

Evolutionary Gabor Filter Optimization with Application to Vehicle Detection

Zehang Sun¹, George Bebis¹ and Ronald Miller²

¹Computer Vision Lab. Department of Computer Science, University of Nevada, Reno

²Vehicle Design R&A Department, Ford Motor Company, Dearborn, MI

(zehang,bebis)@cs.unr.edu, rmille47@ford.com

Abstract—Despite the considerable amount of research work on the application of Gabor filters in pattern classification, their design and selection have been mostly done on a trial and error basis. Existing techniques are either only suitable for a small number of filters or less problem-oriented. A systematic and general evolutionary Gabor filter optimization (EGFO) approach that yields a more optimal, problem-specific, set of filters is proposed in this study. The EGFO approach unifies filter design with filter selection by integrating Genetic Algorithms (GAs) with an incremental clustering approach. Specifically, filter design is performed using GAs, a global optimization approach that encodes the parameters of the Gabor filters in a chromosome and uses genetic operators to optimize them. Filter selection is performed by grouping together filters having similar characteristics (i.e., similar parameters) using incremental clustering in the parameter space. Each group of filters is represented by a single filter whose parameters correspond to the average parameters of the filters in the group. This step eliminates redundant filters, leading to a compact, optimized set of filters. The average filters are evaluated using an application-oriented fitness criterion based on Support Vector Machines (SVMs). To demonstrate the effectiveness of the proposed framework, we have considered the challenging problem of vehicle detection from gray-scale images. Our experimental results illustrate that the set of Gabor filters, specifically optimized for the problem of vehicle detection, yield better performance than using traditional filter banks.

I. INTRODUCTION

Motivated by biological findings on the similarity of 2-D Gabor filters and receptive fields of neurons in the visual cortex [1], there has been increased interest in deploying Gabor filters in various computer vision applications. An important property of Gabor filters that has contributed to this is that they have optimal joint localization both in the spatial and frequency domains [1]. Gabor filters have been successfully applied to various image analysis applications including edge detection [2], image coding [1], texture analysis [3][4][5], handwritten number recognition [6], face recognition [7], vehicle detection [8], and image retrieval [9]. Despite the considerable amount of research work on the application of Gabor filters to computer vision problems, their design is mostly performed on a trial and error basis. A filter design method is needed for selecting filter parameters to maximize the discriminating power of the filters. Previous efforts in designing Gabor filters follow two directions: the

“Filter-design approach” and the “Filter-bank approach” [10], [3].

In the “filter-design approach” the filter parameters are chosen by considering the data available, that is, the parameters are appropriate for the problem at hand only. In one of the pioneering studies on the design of Gabor filters conducted by Bovik et al. [11], the peak detection technique was used. In this approach, the center frequency of each filter corresponds to a peak of the power spectrum of the input image. Slightly different from [11], Okombi-Diba et al. [12] implemented a multi-iteration peak detection method for a texture segmentation problem. Dunn et al. [13] investigated an exhaustive search to find the center frequency. The search was guided by a “filter-quality measurement” (i.e., Rician statistical model) that was determined by the sample mean and sample variance of the values of an averaged windowed Fourier transform. This work was based on a bipartite (two-texture) image segmentation problem and required heuristics to find a proper bandwidth. The “filter-quality measurement” was the image-segmentation error and the filter with the lowest error was selected. Due to the exhaustive search, this method is quite time-consuming. A more computationally efficient method was described in [10], [3], using a segmentation-error criterion similar to [13]. The efficiency was gained using a method to calculate the filter output power for all Gabor filters at certain center frequencies simultaneously. It is worth mentioning that this method does not increase the efficiency of designing a single filter.

In the “filter-bank approach” the filter parameters are chosen in a data independent way. Then, a subset of filters is selected for a particular application. Turner [14] used 32 filters (4 frequencies \times 4 orientations \times 2 phase pairs) in a texture discrimination problem. Based on the observation that a constant bandwidth on the logarithmic scale assures the width of the filters to be inversely proportional to their radial frequencies, Jain et al. [4] chose the filter parameters such that the radial frequencies were one octave apart. To reduce the computational burden, a greedy filter selection method was employed using a selection

criterion based on the error between the original image and the one reconstructed by adding together a subset of the filtered images. To reduce the redundancy in the Gabor feature representation, Manjunath et al. [9] proposed a design method to ensure that the half-peak magnitude support of the filter responses in the frequency domain touch each other. For fast image browsing, they implemented an “adaptive filter selection algorithm”, where spectrum difference information was used to select filters with better performance. In the context of handwritten number recognition, Hamamoto et al. [6] optimized the filters by checking the error rate for all possible combinations of filter parameters and then choosing those minimizing the error rates.

