Wavelet Reduced Support Vector Regression for Efficient and Robust Head Pose Estimation

Matthias Rätsch *(Member IEEE)*,
Philip Quick *(Member IEEE)*
Cognitec Systems GmbH
Grossenhainer Str. 101, D-01127 Dresden, Germany
Email: {raetsch,quick}@cognitec.com

Patrik Huber, Tatjana Frank, Thomas Vetter *(Member IEEE)*
Departement of Computer Science and Mathematics
University of Basel
Bernoullistrasse 16, CH-4056 Basel, Switzerland
Email: {patrik.huber,tatjana.frank,thomas.vetter}@unibas.ch

***Abstract*—In this paper, we introduce concepts to reduce the computational complexity of regression, which are successfully used for Support Vector Machines. To the best of our knowledge, we are the first to publish the use of a cascaded Reduced Set Vector approach for regression. The Wavelet-Approximated Reduced Vector Machine classifiers for face and facial feature point detection are extended to regression for efficient and robust head pose estimation. We use synthetic data, generated by the 3D Morphable Model, for optimal training sets and demonstrate results superior to state-of-the-art techniques. The new Wavelet Reduced Vector Regression shows similarly good results on natural data, gaining a reduction of the complexity by a factor of up to 560. The introduced Evolutionary Regression Tree uses coarse-to-fine loops of strongly reduced regression and classification up to most accurate complex machines. We demonstrate the Cascaded Condensation Tracking for head pose estimation for a large pose range up to $\pm$90 degrees on videostreams.**

***Keywords*-Wavelet Reduced Vector Regression; Reduced Set Vector Machine; Head Pose Estimation; Cascaded Condensation Tracking; Coarse-to-Fine Particle Filter; Evolutionary Regression Tree; Wavelet Vector Machine**

I. Introduction

Humans are able to immediately predict the position, orientation, or expressions of faces. Human Computer Interaction (HCI) should be as natural as a conversation between humans. Embodied Conversational Agents must be able to localize their conversational partner before they initiate contact. To detect and track the location and orientation of objects is an important aspect of robotics and computer vision. Object detection is a binary pattern-classification problem, in contrast to pose estimation, where parameters are estimated, e.g. angles of a head. Ultimately, when given an image or videostream, the aim is to estimate location, scale, yaw, pitch and roll angles of a face.

Several approaches have been published in the field of head pose estimation. Often nonlinear regression techniques are used, e.g. by Ma et al. [1] and others [2], [3], [4]. Chutorian et al. give a good survey [5] over techniques used, data and results. One of the most accurate pose estimation approaches, by Balasubramanian et al., is based on Biased Manifold Embedding [6]. These results are based on the database *FacePix* containing only 30 different subjects. Overfitting and non-uniformly distributed data are two of the core problems of regression [7]. They can be avoided by using synthetically rendered images with infinitely many possible subjects, generated in this work with the 3D Morphable Model of the working group of T. Vetter [8], [9].

Face detection is complex as faces differ in size, rotation, orientation, illumination, and subjects. Furthermore, glasses often occlude parts of the characteristic eyes, and specular highlights occur. In Rätsch et al. efficient classifiers are proposed which can be adjusted to specific complexity [10], [11]. The efficiency is obtained by a reduced set of wavelet approximated support vectors used in a coarse-to-fine Double Cascade. This Wavelet-Approximated Reduced Vector Machine, short Wavelet Vector Machine (WVM), is combined with an extended condensation tracking [12]. Condensation samples more measurement points on regions of interest instead of using a sliding window approach or an equidistant grid for sampling. Our former proposed Cascaded Condensation Tracking (CCT) [13] spends only as much effort as is necessary for easy to discriminate regions of the feature space, and most of the effort on regions with high statistical likelihood of containing the object of interest.

CCT tracks faces robust up to approximately half profile view. The range from left to right profile view is too complex for a tracking of the full face space in real-time. To solve such complex problems efficiently, a strategy of divide and conquer (D&C) is often used. The feature space is divided in ranges where specific classifiers are trained. For example, the pose angles can be divided in subregions. One way, which is very time consuming, is to use the specific classifiers in sequence. Sahbi et al. [14] introduced a realization of divide and conquer using a tree representation for pose-invariant face detection for the roll angle. At every node, a full complex Support Vector Machine [15] is used. This approach is too slow for a pose-invariant face detection of more than one orientation or for tracking in real-time.

In this paper we propose Support Vector Regression (SVR) to estimate the orientation of a human head and use this information for the decision which classifier (trained for

978-0-7695-4683-4/12 $26.00 © 2012 IEEE
DOI 10.1109/CRV.2012.41

a subrange of a pose angle) will be used. Regression supplies no information whether there is a face located at a particular position of the image or not. First approaches that unify classification and regression are published [16], but they are not usable for an efficient coarse-to-fine approach running in real-time on videostreams, in contrast to the approach introduced here. Therefore, in this work we use a regression and a classification stage. A full SVR is too time-consuming to use for all image locations, and the face space for all poses is too complex to first decide with a classifier where the faces are located. Therefore, we will adapt the approaches to reduce the complexity for classification mentioned above to regression. Both stages, the classification and the regression, are adjustable in their complexity. In this work we introduce a Evolutionary Regression Tree as a coarse-to-fine loop of regression and classification stages.

