Fast Gabor Filters for Object Recognition of Mobile Robot

Xiaorong WANG * Yingkai ZHAO ** Jinguo LIN ***

** College of Automation Nanjing University of Technology，*
210009 China (e-mail: zdhwxr@163.com).
*** College of Automation Nanjing University of Technology，*
210009 China (e-mail: zhaoyk43@163.com).
**** College of Automation Nanjing University of Technology，*
210009 China (e-mail: zdhljg@163.com).

Abstract: Gabor filters have been used extensively in areas related to feature extraction of images due to their localization in space and bandlimited properties. Since 2-D Gabor filters have more complexity of computation, they have been used in static image decompositions rather than for mobile robots. In an attempt to reduce complexity of computation of 2-D Gabor filters for mobile robots, in this paper, a method of fast Gabor filters is presented. In the method, 2-D Gabor filters are decomposed into 1-D Gabor filters along non-orthogonal axes with different variances first, and those 1-D Gabor filters are recursively implemented, then the image group after fast Gabor filters is extracted feature by Principle Component Analysis（PCA）, last the image would be classified by support vector machine(SVM). Experiment results indicate that, mobile robots can reach recognition rate of more than 92% and speed of quasi real-time image processing of 8 frames per second by the method.

1. INTRODUCTION

Image has many features such as chrominance, intensity, statistics, and space distribution. Image is a large data set, so image processing often accompany large amount of computation. However, image contains much redundant information, so we can extract useful features from image. Object images gotten by CCD camera on robot lie in highly nonlinear due to the various changes by illumination and observation perspective etc. Difficulties of object recognition of mobile robot can be summed as: robustness, accuracy and speedability of the algorithm (Hou, Z.Q. *et al*., 2006). The Gabor wavelets, whose kernels are similar to the two-dimensional (2-D) receptive field profiles of the mammalian cortical simple cells, exhibit desirable characteristics of spatial locality and orientation selectivity (J. G. Daugman, 1988). It is robust to variations due to illumination and observation perspective changes and is one of the most successful approaches for face recognition (K. Sreekar *et al*., 2006), texture feature extraction (B. Francesco *et al*., 2007) and so on. However 2-D Gabor filters have more complexity of computation, they have been used hardly for mobile robots. In the paper a method of fast Gabor filters is presented.

Usually, by convolving a $N \times M$ 2-D Gabor filter *H* to an image *I*, we obtain the image $\tilde{I}$:

$$\tilde{I}(x,y) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} I(x+n, y+m) H(n,m) \qquad (1)$$

By equation (1), calculation of each pixel of image *I* is more than *NM* – 1 multiplication. For an image *I*, calculation is too much to process in real time for mobile robots.

The Gaussian component of the Gabor filter has several useful properties including Fourier symmetry, implying the extent of the filter in the spatial and frequency domains are similar, and separability, which means that a 2-D filter kernel can be separated into two 1-D components. Convolution of 1-D Gabor filter can be calculated by Recursive implementation (Young I. T. *et al*., 1995), which is only associated with about 30 multiplication per pixel (Chen X.G. *et al*., 2007). The huge amount of biometric data makes it mandatory to perform a dimension reduction prior to any processing. Turk and Pentland presented the classical Principal Component Analysis method (PCA), which maximizes the variance over the data, after converting the images into column vectors (Turk M. *et al*., 1991). Through above methods, only the most important features are used. By these features, we can identify objects in images through Support Vector Machine (SVM).

2. GABOR FILTERS

The general form of the complex 2-D Gabor filter is presented in (2) (J. Daugman, 1985). It consists of a 2-D Gaussian function with standard deviation σ , that modulates a spatial sinusoid.

$$gabor(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} e^{j(\omega_x x + \omega_y y)} \qquad (2)$$

Where ω_x and ω_y represent the radial frequencies in the horizontal and vertical directions respectively. For implementation considerations Equation (2) is often split into two parts, the real part and the imaginary part. Equations (3) and (4) present the real and imaginary (cosine and sine) components of the Gabor function.

$$gabor_c(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \cos(\omega_x x + \omega_y y) \tag{3}$$

$$gabor_s(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \sin(\omega_x x + \omega_y y) \tag{4}$$

Fig. 1 illustrates the imaginary part of a Gabor filter example. The central frequency of this filter is equal to 1 pixel and the values of σ was set to 2 pixels.

