ECSE 6650 Computer Vision Project 3

Ming Jiang

November 26, 2002

1. Introduction

Optical flow is a vector field in the image that represents an approximation of the image motion field. By computing optical flow, we can estimate the motion field from image sequences, which is crucial to understanding the dynamic world. The optical flow of images is obtained from the spatial and temporal variations of the image brightness. This report is organized in the following structure. In section 2, we will discuss the theory of motion analysis using optical flow. In section 3, the experiments and the results will be given. Section 4 will conclude the report with a summary.

2. The theory of motion analysis using optical flow

The first issue in estimating optical flow is how to relate optical flow with the variations of image brightness and. The constancy of the apparent brightness of the observed scene can be written as the stationarity of the image brightness *I* over the time:

$$\frac{dI}{dt} = 0 \qquad (1)$$

This constraint is completely satisfied only under 1) the motion is translational motion; 2) illumination direction is parallel to the angular velocity for lambertian surface.

Since I is a function of (x,y), which, in turn, are function of time t, I(x(t),y(t), t), by using chain rule we have

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial x} \qquad (2)$$

where $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ represent spatial intensity gradient while $\frac{dI}{dt}$ represents temporal intensity gradient.

By a simple transform, we derive the following image brightness constancy equation

$$(\nabla I)^t + I_t = 0 \qquad (3)$$

Where ∇I is called image intensity gradient. This equation may be used to estimate the motion filed v. One limit of this equation is that it can only determine motion flow component along the direction of the intensity gradient, because the constraint on v is lost when the velocity is orthogonal to intensity gradient. In this sense, we can say that optical flow is the projection of motion field in the gradient direction.

To estimate optical flow, we need an additional constraint since equation (3) only provides one equation for 2 unknowns. For each image point p and an nxn neighborhood R, where p is the center, assume every point in the neighborhood has the same optical flow v, we have

$$\nabla^t I(x,y)v(x,y) + I_t(x,y) = 0 \qquad (x,y) \in R \qquad (4)$$

Where v(x,y) can be estimated via:

$$\varepsilon^2 = \sum_{(x,y)\in R} (\nabla^t I(x,y)v(x,y) + I_t(x,y))^2 \qquad (5)$$

The least square solution to v(x, y) is

$$v(x,y) = (A^t A)^{-1} A^t b\,. \qquad (6)$$

Where

$$A = \begin{bmatrix} \nabla^t I(x_1,y_1) \\ \nabla^t I(x_2,y_2) \\ \vdots \\ \nabla^t I(x_3,y_3) \end{bmatrix} \qquad b = -[I_t(x_1,y_1), I_t(x_2,y_2), I_t(x_3,y_3), I_t(x_4,y_4)]^t \quad (7)$$

Besides assuming brightness constancy while objects are in motion, we can assume smoothness constraint on the motion field, i.e., motion field projections in *x*, *y*, and *t* remain the same for a small neighborhood. Mathematically, these constraints can be formulated as follows:

$$\frac{d^2 I}{dtdx} = 0 \quad \frac{d^2 I}{dtdy} = 0 \quad \frac{d^2 I}{dtdt} = 0 \qquad (8)$$

Applying them to equation (2) yields three additional optical constraints:

$$v_x I_{xx} + v_y I_{yx} + I_{tx} = 0$$
$$v_x I_{xy} + v_y I_{yy} + I_{ty} = 0$$

$$v_x I_{xt} + v_y I_{yt} + I_{tt} = 0$$

Then we have four optical constraints totally:

$$\begin{aligned} v_x I_x + v_y I_y + I_t &= 0 \\ v_x I_{xx} + v_y I_{yx} + I_{tx} &= 0 \\ v_x I_{xy} + v_y I_{yy} + I_{ty} &= 0 \\ v_x I_{xy} + v_y I_{yy} + I_{ty} &= 0 \end{aligned} \qquad (9)$$

Then v=v(x,y) can be solved by minimizing

$$\| Av - b \|^2 \qquad (10)$$

Where

$$A = \begin{bmatrix} I_x & I_y \\ I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \\ I_{tx} & I_{ty} \end{bmatrix} \quad b = -\begin{bmatrix} I_t \\ I_{xt} \\ I_{yt} \\ I_{tt} \end{bmatrix} \qquad (11)$$

$$v(x, y) = (A^t A)^{-1} A^t b \qquad (12)$$

The next issue is how to compute image derivatives.