The main contribution of this work is the adaption and extension of the coherent framework we developed for classification [10], [13], [11] to regression (Section II-B). The use of the cascaded reduced vector technique for regression is one of the main novelties (Section II-B2 and II-B3). Also a new approach for Cascaded Regression is introduced (Section II-B5) and the adjustable complexity enables the introduced coarse-to-fine Evolutionary Regression Tree (Section II-C). We demonstrate very accurate pose estimation results (Section III), superior to state-of-the-art methods. Optimally distributed training sets are generated with the 3D Morphable Model (Section II-D). In our application of the Evolutionary Regression Tree we demonstrate an efficient pose-invariant tracking with highly accurate pose estimation of faces on videostreams (Section III-C).

II. Wavelet Reduced Vector Regression

Nonlinear support vector regression [17], like classification, is solved with a kernel function, which leads to high accuracy and optimal generalization performance, but at the same time a high computational effort.

If Support Vector Regression is to be applied to a real-time camera stream, or for estimating several facial features at a time, it is not practical. Lee et al. proposes a ε-smooth SVR [18] formulation, where they only need to solve a system of linear equations iteratively instead of solving a convex quadratic program or a linear program, as is the case with a conventional ε-SVR. Second, they propose a reduction of the kernel, similarly to classification. Those reduced vectors in the kernel are however a subset of the training data.

In previous work, Romdhani et al. [11] could adapt the idea of Burges [19], to replace the Support Set Vectors (SSVs) with a lower number of approximated Reduced Set Vectors (RSVs), successfully to face detection. The RSVs are, in comparison to the SSVs, no longer a subset of the training data, but new data-points. Kukenys et al. [20] follow a hybrid approach by alternately combining the global optimization of [19] with the cascade of [11]. Rätsch et al. [10], [13] could further accelerate the evaluation function significantly by reducing the operations per support vector based on Integral Images, using a Double Cascade, and an Over-Complete Wavelet Transformation (OCWT) to approximate the RSVs by Wavelet Set Vectors (WSVs).

To our knowledge this is the first publication adapting reduced SVM techniques, like the Reduced Set Vector approach that is included in our WVM framework, to regression. Literature research yielded only a technical report by Marconato et al. [21], which is limited to linear kernels.

Now we will introduce step by step the adaption of the reduction techniques of the WVM from Support Vector Machines to Support Vector Regression.

A. Comparison SVM and SVR

Because the concepts of the WVM have been developed for classification, it is reasonable to start by comparing the two methods. While support vector classification and regression are similar methods, some important differences exist. In classification, a binary decision function that separates training data of two classes is calculated. In regression, a continuous function is calculated that fits the given data best. In the following, we will denote all variables concerning regression with a tilde.

In regression, our training set consists of data in the form $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\} \subset \mathbb{R}^d \times \mathbb{R}$. In dual form, support vector regression involves maximizing the constrained optimization problem

$$\begin{aligned}\tilde{L}(\tilde{\alpha}, \tilde{\alpha}^*) \quad = \quad & -\frac{1}{2}\sum_{i,j=1}^{N}\left(\tilde{\alpha}_i - \tilde{\alpha}_i^*\right)\left(\tilde{\alpha}_j - \tilde{\alpha}_j^*\right)\langle \mathbf{x}_i, \mathbf{x}_j\rangle \\ & -\epsilon\sum_{i=1}^{N}\left(\tilde{\alpha}_i + \tilde{\alpha}_i^*\right) + \sum_{i=1}^{N}\tilde{y}_i\left(\tilde{\alpha}_i - \tilde{\alpha}_i^*\right) \quad (1)\end{aligned}$$

with respect to $\{\tilde{\alpha}_n\}$ and $\{\tilde{\alpha}_n^*\}$[17]. Solving for $\tilde{\alpha}_i$, $\tilde{\alpha}_i^*$, $\mathbf{w}$ and b, predictions for a new input image $\mathbf{x} \in \mathbb{R}^d$ can be made using $\tilde{f}(\mathbf{x}) = \sum_{i=1}^{N_x}\left(\tilde{\alpha}_i - \tilde{\alpha}_i^*\right)k(\mathbf{x}, \tilde{\mathbf{x}}_i) + b$, 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 $\Phi : \mathcal{X} \to F, \quad \mathbf{x} \mapsto \Phi(\mathbf{x})$ maps the data $\mathbf{x}$ (in our case, a vector of 1024 gray values of a 32×32 observation window) into F. The SVR decision hyperplane is determined by $\Psi_{\text{SVR}} = \sum_{i=1}^{N_{\tilde{x}}}\left(\tilde{\alpha}_i - \tilde{\alpha}_i^*\right)\Phi(\tilde{\mathbf{x}}_i)$, with N_x support vectors $\tilde{\mathbf{x}}_i$ with coefficients $\tilde{\alpha}_i$ and $\tilde{\alpha}_i^*$.