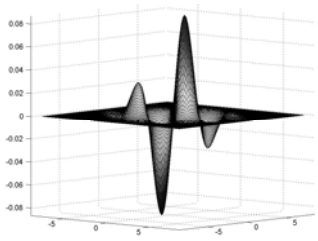

Fig. 1. Spatial-domain odd-symmetric 2-D Gabor filter

The values of the standard deviation of the Gabor can be determined according to the desired frequency and angular frequency-domain bandwidths. Equation (5) provides the relation between the frequency bandwidth B_f, in octaves and the spatial standard deviation σ . f represents the selected central frequency and the cut-off points are set to -6 db. A cut-off point is defined as the location where the filter amplitude reaches a value that is equal to half of its peak magnitude.

$$\sigma = \frac{\sqrt{\ln(2)/2}}{f\pi} \frac{2^{B_f}+1}{2^{B_f}-1} \tag{5}$$

$$\sigma = \frac{\sqrt{\ln(2)/2}}{f\pi} \frac{1}{\tan(B_\theta/2)} \tag{6}$$

Equation (6) describes σ as a function of the desired angular bandwidth B_θ . Similarly, the cut-off points are set to -6 db. The relationship between the frequency and angular bandwidths can be derived by dividing (5) .

3. IMAGE DECOPOSITION

Since the imaginary part of Gabor function is more appropriate for object feature extraction, by rearranging the terms using (3) we obtain the following:

$$gabor_c(x,y) = \left(e^{-\frac{x^2}{2\sigma^2}} \cos(\omega_x x)\right)\left(e^{-\frac{y^2}{2\sigma^2}} \cos(\omega_y y)\right) - \left(e^{-\frac{x^2}{2\sigma^2}} \sin(\omega_x x)\right)\left(e^{-\frac{y^2}{2\sigma^2}} \sin(\omega_y y)\right) \tag{7}$$

This form can be efficiently implemented as it contains sums of separable functions.

It was demonstrated that in order to obtain good frequency and angular discrimination properties for texture segmentation, an angular bandwidth of approximately 30^o coupled with a frequency bandwidth of one octave is recommended (D. A. Clausi, 1996). These guidelines provide a reasonable starting point, however, optimal values are to be determined experimentally on a case by case basis.

In our experiment, the standard deviation σ used was 2, the central frequency used was 1, the original image resolution was 64x64 pixels, angular bandwidth was 30^o, and frequencies were 1, $\sqrt{2}/2$, 1/2, $\sqrt{2}/4$. So the Gabor filter bank is composed of 4×6 Gabor filters.

Young I T. and Vliet L J. present a method of Recursive implementation of the 1-D Gaussian filter in 1995. If the input signal is *IN[n]*, and the output signal after filtering is *OUT[n]*, the course of recurrence can express as:

$$\begin{cases} Temp[n] = a_0 IN[n] + a_1 Temp[n-1] \\ + a_2 Temp[n-2] + a_3 Temp[n-3] \\ OUT[n] = a_0 Temp[n] + a_1 OUT[n+1] \\ + a_2 OUT[n+2] + a_3 OUT[n+3] \end{cases} \tag{8}$$

Where constant $\{a_0, a_1, a_2, a_3\}$ are coefficients of recurrence filtering, which is interrelated with the variance of Gaussian filter. We can calculate convolution of 1-D Gabor filter by using (8), which is only associated with about 30 multiplication per pixel.

4. OBJECTS FEATURES EXTRACTION

An image transformed by above methods is decomposed into an 6×4 image bank. Each image in the bank is 64×64 pixels. Thus, the total of an image feature is $98304(64\times 64\times 24)$, which is too much to recognize objects quickly. Principal component analysis (PCA) is an effective method for choosing the features from image in pattern recognition, which is a process to reduce the data dimension by removing correlations among the data and to describe the image by fewer feature vectors while retaining the necessary information of recognition. 24 images $Q_{i,j}(x,y)$ in the bank are connected by column to 24 4096×1 column vectors X_i . Normalize them as (9):

$$X_i^* = X_i - \left(\sum_{j=1}^{K} X_j \right) / K, i = 1,2,...K \quad (9)$$

Where K is the number of $Q_{i,j}(x, y)$, which was set with 24. From X_i^*, we can get Matrix expansion M:

$$M = \left(\sum_{j=1}^{K} X_j^* \left(X_j^* \right)^T \right) / K \quad (10)$$

Where M is Positive real symmetric matrix of 4096×4096. From M, we can obtain their eigenvalues λ_i and the corresponding eigenvectors v_i. Eigenvectors corresponding to the top values of eigenvalues express general shape of the object, and eigenvectors corresponding to the low values of eigenvalues express details of the object. We can extract eigenvectors corresponding to the top P eigenvalues, which constituted the subspace. The value of P is obtained by using (11).