Although good performance has been reported in the literature, certain limitations still exist. “Filter-design approaches”, for example, divide the design process into two stages: pre-filter and post-filter. Several pre-filter design approaches have been investigated, however, an explicit methodology for selecting an appropriate post-filter step for a given pre-filter step has not been suggested. Moreover the selection of the bandwidth parameter is done mostly heuristically. The design stage in the “filter-bank approach” is mostly problem-independent. Different pattern classification problems, however, might require selecting an optimum set of features and, consequently, an optimum set of Gabor filters. We would not expect, for example, that a set of Gabor filters optimized for a vehicle classification application (compact car v.s. truck) would work well in a vehicle detection application (vehicle v.s. non-vehicle), since more detailed information is required in the former case than in the later. Many researchers have realized that this is a serious problem and have suggested filter selection schemes to deal with it, however, filters are selected from an original small pool of filters that might not be suitable for the problem at hand (e.g., Hamamoto et al. [6] performed filter selection using a pool of 100 predefined filters). The main issue here is that we are not certain whether or not the optimum set of filters are included in the predefined pool of filters.

A systematic and general evolutionary filter optimization approach that yields a more optimal, problem-specific, set of filters is proposed in this paper. We believe that filter design and selection are not two independent problems and should not be treated separately. In this study, GAs have been integrated with an incremental clustering algorithm in the parameter space to enable Gabor filter optimization. GAs allow searching the space of filter parameters efficiently while clustering removes filters having a high degree of redundancy. The final set of filters is both compact and optimized. To customize the filters for a given problem, an application-oriented fitness criterion is used based on Support Vector Machines (SVMs).

The EGFO approach is suitable for optimizing any number of filters for a given application. It encapsulates the main characteristics of both of the previous two approaches. The search space of our method is much larger than that of the filter-bank approaches, providing a higher likelihood of getting close to the optimal solution as in the case of filter design approaches. Moreover, we represent filter optimization as a closed-loop learning problem. The search for an optimal solution is guided by the performance of a SVM classifier on features extracted from the responses of the Gabor filters.

The rest of the paper is organized as follows: In Section II, we define the Gabor filter optimization problem. Section III presents our evolutionary Gabor filter optimization approach in detail. The statistical Gabor filter feature extraction method and the learning engine used in our experiments are described in Section IV. The proposed framework is tested in Section VI on the challenging problem of vehicle detection. The analysis of our experimental results is given in Section VII. Finally, Section VIII summarizes the main results of the paper and presents possible directions for future work.

II. PROBLEM STATEMENT

We begin with a brief review of Gabor filters. One can refer to Daugman’s seminal paper [1] for more details. The general functional of the two-dimensional Gabor filter family can be represented as a Gaussian function modulated by a complex sinusoidal signal. Specifically, a two dimensional Gabor filter $g(x, y)$ can be formulated as:

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{\tilde{x}^2}{\sigma_x^2} + \frac{\tilde{y}^2}{\sigma_y^2}\right)\right] \exp[2\pi jW\tilde{x}] \quad (1)$$

$$\begin{cases} \tilde{x} = x \cos \theta + y \sin \theta \\ \tilde{y} = -x \sin \theta + y \cos \theta \end{cases} \quad (2)$$

where σ_x and σ_y are the scaling parameters of the filter and determine the effective size of the neighborhood of a pixel in which the weighted summation takes place. $\theta(\theta \in [0, \pi))$ specifies the orientation of the Gabor filters. W is the radial frequency of the sinusoid. A filter will respond stronger to a bar or an edge with a normal parallel to the orientation θ of the sinusoid.

The Fourier transform of the Gabor function in Eq. 1 is given by:

$$G(u, v) = \exp\left[-\frac{1}{2}\left(\frac{(u - W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right)\right] \quad (3)$$

where $\sigma_u = \frac{1}{2}\pi\sigma_x$, $\sigma_v = \frac{1}{2}\pi\sigma_y$. The Fourier domain representation in Eq. 3 specifies the amount by which the filter modifies each frequency component of the input image.

Gabor filter optimization corresponds to selecting the proper values for each of the four parameters in the parameter set

