

————— Technical Report No. 099 —————

# Multi-class SVMs for Image Classification using Feature Tracking

Arnulf B. A. Graf<sup>1</sup> & Christian Wallraven<sup>2</sup>

————— August 2002 —————

The authors would like to thank O. Chapelle and B. Schölkopf for useful discussions and M. Giese and F. Wichmann for helpful comments about the manuscript. The authors were supported by a grant from the EC (COGVIS).

<sup>1</sup> AG Bülthoff, E-mail: [arnulf.graf@tuebingen.mpg.de](mailto:arnulf.graf@tuebingen.mpg.de)

<sup>2</sup> AG Bülthoff, E-mail: [christian.wallraven@tuebingen.mpg.de](mailto:christian.wallraven@tuebingen.mpg.de)

---

This report is available via anonymous ftp at <ftp://ftp.kyb.tuebingen.mpg.de/pub/mpi-memos/pdf/TR-099.pdf> in PDF-format or at <ftp://ftp.kyb.tuebingen.mpg.de/pub/mpi-memos/TR-099.ps.Z> in compressed PostScript-format. The complete series of Technical Reports is documented at: <http://www.kyb.tuebingen.mpg.de/bu/techr/>



# Multi-class SVMs for Image Classification using Feature Tracking

Arnulf B. A. Graf & Christian Wallraven

**Abstract.** In this paper a novel representation for image classification is proposed which exploits the temporal information inherent in natural visual input. Image sequences are represented by a set of salient features which are found by tracking of visual features. In the context of a multi-class classification problem this representation is compared against a representation using only raw image data. The dataset consists of image sequences generated from a processed version of the MPI face database. We consider two types of multi-class SVMs and benchmark them against nearest-neighbor classifiers. By introducing a new set of SVM kernel functions we show that the feature representation significantly outperforms the view representation.

---

## 1 Introduction

Traditionally classification of images—at least in the area of computer vision—has been considered using mostly static representations. Natural visual input, however, consists of spatio-temporal patterns and psychophysical studies corroborate that the human visual system is able to exploit inherent temporal characteristics [1].

If one considers classification of image sequences a possible data representation consists of individual frames, i.e. the simplest view-based representation containing only raw pixel data. In this paper, we present a second representation (based on [2]) which takes dynamic information into account. This is done by extraction of interest points in the image (in our case, corners) and construction of visual features by using the corner positions together with their local pixel neighborhoods. A set of salient features is then found by tracking these visual features over the input sequence. This representation can be seen as incorporating the *a priori* knowledge of a sequential image presentation (temporal continuity).

In this paper Support Vector Machines (SVMs) are benchmarked against classifiers derived from nearest-neighbor approaches using various similarity measures to assess the generalization capabilities of the two representations. Multi-class SVMs in a normalized feature space are considered for classification using two strategies: one versus the rest (1-vs-r) in combination with winner-takes-all and one versus one (1-vs-1) via Directed Acyclic Graph (DAG). Novel metrics are developed for the kernel functions in order to accommodate both the classification of images and features. The work which is most closely related to this paper deals with histogram-based classification of im-

ages [3], where appropriate metrics in kernel functions are studied and SVM-learning of parts of faces (or local features) as proposed in [4]. Note, however, that in both studies static representations were used.

Sec. 2 presents theoretical considerations on multi-class SVMs in a normalized feature space, feature tracking and metrics derived from normalized cross-correlations. Sec. 3 describes the database of image sequences and presents numerical classification experiments using images and features and gives results from the comparison of SVMs and nearest-neighbor classifiers.

