Benjamin Shih
16720f13 Computer Vision
Assignment 4
Motion Tracking

Section 1: Lucas-Kanade Tracking

Q1.1 A^TA is the 2x2 Hessian matrix, also known as the structure tensor and second-moment matrix in other contexts. It is a matrix based on the gradients of a function by summarizing the principal directions of the gradient at a specific point in the image.

A^TA must be invertible in order to solve the least squares optimization. In addition, the eigenvalues of the Hessian should not be too small in order to prevent singularities that result from when at least one eigenvalue is 0 (singularity also means that the matrix is not invertible). Lastly, the ratio of the larger eigenvalue to the smaller eigenvalue should not be exceedingly large, such that the small eigenvalue is very small in comparison for similar reasons.

Q1.2 I used the Inverse Compositional Algorithm to estimate the optimal local motion between two frames. This involves precomputing the gradient of the template, the Jacobian of the warp at a given pixel (which results in the 2x2 identity matrix in this case because we make the assumption that the image differs only by a pure translation), the steepest descent images, and the Hessian matrix. We iterate upon the next frame warped to the current template, the error image, the optimized local motion, and the update of the position until an arbitrary loop threshold or an arbitrary convergence of the estimate.

Related files: LucasKanade.m

Q1.3 To track the car using the above Lucas-Kanade implementation, I iterate through every pair of consecutive frames and determine the optimal local motion which is saved and displayed on top of the current image. This results in a bounding box (synonymous with “template”) that tracks the movement of the car through the sequence of provided images.

Related files: testCarSequence.m

Q1.4 There tracker may not be accurate when its assumptions are violated. Lucas-Kanade assumes that the optical flow at a pixel is essentially constant in a local neighbourhood between consecutive frames. This is not necessarily true when we run into problems like moving cameras, unstable frames, or shadows. These problems can be classified into issues regarding illumination and affine transformation, two parameters that we will attempt to deal with in the following sections. In addition, for a parameter like illumination, we can try methods such as Horn-Schunck which introduces a global constraint of smoothness in order to solve the aperture problem.

Section 2: Lucas-Kanade Tracking with Appearance Basis

Q2.1 To optimize equation 2 over **w** in terms of two consecutive frames and the appearance basis, we start with the formula provided by equation 3 of the homework. Equation 3 is an extension of the equation that we previous optimized over in section 1, with the addition of a given set of the book's basis images that trained to correspond to principal components of the book template.

$$\min_{u,v,\mathbf{w}} J(u,v,\mathbf{w}) = \sum_{(x,y)\in R_t} (I_{t+1}(x+u,y+v) - I_t(x,y) - [\sum_c w_c B_c](x,y))^2 \qquad (3)$$

The equation can be rewritten as $[I_x, I_y, B_1, \ldots, B_{10}]*[\Delta u, \Delta v, w_1, \ldots, w_{10}]'$, where u and v are cumulative but $w_1, \ldots, w_{10}$ replace their previous value (because the weight of a specific book appearance is for that specific instance in time), and then we can apply the same optimization technique in order to find the local motion as well as the respective weights for the eigen-books (synonymous with "basis") by solving $[\Delta u, \Delta v, w_1, \ldots, w_{10}]' = (A^T A) \backslash (A^T * \Delta I)$.

Q2.2

The process to account for the appearance basis is very similar to the approach that was initially used to find the optimal local motion. As described in Q2.1, I accounted for both the weight and translation optimizations simultaneously by rearranging the translational and basis weight vectors into one vector and optimizing over the entire vector to get both the optimal local motion and the optimal basis weights between the two frames.

Related files: LucasKanadeBasis.m, carPosition.mat

Q2.3 The process for tracking the book is the same as the previous process for tracking the car. However, the performance in the book example suffers because there are some non-translational transformations that occur while the book is moving. Even if the book returns to the original viewing plane which we started tracking from, the algorithm will not be able to retain its accuracy because the local motion estimates are cumulative and thus the warp will unintentionally continue to account for the non-translational transformations.

Related files: testBookSequence.m, bookPosition.mat

Section 3: Dominant Motion Estimation

Q3.1 I extend the simplified, translational Lucas-Kanade tracker to account for an affine model of flow by estimating a 6x1 vector of affine flow parameters (corresponding to matrix M), rather than a 2x1 vector of translational flow parameters. In addition, because I am interested in estimating the dominant motion and removing it from the image, I use the entire frame as a template and threshold away the flow that I am not interested in, rather than using a small box as the template and retaining all of the movement. A side effect of using the entire image as the template is the matrix of image

derivatives A and temporal derivatives ΔI are only meaningful in the overlapping regions of It and the warped It1. The general process of the affine flow estimation remains the same, including using the pseudo-inverse to find the least squares method optimization.

Related files: LucasKanadeAffine.m

Q3.2 To incorporate the dominant motion estimation, I warp the current frame to the next frame using M, and subtract it from the next frame to find the error image. Next, I arbitrarily threshold the absolute difference in order to partition dominant and non-dominant motion. This allows me to conclude what is background motion and what is an actual moving object. Hysteresis thresholding helps with this process it allows for two input values - in addition to retaining values above the high threshold and tossing out values below the low threshold, it retains values in between the two thresholds only if there is a valid high-threshold edge nearby.

Related files: SubtractDominantMotion.m

Q3.3 In order to improve the performance of the estimate for the moving image following the affine registration step, I incorporated dilations and erosions of the image in order to reduce the effect of noise. I also applied median filters in order to reduce salt and pepper noise from the images.

Related files: SubtractDominantMotion.m