6.869 Advances in Computer Vision
Andrew Moran
Collaborators: None
Nov 20, 2014 (1:00pm)
Problem Set 8

Problem Set 8

Resources with corresponding images and code are on Stellar under `andrewmo@mit.edu`. The files are in the `pset8.zip` folder. To reproduce the below figures, run `demoMotion.m` and `makeMovie.m`.

Problem 8.1

For this problem, we implement the Lucas-Kanade (LK) optical flow algorithm. We will do so using the coarse-to-fine approach, which will be explained in the next section.

(A) Given two images I_1 and I_2 as well as the initial flow $(u0, v0)$, we can compute the optical flow (u, v) from image I_2 to I_1 using Lucas-Kanade. With $I_x(q_i)$, $I_y(q_i)$, $I_t(q_i)$ representing the partial derviatives of image I with respect to x, y, and t; the solution can be viewed as solving the following 2 by 2 system of equations as shown below:

$$\begin{bmatrix} du \\ dv \end{bmatrix} = \begin{bmatrix} \sum\limits_i I_x(q_i)^2 & \sum\limits_i I_x(q_i) I_y(q_i) \\ \sum\limits_i I_y(q_i) I_x(q_i) & \sum\limits_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum\limits_i I_x(q_i) I_t(q_i) \\ -\sum\limits_i I_y(q_i) I_t(q_i) \end{bmatrix}$$

The varaiable q_i is the nieghborhood of pixels in a specified patch. The derivates were calcuated as followed:

```
[Ix, Iy] = gradient(I2);
It = warpI2 - I1;
```

Using the vectorized version, you can evaluate the velocity vector (du, dv) explicity. Below is some code that solves for (u, v) in a vectorized manner.

```
det = Ixx.*Iyy - Ixy.*Ixy;
u = (-Iyy.*Ixt + Ixy.*Iyt )./det;
v = ( Ixt.*Ixy - Ixx.*Iyt )./det;
```

warpI2 is the warped I2 image using the updated flow field on each iteration. We iterate multiple times in order to get a better approximatin of the optical flow field.

(B) Using the Lucas-Kanade implementation, we can apply the coarse-to-fine algorithm (which utilizes Gaussian pyramids) to get the desired results. This is due to the fact that Lucas-Kanade is a greedy algorithm that converges to the local minimum. Below is reeated psuedocode on how the algorithm works:

Algorithm 1 Coarse-to-Fine-LK(I_1, I_2, k)

1. Build k-level Gaussian pyramids G_1, G_2 for I_1, I_2
2. Find the optical flow field (u_k, v_k) from G_2^k to G_1^k at the coarsest pyramid level k using the Lucas-Kanade algorithm
3. Upsample the flow field for level $k-1$, and transform G_2^{k-1} towards G_1^{k-1} using (u_{k-1}, v_{k-1})
4. Update the optical flow estimation (u_{k-1}, v_{k-1}) at level $k-1$
5. Repeat $3-4$ for levels $k-2, k-3, ..., 1$

(B) Below are the results after running `demoMotion.m`:

Figure 1: Original sequence of images for car (left) frame 1 of `car1.jpg` and (right) frame 2 of `car2.jpg`.

Figure 2: Results after running CoarsetoFine LR with Median Filtering. (left) `frame1.png` and (right) `warped_frame2.png`.

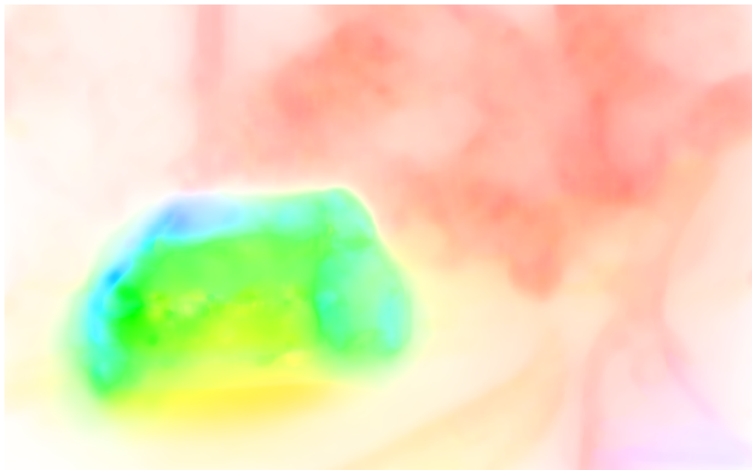

Figure 3: Visualization of the optical flow field (`flow_field.png`).

Problem 8.2

For this problem, we would like to extend our Coarse-to-Fine with Lucas-Kanade (LK) optical flow algorithm to undergoe motion magnification.