## 2 Theoretical considerations

### 2.1 Multi-class SVMs

SVMs as binary classifiers have drawn much attention because of their high classification performance and thorough mathematical foundations rooted in statistical learning theory [5]. They are here considered in a normalized feature space [6] by using modified kernel functions  $\frac{K(\vec{x}, \vec{y})}{\sqrt{K(\vec{x}, \vec{x})K(\vec{y}, \vec{y})}}$ , the latter still satisfying Mercer's theorem. The data in such a space lies on a unit hypersphere. A modification in the SV algorithm taking advantage of this geometrical property is proposed in [6] where the offset of the optimal separating hyperplane (OSH) is shifted. In the standard SVM algorithm, the OSH is placed in the middle of the margins. In a normalized feature space, however, the OSH is chosen so as to separate equidistantly the unit hypersphere in between the margins as shown in figure 1. In other words, the distance metric is projected on the unit hypersphere and the correction of the offset can be

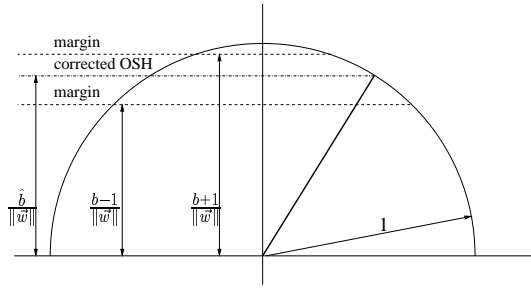


Figure 1: Correction of  $\hat{b}$  in a normalized two-dimensional feature space. Note the asymmetry of the margins around the OSH.

written as:

$$\frac{\hat{b}}{\|\vec{w}\|} = \cos\left(\frac{\arccos\left(\frac{b-1}{\|\vec{w}\|}\right) + \arccos\left(\frac{b+1}{\|\vec{w}\|}\right)}{2}\right) \quad (1)$$

We consider two strategies in order to extend the binary classification scheme to multiple classes, say  $M$ , without modifying the SV formulation (as opposed to [7]). As a first approach,  $M$  classifiers are trained using a 1-vs-r protocol and classification is done according to a *winner-takes-all* strategy over all the classes for each element of the testing subset of the database [5,8]. In the second approach,  $\frac{M(M-1)}{2}$  classifiers are trained for each pair of classes in the training database according to a 1-vs-1 protocol (also called pairwise classification, [9]). Classification is then performed using a *Directed Acyclic Graph* over the classes for each element of the testing database [10] instead of a voting scheme as proposed in [9]. In both cases, the classification error in the experiments presented in this paper is the mean of the binary errors corresponding to each element of the testing set. Data compression and redundancy is assessed by computing the SV ratio, which is the ratio of the mean number of SVs across all classifiers and the number of elements in the training set.

## 2.2 Feature tracking

Evidence from psychophysical studies shows that humans are able to exploit the temporal continuity in the natural visual input during learning of object classes [1]. A computational framework directly motivated by these results is introduced in [2] where a sparse yet powerful representation is generated from dynamic visual input. The main idea behind this approach is to process a sequence of images  $\mathcal{I} = \{I_t\}_{t=t_0}^{t_1}$  to extract salient features via feature tracking.

The proposed framework consists of two main parts: feature *extraction* and *tracking* (see figure 2). First of all, a Gaussian scale pyramid with 3 scales is constructed from  $I_t$  to enable multi-scale analysis. Feature

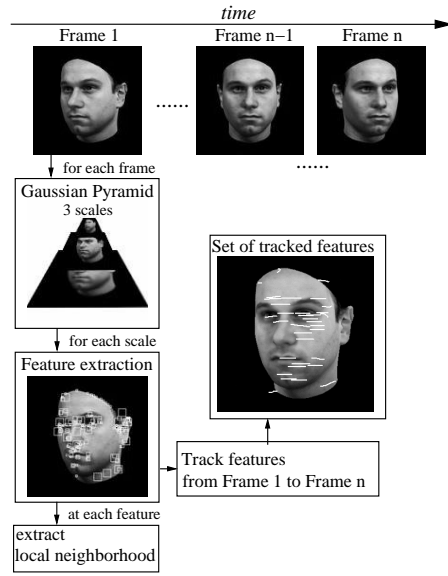


Figure 2: Overview of the feature extraction and tracking process. The large image shows the feature trajectories of the input sequence.

