

# Fast Learning for Customizable Head Pose Recognition in Robotic Wheelchair Control

Christian Bauckhage<sup>1,2</sup>, Thomas Käster<sup>1</sup>, Andrei M. Rotenstein<sup>1</sup> and John K. Tsotsos<sup>1</sup>

<sup>1</sup>Centre for Vision Research  
York University

Toronto, ON, M3J1P3, Canada

<sup>2</sup>Deutsche Telekom AG  
Laboratories

10587 Berlin, Germany

## Abstract

*In the PLAYBOT project, we aim at assisting disabled children at play. To this end, we are developing a semi autonomous robotic wheelchair. It is equipped with several visual sensors and a robotic manipulator and thus conveniently enhances the innate capabilities of a disabled child. In addition to a touch screen, the child may control the wheelchair using simple head movements. As control based on head posture requires reliable face detection and head pose recognition, we are in need of a robust technique that may effortlessly be tailored to individual users. In this paper, we present a multilinear classification algorithm for fast and reliable face detection. It trains within seconds and thus can easily be customized to the home environment of a disabled child. Subsequent head pose recognition is done using support vector machines. Experimental results show that this two stage approach to head pose-based robotic wheelchair control performs fast and very robust.*

## 1 Introduction and Motivation

The work reported in this paper results from a project on assistive technologies for children living with physical impairments. Our overall goal is a semi autonomous wheelchair that assists these children at play. The current prototype is equipped with a robot arm and several cameras so that it can safely navigate and manipulate objects within the play environment. One of the cameras registers head movements of the user. This provides a rather effortless way for the child to control the orientation of the wheelchair: turning the head left or right causes the chair to turn in that direction after a little while. The *face cam* is mounted on the lower left of the person sitting in the chair (see Fig. 1) to be unobtrusive.

In this paper, we focus on a two-staged process for coarse but highly reliable head pose estimation in our scenario. The first step is to detect the user's face in the images

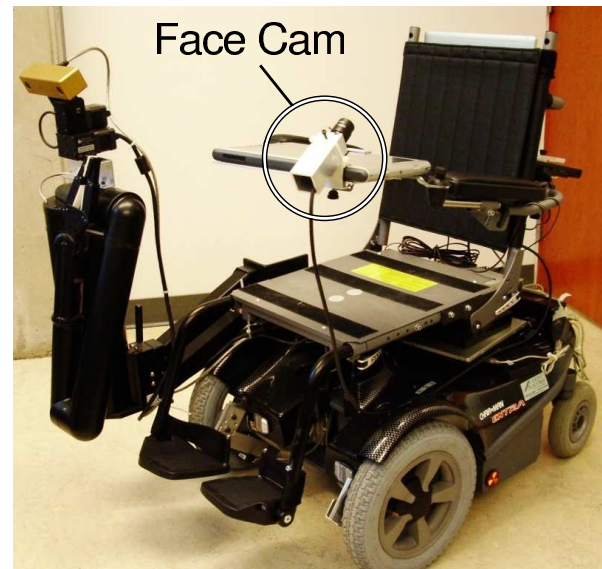


Figure 1. The PLAYBOT robotic wheelchair.

recorded by the face cam. In the second step, head posture is determined from those image patches showing the face.

Since face detection and head pose estimation are very active areas in computer vision (cf. e.g. [6, 8, 11, 18, 20]), it is no surprise to find reports on methods that provide the speed and reliability required in robotic control. Especially, boosting techniques (e.g. [10, 15]) are known to be fast and robust. But, their performance comes at the cost of an extensive and exhaustive training phase. Given our scenario, however, there are three major reasons why we need algorithms that can quickly be adjusted to different environments and users. (1) The view angle of the face cam is unorthodox in that it does not provide frontal views of faces. Moreover, depending on the individual needs of the child, the way the camera is mounted may need to be reconfigured. (2) Pre-trained current state of the art face detectors were trained on data sets that are heavily biased towards

able-bodied persons. Presently, there is no image repository dedicated to the disabled. For both these reasons, training face detection and head pose recognition algorithms in the user's home is desirable. However, (3) state of the art face detectors rely on the analysis of huge training sets. Providing correspondingly exhaustive training data is of course infeasible for a disabled child or his caregivers.