As the minimization problem and its evaluation function are similar to SVM for classification, most of the WVM-theory can be adapted for regression. In the following, we will give a short overview of our WVM framework for classification. We will then show which of the components can be used for regression and which need modifications.

B. Wavelet Reduced Vector Regression

1) Wavelet Vector Machine Core Ideas: The WVM-framework consists of the four SVM reduction concepts:

i. **Reduced set of vectors:** Approximation of the support vectors with a much smaller set of vectors [11].
ii. **Integral Images:** Integral Image method for the efficient calculation of the kernel.
iii. **Wavelet Frame:** An over-complete wavelet system to find the best representation of the WSVs.
iv. **Double Cascade:** Early rejection of non-objects at the evaluation over the Wavelet Set Vectors (WSVs):
   - **Cascade over the number of used WSVs**
   - **Cascade over the resolution levels of each WSV**

2) Reduced Vector Regression - Gradient Descent: Burges [19] proposes an approximation to the decision rule of the SVM in terms of a reduced set of vectors (RSVs), which are not a subset of the training data. Depending on the used kernel, they can be approximated analytically. We will adapt this idea to regression to reduce time necessary for computing. Because we found the RBF-kernel to fit regression data well in previous experiments, we will first show the reduction with RBF-kernels.

The idea how to reduce SVMs can be adapted to regression, because in both cases a mathematically similar function is approximated. The decision hyperplane of the SVR, Ψ_{SVR}, can be approximated with $\Psi_{\text{WVR-R}}$ by replacing the support vectors with a new set of Reduced Regression Vectors (RRVs) $\tilde{\mathbf{z}}_i$: $\Psi_{\text{WVR-R}} = \sum_{i=1}^{N_z} \tilde{\beta}_i \Phi(\tilde{\mathbf{z}}_i)$, where $N_z \ll N_x, \tilde{\beta}_i \in \mathbb{R}$.

The distance between Ψ_{SVR} and $\Psi_{\text{WVR-R}}$ is minimized:

$$\left\|\Psi_{\text{SVR}} - \Psi_{\text{WVR-R}}\right\|^2 = \sum_{i,j=1}^{N_{\tilde{x}}} \left(\tilde{\alpha}_i - \tilde{\alpha}_i^*\right)\left(\tilde{\alpha}_j - \tilde{\alpha}_j^*\right) k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) + \sum_{i,j=1}^{N_z} \tilde{\beta}_i \tilde{\beta}_j k(\tilde{\mathbf{z}}_i, \tilde{\mathbf{z}}_j) - 2\sum_{i=1}^{N_x}\sum_{j=1}^{N_z} \left(\tilde{\alpha}_i - \tilde{\alpha}_i^*\right) \tilde{\beta}_j k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}_j) \quad (2)$$

The Reduced Regression Vectors $\tilde{z}_i$ and the coefficients $\tilde{\beta}_i$ are calculated iteratively as in [22]. This reduction can be applied to regression, yielding for the n^{th} RRV

$$\tilde{\mathbf{z}}_{n+1} = \frac{\sum_{i=1}^{N_{\tilde{x}}} \left(\tilde{\alpha}_i - \tilde{\alpha}_i^*\right) \exp(-||\tilde{\mathbf{x}}_i - \tilde{\mathbf{z}}_n||^2/(2\sigma^2))\tilde{\mathbf{x}}_i}{\sum_{i=1}^{N_{\tilde{x}}} \left(\tilde{\alpha}_i - \tilde{\alpha}_i^*\right) \exp(-||\tilde{\mathbf{x}}_i - \tilde{\mathbf{z}}_n||^2/(2\sigma^2))} \quad (3)$$

and $\tilde{\beta} = \left(K^{\tilde{z}}\right)^{-1}\left(K^{\tilde{z}\tilde{x}}\right)(\tilde{\alpha} - \tilde{\alpha}^*)$, with $K_{ij}^{\tilde{z}} := (\Phi(\tilde{\mathbf{z}}_i) \cdot \Phi(\tilde{\mathbf{z}}_j))$ and $K_{ij}^{\tilde{z}\tilde{x}}\left(\Phi(\tilde{\mathbf{z}}_i) \cdot \Phi(\tilde{\mathbf{x}}_j)\right)$. The new regression function for the WVR-R is $\tilde{f}(\mathbf{x}) = \sum_{i=1}^{N_z} \tilde{\beta}_i k(\mathbf{x}, \tilde{\mathbf{z}}_i) + b$.