extraction is done for each scale by determining at each pixel the symmetric structure matrix  $\mathbf{H}$  (see also [11]) which is evaluated in a small square neighborhood  $\mathcal{N}$  (typically  $7 \times 7$  pixels) centered on this pixel:

$$\mathbf{H} = \begin{pmatrix} \sum_{\mathcal{N}} \|\frac{\partial I}{\partial x}\|^2 & \sum_{\mathcal{N}} \langle \frac{\partial I}{\partial x} | \frac{\partial I}{\partial y} \rangle \\ \sum_{\mathcal{N}} \langle \frac{\partial I}{\partial x} | \frac{\partial I}{\partial y} \rangle & \sum_{\mathcal{N}} \|\frac{\partial I}{\partial y}\|^2 \end{pmatrix}$$

The smaller of the two eigenvalues of  $\mathbf{H}$ ,  $\lambda_2 = \frac{tr(\mathbf{H})}{2} - \sqrt{(\frac{tr(\mathbf{H})}{2})^2 - det(\mathbf{H})}$ , yields information about the structure of  $\mathcal{N}$ . The values of  $\lambda_2$  which are above a pre-defined threshold are considered further and yield a set of  $n$  interest points in each image  $I_t$  of the sequence. A set of visual features  $\mathcal{F}_t = \{\vec{p}_i(I_t), \vec{f}_i(I_t)\}_{i=1}^n$  is then constructed by using the Cartesian pixel positions  $\vec{p}_i(I_t)$  of the interest points together with their pixel neighborhood  $\mathcal{N}$  written in vector form as  $\vec{f}_i(I_t)$ .

The feature tracking algorithm as proposed by [12] and further developed by [13] matches two feature sets,  $\mathcal{F}_{t_1}$  and  $\mathcal{F}_{t_2}$ , by constructing a pair-wise similarity matrix  $\mathbf{A}$  given by:

$$A(i, j) = \left( NCC(\vec{f}_i, \vec{f}_j) + 1 \right) \cdot \exp\left(-\frac{dist(\vec{p}_i, \vec{p}_j)}{2\sigma_{dist}^2}\right)$$

where  $i, j$  index interest points of  $\mathcal{F}_{t_1}, \mathcal{F}_{t_2}$  respectively,  $dist$  is a spatial similarity measure (such as the Euclidean distance in the image plane) and  $NCC$  a feature similarity measure (such as a normalized cross-correlation, see below). If we define the standard SVD

of the similarity matrix to be  $\mathbf{A} = \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{W}^T$ , the matrix defined by  $\mathbf{A}' = \mathbf{U} \cdot \mathbf{W}^T$  then yields a one-to-one feature mapping between  $\mathcal{F}_{t_1}$  and  $\mathcal{F}_{t_2}$  by taking the largest elements each in row and column of  $\mathbf{A}'$  [12]. One of the advantages of this approach is the combination of a correlation term *and* a distance term in the similarity measure which increases the robustness of the matching process considerably [2,13]. A similar approach is considered in section 2.3 for the definition of new metrics used in the kernel functions of SVMs.

For this study the feature representation consists of the set of features  $\mathcal{F}$  which could be tracked over the whole input image sequence  $\mathcal{I}$  (larger image in figure 2). The set  $\mathcal{F}$  thus represents a set of salient features which is determined in a purely data-driven way from dynamic visual input by exploiting temporal continuity through feature tracking.

