Yue Xu

CS101C Final Project Report – Motion Attenuation Using Steerable Pyramids

Introduction

The goal of this project is to attenuate small motions in video recording, such as turbulence, moves caused by shaking hands etc. This idea is driven by the need of removing tiny motions of fixed animals under microscope for imaging neurons in the animals' heads. I followed the method described in Wadhwa et al. [1] to first decompose each video frame into complex-valued steerable pyramids, and then extract the phases of the pyramids and smooth the change of these phases across frames.

There are several reasons why complex-valued steerable pyramids is used. First, the computation expense is lower than other commonly used motion correction methods, such as computing optical flow. Secondly, the pyramids decompose a frame into different spatial frequency bands so we can choose to selectively modify motions at different frequency and orientations. Thirdly, the method avoids artifacts by being able to avoid spatial aliasing and avoid increasing noise from the original video.

Methods

Step 1: Develop complex-valued steerable pyramids for a video frame

(1) Perform discrete Fourier transform on the frame $\boldsymbol{I}$, and Fourier shift the results $\boldsymbol{I'}$.

(2) Develop filters of frequency-domain transfer functions, $\boldsymbol{\Phi'}_{\boldsymbol{a,k}}$, that are scaled and rotated copies of a basic filter, indexed by scale **a** and orientation **k**.

The computation is using a filter design technique for steerable pyramids described in Karasaridis et al (1996)[2]. As shown in Figure (1a) below for filters of one layer steerable pyramids, the square represents the frequency domain spectrum with $\boldsymbol{\omega_x}$ and $\boldsymbol{\omega_y}$ defined in range [-π, π], and the square region is separated into sub-regions that each represent a filter that covers the region with a specific scale and orientation. $\boldsymbol{H_0}$ is a high-pass filter covers the four corner regions of high frequencies. Then four $\boldsymbol{B_k}$ subregions represent two orientations of a base filter with **k** = [0, 1]. And $\boldsymbol{L_1}$ is the low-pass residual filter. To create steerable pyramids for multiple layers, $\boldsymbol{L_1}$ is further separated into sub-regions of $\boldsymbol{B_k}$ and another low-pass residual filter recursively, as shown in Figure (1b).

The computation to find the described filters is expensive due to requirements as shown in Figure (2a). The requirements are defined to keep the reconstructed image from steerable pyramids having a unitary amplitude as before, and to prevent spatial aliasing.

Based on the technique introduced in Portilla et al (2000)[3], the filters can be computed easily in the frequency domain using polar coordinates, where $\boldsymbol{r} = \sqrt{\boldsymbol{\omega_x^2} + \boldsymbol{\omega_y^2}}$, and $\boldsymbol{\theta} = \boldsymbol{arctan2}(\boldsymbol{\omega_y}, \boldsymbol{\omega_x})$. The equations used are shown in Figure (2b), where **K** is the number of orientations and $\boldsymbol{\alpha_K} = \boldsymbol{2^{K-1}} \frac{\boldsymbol{(K-1)!}}{\sqrt{\boldsymbol{K[2(K-1)]!}}}$ (note that the equation was written wrongly in the original paper). The equations are designed so that the edge of subregions is smoothed with a raised cosine. For the 0-th layer to compute the high-pass filter $\boldsymbol{H_0}$ and $\boldsymbol{L_0}$ defined as shown in Figure (3). And for the following layers after subsampling with factor 2, the inputs become $\boldsymbol{r} = \{\boldsymbol{r}, \boldsymbol{2r}, \boldsymbol{4r}, \ldots\}$ respectively (for octave bands) and $\boldsymbol{\theta}$ remains the same. Note that $\boldsymbol{B_k}$ is separated into radial and angular parts. For each layer, the radial part of $\boldsymbol{B_k}$ filters are computed by multiplying the **H** filter with the previous layer's low-pass residual **L**, and the angular parts are the same across layers.