The cubic facet model is used to compute the image derivative analytically. Compared with traditional approach to compute intensity derivatives with numerical approximation of continuous differentiations, this approach is more robust and accurate.

Assume the gray level pattern of each small block in an image sequence is ideally a canonical 3D cubic polynomial of x, y, t:

$$I(x,y,t) = a_1 + a_2 x + a_3 y + a_4 t + a_5 x^2 + a_6 xy + a_7 y^2 + a_8 yt + a_9 t^2 + a_{10} xt + a_{11} x^3 + a_{12} x^2 y + a_{13} xy^2 + a_{14} y^3 + a_{15} y^2 t + a_{16} yt^2 + a_{17} t^3 + a_{18} x^2 t + a_{19} xt^2 + a_{20} xyt \qquad (x,y,t) \in R \qquad (13)$$

The solutions for coefficients a_i in the least square sense minimizes $\| Da - J \|^2$ and is expressed by

$$a = (D^t D)^{-1} D^t J \qquad (14)$$

Where

$$D = \begin{bmatrix} 1 & x_1 & y_1 & t_1 & \cdots & x_1 y_1 z_1 \\ 1 & x_2 & y_2 & t_2 & \cdots & x_2 y_2 z_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & t_N & \cdots & x_N y_N z_N \end{bmatrix} \qquad J = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ \vdots \\ I_N \end{bmatrix} \qquad (15)$$

I_n is the intensity value at (x,y,t).

Image derivatives are readily available from the cubic facet model. Substituting elements of A into (11) gives

$$A = \begin{bmatrix} a2 & a3 \\ 2a5 & a6 \\ a6 & 2a7 \\ a10 & a8 \end{bmatrix} \quad b = -\begin{bmatrix} a4 \\ a10 \\ a8 \\ 2a9 \end{bmatrix} \qquad (16)$$

The above algorithm can be summarized as the following steps:

1. Select an image as central frame (the 3rd frame if 5 frames are used)
2. For each pixel (excluding the boundary pixels) in the central frame:
   - Perform a cubic facet model fitting using equation (13) and obtain the 20 coefficients using equation (14).
   - Derive image derivatives using the coefficients and the A matrix and b vector using equation (16).
   - Compute image flow using equation (12).
   - Mark each point with an arrow indicate its flow if its flow magnitude is larger than a threshold.

In practical calculation of optical flow, the optical flow vectors should be set to zeros when matrix $A^t A$ is singular.

3. Experiments and results

With the above algorithm, I have calculated the optical flow on the three given image sequences.

In doing the experiments, I found that results did get better if optical flow vector is set to zero when $A^t A$ is rank deficient.

However, there is still much noise to be handled so that the optical flow can be displayed in meaningful ways. After checking the magnitude of the detected velocity, I set a certain threshold value so that all the noisy impulsive optical flow vectors are filtered out. The noise now left in the optical flow is noisy in sense of direction. The direction of the noisy vectors has significant deviation from the overall direction of the image or the velocity direction a local region in the image. This kind of noise can be further attenuated with convolution or median filtering on the directions of the optical flow vectors.

I also noticed that noise in the original image is just so strong that using a large window in facet fitting doesn't yield optical flow vectors with much lower noise.

The optical flow of the central frame of each sequence is displayed in the following figures.

