Camera Motion Estimation by Feature Masking and Kalman Filter for Video Stabilization

May 21, 2018
EE374 DSP2

Jay Neil Gapuz
EEEI University of the Philippines- Diliman

Introduction

Not smooth camera movement may result to unsteady image sequences which create a shaky or jerky output.

Types of Video Stabilization:

- Mechanical/Optical (Motion sensor, gyroscopes)
- Post Image Processing

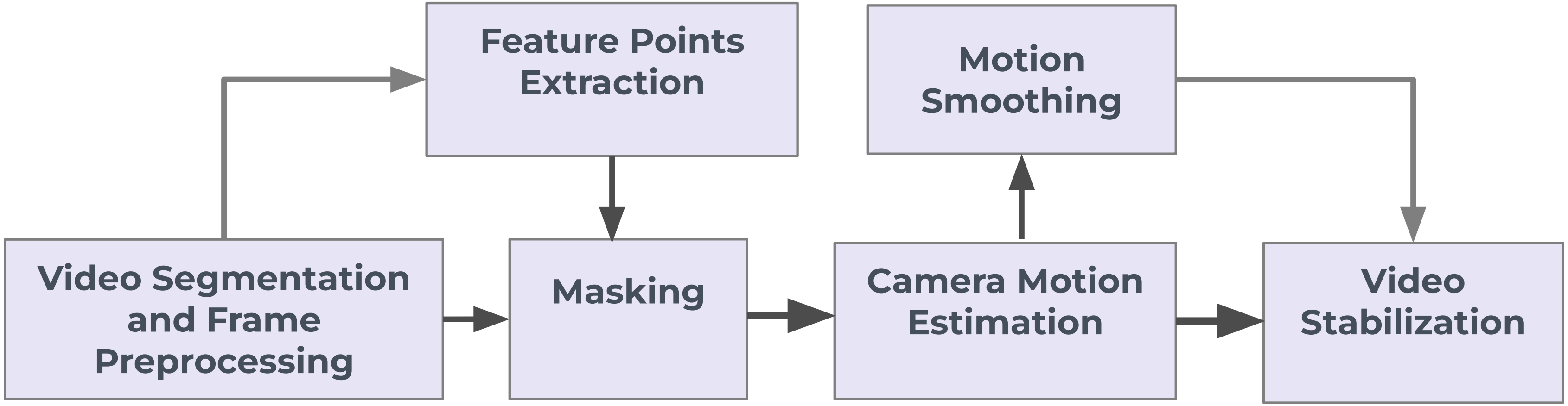
Post Processing Video Stabilization

- Comprised of three major steps:

Proposed System and Algorithm

Introduce a simple and efficient post processing algorithm to track camera motion and filter the unwanted translation

Block Diagram



Segmentation and Preprocessing

(1) Segmentation includes analyzing the frame rate and separating image sequences into blocks/frames

(2) Grayscale image transformation and **Image normalization**
- eliminates the problem of frame ambiguity and motion blur when each succeeding frames suddenly differs in brightness level.

Feature Extraction and Masking

(1) Masking the frame with the same video dimension

(2) Passing the image with an **edge detector. Use Sobel operator** (performs a 2-D spatial gradient measurement on images) **then apply thresholding.**

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Feature Extraction and Masking

(3) Feature points selection using the Shi-Tomasi Good Features to Track. It finds the difference in intensity for a displacement(u,v) in all directions. It finds the difference in intensity for a displacement(u,v) in all directions.