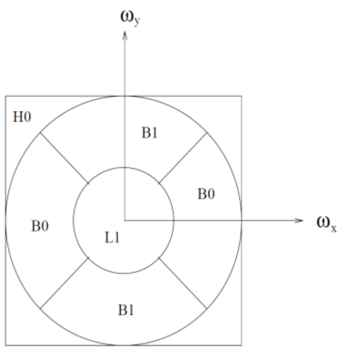

Figure (1a)

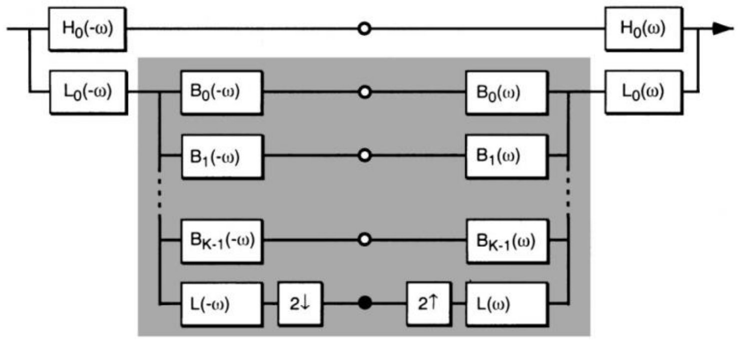

Figure (2a)

1. Unity system response amplitude:

$$|L_0(\vec{\omega})|^2\left[|L_1(\vec{\omega})|^2+\sum_{k=0}^{n}|B_k(\vec{\omega})|^2\right]+|H_0(\vec{\omega})|^2=1. \tag{2}$$

2. Recursion relationship. The low-pass branch of the diagram must be unaffected by insertion of the recursive portion of the system (see caption of figure 2):

$$|L_1(\vec{\omega}/2)|^2\left[|L_1(\vec{\omega})|^2+\sum_{k=0}^{n}|B_k(\vec{\omega})|^2\right]=|L_1(\vec{\omega}/2)|^2. \tag{3}$$

3. Aliasing cancellation (for a circular symmetric filter):

$$L_1(\vec{\omega})=0, \qquad \text{for} \quad |\vec{\omega}|>\pi/2. \tag{4}$$

Figure (2a)

$$L(r,\theta)=\begin{cases}2\cos\left(\frac{\pi}{2}\log_2\left(\frac{4r}{\pi}\right)\right), & \frac{\pi}{4}<r<\frac{\pi}{2}\\ 2, & r\le\frac{\pi}{4}\\ 0, & r\ge\frac{\pi}{2}\end{cases}$$

$$B_k(r,\theta)=H(r)G_k(\theta), \quad k\in[0,K-1],$$

$$H(r)=\begin{cases}\cos\left(\frac{\pi}{2}\log_2\left(\frac{2r}{\pi}\right)\right), & \frac{\pi}{4}<r<\frac{\pi}{2}\\ 1, & r\ge\frac{\pi}{2}\\ 0, & r\le\frac{\pi}{4}\end{cases}$$

$$G_k(\theta)=\begin{cases}\alpha_K\left[\cos\left(\theta-\frac{\pi k}{K}\right)\right]^{K-1}, & \left|\theta-\frac{\pi k}{K}\right|<\frac{\pi}{2}\\ 0, & \text{otherwise,}\end{cases}$$

Figure (2b)

$$L_0(r,\theta)=\frac{L\left(\frac{r}{2},\theta\right)}{2} \qquad H_0(r,\theta)=H\left(\frac{r}{2},\theta\right)$$

Figure (3)

(3) Matrix element-wise multiplicate the Fourier transformed ***I'*** with each filter to compute the steerable pyramids in the frequency domain, for each scale **a** and orientation **k**, $\boldsymbol{I'\Phi'_{a,k}}$.

(4) Perform inverse discrete Fourier transform to all $\boldsymbol{I'\Phi'_{a,k}}$ to get steerable pyramids $\boldsymbol{S_{a,k}}$.

