Feature tracking in videos using WSSD and Kalman's Approach

1. Introduction

There are many real-world applications based on tracking objects over a sequence of images. A video is a set of frames captured over a sequence of time. To track objects in a video, one can simply keep tracking objects between subsequent images. This idea led to many implementation techniques. The techniques used in this assignment are Autocorrelation based on the weighted sum of squared differences and Kalman Feature tracking. For tracking objects using these techniques, good features are first extracted. Points on the images are taken to be features(called feature points). These feature points help in matching features among images.

2. Algorithms

2.1. Feature Detection:

For tracking features in a sequence of images, we need to detect good feature points are worthy of tracking. To do this, an equation has been formulated as- $\sum_i w(x_i)[I_1(x_i+u)-I_0(x_i)]^2$ where I1 is the second image, I0 is the first image, u is the displacement vector and w(x) is the weighted kernel. This equation has been reformulated using the Taylor series and the final equation is formulated as u^TAu. Here A is a matrix of the form [[I_x^2 I_xI_Y]
[I_xI_y I_y^2]].

The next step is to find the minimum Eigen value of this matrix for each pixel and picking out top K values as the feature points.

2.2. Weighted Sum of Squared Differences:

The next step after obtaining features in the first image, we need to perform feature tracking. Algorithm of this technique is as follows:

- We assume the displacement of the feature point in the new image to be within a window of a certain size.
- For each point within this window in the new image, we inturn take a small sliding window and compute the weighted sum of squared differences using the first equation given in section 2.1.
  - We choose the pixel within the bigger window with the minimum weighted sum. This pixel is the new feature point in the new image.
  - This process is repeated for all the feature points in the image.
- This process is repeated for all the images in the video sequence.

2.2. Kalman Tracking:

This technique considers the objects in states. Each state represents the location, velocity and acceleration of pixels. These states change with time. This approach constantly updates the states of the feature points by some matrix calculations. Forward update and backward update are done sequentially. The formulae are as follow:

Forward Update:

$$S_{t+1}^{-} = AS_t$$
$$P_{t+1}^{-} = AP_tA^T + Q$$

Backward Update:

$$K=P_{t+1}^{-}H^T(HP^{-}H^T+R)^{-1}$$
$$S_{t+1}^{-} = S_{t+1}^{-}+K(m_{t+1}-HS_{t+1}^{-})$$
$$P_{t+1}=(I-KH)P_{t+1}^{-}$$

Where S denotes the states, P represents the covariance of uncertainty and m represents the measurement noise(m_t=HS_t+w, w is the zero mean gaussian covariance of A^{-1}).

3. Implementation

In this assignment, feature detection is performed on the first image using the matrix A given in section 2.1. To obtain these values, first-order and zero-order Gaussian convolutions must be performed on the image.

$$I_x=g_x(x)\ g(y) * I(x,y)$$
$$I_y=g(x)\ g_y(y) * I(x,y)$$

Where g_x and g_y are the first-order Gaussian derivatives and g(x) and g(y) are the zero-order gaussian derivatives. By using these formulae, we get A matrices for each pixel in the image. Minimum Eigen values are found out and top K pixels are the feature points. Fig1. is the first frame in a video of the moon. After performing the feature detection as explained above, the top 8 feature points are obtained as shown in Fig2.

Fig1. First image.

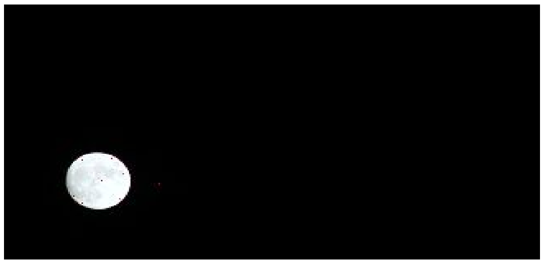
Fig2. First image with feature points.