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

$$R = \min(\lambda_1, \lambda_2)$$

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

(4) Add the feature points to the empty mask

Camera Motion Estimation and Filtering

Assumption, at any motion point:

$$m(x_n,y_n) = s(x_n,y_n) + w(x_n,y_n)$$

To estimate camera motion, we compare the amount of shift of two frames given the feature points by cross correlation

$$p(x_n , y_n) = [f(n) * f(n-k)^*]$$
$$m(x_n , y_n) = p(x_n , y_n) + p(x_{n-1} , y_{n-1})$$

Camera Motion Estimation and Filtering

Kalman Filtering provides the optimal estimate of the displacement Xn and Yn of the camera motion and described by this state representation:

m(k) = A m(k-1) + Bu(k) - motion points: sum of control signal and process points

z(k) = H m(k) + w(k) - combination of the signal value and the measurement noise

The experiment used a Kalman model where constant velocity is presumed during each frame translation on the x and y axis. Transition matrix which contains the xn, xn velocity ,yn and yn velocity. and observation matrix given by:

A

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

H

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

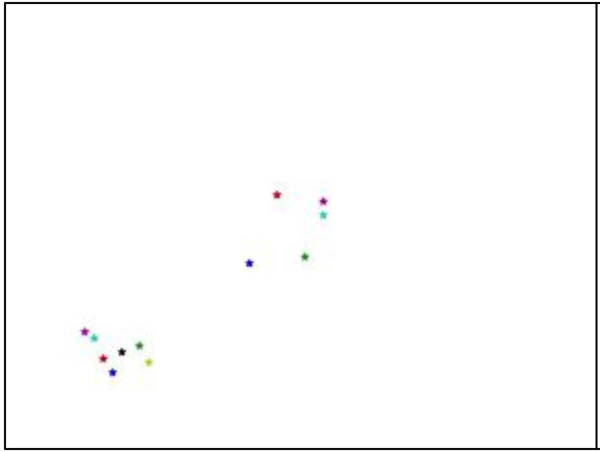
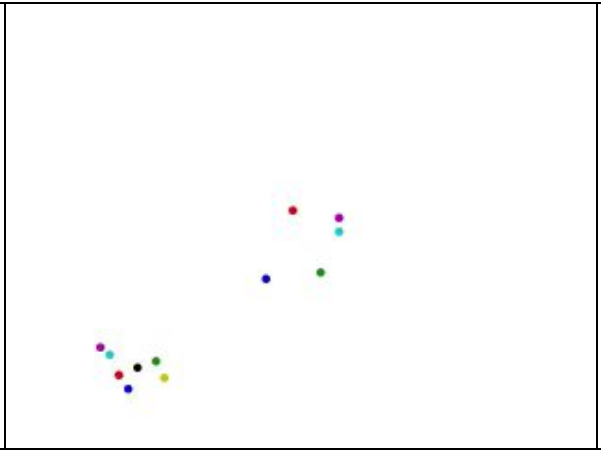
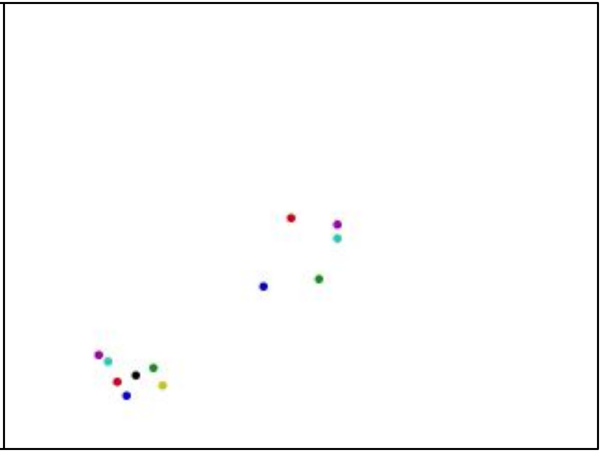
Experimental Results and Discussion

Programming Language used**: Python**
Included libraries**: Numpy, Scipy and OpenCV**

Algorithm evaluation
Simulating 10 second videos with minimally moving objects mimicking a panning motion of a handheld camera. A noise is incorporated to simulate the shaky movement of the camera.

