

A Trainable Object-Tracking Method using Equivalent Retinotopical Sampling and Fisher Kernel

Hiroataka Niitsuma

DENSO IT LABORATORY, INC.

Abstract

In this paper, two object detection techniques in computer vision are proposed. The first method is a trainable object-tracking method, based on maximum likelihood. The second method is an extension of support vector machines (SVMs).

The first method is an extension of Retinotopical Sampling (RS). RS is a Gaussian filter with object detection mechanism. The concept of RS was inspired by human saccadic eye movements. However, when the object size is inferred by RS the result tends to gravitate towards zero. In this paper, Equivalent Retinotopical Sampling (ERS), which is an extension of RS, is proposed. ERS is reformulated RS by introducing an amount of information from each sampled point.

The second method is an extension of discriminant function trained by SVMs for object recognition in an image. The discriminant function is formulated as an analytical function of the object position and the object size in an image. The extension is introducing ERS to SVMs.

Introduction

Support vector machines (SVMs) have yielded good generalization performance on wide range of problems. In the pattern recognition field, SVMs have been applied to isolated handwritten digit recognition, object recognition, speaker identification, charmed quark detection, face detection in images, and text categorization. In most of the applications, SVM's generalization performance either matches or is significantly better than that of competing methods.

In this paper, an extension of a discriminant function trained by SVM for object recognition in an image is suggested. By this extension, the discriminant function is formulated as an analytical function of the object position and the object size in an image. This extension realizes a trainable object-tracking method as gradient decent method for the discriminant function like figure 8.

The extension is introducing a concept of Retinotopical Sampling (RS) (Smeraldi & Bigun 2002) to SVMs. The concept of RS was inspired by human eye mechanism (Smeraldi & Bigun 2002). Using the concept of RS, a statistical object model is defined. Then, Fisher Kernel (Jaakkola & Hausler 1999) using this statistical model is defined. This kernel function is an analytical function of the object position and

Copyright © 2003, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

the object size. Then, the discriminant function becomes an analytical function of the object position and the object size.

Statistical Object Model

In this section, the statistical object model is described. And, a maximum likelihood based trainable object-tracking method is discussed. The concept of RS was inspired by human saccadic eye movements and enables a quick recognition of eyes, mouth in static images (Smeraldi & Bigun 2002). RS is a Gaussian filter with object detection mechanism. Tao et al. formulated object-tracking as a similar model (Tao, Sawhney, & Kumar 2002), with the object position represented by Gaussian prior distribution. This model enables estimation in a maximum a posteriori (MAP) framework using a generalized expectation-maximization (EM) algorithm. Moreover, RS has a high calculation speed. In this paper, instead of Gaussian prior distribution, RS is used.

When the object size is inferred by RS the result tends to gravitate towards zero. To avoid this difficulty, Equivalent Retinotopical Sampling (ERS), which is an extension of RS, is proposed. ERS can infer the size of objects more accurately. Using ERS, a trainable static object-tracking method, with essentially only two parameters, has been formulated.

Notation and Model

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

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

Here, $\mathbf{x}_n = (x_n, y_n)$ denotes the coordinates of the n th pixel, $i_n = i(\mathbf{x}_n)$ denotes intensity of the n th pixel. $\frac{\partial i_n}{\partial \mathbf{x}} = \frac{\partial i}{\partial \mathbf{x}}(\mathbf{x}_n)$ is the intensity gradient at \mathbf{x}_n . \mathbf{X} denotes a state of a pixel at \mathbf{x} . \mathbf{X}_n denotes the state of n th pixel.

The designated objects (for training) are represented by the following Gaussian mixture distribution for the state of

a pixel $p(\mathbf{X}|\Theta)$.

$$p(\mathbf{X}|\Theta) = \sum_{k=1}^M p_k N_5(\mathbf{X}; \varsigma_k, \Sigma_k) \quad (2)$$

$$\Theta = (\varsigma_1, \Sigma_1, \dots, \varsigma_M, \Sigma_M)$$

$N_l(\mathbf{x}; \varsigma, \Sigma)$ = 1 dimensional normal distribution

$$N_l(\mathbf{x}; \varsigma, \Sigma) = \frac{1}{\sqrt{(2\pi)^l |\Sigma|}} \exp(-(\mathbf{x} - \varsigma) \Sigma^{-1} (\mathbf{x} - \varsigma) / 2)$$

Here, parameter Θ is determined to give the maximum likelihood for all pixels in the designated images for training. 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

$$\sum_{k=1}^M p_k N_3((\mathbf{x}_n, i_n); \varsigma_k, \Sigma_k)$$

can not represent sharp objects, the probability distribution for each point is required. A distribution for 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. Object detection and object-tracking in the test image is formulated as determining an appropriate coordinate transformation between coordinate system $\hat{\mathbf{x}}$ and \mathbf{x} . Where

- $\hat{\mathbf{x}}$ = coordinate system used in training process to determine Θ from the training images
- \mathbf{x} = coordinate system on the test image.

