Wavelet Frame Accelerated Reduced Support Vector Machines

Matthias Rätsch, Gerd Teschke, Sami Romdhani, and Thomas Vetter, *Member, IEEE*

***Abstract*—In this paper, a novel method for reducing the runtime complexity of a support vector machine classifier is presented. The new training algorithm is fast and simple. This is achieved by an over-complete wavelet transform that finds the optimal approximation of the support vectors. The presented derivation shows that the wavelet theory provides an upper bound on the distance between the decision function of the support vector machine and our classifier. The obtained classifier is fast, since a Haar wavelet approximation of the support vectors is used, enabling efficient integral image-based kernel evaluations. This provides a set of cascaded classifiers of increasing complexity for an early rejection of vectors easy to discriminate. This excellent runtime performance is achieved by using a hierarchical evaluation over the number of incorporated and additional over the approximation accuracy of the reduced set vectors. Here, this algorithm is applied to the problem of face detection, but it can also be used for other image-based classifications. The algorithm presented, provides a 530-fold speedup over the support vector machine, enabling face detection at more than 25 fps on a standard PC.**

***Index Terms*—Cascaded evaluation, coarse-to-fine classifier, face detection, machine learning, over-complete wavelet transform (OCWT), reduced support vector machine (RVM).**

I. Introduction

IMAGE classification tasks are time consuming. For instance, detecting a specific object in an image, such as a face, is computationally expensive, as all the pixels of the image are potential object centers. Hence, all the pixels must be classified.

Recently, more efficient methods have emerged based on a cascaded evaluation of hierarchical filters: image patches easy to discriminate are classified by a simple and fast filter, while patches that resemble the object of interest are classified by more-involved and slower filters. In the area of face detection [1], cascaded based classification algorithms were introduced by Keren *et al.* [2], by Romdhani *et al.* [3] and by Viola and Jones [4]. To apply the detector, proposed by Keren *et al.* [2], the negative examples need to be Boltzmann distributed and smooth. This assumption could increase the number of false positive in presence of a cluttered background. Romdhani *et al.* [3] use a cascaded reduced set vectors (RSV) expansion of a support vector machine [5]. The bottleneck of [3] is that at least one convolution of a 20×20 filter has to be carried out on the full image, resulting in a computationally expensive evaluation of the kernel with an image patch. Kienzle *et al.* [6] present an improvement of this method, where the first (and only the first) RSV is approximated by a separable filter. Viola and Jones [4] use Haar-like oriented edge filters having a block like structure enabling a very fast evaluation by use of an integral image. These filters are weak, in the sense that their discrimination power is low. They are selected, among a finite set, by the AdaBoost algorithm that yields the ones with the best discrimination. A drawback of their approach is that it is not clear that the cascade achieves optimal generalization performances. Practically, the training proceeds by trial and error, and often, the number of filters per stage must be manually selected so that the false positive rate decreases smoothly. Another drawback of the method is that the set of available filters is limited and has to be selected manually. The training for the classifier is "on the order of weeks" [4, Section 5.2], as every filter (about 10^5) is evaluated on the whole set of training examples and this is done every time a filter is added to a stage of the cascade.

Taking the above mentioned problems into account, we developed a novel classification algorithm. The following features make the algorithm accurate and efficient.

1) **Support Vector Machine:** Use of an SVM classifier that is known to have optimal generalization capabilities.
2) **Reduced Support Vector Machine:** The RVM uses a reduced set of support vectors [3].
3) **Double Cascade:** For nonsymmetric data (i.e., only few positives to many negatives), we achieve an early rejection of easy to discriminate vectors. It is obtained by the two following cascaded evaluations over coarse-to-fine wavelet approximated reduced set vectors (W-RSVs): i) **Cascade over the number of used W-RSVs** and ii) **Cascade over the resolution levels of each W-RSV**. The double cascade constitutes one of the major novelties of our approach. The tradeoff between accuracy and speed is essentially reduced.
4) **Integral Images:** As the RSVs are approximated by a Haar wavelet transform, the integral image method is used for their evaluation, similarly to [4].
5) **Wavelet Frame:** We use an over-complete wavelet system to find the best representation of the RSVs.

The learning stage of our proposed wavelet approximated reduced SVM (**W-RVM**) is fast, straightforward, automatic and

Manuscript received September 05, 2006; revised April 17, 2008. Current version published November 12, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Srdjan Stankovic.

M. Rätsch, S. Romdhani, and T. Vetter are with the Computer Science Department, University of Basel, Bernoullistrasse 16, CH-4057 Basel, Switzerland (e-mail: matthias.raetsch@unibas.ch; sami.romdhani@unibas.ch; thomas.vetter@unibas.ch).

G. Teschke is with the Konrad Zuse Institute Berlin, Takustr. 7, D-14195 Berlin, Germany (e-mail: teschke@zib.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2008.2001393

does not require the manual selection of ad-hoc parameters. For example, the training time (Section III) is 2 h which is a vast improvement over former detectors.

The paradigm of our method is that, instead starting by a poor classifier and getting more complex by heuristical knowledge, we first build a classifier that is proven to have optimal generalization capabilities. The focus then becomes runtime efficiency while maintaining the classifier's optimal accuracy. To avoid complex search over the parameter space, we do not start with the full parameter space, but with the proved optimal performance of an SVM. Then we reduce the complexity by a reduced vector set and the over-complete wavelet approximation. Hence, our approach is straightforward.

In our approach, we apply an over-complete wavelet transform (OCWT) to the reduced support vector machine itself, and not of the input space as a preprocessing like [7] and [8].

This paper presents the coherent and complete frame work of our approach where we summarize and extend the conference papers [9], [10], [3]. The improvement of [9] compared to [10] are the features 3. ii) and 5. (see above): The simulated annealing optimization using morphological filters was replaced by a sparse wavelet frame representation of the RSVs. Simulated annealing does not provide the global optimum of the RVM approximation in all cases and it is difficult to adjust the resolution level.

In this paper, we take advantage of recent progress in wavelet analysis: the optimality of sparse signal approximation (rectangular structure) in wavelet space. Moreover, we show the double cascade structure of the learning and detection process that is obtained by the proposed recursive refinement of the wavelet frame representation of the RSVs.

In addition, we show in Section II-B3 that the wavelet frame approach provides an upper bound of the hyperplane approximation error. Exploring this characteristic the training of the W-RVM works without heuristics and is fast. Also as an expansion, we show in Section II-B3 the relation between the hyperplane approximation error of the decision functions and a training parameter to control the tradeoff between sparsity and approximation. As demonstrated in Section IIII-C1 the parameter for setting the approximation accuracy does not play a decisive role, opposite to former methods, using only one resolution level.

The paper is organized as follows: Section II details our novel training (Section II-B) and detection algorithm (Section II-C). It is shown in Section III that the new expansion yields a comparable accuracy to the SVM while providing a significant speedup. In addition to the mentioned papers [9], [10], we carried out experiments on well-known databases, like FERET [11] to provide the comparability to other approaches.

II. Wavelet Frame Approximated Support Vector Machine

Support vector machines (SVM) [5] are well-known for good generalization capabilities. Their decision function has the form

$$y(\mathbf{x}) = \sum_{i=1}^{N_x} \alpha_i \cdot k(\mathbf{x}, \mathbf{x}_i) + b \tag{1}$$

where $k(\cdot,\cdot)$ represents the kernel, which can be shown to compute the dot products in associated feature spaces F, i.e., $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$. The function $\Phi : \mathcal{X} \to F, \mathbf{x} \mapsto \Phi(\mathbf{x})\cdot$ maps the data $\mathbf{x}$ (in our case, a vector of 400 grey values of the 20×20 observation window) into F. Although F can be high-dimensional, it is usually not necessary using the kernel function to explicitly work in that space. The SVM decision hyperplane is determined by $\Psi_{\mathrm{SVM}} = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$, with N_x support vectors $\mathbf{x}_i$ with nonvanishing coefficients α_i.

