

**MOTION FILTERING: FREQUENCY DOMAIN
APPROACH**

Elena Zorn, Lokesh Ravindranathan

05-03-2010

Boston University

Department of Electrical and Computer Engineering

Technical Report No. ECE-2010-01

**BOSTON
UNIVERSITY**

MOTION FILTERING: FREQUENCY DOMAIN APPROACH

Elena Zorn, Lokesh Ravindranathan

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
www.bu.edu/ece

05-03-2010

Technical Report No. ECE-2010-01

Summary

Motion filtering, is one of the less explored topics in video processing, but has a lot of potential with many applications. Taking advantage of our ability to isolate motion in the frequency domain is one of the basic motivations behind this project. This approach to motion filtering is theoretically elegant but requires detailed understanding of the filters. We have developed a method to understand and test these filters allowing for ease of implementation and analysis. We have designed and implemented gabor filters for the detection of motion in a particular direction (1 of 8) at a time. Two dimensional filters were designed for 2D simulated data followed by 3D filters for 3D simulated data. These filters were then used on natural images of Boston University pedestrian traffic. Results show that motion could be filtered but there were limitations.

Contents

1	Introduction	1
2	Literature Review	1
3	Problem Statement	2
4	Implementation	3
4.1	Ground Truthing in Two Dimensions	3
4.2	Ground Truthing in Three Dimensions	5
5	Experimental Results	10
6	Conclusions	13
7	References	14
8	Appendix	15
8.1	Appendix A: Matlab Code	15
8.2	Appendix B: Parseval's Theorem	16

List of Figures

1	The first row shows the 2D simulated patterns created for this study. The second row shows the absolute value of the Fourier transforms of the test patterns. The third row shows the energy of four Gabor filters with center Frequencies at $(f_{x0}, f_{y0}) = (-1/\sqrt{32}, 1/\sqrt{32}), (1/\sqrt{32}, 1/\sqrt{32}), (1/4, 0), (0, 1/4)$. We kept $f_{x0}^2 + f_{y0}^2$ constant to ensure that the energies could be compared.	4
2	This image was chosen for its structure. 50x50 pixel image patches were used to simulate motion in 8 directions as indicated by the arrows.	5
3	This image was chosen for its minimal high frequency content. 100x100 pixel image patches were used to simulate motion in various directions as indicated by the boxes and arrows. (Picture taken from: http://www.boston.com/bigpicture/2010/02/colorful_india.html . .	6
4	Center frequencies and standard deviations selected for family of 8 Gabor filters. Note that the first four rows are the same as the last four except for the sign of f_t . Changing the sign of f_t changes the directionality of the filter by 180 degrees.	7
5	Iso-surface plot of a Gabor filter with center frequency $(f_x, f_y, f_t) = (0, 1/4, 1/4)$ and $(\sigma_x, \sigma_y, \sigma_t) = (4, 4, 1)$	7
6	Iso-surface plot of last four Gabor filters described in the table above.	8
7	Iso-surface plot of all Gabor filters described in the table above plus an additional 4 stationary filters.	8
8	This plane represents a velocity in the downward motion. The filter represents one with center frequencies of $(f_x, f_y, f_t) = (0, 1/4, 1/4)$ and $(\sigma_x, \sigma_y, \sigma_t) = (4, 4, 1)$	9
9	Images in first column are selected frames from the Marsh Chapel video sequence. Images in second column show the results of the filter for leftward motion. Images in third column show the results for rightward motion.	10
10	Images in first column are selected frames from the Marsh Chapel video sequence. Images in second column show the results for upward motion. Images in third column show the results for downward motion.	11
11	Images in the first column are selected frames from the Lokesh walking video sequence showing deformable object motion and jittery camera motion. Images in the second column are filtered images for rightward motion.	12

1 Introduction

As our society increases its consumption of video data, the required analysis is also becoming more complex. Not only will motion detection in video sequences be required, but some form of motion analysis will be useful. Monitoring access to a store front in the midst of high pedestrian activity, gesture recognition [3], anomaly detection and robotic visual scene interpretation are some of the applications of direction-specific motion estimation. Motion filtering using frequency domain analysis is not a new topic but it is not as well developed as time domain methods such as optical flow. There are many approaches to estimate motion using successive frames but filtering motion in specific directions is difficult. The HornSchunck and Lucas-Kanade Optical flow algorithms are common methods that use 2 frames for motion estimation. The accuracy of the estimates is not dependable because factors like changes in intensity between frames and the movement of highly textured objects [2]. We explored the creation of a family of Gabor filters to which motion filtering in 8 directions was applied. We approached the implementation in a careful manner by first understanding 2 dimensional data and then moved to implementation with 3 dimensional simulated data. Finally, the implementation with videos of pedestrian traffic around the Boston University East Campus was analysed.

