TENSOR-BASED FILTER DESIGN USING KERNEL RIDGE REGRESSION

Christian Bauckhage

Deutsche Telekom Laboratories
10587 Berlin, Germany

ABSTRACT

Tensor-based approaches to visual object detection can drastically reduce the number of parameters in the training process. Compared to their vector-based counterparts, tensor methods therefore train faster, better manage noisy or corrupted training samples, and are less prone to over-fitting. In this paper, we show how to incorporate the kernel trick into tensor-based filter design. Dealing with object detection in cluttered natural environments, the method is shown to cope with substantially varying training data and a cascade of only two kernel tensor-filters is demonstrated to provide very reliable results.

***Index Terms*—** Color object detection, tensor-based filter design, kernel ridge regression

1. INTRODUCTION

The work reported in this paper was motivated by problems we encountered in the context of interactive vision systems. For instance, in a project on assistive technologies for the home environment [1, 2], users were supposed to interactively teach the system about objects in their surroundings. In scenarios like this, data acquisition and annotation happens online so that the data will hardly be flawless but noisy and rather imperfectly aligned. Also, in order for the user to not experience ennui and frustration, the data must be processed quickly and models must be learned rapidly. Moreover, as interactive technologies are usually intended for use in natural and unconstrained environments (see Fig. 5), we are in need of methods that perform reliably under a variety of illumination conditions, view directions, and scene clutter.

While modern classifier ensembles accomplish very robust detection (cf. e.g. [3, 4]), they require vast amounts of training data and are characterized by extensive training times. Traditional linear filters, on the other hand, train quickly but are easily affected by corrupted training data and perform less reliable under incoherent conditions [5]. Recent results, however, indicate that multilinear generalizations of linear approaches provide a reasonable compromise between the two extremes. Sparked by reports that understanding images as *multiindexed objects* or *higher order tensors* improves image coding and classification [6, 7, 8], tensor-based approaches have lately been applied to filter design. In [9, 10] they were reported to provide quickly trainable *and* robust tools for view-based object detection.

In this paper, we build upon these findings. We adopt the approach in [9] and show how to achieve even more robustness by incorporating the kernel trick. Dealing with color object detection in cluttered home environments, we present a simpler ensemble approach than in [9] and demonstrate that a filter cascade of only two levels performs very robust. First, however, we summarize the mathematical framework. Section 3 presents and discusses experimental results and a conclusion will end this contribution.

2. MATHEMATICAL BACKGROUND

Linear filtering of an image $\boldsymbol{\mathcal{I}}$ means to correlate it with a filter $\boldsymbol{\mathcal{W}}$ yielding a response map $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{I}} * \boldsymbol{\mathcal{W}}$. Therefore, if $\boldsymbol{\mathcal{X}}_{ij}$ denotes the image patch centered at image coordinates (i, j), the corresponding response is tantamount to the inner product $\mathcal{Y}_{ij} = \langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}}_{ij} \rangle$. This is the starting point for vector- and tensor-based filter design alike. However, since our method of tensor-based filter design makes use of least squares regression over vectors, we will first summarize least squares techniques for vector-based filter design.

2.1. Least Squares Regression

Given a sample of $l = 1, \ldots, N$ vectors $\mathbf{x}^l \in \mathbb{R}^m$ and a corresponding set of class labels y^l (typically in $\{-1, +1\}$), a suitable filter $\mathbf{w}$ results from minimizing the error

$$E(\mathbf{w}) = \sum_l \left(\langle \mathbf{w}, \mathbf{x}^l \rangle - y^l \right)^2 = \| \mathbf{X}\mathbf{w} - \mathbf{y} \|^2 \qquad (1)$$

where the $N \times m$ sample matrix $\mathbf{X}$ consists of the samples $\mathbf{x}^l$ and $\mathbf{y} \in \mathbb{R}^N$ contains the corresponding labels. This is a convex optimization problem that has a closed form solution. After setting the gradient $\nabla_{\mathbf{w}} E = 0$ and some algebra, one obtains:

$$\mathbf{w} = \left(\mathbf{X}^{\mathrm{T}} \mathbf{X} \right)^{-1} \mathbf{X}^{\mathrm{T}} \mathbf{y}. \qquad (2)$$

In the signal processing literature, this technique is often called synthetic discriminant filtering [5]; in machine learning it is known as linear discriminant analysis [11].