And, the following linear coordinate transformation is used.

$$\mathbf{x}^t = B\hat{\mathbf{x}}^t + \mu^t. \quad (3)$$

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

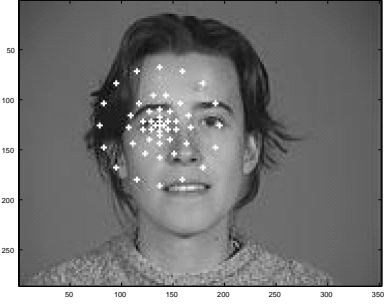
$$B = \begin{bmatrix} B_{xx} & B_{xy} \\ B_{yx} & B_{yy} \end{bmatrix}$$

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 a statistical model defined in the following section: Retinotopical Sampling and Equivalent Retinotopical Sampling.

Method

In the following, two mechanisms (RS, and ERS) to estimate likelihood for certain coordinate transformation are defined.

Retinotopical Sampling The designated object is represented by the Gaussian mixture distribution (2) for one pixel. An image is a set of pixels. To estimate the likelihood for a set of pixels, the distribution (2) is applied for sampled pixels. The sampling is done with a Gaussian probability



Retinotopical Sampling Grid used in (Smeraldi & Bigun 2002)

Figure 1: Retinotopical Sampling

density function, like figure 1. This sampling mechanism is called Retinotopical Sampling. When, the object position μ , and the object size and angle B in the test image are given, the density of sampled pixels is as follows:

$$\begin{aligned} \hat{\Lambda}(\hat{\mathbf{x}}) &= \frac{1}{2\pi} \exp(-|\hat{\mathbf{x}}|^2/2) \\ \Lambda(\mathbf{x}, \Phi) &= \hat{\Lambda}(\hat{\mathbf{x}}(\mathbf{x})) \left| \frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{x}} \right| \\ &= \frac{1}{2\pi} \exp(-|B^{-1}(\mathbf{x} - \mu)|^2/2) / |B| \quad (4) \end{aligned}$$

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} . $\Lambda(\mathbf{x}, \Phi)$ represents the density of sampled pixels on the test image. The result of the sampling is,

$$J = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$$

Log likelihood for the set J is defined as,

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

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

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

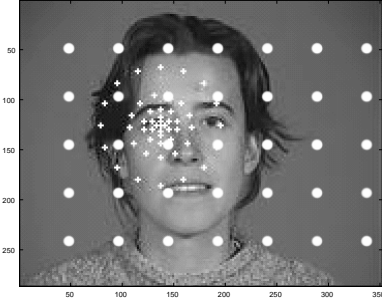


Figure 2: Equivalent Retinotopical Sampling

system $\hat{\mathbf{x}}$, $\hat{\mathbf{X}}$ is regarded as a mapping function from \mathbf{X} , Φ 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 Retinotopical Sampling RS realizes very fast object search in images. However, 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. Figure 2 shows a schematic image of ERS.

ERS samples a state \mathbf{X} at points with broader probability distribution $q(\mathbf{x})$ than $\Lambda(\mathbf{x}, \Phi)$.¹ The result of the sampling is expressed as,

$$J = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}.$$

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

$$\begin{aligned} & \log P(J|\Phi) \\ &= \int \log p(\hat{\mathbf{X}}|\Theta) \end{aligned}$$

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

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

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 in the experiment for the test images used in section .

Φ gives maximum log likelihood represents the object position in the image.

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 6 and 7 shows errors of inferred size for 100 test images. Where B_{xx} and B_{yy} are xx and yy elements of the matrix B . B was defined in the equation (3). B_{xx} represents the object size of the object x direction. B_{yy} represents the object size of the object y direction. 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.

Φ gives maximum log likelihood represents object position in the image. Figure 8 shows a situation using gradient descent to determine a vehicles's position, which is the local maximum of $\log P(I|\Phi)$. Figure 8 shows a situation about facial detection. Here $\log P(I|\Phi)$ by ERS is used.