2 Literature Review

The application of motion filters using multiple frames is desirable. David Heeger's thesis [1] is one of the earliest papers dealing with the filtering of movement using many frames. He uses twelve different Gabor filters to determine velocity on patches of images over multiple frames. 8 of the filters are to determine velocities in the left, right, up, down, and four other corners. The four additional filters are for stationary responses. Several families of these filters could be built to describe various spatiotemporal-frequency bands, but Heeger opted to use a Gaussian pyramid scheme similar to the one described by Adelson Et Al [6][7]. Heeger[1][2] uses Parseval's Theorem to determine motion energy in various directions. A more recent application of gabor filters for spatiotemporal motion estimation was used for interpretation of dynamic hand gestures. Yeasin and Chaudhuri analyze a series of frames that encompass a hand gesture like one that represents come here. Gabor filters are used to determine various motions and a finite state machine developed by the authors is used to decide which hand gesture is being presented.[5]

3 Problem Statement

An object moving with a constant velocity is always represented by a plane in the spatio-temporal frequency domain. The orientation of the plane is dictated by the velocity of the object. We aim to develop a set of filters that can distinguish between varying orientations of this plane. Filters were created for differentiating between 8 different motions. Parseval's theorem states that the sum of a square of a function is equal to the sum of the square of its transform. This theorem is used to describe the intersection of a plane that represents an object in motion and a filter. Energy of the intersection of a plane in the frequency domain with the frequency response of a motion filter is the same as the energy of the filtered output in space time domain.

$$\begin{aligned}
& \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} |f(x, y, t)|^2 dx dy dt \\
&= \frac{1}{8\pi^3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |F(f_x, f_y, f_z)|^2 df_x df_y df_z \\
&= \frac{1}{8\pi^3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |P(f_x, f_y, f_z)| df_x df_y df_z
\end{aligned} \tag{1}$$

Our goal would be to design a set of filters which are tuned to 8 directions of movement (up, down, left, right and 4 diagonals). We will be designing 3 dimensional Gabor filters to capture the energies of the moving objects in a video sequence.

$$G_s(x, y, t) = \frac{1}{(2\pi)^{3/2} \sigma_x^2 \sigma_y^2 \sigma_z^2} \exp\left\{-\left(\frac{x^2}{2\sigma_x^2}\right) + \left(\frac{y^2}{2\sigma_y^2}\right) + \left(\frac{z^2}{2\sigma_z^2}\right)\right\} \sin(2\pi f_{x_0} + 2\pi f_{y_0} + 2\pi f_{z_0})$$

$$G_c(x, y, t) = \frac{1}{(2\pi)^{3/2} \sigma_x^2 \sigma_y^2 \sigma_z^2} \exp\left\{-\left(\frac{x^2}{2\sigma_x^2}\right) + \left(\frac{y^2}{2\sigma_y^2}\right) + \left(\frac{z^2}{2\sigma_z^2}\right)\right\} \cos(2\pi f_{x_0} + 2\pi f_{y_0} + 2\pi f_{z_0})$$

4 Implementation

Our approach to defining a specific set of Gabor filters involved creating simulated data to tune the Gabor filters. We first did this in two dimensions and then repeated the process for three dimensions. This is what we called "Ground Truthing". Finally, the Gabor filters were applied to real data.

4.1 Ground Truthing in Two Dimensions

We first created a two dimensional pattern out of a one dimensional signal. From an image we selected a row of 50 pixels. This one dimensional signal was shifted by one pixel (the end wrapping around to the beginning) and copied to the line below. This step was repeated until a two dimensional image of 50x50 pixels was created. The resulting image is a striped one like the one shown below. In a similar manner, we created images to depict diagonal stripes that were orthogonal to the previous image. In addition, horizontal and vertical striped images were created. As a result, we obtained four images that had energy that closely resembled a line in the frequency domain. For each of these images, the energy in the frequency domain is represented by a line. The energy of the Sine Gabor filters is shown. Since convolution in the space-time domain is equivalent to multiplication in the frequency domain, the amount of energy captured by the filters is well represented with the center frequencies provided[8]. Similarly for the Cosine Gabor filters.

