TOWARDS THE ADAPTIVE IDENTIFICATION OF WALKERS: AUTOMATED FEATURE SELECTION OF FOOTSTEPS USING DISTINCTION-SENSITIVE LVQ

Jaakko Suutala and Juha Röning

Intelligent Systems Group, Infotech Oulu
University of Oulu
PO Box 4500
90014 University of Oulu, Finland
{jaakko.suutala, juha.roning}@ee.oulu.fi

ABSTRACT

We applied a method called Distinction-Sensitive Learning Vector Quantization (DSLVQ) to the classification of footsteps. The measurements were made by a pressure-sensitive floor, which is part of the smart sensing living room in our research laboratory. The aim is to identify walkers based on their single footsteps. DSLVQ is an extended version of Learning Vector Quantization (LVQ), and it can be used for automated feature scaling and selection during the training of an LVQ codebook. The method shows improvements in the classification accuracies compared to a standard LVQ. In addition, due to its capability of automated input pruning, discarding the non-informative features, it was able to detect automatically the most significant features from a large set of features. This is important in an adaptive identification system, where the informative features might change.

1. INTRODUCTION

This paper describes experiments on automated feature selection during the training of Kohonen's LVQ [1] codebook for footstep classification. A method called Distinction-Sensitive Learning Vector Quantization (DSLVQ), proposed in [2], was used. Automatic feature scaling is based on feature relevance measurements. During the learning process of codebook vectors, a weight vector is also determined based on the relevance of single features. Non-informative features get small weights and are automatically discarded from the calculation of the weighted distance metric between an LVQ codebook vector and the input sample. When building an adaptive identification method, where an aware environment can learn and react to the behaviour of occupants, it is essential for the system to be able to choose automatically the most informative features available.

Earlier footstep identification has been studied in a couple of projects. In [3] and [4], identification based on small area force sensors is presented. Our work is based on measurements from a large pressure-sensitive floor, and we utilized such methods as Hidden Markov Models (HMM) [5] and standard LVQ [6]. Recently, we also proposed a very reliable method for identification, which combined independent LVQ classifiers to make a decision based on multiple footsteps. The method includes an option to reject samples, that cannot been classified reliably to any of the known classes. The identification method is presented in [7].

The DSLVQ method can improve the classification of single footsteps compared to our earlier research. Naturally, it can also reduce the classification error in the method of multiple footstep identification, where multiple DSLVQ classifiers can be used to make a joint decision. Furthermore, DSLVQ provides more adaptiveness due to its capability of feature ranking during operation. For example, if a new person (unknown to the system) enters the room, and we are able to detect her as a new person by using the reject-optional classifier [7], the system can autonomously select an optimal subset of input features from a large set during the training of a new classifier.

The outline of the paper is as follows. In the next section, related work on gait-based identification is presented. Section 3 describes details of the sensor material and the floor setting used in our experiments. A brief theory of footstep classification and an automatic feature selection method is presented in section 4. In section 5, experimental results using the algorithm are shown, and finally, in the last section, the conclusions are presented.

2. RELATED WORK

Gait-based identification methods provide a flexible and natural way to model humans in smart environments without any wearable sensors. Most of the works have been based on the use of video cameras with machine vision methods. Identification in these systems is based on modeling the sequence of walking stride using consecutive images from a side-view camera. Different features are calculated from the posture and limb positions of the subject and also from the frequency and phase presentation of walking [8]. Temporal-Templates and eigenspace transform with canonical space transformations [9] and continuous HMM [10] have been applied for modeling more dynamic properties of moving objects.

Although vision-based systems provide reliable methods, which can capture the dynamic properties of human walking, they suffer from the differences in light conditions and background movements. In addition, the devices are not always hidden from the user, and the occupant may hence have a feeling of the "big brother watching". The hidden sensory system mounted on the floor provides a transparent way to model human motion and to perform identification. Furthermore, transparency fulfils one requirement for ubiquitous computing and calm technology in smart spaces, which is not to disturb the user but to quietly support her [11]. However, the information provided by the floor-based system is more limited than that obtained with a vision-based system. Instead of getting a dynamic model of the whole body, we only get the profile of single footsteps, and some information about the distance and time between them.