(A) Using I as the input video sequence, α as the scaled gain, and τ as the threshold; the Simple Motion Magnification pseudocode is defined as follows:

Algorithm 2 Simple-Motion-Magniication(I, α, τ)

1. Estimate the motion (u_t, v_t) between frame I_1 and each frame I_t in I.
2. Modify the flow fields according to

$$(u_t^{'}, v_t^{'})_p = \begin{cases} \alpha(u_t, v_t)_p, & ||(u_t, v_t)_p|| \leq \tau \\ (u_t, v_t)_p, & \text{otherwise} \end{cases}$$

3. Generate the motion-magnified sequence by warping I_1 according to $(u_t^{'}, v_t^{'})$ for each t.

Figure 4 shows the results of magnifying the motion in the `/bookshelf` driectory. This was done with $\alpha = 10$, $\tau = 2$. There are some noticeable artifacts near the hand and arm, however, most flow is correctly represented in the attached video `bookshelfMagMovie.avi`

(B) Optional

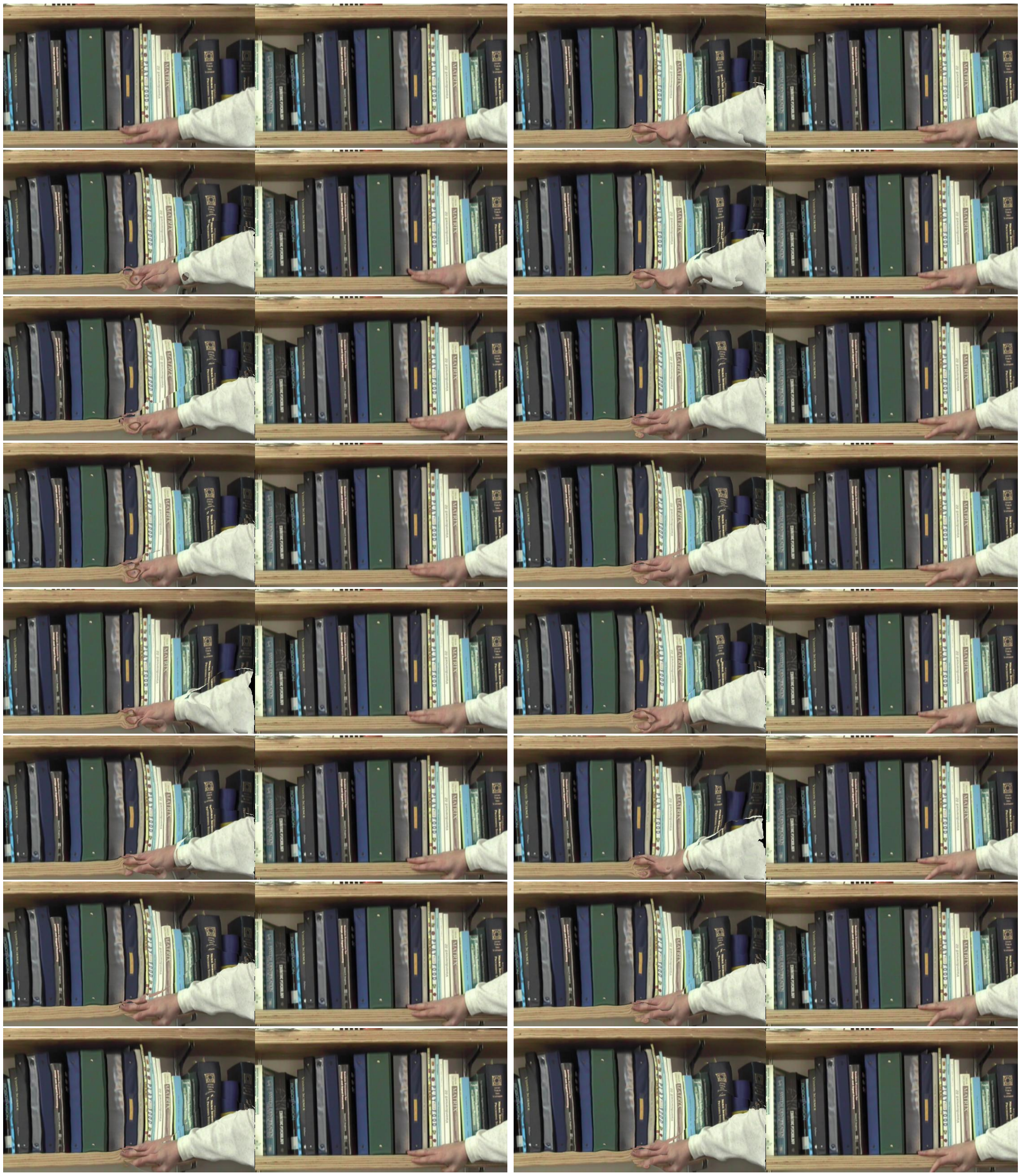

Figure 4: Results after `makeMovie.m`. We display the odd frames out of the original 44. Side by side, we display the magnified frame in the left half of the movie and the original in the right half. Left column displays frames 1 to 15, right column displays frames 17 to 31.