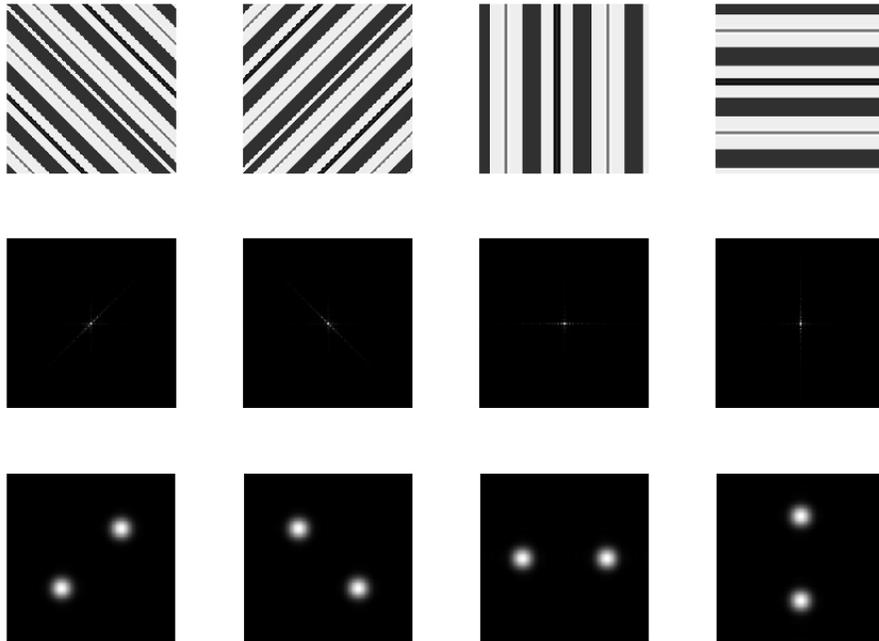


Figure 1: The first row shows the 2D simulated patterns created for this study. The second row shows the absolute value of the Fourier transforms of the test patterns. The third row shows the energy of four Gabor filters with center Frequencies at $(f_{x0}, f_{y0}) = (-1/\sqrt{32}, 1/\sqrt{32}), (1/\sqrt{32}, 1/\sqrt{32}), (1/4, 0), (0, 1/4)$. We kept $f_{x0}^2 + f_{y0}^2$ constant to ensure that the energies could be compared.

4.2 Ground Truthing in Three Dimensions

Motion was simulated in a similar manner for ground truthing in three dimensions. From a highly textured image a small patch was extracted. For each frame the patch that was captured was shifted in various directions by moving the area down, right, up, left or diagonally by a pixel. We created 50-frame simulations of 50x50 and 100x100 pixel images. These frames were tested with a family of 8 Gabor filters that represented the 4 cardinal directions and 4 diagonals. The center frequencies were selected in a similar manner so that $f_x^2 + f_y^2$ remained constant. It was selected to be $1/4$. This was a recommendation that came from Heeger's paper on Optical Flow [4]. The images and filters that were tested are shown below.



Figure 2: This image was chosen for its structure. 50x50 pixel image patches were used to simulate motion in 8 directions as indicated by the arrows. (Picture taken from: http://www.boston.com/bigpicture/2010/02/colorful_india.html)

In order to understand the parameters of the filters and to ensure that they were implemented correctly, we visualized the magnitude of the filters in the frequency domain. They were normalized to have a max of one and were displayed in an iso-surface plot where the surface displayed is at iso-surface = .5.

We could then see that any plane representing velocity of an object would pass



Figure 3: This image was chosen for its minimal high frequency content. 100x100 pixel image patches were used to simulate motion in various directions as indicated by the boxes and arrows. (Picture taken from: http://www.boston.com/bigpicture/2010/02/colorful_india.html)

	f_{x0}	f_{y0}	f_{t0}	σ_x	σ_y	σ_t
→	1/4	0	-1/4	4	4	1
↗	1/√32	1/√32	-1/4	4	4	1
↑	0	1/4	-1/4	4	4	1
↖	-1/√32	1/√32	-1/4	4	4	1
←	1/4	0	1/4	4	4	1
↙	1/√32	1/√32	1/4	4	4	1
↓	0	1/4	1/4	4	4	1
↘	-1/√32	1/√32	1/4	4	4	1

Figure 4: Center frequencies and standard deviations selected for family of 8 Gabor filters. Note that the first four rows are the same as the last four except for the sign of f_t . Changing the sign of f_t changes the directionality of the filter by 180 degrees.