We performed experiments with linear, polynomial and RBF kernels and it turned out that RBF performed best for our specific classification problem. We also focus in this paper on Gaussian kernel, because we can show in Section II-B3 in an analytically way the necessary approximation bounds. The advantage of polynomial kernels is that the reduced set vectors can be derived explicitly, even for nonhomogenous kernels [12], [13]. However, for good performance with polynomial kernels, a feature space normalization is necessary. The focal point of this paper is the transform of the SVM and not of the feature space.

In order to improve the runtime performance, it is proposed in [14] to approximate the SVM by a reduced SVM (RVM) in combination with a cascaded evaluation as in [3]. The RVM aims to approximate the SVM by a *smaller* set of new reduced set vectors (RSVs), $\mathbf{z}_i$ instead of the support vectors, $\mathbf{x}_i$. The RVM approach provides a significant speedup over the SVM, but is still not fast enough, as the image has to be convolved in steps of full convolutions, e.g., by 20×20 RSVs. The algorithm presented in this paper improves this method since it does not require this convolution to be performed explicitly. Instead, it approximates the RSVs by Haar-like vectors and computes the evaluation of a patch using an integral image of the input image. They can be used to compute very efficiently the dot (or inner) product of an image patch with an image that has a block-like structure, i.e., rectangles of constant values.

A. Integral Images Based on Haar-Like W-RSVs

During an RVM evaluation, most of the time is spent for kernel evaluations. In the case of the Gaussian kernel, $k(\mathbf{x}, \mathbf{z}_i) = \exp(-\|\mathbf{x} - \mathbf{z}_i\|^2/(2\sigma^2))$, chosen here, the computational cost is spent in evaluating the norm of the difference between a patch and an RSV. This norm can be expanded as follows: $\|\mathbf{x} - \mathbf{z}_i\|^2 = \mathbf{x}'\mathbf{x} - 2\mathbf{x}'\mathbf{z}_i + \mathbf{z}_i'\mathbf{z}_i$. As $\mathbf{z}_i$ is independent of the input image, it can be precomputed. The sum of squares of the pixels of a patch of the input image, $\mathbf{x}'\mathbf{x}$ is efficiently computed using the integral image ([15], [4]) of the squared pixel values of the input image. As a result, the computational cost of this expression is determined by the term $2\mathbf{x}'\mathbf{z}_i$.

The novelty of our approach is the approximation of the RSVs, $\mathbf{z}_i$, by optimally wavelet frame approximated reduced set vectors (W-RSVs), $\mathbf{u}_i$ which have a block-like structure, as seen in Fig. 1. Optimally approximated means here the usage of an optimally shifted wavelet basis that represents the image as sparse as possible. If $\mathbf{u}_i$ is an image patch with rectangles of constant grey levels, then the term $2\mathbf{x}'\mathbf{u}_i$ can be evaluated very efficiently using the integral image. The term can be re-sorted by $2\mathbf{x}'\mathbf{u}_i = 2\sum_{k=1}^{D} x_k u_{i,k} = 2\sum_{r=1}^{R_i} v_{i,r} \sum_{j=1}^{D_r} x_j$ where D is the dimension of the vectors (e.g., 400 pixel by a patch-size 20×20), R_i is the number of rectangles of $\mathbf{u}_i, v_{i,r}$ the grey values of the rectangle r and $x_j, j = 1, \ldots, D_r$ all pixel-values of $\mathbf{x}$ within the r-th rectangle. Because $\sum_{j=1}^{D_r} x_j$ can be computed by the addition of three pixels of the integral image of the input image [15], the dot product is evaluated in constant

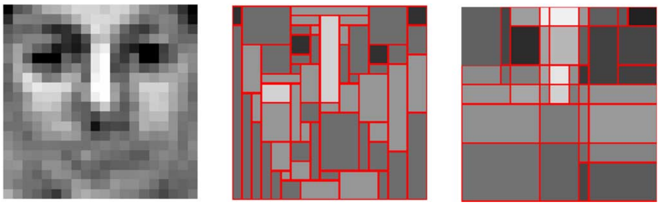

Fig. 1. Examples for Haar-like approximations: RSV (*left*) approximated using morphological filter (H-RSV [10], *middle*) and using an OCWT (W-RSV, *right*). The OCWT representation meets optimally the local image structure. The ratio of the decreasing of the hyperplane distance to the used operations (see Section II-B5) is more efficient for the W-RSV (0.73) than for the H-RSV (0.51).

time by four additions per rectangle and one multiplication per grey value.

B. Learning Process

In contrast to several approaches like [7] and [8], we do not wavelet-transform the input images as a preprocessing at runtime. The novelty is that we apply the OCWT to the RVM itself.

1) Soft-Shrinkage to Build Rectangular Structured W-RSVs: In order to exploit the integral image method a block-like approximation of the reduced set vectors must be used, i.e., they must have a rectangular (Haar-like) structure with piecewise constant grey values. Therefore, we use Haar wavelets and not wavelets with more vanishing moments (e.g., Daubechies wavelets of higher order), even if they would in general result in a more sparse approximation.

We are searching for an approximation of a given image $\mathbf{z}$ by a piecewise block structured image $\mathbf{u}$ which is as sparse as possible. This optimization problem can be casted in the following variational form:

$$\min_{\hat{\mathbf{u}}} \left\{ \|\mathbf{z} - \hat{\mathbf{u}}\|_{L_2}^2 + 2\mu|\hat{\mathbf{u}}|_{B_1^1(L_1)} \right\} \quad (2)$$

where $B_1^1(L_1)$ denotes a particular Besov semi-norm (for more details, we refer the reader to [16], [17] and for a detailed discussion of the problem to [18]). It is known that the Besov (semi) norm of a given function can be expressed by means of its wavelet coefficients. In two spatial dimensions the Besov penalty is nothing else than a ℓ_1 constraint on the wavelet coefficients (promoting sparsity as required).

The minimization of (2) is easily obtained: Let $\{\psi_\lambda\}_{\lambda\in\Lambda}$ be the underlying wavelet basis, where Λ is the index set over all possible locations, scalings and wavelet species. Then we may express $\mathbf{z}$ and $\hat{\mathbf{u}}$ as follows: $\mathbf{z} = \sum_{\lambda\in\Lambda} z_\lambda \psi_\lambda, \hat{\mathbf{u}} = \sum_{\lambda\in\Lambda} \hat{u}_\lambda \psi_\lambda$, where $z_\lambda = \langle \mathbf{z}, \psi_\lambda\rangle$ and $\hat{u}_\lambda = \langle \hat{\mathbf{u}}, \psi_\lambda \rangle$. Thus, we may completely rewrite (2) as

$$\mathbf{u} = \arg\min_{\hat{\mathbf{u}}} \sum_{\lambda\in\Lambda} \left\{ (z_\lambda - \hat{u}_\lambda)^2 + 2\mu|\hat{u}_\lambda| \right\}. \quad (3)$$

Minimizing summand-wise, we obtain the following explicit expression for the optimum u_λ, see, e.g., [19]:

$$u_\lambda = S_\mu(z_\lambda) = \mathrm{sgn}(z_\lambda)\max\{|z_\lambda| - \mu, 0\} \quad (4)$$

where S_μ is the soft-shrinkage operation with threshold μ. Consequently, the optimum $\mathbf{u}$ is simply obtained by soft-shrinking the wavelet coefficients of $\mathbf{z}$, i.e.,

$$\mathbf{u} = \sum_{\lambda\in\Lambda} S_\mu(z_\lambda)\psi_\lambda = W^{-1}S_\mu(W\mathbf{z}) \quad (5)$$

where W stands for the wavelet transform operator.