In the next section, we introduce multilinear classifiers. We shall see that this technique achieves very fast learning for face detection in color images. The results in section 3 underline that it also provides fast runtime and reliable results. Afterwards, we will briefly discuss support vector machines for coarse but reliable head pose estimation for wheelchair control. A conclusion closes this contribution.

## 2 Multilinear Discriminant Classifiers

Many pattern recognition techniques that are being applied in computer vision require vectorial representations of image data. Recent findings, however, suggest, that treating images for what they really are, namely *multiindexed objects* or *higher order tensors*, improves performance in tasks such as image coding or classification [1, 12, 14, 16, 17, 19]. Extending an approach by Bauckhage and Tsotsos [1] to third-order tensors, we will next discuss *multilinear discriminant analysis* of color image data.

### 2.1 Classical Discriminant Analysis

Given a set of pairs  $\{(\mathbf{x}_l, y_l) | l = 1, \dots, L\}$  where the  $\mathbf{x}_l \in \mathbb{R}^n$  are feature vectors sampled from two classes and the  $y_l \in \mathbb{R}$  (usually either +1 or -1) denote class membership, linear discriminant analysis (LDA) seeks a projection  $\mathbf{w} \cdot \mathbf{x}_l$  of the samples that maximizes the inter-class distance of the resulting scalars. The arguably most popular way of finding the projection vector  $\mathbf{w}$  is to maximize the Rayleigh quotient of the between- and within-class scatter of the data [3]. Alternatively, one can minimize a least squares error criterion [2, 3, 7]. If the samples  $\mathbf{x}_l$  are normalized to zero mean, the optimal  $\mathbf{w}$  is given by

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}} E(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{l=0}^L (y_l - \mathbf{w} \cdot \mathbf{x}_l)^2. \quad (1)$$

Often, the error is written as  $E = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$  where the sample vectors form the rows of the  $L \times n$  matrix  $\mathbf{X}$  and the class labels are gathered in  $\mathbf{y} \in \mathbb{R}^L$ . This results in a simple closed form expression for the optimal projection direction:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (2)$$

Below, we will refer to the matrix  $(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$  as the *Moore-Penrose inverse*.

### 2.2 Higher Order Discriminant Analysis

The least squares approach to LDA can be generalized to discriminant analysis of multilinear objects of arbitrary order. However, to simplify matters, we restrict our discussion to third-order tensors  $\mathcal{A} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$  for these lend themselves to the modeling of color images: an RGB image consists of  $m_3 = 3$  layers of a resolution of  $m_1 \times m_2$  each.

The *inner product* of two third-order tensors  $\mathcal{A}$  and  $\mathcal{B}$  is defined as

$$\mathcal{A} \cdot \mathcal{B} = \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} \sum_{i_3=1}^{m_3} \mathcal{A}_{i_1 i_2 i_3} \mathcal{B}_{i_1 i_2 i_3}. \quad (3)$$

Therefore, if  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{m_1 m_2 m_3}$  are vector unfoldings of  $\mathcal{A}$  and  $\mathcal{B}$ , we see that  $\mathcal{A} \cdot \mathcal{B} = \mathbf{a} \cdot \mathbf{b}$ . Dealing with discriminant analysis of tensorial data, the minimization problem in (1) can thus be cast as

$$\mathcal{W} = \underset{\mathcal{W}}{\operatorname{argmin}} \sum_{l=0}^L (y_l - \mathcal{W} \cdot \mathcal{X}_l)^2. \quad (4)$$

This tensorial formulation leads to a highly efficient learning process, if we require  $\mathcal{W}$  to be expressible as a sum of  $R$  rank-1 tensors

$$\mathcal{W} = \sum_{r=1}^R \mathbf{u}^r \otimes \mathbf{v}^r \otimes \mathbf{w}^r. \quad (5)$$

Here,  $\otimes$  denotes the outer product of two vectors. In the following, we will drop summation signs and apply Penrose's *abstract index notation*. In this formalism indices of tensor components assume the role of abstract markers in terms of which the algebra is formulated<sup>1</sup>.