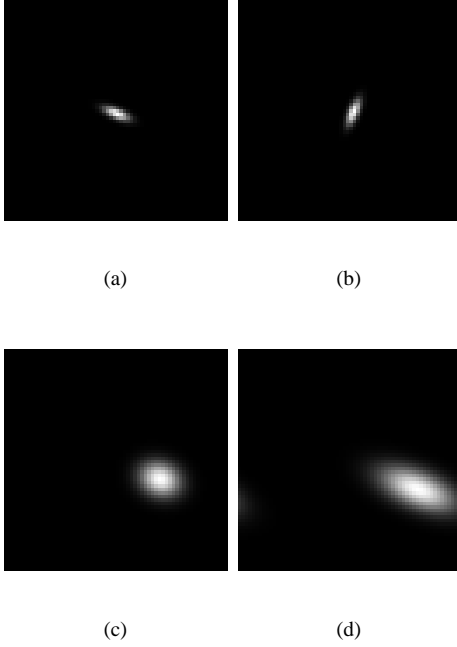


Fig. 1. The Gabor filter with different parameter $\Phi = \{\theta, W, \sigma_x, \sigma_y\}$ in frequency domain (the Fourier transform of the Gabor functions with different parameters). (a) $\Phi_a = \{0^\circ, 0.0961, 0.0204, 0.01219\}$, (b) $\Phi_b = \{0^\circ, 0.3129, 0.06, 0.359\}$, (c) $\Phi_c = \{90^\circ, 0.3129, 0.06, 0.359\}$, (d) $\Phi_d = \{90^\circ, 0.3921, 0.0503, 0.3066\}$

$\Phi = \{\theta, W, \sigma_x, \sigma_y\}$. Gabor filters act as local bandpass filters. Fig. 1 shows four Gabor filters with different parameter settings in frequency domain. The light areas of the power spectrum indicate frequencies and wave orientations. It is obvious from Fig. 1 that different parameter settings will lead to quite different filter responses, an important issue in pattern classification problems. Each filter is fully determined by choosing the four parameters in Φ . Therefore, choosing a filter for a particular application involves optimizing these four parameters. Assuming that N filters are needed in an application, $4N$ parameters need to be optimized. Solving this high dimensional multivariate optimization problem is very difficult in general. In contrast to previous filter design methods, a global optimization approach using GAs is investigated here to deal with this problem.

III. EVOLUTIONARY GABOR FILTER OPTIMIZATION

In this section, we describe the proposed evolutionary Gabor filter optimization approach.

A. A brief review of GAs

GAs are a class of optimization procedures inspired by the natural selection mechanisms[15]. GAs operate iteratively on a population of structures, each of which represents a candidate solution to the problem, encoded as a string of symbols (chro-

mosome). A randomly generated set of such strings forms the initial population from which the GA starts its search. Three basic genetic operators guide this search: selection, crossover and mutation.

B. Encoding and decoding

Using a binary encoding scheme, each Gabor filter is represented by M bits that encode its four parameters. To design N filters, we use a chromosome of length MN bits. Each of the four parameters in Φ is encoded using $n = M/4$ bits as illustrated in Fig. 2.

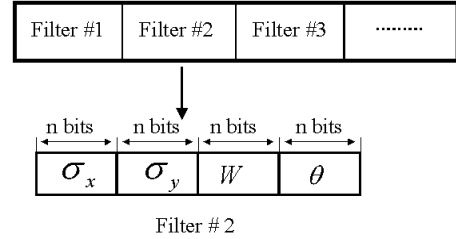


Fig. 2. Encoding scheme

It is worth mentioning that:

- The encoding scheme is quite flexible, and allows us to encode any number of filters by simply varying the length of the chromosome;
- The numbers of bits associated with each parameter need not to be the same, we can make the search for a particular parameter finer or coarser by simply adding or removing bits for this parameter;
- If we need to fix certain parameter(s) using prior knowledge, we can remove the parameter(s) from the chromosome. In this case, the GA will optimize the remaining parameters;

Each of the parameters in Φ has its own constraints and ranges. The encoding/decoding scheme was designed to ensure that the generated filters satisfy these requirements.

The orientation parameter θ should satisfy: $\theta \in [0, \pi)$. If D_θ denotes the decimal number corresponding to the chunk of bits associated with θ (see Fig. 2) then the value of θ is computed by

$$\theta = D_\theta * \pi / 2^n. \quad (4)$$

which always satisfies the range requirement.

W is the radial frequency of the Gabor filter, which is application dependent. Using some prior knowledge, we can limit the range of W into $[W_{min}, W_{max}]$. Then the decoding formula is given by

$$W = W_{min} + (W_{max} - W_{min}) * D_W / 2^n \quad (5)$$

where D_W is the decimal number corresponding to the chunk of bits associated with W . In this study, we have used $W_{min} = 0$ and $W_{max} = 0.5$.

σ_x , and σ_y are essentially the effective sizes of the Gaussian functions and are within the range $[\sigma_{min}, \sigma_{max}]$. The upper limit σ_{max} is determined by the mask width w [16]. A relation between σ_{max} and the mask size w can be obtained by imposing that w subtends most of the energy of the Gaussian. An adequate choice is $\sigma_{max} < w/5$, which subtends 98.76% of the energy of the Gaussian filter. The lower limit can be derived using the *Sampling Theorem*. If the pixel width is taken as our unit step, we cannot reconstruct completely a signal containing frequencies higher than 0.5pixel^{-1} from its samples, which means that any frequency component at $|\omega| > \omega_c = 2\pi(0.5) = \pi$ is distorted. The ω_c is determined by the pixelization, not by the signal. To avoid aliasing, the best we can do is to keep most of the energy of the Gaussian function within the interval $[-\pi, \pi]$. Applying the ‘‘98.86% of the energy’’ criterion, we find $\sigma_{min} > 0.796$. To meet the range constraint ($[\sigma_{min}, \sigma_{max}]$), our decoding scheme follows:

$$\sigma_x = \sigma_{min} + (\sigma_{max} - \sigma_{min}) * D_{\sigma_x} / 2^n \quad (6)$$

for σ_x and

$$\sigma_y = \sigma_{min} + (\sigma_{max} - \sigma_{min}) * D_{\sigma_y} / 2^n \quad (7)$$

for σ_y . D_{σ_x} and D_{σ_y} are again the decimal numbers corresponding to the chunk of bits associated with σ_x and σ_y correspondingly.

C. Filter Selection

During parameter optimization, some of the Gabor filters encoded in a chromosome might end up being very similar to each other or even identical. These filters will result in similar/identical responses, therefore, introducing great redundancy and increasing time requirements. To eliminate redundant filters, we perform filter selection, implemented through filter clustering in the parameter space. An incremental clustering algorithm [17] has been adopted in this paper for its simplicity. A high level description of the clustering algorithm is given below:

1. Assign the first Gabor filter to a cluster.
2. Compute the distance of the next Gabor filter from the centroid of each cluster.
3. Find the smallest distance.
4. If the distance is less than a threshold, assign the filter to the corresponding cluster; otherwise, assign the filter to a new cluster.
5. Repeat step 2-4 for each of the remaining filters.
6. Represent the filters in each cluster by a single filter whose parameters correspond to the cluster’s centroid.

The optimized filters are evaluated using the fitness function defined in Section III-E.

In our implementation, clustering is carried out in the parameter domain. Representing the parameters of a Gabor filter with $\{\theta^n, W^n, \sigma_x^n, \sigma_y^n\}$ and the centroids of the clusters with $\{\theta^i, W^i, \sigma_x^i, \sigma_y^i\}$ with $i \in [1, N]$, where N is the number of currently existing clusters, we assign the filter to the i th cluster only if all of the following conditions are satisfied:

$$\theta^i - \frac{1}{2} \times Thre_\theta \leq \theta^n \leq \theta^i + \frac{1}{2} \times Thre_\theta \quad (8)$$

$$W^i - \frac{1}{2} \times Thre_W \leq W^n \leq W^i + \frac{1}{2} \times Thre_W \quad (9)$$

$$\sigma_x^i - \frac{1}{2} \times Thre_\sigma \leq \sigma_x^n \leq \sigma_x^i + \frac{1}{2} \times Thre_\sigma \quad (10)$$

$$\sigma_y^i - \frac{1}{2} \times Thre_\sigma \leq \sigma_y^n \leq \sigma_y^i + \frac{1}{2} \times Thre_\sigma \quad (11)$$

Otherwise, the filter is assigned to a new cluster. The above conditions are quite strict to make sure that filters falling in the same cluster are very similar to each other. We can always relax the criterion by increasing the predefined thresholds. The following thresholds were used in our experiments:

$\Phi = \{\theta, W, \sigma_x, \sigma_y\}$ are $Thre_\theta = \pi/K$, $Thre_W = (W_{max} - W_{min})/K$, and $Thre_{\sigma_x} = Thre_{\sigma_y} = Thre_\sigma = (\sigma_{max} - \sigma_{min})/K$. Depending on different applications and desired trade-off between model compactness and accuracy, K can be set to different values.

D. Selection, Mutation and Crossover

Mutation is a very low probability operator and just flips a specific bit. It plays the role of restoring lost genetic material. Our selection strategy was cross generational. Assuming a population of size N , the offspring double the size of the population and we select the best N individuals from the combined parent-offspring population. Uniform crossover is used here.

E. Fitness evaluation

Each individual’s fitness will determine whether or not it will survive in subsequent generations. The fitness value used here is the performance of a SVM classifier on a validation set using features extracted from the responses of the selected Gabor filters. In this way, the Gabor filter optimization design is implemented as a closed-loop learning scheme, which is more powerful, more problem-specific, and less heuristic than in previous approaches.

IV. FEATURE EXTRACTION AND CLASSIFICATION

Designing an optimal set of Gabor filters is the first step in building a pattern classification algorithm. Then, we need to extract features using the responses of the selected filters and train a classifier using those features. To demonstrate the proposed filter design approach, redundant statistical Gabor features and SVMs are utilized.