2) Translated Wavelet Bases to Overcome the Windowing Effect: Typically, orthogonal or so-called nonredundant representations and filtering very often creates artifacts in terms of undesirable oscillations or nonoptimally represented details, which manifest themselves as ringing and edge blurring (also called Gibbs or windowing effect). For our purpose, it is essential to pick a representation that optimally meets the local image structure (see Fig. 1). The most promising method for adequately solving the windowing problem has its origin in translation invariance (the method of cycle spinning, see, e.g., [20]), i.e., representing the image by all possible shifted versions of the underlying (Haar) wavelet basis. But contrary to the idea of introducing redundancy by averaging over all possible representations of $\mathbf{z}$, we aim to pick only that one that optimally meets the given image structure.

In order to give a rough sketch of this technique, assume that we are given an RSV $\mathbf{z}$ with $2^M \times 2^M$ pixel. Following the cycle-spinning approach, see again [20], we have to compute $2^{2(M+1-j_0)}$ different representations of $\mathbf{z}$ with respect to the $2^{2(M+1-j_0)}$ translates, s of the underlying wavelet basis. The scale j_0 denotes the coarsest resolution level of $\mathbf{z}$. The family $\{\mathbf{z}^s\}_s$ generated this way serves now as our reservoir of possible wavelet representations of one single $\mathbf{z}$. The best shift s^* is that one for which we have a minimal discrepancy to the SVM hyperplane per operations for the kernel-evaluation. We evaluate all possible local shifts (in our case $s = 64$); hence, the global optimum shift is guaranteed (see Section II-B5).

3) Hyperplane Approximation: We use a two stage hyperplane approximation from the original SVM to the reduced SVM (RVM) and from the RVM to the wavelet approximated reduced SVM (W-RVM). The first reduction step was computing the RVM by minimizing the hyperplane distance $\|\Psi_{\mathrm{SVM}} - \Psi_{\mathrm{RVM}}\|_F$ in the feature space F [10] and [3]. This yields $\Psi_{\mathrm{RVM}} = \sum_{i=1}^{N_z} \beta_i \Phi(\mathbf{z}_i)$ with the mapping function $\Phi : \mathcal{X} \to F, \mathbf{z} \mapsto \Phi(\mathbf{z})$ as used for the SVM. As outlined above, an essential improvement can be achieved by accelerating the numerical integration. To this end, we have suggested the use of Haar-like sparse approximations $\mathbf{u}_i$ of $\mathbf{z}_i$ that generates rectangular representations of the images and fits thus well with the concept of integral images. Replacing $\mathbf{z}_i$ by $\mathbf{u}_i$ amounts to $\sum_{i=1}^{N_z} \beta_i \Phi(\mathbf{u}_i)$. The change of the supporting vectors might likely require a slight adjustment of the β_i's which is done iteratively (see below), i.e., the second hyperplane approximation we are proposing finally reads as

$$\Psi_{\mathrm{W-RVM}} = \sum_{i=1}^{N_z} \gamma_i \Phi(\mathbf{u}_i). \quad (6)$$

The natural question that arises is how well approximates the reduced and Haar-like designed $\Psi_{\mathrm{W-RVM}}$ (6) the original SVM Ψ_{SVM}, i.e., we have to consider the quantity

$$\begin{aligned} \|\Psi_{\mathrm{SVM}} - \Psi_{\mathrm{W-RVM}}\|_F \leq & \|\Psi_{\mathrm{SVM}} - \Psi_{\mathrm{RVM}}\|_F \\ & + \|\Psi_{\mathrm{RVM}} - \Psi_{\mathrm{W-RVM}}\|_F \end{aligned} \quad (7)$$

where the first misfit term on the right hand side is minimized through the iterative method in [10] and [3]. It remains to analyze the second discrepancy between Ψ_{RVM} and $\Psi_{\mathrm{W-RVM}}$.

By making use of kernel-based evaluations of the inner products (and using $k(\mathbf{z}_i, \mathbf{z}_i) = 1$) and Cauchy–Schwarz, we obtain

$$\begin{aligned}
&\|\Psi_{\mathrm{RVM}} - \Psi_{\mathrm{W-RVM}}\|_F^2 \\
&\le \left(\sum_{i=1}^{N_z} \|\beta_i \Phi(\mathbf{z}_i) - \gamma_i \Phi(\mathbf{u}_i)\|_F \right)^2 \\
&= \langle \mathbf{1}_{N_z \mathrm{x} 1}, (\|\beta_1 \Phi(\mathbf{z}_1) - \gamma_1 \Phi(\mathbf{u}_1)\|_F, \ldots, \\
&\quad |\beta_{N_z} \Phi(\mathbf{z}_{N_z}) - \gamma_{N_z} \Phi(\mathbf{u}_{N_z})\|_F) \rangle^2 \\
&\le N_z \sum_{i=1}^{N_z} \|\beta_i \Phi(\mathbf{z}_i) - \gamma_i \Phi(\mathbf{u}_i)\|_F^2 \\
&= N_z \sum_{i=1}^{N_z} \left\{ \beta_i^2 + \gamma_i^2 - 2\gamma_i \beta_i k(\mathbf{z}_i, \mathbf{u}_i) \right\} \\
&= N_z \left\{ \sum_{i=1}^{N_z} (\beta_i - \gamma_i)^2 \right. \\
&\quad \left. + 2 \sum_{i=1}^{N_z} \gamma_i \beta_i (1 - k(\mathbf{z}_i, \mathbf{u}_i)) \right\} \\
&= N_z \left\{ \|\beta - \gamma\|^2 + 2 \sum_{i=1}^{N_z} \gamma_i \beta_i (1 - k(\mathbf{z}_i, \mathbf{u}_i)) \right\}. \qquad (8)
\end{aligned}$$

Now, when choosing the Gaussian kernel with kernel parameter, σ (optimized by the SVM training [5]), we may approximate $1 - k$ in (8) as follows:

$$\begin{aligned}
1 - k(\mathbf{z}_i, \mathbf{u}_i) &= 1 - \exp\left(\frac{-\|\mathbf{z}_i - \mathbf{u}_i\|^2}{2\sigma^2} \right) \\
&= \frac{\|\mathbf{z}_i - \mathbf{u}_i\|^2}{2\sigma^2} + \mathcal{O}(\|\cdot\|^4). \qquad (9)
\end{aligned}$$

Thus, the data misfit discrepancy is directly controlled by the ℓ_2 distance of the sparse approximation $\mathbf{u}_i$ of $\mathbf{z}_i$ (which is minimized under sparsity constraints) and the distance $\|\beta - \gamma\|$. Thus, up to higher order terms, we achieve

$$\begin{aligned}
\|\Psi_{\mathrm{RVM}} - \Psi_{\mathrm{W-RVM}}\|_F^2 \lessapprox N_z &\left\{ \|\beta - \gamma\|^2 \right. \\
&\left. + \sigma^{-2} \sum_{i=1}^{N_z} \gamma_i \beta_i \|\mathbf{z}_i - \mathbf{u}_i\|^2 \right\} \qquad (10)
\end{aligned}$$

where the relation between the error of the wavelet approximated reduced set vectors and the threshold parameter μ needs to be made. This is important to control the tradeoff between sparsity (i.e., computational cost) and the approximation (classification) preciseness per approximated vector.

