

Adaptive texture representation methods for automatic target recognition

K. Messer*, D. de Ridder[†] and J. Kittler*

*Centre for Vision, Speech and Signal Processing, Dept. of Electronic and Electrical Eng.,
University of Surrey, Guildford, Surrey GU2 5XH, United Kingdom

[†]Pattern Recognition Group, Dept. of Applied Physics, Faculty of Applied Sciences,
Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands

*{K.Messer, J.Kittler}@surrey.ac.uk, [†]dick@ph.tn.tudelft.nl

Abstract

Automatic Target Recognition (ATR) is a demanding application that requires separation of targets from a noisy background in a sequence of images. In this paper, two adaptive methods for describing such a background are proposed which are based on Principal and Independent Component Analysis of sampled image patches. Coupled together with feature selection and outlier detection techniques they enable the ATR system to adapt to certain backgrounds and identify non-standard elements in the images as targets. The methods proposed are compared with a standard wavelet-based approach and are shown to perform somewhat better on a difficult image sequence.

1 Introduction

Automatic Target Recognition (ATR) is concerned with the detection, tracking and recognition of small targets using input data obtained from a multitude of sensor types such as forward looking infrared (FLIR), synthetic aperture radar (SAR) and laser radar (LADAR). Applications of ATR are numerous and include the assessment of battlefield situations, monitoring of possible targets over land, sea and air and the re-evaluation of target position during unmanned missiles weapon firing.

An ideal system will exhibit the properties of a low false positive rate (detection of a non-target as a target), whilst obtaining a high true positive rate (the detection of a true target). This performance should be invariant to the following parameters: sensor noise; time of day; weather types; target size/aspect and background scenery. It should be flexible such that it has the ability to detect previously unseen targets and be able to retrain itself if necessary. It is unlikely that one single system will cope well with all these possible scenarios [2]. The many challenges produced by ATR have been previously well documented in [3], [13] and [1].

In this paper an adaptive ATR system is proposed, which decides how to best distinguish the target from a particular background or clutter. In the bootstrap phase a statistical model of the background is built by using a set of texture filters. In operation, the same features are computed for each new pixel arriving at the sensor input. A statistical test is then applied to this pixels feature vector to determine whether it belongs to the same region as the background or it is an outlier, i.e. a potential target. This general background/target concept is not new with some systems already using such an approach [10].

The novelty of this work is in the techniques applied to obtain a suitable set of filters which ensure that the background/target separation is maximised during training. Usually a standard set of texture filters such as Gabor and wavelet transforms are computed to attempt to model the background and distinguish the target. The problem with this approach is that the filters will respond differently to different textured background. In some circumstances the targets will not be found as outliers and will be lost in the background.

In this system, Principal Component and Independent Component analysis are used to design a set of texture filters from randomly sampled image patches taken from a training image. This ensures that these filters have a mean response when presented with a similar looking texture. If an object with different texture, such as a target, is presented to the filter the resulting response should be non-mean, making its detection as an outlier easier. To further enhance the separation a feature selection stage has been added which selects a subset of the filters which maximise the distance between the background and target.

It will be shown that the adaptive methods of PCA and ICA can work in target recognition applications and that they outperform a more traditional wavelet-based approach. This is demonstrated on two sequences of targets appearing on a sea-scaped background. Feature selection is shown to be a useful tool which helps achieve a higher recognition performance. It will also be demonstrated that using prior knowledge improves the recognition rate.

The rest of this paper is organised as follows: in the next section the target detection algorithm is detailed in full. In section 3 the filter generation methods of PCA and ICA are explained. Section 4 presents the results of the experiments on the two image sequences. Finally, some conclusions are drawn and recommendations for future research are given.

2 The method

2.1 Statistical background modelling

The target detection problem can be viewed as an outlier detection problem. That is, anything that does not normally occur in the background can be seen as a target. Following this idea, our algorithm is based on the following steps:

- describe the background using a statistical model;
- optimise both the model and model size using training data;
- find outliers by deciding, per pixel, whether it is accurately described by the model.

The first step therefore is to generate a statistical model of the background by computing a series of n features for each pixel in a training image. These features can be arbitrarily chosen, but in this work we chose to use a set of Daubechies wavelets, a PCA subspace of image patches and an ICA subspace of image patches. These three feature extraction methods are explained in sections 3.1, 3.2 and 3.3, respectively.