### 2.3 Normalized cross-correlation functions

Presented below are various metrics to be incorporated into the kernel functions of the SVM for image or feature classification. When comparing images whose intensities are represented by the vectors  $\vec{x}$  and  $\vec{y}$ , the most basic metric is the pixel-based Euclidean norm:  $\|\vec{x} - \vec{y}\|$ . Another possibility is to use the normalized cross-correlations (NCCs) between the two images as:

$$C(\vec{x}, \vec{y}) = \frac{\langle \vec{x} - \vec{\mu}_{\vec{x}} | \vec{y} - \vec{\mu}_{\vec{y}} \rangle}{\|\vec{x} - \vec{\mu}_{\vec{x}}\| \|\vec{y} - \vec{\mu}_{\vec{y}}\|} \quad (2)$$

where  $\vec{\mu}_{\vec{x}}, \vec{\mu}_{\vec{y}}$  are the vectors of the mean value of  $\vec{x}, \vec{y}$  respectively. Clearly,  $-1 \leq C(\vec{x}, \vec{y}) \leq 1$  and  $C(\vec{x}, \vec{y}) = C(a\vec{x} + b, c\vec{y} + d)$  for  $a, b, c, d \in \mathbb{R}$ .  $C(\vec{x}, \vec{y})$  is thus invariant to an affine transformation of image intensities.

For the classification of the features as presented above, the following two approaches can be used as metrics in the kernel functions. Assume the two images are represented by position and feature vectors as defined before:  $\{\vec{p}_i(\vec{x}), \vec{f}_i(\vec{x})\}_{i=1}^n$  and  $\{\vec{p}_i(\vec{y}), \vec{f}_i(\vec{y})\}_{i=1}^m$ . The metric evaluated on the features only is defined as:

$$C_f(\vec{x}, \vec{y}) = \frac{1}{n} \sum_{i=1}^n \max_{j=1, \dots, m} C(\vec{f}_i(\vec{x}), \vec{f}_j(\vec{y})) \quad (3)$$

The more complete metric using position *and* feature information is expressed as:

$$C_{pf}(\vec{x}, \vec{y}) = \frac{1}{n} \sum_{i=1}^n \max_{j=1, \dots, m} [C(\vec{f}_i(\vec{x}), \vec{f}_j(\vec{y})) + 1] \exp(-\sigma \|\vec{p}_i(\vec{x}) - \vec{p}_j(\vec{y})\|^2) - 1 \quad (4)$$

The above NCC metrics are incorporated in the kernel functions of the SVM as shown below. These kernels

are positive definite and thus allow SVM training. It is not known, however, whether they satisfy Mercer's conditions and there is thus no guarantee that the margin in feature space is maximized [3].

## 3 Numerical experiments

### 3.1 Database and representation

The database used in the following experiments is the MPI human head database as developed and described in [14] and is composed of 100 male and 100 female three-dimensional head models. From this database sequences showing a horizontal 180° rotation of the heads are generated. Individual frames consist of 256x256 pixel 8 bit greyscale images with a black background. The images are pre-processed such that the heads have the same mean intensity, same number of pixels and same center of mass. The removal of these obvious cues is done to make classification a harder task. In the following experiments we report results from a random subset of 100 heads taken from the whole processed database. We first split the full rotation sequence into 3 subsequences defined by the following views: ( $-90^\circ \rightarrow -54^\circ, -18^\circ \rightarrow 18^\circ$  and  $54^\circ \rightarrow 90^\circ$ ) yielding 6 training vectors. The separate test set is chosen to study the interpolation capabilities of the classifiers and consists of 2 *intermediate* subsequences, namely ( $-72^\circ \rightarrow -36^\circ$  and  $0^\circ \rightarrow 36^\circ$ ) yielding 4 testing vectors.

The simplest representation of the above image sequences consists of the raw pixel data from the *views* as defined above, which are reduced to 32x32 pixel images for the sake of numerical tractability. Training and testing are then performed on the 1024-dimensional vector obtained from the corresponding image matrix. Feature representations are extracted by feature tracking according to section 2.2 from the *subsequences* of the training set and testing set, resulting in 6 feature sets  $\mathcal{F}$  used for training and 4 sets for testing. The size of the local pixel neighborhood  $\mathcal{N}$  of each interest point is chosen to be 7x7 and the number of tracked features is set to 21 (see section 3.3).

