A Non-Parametric Trainable Object-Detection Model Using a Concept of Retinotopic Sampling

Hirotaka Niitsuma
DENSO IT LABORATORY, INC.
Nikko Shibuya Nanpeidai Bldg. 5th Floor 2-17 Nanpeidai-cho Shibuya-ku Tokyo, 150-0036 Japan
niitsuma@mub.biglobe.ne.jp

Abstract

A retina has a space-variant sampling mechanism and an orientation-sensitive mechanism. The space-variant sampling mechanism of the retina is called retinotopic sampling (RS). With these mechanisms of the retina, the object-detection is formulated as finding appropriate coordinate transformation from a coordinate system on an input image, to a coordinate system on the retina. However, when the object size is inferred by this mechanism, the result tends to gravitate towards zero. To cancel this gravity, the space-variant sampling mechanism is modified to uniform sampling mechanism, but a concept of RS is equivalently introduced by using space-variant weights. This object-detection mechanism is modeled as a non-parametric method.

By using the model based on RS, we formulate a kernel function as an analytical function of information of an object, a position and a size of the object in an image. Then the object-detection is realized as a gradient decent method for a discriminant function trained by Support Vector Machine (SVM) using this kernel function. This detection mechanism realizes faster detection than exploring a visual scene in raster-like fashion.

The discriminant function outperforms results of SVMs using a kernel function using intensities of all pixels (based on independently published results), in face detection experiments over the 24,045 test images in the MIT-CBCL database.

Introduction

The human visual system is a real-time, accurate and non-parametric (don't need tune by programmers) system. In this paper, we propose an object-detection mechanism inspired by the human visual system.

The human visual system is that visual acuity is not uniform, but decreases from the centre of a retina towards the periphery(Hubel & Wiesel 1979). For the decrease of visual acuity, exponential decrease structure is in good agreement with the measured structure of the cat(Schwartz 1980). Tao et al. formulated an object-tracking method with a similar decrease structure(Tao, Sawhney, & Kumar 2002). Tao et al. used Gaussian decrease structure instead of the exponential decrease structure. In the method, the object position represented by Gaussian prior probability density function (PDF). The method based on Gaussian PDF, enables estimation in a maximum a posteriori (MAP) framework using a generalized expectation-maximization (EM) algorithm. This EM based formulation realizes faster detection than exploring a visual scene in raraster-like fashion.

There are orientation-sensitive cells in the retina(Hubel & Wiesel 1959; 1979; Hubel 1988). It has been argued(Daugman 1980; Orban) that a suitable computational model for such cells is represented by Gabor filters(Bastiaan 1981; Gabor 1946; Orban). Smeraldi et al. formulated an object-detection mechanism by using the Gabor filters(Smeraldi & Bigun 2002; Smeraldi 2000). This mechanism realized the real time detection of eyes and mouth in static images. However, this method has many parameters.

In this paper, an orientation-sensitive mechanism is introduced by using the intensity gradient instead of the Gabor filters. And the decrease of visual acuity is introduced by Gaussian decrease structure instead of the exponential decrease structure. By using these modified mechanisms of the retina, we formulate a non-parametric object-detection model.

For an isolated object in an image, SVM's generalization performance either matches or is exceeds that of competing methods(Alvira & Rifkin 2001). For non-isolated objects, Papageorgiou et al.(Papageorgiou & Poggio 1999) proposed a method based on SVM with preprocessing which isolates objects from a background image (Itti & Koch 2001). S.Avidan(Avidan 2001) proposed a method "Support Vector Tracking" (SVT) to find the object in an image using the gradient of the decision function trained by SVM. But, because SVM was able to handle only feature vectors of fixed lengths, longer or shorter object images cannot be handled. Jaakkola et al.(Jaakkola & Haussler) introduced Fisher kernel which can handle feature vectors of various lengths. A discriminant function using the Fisher kernel based on a generative model of our object-detection mechanism, can detect the longer and shorter object images as well.

We report face detection experiments over the 24,045 test images of the MIT-CBCL face database(M.C. for Biological and C. Learning). The performance of SVMs using Fisher kernel based on our model, is compared with SVMs using kernel function using intensities of all pixels, also accord-

ing to results published by other research groups(Alvira & Rifkin 2001). Experiment shows that SVM using the kernel function based on our model, significantly improve performance, thus proving to be a promising technique for pattern recognition on high noise, low resolution images.