At first, we consider the difference of the reduced and wavelet approximated reduced set vectors and express them by means of the corresponding wavelet coefficients, i.e.,

$$\|\mathbf{z}_i - \mathbf{u}_i\|^2 = \sum_{\lambda \in \Lambda} (z_{i,\lambda} - S_\mu(z_{i,\lambda}))^2.$$

Assuming further that $\mathbf{z}$ consists of $2^M \times 2^M$ pixel and $(z_{i,\lambda} - S_\mu(z_{i,\lambda}))^2 \le \mu$ using (4), we have

$$1 - k(\mathbf{z}_i, \mathbf{u}_i) \le 1 - \exp\left(\frac{-2^{2M}\mu^2}{2\sigma^2} \right).$$

Applying this to (8), an upper bound E for the worst case error is then given by

$$\begin{aligned}
&\|\Psi_{\mathrm{RVM}} - \Psi_{\mathrm{W-RVM}}\|_F^2 \\
&\le N_z \left\{ \|\beta - \gamma\|^2 \right. \\
&\quad \left. + 2\left(1 - \exp\left(\frac{-2^{2M}\mu^2}{2\sigma^2} \right) \right) \sum_{i=1}^{N_z} \beta_i \gamma_i \right\} \\
&=: E(\mu).
\end{aligned}$$

Neglecting higher order terms of the $\exp$ series, we may write

$$E(\mu) \approxeq N_z \left(\sigma^{-2} 2^{2M} \mu^2 \sum_{i=1}^{N_z} \beta_i \gamma_i + \|\beta - \gamma\|^2 \right). \qquad (11)$$

From the last formula, we see that the influence of μ is of quadratic nature which assures a rapid error decay of the left hand summand. The quantity $\|\beta - \gamma\|^2$ will be studied below when we have exploited a rule for deriving the vector γ. In the limit case, $\mu \to 0$, we then achieve $\lim_{\mu \to 0} E(\mu) = 0$, which shows that the proposed scheme acts in the limit case as the RVM. For the case in which we really achieve complexity reduction by sparsity and, thus, a significant gain in computational time and cost, we refer to Section III.

4) Hierarchical Evaluation via Resolution Levels: The early rejection of easy to discriminate vectors is achieved by a double cascade. The inner cascade is a hierarchy over the number $i = 1, \ldots, N_z$ of incorporated W-RSVs, $\mathbf{u}_i^l$. After incorporating a certain number of W-RSVs with a constant resolution level l it is more efficient to improve the approximation accuracy of the first (already incorporated) vectors. Hence, we train in Section II-B5 $l = 0, \ldots, L$ sets of W-RSVs for the outer cascade of coarse-to-fine resolution levels. To use the cascade over the resolution levels as inner loop and over the W-RSVs as outer loop should result in similar performance. To keep the method simple, we only propose one realization of the double cascade. The tradeoff between the two cascades is determined in Section II-C. To exploit these cascades is the superior way to reject most image points by only few operations. Moreover this novel method is robust since the adjustment of only one optimal resolution level was sensitive in [10]. The proposed evaluation selects the most efficient approximation accuracy automatically at detection time based on the image patch to be classified. In contrast to former methods, the tradeoff between accuracy and speed is smooth, so that image points are rejected earlier. Therefore, the approach is robust, not sensitive to the parameter choice at training time, simple to use, and fast.

5) Algorithm to Generate Hierarchically Refined W-RSVs: The algorithm is based on residual Haar wavelet approximations of the RSVs $\mathbf{z}_i$ which are precomputed by minimizing $\|\Psi_{\mathrm{SVM}} - \Psi_{\mathrm{RVM}}\|_F^2$ via the algorithm suggested in [3].

Before presenting the algorithm, we introduce the basic quantities. To find the optimal match (see OCWT in Section II-B2), we use a translated wavelet bases with an offset up to $2^J \times 2^J$. To avoid the ringing effect $J = \log_2(2^M/4)$ (i.e., about a quarter of the dimensions of $\mathbf{z}$) is sufficient. Starting with computing 2^{2J} different initial Haar-like approximations $\mathbf{r}_i^{0,s}$ by (5), where

$s \in \{1,\dots,2^J\}^2$ is the shift of the underlying Haar wavelet basis, we recursively define for $l = 0,\dots,L$ and $i = 1,\dots,N_z$

$$\mathbf{u}_i^l = \sum_{j=0}^{l} \mathbf{r}_i^{j,s^*}$$

$$\mathbf{r}_i^{l+1,s} = (W^s)^{-1} S_\mu \left(W^s \left(\mathbf{z}_i - \mathbf{u}_i^l \right) \right) \qquad (12)$$

where the shift s^* denotes the best shift (selected by an optimally criterion introduced below) of the residual at resolution level l, see Fig. 2. Note that s^* may differ for each $\mathbf{r}_i^{l,s}$. Within this setting each reduced set vector $\mathbf{z}_i$ is then approximated at level l by $\mathbf{u}_i^l$. The benefit of the residual structure is that i) $\mathbf{u}_i^l$ converge to $\mathbf{z}_i$, if $l \to \infty$, ii) we can store all the residuals, and, thus, they do not need to be recomputed in the cascade step when tuning the resolution (i.e., the accuracy of the W-RSV representation) from coarse to fine, and iii) the evaluation of the kernel at runtime is more efficient [detailed later in (20) in Section II-C]. incorporate the next optimal W-RSV, we have to evaluate the computational cost and the discrepancy of the cascaded W-RVM to the original SVM. Such a discrepancy depends on the resolution level l and the number i of incorporated W-RSVs. Only $\mathbf{r}_i^{l,s}$ changes for the optimization steps over all offsets s. Therefore, using the expanded form (12) in (6) the discrepancy of the hyperplanes becomes

$$\delta_i^l(s) = \left\| \Psi_{\text{SVM}} - \sum_{k=1}^{i-1} \gamma_k^{l,i} \Phi\left(\mathbf{u}_k^l\right) - \gamma_i^{l,i} \Phi\left(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}\right) - \sum_{k=i+1}^{N_z} \gamma_k^{l,i} \Phi\left(\mathbf{u}_k^{l-1}\right) \right\|_F^2 \qquad (13)$$

where we set $\mathbf{u}_i^{-1} = 0$. The cascade structure is, thus, achieved when adding residuals $i \to i+1$ and then, after reaching $i = N_z$, passing to the next level $l \to l+1$, i.e., subsequently adding $\mathbf{r}_i^{l,s}$. Note that for each added residual $\mathbf{r}_i^{l,s}$, we have to compute a new vector $\gamma^{l,i} = (\gamma_1^{l,i},\dots,\gamma_{N_z}^{l,i})'$. Since we are searching for the best shift s for $\mathbf{r}_i^{l,s}$ and the optimal $\gamma^{l,i}$, we have to minimize $\delta_i^l(s)$. The optimal vector $\gamma^{l,i}$ can be computed explicitly. Introducing the $N_x \times N_z$ matrix

$$\Phi_{\mathbf{x},\mathbf{u}}^{l,i,s} = \begin{pmatrix} \left\langle \Phi(\mathbf{x}_1), \Phi\left(\mathbf{u}_1^l\right) \right\rangle & \dots & \left\langle \Phi\left(\mathbf{x}_{N_x}\right), \Phi\left(\mathbf{u}_1^l\right) \right\rangle \\ \vdots & \ddots & \vdots \\ \left\langle \Phi(\mathbf{x}_1), \Phi\left(\mathbf{u}_{i-1}^l\right) \right\rangle & \dots & \left\langle \Phi\left(\mathbf{x}_{N_x}\right), \Phi\left(\mathbf{u}_{i-1}^l\right) \right\rangle \\ & \mathbf{v}^s & \\ \left\langle \Phi(\mathbf{x}_1), \Phi\left(\mathbf{u}_{i+1}^{l-1}\right) \right\rangle & \dots & \left\langle \Phi\left(\mathbf{x}_{N_x}\right), \Phi\left(\mathbf{u}_{i+1}^{l-1}\right) \right\rangle \\ \vdots & \ddots & \vdots \\ \left\langle \Phi(\mathbf{x}_1), \Phi\left(\mathbf{u}_{N_z}^{l-1}\right) \right\rangle & \dots & \left\langle \Phi\left(\mathbf{x}_{N_x}\right), \Phi\left(\mathbf{u}_{N_z}^{l-1}\right) \right\rangle \end{pmatrix}$$

with the ith row

$$\mathbf{v}^s = \left(\left\langle \Phi(\mathbf{x}_1), \Phi\left(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}\right) \right\rangle, \dots, \left\langle \Phi\left(\mathbf{x}_{N_x}\right), \Phi\left(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}\right) \right\rangle \right)$$

and the same way the $N_z \times N_z$ matrix $\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s}$ with entries $\langle \Phi(\mathbf{u}_i^l), \Phi(\mathbf{u}_{i'}^{l'}) \rangle$ but where the ith row is replaced with

$$\mathbf{w}^s = \left(\left\langle \Phi\left(\mathbf{u}_1^l\right), \Phi\left(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}\right) \right\rangle, \dots, \left\langle \Phi\left(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}\right), \Phi\left(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}\right) \right\rangle, \dots, \left\langle \Phi\left(\mathbf{u}_{N_z}^{l-1}\right), \Phi\left(\mathbf{u}_i^{l-1} + \mathbf{r}_i^{l,s}\right) \right\rangle \right)$$