With the rank-1 constraint on  $\mathcal{W}$ , there is no closed form solution for (4). However, by means of an alternating least squares scheme, the problem can be reduced to solving several ordinary least squares equations similar to (1). For the simple case of an  $R = 1$  term solution for  $\mathcal{W}$ , the algorithm consists of  $t = 1, \dots, t_{\max}$  iterations of the following steps. Given an initial, random guess for the vectors  $\mathbf{u} \in \mathbb{R}^{m_1}$  and  $\mathbf{v} \in \mathbb{R}^{m_2}$ , compute the *tensor contractions*

$$x_{i_3}^l = \mathcal{X}_{i_1 i_2 i_3}^l u_{i_1}(t) v_{i_2}(t), \quad l = 1, \dots, L. \quad (6)$$

Stacking the resulting vectors  $\mathbf{x}_l \in \mathbb{R}^{m_3}$  into a sample matrix  $\mathbf{X}$  yields  $\mathbf{w}(t) = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ . Given  $\mathbf{w}$ , the training set can be contracted over  $\mathbf{u}$  and  $\mathbf{w}$  in order to compute a new estimate for  $\mathbf{v}$ ; finally, given  $\mathbf{v}$ , a new estimate for  $\mathbf{u}$  can be computed.

<sup>1</sup>Consider the matrix-vector multiplication  $\mathbf{M}\mathbf{u} = \mathbf{v}$ . As the components of  $\mathbf{v}$  are given by  $v_i = M_{ij}u_j$ , the following expressions are equivalent using abstract index notation:  $\mathbf{M}\mathbf{u} = \mathbf{v} \Leftrightarrow v_i = M_{ij}u_j$ .

---

**Input:** a training set  $\{\mathcal{X}_l, y_l\}_{l=1, \dots, L}$  of image patches  
 $\mathcal{X}_l \in \mathbb{R}^{m_1 \times m_2 \times m_3}$  with class labels  $y_l$   
**Output:** a rank- $R$  approximation of a third-order  
projection tensor  $\mathcal{W} = \mathbf{u}^r \otimes \mathbf{v}^r \otimes \mathbf{w}^r$

---

```

for  $r = 1, \dots, R$ 
   $t = 0$ 
  randomly initialize  $\mathbf{u}^r(t)$ 
  orthogonalize  $\mathbf{u}^r(t)$  w.r.t.  $\{\mathbf{u}^1, \dots, \mathbf{u}^{r-1}\}$ 
  randomly initialize  $\mathbf{v}^r(t)$ 
  orthogonalize  $\mathbf{v}^r(t)$  w.r.t.  $\{\mathbf{v}^1, \dots, \mathbf{v}^{r-1}\}$ 
  repeat
     $t \leftarrow t + 1$ 
    contract  $x_{i_3}^l = \mathcal{X}_{i_1 i_2 i_3}^l u_{i_1}^r(t) v_{i_2}^r(t)$ 
    compute  $\mathbf{w}^r(t) = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ 
    orthogonalize  $\mathbf{w}^r(t)$  w.r.t.  $\{\mathbf{w}^1, \dots, \mathbf{w}^{r-1}\}$ 
    similarly compute  $\mathbf{v}^r(t)$ 
    similarly compute  $\mathbf{u}^r(t)$ 
  until  $\|\mathbf{u}^r(t) - \mathbf{u}^r(t-1)\| \leq \epsilon \vee t > t_{\max}$ 
endfor

