High Speed Gabor Filter using Haar Wavelets

Woochan Seo, Minsue Park, Jinguk Kwon and Hoon Kang
School of Electrical and Electronic Engineering, Chung-Ang University
221, Huksuk-Dong, Dongjak-Gu, Seoul 156-756, Korea
Tel: +82-2-816-8234, Fax: +82-2-816-8234
E-mail: woochan@robot.cau.ac.kr

Abstract : In this paper, we propose a new way of improving the operation speed of Gabor filters. If the mask is applied to some images, a number of mathmatical operations increase due to complexity. If the mask size is reduced as small as possible in an allowable range, the operation time may decrease. But it would make a problem because the small mask is difficult to represent both frequency and orientation characteristics. Therefore, the operation speed is overcome by substituting multiplications for additions, instead of changing the mask size. First, Haar wavelets are generated in terms of binary data. We divide the mask into two groups, 1's and -1's. Thus, Haar wavelets show both frequency and orientation characteristics. Then, by applying 'Matching Pursuit', we obtain the correlations between the original Gabor filters and our binary Haar wavelets. The Gabor filter can be described by linear combinations of Haar wavelets. As a result, the binary wavelets give a benefit to the operation speed. However, some trade-off is necessary between accuracy and speed since it may cause some errors.

1. Introduction

In the field of robot or unmanned systems, the image processing accompanies an important role to acquire information about the real world and to decide their action. Particularly, according to developing the image analysis method to meet our goal, the relevance of the system is decided. Therefore, the image processing is an important object of this study.

Object tracking techniques are used for military equipments and security devices. Also if we use it for robots or machineries, we can realize the interface which the users are more familiar to. To track objects in real time, we extract features, recognize the target and predict its movement. But that processing takes long time. Therefore, it is a key point to reduce the operation time and to keep the accuracy. In this paper, we propose a new method that reduces the operation time and keeps the accuracy.

2. Gabor Filter

In 1946, Dennis Gabor suggested a method which represents some signal with sum of basic functions. The Gabor Filter is a non-orthogonal function composed of Gaussian functions and Fourier functions. Besides the characteristics of the 2-dimensional Gabor Filter is similar to one of the simple cells in the visual cortex for mammals. Therefore we can extract features in an image along the same way humans see objects. This filter mainly uses for the pattern analysis, character classification, face recognition and edge detection.

$$\psi(x,y) = \frac{f^2}{\pi\gamma\eta} e^{-\left(\frac{f^2}{\gamma^2}x'^2 + \frac{f^2}{\eta^2}y'^2\right)} e^{j2\pi f x'} \quad (1)$$

$$\begin{cases} x' = x\cos\theta + y\sin\theta \\ y' = -x\sin\theta + y\cos\theta \end{cases}$$

※ γ, η : filter bandwidth
f : frequency
θ : angle

Eq.(1) defines a 2-dimensions Gabor filter. We know the Gabor filter is consists of a real part and an imaginary part.

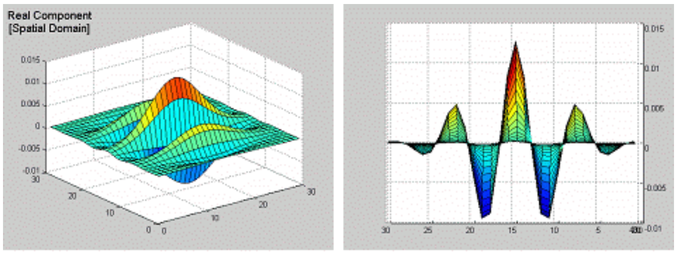

Fig. 1. A real part of 2-dimensional Gabor Filter

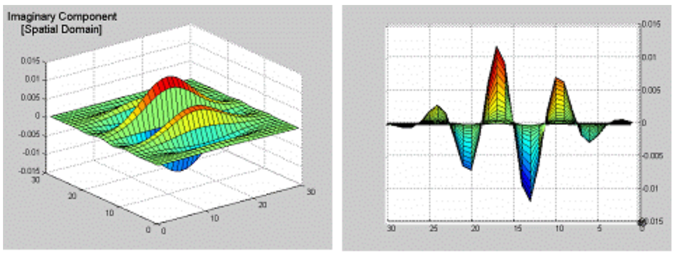

Fig. 2. An imaginary part of 2-dimensional Gabor Filter

We can organize Gabor filters with different patterns through changing the angle or the frequency. And using those filters, we can extract features which wanted on images.