Step 2: Modify phases and collapse pyramids back to the video frame

(1) Extract phase components of each element in the steerable pyramid results

(2) Compute the median phase of each element across steerable pyramids of different frames (frames within a window centered around the video frame)

(3) Replace the phase components with the median phase values.

(4) Add new phase components back to the steerable pyramids, to get the modified steerable pyramids $\boldsymbol{S_{a,k\,(new)}}$.

(5) Collapse the steerable pyramids back into the video frames. With the equation that $I_{new} = \text{ifft}(\sum_{a,k} 2[\text{fft}(S_{a,k(new)})\boldsymbol{\Phi}'_{\boldsymbol{a,k}}])$, where we multiply by 2 because the transfer functions of a complex steerable pyramid only contain the positive frequencies of the corresponding real steerable pyramid's filters, and the amplitude of filters must add up to unity as described in Figure (2a).

Results

The method was performed on three videos "fly03_short.avi", "subway.mp4", and "moon.avi". And the steerable pyramids are computed for all three R, G, B channels (RGB method) or computed for the Y channel in Y, I, Q color space (YIQ method). The steerable pyramids filters used have 4 orientations over octave bands, and number of levels are dependent on frame sizes. Two window sizes are used to compute median phase across frames, 11-frame window and 25-frame window. (Please refer to the code package for video inputs and results, but results of steerable pyramids amid constructions are shown below).

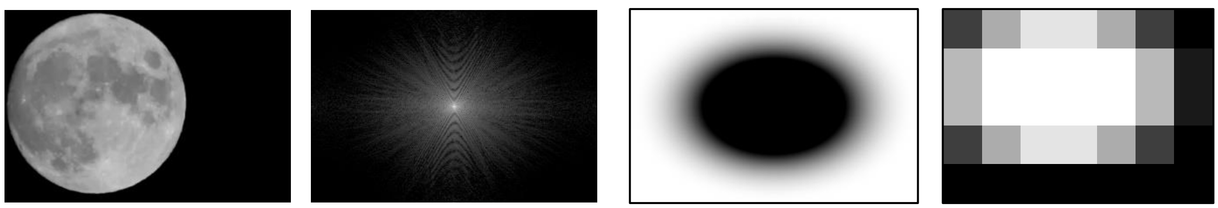

Figure (4a) Figure (4b) Figure (4c) Figure (4d)

Figure 4: a. an example image; b. the shifted Fourier transform in (a); c. the H0 filter; d. the low-pass residual filter (maximized to 7 times the original size).

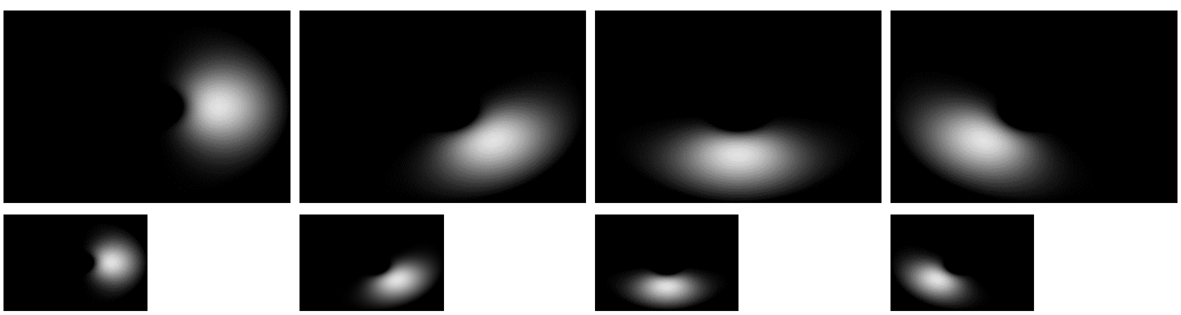

Figure 5: Top $\mathbf{B_k}$ filters of the first layer. Bottom $\mathbf{B_k}$ filters of the second layer.