For every pixel in the image we thus obtain a feature vector $\mathbf{f} = [y_0, y_1, \dots, y_n]$. We model background pixels using a Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$:

$$p(\mathbf{f}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (\mathbf{f} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{f} - \boldsymbol{\mu}) \right] \quad (1)$$

To detect possible targets in test frames the same set of n features is generated for every pixel in the image. Each feature vector, \mathbf{f}_{test} , is tested in turn to see whether it belongs to the same distribution as the background or is an outlier (i.e. possible target). This is done using a measure known as the Mahalanobis distance:

$$d_M = (\mathbf{f}_{test} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{f}_{test} - \boldsymbol{\mu}) \quad (2)$$

If d_M is higher than a set threshold d_{thr} the corresponding pixel is considered a target.

2.2 Feature selection

A problem is to find the features that maximise the Mahalanobis distance between background and target. Using many features increases computational burden whilst it can also degrade results due to the so-called *peaking phenomenon*.

More formally the problem of feature selection is defined as follows. Given a set Y of n measurements, $Y_n = [y_1, y_2 \dots y_n]$ we wish to select a sub-set $X_k = [x_1, x_2 \dots x_k]$ of k features, $k < n$, such that each feature x_i is identical to a distinct measurement y_j . We wish to do this such that the set X_k is optimal with respect to a criterion function $J(\chi_k)$ defined over all possible sets of k out of n measurements, i.e.

$$J(X_k) = \max_{\chi_k} (\chi_k) \quad (3)$$

where χ_k denotes a candidate set of features, $\chi_k = \{\zeta_i \mid i = 1 \dots k, \zeta_i \in Y\}$.

A criterion function is required which gives a reliable measure of a candidate feature sub-set. For this application, the maximum Mahalanobis distance between target pixels and the mean of the background pixels was used. The criterion function is evaluated as follows:

1. select a sub-set of features, χ_k , for which the performance has to be evaluated;
2. estimate the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ of the background in this feature space;
3. calculate the Mahalanobis distance of all the target pixel features to the mean of the background pixel features;
4. the maximum distance over all the target pixels is the criterion value $J(\chi_k)$.

This measure was chosen because it ensures that the selected features maximise the distance between target and background. Once a criterion function is defined, feature selection is reduced to a problem that involves searching for the optimal feature sub-set from among all possible feature sub-sets, i.e. the one with the lowest error. For this system the *sequential floating forward search* (SFFS) [12] was implemented which gives near optimal performance but with low computational cost. Once the best features are identified for each feature set cardinality, a graph which plots the criterion function value as a function of feature sub-set size can be used to select an optimal feature sub-set.

BMVC99

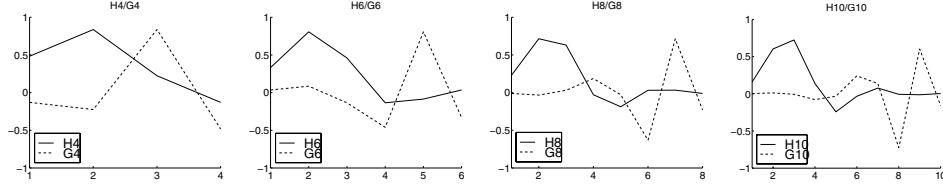


Figure 1: The Daubechies wavelets used.

3 Feature sets

3.1 Wavelets

As a baseline method, a series of Daubechies wavelet transforms is used. The filter coefficients (4-, 6-, 8- and 10-tap) are shown in figure 1. Horizontal and vertical convolution with the 4 wavelet bases gives a set of 16 texture features [4].

3.2 Principal Component Analysis (PCA)

Instead of using pre-defined features such as wavelets, one can also try to describe the background adaptively. In this work, both Principal Component Analysis and a relatively new technique called Independent Component Analysis (which will be discussed in section 3.3) are used.