1-4244-1437-7/07/$20.00 ©2007 IEEE

2.2. Ridge Regression

Ordinary least squares regression is overly sensitive against outliers in the training data. The ridge regression approach aims to alleviate this and to control over-fitting by penalizing the norm of $\mathbf{w}$. This is done by introducing a regularization term into the error criterion: $E(\mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda\|\mathbf{w}\|^2$. Minimizing this error with respect to $\mathbf{w}$ is a convex problem, too, whose closed form solution is given by:

$$\mathbf{w} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}. \tag{3}$$

2.3. Kernel Ridge Regression

With some matrix algebra [11], one can show that $\mathbf{w}$ actually lies in the span of the training samples, i.e. $\mathbf{w} = \mathbf{X}^T\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is called the dual vector. The error criterion may thus be cast as $E(\boldsymbol{\alpha}) = \|\mathbf{X}\mathbf{X}^T\boldsymbol{\alpha} - \mathbf{y}\|^2 + \lambda\|\mathbf{X}^T\boldsymbol{\alpha}\|^2$ which is solved by $\boldsymbol{\alpha} = \left(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I}\right)^{-1}\mathbf{y}$. Now the matrix $\mathbf{X}\mathbf{X}^T$ of inner products between samples can be replaced by a kernel matrix $\mathbf{K}$. Since the inner products in $\mathbf{K}$ can be inner products in any space, one may also introduce nonlinear functions of the samples. In terms of $\mathbf{w}$, the kernel trick provides the solution:

$$\mathbf{w} = \mathbf{X}^T\left(\mathbf{K} + \lambda\mathbf{I}\right)^{-1}\mathbf{y}. \tag{4}$$

2.4. Tensor-Based Filter Design

Since our main interest is in color object detection and since color image patches can be thought of as third-order tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ where m_1 and m_2 denote the x- and y-resolution and m_3 counts the number of color channels (usually 3), we restrict the following discussion to third-order tensors.

Using Einstein's summation convention, the inner product of two third-order tensors $\boldsymbol{\mathcal{W}}$ and $\boldsymbol{\mathcal{X}}$

$$\langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle = \sum_{i,j,k} \mathcal{W}_{ijk}\mathcal{X}_{ijk}. \tag{5}$$

may be written $\langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}} \rangle = \mathcal{W}_{ijk}\mathcal{X}_{ijk}$. Given a training set $\{(\boldsymbol{\mathcal{X}}^l, y^l)\}$, where the $\boldsymbol{\mathcal{X}}^l$ are color image patches from two classes and the y^l denote class membership, we seek to solve

$$\boldsymbol{\mathcal{W}} = \underset{\tilde{\boldsymbol{\mathcal{W}}}}{\operatorname{argmin}} \sum_l \left(\mathcal{W}_{ijk}\mathcal{X}^l_{ijk} - y^l\right)^2. \tag{6}$$

Towards efficiency, we impose a structural constraint on $\boldsymbol{\mathcal{W}}$ and require it to be decomposable into R tensors of *rank* 1:

$$\boldsymbol{\mathcal{W}} = \sum_{r=1}^{R} \mathbf{u}^r \otimes \mathbf{v}^r \otimes \mathbf{w}^r, \tag{7}$$

```
Input: a training set ${\boldsymbol{\mathcal{X}}^l, y^l}_{l=1,...,N}$ of image patches
    $\boldsymbol{\mathcal{X}}^l \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ with class labels $y^l \in \{-1, +1\}$
Output: a rank-$R$ solution of a third-order
    filter tensor $\boldsymbol{\mathcal{W}} = \sum_r \mathbf{u}^r \otimes \mathbf{v}^r \otimes \mathbf{w}^r$

for $r = 1, \ldots, R$
  $t = 0$
  randomly initialize $\mathbf{u}^r(t)$
  orthonormalize $\mathbf{u}^r(t)$ w.r.t. $\{\mathbf{u}^1, \ldots, \mathbf{u}^{r-1}\}$
  randomly initialize $\mathbf{v}^r(t)$
  orthonormalize $\mathbf{v}^r(t)$ w.r.t. $\{\mathbf{v}^1, \ldots, \mathbf{v}^{r-1}\}$
  repeat
    $t \leftarrow t + 1$
    contract $x^l_k = \mathcal{X}^l_{ijk}\ u^r_i(t)\ v^r_j(t)$
    compute $\mathbf{w}^r(t) = \operatorname{argmin}_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$
    similarly update $\mathbf{v}^r(t)$
    similarly update $\mathbf{u}^r(t)$
  until $\|\mathbf{u}^r(t) - \mathbf{u}^r(t-1)\| \leq \epsilon \ \vee \ t > t_{\max}$
endfor
```