$$\left(\sum_{j=1}^{P} \lambda_j \right) / \left(\sum_{j=1}^{N} \lambda_j \right) \geq T \quad (11)$$

Where N is the dimension of X_i, T is set by experience and actual needs. In our experiment, T was 90.

5. SVM ALGORITHM

SVM based on the theory of uniform convergence in probability and structural risk minimization (SRM) principle, maps a set of input data to a high-dimensional feature space through a kernel function, avoids overfitting by choosing an optimal separating hyperplane (OSH) in the feature space that maximizes the width of the margin between classes. In general, the hyperplane corresponds to a non-linear decision boundary in the input space and depends only on a subset of the original input data called the support vectors.

We applied SVM to objects recognition of mobile robots, and it is only for one or a few specific objects, which are trained advance. For a practical problem, the suitable kernel function is chosen for constructing SVM algorithm according to the actual data model. We choose a RBF kernel function as (12):

$$k(x, y) = \exp\left(- \|x - y\|^2 / \left(2\sigma^2 \right) \right) \quad (12)$$

If the function meets Mercer's condition, we can avoid the complex calculation in direct solution. At this point, optimal recognition function is:

$$f(x) = \operatorname{sgn}\left\{ \sum_{j=1}^{n} \alpha_i^* y_i k(x, x_i) + b^* \right\} \quad (13)$$

Where n is the number of training samples, α_i^* is Lagange coefficient, and b_i^* is classifier threshold. The introduction of kernel functions in the SVM framework enables one to define the feature space implicitly and thus overcomes the problem of computational burden of explicitly mapping the input data to the higher-dimensional space via non-linear mapping. In the kernel function, some α_i^* corresponding to x_i is zero. According to the demand of real-time image processing for mobile robots, the SVM classifier adopts a "one-to-many" strategy. Given a gallery $\{q_j\}$ of m known individuals and a probe p to be identified, both classifiers will first compute the Gabor feature differences $\{x_j = [d_1 ... d_t ... d_T]\}$ between the probe and each of the gallery images.

6. EXPERIMENTAL RESULTS

Experiments have done in the self-developed mobile robot (CY-1), which is shown in Fig.2. The CPU is P4 3.0GHz, and core store is 512M.

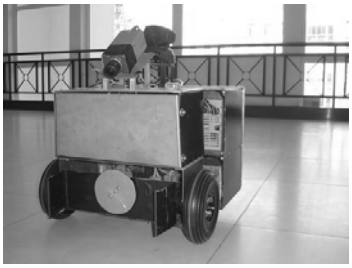

Fig.2. CY-1 Mobile Robot

To evaluate the performance of the proposed method for object recognition of the robot, a database is established, which consists of 648 eight-bit grayscale images of football, toolbox and small robot. The sample images were captured in 3 distances of 0.5m, 1m and 2m in different angles of per 5 degree. 36 images of each subject in different angles and distances are chosen for training, and the remaining 180 images are used for testing. Some sample images are shown in Fig.3.

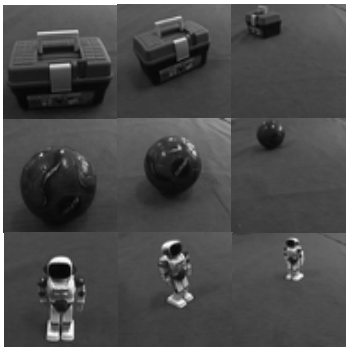

Fig.3. Part of sample images

To test and evaluate the performance of the fast Gabor filter (FGF) method, we compare with other methods of 2D Gabor filter (2DGF), which used 2-D convolution and BP neural

network classification. The comparison results are shown in table 1.