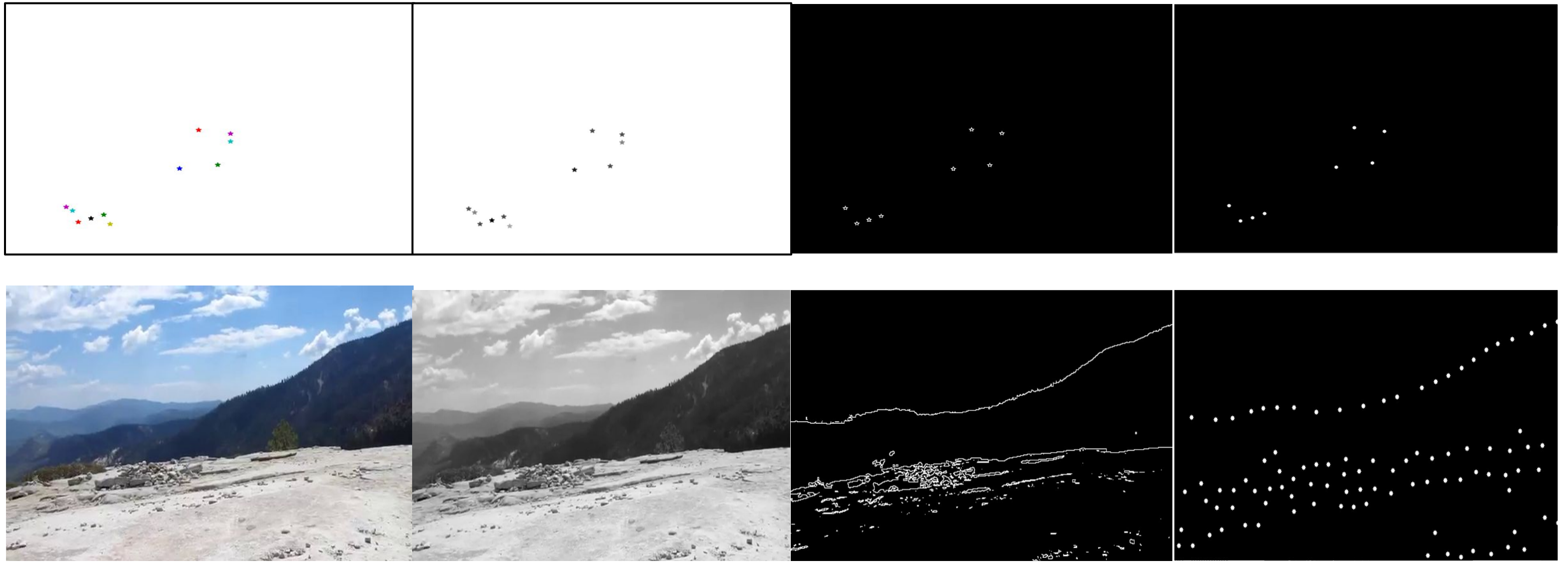
Experimental Results and Discussion

Some of the simulation videos used:

Experimental Results and Discussion

Preprocessing to Feature Extraction



Original Frame

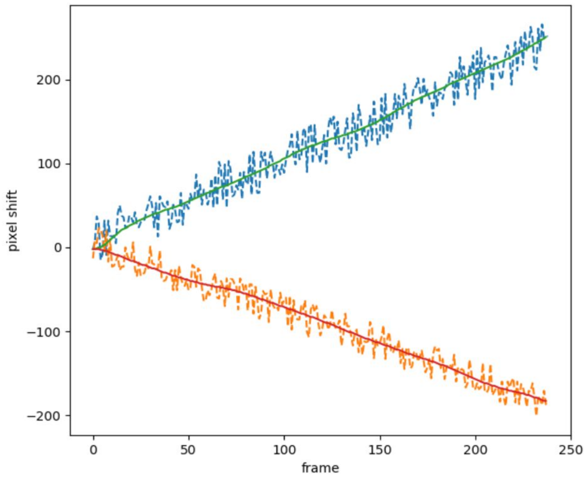
Grayscale

Normalization and Edge Detection

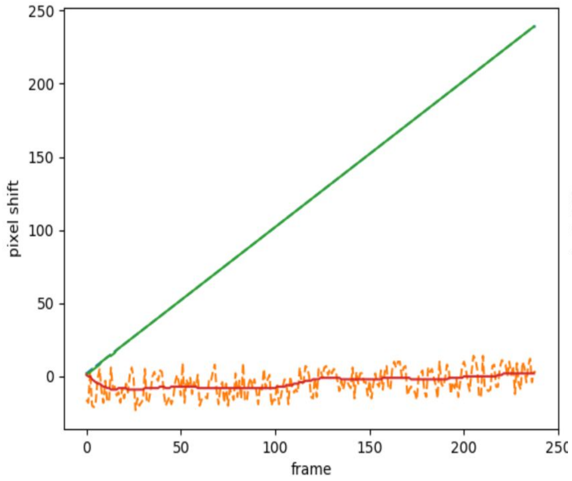
Masking and Feature Extraction

Experimental Results and Discussion

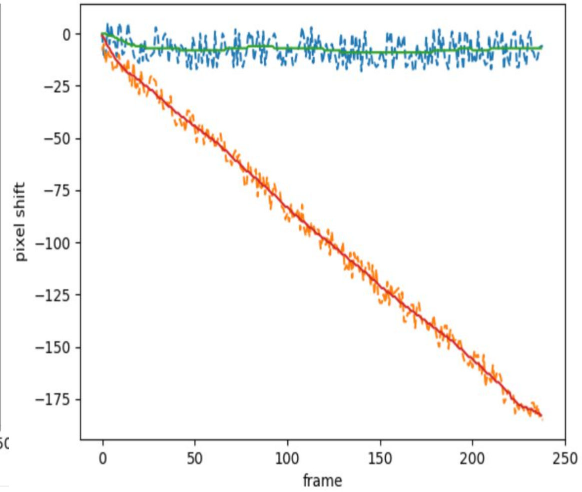
Motion Estimation and Filtering



Moving both at x and y and noise present on both axes



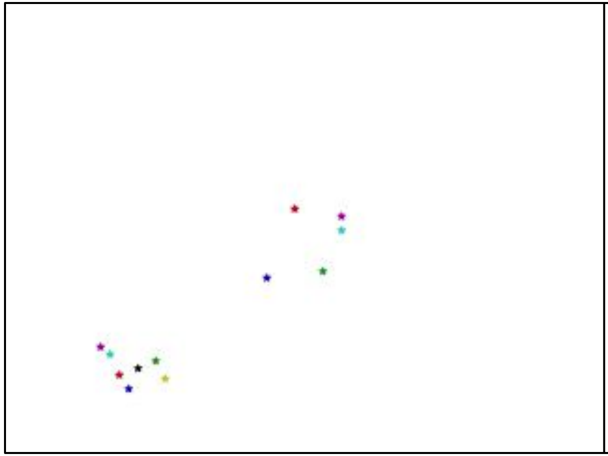
Moving along x axis, vertical noise is present, no x axis noise



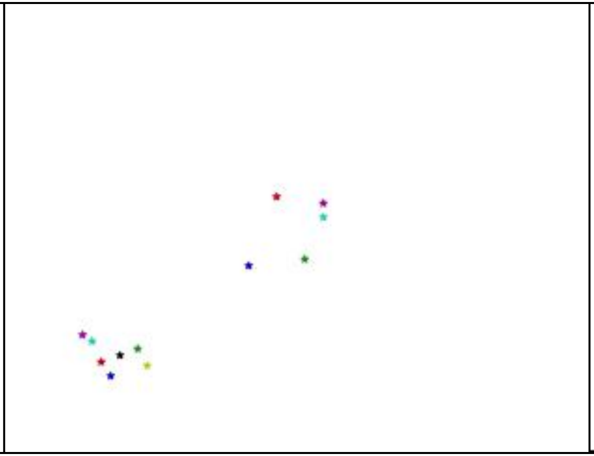
Moving along y axis, noise present on both axes