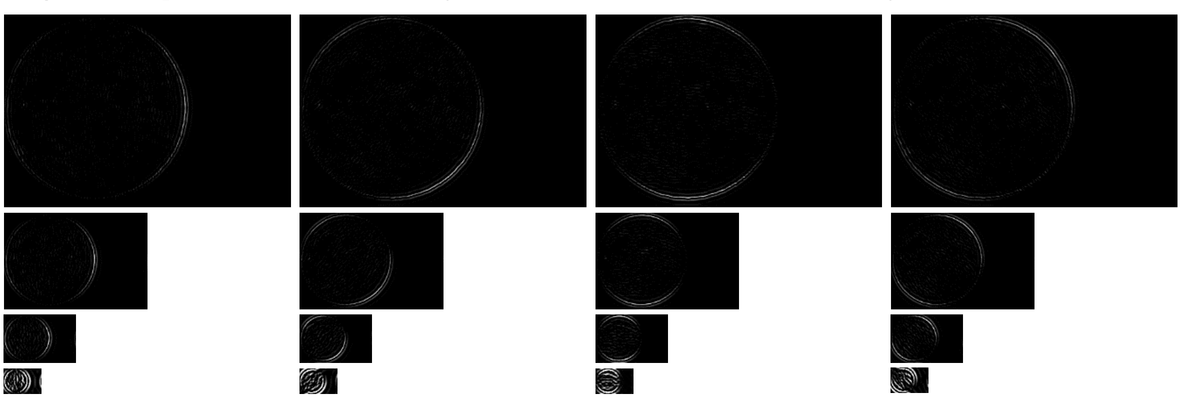

Figure 6: Top 4 layers of $\mathbf{B_k}$ filtered results of Figure (4a).

To compare results of different schemes, there are no perceivable differences between results of RGB and YIQ methods, but YIQ method computes much faster. A larger window size removes more small motions compared to a smaller one. To compare results of “moon.avi” with the results of original paper [1], the performances are similar. To compare results of “fly03_short.avi” with the results of NoRMCorre method [4], both results still contain larger-scale motions with most small motions removed. However, NoRMCorre method takes longer time to compute.

In conclusion, the project successfully reproduced motion attenuation results described in the paper of Wadhwa et al [1] for the “moon.avi”. And the project is able to generalize the performance to two other videos. However, not all unwanted motions are removed from the videos, and further developments are needed such as decomposing image into finer scales and orientations and using band-pass filter to select specific phase-changes to modify. During the project, I encountered difficulties in understanding how to implement the steerable pyramids and the mathematical equations. To solve the difficulties, I traced back to reference papers to find clear descriptions of how to construct steerable pyramid filters.

Reference

[1] Wadhwa, N., Rubinstein, M., Durand, F., Freeman, W. 2013. Phase-based Video Motion Processing. ACM Trans. Graph. 32, 4, Article 80 (July 2013), 9 pages. DOI = 10.1145/2461912.2461966 http://doi.acm.org/10.1145/2461912.2461966

[2] Karasaridis, A., Simoncelli, E. 1996. A filter design technique for steerable pyramid image transforms. Proc. Int'l Conf Acoustics Speech Signal Processing (ICASSP),.vol.IV (May 1996): 2387—2390. DOI = 10.1109/ICASSP.1996.547763 https://www.cns.nyu.edu/pub/eero/karasaridis96.pdf

[3] Portilla, J., Simoncelli, E. 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision 40(1) (Oct 2000): 49–71. DOI = 10.1023/A:1026553619983 https://www.cns.nyu.edu/pub/eero/portilla99-reprint.pdf

[4] Pnevmatikakis, E.A., Giovannucci, A. 2017. NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. Journal of Neuroscience Methods. 291 (August 2017): 83-94. http://dx.doi.org/10.1016/j.jneumeth.2017.07.031