3. Matching Pursuit

The Matching Pursuit is a method that represents a vector as given some vectors.

$$f = \sum_{i=1}^{k} a_i x_{n_i} + R_k = f_k + R_k \qquad (2)$$

$$\begin{cases} a_{k+1} = < R_k, x_{n_i} > \\ f_{k+1} = f_k + < R_k, x_{n_{k+1}} > x_{n_{k+1}} \\ R_{k+1} = R_k - < R_k, x_{n_{k+1}} > x_{n_{k+1}} \end{cases}$$

※ a_k : coefficients
f : given vector
x_k : base vectors
R_k : residual errors

Eq.(2) expresses a vector f as k number of given vectors x. At first, we operate the inner-product between a vector f and given vectors x_k, then we get the results as coefficients a_k. Among k number of coefficients a_i, the biggest absolute value of a_k and the concerned vector x_k are chosen as the first term. And the error between the original vector and the chosen vector x_k is represented R_k. The reason for choosing the biggest absolute value $|a_k|$ is that the inner-product represents the correlation between a vector f and given vectors x.

After picking up the first term, we choose the biggest absolute inner-product value $|a_k|$ between residual and given vector as the next term, except for the selected vector in first them.

We do over again this process until the residual reaches a permissible range. If all given vector are picked before the residual reaches a permissible range, we can choose all given vector again.

4. Haar Wavelet

In JPEG2000, the Wavelet is applied to image compression. But we use the Wavelet to represent the 2-dimensional Gabor filter. The Wavelet can be defined in variety of methods. In this paper, our goal is to organize the approximate Gabor Filter with binary features and to reduce the operation time. Thus, we use a binary Haar Wavelet.

We apply a binary version of the Gabor filter. Applying the Gabor filter, it needs a lot of convolution operation time for multiplications and additions. If the multiplication operation is replaced by the addition operation, the total operation time can be reduced.

The Haar Wavelet makes some feature sets. Because those sets are orthogonal to each other, any signal can be represented by their associations. By suitable associations of binary features, an input signal can be represented approximately.

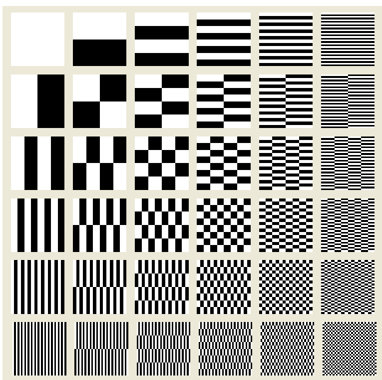

Fig. 3. Binary Haar Wavelets(32 by 32)

5. Experiment

In this chapter, the way which improves the operation speed problem of Gabor filters would be suggested. We make some Gabor filters and constitute proper Haar wavelets. Then, by applying Matching Pursuit, we obtain the correlations between the original Gabor filters and our binary Haar wavelets. Finally, we are going to show a binary version of the Gabor filter and the effect of speed improvement.

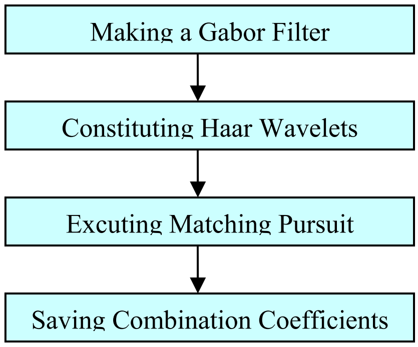

Fig. 4. The speed improvement way of Gabor Filter

5.1 Making a Gabor Filter

Above all, We make a Gabor filter. A size of the Gabor filter sets up 2^n, because Haar wavelets divides the mask into two equal parts. In this section, a size of the Gabor filter is $32(=2^5)$ pixels.