Principal Component Analysis (PCA, also known as the Karhunen-Loève transform [6]) finds a linear $r.c$ -dimensional base to describe the dataset. It finds axes which retain the maximum amount of variance in the data. To construct a PCA base, firstly N random rectangles of size $r \times c$ are taken from a set of training images. Optionally, these image patches can be weighted using a 2D Gaussian (with $\sigma_x = \frac{c}{4}, \sigma_y = \frac{r}{4}$). These rectangles are then packed into a $r.c$ -dimensional vector \mathbf{x}_i , usually in a row-by-row fashion. This results in a data set \mathbf{X} containing N samples. The principal components are the eigenvectors of the covariance matrix of \mathbf{X} . These are the columns of the matrix \mathbf{E} , satisfying

$$\mathbf{E}\mathbf{D}\mathbf{E}^{-1} = E(\mathbf{X}\mathbf{X}^T) \quad (4)$$

where \mathbf{D} is a diagonal matrix containing the eigenvalues corresponding to the eigenvectors in \mathbf{E} . The PCA filters can be expressed as

$$\mathbf{W} = \mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T \quad (5)$$

The corresponding basis vectors, the columns of \mathbf{A} , are simply the inverse of the filters:

$$\mathbf{A} = \mathbf{W}^{-1} \quad (6)$$

3.3 Independent Component Analysis (ICA)

Independent Component Analysis, or ICA, has gained widespread interest in the last few years. It is a linear model, like PCA, but finds independent components instead of uncorrelated components.

BMVC99

The basic ICA model supposes, like PCA, a linear model for the data. In general, a data vector \mathbf{x} is viewed as being a mixture of a number of unknown sources \mathbf{s} :

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (7)$$

where \mathbf{A} is an unknown mixing matrix, which also has to be estimated. The goal is to find the separating matrix \mathbf{W} :

$$\mathbf{S}\mathbf{P}\mathbf{s} = \mathbf{W}\mathbf{x} \quad (8)$$

where \mathbf{S} is a scaling matrix and \mathbf{P} a permutation matrix. Note that simple PCA can also solve this problem. However, if one demands that the sources \mathbf{s} are not merely uncorrelated, but independent, one has to perform ICA.

As there is no simple mathematical derivation of the independent components, usually iterative algorithms are used to find them. In this work, the `fastica` algorithm proposed by Hyvärinen et al. [8] was used, which tries to identify independent components by the following reasoning.

From the central limit theorem, we know that summing a number of independent, non-Gaussian distributions leads to a Gaussian-like distribution. Therefore, if we project the data we have onto an axis and we obtain a Gaussian-like distribution, we will not have unmixed the various distributions. However, if we find a projection which gives a non-Gaussian distribution, we may infer that we have separated one single distribution. Note that this means that we cannot separate different Gaussian distributions.

The measure most often used for non-Gaussianity is the absolute value of the kurtosis of a distribution, i.e.

$$\kappa_4(u) = E(u^4) - 3(E(u^2))^2 \quad (9)$$

As uncorrelatedness is a necessary (but not sufficient) prerequisite for independence, it makes sense to first pre-whiten, or sphere, the data \mathbf{x} by centering it (i.e. make it zero-mean) and projecting it onto a PCA basis. Note that this pre-processing step is a prerequisite for the fixed point algorithm.

Since ICA is based on higher order statistics, as opposed to PCA which only uses second order statistics, one would expect that ICA might be able to use phase information rather than just frequency magnitude information. It is well-known that phase plays a more important role in image formation than magnitude (e.g. [11]).

3.4 Algorithm parameters

The final algorithm, i.e. texture representation, feature selection and classification, has a number of parameters, each of which influence the final results. These parameters are:

- Data selection (PCA, ICA only):
 - window size (r, c)
 - window weighting (uniform or Gaussian)
 - sample size (N)
- Feature selection:

- number of features to use (χ_k)
- threshold for detecting outliers (d_{thr} , expressed in terms of the maximum Mahalanobis distance in the training set: $d_{thr} = p_{thr} \cdot J(\chi_k)$)

Although in principle these parameters could be optimised automatically, in our experiments these parameters were tuned manually.

4 Experiments

4.1 The data

The proposed system was applied to two sequences, made available by DERA Farnborough. The first sequence consisted of 100 simulated images of an airplane flying over a sea, from left to right. In the first and last 10 images, the airplane was not visible. All 100 images were accompanied by ground truth. An example is shown in figures 2 (a) and (b). The second sequence consisted of 20 simulated images of 5 very small targets (1 pixel each) in a sea, some of which were moving. Finding the targets in this sequence was very hard even for human observers. A ground truth was available only for the first image, shown in figures 2 (c) and (d).