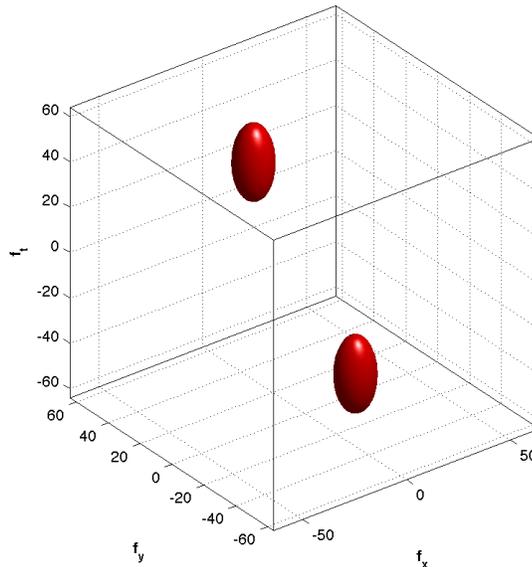


Figure 5: Iso-surface plot of a Gabor filter with center frequency $(f_x, f_y, f_t) = (0, 1/4, 1/4)$ and $(\sigma_x, \sigma_y, \sigma_t) = (4, 4, 1)$

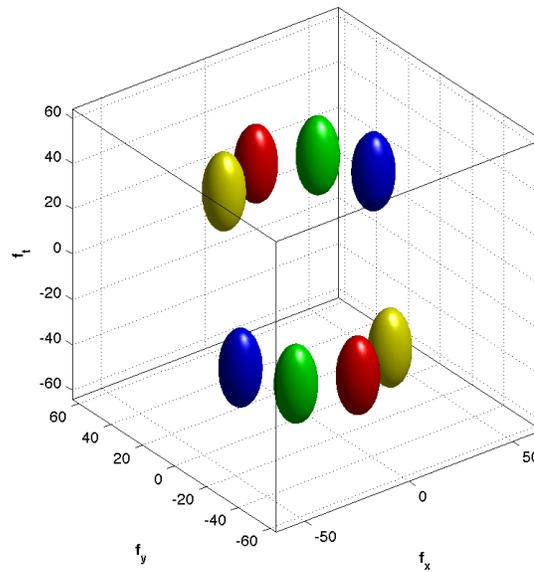


Figure 6: Iso-surface plot of last four Gabor filters described in the table above.

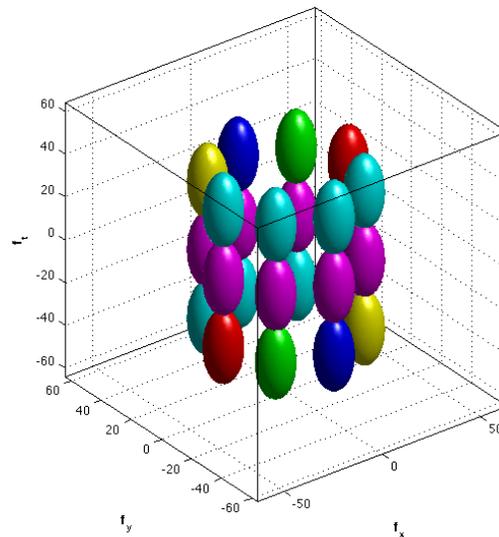


Figure 7: Iso-surface plot of all Gabor filters described in the table above plus an additional 4 stationary filters.

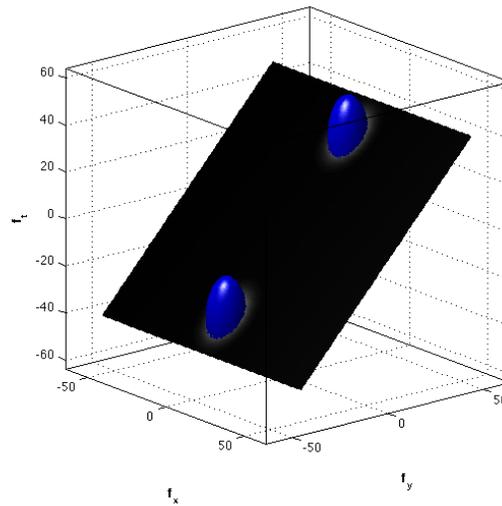


Figure 8: This plane represents a velocity in the downward motion. The filter represents one with center frequencies of $(f_x, f_y, f_t) = (0, 1/4, 1/4)$ and $(\sigma_x, \sigma_y, \sigma_t) = (4, 4, 1)$

through a combination of these filters. The goal would be to design these filters so that there is minimal ambiguity between the directional filters. The standard deviations of the filters were selected to this effect.

5 Experimental Results

One of the goals of our project was to design a system to track the number of people entering a shop and the number of people walking past the shop. To this end, we created a scenario (Marsh Chapel Plaza) where there is a high pedestrian density walking past the plaza and a lower pedestrian density walking across the plaza. We designed 4 motion filters, 2 of them filtering motion across the plaza and 2 of them filtering motion past the plaza. The image and the respective filtered image using the 4 motion filters are shown below.