```

---

**Figure 2. Alternating least squares scheme to compute a third-order tensor classifier  $\mathcal{W}$  given as a sum of completely orthogonal basis tensors  $\mathbf{u}^r \otimes \mathbf{v}^r \otimes \mathbf{w}^r$ .**

As the algorithm starts with arbitrary vectors  $\mathbf{u}$  and  $\mathbf{v}$ , it has to be iterated until convergence. To this end, we consider the refinement of the vector  $\mathbf{u}$ . If, in iteration  $t$ ,  $\|\mathbf{u}(t) - \mathbf{u}(t-1)\| \leq \epsilon$ , the process is stopped. In our experiments, this usually led to convergence in less than 10 iterations.

Extending the procedure to multi-term tensors is straightforward. If  $\mathcal{W} = \sum_{r=1}^k \mathbf{u}^r \otimes \mathbf{v}^r \otimes \mathbf{w}^r$  is a  $k$  term solution for the projection tensor, a  $k+1$  term representation can be found by minimizing  $E(\mathbf{u}^{k+1}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1})$ . Note that it is appropriate to require that every newly found rank-1 tensor  $\mathbf{u}^{k+1} \otimes \mathbf{v}^{k+1} \otimes \mathbf{w}^{k+1}$  be orthogonal to its predecessors. In this way, the resulting projection tensor  $\mathcal{W}$  favors directions of maximum variance in the data tensor space. Therefore, the (modified) Gram-Schmidt procedure is applied to each of the vectors  $\mathbf{u}^r$ ,  $\mathbf{v}^r$  and  $\mathbf{w}^r$  derived in the procedure. Figure 2 summarizes the overall algorithm to find an  $R$ -term sum of third-order tensors of rank 1 for multilinear discriminant analysis.

## 2.3 Benefits of the Tensorial Approach

The alternating least squares approach to higher order tensor discriminant classification provides valuable advantages for face detection.

First, tensor classifiers provide fast runtime. Classifying the content of a color image  $\mathcal{I}$  using a projection tensor  $\mathcal{W}$  amounts to a convolution  $\mathcal{I} * \mathcal{W}$ . If  $\mathcal{W}$  is a sum of rank-1 tensors, this reduces to a sequence of one-dimensional convolutions  $\sum_r ((\mathcal{I} * \mathbf{u}^r) * \mathbf{v}^r) * \mathbf{w}^r$ . This requires  $O(R(m_1 + m_2 + m_3))$  operations per pixel and therefore enables fast object detection. For larger mask sizes  $m_1 \times m_2 \times m_3$ , further speedup is gained, if the classifier is applied in the frequency domain:  $\mathcal{F}(\mathcal{I} * \mathcal{W}) = \mathcal{F}(\mathcal{I})\mathcal{F}(\mathcal{W})$ . Using a very fast Fourier transformation [4], we achieved processing times of almost 20Hz in our experiments.

Second, Multilinear discriminant classifiers train quickly. If multivariate data of size  $m_1 \times m_2 \times m_3$  were unfolded into vectors, conventional LDA based on Fisher's criterion or on least squares would have to invert matrices of sizes  $m_1 m_2 m_3 \times m_1 m_2 m_3$ . Even for moderate values of  $m_1$  and  $m_2$  and not too many training examples, this may become infeasible. However, the Moore-Penrose inverses that appear in our algorithm only require inversion of matrices of considerably reduced sizes  $m_3 \times m_3$ ,  $m_2 \times m_2$  and  $m_1 \times m_1$ . Therefore, our technique significantly shortens training time. In practice, we found that, compared to LDA on very high dimensional vector spaces, the tensor based method reduces training times from hours to seconds.