In SmartFloor [3] and ActiveFloor [4], footstep identification was accomplished by utilizing small force plates, whose measurements were done by load cells installed at each corner of the steel plate. SmartFloor consisted of 4 sensors and ActiveFloor of a group of 16 force plates. The sensors measure the *ground reaction force* (GRF) caused by the weight and inertia forces of the body. The vertical component of the GRF profile was used in identification along with the Nearest-Neighbour classifier and HMM. Besides identification and tracking, force plates have been used to detect and classify simple activities of human body movements, such as crouches, and jumps as well as standing up and sitting down [12].

UbiFloor [13] uses 144 low-cost and simple ON/OFF switch sensors, which are 14x2.5cm in size. The sensors are placed horizontal to the walking direction, so that a group of 4 sensors cover a single footstep. Feature extraction is based on the use of both single footstep and walking features calculated from 5 consecutive footsteps on the floor. The Multi-Layer-Perceptron Neural Network was used as an experimental identification method.

The combination of a camera-based system and GRF floor sensors has also been studied. In the work [14], a human gait-based authentication system was developed to be used as a surveillance system. The system showed very reliable recognition, but more flexible settings may be needed in intelligent environments.

All the methods of footstep identification presented in this section are based on the use of off-line trained classifiers. In other words, once a classifier has been trained, it remains static. Our approach here is to make the classification methods more adaptive.

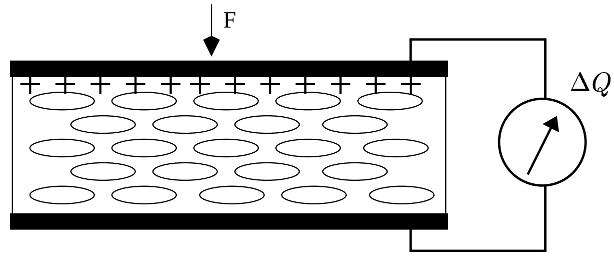

Fig. 1. The cross-section of EMFi. An external force affecting the film's surface causes a change in its thickness, resulting in a change in the charge between the metal layers.

3. EMFI FLOOR

Instead of using force plates measuring the GRF profile, our floor setting consist of long sensor stripes of material called ElectroMechanical Film (EMFi) [15]. EMFi is a thin, flexible, inexpensive material, which consist of cellural, biaxially oriented polypropylene film coated with metal electrodes. EMFi material is used in many commercial applications, such as keyboards, microphones and loudspeakers. When EMFi is used as a sensor, an external force affecting its surface causes a change in the film thickness, resulting in a change between the metal layers. This charge can be detected as a voltage. In figure 1, the cross-section of EMFi is presented.

We have used EMFi material as a wide area floor sensor, which is installed in our research laboratory as a part of smart living room covering a 100 square meter area. The EMFi floor contains 34 horizontal and 30 vertical stripes, which are 30 cm wide. The sensor material is placed under the normal floor, where it make up a 34x30 matrix with a cell size of 30x30 cm (see fig. 2).

The advantage of using long stripes instead of small squares is that there are a smaller number of wires and a smaller number of channels to process. Using this setting, we have only 64 channels to process, while small squares 30x30 cm in size would result in over a thousand pieces. However, the use of a sensor stripe matrix, poses challenges in signal processing. For example, when multiple persons are being simultaneously tracked on the floor, pressure events overlap even when the people are far from each other. Moreover, it is difficult to get "high-quality" footsteps for identification, if the person is walking freely around the room, due to the fact that a given footstep impact may fall on multiple sensor stripes.

Each sensor stripe produces a continuous signal that is streamed into a PC (Pentium 1700 MHz, 256 MB main

Fig. 2. The setting for EMFi sensor stripes under the laboratory's normal flooring. Some stripes are artificially visualized in the picture.

memory). The analogous signal is sampled at a rate of 100 Hz, using a National Instruments DAQ card (PCI-6033E) containing an amplifier. At the first phase, the digitalized signal is saved on a disk for analysis.

4. DISTINCTION-SENSITIVE LVQ

The LVQ [1] is an effective and simple distance measurement based statistical classification method, where labeled codebook vectors are trained in a supervised manner in an N-dimensional feature space. Then, Euclidean distance is used to classify an unknown sample to the closest codebook vector. In order to be successful, however, LVQ needs proper feature selection and scaling at the pre-processing stage. Due to the distance metric, it treats all features equally, regardless of the fact that some of them might be more informative than the others. Therefore, many automated feature ranking methods have been proposed to overcome this problem of standard LVQ. In [16], the relevance LVQ based on an LVQ1 training algorithm is introduced, and [17] provides a more stable solution derived from Generalized LVQ [18]. DSLVQ [2] is otherwise quite similar to Generalized Relevance LVQ, but it is based on an LVQ3 training algorithm.