Fig. 1. Alternating least squares scheme to compute a filter $\boldsymbol{\mathcal{W}}$ given as a sum over outer products $\mathbf{u}^r \otimes \mathbf{v}^r \otimes \mathbf{w}^r$.

where $\otimes$ denotes the vector outer product. This constraint reduces the number of adjustable parameters from $m_1 \cdot m_2 \cdot m_3$ to $R \cdot (m_1 + m_2 + m_3)$ and allows for solving the problem in (6) in a series of simpler tasks. Consider the simplest case where $\boldsymbol{\mathcal{W}} = \mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}$. We can solve for $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ by means of the following steps. First, given random guesses for $\mathbf{u} \in \mathbb{R}^{m_1}$ and $\mathbf{v} \in \mathbb{R}^{m_2}$, compute the *tensor contractions*

$$x^l_k = \mathcal{X}^l_{ijk}\ u_i\ v_j, \quad l = 1, \ldots, N. \tag{8}$$

Stacking the resulting vectors $\mathbf{x}^l \in \mathbb{R}^{m_3}$ into a sample matrix $\mathbf{X}$ yields the familiar optimization problem for $\mathbf{w}$:

$$\mathbf{w} = \underset{\tilde{\mathbf{w}}}{\operatorname{argmin}} \|\mathbf{X}\tilde{\mathbf{w}} - \mathbf{y}\|^2. \tag{9}$$

Note that at this point either (2), (3), or (4) can be applied! Second, after solving for $\mathbf{w}$, the training set is contracted over $\mathbf{u}$ and $\mathbf{w}$ in order to update the estimate of $\mathbf{v}$. Third, a new estimate of $\mathbf{u}$ can be computed from the estimates of $\mathbf{v}$ and $\mathbf{w}$. Since the procedure starts with arbitrary vectors $\mathbf{u}$ and $\mathbf{v}$, it must be iterated until convergence. In our implementation, it stops, if $\|\mathbf{u}(t) - \mathbf{u}(t-1)\| \leq \epsilon$. Practical experience shows that this usually converges in less than 10 iterations.

The algorithm in Fig. 1 extends this alternating scheme to the derivation of tensor-templates of rank R. If $\boldsymbol{\mathcal{W}} = \sum_{r=1}^{k} \mathbf{u}^r \otimes \mathbf{v}^r \otimes \mathbf{w}^r$ is a k term solution for the projection tensor, a next triplet of vectors $(\mathbf{u}^{k+1}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1})$ can be found using the same procedure. Redundancy is avoided by otrhogonalizing the vectors $\mathbf{u}^{k+1}$ and $\mathbf{v}^{k+1}$ with respect to their predecessors.

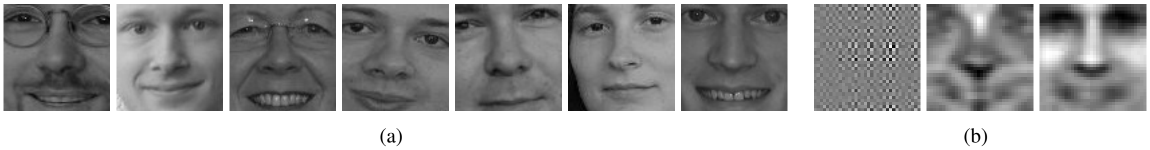

(a) (b)

Fig. 2. 2(a) Seven examples from a set of 35 face images used to train the templates on the right. 2(b) Templates resulting from applying ordinary (left), regularized (middle), and kernelized (right) least squares estimators in the algorithm in Fig. 1.

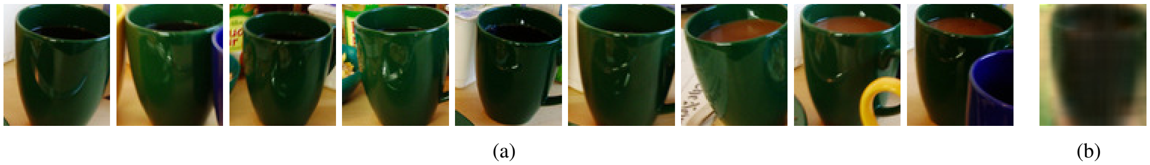

(a) (b)

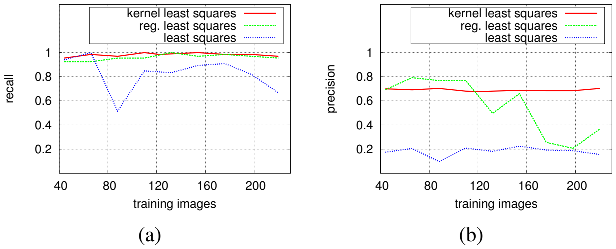
Fig. 3. 3(a) Nine examples from a set of 22 color image patches showing a green cup used to train the template on the right. 3(b) Template resulting from using Gaussian kernel least squares estimators in the alternating least squares algorithm in Fig. 1.