A. Gabor Filter Features

Given an input image $I(x, y)$, Gabor feature extraction is performed by convolving $I(x, y)$ with a set of Gabor filters:

$$r(x, y) = \int \int I(\xi, \eta)g(x - \xi, y - \eta)d\xi d\eta \quad (12)$$

Although the raw responses of the Gabor filters could be used directly as features, some kind of post-processing is usually applied (e.g., Gabor-energy features, thresholded Gabor features, and moments based on Gabor features [18]). In this study, we use moments derived from Gabor filter outputs on subwindows defined on subimages extracted from the whole input image.

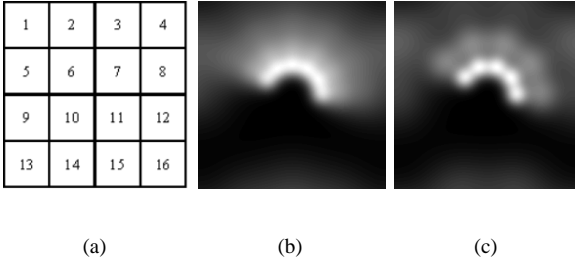


Fig. 3. (a) feature extraction patches; (b) Gabor filter bank with 4 scales and 6 orientations; (c) Gabor filter bank with 3 scales and 5 orientations;

First, each subimage is scaled to a fixed size of 32×32 . Then, it is divided into 9 overlapping 16×16 subwindows. Each subimage consists of 16 8×8 patches as shown in Figure 3(a), patches 1,2,5, and 6 comprise the first 16×16 subwindow, 2,3,6 and 7 the second, 5, 6, 9, and 10 the fourth, and so forth. The Gabor filters are then applied on each subwindow separately. The motivation for extracting -possibly redundant - Gabor features from several overlapping subwindows is to compensate for the error due to the subwindow extraction step (e.g. subimages containing partially extracted objects or background information), making feature extraction more robust.

The magnitudes of the Gabor filter responses are collected from each subwindow and represented by three moments: the mean μ_{ij} , the standard deviation σ_{ij} , and the skewness κ_{ij} where i corresponds to the i -th filter and j corresponds to the j -th subwindow. Using moments implies that only the statistical properties of a group of pixels is taken into consideration,

while position information is discarded. This is particularly useful to compensate for errors in the extraction of the subimages. Suppose we are using $N = 6$ filters. Applying the filter bank on each of the 9 subwindows, yields a feature vector of size 162, having the following form:

$$[\mu_{11}\sigma_{11}\kappa_{11}, \mu_{12}\sigma_{12}\kappa_{12} \cdots \mu_{69}\sigma_{69}\kappa_{69}] \quad (13)$$

B. SVM classifier

SVMs are primarily two-class classifiers that have been shown to be an attractive and more systematic approach to learning linear or non-linear decision boundaries [19] [20]. Given a set of points, which belong to either one of the two classes, SVM finds the hyperplane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. This is equivalent to performing structural risk minimization to achieve good generalization [19] [20]. Given l examples from two classes

$$(x_1, y_1)(x_2, y_2) \dots (x_l, y_l), \quad x_i \in R^N, y_i \in \{-1, +1\} \quad (14)$$

finding the optimal hyper-plane implies solving a constrained optimization problem using quadratic programming. The optimization criterion is the width of the margin between the classes. The discriminating hyperplane is defined as:

$$f(x) = \sum_{i=1}^l y_i a_i k(x, x_i) + b \quad (15)$$

where $k(x, x_i)$ is a kernel function and the sign of $f(x)$ indicates the membership of x . Constructing the optimal hyperplane is equivalent to finding all the nonzero a_i . Any data point x_i corresponding to a nonzero a_i is a support vector of the optimal hyperplane.

Kernel functions, which satisfy the Mercer's condition, can be expressed as a dot product in some space [19]. By using different kernels, SVMs implement a variety of learning machines (e.g., a sigmoidal kernel corresponds to a two-layer sigmoidal neural network while a Gaussian kernel corresponds to a radial basis function (RBF) neural network). The Gaussian radial basis kernel, which is used in this study, is given by

$$k(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\delta^2}\right) \quad (16)$$

Our experiments with different kernels have shown that the Gaussian kernel outperforms the others in the context of our application.

V. VEHICLE DETECTION USING OPTIMIZED GABOR FILTERS

In this section, we consider the problem of vehicle detection from gray-scale images. The first step in vehicle detection is usually hypothesizing the vehicle locations in an image.

Then, verification is applied to test the hypotheses. Both steps are equally important and challenging. Approaches to generate the hypothetical locations of vehicles in images include using motion, symmetry, shadows, and vertical/horizontal edges. Our emphasis here is on improving the performance of the verification step by optimizing the Gabor filters.