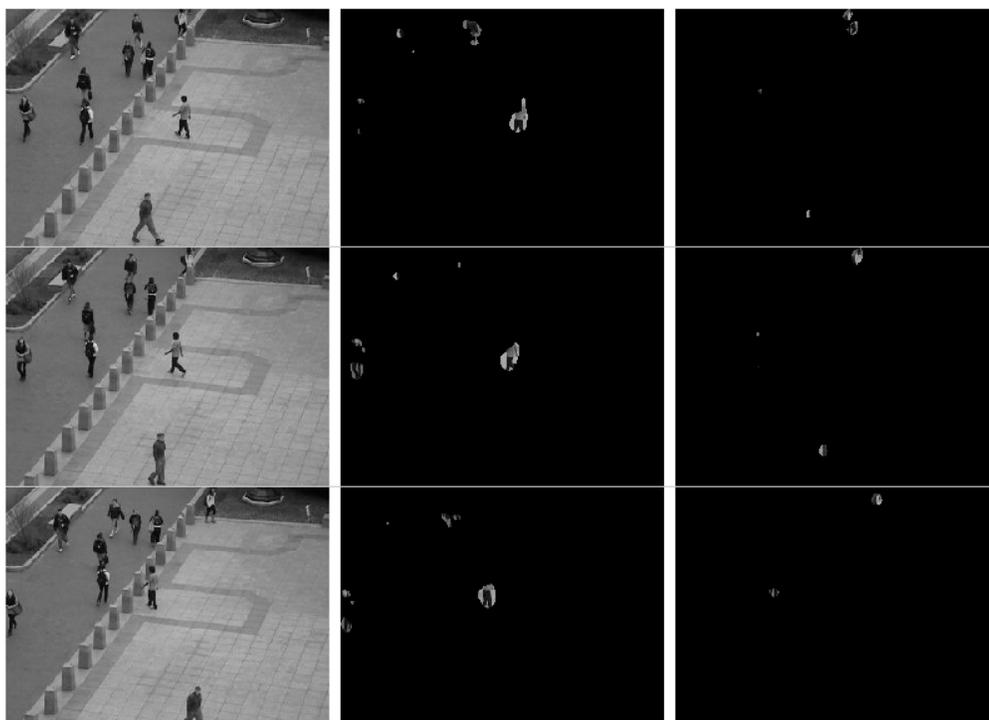


Figure 9: Images in first column are selected frames from the Marsh Chapel video sequence. Images in second column show the results of the filter for leftward motion. Images in third column show the results for rightward motion.

The video sequence shown above was captured using a Nikon Camera and a stable tripod. The goal of the filters was to filter the motion in their respective directions. The first filter filters out the person moving right to left. The second filter filters out the person moving left to right as seen in the top portion of the second frame.