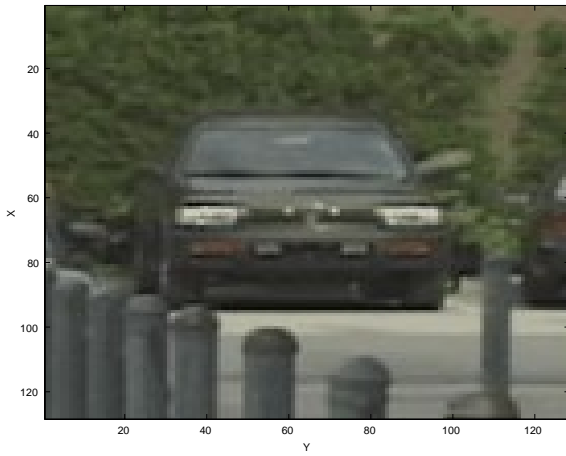
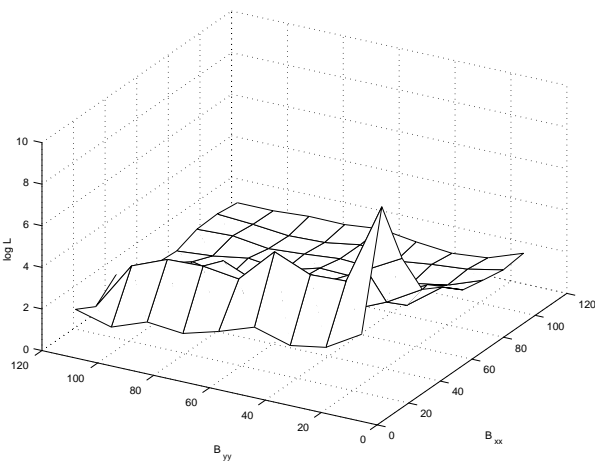
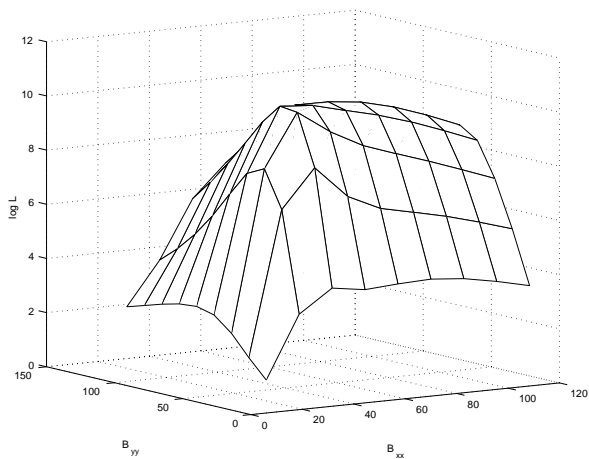


Figure 3: Test image



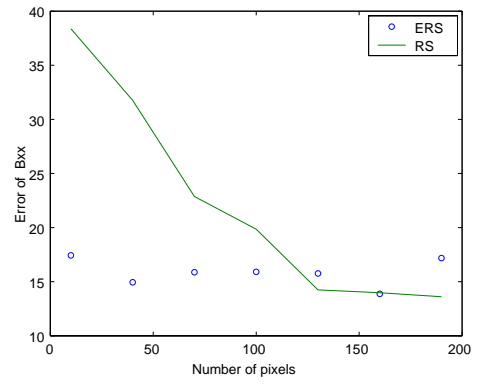
Log likelihood by RS versus B_{xx}, B_{yy} .

Figure 4: Size inference by RS



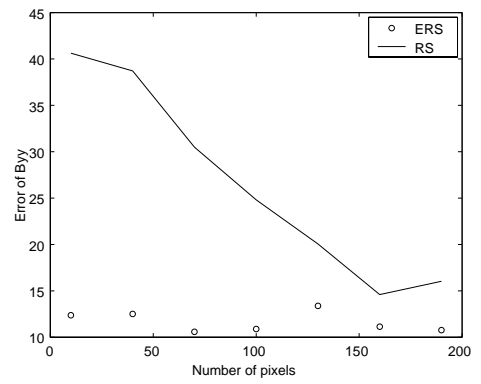
Log likelihood by ERS versus B_{xx}, B_{yy} .