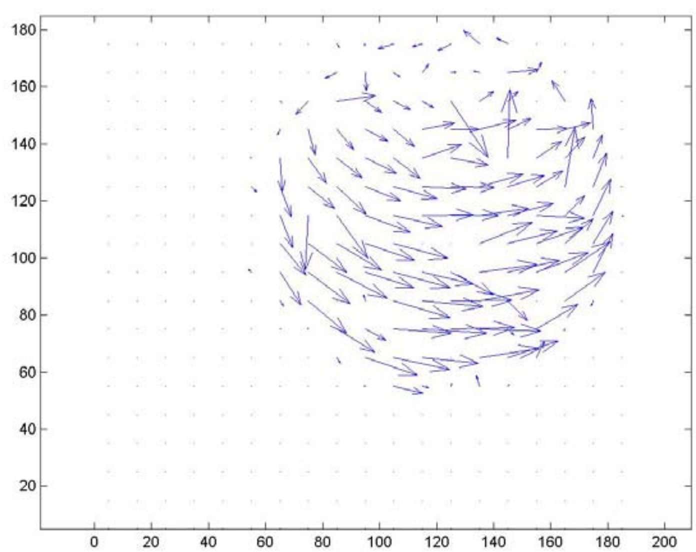

Fig 1: optical flow of frame 3 of sequence 1; motion direction: the direction follows right rule with the upward thumb as the rotating axis

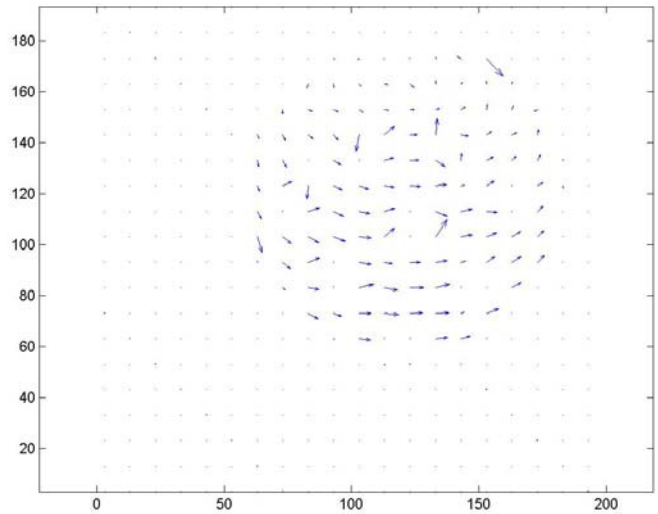

Fig 2: optical flow of frame 3 of sequence 1 with a different scale from Fig 1.

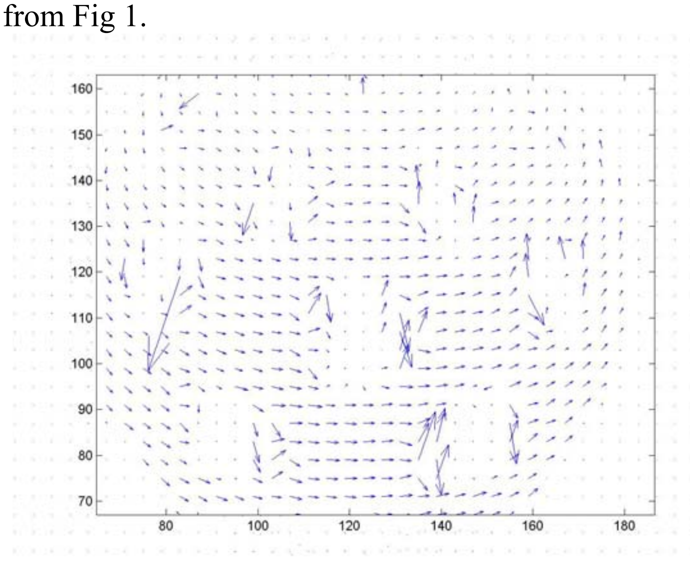

Fig 3: optical flow of frame 3 of sequence 1 in a closer look without filtering out the flow vectors with large magnitude.

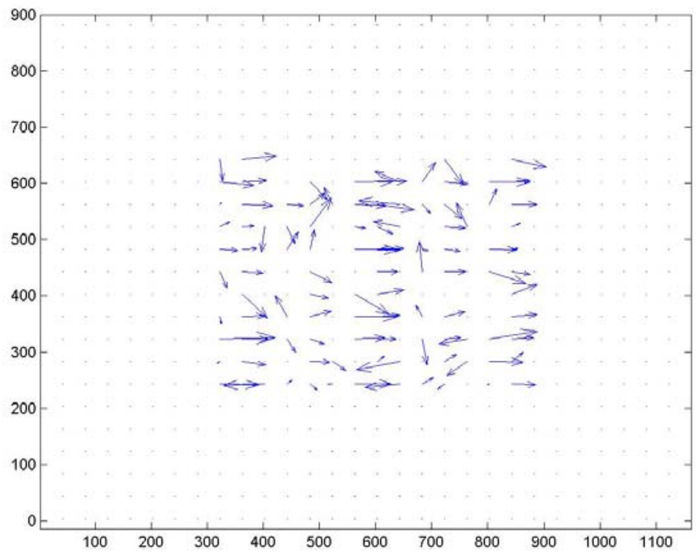

Fig 4: optical flow of frame 3 of sequence 2; motion direction: move to the right along horizontal direction of the image