4.2 Sequence 1

In all experiments, the training image used was the 50th (shown in figure 2). From this image, a number of samples were drawn by placing $r \times c$ windows at random positions. As a pre-processing step, the mean was subtracted from each vector [7].

Wavelet, PCA and ICA features were extracted from the data set. Classification was performed as follows: first, an optimal number of features (10) and an optimal corresponding threshold were chosen by hand. Then, for each image, all possible image patches were considered (i.e. overlapping) and projected onto the first 10 basis vectors. In the resulting 10-dimensional space, where each point represents a pixel in the original image, only points with a distance larger than p_{thr} times the maximum Mahalanobis distance in the training set were counted as target points.

As many false negatives were found to lie on the border of the actual target region, the binary ground truth image was dilated twice with an 8-connected neighbourhood. This means that false negatives within a range of 2 pixels from the actual targets are not counted as errors.

For each image, we obtain a number of false positives and false negatives. These numbers are plotted in figures 3. Only the optimal results, obtained for $N = 10000$, $r = c = 4$, Gaussian weighting, and $|\chi_k| = 10$ are shown. Clearly, the results are satisfactory for the application. A simple motion tracking can help rejecting spurious false positives. In some cases the target is lost, but it usually resurfaces within a number of frames.

There is a clearly periodic behaviour in each of the graphs. We believe this to be the result of the target passing through waves, which obscures the high frequency edge which triggers recognition. The periodicity could be the result of the waves rolling in subsequent frames.

The three texture representation methods work almost equally well on this problem. The adaptive methods do not outperform the simple wavelet approach, and there is hardly

BMVC99

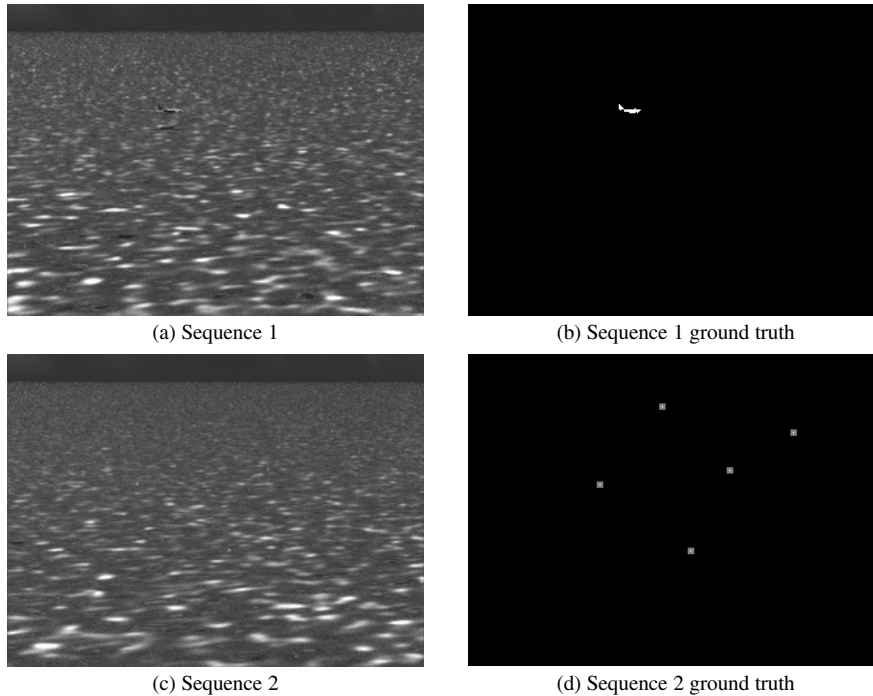


Figure 2: Examples of the original images and ground truths for sequences 1 and 2. For sequence 2, the ground truth has been enhanced: the original size of each of the objects is 1 pixel.

any difference between the PCA and ICA results. This is probably due to the feature selection process, which equalises performance by choosing only features that contribute enough. Also, it is quite easy to find the target in most of the frames, giving all methods the opportunity to perform well.