Figure 5: Size inference by ERS



Error of inferred x size: $B_{xx} = \operatorname{argmax}_{B_{xx}} \log P(I|\Phi)$ versus number of sampled pixels: N

Figure 6: Error of x size inference



Error of inferred y size: $B_{yy} = \operatorname{argmax}_{B_{yy}} \log P(I|\Phi)$ versus number of sampled pixels: N

Figure 7: Error of y size inference

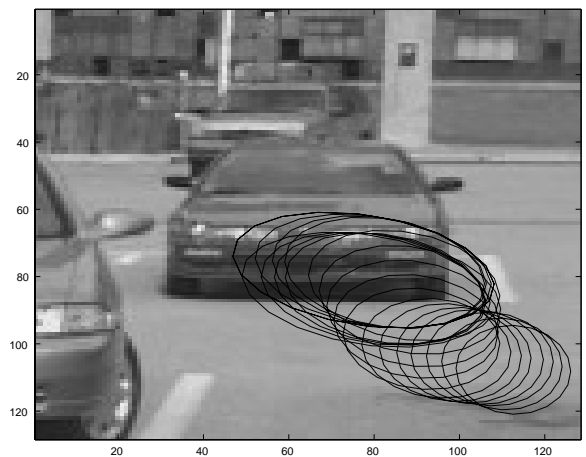


Figure 8: Vehicle tracking using gradient decent method

Fisher Kernel

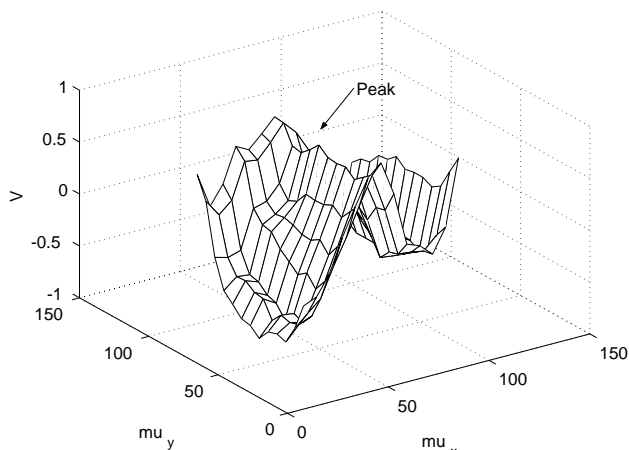
Supposing a situation where none of the designated objects exists in an image, this would not be recognized by a system using only maximum likelihood. Support Vector Machines (SVM's) can provide thresholds for distinguishing object and background and object. The system with the threshold can recognize the state.

For an isolated object in an image, SVM's generalization performance either matches or 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 1999) introduced Fisher kernel which can handle feature vectors of various lengths. The experiment about ERS shows that, a generative model based on ERS can detect the longer and shorter object images as well. By using the Fisher Kernel, the discriminant function V becomes an analytical function of the object position: μ and the object size: B (also input image: I): $V = V(I, \mu, B)$. And a gradient of V , $(\frac{\partial V}{\partial \mu}, \frac{\partial V}{\partial B})$ can be calculated analytically. This gradient calculation part is similar to the SVT (Avidan 2001). Using this gradient, object detection can be formulated as a gradient descent 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 9 shows the Discriminant Function $V(\mu_x, \mu_y)$ trained (Collobert & Bengio 2001) for detection of vehicles. The center peak of $V(\mu_x, \mu_y)$ represents the correct position of the object in the image. And other peaks are more than 10% away from the correct peak.

Figure 11 shows the accuracy of the discriminant function trained (Collobert & Bengio 2001) for face detection. The ROC curve (Provost & Kohavi 1998) is calculated for the same training and test data (Alvira & Rifkin 2001). The result of our system is similar to the results of SNoW (Carlson et al. 1999; Yang & Ahuja 1999) and linear SVM in (Alvira & Rifkin 2001). In this experiment, the number of Gaussians used for the Gaussian mixture model is only 11, due to the limitation of computational time. Figure 11 is a good result for such a small model. By using this above simple statistical model, computational time to calculate the gradient of V on MATLAB is about $(0.1 * (\text{number of sampled pixels in ERS}))$ seconds. The gradient is calculated by numerical differentiation. By replacing numerical differentiation with analytical differentiation and by efficient programming, this system will be close to real-time system.