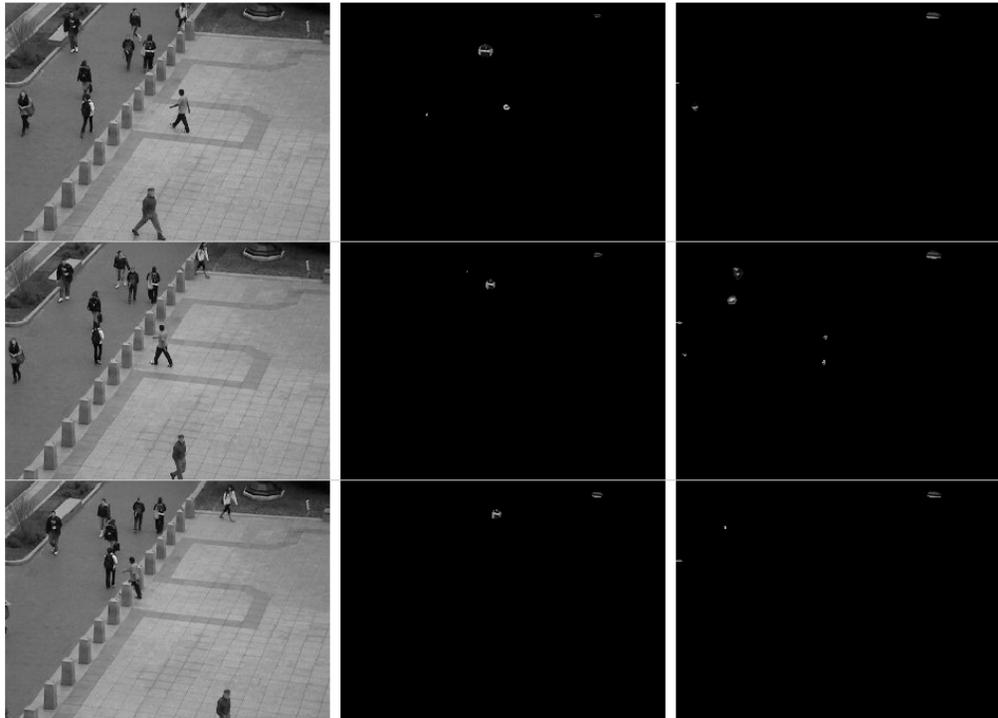


Figure 10: Images in first column are selected frames from the Marsh Chapel video sequence. Images in second column show the results for upward motion. Images in third column show the results for downward motion.

The video sequence shown above was captured using a Nikon Camera and a stable tripod. The goal of the filters was to filter the motion in their respective directions. The first filter filters out the pedestrian traffic moving from the bottom of the frame to the top. The highly textured area of the pedestrian's shirt was detected easily in the three frames shown. However, there is less texture in the pedestrian's shirt moving from the top of the frame to the bottom and has been detected only in the second frame of the filtered frames.

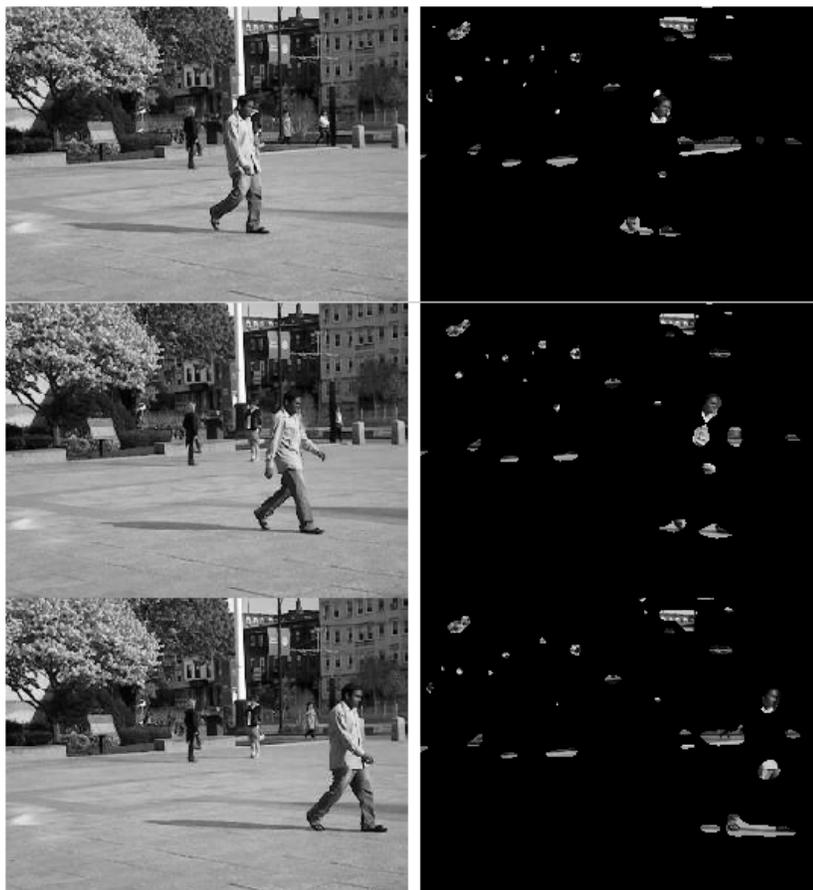


Figure 11: Images in the first column are selected frames from the Lokesh walking video sequence showing deformable object motion and jittery camera motion. Images in the second column are filtered images for rightward motion.

The video sequence shown above captured using a Nikon Camera. The filter filters out the person moving right to left. Person's head is clearly filtered out but the rest of the body is not detected because of the non-linear motion. The camera is jittery and hence the filtered image shows some portions of the background. Because of the small movements in the camera, objects at a distance were detected as moving.

6 Conclusions

Texture is important for performing motion filtering in the frequency domain. Analysis in the frequency domain is based on frequencies seen in x and y directions of an image in an image sequence. An object with little texture will have low frequencies and it would be tough to find the corresponding shift in the frequency domain. An object with very high texture will have similar problems too.

Temporal Aliasing creates a huge hurdle in filtering motion. This kind of aliasing causes the filter to detect wrong direction of motion.