4.3 Sequence 2

On first application of the techniques to the second sequence, performance was poor for all feature sets. Therefore, the following pre- and post-processing steps were added:

- As the granularity of the texture in the image changes gradually from top to bottom (which is prior knowledge in many real-life situations), it was decided to divide the image into regions (horizontal bands), which were individually processed. For each region, a different optimal window size and set of filters were found.
- As from one frame to the next the objects will stay in approximately the same place, a temporal constraint was added: for a pixel to be classified as a target, it had to be present in at least some of a number of consecutive frames. The segmented images were dilated once using a 8-connected neighbourhood mask, to take moderate

BMVC99

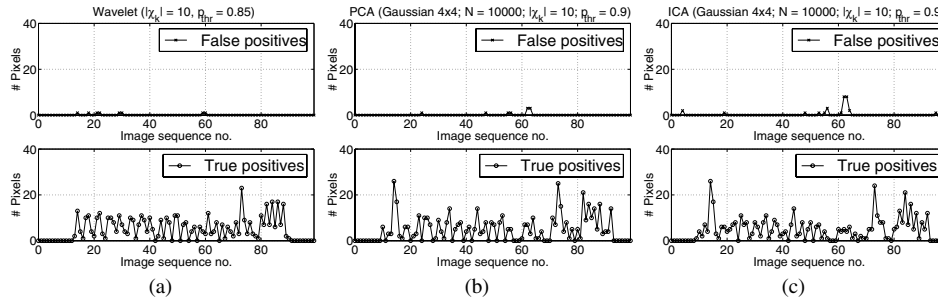


Figure 3: Final classification results for (a) wavelet features, for $p_{thr} = 0.85$; (b) PCA and (c) ICA, both with $p_{thr} = 0.9$.

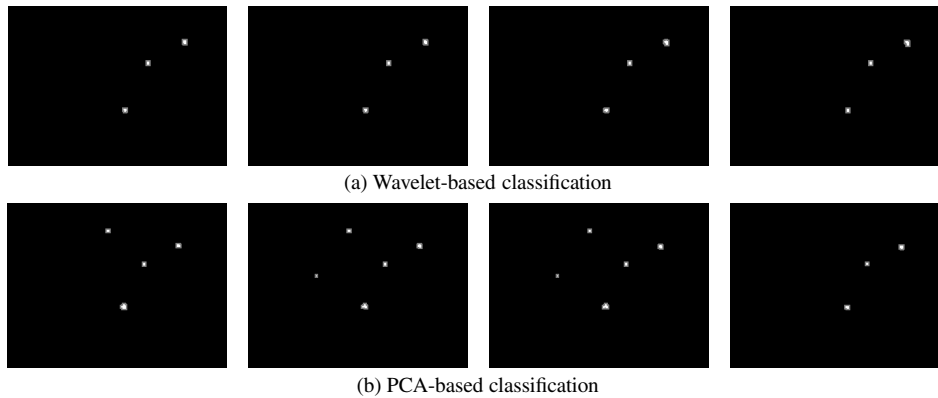


Figure 4: Classification results for the first four frames of sequence 2, using wavelet features (a) and PCA features (b). All objects found were enlarged for presentation purposes.

motion into account, and added. Only pixels with a count above a certain threshold were considered to be target pixels.

This adds a number of new parameters: the number of regions to split the image into (R), the number of consecutive frames considered (F) and a threshold on the number of frames in which the target should appear (F_{thr}). These parameters were chosen as follows: $R = 4$ equally sized regions after deleting the region above the horizon; $F = 7$ and $F_{thr} = 3$.

Only the best results obtained for the wavelet and PCA features are presented. The settings were: $r = c = 4$; $N = 1000$; uniform weighting window; $\chi_k = 10$; $d_{thr} = 0.6$ (wavelets) and 0.9 (PCA). The ICA results were nearly identical to those obtained using PCA, and are not presented. As there was no ground truth available, it is impossible to calculate the number of false and true positives. Therefore, we plot the first four target segmentation results for each of the methods. These are shown in figure 4.

The wavelet features give an optimal result of 3 targets found, whereas the PCA (and ICA) features find all targets in 2 of the frames. In this case, adaptive methods prove better than the wavelet approach.

5 Conclusions and recommendations

We have shown that adaptive methods can work in target recognition applications. In general it is possible to recognise a substantial portion of the target without detecting any false positives. Where false positives do occur, they are usually lost within a small number of frames. Although on Sequence 1 these methods performed no better than a standard wavelet approach, in the more complicated Sequence 2 their performance was much better.