A. Vehicle Data

The images used in our experiments were collected in Dearborn, Michigan in two different sessions, one in the Summer of 2001 and one in the Fall of 2001. To ensure a good variety of data in each session, the images were captured at different times of different days and on five different highways. The training set contains subimages of rear vehicle views and non-vehicles, which were extracted manually from the Fall 2001 data set. A total of 1051 vehicle and 1051 non-vehicle subimages were extracted manually(see Figure 4). In [21], the subimages were aligned by warping the bumpers to approximately the same position. However we have not attempted to align the data since alignment requires detecting certain features on the vehicle accurately. Moreover, we believe that some variability in the extraction of the subimages can actually improve performance. Each subimage in the training and test sets was scaled to a size of 32×32 and preprocessed to account for different lighting conditions and contrast using the method suggested in [22].

To evaluate the performance of the proposed approach, the error rates (*ER*) are recorded using a three-fold cross-validation procedure. Specifically, we sample the training dataset randomly three times (*Set1*, *Set2* and *Set3*) by keeping 280 of the vehicle subimages and 280 of the non-vehicle subimages for training. 300 subimages (150 vehicle subimages and 150 non-vehicle subimages) are used for validation during the filter optimization design. For testing, we used a fixed set of 231 vehicle and non-vehicle subimages which were extracted from the Summer 2001 data set.



Fig. 4. Examples of vehicle and nonvehicle images used for training.

VI. EXPERIMENTAL RESULTS

For comparison purposes, we also report the detection error rates using two different Gabor filter banks without optimization: one with 4 scales and 6 orientations Fig.3(b), the other with 3 scales and 5 orientations Fig.3(c). These filter banks were designed by following the method proposed in [9].

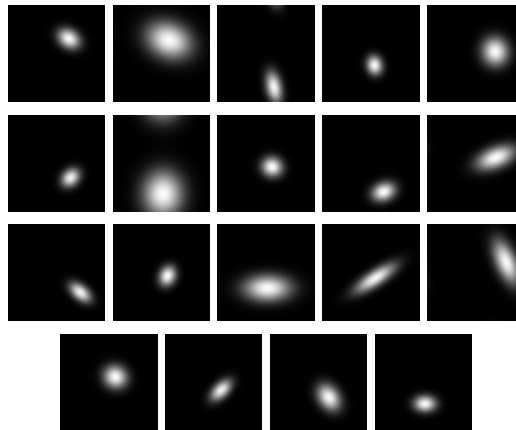


Fig. 5. 19 optimized Gabor filters for the vehicle detection problem with $K = 3$

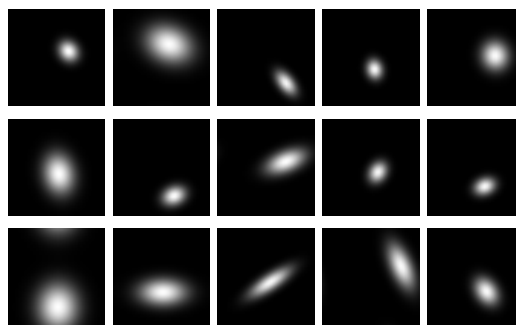


Fig. 6. 15 Gabor filters for the vehicle detection problem with $K = 2$

A. Results

We have performed a number of experiments and comparisons to demonstrate the proposed Gabor filter optimization approach in the context of vehicle detection. First, a Gabor filter bank with 3 scales and 5 orientations was tested using SVMs for classification. Using the feature extraction method described in Section IV-A, the size of each Gabor feature vector was 405 in this experiment. The average error rate was found to be 10.38%, (see Table I). Then, we tested a Gabor filter bank with 4 scales and 6 orientations which yielded features vectors of size 648. The error rate in this case was 9.09% which is slightly better than before.

Second, we used the EGFO approach to customize a group of filters, up to 24, for the vehicle detection problem. We limited the number of filters to 24 to make the comparison with the traditional filter bank design methods fair. The GA parameters used were as follows: population size: 700, number of generations: 100, crossover rate: 0.66 and mutation rate: 0.03. In all the experiments, the GA converged in less than 100 generations. Each parameter in $\Phi = \{\theta, W, \sigma_x, \sigma_y\}$ was encoded using 4 bits. The total length of the chromosome was $384(4 \times 4 \times 24)$, which corresponds to a huge search space (i.e., 2^{384}). The threshold factor K for the clustering was set to 3 in our experiments. The aver-

age error rate in this case was 6.36%, and the average number of customized filters was 19.3. The optimized 19 filters generated for data *Set3* are shown in Fig.5. The individual results from the three data sets are shown in Table I. Fig. 7(a) shows the average detection error rates for all methods.

We also ran the filter optimization method without clustering on the same data sets, using the same parameters. The average error rate was 6.19%, slightly better than that yielded by the method with clustering. Obviously, clustering has the advantage of producing a more compact set of filters (i.e., 19 v.s. 24).

To get an idea regarding the effectiveness of the clustering subcomponent, we performed more experiments using different threshold settings for the factor $k = 2$. The average error rate was 8.23%, and the average number of customized filter was 14.7. The 15 filters generated for data *Set3* are shown in Fig.6.