Setting the right filter design parameters is challenging. Small spatio-temporal standard deviations cause ambiguity in the frequency domain. Large spatio-temporal standard deviations can miss certain frequency planes.

Some of the difficulties we faced with the implementation were:

- (i) Brightness contrast between moving objects and background is required for detecting motion.
- (ii) Diagonal movements are tougher to filter out since they are a combination of two planes of motion.

Possible improvements using existing Gabor filters would be to:

- a. Estimate motion direction with a subset of Gabor Filters by taking advantage of the known ambiguities.
- b. Use a hierarchical approach by dividing the image into blocks and estimate dense vector field by using overlapping blocks.

7 References

- [1] David J. Heeger, Models for motion perception, Ph.D. Thesis, CIS Department, Univ. of Pennsylvania, 1987. (Technical report MS-CIS-87-91.)
- [2] David J. Heeger, Optical Flow Using Spatiotemporal Filters International Journal of Computer vision, pp. 279-302, 1988.
- [3] M. Yeasin, S. Chaudhuri, Visual understanding of dynamic hand gestures, Pattern Recognition, Vol. 33, pp. 1805-1817, 2000.
- [4] David J. Heeger, Model for the extraction of image flow, Journal of Optical Society America, Vol. 4, No. 8, pp.1455-1471, August 1987.
- [5] E. Bruno, D. Pellerin, Robust motion estimation using spatial Gabor-like filters, Signal Processing, Vol. 82, pp. 297-309, 2002.
- [6] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, J. M. Ogden, Pyramid methods in image processing, RCA Engineer, Vol. 29-6, pp. 33-41, Nov/Dec 1984.
- [7] E.P. Simoncelli, E.H. Adelson, Computing Optical Flow Distributions Using Spatiotemporal Filters, MIT Media Laboratory Vision and Modeling Technical report-165,1991.
- [8] J.G. Daugman, Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two dimensional visual cortical filters, J. Opt. Soc. Am. A2 pp. 1160-1169, 1985.
- [9] S.Schauland, J.Velten, A.Kummert, Detection of Moving object in Image Sequences using 3D Velocity Filters, Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18, No. 1, 2131

8 Appendix

8.1 Appendix A: Matlab Code

Gabor Filters Code

```
function [y1,y2]=gaborfilto(img,fx,fy,ft,sx,sy,st)
%%% 3-D Gabor Filter Output Implementation
%%% By Lokesh A. Ravindranathan, Elena Zorn (LM: 04/10/2010)
for x = -3*sx:3*sx,
    for y = -3*sy:3*sy,
        for t = -3*st:3*st,
            G(3*sx+x+1,3*sy+y+1,3*st+t+1) = (1/(sqrt(2)*((pi)^(1.5))*sx*sy*st))*...
            exp(-((x^2/(2*(sx^2)))+(y^2/(2*(sy^2)))+(t^2/(2*(st^2))))+2*pi*1i*((fx*x)+(fy*y)+(ft*t)));
        end
    end
end
end

cosconv = convn(img,real(G),'same');
sinconv = convn(img,imag(G),'same');
gabout1 = cosconv;
gabout2 = sinconv;
y1 = gabout1;
y2 = gabout2;

Code for filtering and Thresholding
[g2(:,:,:) g3(:,:,:)] = gaborfilto(double(x),0,1/4,1/4,4,4,1);
[g7(:,:,:) g8(:,:,:)] = gaborfilto(double(x),1/4,0,1/4,4,4,1);
[g9(:,:,:) g10(:,:,:)] = gaborfilto(double(x),0,1/4,-1/4,4,4,1);
[g11(:,:,:) g12(:,:,:)] = gaborfilto(double(x),1/4,0,-1/4,4,4,1);
[g13(:,:,:) g14(:,:,:)] = gaborfilto(double(x),1/4,1/4,1/4,4,4,1);
[g15(:,:,:) g16(:,:,:)] = gaborfilto(double(x),1/4,-1/4,1/4,4,4,1);
[g17(:,:,:) g18(:,:,:)] = gaborfilto(double(x),1/4,1/4,-1/4,4,4,1);
[g19(:,:,:) g20(:,:,:)] = gaborfilto(double(x),1/4,-1/4,-1/4,4,4,1);

g4=g2.^2+g3.^2;
g5=g4./max(max(max(g4)));
g6 = (g5>0.05);
```

8.2 Appendix B: Parseval's Theorem

A 1-Dimensional Gabor odd phase(sine phase) filter is defined as a product of a Gaussian signal and sine signal of a given frequency:

$$g_s(t) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{t^2}{2\sigma^2}} \sin(2\pi f_0 t)$$

