t} = 0. \tag{5}$$

We can solve for the veclocity vector $\vec{v}$ by minimizing the following quadratic error function:

$$E(\vec{v}) = \|A\vec{v} + \vec{t}\|^2, \tag{6}$$

The error function can be minimized using least-squares. The error function is differentiated with respect to $\vec{v}$:

$$\frac{dE(\vec{v})}{d\vec{v}} = 2A^T(A\vec{v} + \vec{t}), \tag{7}$$

setting the result equal to zero and solving:

$$\begin{aligned} 2A^T(A\vec{v} + \vec{t}) &= \vec{0} \\ A^TA\vec{v} + A^T\vec{t} &= \vec{0} \\ \vec{v} &= -(A^TA)^{-1}A^T\vec{t}, \end{aligned} \tag{8}$$

where

$$\begin{aligned} A^TA &= \begin{pmatrix} f_x(x_1,y_1) & \dots & f_x(x_9,y_9) \\ \vdots & \dots & \vdots \\ f_y(x_1,y_1) & \dots & f_y(x_9,y_9) \end{pmatrix} \begin{pmatrix} f_x(x_1,y_1) & f_y(x_1,y_1) \\ f_x(x_2,y_2) & f_y(x_2,y_2) \\ \vdots & \vdots \\ f_x(x_9,y_9) & f_y(x_9,y_9) \end{pmatrix} \\ &= \begin{pmatrix} \sum_\Omega f_x^2 & \sum_\Omega f_xf_y \\ \sum_\Omega f_xf_y & \sum_\Omega f_y^2 \end{pmatrix}, \end{aligned} \tag{9}$$

and

$$\begin{aligned} A^T\vec{t} &= \begin{pmatrix} f_x(x_1,y_1) & \dots & f_x(x_9,y_9) \\ \vdots & \dots & \vdots \\ f_y(x_1,y_1) & \dots & f_y(x_9,y_9) \end{pmatrix} \begin{pmatrix} f_t(x_1,y_1) \\ \vdots \\ f_t(x_9,y_9) \end{pmatrix} \\ &= \begin{pmatrix} \sum_\Omega f_xf_t \\ \sum_\Omega f_yf_t \end{pmatrix}. \end{aligned} \tag{10}$$

and where Ω corresponds to the pixel neighborhood over which we are assuming that the motion is constant (a 3×3 pixel neighborhood in the above example).

In order to find a solution, the matrix A^TA must be invertible. Generally speaking this matrix is rank deficient, and hence not invertible, when the intensity variation in a local image neighborhood varies only one-dimensionally

(e.g., $f_x = 0$ or $f_y = 0$) or zero-dimensionally ($f_x = 0$ and $f_y = 0$). These singularities are sometimes referred to as the aperture and blank wall problem. The motion at such points simply cannot be estimated.

Motion estimation then reduces to computing, for each point in space and time, the spatial/temporal derivatives f_x, f_y, and f_t. Of course the temporal derivative requires a minimum of two images, and is typically estimated from between two and seven images. The spatial/temporal derivatives are computed as follows. Given a temporal sequence of N images, the spatial derivatives are computed by first creating a temporally prefiltered image. The spatial derivative in the horizontal direction f_x is estimated by prefiltering this image in the vertical y direction and differentiating in x. Similarly, the spatial derivative in the vertical direction f_y is estimated by prefiltering in the horizontal x direction and differentiating in y. Finally, the temporal derivative is estimated by temporally differentiating the original N images, and prefiltering the result in both the x and y directions. The choice of filters depends on the image sequence length: an N tap pre/derivative filter pair is used for an image sequence of length N.