DSLVQ uses a weighted distance function to scale and rank features during the training. The weighted Euclidean distance function w_{dist} is presented, as follows,

$$w_{dist}(\mathbf{w},\mathbf{x},\mathbf{m}) = \sqrt{\sum_{n=1}^{N} [\max(0,w_n)(x_n - m_n)]^2}, \quad (1)$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_N) \geq 0$ is a weight vector, $\mathbf{x}$ is an input feature vector and $\mathbf{m}$ a codebook vector.

During the training of DSLVQ, the codebook vectors are updated in a similar manner as with LVQ3. Moreover, the weight vector is adapted in parallel based on the relevance of single features in the training samples. In iteration t, the two closest codebook vectors are $\mathbf{m}_j(t)$ and $\mathbf{m}_i(t)$, which belong to the same and to a different class compared to the input sample $\mathbf{x}$. The input sample $\mathbf{x}$ must also fall into a window of certain width [1]. Then, the weights are adapted with equations:

$$\mathbf{w}(t+1) = norm(\mathbf{w}(t) + \alpha(t)[\mathbf{nw}(t) - \mathbf{w}(t)]), \quad (2)$$

where

$$nw_n(t) = norm\Big(\frac{d_{i_n}(t) - d_{j_n}(t)}{\max(d_{i_n}(t), d_{j_n}(t))}\Big) \quad (3)$$

$$norm(\mathbf{y}) = \frac{\mathbf{y}}{\sum_{n=1}^{N} |y_n|} \quad (4)$$

$$d_{k_n}(t) = |x_n(t) - m_{k_n}(t)|, \qquad k \in \{i, j\}. \quad (5)$$

In Eq. (5), the distance for a single feature, $d_{k_n}(t)$, is calculated between all the features in an input sample $\mathbf{x}(t)$. Also, both of the closest codebook vectors $\mathbf{m}_i(t)$ and $\mathbf{m}_j(t)$ are examined. The normalizing function $norm(\mathbf{y})$ (Eq. (4)) is a scaling transformation for the vector $\mathbf{y} = (y_1, y_2, \ldots y_n)$. The scaling factor results in a vector where the sum over all elements is one. Normalization of both $\mathbf{w}$ and $\mathbf{nw}$ (Eq. (2)) provides a certain amount of credits to share between the relevant input-features. A new weight vector $\mathbf{nw}$ is calculated in every learning iteration t, according to an evaluation function for single features. The normalization process causes the scaling factor to be larger if the number of relevant features is small. The weight vector $\mathbf{w}$ is then shifted toward $\mathbf{nw}$, depending on the learning factor α (Eq. (2)). The Evaluation function compares single feature distances between the training vector $\mathbf{x}(t)$ and the two closest codebook vectors $\mathbf{m}_i(t)$ and $\mathbf{m}_j(t)$. The new weight value nw_n is larger if the feature n is closer to the corresponding value of the proper codebook vector (m_{j_n}) and further away from the value of the other codebook vector (m_{i_n}). Division by the maximum distance (Eq. (3)) makes the process independent of proper normalization of the features.

As a result, weight values become very small for non-informative features, and the influence of these features is reduced in the weighted distance metric when classifying unknown samples. The learning rate α for the weights can be set as the learning rate in LVQ3 training. However, with a large number of codebook vectors the learning rate should be smaller for the weights than for codebook training, because the weights are updated more often than the average codebook vector.

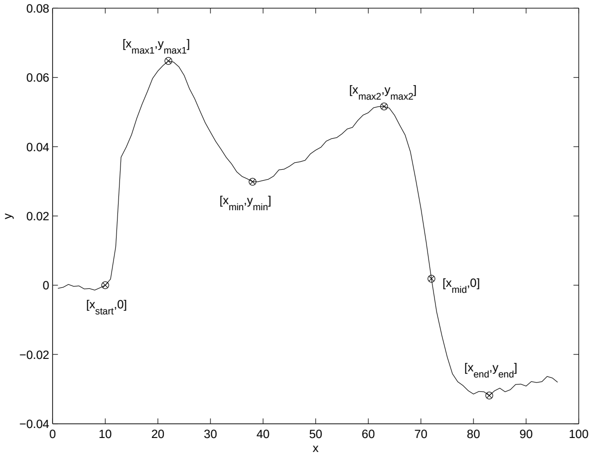

Fig. 3. The coordinate points derived from each footstep.