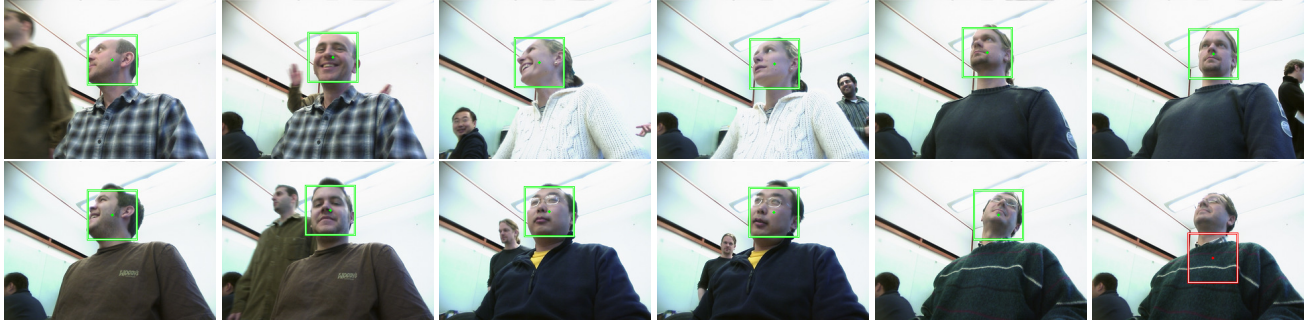
Third, tensor-based discriminant classification tackles the *small sample size problem*. This term refers to the effect that for conventional LDA the within-class scatter matrix may be singular because the number of training samples is much smaller than the dimension of the embedding space [5]. Again, as the Moore-Penrose matrices necessary for computing  $R$ -term rank-1 tensor classifiers are of small sizes, small sample sizes will not hamper multilinear discriminant analysis.

In the next section, we will present empirical results which corroborate these considerations.

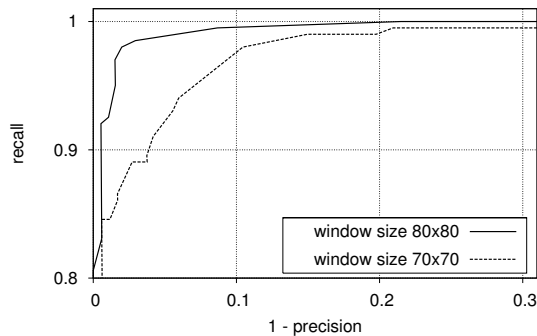
## 3 Face Detection

In order to test the general applicability of tensor classifiers to our scenario, we collected a set of images of 10 individuals sitting in our robotic wheelchair. Recorded by the face cam mounted on the robot, the images have a resolution of  $320 \times 240$  pixels and show faces of different gender, race, facial expression and head pose.

For 302 images, ground truth was provided manually; 101 images were used for training, 201 images formed our test set. From each training image, 4 image patches were



**Figure 3. Face detection examples with an  $R = 3$  term tensor discriminant classifier; on a 3Ghz Xeon PC, training with 1616 color image patches took 7 seconds; using frequency domain computations, 201 test images of size  $320 \times 240$  were processed in 11 seconds.**



**Figure 4. Precision vs. recall for face detection. The classifier of dimensions  $80 \times 80 \times 3$ , achieved an EER of 98%.**

randomly cut around the indicated center of the face; 12 patches randomly cut from the background served as counterexamples. Given the resulting 1616 examples, we trained different 3-term third-order tensor classifiers. In all experiments, the data was normalized to zero mean  $\mathcal{X} - \mathcal{M}$ , where  $\mathcal{M}$  denotes the mean of the training samples. During runtime, this accounts only for a single operation per pixel, since  $(\mathcal{X} - \mathcal{M}) \cdot \mathcal{W} = \mathcal{X} \cdot \mathcal{W} - \mathcal{M} \cdot \mathcal{W}$ , where the scalar constant  $\mathcal{M} \cdot \mathcal{W}$  can be computed beforehand.

Figure 4 shows precision recall curves we obtained for classifiers of dimensions  $70 \times 70 \times 3$  and  $80 \times 80 \times 3$ . While training the former took 9 seconds, training the latter required 7 seconds. In spite of the small sample size and short training time both classifiers achieved very high detection accuracy; the  $80 \times 80 \times 3$  variant, in particular, yielded an equal error rate of 98%. Figure 3 shows some of the results obtained using this classifier.