Statistical Model

Notation and Model

In this paper, an image is represented as a following set.

$$\begin{aligned} I &= \left\{(x_1, y_1, i_1, \frac{\partial i_1}{\partial \mathbf{x}}), (x_2, y_2, i_2, \frac{\partial i_2}{\partial \mathbf{x}}), ...\right\} \\ &= \{\mathbf{X}_1, \mathbf{X}_2, ...\}, \qquad (1) \\ &\quad \mathbf{X}_n = \mathbf{X}(\mathbf{x}_n) = (\mathbf{x}_n, i_n, \frac{\partial i_n}{\partial \mathbf{x}}), \\ &\quad \mathbf{X}(\mathbf{x}) = (\mathbf{x}, i(\mathbf{x}), \frac{\partial i}{\partial \mathbf{x}}(\mathbf{x})), \end{aligned}$$

where, $\mathbf{x}_n = (x_n, y_n)$, $i_n = i(\mathbf{x}_n)$, $\frac{\partial i_n}{\partial \mathbf{x}} = \frac{\partial i}{\partial \mathbf{x}}(\mathbf{x}_n)$ denote the coordinates of the nth pixel, intensity of the nth pixel, and the intensity gradient at $\mathbf{x}_n$ respectively. $\mathbf{X}$ denotes a state of a pixel at $\mathbf{x}$. $\mathbf{X}_n$ denotes the state of the nth pixel.

The designated objects (for training) are represented as a set of images $S_{train} = \{I_1, I_2, I_3, ...I_{N_{train}}\}$. To ignore background image in the training image I_n, we sample a set of states $J_n = \{X^n_{m_1}, X^n_{m_2}, ...\}$ on the pixels which are chosen from the image I_n with the PDF on the coordinate system $\hat{\mathbf{x}}$,

$$\hat{\Lambda}(\hat{\mathbf{x}}) = \frac{1}{2\pi} \exp(-|\hat{\mathbf{x}}|^2/2). \qquad (2)$$

The coordinate system $\hat{\mathbf{x}}$ is given by a linear transformation of $\mathbf{x}$ so that the center of the object becomes the origin, $\mathbf{x} = 0$, and its size is normalized as unity:

$$\mathbf{x}_n^{train^t} = B_n^{train} \hat{\mathbf{x}}^t + \mu_n^{train^t},$$

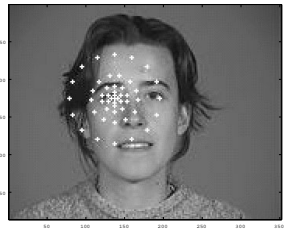
where, $\mathbf{x}_n^{train}$ denotes the original coordinate system on the image I_n. $\mu_{\mathbf{n}}^{\mathbf{train}} = (\mu_{n,x}^{train}, \mu_{n,y}^{train})$ is the position of the trained object in I_n. B_n^{train} is a 2×2 matrix which represents the size and the angle of the trained object in I_n.

The states defined on the coordinate systems $\hat{\mathbf{x}}$ is transformed as

$$\begin{aligned} \hat{X}^n_m &= (\hat{\mathbf{x}}^n_m, \hat{i}^n_m, \frac{\partial \hat{i}^n_m}{\partial \hat{\mathbf{x}}}), \\ &\quad \hat{i}^n_m = i^n_m, \\ &\quad \frac{\partial \hat{i}^n_m}{\partial \hat{\mathbf{x}}} = \frac{\partial i^n_m}{\partial \mathbf{x}_n^{train}} \frac{\partial \mathbf{x}_n^{train}}{\partial \hat{\mathbf{x}}}, \\ X^n_m &= (\mathbf{x}^n_m, i^n_m, \frac{\partial i^n_m}{\partial \mathbf{x}_n^{train}}), \end{aligned}$$

where, $\hat{X}^n_m$ is the state on the coordinate system $\hat{\mathbf{x}}$ of the mth pixel in image I_n. We represent the set of whole states from all training images as

$$J^{train} = \hat{J}_1 \cup \hat{J}_2 \cup \hat{J}_3 ...$$



Retinotopic Sampling Grid used in (Smeraldi & Bigun 2002)

Figure 1: Retinotopic Sampling

where $\hat{J}_n$ is a set of states converted from J_n, as states on the coordinate system $\hat{\mathbf{x}}$.