### 3.2 Classification of images

This section presents results from classification of images using the raw pixel representation. We first consider two types of benchmark classifiers, namely the *K-nearest-neighbors* (KNN) and *K-highest-correlations* (KHC) classifiers. A KNN classifier assigns an unknown pattern  $\vec{z}$  to the class  $1 \leq \mathcal{C}(\vec{z}) \leq M$  according to:  $\mathcal{C}(\vec{z}) = \arg \min_{i=1, \dots, N} \sum_{k=1}^K \|\vec{z} - \vec{\eta}_k^i\|$  where  $\vec{\eta}_k^i \in \mathcal{N}_E(\vec{z})$  is in the Euclidean neighborhood of  $\vec{z}$ . Consequently, we define KHC classification according to:  $\mathcal{C}(\vec{z}) = \arg \max_{i=1, \dots, N} \sum_{k=1}^K \text{Corr}(\vec{z}, \vec{\eta}_k^i)$  where

$\vec{\eta}_k^i \in \mathcal{N}_C(\vec{z})$  belongs to the vectors with highest correlations to  $\vec{z}$ . Setting  $Corr(\vec{x}, \vec{y}) = C(\vec{x}, \vec{y})$  yields classification error curves as shown in figure 3. The

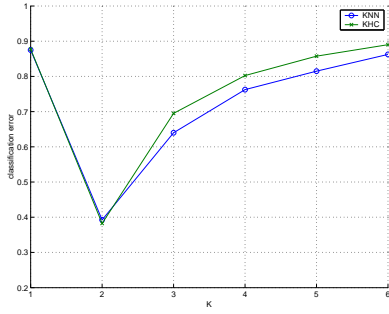


Figure 3: Benchmark KNN and KHC classification: the minimum classification error for KNN is **39.25%** and for KHC is **38.25%**, both for **K=2**.

behavior of the KNN and KHC classifiers are similar with a slight performance advantage for KNN on average. The large decrease of classification error for  $K = 2$  is due to the selection of the training and testing sets. The views in the testing set are *in between* the training views such that  $K = 2$  yields the best generalization.

The numerical results reported in table 1 are obtained using multi-class SVMs with or without the NCC metric for heuristically-determined optimal parameters of the kernel functions. The SVM trade-off parameter  $C$  is set to a relatively high value of 100 which favors separability over correct classification. Further numerical experiments, however, show that the classification performance is to a large extent insensitive to this value. When considering RBF kernels, the correlation function slightly increases classification error with respect to the Euclidean norm. In the case of the polynomial kernel, the classification performance is unaffected by the choice of the correlation function or the norm. This behavior is valid for both multi-class strategies. Since computational costs for NCC are larger than for the Euclidean norm, it seems sufficient to use the simpler Euclidean norm in this case.

The optimal parameters found for both multi-class strategies are different which is due to the intrinsic structure of the two protocols. The 1-vs-1 strategy outperforms here the 1-vs-r strategy and we also find higher invariance of the classification error with respect to the kernel function. As far as the number of SVs is concerned, all elements of the training set are SVs for a 1-vs-1 strategy, whereas one sixth of the training set are SVs in the case of a 1-vs-r strategy. The 1-vs-1 strategy thus needs more support vectors in total for a slight performance gain.

### 3.3 Classification of features

In this section, we present classification results for the feature representation. As mentioned before, the nature of the interpolation problem considered here implies a minimal classification error for the KHC classifier at  $K = 2$ . The benchmark experiments using KHC classifiers are thus done for this case and yield classification errors of 55.00% for  $Corr(\vec{x}, \vec{y}) = C_f(\vec{x}, \vec{y})$  and of 23.50% for  $Corr(\vec{x}, \vec{y}) = C_{pf}(\vec{x}, \vec{y})$  with  $\sigma = 3 \cdot 10^{-5}$ . Considering features alone thus degrades the classification performance compared to a KNN or KHC classifier applied directly to the raw image data (even though classification performance is above chance level). However, the combination of position and feature information clearly outperforms both benchmark classifiers on the images (23.50% compared to 38.25%, cf. figure 3). The information contained in the tracked feature set is thus better suited for classification than the one contained in the images without explicit tracking. In the following, we only consider the combination of spatial and feature information.