#### 4 Head Posture for Wheelchair Control

As pointed out in the introduction, wheelchair control by simple head movements not only requires rapid and robust face detection but also reliable head pose recognition. In order to achieve the latter, we opted for support vector machines (SVMs) because many empirical studies have shown that SVMs possess the generalization properties to solve real life problems [13]. Since online training is essential for our application, we apply the SVM<sup>light</sup> algorithm [9], which provides fast training and classification, especially for the linear kernel.

In order to test the feasibility of the SVM, we recorded several colored videos in different environments. All of these videos show the same individual sitting in the robotic wheelchair and moving the head in different directions. Again, all image sequences were labeled manually. This time, labeling was done with regard to the following view directions: left, center, right. Consequently, ground truth for testing was available. Our training set consists of 260 image patches of size  $80 \times 80$  pixels which were chosen randomly from two videos showing different environments. The dimension of the data was reduced by transforming color images to grey value images. The resulting training vectors were 1600-dimensional. On a 3Ghz Xeon PC, training a linear kernel SVM then took 80 seconds on average. This satisfies our requirement for rapid learning. The test set consisted of 1116 patches from two different videos. Head pose classification of this test data took an average of 0.75 seconds (without I/O) and yielded a success rate of 95.7%. Therefore, combining this classification scheme with the face detection algorithm from section 3 does not threaten our realtime requirements but nevertheless achieves very reliable results.

Applying our two stage pose estimation approach to wheelchair control requires a mapping of recognized head poses to motor commands of the robotic chair. One problem





**Figure 5. Face detection and head pose classification examples in two different environments. The first column shows the input frames. The second and the third column depict the corresponding detection and classification results, respectively. Head pose classification results are translated into wheelchair rotations by analyzing the temporal context of each classification. The histograms in the rightmost column display the classification results (w.r.t. the point of view of the user) of the last 75 frames. This corresponds to a temporal buffer of three seconds. The wheelchair is rotated only if a label exceeds a specific threshold during this time and if the user does not face forward.**

associated with this mapping is that occasional misclassifications of head orientation could result in sudden changes of motion. This is solved by analyzing the temporal context of each classification. Instead of issuing motor commands based on the currently produced classification the percentage of orientation labels generated during the last 75 frames is considered. This corresponds to a temporal buffer of three seconds (see rightmost column of Fig. 5). Only if an orientation label exceeds a specific threshold during this time, a turning command should be initiated. If the user is found to face forward, no motor command should be issued at all. A second problem, resulting from this strategy, is as follows. If the user intends to halt turning by facing forward, but the classifier fails to immediately recognize this, then the wheelchair could enter an unsafe configuration. As this can be frustrating for the user, it needs to be filtered out. Our solution is to treat each classification percentage and its rate of change as an input to a fuzzy logic controller. Its rule base implements the following expert notions: if either the

percentage of left or right classification is high, it outputs a strong signal to turn left or right, respectively. However, if either the percentage of center classification is high or its rate of change is large and positive, then it outputs a zero turning signal. In this way the controller errs on the side of caution.

## 5 Conclusion

In this paper, we considered the problem of fast learning for customizable face detection and subsequent head pose recognition.

Dealing with an application in robotic wheelchair control, we presented an algorithm for multilinear discriminant analysis of color images. Using an alternating least squares scheme, rank-1 decomposable projection tensors can rapidly be learned from a small set of examples. In experiments with third order tensors of considerably large spatial extension, we measured training times between 6

and 15 seconds. These experiments also revealed that tensor classifiers achieve very fast runtime; applied in the frequency domain, our multilinear classifiers process about 19 images per second. With respect to reliability, they detect faces very accurately if tested on color images of different subjects with different facial expressions and head poses. In experiments with videos of a single person, tensor-based face detection achieved 100% accuracy.