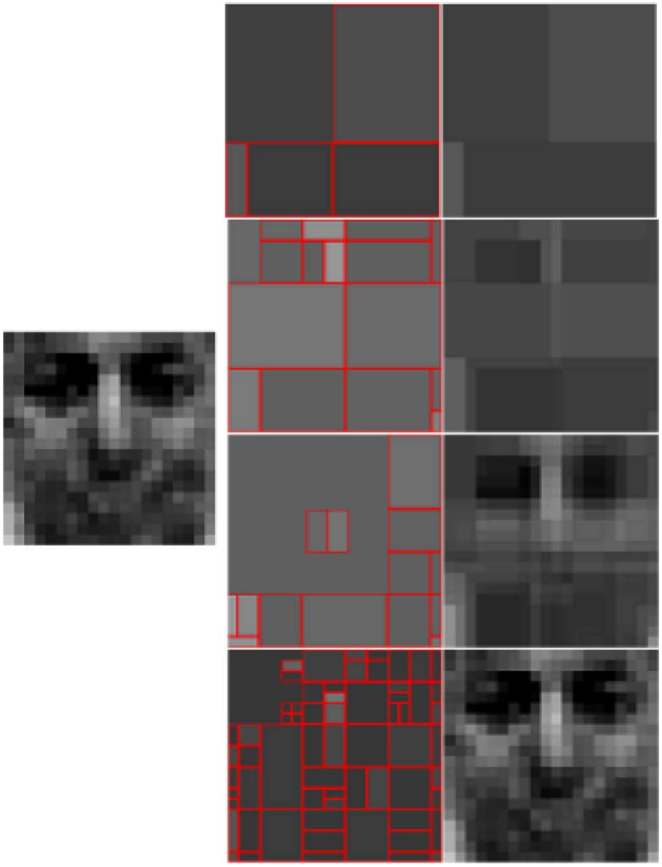

Fig. 2. Example of recursively approximating an RSV. *Left*: an RSV $\mathbf{z}_i$; *right*: W-RSV $\mathbf{u}_i^l$ at different resolution levels (top to bottom: $l = 0, 1, 9, 18$); *middle*: related residuals $\mathbf{r}_i^{l,s^*}$ (top to bottom: $l = 0, 1, 9, 18$).

and ith column with $(\mathbf{w}^s)'$, we recast the discrepancy $\delta_i^l(s)$ (13) as follows:

$$\delta_i^l(s) = \|\Psi_{\text{SVM}}\|_F^2 - 2(\gamma^{l,i})' \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s} \alpha + (\gamma^{l,i})' \Phi_{\mathbf{u},\mathbf{u}}^{l,i,s} \gamma^{l,i} \qquad (14)$$

where α is the vector of the nonvanishing coefficients of the SVM hyperplane $\Psi_{\text{SVM}} = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$.

Evaluating the derivative of the discrepancy (14) and setting it to 0, the optimal $\gamma^{l,i}$ is then obtained by

$$\gamma^{l,i}(s) = \left(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s}\right)^{-1} \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s} \alpha \qquad (15)$$

and depends thus on s. With the explicit expression (15), the discrepancy (14) becomes

$$\delta_i^l(s) = \|\Psi_{\text{SVM}}\|_F^2 - \alpha' \left(\Phi_{\mathbf{x},\mathbf{u}}^{l,i,s}\right)' \left(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s}\right)^{-1} \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s} \alpha. \qquad (16)$$

This, of course, requires the existence of $(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s})^{-1}$ what clearly means then linear independency of all involved $\Phi(\cdot)$'s. If this cannot be assured, we have to consider a regularized version of $\delta_i^l(s)$, namely

$$\delta_i^l(s) = \|\Psi_{\text{SVM}}\|_F^2 - 2(\gamma^{l,i})' \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s} \alpha + (\gamma^{l,i})' \left(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s} + \rho\right) \gamma^{l,i}.$$

This yields

$$\gamma^{l,i}(s) = \left(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s} + \rho\right)^{-1} \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s} \alpha \qquad (17)$$

and, thus

$$\delta_i^l(s) = \|\Psi_{\text{SVM}}\|_F^2 - \alpha' \left(\Phi_{\mathbf{x},\mathbf{u}}^{l,i,s}\right)' \left(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s} + \rho\right)^{-1} \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s} \alpha. \qquad (18)$$

With the matrix notation, the double-cascade structure becomes now more visible: beside the residual cascade with respect to l in the approximation of each $\mathbf{z}_i$ by $\mathbf{u}_i^l$, there is for each l a matrix cascade structure with respect to i that allows to store the entries up to the ith row in $\Phi_{\mathbf{x},\mathbf{u}}^{l,i,s}$ and up to ith row and ith column in $\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s}$. The remaining entries $(\Phi_{\mathbf{x},\mathbf{u}}^{l,i,s})_{n,m}$ for $m > i$ and $(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s})_{n,m}$ for $n, m > i$ can be taken from the previous level $l-1$.

We summarize our findings and design the algorithm for the learning stage of the W-RVM.

Learning Stage of the W-RVM:

$$\begin{array}{rl} \text{Input}: & \text{SVM with } \alpha_i, \mathbf{x}_i, i = 1, \ldots, N_x \\ & \text{RVM with} \beta_i, \mathbf{z}_i, i = 1, \ldots, N_z \end{array}$$

$$\begin{array}{rl} \text{Output}: & \text{W}-\text{RVM with the rectangle} \\ & \text{structures of } \mathbf{r}_i^l \text{ and the coefficient} \\ & \text{vectors } \hat{\gamma}^{l,i}, i = 1, \ldots, N_z(l), \\ & l = 0, \ldots, L \end{array}$$

1) Set $\Psi_{\text{SVM}} = \sum_{i=1}^{N_x} \alpha_i \Phi(\mathbf{x}_i)$, $\mathbf{u}_i^{-1} = 0$ and set $l = 0$.

2) Start with $i = 1$.

3) Compute for $s \in \{1, \ldots, 2^J\}^2, J = \log_2(2^M/4))$

$$\mathbf{u}_i^{l-1} = \sum_{j=0}^{l-1} \mathbf{r}_i^{j,s^*}$$
$$\mathbf{r}_i^{l,s} = (W^s)^{-1} S_\mu \left(W^s \left(\mathbf{z}_i - \mathbf{u}_i^{l-1} \right) \right)$$

where s^* denotes the best shift, S_μ is the shrinkage function (4) with the threshold parameter μ discussed in Section II-C1 and W is the wavelet transform operator.