We apply the Gaussian mixture probability density function (PDF): $p(\hat{\mathbf{X}}|\Theta)$ as generative model of the object,

$$\begin{aligned} p(\hat{\mathbf{X}}|\Theta) &= \sum_{k=1}^{M} p_k N_5 \left(\hat{\mathbf{X}}; \varsigma_{\mathbf{k}}, \Sigma_k\right), \qquad (3) \\ \Theta &= (\varsigma_1, \Sigma_1, ...\varsigma_M, \Sigma_M), \\ N_l(\mathbf{x}; \varsigma, \Sigma) &= \frac{1}{\sqrt{(2\pi)^l |\Sigma|}}, \\ &\exp\left(-(\mathbf{x} - \varsigma) \Sigma^{-1} (\mathbf{x} - \varsigma)/2\right). \end{aligned}$$

The parameter Θ is determined so that likelihood of $p(\mathbf{X}|\Theta)$ for the sampled states J^{train} takes the maximum value. Θ is determined by the method Verbeek et al.(Verbeek, Vlassis, & Krose) proposed.

In the model Tao et al.(Tao, Sawhney, & Kumar 2002) proposed, an appearance model as $p(i|n : \text{pixel number})$ is used. Because, a simple Gaussian mixture model $\sum_{k=1}^{M} p_k N_3((\mathbf{x}_n, i_n); \varsigma_{\mathbf{k}}, \Sigma_k)$ can not represent sharp objects, a PDF for each point is required. Intensity gradient $\frac{\partial i}{\partial \mathbf{x}}$, which represents edge density distribution, is used to detect sharp objects.

Let us consider detection of the trained object in a test image. The object detection is formulated as determining an appropriate coordinate transformation between coordinate system $\hat{\mathbf{x}}$ and $\mathbf{x}$,

$$\mathbf{x}^t = B\hat{\mathbf{x}}^t + \mu^t,$$

where $\mathbf{x}$ is coordinate system on the test image.

$\mu = (\mu_x, \mu_y)$ denotes a position of the trained object in the test image.

$$B = \begin{bmatrix} B_{xx} & B_{xy} \\ B_{yx} & B_{yy} \end{bmatrix},$$

is a 2×2 matrix which represents a size and an angle of the trained object in the test image. The appropriate coordinate transformation is a coordinate transformation which gives maximum likelihood for (5),(6)

Retinotopic Sampling

The designated object is represented by the PDF (3) for one pixel. An image is a set of pixels. To estimate the likelihood

Figure 2: Test image

for a set of pixels, the PDF (3) is applied for sampled pixels. The sampling is done with a Gaussian PDF, like figure 1. The density of sampled pixels is as follows:

$$\hat{\Lambda}(\hat{\mathbf{x}}) = \frac{1}{2\pi}\exp(-|\hat{\mathbf{x}}|^2/2)$$
$$\Lambda(\mathbf{x},\Phi) = \hat{\Lambda}(\hat{\mathbf{x}}(\mathbf{x}))|\frac{\partial\hat{\mathbf{x}}}{\partial\mathbf{x}}| \quad (4)$$

Here $\Phi = (\mu, B)$ denotes the coordinate transformation, $\hat{\Lambda}(\hat{\mathbf{x}})$ is the density of sampled pixel at the coordinate system $\hat{\mathbf{x}}$, $\Lambda(\mathbf{x},\Phi)$ is the density of pixels sampled at the coordinate system $\mathbf{x}$. Let us denote the result of the sampling as $J = \{\mathbf{X}_1, \mathbf{X}_2, ...\mathbf{X}_N\}$. Log likelihood for the set J is defined as,

$$\begin{aligned}
&\log P_{RS}(J|\Phi,\Theta)\\
&= \int \log p(\hat{\mathbf{X}}|\Theta)(\sum_{j=1}^{N}\delta(\mathbf{X}(\hat{\mathbf{X}}) - \mathbf{X}_j)d\hat{\mathbf{X}}\\
&= \sum_{j=1}^{N}\log p(\hat{\mathbf{X}}(\mathbf{X}_j,\Phi)|\Theta)|\frac{\partial\hat{\mathbf{X}}}{\partial\mathbf{X}}|\\
&= \sum_{j=1}^{N}\log p(\hat{\mathbf{X}}(\mathbf{X}_j,\Phi)|\Theta) \quad (5)
\end{aligned}$$