5. EXPERIMENTAL RESULTS

The experimental data set consisted of footsteps from eleven different walkers, including about 200 footsteps of both feet (about 40 footsteps / person). The footsteps were first segmented from the raw data and then featurized. At the first phase, segmentation was made with edge detection using a FIR median hybrid filter [19], convolution, and thresholding. We also developed a flexible statistical model-based method for segmentation, based on Segmental Semi-Markov Models (SSMM) [20]. The SSMM method is able to match "high-quality" footstep from a noisy channel, which is essential in building an online person identification and tracking system.

For feature extraction, a single footstep was divided into two sections: the ball of the footstep and the heel of the footstep. The partition was made based on the local minimum (x_{min}, y_{min}), as shown in figure 3. Altogether 31 features were extracted, including features from both the spatial and the frequency domain.

Spatial features included mainly coordinate points of the signal, as shown in figure 3. The relations between the coordinate points were also included. In extracting the frequency domain features, footsteps were interpolated or decimated to have a uniform length of 64 time units. Then, the Fast Fourier Transformation (FFT) [21] was used to calculate the frequency domain presentation. First, four FFT coefficients were chosen as a part of the feature vector. All features were normalized between 0 and 1 to have an equal influence on a standard Euclidean distance.

At the beginning of the project, a software package LNKnet [22] developed at MIT Lincoln Laboratory was used to analyze the relevance of features. It uses a kNN classifier to recognize the best subset of features in a given data set, minimizing the classification error. It selected 13 features, all of which belonged to the spatial domain. This clearly shows that a lot of non-informative features were included in the original data set. However, in an adaptive system, the informative features may change, and it is therefore important to have a large set available, from which the best can be chosen online. Two data sets were used to analyze the DSLVQ algorithm. The first data set consisted of the 13 features chosen by the kNN classifier. The second data set contained all of the 31 features, including the non-informative ones. These 31 features are presented in table 1. The features 1 - 13 were chosen by the kNN classifier.

The LVQ codebooks consisted of 18 prototype vectors for each class. At first, the codebooks were trained by the OLVQ1 to get raw decision boundaries, after which training was continued with LVQ1. Finally, LVQ3 and DSLVQ were used for fine tuning. The weight learning rate parameter α in DSLVQ was set to 0.01, which was ten times smaller than the codebook learning rate. This was found reasonable when quite a large number of codebook vectors were used. The standard LVQ training algorithms and the codebook initializaton were done using the LVQ_PAK [23] developed at the Faculty of Information Technology at Helsinki University of Technology. The DSLVQ algorithm was implemented in Matlab technical language [24].

When using DSLVQ with 13 features, the value of equally weighted features was 0.0769. The most informative single features were y_{end} and x_{min}, and the two smallest weight values were assigned to y_{max1} and std_2, so that the last two will have the smallest influence on the weighted distance metric. The rest of the features were assigned with weight values close to the average (equally weighted value). In the data set of 31 features, the average weight value was 0.0323, and 17 features were assigned with a value above that. The most descriptive features, in addition to the 13 features selected earlier, were obtained from the frequency domain (FFT coefficients) and also mean and standard deviation values in the end part of the footstep ($mean_3$ and std_3). The non-informative features consisted of relational features (y_{rel1}, y_{rel2}). The description of single features presented above can be found in table 1. Moreover, the weight values using these two different data sets are shown summarized in table 2.

The overall recognition accuracies of eleven walkers are presented in table 3. The results show the average accuracies and the standard deviations for 5-fold cross-validation (CV) [25]. The data sets were trained by LVQ1, LVQ3, and DSLVQ. The results show that DSLVQ improves classification accuracies in both sets compared to standard LVQ algorithms. The standard deviations in the results are very high. This is indicative of instabilities between the classifiers constructed for CV. In the most cases, the instabilities are caused by the noisy measurements (footsteps) in the data set.