4) Compute $\forall s \in \{1, \ldots, 2^J\}^2$ the decrement of the discrepancy (18)

if $i = 1, l = 0$: ${}_\delta\Delta_i^l(s) = \|\Psi_{\text{SVM}}\|_F^2 - \delta_1^0(s)$
if $i = 1, l > 0$: ${}_\delta\Delta_i^l(s) = \delta_{N_z}^{l-1}(s^*) - \delta_1^l(s)$
else: ${}_\delta\Delta_i^l(s) = \delta_{i-1}^l(s^*) - \delta_i^l(s)$

and the number of operations

$${}_\omega\Delta_i^l(s) = 4 * \# \left[\mathbf{r}_i^{l,s} \right] + v \left(\mathbf{r}_i^{l,s} \right)$$

where $\#[\mathbf{r}_i^{l,s}]$ is the number of piecewise constant rectangles and $v(\mathbf{r}_i^{l,s})$ the number of grey values of $\mathbf{r}_i^{l,s}$.

5) Select the best shift s^* out of $\{1, 2, \ldots, 2^J\}^2$ by

$$s^* = \arg\max_s \frac{{}_\delta\Delta_i^l(s)}{{}_\omega\Delta_i^l(s)}.$$

6) Save the rectangle structure of $\mathbf{r}_i^{l,s^*}$ and the coefficient vector

$$\hat{\gamma}^{l,i} = \gamma^{l,i}(s^*) = \left(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s^*} \right)^{-1} \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s^*} \alpha.$$

7) If $i < N_z(l)$, increment i and proceed to step 3. If $i = N_z(l)$ and $l < L$, increment l and proceed to step 2 ($N_z(l)$ and L are obtained using (21) and (22)); else, stop.

Finally, as a by-product of this section and as a contribution to Section II-B3, we are now able to quantify $\|\beta - \gamma\|$. Assume, the SVM is given by N_x support vectors $\mathbf{x}_i$ and the RVM by N_z reduced set vectors $\mathbf{z}_i$, then with $(\Phi_{\mathbf{z},\mathbf{z}})_{i,j} = \langle \Phi(\mathbf{z}_i), \Phi(\mathbf{z}_j) \rangle$ and $(\Phi_{\mathbf{x},\mathbf{z}})_{i,j} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{z}_j) \rangle$ it is common that $\beta = \Phi_{\mathbf{z},\mathbf{z}}^{-1} \Phi_{\mathbf{z},\mathbf{x}} \alpha$, see [3]. Consequently

$$\|\beta - \hat{\gamma}^{l,i}\| \leq \left\| \Phi_{\mathbf{z},\mathbf{z}}^{-1} \Phi_{\mathbf{x},\mathbf{z}} - \left(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s^*} \right)^{-1} \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s^*} \right\| \|\alpha\| \quad (19)$$

and since we have $\|\mathbf{u}_i^l - \mathbf{z}_i\| \leq C_{\mu,l}$, by perturbation arguments we also have an entry-wise perturbation estimate for the full matrices which in turn yield an estimate for $\|\beta - \hat{\gamma}^{l,i}\|$ in dependence on μ and l (we omit a detailed examination here). Moreover, as the approximations $\mathbf{u}_i^l$ at resolution level l tend to $\mathbf{z}_i$ as μ tends to 0, we have an entry-wise convergence

$$\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s^*} \to \Phi_{\mathbf{z},\mathbf{z}}, \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s^*} \to \Phi_{\mathbf{x},\mathbf{z}}$$

and, hence

$$\left\| \Phi_{\mathbf{z},\mathbf{z}}^{-1} \Phi_{\mathbf{x},\mathbf{z}} - \left(\Phi_{\mathbf{u},\mathbf{u}}^{l,i,s^*} \right)^{-1} \Phi_{\mathbf{x},\mathbf{u}}^{l,i,s^*} \right\| \xrightarrow{\mu \to 0} 0.$$

C. Detection Process

The classification function of the W-RVM, denoted by $y_i^l(\mathbf{x})$ of the input patch $\mathbf{x}$, using $N_z(l)$ W-RSVs at the levels $0, \ldots, l-1$ and i W-RSVs at the level l is as follows:

$$\begin{aligned} y_i^l(\mathbf{x}) &= \operatorname{sgn} \left(\sum_{k=1}^{i} \hat{\gamma}_k^{l,i} k\left(\mathbf{x}, \mathbf{u}_k^l\right) \right. \\ &\qquad \left. + \sum_{k=i+1}^{N_z} \hat{\gamma}_k^{l,i} k\left(\mathbf{x}, \mathbf{u}_k^{l-1}\right) + b_i^l \right) \\ k\left(\mathbf{x}, \mathbf{u}_i^l\right) &= \exp\left(-\frac{1}{2\sigma^2} \left(\mathbf{x}'\mathbf{x} - 2\mathbf{x}'\mathbf{u}_i^l + \mathbf{u}_i'^l \mathbf{u}_i^l \right) \right) \end{aligned} \quad (20)$$

where the kernel k is efficiently evaluated using integral images (Section II-A). For the term $2\mathbf{x}'\mathbf{u}_i^l = 2\mathbf{x}'\mathbf{u}_i^{l-1} + 2\mathbf{x}'\mathbf{r}_i^{l,s^*}$ only $2\mathbf{x}'\mathbf{r}_i^{l,s^*}$ has to be computed, since $2\mathbf{x}'\mathbf{u}_i^{l-1}$ can be stored at the previous level. The thresholds b_i^l are obtained automatically from an R.O.C. (Receiver Operating Characteristic) for a given accuracy. These thresholds are set to yield a given false rejection rate (FRR) so that the accuracy of the W-RVM can be the same as the one of the full SVM (see [3] for details). The given false rejection rate also controls the tradeoff between computational cost and detection performance and depends on the requirements of the application. If only few false rejections are acceptable (yields higher computational cost and more false acceptances), a smaller FRR should be adjusted. This ratio between FRR and FAR (false acceptance rate) is the only parameter of our algorithm to be set by the user. This ensures a simple to adjust detection approach.

Realizing our double cascade algorithm (Section II-B4) the detection process goes as follows.

Working Stage of the W-RVM:

1) Start at the first resolution level $l = 0$.

2) Start with the first W-RSV, $\mathbf{u}_1^l$ at the level l.

3) Evaluate $y_i^l(\mathbf{x})$ for the input patch $\mathbf{x}$ using (20).

4) If $y_i^l < 0$ then the patch is classified as not being the object of interest, the evaluation stops.

5) If $i < N_z(l)$, i is incremented and the algorithm proceeds to step 3; else if $l < L, l$ is incremented and the algorithm proceeds to step 2; otherwise the full SVM is used to classify the patch.

1) Adjustment of Resolution Levels and Number of W-RSVs Per Level: When computing an approximation of an SVM, it is not clear how many approximation vectors N_z should be computed (see [3]). This number of vectors may vary depending on the level l of the approximation. To this end, it may be useful to let N_z depend on l. The reason is that at a certain point of the evaluation algorithm it is more efficient to increment l (and reset i), rather than to increment i. The best value of $N_z(l)$ is computed in an offline process using a validation dataset: $N_z(l)$ is set to the smallest i for which empirically

$$\frac{\mathrm{Nops}\left(y_{i+1}^l\right)}{\mathrm{Nrecs}\left(y_{i+1}^l\right)} > \frac{\mathrm{Nops}\left(y_1^{l+1}\right)}{\mathrm{Nrecs}\left(y_1^{l+1}\right)} \tag{21}$$

where Nops stands for the number of operations and Nrecs stands for the number of rejections of the negative examples.