Compared to vector-based template design, the tensor-based method trains quicker. While vectorizing multivariate data of size $m_1 \times m_2 \times m_3$ would require the inversion of matrices of sizes $m_1 m_2 m_3 \times m_1 m_2 m_3$ during training, the matrix inverses in our algorithm are of considerably reduced sizes $m_3 \times m_3$, $m_2 \times m_2$ and $m_1 \times m_1$, respectively. In practice, we found that this accelerates training by several orders of magnitude. Also, the tensor-based approach does not suffer from *small sample sizes*. While for the vector-based approach the sample covariance matrices may be singular because the number of samples is much smaller than the dimension of the embedding space, the matrices in our algorithm will allow for inversion even if the sample set is small.

3. EXPERIMENTS

Figure 2 illustrates an experiment meant to convey the robustness of tensor-based template design using kernel ridge regression. We considered a sample of $N = 35$ grey-valued face images and, setting all labels y^l to $+1$, computed second-order tensor-templates ($R = 6$). Obviously, the ordinary least squares variant of our algorithm could not cope with the varying illuminations, head poses, and facial expressions in the sample (see the noisy template on the left of Fig. 2(b)). While the regularized variant of the algorithm learned a better but still ghostly face template, a kernelized variant using a Gaussian kernel produced the template on the right of Fig. 2(b). Here, we clearly recognize a smoothed, averaged face.

In another experiment, we considered object detection in natural home environments. Given a set of 88 pictures of a breakfast scene, 22 of these pictures were used for training, the remaining 66 for testing. A user was asked to quickly indicate the locations of a green cup seen in all the training images. Centered at the resulting coordinates, patches of size $91 \times 71 \times 3$ were cropped from the images, leading to a set of badly aligned examples of that cup (see Fig. 3(a)). A number of up to 198 counterexamples was randomly cropped from the background of the images, providing us with differently sized training sets of positive and negative examples. Given a C implementation running on a 3GHz Xeon PC, in each experiment, each of the variants of our algorithm produced third-order templates ($R = 6$) in less than a second.



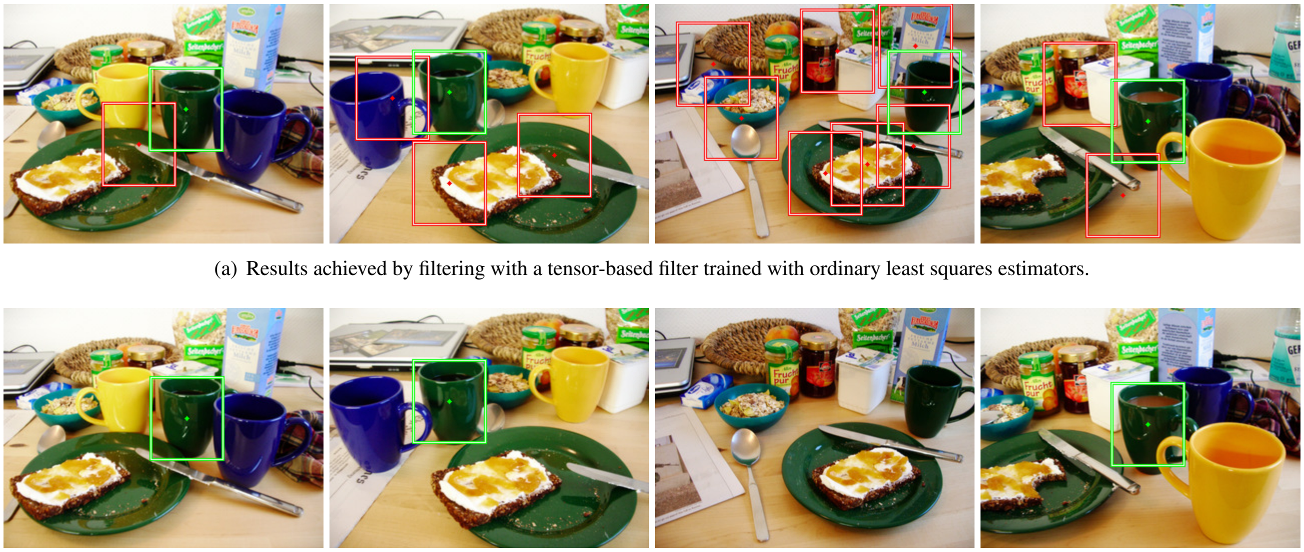
(a) (b)

Fig. 4. Recall and precision on the breakfast scene test set.