Table 1. Performance of FGF and 2DGF

	sample	FGF	2DGF
Number of samples	football	180	180
	toolbox	180	180
	small robot	180	180
False negative	football	10	7
	toolbox	9	9
	small robot	3	1
False positive	football	8	5
	toolbox	8	4
	small robot	2	0
Accuracy (%)	football	90.0	93.3
	toolbox	90.6	92.8
	small robot	97.2	99.4
	total	92.6	95.2
Average Recognition time(ms)	football	122.6	1584.7
	toolbox	125.1	1566.4
	small robot	109.3	1563.9

From Table 1, we can see, the recognition rate of 2DGF and FGF are all more than 90%, and the average recognition time of FGF is less than 2DGF obviously.

We conducted another experiment to test the property of FGF. In indoors carpet, randomly placed toolbox, ball, small objects, which distance were no more than 3 m. Two methods were used to compare the performance: one is FGF; the other is HSI (Chen F.D. *et al*., 2004), which depended on defining the color regions by thresholds through HSI model. The program of mobile robot can be described as: if the robot find the object, the robot get in front of the object, and if there were no object, the robot will whirl. The experiments were conducted at different times under different illumination conditions. The comparison results are shown in table 2.

From Table 2, we can see, although HSI only spend 27.2ms to process an image frame, it failure 47 times in the experiments. The case of too much failures of HSI is due to its poor robustness. Times of success by HSI are strongly influenced by different times such as morning, afternoon. Although the recognition rate of FGF is about 92.6%, its failure times are zero under consecutive frames.

Table 2. Performance of FGF and HSI

	FGF	HSI
Times of the test	120	120
Times of failure	0	47
Rate of success	100.0%	60.8%
Average process time	130.3 ms	27.2ms

From Table 2, we can see, although HSI only spend 27.2ms to process an image frame, it failure 47 times in the experiments. The case of too much failures of HSI is due to its poor robustness. Times of success by HSI are strongly influenced by different times such as morning, afternoon. Although the recognition rate of FGF is about 92.6%, its failure times are zero under consecutive frames.

7. CONCLUSIONS

We have proposed a Fast Gabor Filters (FGF) method of object recognition. The method is robust to variations due to illumination and observation perspective changes and is one of the most successful approaches for object recognition. The method can be apply to mobile robots, and improve Robots the robustness of object recognition obviously. The average recognition time per frame of the method is about 120ms, and can reach speed of quasi real-time image processing of 8 frames per second. The average time of the method is still longer than that of traditional method, but it can be settled by other ways such as improving the speed of computer, using SSE optimized C++ library.

REFERENCES

B. Francesco and F. Antonio (2007). Evaluation of the effects of Gabor filter parameters on texture classification. In: Pattern Recognition, 40: 3325-3335. Elsevier Ltd, Oxford, United Kingdom.

Chen F.D., Hong B.R. and Zhu Y. (2004). Multi-robots segmentation and identification based on HSI space. In: *Journal of Harbin Institute of Technology*, 36: 928-930.

Chen X.G. and Feng J.F.(2007). Fast Gabor Filtering. In: *Acta Automatica Sinica*, 33: 456-461.Science Press, Beijing, China.

D. A. Clausi (1996). Texture Segmentation of SAR Sea Ice Imagery. In: *Ph.D. Thesis Department of Systems Design Engineering*, University of Waterloo, Ontario.

Hou, Z.Q. and Han C.Z. (2006). A Survey of Visual Tracking. In: *Acta Automatica Sinica*, 32: 603-617. Science Press, Beijing, China.

J. Daugman (1985). Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters. In: *Journal of the Optical Society of America A*, 2: 1160-1169.

J. G. Daugman (1988). Complete discrete 2-D gabor transform by neural networks for image analysis and compression. In: *IEEE Trans. on Acoustics, Speech and Signal Processing*, 36:1169-1179.

K. Sreekar, B. John and P. Sethuraman (2006). Using genetic algorithms to find person-specific Gabor feature detectors for face indexing and recognition. In: *International Conference on Biometrics*,182-191. Springer Verlag, Heidelberg, Germany.

Turk M. and Pentland A. (1991). Eigenfaces for Recognition. In: *Journal of Cognitive Neuroscience*, 3: 71-86.

Young I. T. and Vliet L. J. (1995). Recursive implementation of the Gaussian filter. In: *Signal Processing*, 1995, 44: 139-151.