3) Reduced Vector Regression - Analytical for Inhom. Polynomial Kernel: Burges [19] provides an analytical solution for finding the best RSVs for homogeneous quadratic kernels. Thies and Weber [23] developed this further and provide an explicit solution in case of an inhomogeneous quadratic kernel. The key idea is to follow the approach of Burges by expressing the inhomogeneous kernel as a homogeneous kernel on a space having one dimension more than the original one. This reduction is more efficient than gradient descent, but not applicable for RBF-kernels.

4) Integral Images for Efficient Kernel Evaluation based on Wavelet Frame Methods: During evaluation of an SVR or RVR, most of the time is spent for kernel evaluations. In the case of the RBF kernel $k(\mathbf{x}, \tilde{\mathbf{z}}_i) = \exp(-||\mathbf{x} - \tilde{\mathbf{z}}_i||^2/(2\sigma^2))$, chosen here, the computational cost is spent in evaluating the norm of the difference between a patch and an RRV. The expanded norm is $\mathbf{x}'\mathbf{x} - 2\mathbf{x}'\tilde{\mathbf{z}}_i + \tilde{\mathbf{z}}_i'\tilde{\mathbf{z}}_i$. As $\tilde{\mathbf{z}}_i$ is independent of the input image, it can be precomputed. $\mathbf{x}'\mathbf{x}$ can be computed efficiently using the integral image [24] of the squared pixel values of the input image. Hence, the computational cost of the norm is determined by the term $2\mathbf{x}'\tilde{\mathbf{z}}_i$.

The Reduced Regression Vectors $\tilde{\mathbf{z}}_i$ can be approximated by optimal wavelet frame approximated reduced regression vectors (WRVs) $\tilde{\mathbf{u}}_i$, which have a block-like structure, as seen in Fig. 5. If $\tilde{\mathbf{u}}_i$ is an image patch with rectangles of constant gray levels, then the term $2\mathbf{x}'\tilde{\mathbf{u}}_i$ can be evaluated very efficiently using the integral image. The term can be resorted by $2\mathbf{x}'\tilde{\mathbf{u}}_i = 2\sum_{k=1}^{D} x_k \tilde{u}_{i,k} = 2\sum_{r=1}^{R_i} v_{i,r} \sum_{j=1}^{D_r} x_j$ where D is the dimension of the vectors (e.g., 1024 pixel with a patch-size of 32×32), R_i is the number of rectangles of $\tilde{\mathbf{u}}_i$, $v_{i,r}$ the gray values of the rectangle r and $x_j, j = 1, ..., D_r$ 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 [24], the dot product is evaluated in constant time.

The Integral Image method works for the RBF kernel, as well as the polynomial kernel. In case of a polynomial kernel, $\mathbf{x}'\mathbf{x}$ and $\tilde{\mathbf{u}}_i'\tilde{\mathbf{u}}_i$ vanish and the remaining $\mathbf{x}'\tilde{\mathbf{u}}_i$ can be calculated efficiently using integral images with the method described above.

5) Cascaded Regression: For the analytical solution from Section II-B3, the different sets of $\{\tilde{\beta}_j\}$, $j = 1...i$ for each WRV $\tilde{\mathbf{z}}_i$, necessary to run the evaluation function cascaded, are not available. Thus, the analytical WVR-R is not usable for cascaded regression without modifications.

We cannot directly adapt the cascade from the WVM classification framework, because there are no negative patches that could be rejected early. We introduce a new method for fast, efficient cascaded regression for RBF kernels.

For most of the image locations, a few WRVs at low resolution levels suffice to get a rough estimate of the angle. This is due to the fact that the WRVs $\tilde{\mathbf{u}}_i, i = 1, ..., N^{\tilde{u}}$ are ordered, each one with its own set of weights. As seen in Fig. 1, often, the estimation of the angle converges early and does not significantly improve further, so that an evaluation of all the available WRVs is unnecessary. To that end, we measure the gradient $\delta = \partial/\partial\tilde{\mathbf{u}}\ g(\tilde{f}(\tilde{\mathbf{u}}_i, \mathbf{x}))$ at $\tilde{\mathbf{u}}_i$, where g is obtained from the evaluation function $\tilde{f}$ by applying a moving average operator of size k, and the scattering $\eta =$

$\sum_{j=i-k}^{i}(g(\tilde{f}(\tilde{\mathbf{u}}_j,\mathbf{x}))-\tilde{f}(\tilde{\mathbf{u}}_j,\mathbf{x}))^2$ over the k last WRVs. The next $\tilde{\mathbf{u}}_{i+1}$ is only incorporated if δ and η are larger than given thresholds t_1 and t_2. The parameters of the Cascaded Regression algorithm k and $\mathbf{t}$ are optimized such that the average of the used number of WRVs over all patches is smaller than using a constant number of WRVs.

With this cascade of wavelet vectors, a WVR is very efficient compared to a WVR-R. The complexity is smoothly adjustable, and the WVR is optimally suited for the first stages of the Evolutionary Regression Tree introduced in Section II-C. For example, a cascaded WVR that uses 490 WRVs on average, performs with an average error of $8.26°$, while a WVR-R with the same complexity performs approximately 2.9 times worse ($23.50°$) on the same test data. Thus, the WVR is very well suited for a fast, computationally efficient approximation of the angle.