Number	Name	Description
1.	x_{max1}	Maximum time value of heel strike
2.	y_{max1}	Maximum amplitude value the heel strike
3.	x_{min}	Minimum time value between heel and ball of the foot strike
4.	y_{min}	Minimum amplitude value between heel and ball of the foot strike
5.	x_{max2}	Maximum time value of ball of the foot strike
6.	y_{max2}	Maximum amplitude value of ball of the foot strike
7.	x_{end}	End point before the film is reset
8.	y_{end}	Amplitude value of the end point
9.	$mean_1$	Mean value from the beginning to the minimum point (x_{min})
10.	std_1	Standard deviation from the beginning to the minimum point (x_{min})
11.	$mean_2$	Mean value from the minimum point (x_{min}) to the middle point(x_{mid})
12.	std_2	Standard deviation from the minimum point (x_{min}) to the middle point (x_{mid})
13.	$mean_{max}$	Mean value of difference between y_{max1}, y_{max2} and y_{min}
14.	$area_1$	Area from the beginning to the minimum point (x_{min})
15.	$area_2$	Area from the minimum point (x_{min}) to the middle point (x_{mid})
16.	x_{heel}	Start point of heel strike (when amplitude is above x_{min})
17.	x_{ball}	End point of ball strike (when amplitude is below x_{min})
18.	$length_{heel}$	Length of the heel impact (x_{heel}, x_{min})
19.	$length_{ball}$	Length of the ball of the foot impact (x_{min}, x_{ball})
20.	$mean_3$	Mean of the end part (x_{mid}, x_{end})
21.	std_3	Standard deviation of the end part (x_{mid}, x_{end})
22.	$shape_{heel}$	$((y_{max1} - y_{min})/(x_{min} - x_{heel}))$
23.	$shape_{ball}$	$((y_{max2} - y_{min})/(x_{ball} - x_{mid}))$
24.	x_{rel1}	Relation (x_{max1}/x_{end})
25.	x_{rel2}	Relation (x_{max2}/x_{end})
26.	y_{rel1}	Relation (y_{max1}/x_{end})
27.	y_{rel2}	Relation (y_{max2}/x_{end})
28.	fft_1	Amplitude spectrum, 1. coefficient
29.	fft_2	Amplitude spectrum, 2. coefficient
30.	fft_3	Amplitude spectrum, 3. coefficient
31.	fft_4	Amplitude spectrum, 4. coefficient

Table 1. The features derived from each footstep.

N	w_{ave}	w_{max}	w_{min}	nof $> w_{ave}$
13	0.0769	0.1407	0.0379	7
31	0.0323	0.0487	0.0056	17

Table 2. The weight values after DSLVQ training are shown. The first column presents the number of features in the data set. w_{ave} is the value of equally weighted features. w_{max} and w_{min} are the maximum and minimum values of single feature weights. The last column shows the number of features (nof) above the average value in the data set.

N	LVQ1	LVQ3	DSLVQ
13	66.8% (±5.4)	67.4% (±5.4)	70.2% (±5.7)
31	65.8% (±5.0)	66.5% (±4.7)	69.4% (±6.4)

Table 3. The recognition results of different LVQ algorithms. The recognition rate is the average value of 5-fold cross-validation. The standard deviation is presented in parentheses. N is the number of features used in a data set.

6. CONCLUSIONS

In this work, Distinction-Sensitive LVQ was applied to footstep classification. The method provides automated feature selection and scaling to improve the identification results and to increase the adaptiveness of the identification system.

The DSLVQ method was able to discard automatically non-informative features in a data set of 31 features during the learning process of the classifier and showed almost the same accuracy as that attained by using 13 informative features. In our system, where the informative features might change, it is essential to be able to choose the relevant features online. This adaptiveness is a very important property of the online adaptive identification system.

In addition, DSLVQ provides a possibility to move the pre-processing stage to the training phase, thereby decreasing the computing time needed to build a new classifier compared to off-line pre-processing methods. Clearly, this results in faster adaptation to previously unknown (to the system) persons in our smart living room.

Acknowledgments

This work was supported by TEKES and Academy of Finland.

7. REFERENCES

[1] T. Kohonen, *Self-Organizing Maps (3rd edition)*, Springer-Verlag Berlin Heidelberg New York, 2001.

[2] M. Pregenzer, G. Pfurtscheller, and D. Flotzinger, "Automated feature selection with a distinction sensitive learning vector quantizer," *Neurocomputing*, vol. 11, pp. 19–20, 1996.

[3] R.J. Orr and G.D. Abowd, "The smart floor: A mechanism for natural user identification and tracking," in *Proc. 2000 Conf. Human Factors in Computing Systems (CHI 2000)*, New York, 2000, ACM Press.

[4] M.D. Addlesee, A. Jones, F. Livesey, and F. Samaria, "ORL active floor," *IEEE Personal Communications*, vol. 4, no. 5, pp. 35–41, October 1997.