A 1-Dimensional Gabor even phase(cosine phase) filter is defined as a product of a Gaussian signal and cosine signal of a given frequency:

$$g_c(t) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{t^2}{2\sigma^2}} \cos(2\pi f_0 t)$$

Gabor function can also be seen as: $g(t) = \text{Product } [h_1(t), h_2(t)]$ where $h_1(t) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{t^2}{2\sigma^2}}$ and $h_2(t) = \sin(2\pi f_0 t)$

Fourier Transform of Gabor Function is defined as $F[g(t)]$ and can be re-written as follows: $F[g(t)] = F[h_1(t)] * F[h_2(t)]$ – Eqn(1)

Fourier transform of a Gaussian signal is: $F[e^{-ax^2}(k)] = \sqrt{\frac{\pi}{a}} e^{-\frac{\pi^2 k^2}{a}}$ – Eqn(2)

Fourier transform of a sine signal of a given frequency is:

$$F[\sin(2\pi f_0 t)] = \frac{i}{2}(\delta(f + f_0) - \delta(f - f_0))$$
 – Eqn(3)

Fourier transform of a cosine signal of a given frequency is:

$$F[\cos(2\pi f_0 t)] = \frac{1}{2}(\delta(f + f_0) + \delta(f - f_0))$$
 – Eqn(4)

Substituting Eqn(2), Eqn(3) in Eqn(1),

$$G_s(f) = (e^{-\pi^2 f^2 2\sigma^2}) * (\frac{i}{2}(\delta(f + f_0) - \delta(f - f_0)))$$

$$G_s(f) = \frac{i}{2}(e^{-\pi^2(f+f_0)^2 2\sigma^2} - e^{-\pi^2(f-f_0)^2 2\sigma^2})$$

For odd phase filter, energy is calculated using Parseval's theorem by convolving the filter with an arbitrary sine wave,

$$\begin{aligned} \int_{-\infty}^{\infty} |G_s(f_0) * \sin(2\pi f_1)|^2 dx &= \int_{-\infty}^{\infty} |F[G_s(f_0)]F[\sin(2\pi f_1)]|^2 df \\ &= |(\frac{1}{4})[g(f) - h(f)][\delta(f + f_1) - \delta(f - f_1)]|^2 \\ &= (\frac{1}{16})[g(f) - h(f)]^2 [\delta(f + f_1) + \delta(f - f_1) - 2\delta(f + f_1)\delta(f - f_1)] \\ &= (\frac{1}{16})[g(f_1) - h(f_1)]^2 + (\frac{1}{16})[g(-f_1) - h(-f_1)]^2 \\ &= (\frac{1}{8})[g(f_1) - h(f_1)]^2 \\ &\text{where } g(f) = e^{-2\pi^2\sigma^2(f-f_0)^2} \text{ and } h(f) = e^{-2\pi^2\sigma^2(f+f_0)^2} \end{aligned}$$

For even phase filter, energy is given by,

$$\int_{-\infty}^{\infty} |F[G_c(f_0)]F[\sin(2\pi f_1)]|^2 df = (\frac{1}{8})[g(f_1) + h(f_1)]^2$$

Energy of a Gabor phase is the sum of the energies of odd phase and even phase filters.

$$E_{total} = (\frac{1}{4})[g(f_1)^2 + h(f_1)^2]$$

$$= \left[\left(\frac{1}{4}\right)e^{-4\pi^2\sigma^2(f_1-f_0)^2}\right] + \left[\left(\frac{1}{4}\right)e^{-4\pi^2\sigma^2(f_1+f_0)^2}\right]$$

Extending the concepts to a 3-D Gabor filter,

$$g(x, y, t) = \frac{1}{(2\pi)^{3/2}} e^{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2} + \frac{t^2}{2\sigma_t^2}\right)} \sin(2\pi f_{x0}x + 2\pi f_{y0}y + 2\pi f_{t0}t)$$

where f_{x0} , f_{y0} , f_{t0} are the set of spatial and temporal centre frequencies.

$$G(f_x, f_y, f_t) = \left(\frac{1}{4}\right)e^{-4\pi^2[\sigma_x^2(f_x-f_{x0})^2 + \sigma_y^2(f_y-f_{y0})^2 + \sigma_t^2(f_t-f_{t0})^2]} \\ + \left(\frac{1}{4}\right)e^{-4\pi^2[\sigma_x^2(f_x+f_{x0})^2 + \sigma_y^2(f_y+f_{y0})^2 + \sigma_t^2(f_t+f_{t0})^2]}$$