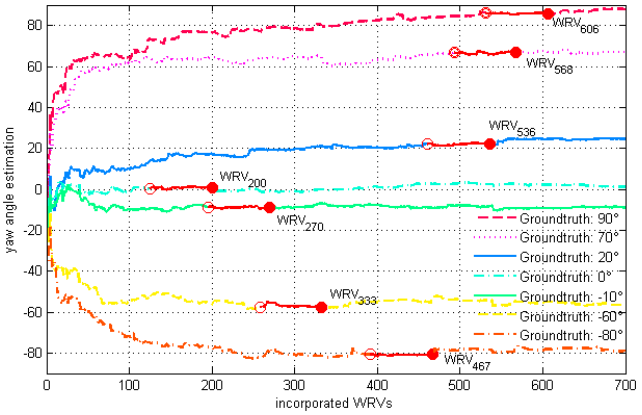

Figure 1. Cascaded Regression: Use only as many WRVs per patch as necessary to compute the angle with an adjustable accuracy. For many patches, the WVR converges early and does not significantly improve further (e.g. cyan dash-dotted curve; the red line shows the k last WRVs of the cascade). However, in some cases, convergence takes longer (e.g. red dashed curve). Thus, a cascade over the WRVs is used.

6) Evaluation of the Comprehensive Wavelet Reduced Vector Regression: Replacement of the RRVs $\tilde{\mathbf{z}}_i$ with the WRVs $\tilde{\mathbf{u}}_i$ leads to the new Haar-like hyperplane for regression $\Psi_{\text{WVR}}=\sum_{i=1}^{N_{\tilde{z}}}\tilde{\gamma}_i\Phi(\tilde{\mathbf{u}}_i)$, which is an optimal approximation of the original hyperplane. The $\{\tilde{\gamma}_i\}$ correspond to the $\{\tilde{\beta}_i\}$ of the WVR-R and are calculated similarly.

The WVR evaluation function for the comprehensive WVR (with Wavelet-Frame, Cascade and Integral Image) of resolution level l at the i^{th} reduced vector for an input image $\mathbf{x}$ becomes:

$$\tilde{f}_i^l(\mathbf{x})=\sum_{h=1}^{l-1}\sum_{j=1}^{N_{\tilde{z}}^h}\tilde{\gamma}_{h,j}^{l,i}k(\mathbf{x},\tilde{\mathbf{u}}_j^h)+\sum_{j=1}^{i}\tilde{\gamma}_{l,j}^{l,i}k(\mathbf{x},\tilde{\mathbf{u}}_j^l)+b. \quad (4)$$

C. Evolutionary Regression Tree

The goal of the proposed work is to track and to estimate the pose of objects in videostreams. The orientation of a head can be estimated by regression, but no certainty is obtained whether there is an object located or not. Therefore, we use a two stage approach, consisting of a regression and a classification step.

Detecting a specific object in an image is computationally expensive, as all the pixels of the image are potential object centers. The hypothesis space for faces is very large, because of the different appearances in expression, in pose and individual differences. If the feature space is too complex to obtain reasonable classification results with a single classifier by acceptable effort, often, a strategy of D&C is used. The feature space is divided, e.g. in ranges of different pose angles and specific classifiers are trained for each subspace. To use all these classifiers in a sequence one after the other is often too time consuming, especially if the space is divided in different dimensions, e.g. yaw angles, and each of these parts is divided again in different pitch angles, and so on.

The here proposed alternative is to estimate the pose angle by regression first and then to use the classifier specifically trained for the estimated pose range. Hence, we use two stages: One regression step to estimate the parameter for the divide and conquer strategy and one stage for the classification. The problem is in which order to use them. To first use the complex regression for all possible locations is too time-consuming. Also to first classify all locations for all poses is too complex for one classifier in reasonable time.

Similar to evolutionary programming we use a coarse-to-fine looped strategy, called Evolutionary Regression Tree (ERT). Starting with a weak but very efficient regression step we obtain a rough estimation of the parameter to decide which specific classifier will be used at the next stage. These weak classifiers reject first feature space locations (e.g. pixels of the image as potential object centers) for the specific areas. In the next loop of the evolutionary strategy a more complex regression can be used for the remaining locations, leading to more accurate estimations to decide which specific trained classifier will be used next in this loop. For the much smaller amount of remaining patches, stronger specific classifiers are used to reject objects of no interest much more precisely, and the next loop is started. The evolutionary loops can be repeated until the final remaining object locations are found and the estimation of the now full complex Support Vector Regression is used to obtain the best final estimation results.

In Section III-C we propose a first version of the introduced ERT based on coarse-to-fine Cascaded Regression stages introduced in Section II-B5 and also adjustable Double Cascaded WVM classifiers summarized in Section II-B1.