$$\begin{aligned}
&\hat{\mathbf{X}} = (\hat{\mathbf{x}}, \hat{i}, \frac{\partial\hat{i}}{\partial\hat{\mathbf{x}}})\\
&\hat{\mathbf{x}}^t = B^{-1}(\mathbf{x}^t - \mu^t)\\
&\hat{i} = i\\
&\frac{\partial\hat{i}}{\partial\hat{\mathbf{x}}} = \frac{\partial i}{\partial\mathbf{x}}\frac{\partial\mathbf{x}}{\partial\hat{\mathbf{x}}} = \frac{\partial i}{\partial\mathbf{x}}B
\end{aligned}$$

Here $\mathbf{X}$ denotes the state of the pixels on the coordinate system $\mathbf{x}$, $\hat{\mathbf{X}}$ denotes the state of the pixels on the coordinate system $\hat{\mathbf{x}}$, $\hat{\mathbf{X}}$ is regarded as a mapping function from $\mathbf{X}, \Phi$ to $\hat{\mathbf{X}}$: $\hat{\mathbf{X}} = \hat{\mathbf{X}}(\mathbf{X},\Phi)$. $\delta(\mathbf{X}(\hat{\mathbf{X}}) - \mathbf{X}_j)$ represents the amount of information on the coordinate system $\hat{\mathbf{x}}$ for the pixel j on the coordinate system $\mathbf{x}$. $|\frac{\partial\hat{\mathbf{X}}}{\partial\mathbf{X}}|$ is the density ratio of information on the coordinate system $\mathbf{x}$ and $\hat{\mathbf{x}}$. Because the state of the pixels represented by intensity and intensity gradient, $|\frac{\partial\hat{\mathbf{X}}}{\partial\mathbf{X}}| = 1$. This rate shows that amount of information for one pixel does not vary with any linear coordinate transformation. This is the reason why no higher order differentiation of intensity like $\frac{\partial^2 i}{\partial\mathbf{x}^2}$ is not used.

Equivalent Retinotopic Sampling

RS tends to ignore feature points that are not near the center of the sampled region. For example, ears and beard of face are not in the center region. Thus, with RS it is hard to detect objects where such feature points are important. RS becomes sparse near the boundary between an object and the background. Then the size of the object estimated by RS will be incorrect. To overcome the above difficulties, an Equivalent RS (ERS) is proposed. ERS is a method that converts broader sampling to RS.

ERS samples a state $\mathbf{X}$ at points with broader PDF $q(\mathbf{x})$ than $\Lambda(\mathbf{x},\Phi)$. [1] The result of the sampling is expressed as, $J = \{\mathbf{X}_1, \mathbf{X}_2, ...\mathbf{X}_N\}$.. Log likelihood for J is defined as (6). If we denote the coordinate system $\mathbf{u}$, such that a coordinate transformation from $\mathbf{u}$ to $\mathbf{x}$ projects/converts uniform sampling on $\mathbf{u}$, into sampling with density $q(\mathbf{x})$ on $\mathbf{x}$. And by designating the coordinate system $\mathbf{v}$, such that a coordinate transformation from $\mathbf{v}$ to $\hat{\mathbf{x}}$ enables uniform sampling of $\mathbf{v}$ into sampling with density $\hat{\Lambda}(\hat{\mathbf{x}})$ on $\hat{\mathbf{x}}$. Then, log likelihood is

$$\begin{aligned}
&\log P_{ERS}(J|\Phi,\Theta)\\
&= \int \log p(\hat{\mathbf{X}}|\Theta)\\
&(\sum_{j=1}^{N}\delta(\mathbf{u} - \mathbf{u}(\mathbf{x}_j))\delta(i - i_j)\delta(\frac{\partial i}{\partial\mathbf{x}} - \frac{\partial i_j}{\partial\mathbf{x}}))\\
&d\mathbf{v}\cdot di\cdot d\frac{\partial i}{\partial\hat{\mathbf{x}}}\\
&= \sum_{j=1}^{N}\log p(\hat{\mathbf{X}}_j|\Theta)|\frac{\partial\mathbf{v}}{\partial\hat{\mathbf{x}}}||\frac{\partial\hat{\mathbf{X}}}{\partial\mathbf{X}}||\frac{\partial\mathbf{x}}{\partial\mathbf{u}}|\\
&= \sum_{j=1}^{N}\log p(\hat{\mathbf{X}}_j|\Theta)\hat{\Lambda}(\hat{\mathbf{x}}_j)/q(\mathbf{x}_j) \quad (6)
\end{aligned}$$