The classification results reported in table 2 are obtained by integrating the  $C_{pf}(\vec{x}, \vec{y})$  metric in the kernel function. In this case, experimental pre-runs indicate the importance to compute the sum in equation 4 over a subset of values given by the max-function, which are higher than a pre-defined threshold  $\theta$  (in the following experiments, we use the optimal value of  $\theta = 0.2$ ) in order to avoid noise in the correlation function. In addition, only the  $Q = 21$  highest values of the max-function in equation 4 are considered, i.e., all features are fed into the kernel and only 21 are evaluated. In this way, the number of effective components used by the SVM when considering features ( $21 \cdot 49$ ) is of the same order of magnitude as for images ( $32 \cdot 32$ )<sup>1</sup>.

The main result is that the use of the feature representation yields a significant performance increase for both multi-class protocols (a decrease in classification error from 19.50% to 5.50% for 1-vs-r and 16.25% to 8.00% for 1-vs-1). This confirms that a representation which exploits only a small amount of prior knowledge about the data (temporal continuity) can result in a large performance gain without increase of the dimensionality of the input vectors.

The feature representation increases the number of SVs for the 1-vs-r strategy when compared to the image representation (16.00% to 37.67%). This indicates on the one hand a more difficult classification task but on the other hand also shows that the feature representation is less redundant. As the number of SVs for the 1-vs-1 strategy is already at ceiling for the image rep-

<sup>1</sup>Using all values instead of the 21 highest did not improve the classification performance.

Table 1: Classification error and number of SVs for SVM image classification.

SVM 1-vs-r classifier and winner-takes-all		
$K(\vec{x}, \vec{y})$	class. error in %	SV ratio in %
$\exp(-\gamma\ \vec{x} - \vec{y}\ ^2)$ with $\gamma = 26 \cdot 10^{-8}$	18.75	15.67
$\exp(-\gamma(1 - C(\vec{x}, \vec{y})))$ with $\gamma = 1.9$	19.50	16.00
$(1 + \langle \vec{x}   \vec{y} \rangle)^d$ with $d = 4$	19.25	15.83
$(1 + C(\vec{x}, \vec{y}))^d$ with $d = 4$	19.25	16.00
SVM 1-vs-1 classifier and DAG		
$K(\vec{x}, \vec{y})$	class. error in %	SV ratio in %
$\exp(-\gamma\ \vec{x} - \vec{y}\ ^2)$ with $\gamma = 7 \cdot 10^{-8}$	16.00	100
$\exp(-\gamma(1 - C(\vec{x}, \vec{y})))$ with $\gamma = 1.1$	16.25	100
$(1 + \langle \vec{x}   \vec{y} \rangle)^d$ with $d = 3$	16.00	100
$(1 + C(\vec{x}, \vec{y}))^d$ with $d = 3$	16.00	100

Table 2: Classification error and number of SVs for SVM feature classification.

SVM 1-vs-r classifier and winner-takes-all		
$K(\vec{x}, \vec{y})$	class. error in %	SV ratio in %
$\exp(-\gamma(1 - C_{pf}(\vec{x}, \vec{y})))$ with $\gamma = 12, \sigma = 5 \cdot 10^{-5}$	5.50	37.67
SVM 1-vs-1 classifier and DAG		
$K(\vec{x}, \vec{y})$	class. error in %	SV ratio in %
$\exp(-\gamma(1 - C_{pf}(\vec{x}, \vec{y})))$ with $\gamma = 8, \sigma = 6 \cdot 10^{-5}$	8.00	100

representation a further increase in number of SVs is not possible. This might be the reason for the lower classification performance of the 1-vs-1 strategy compared to 1-vs-r (8.00% compared to 5.50%).