By a similar evaluation the last used resolution level L can be achieved. The optimal L is the smallest l that fulfills

$$\frac{\mathrm{Nops}\left(y_1^{l+1}\right)}{\mathrm{Nrecs}\left(y_1^{l+1}\right)} > \frac{\mathrm{Nops}(y)}{\mathrm{Nrecs}(y)} \tag{22}$$

where y denotes the decision function of the full SVM (1). For this L, it is more efficient to classify the last few remaining patches by the SVM, instead of incrementing l. L depends also on the threshold parameter μ. The smaller μ, the closer is $\mathbf{u}_i^l$ to $\mathbf{z}_i$ and the fewer resolution levels are required. However, the number of levels does not play a decisive role as the higher L, the sooner the evaluation process selects the next level, i.e., the less $N_z(l)$. Therefore, our proposed approach is not very sensitive to the parameter for setting the approximation accuracy [e.g., for μ in (4) a constant $\mu = 0.8\max(\mathrm{abs}(z_\lambda))$ can be used]. Opposite to former methods, using only one resolution level, the approach is simple and not sensitive to the parameter choice. The evaluation selects the most efficient approximation accuracy automatically at detection time.

III. Experimental Results

We applied our novel wavelet approximated reduced SVM to the task of face detection. For the training and validation of the classifier, we used two databases. The first set was crawled from the WWW (see Acknowledgment) and as second face database we used the greyscale version of FERET [11]. We chose this well-known dataset to provide the comparability to other approaches.

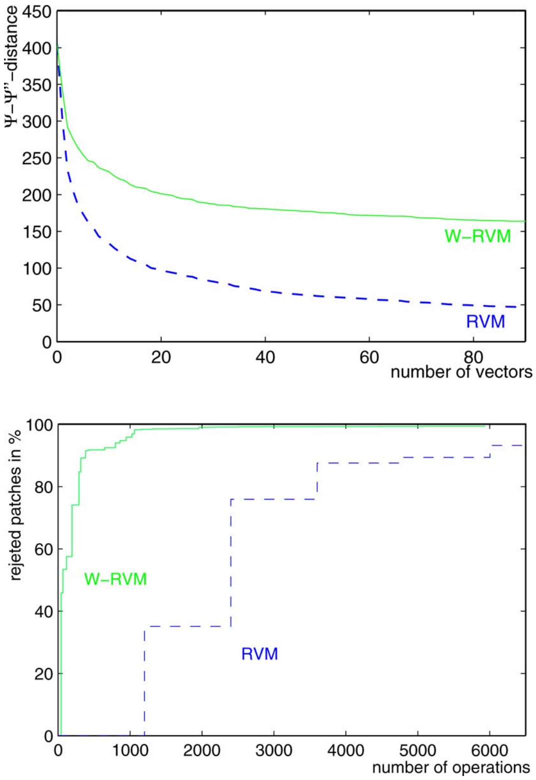

Fig. 3. *Top*: $\Psi_{\mathrm{SVM}} - \Psi_{\mathrm{W-RVM}}$ distance as function of the number of vectors for the RVM (*dashed line*), and the W-RVM (*solid line*). *Bottom*: Percentage of rejected nonface patches as a function of the number of operations required.

The training set includes 3500, 20×20, face patches, and 20000 nonface patches from the first dataset. The SVM computed on the training set yielded about 8000 support vectors that we approximated by $N_z = 90$ W-RSVs at $L = 5$ resolution levels by the method detailed in the previous section [e.g., L using (22)]. For the OCWT [Section II-B2], we used the classical mirroring for adequately continuing the image beyond the boundaries.

As first validation set (set I) we used 1000 face patches, and 100 000 nonface patches randomly chosen also from WWW images, but disjoint from the training examples. The first graph on Fig. 3 plots the residual distance of the RVM (dashed line) and of the W-RVM (plain line) to the SVM (in terms of the distance $\Psi_{\mathrm{SVM}} - \Psi_{\mathrm{RVM}}$ and $\Psi_{\mathrm{SVM}} - \Psi_{\mathrm{W-RVM}}$) as a function of the number of vectors used. It can be seen that for a given accuracy more wavelet approximated set vectors are needed to approximate the SVM than for the RVM. However, as shown on the second plot, for a given computational cost, the W-RVM rejects much more nonface patches from the validation set I than the RVM. This explains the improved runtime performances of the W-RVM. Additionally, it can be seen that the curve is smoother for the W-RVM; hence, a better tradeoff between accuracy and speed can be obtained by the W-RVM.

Fig. 4 shows the R.O.C.s, computed on the validation set I, for the SVM, the RVM and the W-RVM. It can be seen that the accuracies of the three classifiers are similar without (top

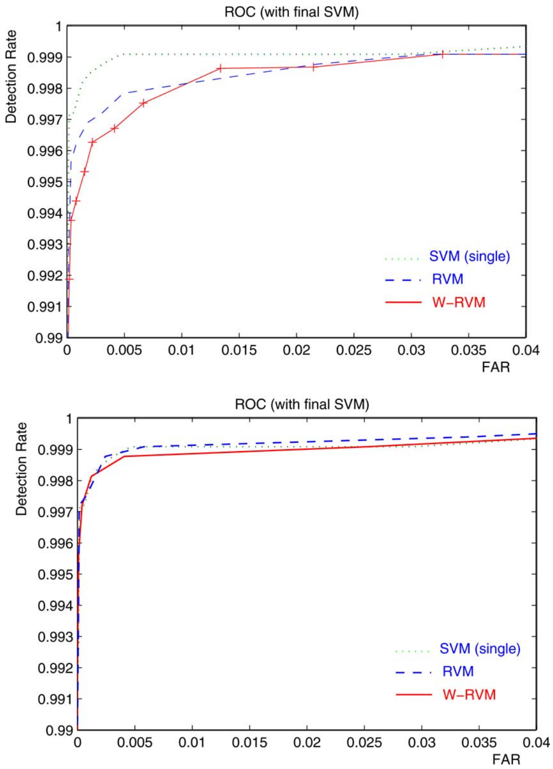

Fig. 4. R.O.C.s for the SVM, the RVM, and the W-RVM (*top*) without and (*bottom*) with the final SVM classification for the remaining patches. The FAR is related to the number of nonfaces.

TABLE I
COMPARISON OF ACCURACY AND SPEED IMPROVEMENT OF THE W-RVM TO THE RVM AND SVM

method	FRR	FAR	time per patch
SVM	1.4%	0.002%	787.34μs
RVM	1.5%	0.001%	22.51μs
W-RVM	1.4%	0.002%	1.48μs

plot) and almost equal with the final SVM classification for the remaining patches (bottom plot), see step 5. of the evaluation algorithm.

Table I compares the accuracy and the average time required to evaluate the patches of the validation set I. The speedup over the former approach [10] is about a factor 2.5 (3.85 μs). The novel W-RVM algorithms provides a significant speedup (530-fold over the SVM and more than 15-fold over the RVM), for no substantial loss of accuracy.

The validation set II contains 500 frontal and half profile images from the FERET database [11]. We compared our approach with the Viola and Jones method [4] implemented in OpenCV (version b5a). The Viola and Jones detector yields on set II a detection rate of 90.9% by 0.32 false acceptances (FA) and 0.29 s per image (on a Pentium M Centrino 1600 CPU). Compared to the results given in [4] the processing time is slower since the image size of the FERET images is larger. The results on FERET are more accurate because of the higher quality of the images. With the W-RVM we obtained on the same PC and set II a detection rate of 90.1% by 0.25 FA and 0.15 s processing time per image.

Our proposed classifier is more efficient at detection, but mainly at training time than the AdaBoost method [4] and classifies about 25 times faster than the Rowley–Baluja–Kanade detector [1] and about 1000 times faster than the Schneiderman–Kanade detector [21].

To demonstrate the efficient and accurate detection algorithm, we implemented an application using a standard webcam. Accurate face detection one obtained at real-time by 25 fps (on a Intel Pentium M Centrino 1600 CPU, at a resolution of 320 × 240, step size 1 pixel, five scales).