D. Synthetic Images for WVR

1) Disadvantages of Natural Data: SV Regression faces the problem that training with non-uniform distributed data introduces a bias in the estimation of new values. When particular angles are overrepresented in the training data, e.g. the yaw angles $0°$ or $30°$, as is often the case, the regression estimation will tend more towards these values [7]. This

makes the training process difficult, as natural data at bigger angles are difficult to obtain. For instance, in face databases like *MultiPie* and *CasPeal* (see Section III-A), such data are highly underrepresented. Also, in each database, only a subset of angles is available, so that it is difficult to train a regression machine for a continuous function from $-90°$ to $90°$ yaw angle. Another problem is the inter-database angle annotation variance - an example subject in *MultiPie* at $45°$ does not look like it was taken from the same angle as a subject from *CasPeal* at $45°$. This is due to different conditions in which the pictures are taken, e.g. single-shot versus camera rotating around the subject, etc.

For those reasons, we start with using synthetic data generated by the 3D Morphable Model (3DMM) [8], [9]. With the 3DMM, faces of any pose angle, illumination, shape and texture variation can be generated freely (see Fig. 2). For our application, this has the advantage that we can generate faces from a yaw angle of $-90°$ to $90°$, in a $1°$ interval, to train a WVR that can accurately handle the whole domain of angles. Additionally, we do not introduce a bias because particular angles are not underrepresented. Furthermore, we have a very accurate labeling.

Evidently, synthetic data have some disadvantages of their own. With the current 3DMM, it is not possible to model facial hair or glasses, so they clearly lack some variance that is occurring in natural data. This can be compensated by also including some natural data to the training set, while keeping the advantages of synthetic data.

III. Performance of Reduced Support Regression and Applications

In this section, first the datasets used are described, then results are shown for the SVR and WVR-R. The performance is compared to leading head pose estimation approaches, and finally, a first application for the efficient comprehensive WVR head pose estimation is introduced.

A. Datasets

In our experiments we used synthetic data *Synth* consisting of data generated by the 3DMM. We generated 150 random faces per degree, from a yaw angle of $-90°$ to $90°$, in a $1°$ interval, resulting in a total of 27,150 different faces. Each face was generated randomly with a quite large variance in the principal components of the 3DMM for the shape and texture, so that each face is a different subject. Additionally, a realistically large variance of the illumination was chosen, to mimic conditions that occur in real images. Some example pictures are shown in Fig. 2. To make the scenario as realistic as possible, a background was added to each image, chosen randomly from a list of 4,135 background images. These images were cropped and scaled to a resolution of 32×32 pixels. 100 of the images per degree were used for training and 50 for testing.

For natural data, the *CasPeal* [25] database was used for both training and testing. This database is a large database of

Figure 2. Example data from the 3DMM. Different, random generated data at different yaw angles. From left to right: $35°$, $55°$, $-45°$, $80°$, $-20°$.

Chinese face images, with 1,040 individuals acquired with nine cameras spread across varying yaw angles and varying illumination. Three pitch angles were acquired by asking the subject to look forward, up and down in front of the nine cameras. The pitch angles were approximately $-30°$, $0°$ and $30°$. The pictures from the nine cameras were taken simultaneously within two seconds.

Also used for training were images from the *MultiPie* [26] database and *FacePix* database [27]. The *MultiPie* database consists of face images of 337 individuals, acquired with 15 cameras and 19 illumination conditions. The *FacePix* database consists of face images of 30 individuals with frontal pitch and yaw angles ranging from $-90°$ to $90°$ with one image per degree.

B. Head-pose Estimation Results of the Reduced Support Regression Training

1) Support Vector Regression Training: First a full SVR is trained to estimate the yaw angle of the head pose. Support Vector Regression is known to obtain best regression performance, but is very time consuming. We compare our trained SVR with the most promising existing approaches for head pose estimation, introduced in Section I. As nearly every work uses different data for testing and different pose angle variations, we use an SVR trained on a yaw angle of $\pm 90°$ to obtain a fair comparison. For example, [6] train on a yaw angle of $-90°$ to $90°$, in steps of $2°$, and are able to achieve an average error in yaw angle of $1.44°$ using *FacePix* as training and test set. Chutorian et al. [3], using support vector regression, but another feature space and no reduction, operate on a yaw angle of $-80°$ to $80°$ and a pitch angle of $-30°$ to $30°$. They are able to estimate the yaw angle with a mean absolute error of $6.40°$. Ma et al. [2] operate on a yaw angle of $-60°$ to $60°$ and a pitch angle of $-30°$ to $30°$ and report a yaw error of $< 7.5°$ for 88.6% of the test data. Fig. 3 shows the obtained head pose estimation results for our SVR trained on the set *Synth*. With an average error of $1.30°$ on a range of $\pm 90°$ we obtain the best results compared with leading approaches. This results in an error of $\leq 5°$ on 90.7% of the subjects.