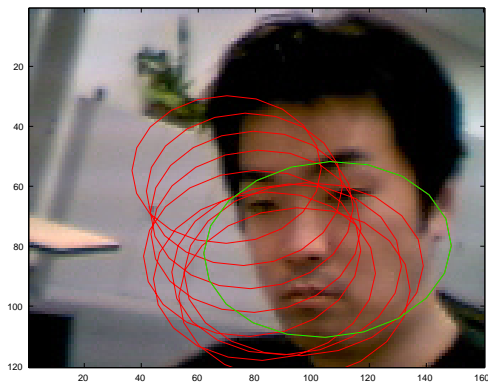
Application

The techniques described in this paper are useful not only in vision, but also in many pattern recognition fields. Using a Gaussian filter for certain feature space, similar mecha-



Discriminant Function by SVM V versus μ_x, μ_y for the image in Figure 8.

Figure 9: Fisher kernel



Training images in <http://www.ai.mit.edu/projects/cbcl/software-datasets/FaceData2.html> are used.

Figure 10: Facial detection using gradient decent method

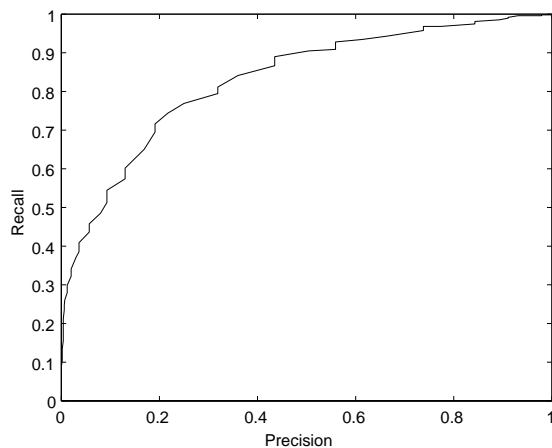


Figure 11: Face Detection ROC Curve

nism 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 about fatal error like running over pedestrians cannot be zero. We think following formulation can avoid such fatal error.

$$\begin{aligned} \max V \\ \mathbf{f} < \mathbf{0} \end{aligned} \quad (7)$$

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

Conclusion

In this paper, two trainable object-tracking method is proposed. The first method is a maximum likelihood based trainable object-tracking method is proposed. The second method is expansion of SVMs.

The first method is formulation of a gradient decent method for log likelihood for object-tracking. This method essentially has two parameters. The first parameter is the number of Gaussians in the Gaussian mixture model. The second parameter is the number of sampled pixels. When the gradient decent method is replaced by Newton method, other parameters are not needed.

The second method is based on the statistical model used in the first method. Using this statistical model, the discriminant function V trained by SVM can be formulated as an analytical function of the object position: μ and the object size: B (also input image : I). By this analytical function, object-tracking can be formulated as a gradient descent

method for the discriminant function $V(I, \mu, B)$. The accuracy of the discriminant function similar to SNoW(Carlson *et al.* 1999; Yang & Ahuja 1999) and linear SVM. Computational time resource our method requires is not large.

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Carlson, A. J.; Cumby, C. M.; Rosen, J. L.; and Roth, D. 1999. Snow user's guide. Technical report, UIUC.
- Collobert, R., and Bengio, S. 2001. SVM Torch: Support Vector Machines for Large-Scale Regression Problems. *Journal of Machine Learning Research* 1:143–160.
- Dietterich, T., and Wang, X. 2001. Batch value function approximation via support vectors. In *Neural Information Processing Systems (NIPS)*, volume 14.
- Itti, L., and Koch, C. 2001. Feature combination strategies for saliency-based visual attention. *Systems Journal of Electronic Imaging* 10:161–169.
- Jaakkola, T., and Haussler, D. 1999. Exploiting generative models in discriminative classifiers. In *Neural Information Processing Systems (NIPS)*, 487–493.
- Niitsuma, H., and Ishii, S. 2000. Learning of minimax strategy by a support vector machine. In *International Conference on Neural Information Processing (ICONIP)*, volume 2, 1432–1437.
- Papageorgiou, C., and Poggio, T. 1999. A pattern classification approach to dynamical object detection. In *International Conference on Computer Vision (ICCV)*, 1223–1228.
- Provost, F. Fawcett, T., and Kohavi, R. 1998. The case against accuracy estimation for comparing induction algorithms. *IMLC*.
- Smeraldi, F., and Bigun, J. 2002. Retinal vision applied to facial features detection and face authentication. *Pattern Recognition Letters* 23:463–475.
- Tao, H.; Sawhney, H.; and Kumar, R. 2002. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(1):75–89.
- Verbeek, J.; Vlassis, N.; and Krose, B. Efficient greedy learning of gaussian mixture models. *Neural Computation* to appear.
- Yang, M.H. Roth, D., and Ahuja, N. 1999. A snow-based face detector. In *Neural Information Processing Systems (NIPS)*, volume 12.