[5] S. Pirttikangas, J. Suutala, J. Riekki, and J. Röning, "Footstep identification from pressure signals using Hidden Markov Models," in *Proc. Finnish Signal Processing Symposium (FINSIG'03)*, Tampere, Finland, May 2003, pp. 124–128.

[6] S. Pirttikangas, J. Suutala, J. Riekki, and J. Röning, "Learning vector quantization in footstep identification," in *Proc. 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA 2003)*, M.H. Hamza, Ed., Benalmadena, Spain, September 8-10 2003, IASTED, pp. 413–417, ACTA Press.

[7] J. Suutala, S. Pirttikangas, J. Riekki, and J. Röning, "Reject-optional LVQ-based two-level classifier to improve reliability in footstep identification," in *Proc. 2nd International Conference on Pervasive Computing (PERVASIVE 2004)*, Linz/Vienna, Austria, April 18-23 2004, Lecture Notes in Computer Science, Vol. 3001, pp. 182–187, Springer-Verlag.

[8] J. Little and J. Boyd, "Recognizing people by their gait: the shape of motion," *MIT Press Journal Videre*, 1996.

[9] P.S. Huang, C.J. Harris, and M.S. Nixon, "Comparing different template features for recognizing people by their gait," in *Proceedings of the British Machine Vision Conference 1998 (BMVC 1998)*, J.N. Carter and M.S. Nixon, Eds., Southamton, United Kingdom, 1998, British Machine Vision Association.

[10] A. Kale, A.N. Rajagopalan, N. Cuntoor, and V. Krüger, "Gait based recognition of humans using continuous HMMs," in *Proceedings of the International Conference on Face and Gesture Recognition*, Washington DC, USA, 2002.

[11] M. Weiser and J.S. Brown, *The Coming Age of Calm Technology. In: Beyond Calculation: The Next Fifty years of Computing*, Springer-Verlag, New York, 1997.

[12] R. Headon and R. Curwen, "Recognizing movements from the ground reaction force," in *Proceeding of the 2001 Workshop on Perceptive User Interfaces*, Orlando, Florida, USA, November 15-16 2001.

[13] J-S. Yun, S-H Lee, W-T Woo, and J-H Ryu, "The user identification system using walking pattern over the ubifloor," in *Proceedings of International Conference on Control, Automation, and Systems (ICCAS2002)*, Gyeongju, Korea, October 2003.

[14] P. Cattin, *Biometric Authentication System Using Human Gait*, Ph.D. thesis, ETH-Zürich, Institute of Robotics, 2002.

[15] M. Paajanen, J. Lekkala, and K. Kirjavainen, "Electromechanical film (EMFi) - a new multipurpose electret material," *Sensors and actuators A*, vol. 84, no. 1-2, August 2000.

[16] T. Bojer, B. Hammer, D. Schunk, and K. Tluk von Toschanowitz, "Relevance determination in learning vector quantization," in *European Symposium*

on Artificial Neural Networks 2001 (ESANN'01), M.Verleysen, Ed. 2001, pp. 271–276, D-facto publications.

[17] T. Villmann B. Hammer, "Generalized relevance learning vector quantization," *Neural Networks*, vol. 15, pp. 1059–1068, 2002.

[18] A. Sato and K. Yamada, "Generalized learning vector quantization," in *Advances in Neural Information Processing systems (NIPS)*, G. Tesauro, D. Touretzky, and T. Lee, Eds. 1995, vol. 7, pp. 423–429, MIT Press.

[19] P. Heinonen and Y. Neuvo, "FIR-median hybrid filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSAP-35, no. 6, pp. 832–838, 1987.

[20] K. Koho, J. Suutala, T. Seppänen, and J. Röning, "Footstep pattern matching from the pressure signals," in *Proc. of 12th European Signal Processing Conference (EUSIPCO 2004)*, Vienna, Austria, September 6-10 2004, (accepted).

[21] E. Oran Brigham, *The Fast Fourier Transform*, Prentice-Hall, 1974.

[22] L. Kukolich and R. Lippmann, *LNKnet User's Guide*, Massachusetts Institute of Technology, Lincoln Laboratory, August 1999.

[23] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, and K. Torkkola, "LVQ_PAK: The learning vector quantization program package," Tech. Rep., Helsinki University of Technology, 1996, Report A30.

[24] "The Mathworks: MATLAB for technical computing," http://www.mathworks.com, Available 14.5.2004.

[25] R. Kohavi, "A study of cross-validation and bootsrap for accuracy estimation and model selection," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Montreal, Quebec, Canada, 1995.