TABLE I

VEHICLE DETECTION ERROR RATES USING DIFFERENT FILTERS. THE NUMBERS IN THE PARENTHESES INDICATE THE NUMBER OF OPTIMIZED FILTERS

	3×5	4×6	EGFO
Data Set1	10.82%	9.09%	6.93%(21)
Data Set2	11.69%	11.26%	7.79%(18)
Data Set3	8.66%	6.93%	4.33%(19)
Average	10.38%	9.09%	6.36%(19.3)

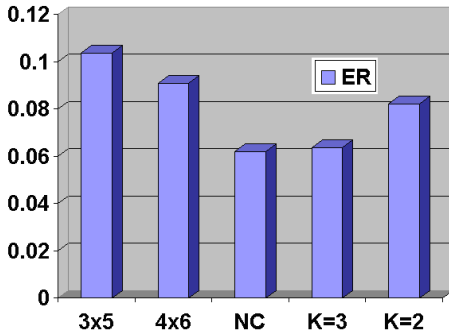


Fig. 7. Vehicle detection error rate. 3×5 : the Gabor filter bank with 3 scales and 5 orientations; 4×6 : 4 scales and 6 orientations; NC: EGFO method without clustering; K=3: EGFO method with K=3; and K=2: EGFO with K=2.

VII. DISCUSSION

To get a better idea about the filter parameters chosen by the EGFO approach, we computed a histogram for each of the parameters (Fig. 8), showing the average distribution of its values over the three data sets. In each graph, the x -axis corresponds

to a parameter from $\Phi = \{\theta, W, \sigma_x, \sigma_y\}$, and has been divided into 10 bins to compute the histogram. The y -axis corresponds to the average number of Gabor filters whose parameters are within a given interval. For example, Fig. 8.a shows the average distribution of θ , where the width of each bin is 18° , given $\theta \in [0 \ 180^\circ)$. The bar associated with the first bin indicates that there were 4 filters (average number over the three training data sets) in the optimized Gabor filter set, whose orientation parameter satisfies: $\theta \in [0 \ 18^\circ)$. The only difference for the rest parameters is the bin size, for instance, the i th bin in Fig. 8(b) corresponds to the interval $[(i - 1) * STEP_W \ i * STEP_W)$, where $STEP_W = (W_{max} - W_{min}/10)$.

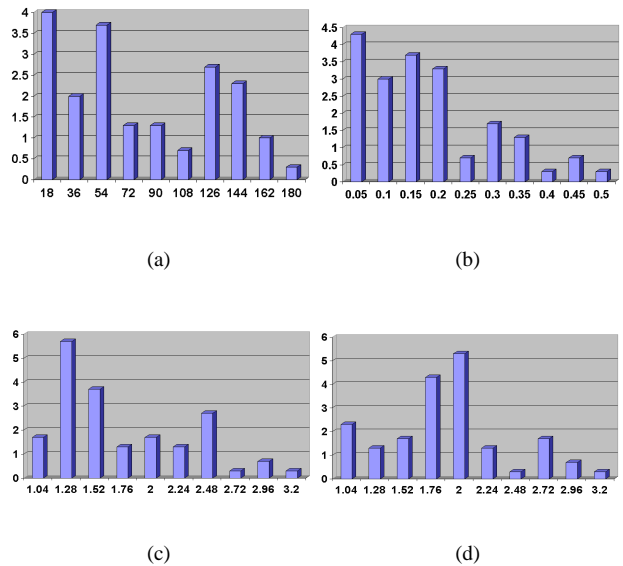


Fig. 8. Distributions of the Gabor filter parameters for vehicle Detection. (a) θ ; (b) W ; (c) σ_x ; (d) σ_y

Several interesting comments can be made based on the experimental results presented in Section VI, the filters shown in Fig. 5, and the parameter distributions shown in Fig. 8:

- The Gabor filters customized using the proposed approach yielded better results in vehicle detection. The most important reason for this improvement is probably that the Gabor filters were designed specifically for the pattern classification problems at hand (i.e., the proposed method is more application-specific than existing filter design methods).
- The orientation parameters of the filters optimized by the GA were tuned to appreciate the implicit information available in vehicle data. Specifically, a Gabor filter is essentially a bar, edge, or grating detector, and will respond most strongly if the filter's orientation is consistent with the orientation of specific features in an image (i.e., bar, edge, etc.). We can see that horizontal, 45° , and 135° structures appear more often in a rear view of a vehicle image, which explains why most of the filter orientations

chosen were close to 0° , 45° , and 135° (see Fig. 8(a)).