IV. CONCLUSION

In this paper, we presented a novel efficient method for SVM classifications on image-based vectors. The essential ingredient was an recursively applied optimally matched wavelet transform of the reduced set vectors. It was demonstrated on the task of face detection.

As opposed to the RVM, the sparseness of operations required for classification is not only controlled by the number of reduced set vectors but also by the number of wavelets basis functions used to approximate a reduced set vector. Hence, negative examples can be rejected with much less number of operations, making the runtime of the algorithm very efficient. Moreover, as the Haar wavelets are used, the SVM kernel may be evaluated extremely efficient using integral images.

The main advantage of this algorithm compared to other classifiers is that the learning stage of our proposed wavelet approximated reduced SVM is fast, straightforward, automatic, and does not require the manual selection of ad-hoc parameters and is, therefore, simple. The approach is straightforward because of our paradigm to avoid a complex search over the parameter space, by starting with the proved optimal performance of an SVM. Then we reduce the complexity by a reduced vector set and the over-complete wavelet approximation. The W-RVM is simple to re-implement. In Section II-B5, we propose a detailed pseudo code. The only input is the SVM and RVM. The used matrix notation makes the double-cascaded structure visible, supports vectorized code and reduces the update rule. This speeds up the training significantly. The parameter are adjusted automatically by the algorithm, e.g., for the number of resolution levels and the number of approximated vectors per level (Section II-C1). Also, the thresholds $\mathbf{b}$ in (20) are obtained automatically. These thresholds are set to yield a given false rejection rate (FRR). The tradeoff between FRR and FAR is the only parameter of our algorithm to be set by the user because it depends on the requirements of the application (Section II-C). All other parameters are automatically adjusted. The learning stage is fast, because the training of the W-RVM takes about 2 h instead of weeks.

ACKNOWLEDGMENT

The authors would like to thank H. Burkhardt's group at the University of Freiburg and, in particular, R. Jüngling for providing the face images used in the experiments. They would also like to thank the referees for thorough reviews, valuable comments, and suggestions that significantly improved this work, as well as J. Sherrah for his insights on the Viola and Jones detector.

REFERENCES

[1] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *Pattern Anal. Mach. Intell.*, vol. 20, pp. 23–38, 1998.

[2] D. Keren, M. Osadchy, and C. Gotsman, "Antifaces: A novel, fast method for image detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 7, pp. 747–761, Jul. 2001.

[3] S. Romdhani, P. Torr, B. Schölkopf, and A. Blake, "Computationally efficient face detection," presented at the 8th Int. Conf. Computer Vision, Jul. 2001.

[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," presented at the IEEE Conf. Computer Vision and Pattern Recognition, 2001.

[5] V. Vapnik*, Statistical Learning Theory*. New York: Wiley, 1998.

[6] W. Kienzle, G. H. Bakir, M. O. Franz, and B. Schölkopf, "Efficient approximations for support vector machines in object detection," in *Proc. DAGM*, 2005, pp. 54–61.

[7] D. Karras, "Improved defect detection in textile visual inspection using wavelet analysis and support vector machines," *ICGST Int. J. Graph., Vis., Image Process.* 2005.

[8] C. Garcia, G. Zikos, and G. Tziritas, "Face detection in color images using wavelet packet analysis," presented at the IEEE Int. Conf. Multimedia Computing and Systems, 1999.

[9] M. Rätsch, S. Romdhani, G. Teschke, and T. Vetter, "Over-complete wavelet approximation of a support vector machine for efficient classification," presented at the 27th Pattern Recognition Symp., Vienna, Austria, 2005.

[10] M. Rätsch, S. Romdhani, and T. Vetter, "Efficient face detection by a cascaded support vector machine using haar-like features," in *Proc. 26th Pattern Recognition Symp.*, 2004, pp. 62–70.

[11] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi, "The feret evaluation methodology for face recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[12] C. Burges, "Simplified support vector decision rules," in *Proc. 13th Int. Conf. Machine Learning*, 1996, pp. 71–77.

[13] T. Thies and F. Weber, "Optimal reduced-set vectors for support vector machines with a quadratic kernel," *Neural Comput.*, vol. 16, no. 9, pp. 1769–1777, Sep. 2004.

[14] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola, "Input space vs. feature space in kernel-based methods," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1000–1017, May 1999.

[15] F. Crow, "Summed-area tables for texture mapping," in *Proc. SIGGRAPH*, 1984, vol. 18, no. 3, pp. 207–212.

[16] H. Triebel*, Interpolation Theory, Function Spaces, Differential Operators*. Berlin, Germany: Verlag der Wissenschaften, 1978.

[17] H.-J. Schmeisser and H. Triebel*, Topics in Fourier Analysis and Function Spaces*. New York: Wiley, 1987.

[18] A. Cohen, R. DeVore, P. Petrushev, and H. Xu, "Nonlinear approximation and the space $BV(\mathbb{R}^2)$," *Amer. J. Math.*, vol. 121, pp. 587–628, 1999.

[19] I. Daubechies and G. Teschke, "Variational image restoration by means of wavelets: simultaneous decomposition, deblurring and denoising," *Appl. Comput. Harmon. Anal.*, vol. 19, no. 1, pp. 1–16, 2005.

[20] R. Coifman and D. Donoho, "Translation-invariant de-noising," in *Wavelets and Statistics*, A. Antoniadis and G. Oppenheim, Eds. New York: Springer-Verlag, 1995, pp. 125–150.

[21] H. Schneiderman and T. Kanade, "A statistical method for 3d object detection applied to faces and cars," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, vol. 1, pp. 746–751.

Matthias Rätsch studied mathematics, physics, and computer science at the University of Potsdam, Germany. In 2002, he received the Diploma degree from the University of Potsdam for his thesis on segmenting touching characters (Patent No: 195 33 585). He is currently pursuing the Ph.D. degree in the graphics and vision research group (GraVis) at the University of Basel, Switzerland.

He worked in the fields of optical and intelligent character recognition at the University of Bremen, Germany. His research interests are in the fields of face and facial feature detection, machine learning, and human computer interaction.

Gerd Teschke received the Diploma degree in mathematics from the University of Potsdam, Germany, in 1998, and the Dr. rer. nat. and Habilitation degrees from the University of Bremen, Germany, in 2001 and 2006, respectively.

From 2002 to 2003, he was with the PACM at Princeton University, Princeton, NJ. In 2002, he became an Assistant Professor at the University of Bremen. In 2005, he moved to Berlin, Germany, and became the Head of the research group Inverse Problems in Science and Technology at the Konrad Zuse Institute. His current research centers around inverse problems, frame theory, and sparse signal recovery.

Prof. Teschke received the Bremer Studienpreis (Bruker prize).

Sami Romdhani studied electronics engineering at the Universite Libre de Bruxelles, Belgium, and received the M.Sc. degree in electronics engineering from the University of Glasgow, Glasgow, U.K., and the Ph.D. degree from the University of Basel, Switzerland, in 2005.

In 1998, he started his research on face image analysis at the University of Westminster, U.K., and joined the University of Freiburg, Germany, in 2000, where he started working on 3-D morphable models. He is currently a Postdoctoral Researcher at the University of Basel, where he works mainly on extending computer vision methods using accurate probabilistic prior models. His research interests include image modeling and understanding, inverse rendering, and 3-D modeling and reconstruction.

Thomas Vetter (M'03) studied mathematics and physics and received the Ph.D. degree in biophysics from the University of Ulm, Germany.

As a Postdoctoral Researcher at the Center for Biological and Computational Learning, Massachusetts Institute of Technology, Cambridge, he started his research on computer vision. In 1993, he moved to the Max-Planck-Institute, Tübingen, Germany, and, in 1999, he became a Professor of computer graphics at the University of Freiburg, Germany. Since 2002, he has been a Professor of applied computer science at the University of Basel, Switzerland. His current research is on image understanding, graphics, and automated model building.