Definition using a simple probability rate

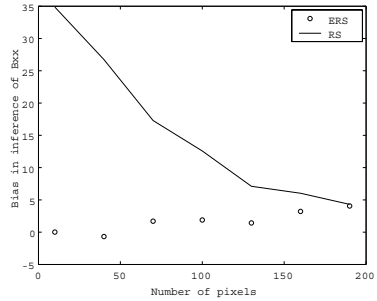
$$\sum_{j=1}^{N}\log p(\hat{\mathbf{X}}_j|\Theta)\Lambda(\mathbf{x}_j|\Phi)/q(\mathbf{x}_j)$$

did not work.

Experiment

In this section, ERS and RS are compared in experiments involving images of vehicles, trained by images from http://www.ai.mit.edu/projects/cbcl/software-datasets/CarData.html. For size inference, a remarkable difference between ERS and RS was seen. Figure 3 shows the bias of inferred size B_{xx} for 100 test images. There is bias

[1] With increasing object size, q =uniform distribution is appropriate.

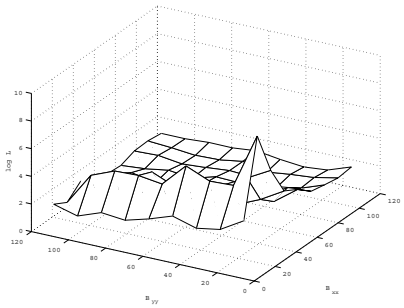


Bias in inference of Bxx:

$$E(B_{xx} - arg\max_{B_{xx}} \log P(I|\Phi))$$

verses number of sampled pixels:N

Figure 3: Bias in inference of B_{xx}



Log likelihood by RS verses B_{xx}, B_{yy}.

Figure 4: Size inference by RS

in inference by RS. The bias for ERS is almost zero. When many pixels are not be sampled, the accuracy of ERS is better than RS. For real-time tracking system, the number of sampled pixels could be $N < 200$. Figure 4 shows an example of inference by RS. In this figure, B_{yy} gives a maximum likelihood that is almost zero. As in figure 4, RS tend to infer that the size is zero. Thus $E(B_{yy} - arg\max_{B_{yy}} \log P(I|\Phi))$ for test images by RS is greater than zero. The log likelihood by RS is up and down because of random sampling. Figure 5 shows an example of inference by ERS. ERS estimates almost accurate object size. Figure 6 shows a situation using gradient descent to determine a vehicle's position

Support Vector Machine

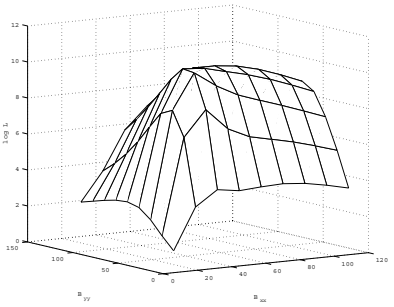
The experiment about ERS shows that, the Fisher kernel based on a generative model of ERS can detect the longer and shorter object images as well.

Using ERS, the Fisher kernel for images J, J' can be defined as follows:

$$K(J, J') = f(\mathbf{s}(J, \Theta)^t Z^{-1}(\Theta)\mathbf{s}(J', \Theta))$$

,where $\mathbf{s}$ is the Fisher score

$$\mathbf{s}(J, \Theta) = (\partial_\Theta \log P_{ERS}(J|\Phi, \Theta))$$



Log likelihood by ERS verses B_{xx}, B_{yy}.

Figure 5: Size inference by ERS

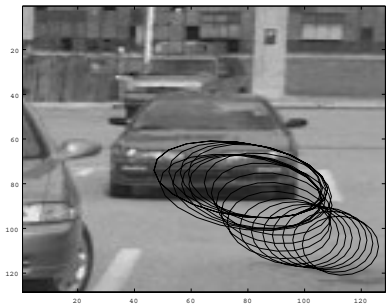


Figure 6: Vehicle tacking using gradient decent method

and Z is the Fisher information matrix: $Z(\Theta) = E_{\mathbf{J}}[\mathbf{s}(J, \Theta)^t \mathbf{s}(J, \Theta)|\Theta]$. f is a liner function ,a polynomial function or exponential function. In all cases for f, a liner function, a polynomial function and exponential function, the accuracy of trained discriminant function is wrong. And the kernel function diverges for some images. To avoid these difficulties, we use following kernel function:

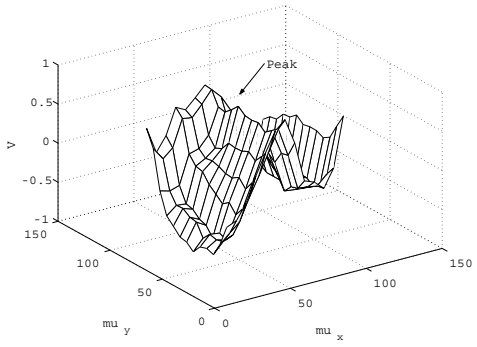
$$K(J, J') = \exp(-\beta\delta\mathbf{s}(J, J', \Theta)^t Z^{-1}(\Theta)\delta\mathbf{s}(J, J', \Theta)) \quad (7)$$
$$\delta\mathbf{s}(J, J', \Theta) = \mathbf{s}(J, \Theta) - \mathbf{s}(J', \Theta)$$

where β is parameter.

Experiment

We present the results of face detection experiment over the images of the MIT-CBCL face database (M.C. for Biological and C. Learning), that consists of low-resolution grey level images ($19 \times 19 = 361$ pixels) of faces and non-faces. A training set of 2,429 faces and 4,548 non-faces are provided. A test set of 472 faces and 23,573 non-faces are provided. All facial images nearly occupy the entire frame; considerable pose changes are represented. The negative test images were selected by a classifier as those that looked most similar to faces among a much larger set of patterns (Heisele, Poggio, & Pontil 2000).

We train our model by using SVMTorch(Collobert & Bengio 2001) with our kernel function for this training data set. ROC curve(Provost & Kohavi 1998) for our model based on SVM is shown in figure 8. Equal Error Rate(EER)



Discriminant Function by SVM V verses μ_x, μ_y for the image in Figure 6.

Figure 7: Fisher kernel

is 7.19 %. SVMs using kernel function using intensities of all pixels, are reported in (Alvira & Rifkin 2001) to yield EERs in excess of 15% on the same database. Our method outperforms the reported result for SVMs.

By using the above Fisher Kernel, the discriminant function V becomes a analytical function of Φ: $V = V(I, \Phi)$. Using a gradient $\frac{\partial V}{\partial \Phi}$, object detection can be formulated as a gradient decent method for V. SVT can handle translations only up to about 10% of the object size. Whereas, our method can handle translations more than 10% . Figure 7 shows the discriminant function $V(\mu_x, \mu_y)$trained with our kernel function for detection of vehicles. The center peak of $V(\mu_x, \mu_y)$ represents correct position of the object in the image. And other peaks are more than 10% away from the correct peak. Computational time to calculate gradient of V on MATLAB is about $(0.3 * (\text{number of sampled pixels in ERS}))$ seconds.

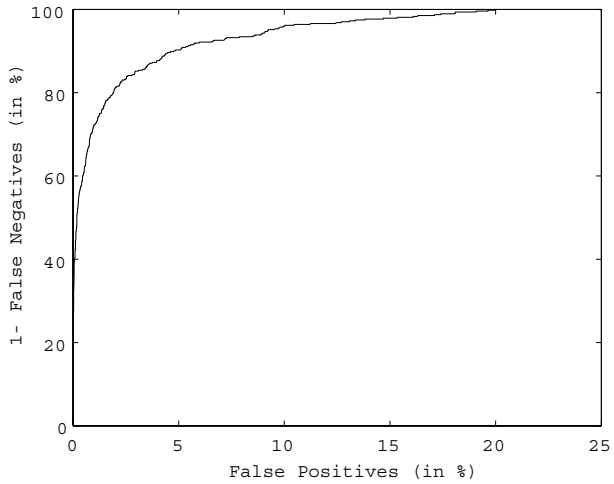
Application

The techniques described in this paper are useful not only in vision, but also in many pattern recognition fields. Using Gaussian filter for certain feature space, similar mechanism to RS can be defined. By this mechanism, an analytical discriminant function in the certain feature space (like object position in image recognition) can be introduced.