The performance of both PCA and ICA on both sequences seemed to be comparable. We believe this to be partly explained by the fact that the feature selection process will obscure any differences by choosing that subset of features which performs best. In that light, it seems to be preferable to use PCA. Firstly, it has a much lower computational complexity. Secondly, there is a guaranteed ordering in the PCA features, whereas with ICA there is no telling what independent components will be found and in which order.

Some recommendations for further research are:

- At the moment, the system has a number of free parameters which, in the experiments, were tuned manually. It should not be difficult (albeit computationally intensive) to devise a way of optimising them automatically.
- The current system needs a ground truth to train on. One could look at ways of training without the need for a ground truth, e.g. by choosing background model compactness as an optimisation criterion instead of the maximum Mahalanobis distance.
- Investigate the use of different models for the background. At this moment, the use of the Mahalanobis distance implicitly assumes the background can be described as a normally distributed data set. The use of more general data description methods (or outlier detection methods) such as forcedly closed decision boundaries [10], clustering [9], radial basis function networks [5] or local density estimation [14] could improve performance.
- It seems advisable to pre-segment the image roughly into areas of homogeneous texture, as was done for sequence 2. One could, for example, locally measure texture granularity and cluster the measures found into a certain (small) number of clusters. Using the knowledge that the image should be split horizontally, one could then find the region locations automatically.
- A logical extension is the use of motion information. In this work, motion has been only used crudely in the temporal summation of segmented images. True motion estimation might help in finding back moving targets. Having said that, a prerequisite for many motion estimation techniques is to have a good spatial segmentation, which is exactly the problem that has to be solved.
- It is always advisable to use prior knowledge, whenever possible. The knowledge that most natural scenes can be split up horizontally into regions of homogeneous texture (sky, sea) is an example of this. If more such knowledge could be included, this might benefit the application.

Acknowledgements

This work has been carried out with the support of the Defence and Electronics Research Agency, Farnborough, Hants., UK. and was partly supported by the Foundation for Computer Science in the Netherlands (SION) and the Dutch Organisation for Scientific Research (NWO).

References

- [1] B. Bhanu. Automatic target recognition: State of the art survey. *IEEE Transactions on Aerospace and Electronic Systems*, 4:364–379, July 1986.
- [2] B. Bhanu, D. Dudgeon, E. Zelnio, A. Rosenfeld, D. Casasent, and I. Reed. Introduction to the special issue on automatic target detection and recognition. *IEEE Transactions on Image Processing*, 6:1:1–3, January 1997.
- [3] B. Bhanu and T. Jones. Image understanding research for automatic target recognition. *IEEE AES Systems Magazine*, pages 15–22, October 1993.
- [4] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communication on Pure and Applied Mathematics*, 41:909–996, 1988.
- [5] D. de Ridder, K. Schutte, and P. Schwering. Vehicle recognition in infrared images using shared weights neural networks. *Optical Engineering*, 37(3):847–857, March 1998.
- [6] P.A. Devijver and J. Kittler. *Pattern Recognition, A statistical approach*. Prentice-Hall International, London, 1982.
- [7] J. Hurri, A. Hyvärinen, and E. Oja. Wavelets and natural image statistics. In M. Frydrych, J. Parkinen, and A. Visa, editors, *Proceedings of the 10th Scandinavian Conference on Image Analysis, Vol. I*, pages 13–18, Lappeenranta, Finland, 1997. IAPR, Pattern Recognition Society of Finland.
- [8] A. Hyvärinen. A family of fixed-point algorithms for independent component analysis. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 3917–3920. ICASSP, 1997.
- [9] T. Kohonen. *Self-organizing maps*. Springer-Verlag, Heidelberg, Germany, 1995.
- [10] M.R. Moya and D.R. Hush. Network constraints and multi-objective optimization for one-class classification. *Neural Networks*, 9(3):463–474, 1996.
- [11] A.V. Oppenheim and J.S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981.
- [12] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.
- [13] M. Roth. Survey of neural network technology for automatic target recognition. *IEEE Transactions on Neural Networks*, 1:28–43, March 1990.
- [14] K. Urahama and Y. Furukawa. Gradient descent learning of nearest neighbor classifiers with outlier rejection. *Pattern Recognition*, 28(5):761–768, 1995.