But similar to [6], despite having generated the most natural looking data as possible, the train and test sets are not as complex as images from real-life use cases. To overcome this limitation, we included subjects from the *CasPeal*-database into the training set, now consisting of synthetic and natural images. Results on the natural database *CasPeal*, for subjects not in the training set, are shown in Fig. 3. Our approach shows promising results, but it has the drawback of a very complex, slow SVR. We will address this next.

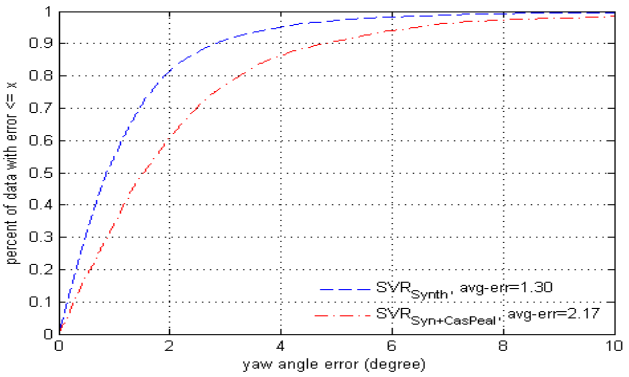

Figure 3. SVR performance on *Synth* and *CasPeal*: Blue (dashed): Performance of the synthetic full SVR tested on synthetic test data. Red (dash-dotted): SVR trained with the same synthetic data, additionally with real data from *CasPeal*. Tested on real *CasPeal* images of different subjects. We see a small drop in performance for the real dataset, but still with very good results (90.7% have an error of $\leq 5°$).

2) Results of Reduced Support Regression Training: Now the results of the reduction of the full SVR$_{syn+cas}$ shown in Fig. 3 are demonstrated. First we approximate the full set of support vectors of the SVR by a smaller set of Reduced Regression Vectors (RRVs), as described in Section II-B2. An RBF kernel is used, with a grayscale feature space and histogram equalization as normalization. As seen in Fig. 4, the SVM with 17,362 SSVs can be highly approximated by a WVR-R with 300 RRVs, a decrease in computation complexity by a factor of 58. The average error on the *CasPeal* test set has only increased by $0.87°$. This makes our approach suitable for real-time application to videostreams without losing much accuracy. The SVR can even be reduced further to 100 RRVs without suffering from a large performance decrease. With only 100 RRVs, still 74% of the images have an error of $\leq 5°$, yielding a 174 times faster regression for a rough first estimation stage.

As demonstrated in Section II-B4 and II-B5, the gained runtime performance is improved in the final application by taking advantage of the Integral Image and Cascaded Regression concept of the Wavelet Reduced Vector Regression.

Optimal results are obtained by the comprehensive WVR including all the core ideas introduced in Section II-B. Fig. 5 shows some examples for Wavelet Regression Vectors from the Double Cascade of a WVR. By taking advantage of the Cascaded Regression (introduced in Section II-B5) the coarse-to-fine method is used to find a good approximation of the angle after just a few resolution levels, not running up to the last vector if the resulting angle is already clear and only marginally changing. This leads to an optimal and adjustable balance between run-time performance and accuracy. The adjustable complexity and efficiency of the WVRs gives us the opportunity to build up the Evolutionary Regression Tree (Section II-C). Therefore, head pose estimation in real-time is feasible. A high accuracy of the pose estimation is obtained because the full Support Vector Regression estimation is used at the last coarse-to-fine stage of the Evolutionary Regression Tree.

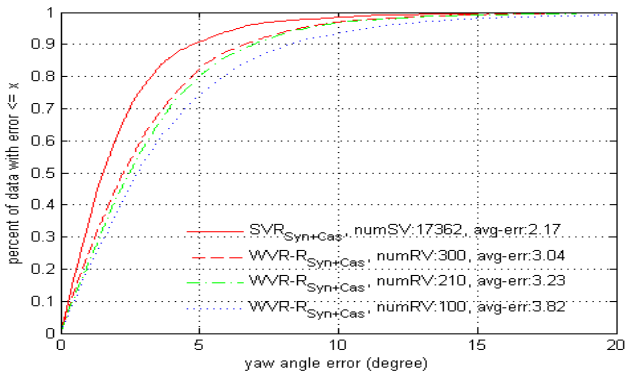

Figure 4. Reduction: Full SVR from Section III-B1 with additonal natural data tested on *CasPeal* test set (red line); WVR-Rs with reduced sets of vectors - 300 (red dashed), 210 (green dashed-dotted) and 100 (blue line with dots), instead of the 17,362 support vectors of the full SVR. A reduction of the computational complexity by a factor of 58 to 174 is obtained.