Next, we must decide a part which we use in, the real part or the imaginary part. To tell the truth, both operation results are similar because both of them are masks that excute the operation in relation to frequencies and orientations. By the way, Haar wavelets is odd functions. Hence the Gabor filter is defined in the imaginary part in Fig. 5.

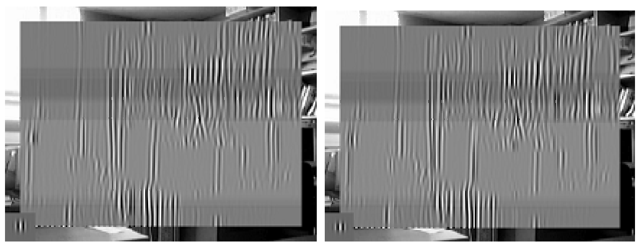

Fig. 5. The difference of
the real part(left) and the imaginary part(right)

5.2 Constituting Haar Wavelets

The problem of Gabor filters is that its mask size is big. The big mask makes many operations as well as many multiplications when the convolution executes. If wavelets are constituted Haar wavelets, then it is possible to calculate combination coefficients quickly in executing Matching Pursuit. Also,when the Gabor filter can be

described by the linear combinations of Haar wavelets, the operation time may decrease.
A size of the wavelet equals to the Gabor filter's. At this time, 36 wavelets are used. When a size of a Gabor filter is given in Eq.(3), the number of wavelets calculates in Eq.(4).

A Size of a Gabor filter : 2^n pixels (3)
The number of wavelets : $(n+1)^2$ (4)

5.3 Executing Matching Pursuit
Matching Pursuit expresses a Gabor Filter as summation between Haar Wavelets. Fig 6 shows Gabor Filters and best binary features.
Gabor Filters set up that sizes are 32 pixels, frequencies are 0.5, 1 and 2, γ and η are 1, parts are in the imaginary part.
And by calculating the convolution between Gabor Filter and each Haar Wavelet, we obtain best binary features.

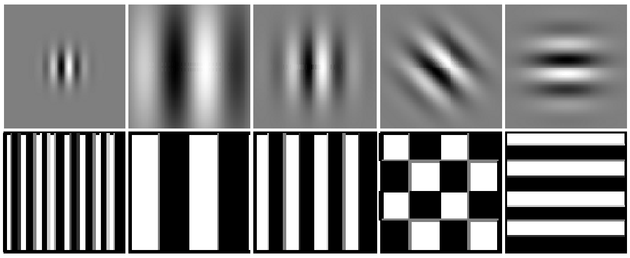

Fig. 6. The Gabor Filter(top) and its the best binary feature(bottom)

Fig. 7 show first result after executing Matching Pursuit.

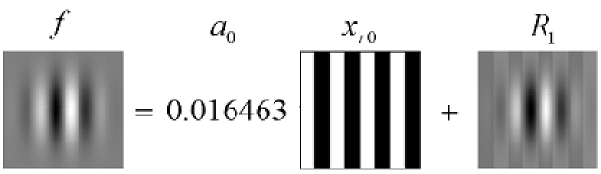

Fig. 7. The result after executing Matching Pursuit

5.4 Saving Combination Coefficients
In this way, we can get a binary version of Gabor Filter in Fig. 8.

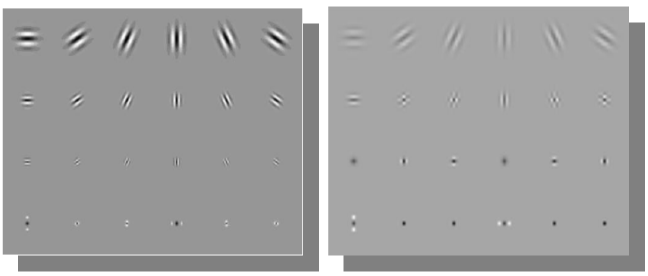

Fig. 8. The Original Gabor Bank and Reconstructed Filter

Most of binary version filters are similar to the original Gabor filters. But the problem may be serious in high frequencies, because a characteristic zone is too small. We use only a few wavelets. So it has an effect of the loss compression, and it makes some error.