Figure 4 compares recall and precision rates we obtained from testing different filters. Tensor-templates trained with ridge- and kernel-ridge-regression clearly outperform the ones trained with ordinary least squares estimators. We attribute this to variances in the training sets and the ability of the former two methods to cope with these. However, only the kernel-based method seems unaffected by the size of the training set: For the the filters trained with ridge regression estimators, increasing the set size improves recall but diminishes precision and therefore obviously impairs their ability to cope with outliers. The filters trained with kernel estimators, in contrast, yield almost constant rates for both measures.

Trained with 66 examples the ordinary least squares approach actually produced a recall of 100% and a precision of 20%. Figure 5(a) illustrates that, despite the perfect recall, the many false positives prohibit the practical use of this filter.

For the same training set, the ridge- and kernel-ridge regression variants produced recall/precision of 92%/79% and 98%/71%, respectively. Since almost all false positives returned by these filters were systematically confused with the blue cup or the green platter in the scene, we experimented



(a) Results achieved by filtering with a tensor-based filter trained with ordinary least squares estimators.

(b) Results achieved by filtering with a tensor-based filter trained with kernel ridge regression estimators followed by a template matching step .

Fig. 5. Exemplary detection results obtained on the breakfast scene test set.

with a second filter stage, where image regions with high responses were matched against a template that was trained by applying the corresponding method to positives examples only; Fig. 3(b) shows such a template for the kernel variant. Again considering the training set of 66 samples, for the ordinary least squares variant this increased the precision to 24%; the other two variants now both achieved perfect precision. Exemplary results obtained from the kernel-based tensor-template with rates of 98%/100% for recall/precision are shown in Fig. 5(b).

4. CONCLUSION

This paper discussed a tensor-based approach to filter design that incorporates the kernel trick. The method was shown to be robust against outliers and substantial variation in the training data. Even from small sets of sloppily aligned examples, it derives filters that very reliably detect color objects in cluttered natural scenes. Therefore and since it trains rapidly, the framework presented in this paper appears well suited for application in interactive vision systems where online learning is pivotal.

5. REFERENCES

[1] C. Bauckhage, M. Hanheide, S. Wrede, T. Käster, M. Pfeiffer, and G. Sagerer, "Vision Systems with the Human in the Loop," *EURASIP J. on Applied Signal Processing*, vol. 2005, no. 14, pp. 2375–2390, 2005.

[2] S. Wrede, M. Hanheide, S. Wachsmuth, and G. Sagerer, "Integration and Coordination in a Cognitive Vision System," in *Proc. ICVS*, 2006, pp. 1–8.

[3] J. Kittler and A.R. Ahmadyfard, "Multiple Classifier System Approach to Model Pruning in Object Recognition," in *Proc. ECCV*, 2004, pp. 342–353.

[4] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *Int. J. of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[5] R. Brunelli and T. Poggio, "Template matching: Matched spatial filters and beyond," *Pattern Recognition*, vol. 30, no. 5, pp. 751–768, 1997.

[6] A. Shashua and A. Levin, "Linear Image Coding for Regression and Classification using the Tensor-rank Principle," in *Proc. CVPR*, 2001, vol. I, pp. 42–40.

[7] M. Vasilescu and D. Terzopoulos, "Multilinear Analysis of Image Ensembles: Tensorfaces," in *Proc. ECCV*, 2002, pp. 447–460.

[8] H. Wang and N. Ahuja, "Compact representation of multidimensional data using tensor rank-one decomposition," in *Proc. ICPR*, 2004, vol. I, pp. 44–47.

[9] C. Bauckhage, T. Käster, and J.K. Tsotsos, "Applying Ensembles of Multilinear Classifiers in the Frequency Domain," in *Proc. CVPR*, 2006, vol. I, pp. 95–102.

[10] S. Yan, D. Xu, L. Zhang, X. Tang, and H.-J. Zhang, "Discriminant Analysis with Tensor Representation," in *Proc. CVPR*, 2005, vol. I, pp. 526–532.

[11] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.