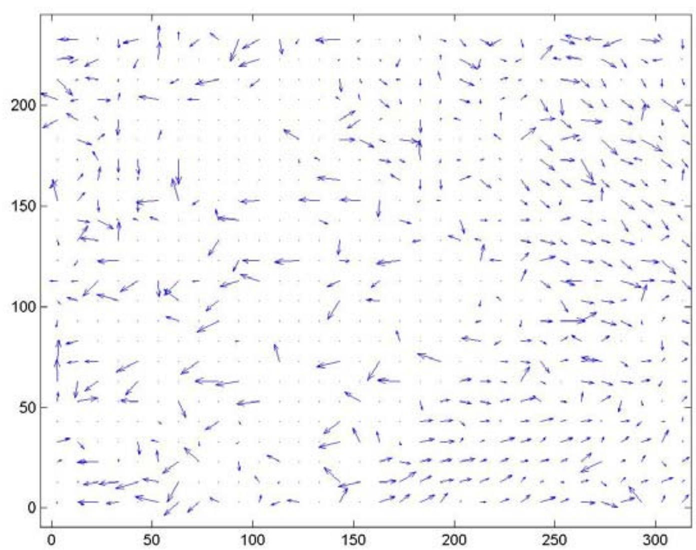

Fig 5: optical flow of frame 3 of sequence 3; motion direction: the right person is moving to the right and the left person is moving to the left and with his head moving away from the camera

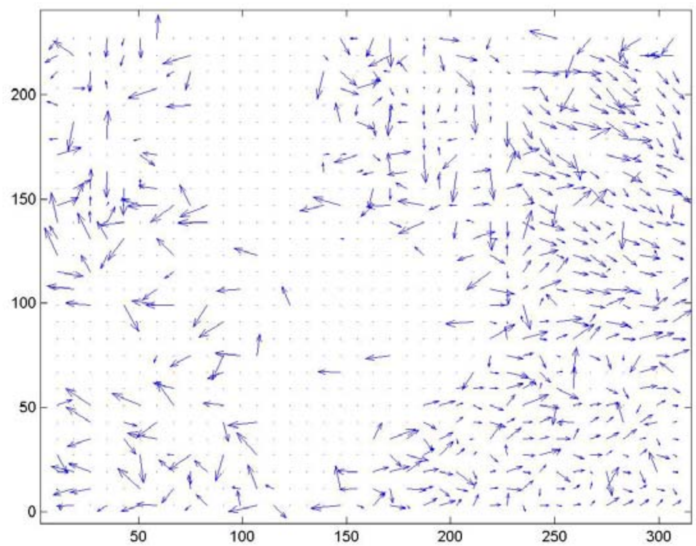

Fig 6 optical flow of frame 3 of sequence 3 with a better display

4. Summary and conclusion

In this project, I have implemented the algorithm to estimate the optical flow for image sequences. The computed optical flow of sequence 1 (rotating sphere) and sequence 2 (moving rectangles) gives satisfying results, indicating clearly the moving direction of the objects in the images. The calculation of optical flow on the third sequence is not satisfying, due to the complexity of the images and the noise.
The optical flow estimation can be improved if the images are preprocessed to filter out the noise. In the above implementation, some noise is filtered out after the computing of the optical flow. This can be improved by de-noising the image before hand. Besides, the optical flow can be improved also by filtering out the vectors with different directions from the principle direction of the motion in the images. Of course, this issue becomes complicated for images including multiple objects.

Code:
Main program
PRJ3seq1
PRJ3seq2
PRJ3seq3
Output module
Mydrawoutput1
Mydrawoutput2
Mydrawoutput3
Core subroutine