5.5 Result
We accomplished tests in the Pentium 4 Dual Core 3.0Ghz system. When a size of a Gabor filter was 32 by 32 and a size of an image was 320 by 240, it took 0.635sec to operate the Gabor filter. In comparison, when applying binary version of Gabor Filter, it took 0.084sec.

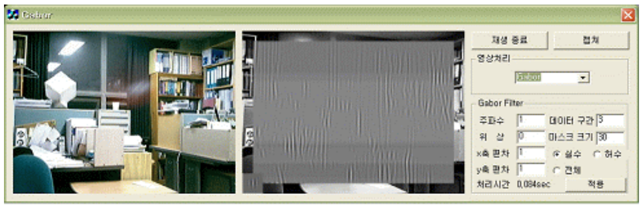

Fig. 9. Applying the Gabor Filter in real time

Application	Result
Gabor Filter	0.635 sec
Binary Version of Gabor Filter	0.084 sec
Speed Up	7.56 times

Tab. 1. Speed up effect of High Speed Gabor Filter

6. Conclusion

The operation speed of binary versions is faster than the original Gabor filter's. And the operation result of binary version is similar to the original Gabor filter's aside from in high frequencies.
The operation speed increase because the original Gabor filter can be described by linear combinations of Haar wavelets. But it takes so long time to execute Matching Pursuit. When executing Matching Pursuit once, there are many convolution operations between the original data and each wavelet. And we must repeat Matching Pursuit more than 100, for its accuracy. So we have need for reducing operation time of Matching Pursuit. But the time is required before applying a filter to an image. If you use the proposed way, you can just obtain fast results. The multiplications are removed by linear combinations and only the additions are remained.

Application	Result
20 wavelets	0.027 sec
30 wavelets	0.062 sec
36 wavelets	0.084 sec

Tab. 2. Speed up effect of High Speed Gabor Filter

It is possible to reduce wavelets. The wavelets which well represent a Gobor filter are chosen. And just chosen wavelets are able to express Gabor Filter. In this case, error is bigger than using all wavelets. But the operation times for resolving a Gabor Filter to wavelets and restoring wavelets to Gabor Filter can be reduced.

You have to remind if you choose only a few wavelets, you may not exactly express Gabor Filters. some trade-off is necessary between accuracy and speed since it may cause some errors. Normally, when wavelets are constituted more than 20 Haar Wavelets, they closely represent some Gabor Filters.

References

[1] Youhong Lu and Morris J.M, "Fast design of Gabor Filters," *IEEE Proceedings. on Engineering in Medinine and Biology Society*, Vol. 2, pp.1334-1335, Nov 1994.

[2] Laura Rebollo-Neira, and David Lowe, "Optimized Orthogonal Matching Pursuit Approach," *IEEE Signal Processing Letters*, pp.137-140, April 2002.

[3] H.R.Karimi, R.J.Maralani, B.Lohman and B.Moshiri, "Haar Wavelet-Based Optimal Control of Time-Varying State-Delayed Systems:A Computational Method, " *IEEE Procedding on Computer-Aided Control Systems Design*, pp. 1528-1533, Oct. 2006

[4] Feng Tang and Hai Tao, "Non-orthogonal Binary Expansion of Gabor Filters with Applications in Object Tracking," *IEEE Workshop on Motion and Video Computing*, pp.24, Feb. 2007.

[5] David Lowe, "Distinctive Image Features from scale-invariant keypoints," *International Journal of Computer Vision*, Nov. 2004

[6] Paul Viola and Michael J. Jones, "Robust Real-time Face Detection," *International Journal of Computer Vision,* May. 2004

[7] P. Czerepinski and B. Macq, "Subband Dictionaries for Low Cost Matching Pursuits of Viseo Residues," *IEEE Trans. Circuits and System for Video Technology*, Oct. 1999