One of an application of our method is image recognition-based vehicle control. Since the error of statistical pattern recognition cannot be zero, if it controls a main unit like a brake in an automobile, then the probability of fatal error like running over pedestrians cannot be zero. We think following formulation can avoid such fatal error.

$$\begin{aligned} &\max V \\ &\mathbf{f} < \mathbf{0} \end{aligned} \tag{8}$$

where, V represents the discriminant function. $\mathbf{f} < \mathbf{0}$ represents safer condition. To solve above problem, an analytical discriminant function of certain feature valuable is required. And, for our application, the Fisher kernel function defined in section, is required. Then, a control system based on ker-



EER=7.19%

Figure 8: Face Detection ROC Curve

nel machine (Niitsuma & Ishii 2000),(Dietterich & Wang 2001) is used.

Conclusion

The paper formulates non-parametric statistical object-detection mechanism inspired by the human visual system. The analogy of human visual system, formulates a "natural" statistical model. However, when the object size is inferred by the "natural" statistical model, the result tends to gravitate towards zero. To cancel this gravity, we propose ERS.

By using ERS, we formulate kernel function appropriate for image recognition. The kernel function is Fisher kernel based on ERS. The kernel function gives extension of "Support Vector Tracking" (SVT)(Avidan 2001). SVT can not infer the size of the object. This extension enables size inference.

Experimental results over a test set of 24,045 images show the kernel function outperform a kernel function using intensities of all pixels.

Acknowledgements

The author is grateful to Dr. Shin'ichi Tamura for his helpful discussions and to Siva Kumar.S. for his corrections of the manuscript.

References

Alvira, M., and Rifkin, R. 2001. An empirical comparison of ! ! snow and svms for face ! ! detection. *AI Memo 2001-004 January 2001,CBCL Memo 193.*

Avidan, S. 2001. Support vector tracking. In *CVPR*.

Bastiaan, M. 1981. A sampling theorem for the complex spectrogram, and gabor's expansion of a signal in gaussian elementary signals. *Optical engineering* 20:594–598.

Collobert, R., and Bengio, S. 2001. SVMTorch: Support Vector Machines for Large-Scale Regression Problems. *JMLR*.

Daugman, J. 1980. Two-dimensional spectral analysis od cortical receptive field profiles. *Vision Research* 20:847–856.

Dietterich, T., and Wang, X. 2001. Batch value function approximation via support vectors. In *NIPS*.

Gabor, D. 1946. Theory of communication. *Journal of the IEE* 93:429–457.

Heisele, B.; Poggio, T.; and Pontil, M. 2000. Face detection in still gray images. A.I. memo 1687, Center for Biological and Computational Learning, MIT, Cambridge, MA.

Hubel, D. H., and Wiesel, T. 1959. Receptive fields of single neurones in the cat's striate cortex. *J.Physiology(London)* 148.

Hubel, D. H., and Wiesel, T. 1979. Brain mechanism of vision. *Scientific American* 241(3):150–162.

Hubel, D. H. 1988. Eye, brain and vision. *Scientific American Library*.

Itti, L., and Koch, C. 2001. Feature combination strategies for saliency-based visual attention. *Systems Journal of Electronic Imaging*.

Jaakkola, T., and Haussler, D. Exploiting generative models in discrimative classifiers. In *NIPS*.

M.C. for Biological and C. Learning. MIT-CBCL face database #1. *http://www.ai.mit.edu/projects/cbcl/*.

Niitsuma, H., and Ishii, S. 2000. Learning of minimax strategy by a support vector machine. In *ICONIP*.

Orban, G. Neuronal operations in the visual cortex.

Papageorgiou, C., and Poggio, T. 1999. A pattern classification approach to dynamical object detection. In *ICCV*.

Provost, F. Fawcett, T., and Kohavi, R. 1998. The case against accuracy estimation for comparing induction algorithms. *IMLC*.

Schwartz, E. 1980. Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding. *Visual Research* 20:645–669.

Smeraldi, F., and Bigun, J. 2002. Retinal vision applied to facial features detection and face authentication. *Pattern Recognition Letters*.

Smeraldi, F. 2000. Attention-driven pattern recognit. *Ph.D. Thesis*.

Tao, H.; Sawhney, H.; and Kumar, R. 2002. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Tpami*.

Verbeek, J.; Vlassis, N.; and Krose, B. Efficient greedy learning of gaussian mixture models. *Neural Computation* to appear.