

Computational Perception

15-485/785

Assignment 4: Motion Perception and Computation

due: Thursday, March 6

Introduction

1 The Motion Gradient Equation

The motion gradient equation relates changes in intensity to changes in motion

$$\begin{aligned}\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} &= 0 \\ I_x u + I_y v + I_t &= 0 \\ \Rightarrow (I_x, I_y) \cdot (u, v) &= -I_t\end{aligned}\tag{1}$$

1. (5 points) Define a function $g(x, y, u, v, t)$ that describes the linear motion of a 2D circular Gaussian with a standard deviation of one. The Gaussian center is defined by (x, y) , and the motion by (u, v) .
2. (5 points) Derive the components of the motion gradient equation for this function and show that the equation is satisfied.

2 Motion Constraint Lines

1. (5 points) Create a moving Gaussian blob using the function `blobmotion`. Use the command `help blobmotion` to learn how to use it. You can use the function `mmplay` to display the motion signal. Write down your choice of parameters for defining the signal. Also explain any rationale you had behind your choice of parameters.
2. (10 points) Write a function called `mge` (motion gradient estimate) that estimates the components of the motion gradient $[I_x, I_y, I_t]$ for each pixel in the movie, for every frame. The size of the components should be identical to that of the movie. The form of the function should be as follows:

```
function [Ix, Iy, It] = mge(M, hx, hy, ht)
%input:
% M - the movie matrix
```

```

% hx, hy - the width and height of each pixel
% ht - the duration of each frame
% output:
% Ix, Iy, It - the estimated motion gradient components

```

3. (10 points) Write a function to plot the motion constraint line at position (x, y, t) in (u, v) space. Be sure that the constraint lines are plotted in the correct units, i.e. they should match the velocity vector `DX` in `blobmotion`. The function should have the following form:

```

function plotmcl(Ix, Iy, It, x, y, t)
%input:
% Ix, Iy, It - the motion gradient components
% x, y, t - the position for which the constraint line is plotted

```

Does the motion constrain line contain the true motion? Be sure to explain what underlies any discrepancies between the true motion and your estimate.

4. (10 points). Next, create a horizontally moving diamond, again using the `blobmotion` function. Use your function `plotmcl` to plot the constraint lines at two points on the upper and lower leading edges.

3 Optical Flow

1. (10 points) Write down the motion constraint equations for the previous problem (the two edges of the moving diamond), and solve for the motion. Explain the conditions under which a solution is possible.
2. (20 points) Generalize the equations in the previous problem to a system of equations in which the motion is estimated from an $r \times r$ region. In most circumstances, the motion gradient equations will not admit an exact solution, and will be overdetermined. Therefore, your equations should express the motion as a least squares problem of the form $Ax = b$, where $x = (u, v)^T$. The value of x for a given region is the one that minimizes the constraint error $\|Ax - b\|$.
3. (20 points) Write a function called `optflow` that uses your derivation in the previous problem to estimate the optical flow at each pixel in the movie. The least squares problem can be solved with the function `pinv`. Your function should have the following form:

```

function [U,V] = optflow(M, hx, hy, ht, r)
%inputs:
% hx, hy - the width and height of each pixel
% ht - the duration of each frame
% r - the size of the region over which the motion
%      estimate for each pixel is computed.
%outputs:
% U, V - the estimated horizontal and vertical motion

```

```
%         components for each pixel in M.  These have the
%         same size as M itself.
```

4. (5 points) Write a function to `plotflow` to plot the optical flow. The matlab function `quiver` is useful for this purpose. Your function should have the form

```
function plotflow(M,i,U,V)
%inputs:
% M - the motion signal
% i - the frame at which to plot the motion
% U,V - the estimated horizontal and vertical motion
%         components for each pixel in M.
```

Test your function on the examples above and show the output. Feel free to create your own examples.