In the second stage of the proposed head pose estimation scheme, image patches returned by the multilinear face detector are fed into a support vector classifier. Using linear kernels in order to classify face images into three classes of head orientation, we were able to achieve 95.7% accuracy where the training only took 80 seconds. In order to translate the per-frame recognition results into motor commands for the wheelchair, temporal context is taken into account and a fuzzy control scheme is applied. Motor commands are issued only once the user has been recognized to be looking to either side over a short period of time.

Since both algorithms perform rapidly and reliably, our two-stage approach to wheelchair control using head movements meets the requirements encountered in robotic control. Moreover, due to their short training times, both algorithms can be adopted quickly to the conditions in the home environment of a disabled user. With help from experts in human computer interaction, we are currently developing an easy-to-use touch screen interface that allows for an effortless acquisition of training data. Our hope is to provide a simple, on-site calibration and customization procedure for the PLAYBOT that can be carried out by caregivers or even by the physically impaired child.

## 6 Acknowledgements

During his stay as a visiting scientist at the Centre for Vision Research in Toronto, Thomas Käster was generously supported by a scholarship from the Applied Computer Science Group at Bielefeld University, Germany. We want to thank Gerhard Sagerer, who is heading the group, for supporting this work by providing these grants.

## References

- [1] C. Bauckhage and J. Tsotsos. Separable Linear Discriminant Classification. In W. Kropatsch, R. Sablatnig, and A. Hanbury, editors, *Pattern Recognition*, volume 3663 of *LNCS*, pages 318–325. Springer, 2005.
- [2] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.
- [3] R. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Ann. Eugenics*, 7:179–188, 1936.
- [4] M. Frigo and S. Johnson. The design and implementation of FFTW3. *Proc. of the IEEE*, 93(2):216–231, 2005.
- [5] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
- [6] R. Gross, S. Baker, I. Matthews, and T. Kanade. Face Recognition Across Pose and Illumination. In A. Jain and S. Li, editors, *Handbook of Face Recognition*. Springer, 2004.
- [7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- [8] R.-L. Hsu, M. Abdel-Mottaleb, and A. Jain. Face detection in color images. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(5):696–706, 2002.
- [9] T. Joachims. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press, 1999.
- [10] S. Li, Z. Zhang, H.-Y. Shum, and H. Zhang. Floatboost learning for classification. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 993–1000. MIT Press, 2003.
- [11] J. Ng and S. Gong. Composite Support Vector Machines for Detection of Faces Across Views and Pose Estimation. *Image Vision Comput.*, 20(5–6):359–368, 2002.
- [12] A. Shashua and A. Levin. Linear Image Coding for Regression and Classification using the Tensor-rank Principle. In *Proc. CVPR*, volume I, pages 42–40, 2001.
- [13] V. Vapnik. Universal Learning Technology: Support Vector Machines. *NEC J. of Adv. Tech.*, 2(2):137–144, 2005.
- [14] M. Vasilescu and D. Terzopolos. Multilinear Analysis of Image Ensembles: Tensorfaces. In *Proc. ECCV*, volume 2350 of *LNCS*, pages 447–460. Springer, 2002.
- [15] P. Viola and M. J. Jones. Robust Real-Time Face Detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004.
- [16] H. Wang and N. Ahuja. Compact representation of multidimensional data using tensor rank-one decomposition. In *Proc. ICPR*, volume I, pages 44–47, 2004.
- [17] S. Yan, D. Xu, L. Zhang, X. Tang, and H.-J. Zhang. Discriminant Analysis with Tensor Representation. In *Proc. CVPR*, volume I, pages 526–532, 2005.
- [18] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting Faces in Images: A Survey. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(1):34–58, 2002.
- [19] J. Ye, R. Janardan, and Q. Li. Two-Dimensional Linear Discriminant Analysis. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1569–1576. MIT Press, 2005.
- [20] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face Recognition: A Literature Survey. *ACM Comput. Surv.*, 35(4):399–458, 2003.