- The radial frequency parameters (W) of the filters found by the GA approach were also tuned to encode the implicit information present in vehicle images. Generally speaking, we have more filters with lower radial frequencies than with higher radial frequencies (see Fig. 8(b)). This is reasonable given that vehicle images contain large structures (windows, bumper, etc.), requiring filters with lower radial frequencies.
- The parameters σ_x , σ_y were also tuned to respond to the basic structures of a vehicle. Fig. 8(c) and Fig. 8(d) show that the σ_y parameter has bigger values than the σ_x parameter. Bigger σ_y values implies a wider Gaussian mask in the y direction. This is consistent with the observation that horizontal structures in vehicle images spread more widely than structures in vertical direction.
- By setting the threshold factor to 2, we ended up with 14.7 filters on average. The error rate went up to 8.23% from 6.36%, which is still better than using the traditional Gabor filter bank with 3 scales and 5 orientations. When we build a pattern classification system, among other factors, we need to find the best balance point between model compactness and performance accuracy. Under some scenarios, we prefer the best performance, no matter what the cost might be. Under different situations, we might favor speed over accuracy, as long as the accuracy is within a satisfactory range. The EGFO approach provides a good base for compromising between model compactness and performance accuracy.

VIII. CONCLUSION AND FUTURE WORK

A systematic evolutionary filter optimization method was proposed in this paper. Specifically, Gabor filter parameters were optimized using GAs, followed by further filter clustering in the parameter domain to eliminate redundancy. The proposed approach provides a simple, general, and powerful framework for optimizing the parameters of a family of filters such as Gabor filters or steerable filters [23]. We have tested the proposed method on the challenging problem of vehicle detection. The filters customized by our method yielded better performance than using traditional filter banks. For future work, we plan to evaluate this framework using different data sets, and different types of filters. We also plan to test different filter selection schemes by encoding selection in the chromosome explicitly.

Acknowledgements: This research was supported by Ford Motor Company under grant No.2001332R, the University of Nevada, Reno under an Applied Research Initiative (ARI) grant, and in part by NSF under CRCD grant No.0088086.

REFERENCES

- [1] J. Daugman, "Complete discrete 2-d gabor transforms by neural network for image analysis and compression," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, 1988.
- [2] R. Mehrotra, K. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern Recognition*, vol. 25, pp. 1479–1493, 1992.
- [3] T. Weldon, W. Higgins and D. Dunn, "Efficient gabor filter design for texture segmentation," *Pattern Recognition*, vol. 29, no. 12, pp. 2005–2015, 1996.
- [4] A. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*, vol. 23, pp. 1167–1186, 1991.
- [5] T. Hofmann, J. Puzicha, and J. Buhmann, "Unsupervised texture segmentation in a deterministic annealing framework," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 803–818, 1998.
- [6] Y. Hamamoto, S. Uchimura, M. Watanabe, T. Yasuda, Y. Mitani, and S. Tomota, "A gabor filter-based method for recognizing handwritten numerals," *Pattern Recognition*, vol. 31, no.4, pp. 395–400, 1998.
- [7] K. Chung, S. Kee, and S. Kim, "Face recognition using independent component analysis og gabor filter responses," *IAPR Workshop on machine vision applications*, pp. 331–334, 2000.
- [8] Z. Sun, G. Bebis, and R. Miller, "Improving the performance of on-road vehicle detection by combining gabor and wavelet features," *The IEEE Fifth International Conference on Intelligent Transportation Systems*, September, 2002, Singapore.
- [9] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.
- [10] T. Weldon, W. Higgins and D. Dunn, "Gabor filter desing for multiple texture segmentation," *Optical Engineering*, vol. 35, pp. 2852–2863, 1996.
- [11] A. Bovik, M. Clark, and W. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 55–73, 1990.
- [12] B. Okombi-Diba, J. Miyamichi, and K. Shoji, "Edge-based segmentation of textured images using otimally selected gabor filters," *IAPR Workshop on machine vision applications*, pp. 267–270, 2000.
- [13] D. Dunn, and W. Higgins, "Optimal gabor filters for texture segmentation," *IEEE Transactions on Image Processing*, vol. 4, pp. 947–964, 1995.
- [14] M. Turner, "Texture discrimination by gabor functions," *Biological Cybernetics*, vol. 55, pp. 71–82, 1986.
- [15] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989.
- [16] E. Trucco, and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.
- [17] A. Jain, M. Murty and P. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, pp. 265–323, 1999.
- [18] P. Kuizinga, N. Petkov and S. Grigorescu, "Comparison of texture features based on gabor filters," *Proceedings of the 10th International Conference on Image Analysis and Processing*, pp. 142–147, 1999.
- [19] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.
- [20] C. Burges, "Tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 955–974, 1998.
- [21] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.
- [22] G. Bebis, S. Uthiram, and M. Georgiopoulos, "Face detection and verification using genetic search," *International Journal on Artificial Intelligence Tools*, vol. 9, no. 2, pp. 225–246, 2000.
- [23] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 891–906, 1991.