In contrast to the previous results where polynomial kernels performed as well as RBF kernels, the feature representation yields rather inconsistent performance for these kernels and the results are thus not reported here. Further analysis shows that they result in overfitting of the data which indicates poor generalization ability.

## 4 Conclusions

In this paper we have proposed a representation for image sequences which consists of a set of tracked visual features. This representation significantly decreased the classification error when compared with a simpler representation containing only raw pixel data for both SVM and benchmark classifiers. For SVM classifiers, a new metric was developed which performs feature matching directly in the kernel function using both position and pixel information of the tracked visual features. The proposed framework seems to be an effective and efficient way to exploit the dynamic information in visual input.

We also investigated the effectiveness of the DAG strategy for multiple class problems which resulted in sometimes significant gains in classification performance. This behaviour might be due to the fact that the

number of classes considered here is large ( $M = 100$ ). However, one cannot draw the general conclusion that the 1-vs-1 strategy always performs best, which motivates further research into multi-class extensions of binary classification algorithms.

Future research will investigate the properties of the SVs (number, relative encoding and sparseness) which should provide further insights into the compactness of the different representations. Another line of research concerns the generalization capabilities with regard to noise in the learning and testing data (e.g., occlusions and image noise). In addition, we are currently exploring different kinds of visual features which are motivated by biological findings and which could further improve the performance of the proposed framework.

## References

- [1] G. Wallis and H. Bülthoff. *Effects of temporal association on recognition memory*. In Proc. of the National Academy of Sciences of the United States of America, 98, 2001.
- [2] C. Wallraven and H. Bülthoff. *Automatic acquisition of exemplar-based representations for recognition from image sequences*. In Proc. CVPR'01 - Workshop Models vs. Exemplars, 2001.
- [3] O.Chapelle, P. Haffner and V. Vapnik. *SVMs for Histogram-Based Image Classification*. IEEE Transaction on Neural Networks, 9, 1999.
- [4] B. Heisele, T. Serre, M. Pontil, T. Vetter and T. Poggio. *Categorization by Learning and Combining Object Parts*. Advances in Neural Information Processing Systems 13, S.A. Solla, T.K. Leen and K.-R. Müller (eds). The MIT

Press, 2001.

[5] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[6] A. Graf and S. Borer. *Normalization in Support Vector Machines*. Proceedings of the DAGM - Pattern Recognition. Springer, 2001.

[7] J. Weston and C. Watkins. *Support Vector Machines for Multi-Class Pattern Recognition*. In the proceedings of ESANN'99.

[8] B. Schölkopf and A.J. Smola. *Learning with Kernels*. The MIT Press, 2002.

[9] U. H.-G. Kressel. *Pairwise Classification and Support Vector Machines*. In "Advances in Kernel Methods - Support Vector Learning", B. Schölkopf, C.J.C. Burges and A.J. Smola (eds). The MIT Press, 1999.

[10] J.C. Platt, N. Cristianini and J. Shawe-Taylor. *Large Margin DAGs for Multiclass Classification*. Advances in Neural Information Processing Systems 12, S.A. Solla, T.K. Leen and K.-R. Müller (eds). The MIT Press, 2000.

[11] C. Tomasi and T. Kanade. *Detection and tracking of point features*. Carnegie-Mellon Tech Report CMU-CS-91-132, 1991.

[12] G. Scott and H. Longuet-Higgins. *An algorithm for associating the features of two images*. In Proc. Royal Society of London, B(244), 1991.

[13] M. Pilu. *A direct method for stereo correspondence based on singular value decomposition*. In Proc. CVPR'97, 1997.

[14] V. Blanz and T. Vetter. *A Morphable Model for the Synthesis of 3D Faces*. In Proc. Siggraph99, pp. 187-194. Los.Angeles: ACM Press, 1999.