Homework 5
Computer Vision CS 4731, Fall 2011
Due Date: December 1, 2011
Total Points: 22

Note 1: Both analytical problems and programming assignments are due at the beginning of class on December 1, 2011. Please start working on your assignment early and note that there is no credit for late submissions.
Note 2: Read the guidelines for the programming assignments carefully. They are available on CourseWorks at Class Files/Shared Files/Programming Guidelines.
Note 3: It is your responsibility to make sure that the submission includes all the source code and necessary files. You will be graded based on the submission received.

Problem 1: Consider a point in the scene moving with velocity (u, v, w) from the starting point (x_0, y_0, z_0). Write the equation for the image coordinates (x, y) at time t. Show that the image point moves along a straight line as t increases. (3 points)

Problem 2: We discussed the optical flow constraint equation. This contraint states that the optical flow estimates (u, v) at a point (x, y) in image space lie on a straight line whose coefficients are the derivatives of image brightness in the x, y (spatial), and t (temporal) dimensions. Clearly, this contraint does not yield a unique (u, v) solution for each point (x, y). This forced us to develop a computationally intensive iterative scheme. Now consider a structured environment (say, industrial) where illumination can be controlled (changed) at a speed much faster than the motion of objects in the scene. Using the optical flow constraint equation show how two (perhaps unknown) different illuminations can be used to obtain a unique solution for (u, v) at each image point. Hint: Two illuminations provide two constraints rather than one. (4 points)

Programming Assignment

In this programming assignment you are asked to develop an optical flow system. You are given a sequence of 6 images (**flow1.pgm - flow6.pgm**) of a dynamic scene. Your task is to develop an algorithm that computes optical flow estimates at each image point using the 5 pairs (1 & 2, 2 & 3, 3 & 4, 4 & 5, 5 & 6) of consecutive images.

For debugging purposes use the images **simple1.pgm** and **simple2.pgm**. In this synthetic case, the flow field consists of horizontal vectors of the same magnitude (translational motion parallel to the image plane). Note that in the real case, foreshortening effects, occlusions, and reflectance variations (as well as noise) complicate the result.

Optical flow estimates can be computed using the optical flow constraint equation and the iterative algorithm presented in class. For smooth motions, this algorithm should produce robust flow estimates. However, given that the 6 images were taken with fairly large time intervals in between consecutive images, the brightness and temporal derivatives used by the iterative algorithm are expected to be unreliable.

Therefore, you are advised to implement a different (and simpler) optical flow algorithm. Given two consecutive images (say 1 and 2), establish correspondences between points in the two images using correlation (template matching). For each image point in the first image, take a small window (say 7x7) around the point and use it as a template to find the same point in the second image. While searching for the corresponding point in the second image, you can confine the search to a window (say 40x40) around the pixel in the second image that has the same image coordinates as the one in the first image. This 40x40 search window should work since motion between consecutive frames is small; each point moves only a short distance between frames. The center of the 7x7 image window in the second image that is maximally correlated with the 7x7 window in the first image is assumed to be the corresponding point. The vector between two corresponding points is the optical flow (u, v).

There is only one part to this assignment:

a. Write a program **flow** that computes optical flow between two gray-level images, and produces a gray-level output image, showing the optical flow vector field as a "needle map" of a given resolution, overlapped on the first of the two images.

   The needle map should only be computed and drawn on an N x N *grid* (i.e. the optical flow vector field sampled every N pixels along x and y axes). From each grid point draw a line in the direction of the optical flow vector, with length proportional to the vectors magnitude. Draw these "needles" in white (value 255). Show the grid points in black (value 0). You may want to draw a small circle (of radius 1 pixel) around each grid point to make it easily visible.

The program's parameters are as follows:

flow {*input image 1*} {*input image 2*} {*grid*} {*output image*}

You need to choose a value for the grid spacing that gives good results without taking excessively long to compute (as always, set this value in the Makefile and also explain it in the README). (15 points)

Notes:

- The data for the programming assignment can be downloaded from CourseWorks as **hw5data.zip**. This zip file contains the test images and a Makefile which contains targets for building the program (you will need to fill some values, such as the names of your files, thresholds, etc.), a target for testing it out ('make test'), and a target for submitting ('make submit').
- You should also download the vision utilities, available from CourseWorks. This contains useful functions for reading and writing images, accessing image data, and drawing lines. Be sure to copy the files **vision_utilities.c** and **vision_utilities.h** to your working directory so that you can compile and use them in your program.