Experimental Results and Discussion

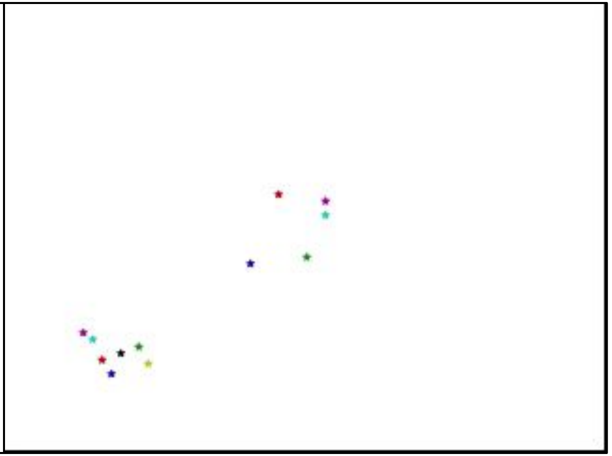
Simulation Results



Original video without noise



With shaky movement

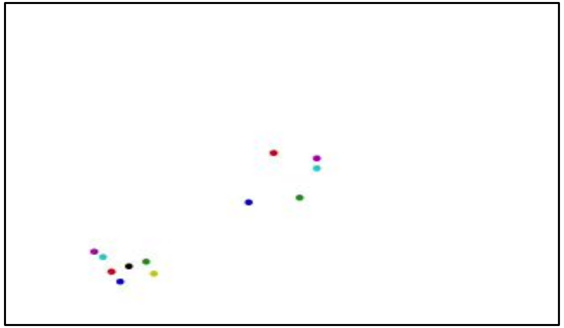


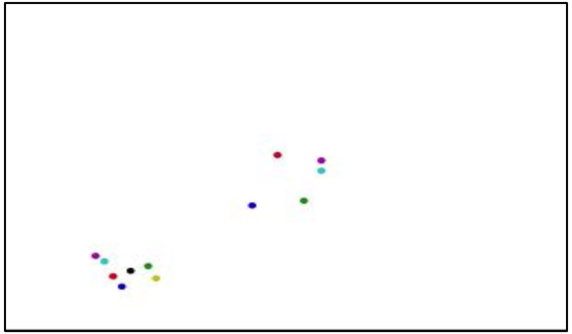
Stabilized video

Experimental Results and Discussion

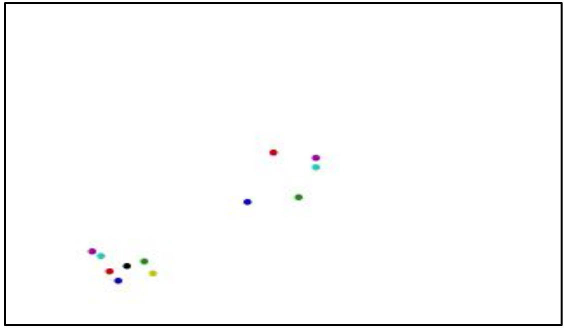
Simulation Results

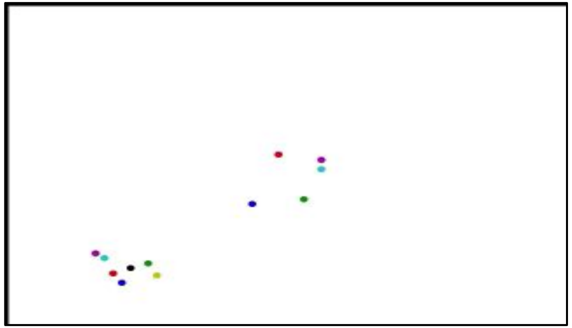
Stabilized





Movement along y-axis only





Movement along x-axis only

Experimental Results and Discussion

Applied on videos




Experimental Results and Discussion

Applied on videos

Experimental Results and Discussion

On Rolling Shutter

Conclusion

This method can compensate unwanted translation on both axes and can provide video stabilization.

However this method can be further improved to
a. Compensate rolling shutter
b. Perform image warping during image wobble
c. Analyze camera rotation if the goal is to provide still objects