Project

The objective of this project is for the student to gain experience with extraction of a 2D motion field from a two image sequence and extraction of moving objects from analysis of the motion field.

Requirements

The term program below, refers to a single program, or a set of programs, some of which might be existing tools not created by the student. A program implementing requirements 1-4 below is worth 25 of the 30 points possible. For 30 points, the student should implement requirements 5 and 6 as well.

1. By default, the program will read two grayscale images in .PGM format (magic # is “P2”). (If you find another common format more convenient, then you may use it, provided you are responsible for all image conversions from the test images supplied.)

2. The program will output a color image in .PGM format (magic number “P3”) showing the tail of each motion vector as a blue spot and the head as a red spot and the shaft of the arrow in green. (As above, another common color image format is OK.) See Adam Clark’s output in the colorplates of the S&S text.

3. The program must somehow match neighborhoods across the two input images. The student may implement an interest operator as described in either Ch 9:Alg. 2 or Ch 9:Exercise 3 and motion vector extraction as in Ch 9: Alg. 3. Or, the student may implement normalized neighborhood correlation. Size of image window is to be chosen by the student.

4. A report on the results of processing must be in the standard course format (see Project 1 specs).

5. The program should extract moving objects by motion coherence using tolerances on the magnitude and direction of the motion vectors, their distance from each other, and the number of them in a cluster (TMag, TDir, TDis, TNum).

6. The program will report on detected moving objects, giving the features computed for each object.
Resources

Students may use the following resources.

1. Any programs developed for or used in Projects #1 or #2.
2. Any image tool, such as xv, hips, gimp, MATLAB.
3. Test files provided in the project directory.
4. A function dda.cpp is provided in the project directory; it will return all the coordinates of points on a thin digital line between two given endpoints.

Submission

Program code must be submitted by 11:59 PM Friday, 2 Nov. via email to cse803@cse.msu.edu using only one text file. The text file must have the following sequential organization: (1) a header with your name, date, project title and course title; (2) the text results of your processing on two test sets; (images will be shown in your paper report) (3) the source code of your program. You may submit your "program file" more than once; however, only the last submission will be graded.

Submit your report by Saturday, 3 Nov, 10 AM at EB3134.

Notes

Notes may be added in the future as students ask questions. Look for the file cse803/Projects/Project3/README.txt.

1. The instructor may have new test images available for demonstrations that students will not have previously seen.

2. We have not formally studied methods for clustering, which is part of CSE 802. Consider the following simple $O(N^2)$ method for detecting coherence of motion vectors. Compare each motion vector with every other motion vector and count the number that are close. For that vector having the highest count, go back and mark or delete those close to it so that they will not be processed again. An alternate, but sometimes weaker method, is to work in one dimension at a time, clustering first on $x$ and then on $y$. 