The next step to iterate over all the images and track features. For calculating the weighted sum of squared differences, a bigger window of size 9x9 is chosen in this project. A smaller sliding window of size 5x5 is used around every pixel within the bigger window. The minimum of these computed weighted sums is the new corresponding feature point. This is continued for all the feature points and repeated for all the images. Fig 3. shows the second image with new feature points. Fig 4. shows the 166th image in the video sequence.

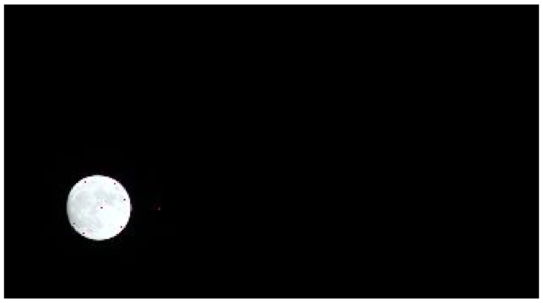
Fig3. Second image with new feature points.

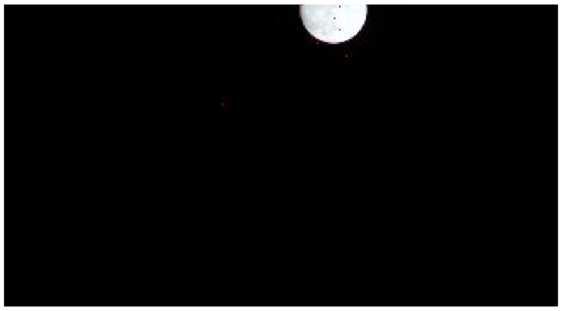
Fig4. 166th image with feature points.

Another technique used in this project is Kalman's approach. As explained in section 2.3, states and covariance matrices are constantly updated using the respective equations. The states updated in the present iteration are sent as parameters for the next iteration. A, H, R and Q matrices are predefined and are fixed. S_{t+1}^{-} and P_{t+1}^{-} are the matrices that change for each feature point. The first and second elements in the new state give the new x and y locations of the feature points. Fig5. and Fig6. show the second and 166th images obtained with the help of Kalman's approach.

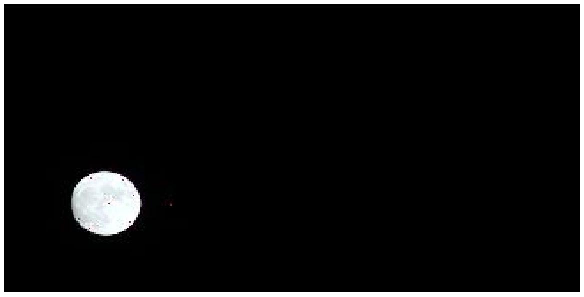
Fig5. Second image after Kalman tracking.

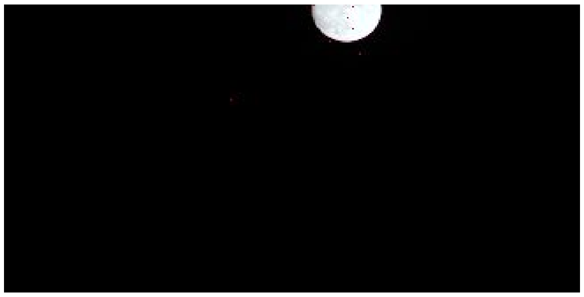
Fig6. 166th image after Kalman tracking.

4. Conclusion

After performing these techniques on the same data set, it can be observed that Kalman's tracking is easier to implement and faster than the WSSD technique. As the WSSD technique requires computing the sums of all the pixels in the window, it takes up a lot of time. In Kalman's tracking, as most of the operations are on matrices, it takes up less code space and completes execution in less amount of time. Results are almost similar for both techniques but in the broader spectrum, Kalman's approach gives better results than WSSD.

5. Issues

Although these two approaches are smooth with feature tracking in images, it can be one hectic of a task when it comes to boundary checking. As a feature goes near boundaries, the window goes out of bounds for which boundary checks had to be performed whenever a window is used.

Sai Srinija Sakinala
U49418724