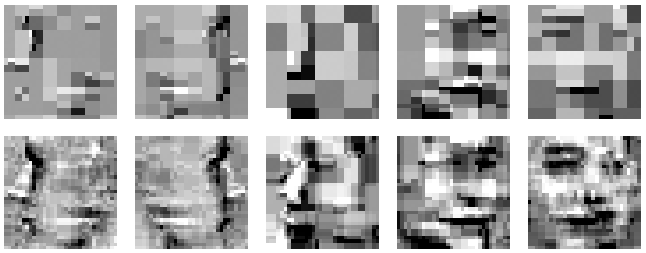

Figure 5. Examples for Wavelet Regression Vectors from the Double Cascade. The first row shows the 1^{st}, 2^{nd}, 8^{th}, 17^{th} and 36^{th} WRV at one of the first, very coarse approximation levels. The block-structure is clearly visible. The second row shows the same WRVs at a finer resolution level.

3) Improvements for Natural Data based on Feature Space Transformation and Analytical Reduction: As an experiment with natural data and analytical SVR reduction over a smaller yaw range, a training set for yaw pose angle was formed from portions of the *CasPeal*, *MultiPie* and *FacePix* data (see Section III-A). For *CasPeal* training the yaw angle ranged from $-30°$ to $30°$ yaw, with all three pitch poses included and 771 individuals and 11,099 images. For *MultiPie*, 3,684 images were used, randomly selected from images with $-30°$ to $30°$ yaw angle and $0°$ pitch angle and varying illumination. For *FacePix*, 1,606 images were used, with yaw angles ranging from $-36°$ to $36°$ yaw and $0°$ pitch.

For testing, a portion of *CasPeal* with 269 individuals and 3,885 images and again yaw angles from $-30°$ to $30°$ yaw and the three pitch angles were used. Individuals in the training set were not used in the testing set.

Faces and eyes were found using commercial detectors. Advanced features with better discriminative power were used as input for the Support Vector Regression rather than simple grayscale features. These features are proprietary and used in a commercial system, so no further details can be made public. For the regression, an inhomogeneous quadratic kernel was used in order to utilize an efficient analytic reduction technique [23] rather than gradient descent (see Section II-B3). Using this technique allowed a reduction from 11,463 SSVs to 25 RRVs with almost no change in performance. These results can be seen in Fig. 6 for the *CasPeal* testing results. Approximately 80% of the images had errors less than $4°$ for a fully automatic algorithm.

Tests were only done with a WVR-R and not with a comprehensive WVR using Cascaded Regression and Integral Images, because the cascade idea is not easily applicable to the analytically found vectors, and the Integral Image approach is not usable for these advanced feature vectors.

The results of the improvements introduced here show a stronger reduction of the computational complexity by a factor of up to 560 and a higher estimation accuracy. The approach is suitable for natural data in uncontrolled and uncooperative conditions (e.g. large range of pose angles).

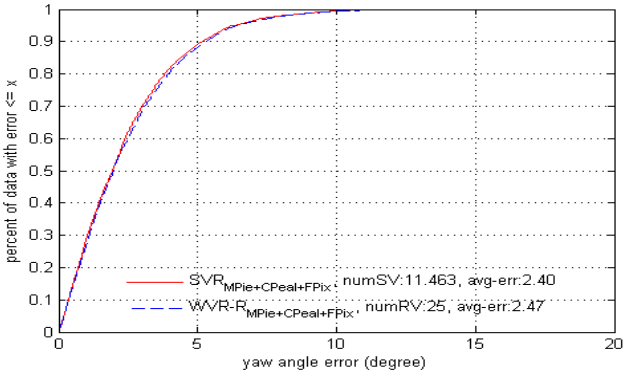

Figure 6. Improvements for Natural Data using feature space transformation and polynomial kernel, enabling analytical optimization technique. Extreme reduction of computational effort by no significant loss of accuracy.

C. Application

1) Measurement: The performance and accuracy of the object tracking and pose estimation is demonstrated on an example of pose invariant face tracking in videostreams. The Evolutionary Regression Tree (ERT) introduced in Section II-C is applied on the former proposed Cascaded Condensation Tracking (CCT) [13]. Coarse-to-fine evolutionary loops based on the adjustable complexity of WVM classifiers (Section II-B1) and WVR pose angle estimations (Section II) are realized.

CCT uses a contraction of the sampling locations in the features space to areas based on the probability density function (PDF) obtained by its measurement function. A WVM consists of four stages, the first stage (i) is a WVM Double Cascade to reject most of the non-object sampling points. It is followed by: (ii) a first Overlap Elimination (OE) reducing the number of samples per object cluster, (iii) the full complex SVM, (iv) a second OE and (v) a reduction of clusters to a number of expected objects.

The loops for the PDF measurement are realized starting with the WVM stage (i) of a global WVM classifier, trained as root node of the ERT over the full pose range from full left to full right profile ($\pm 90°$ yaw). Less than 30% of the samples are remaining depending on the complexity of the background of the video stream. After the 1^{st} OE stage (ii), the first, weak and most efficient WVR estimation of the pose angle is used to decide which one of the branches will be continued. In case the angle lies near the border of two subranges, two branches are used. At the moment the three ranges -90° to -40°, -40° to 40° and 40° to 90° are used. Both WVR stages are very efficient, taking advantage of the Cascaded Regression method (Section II-B5). Efficient WVM Double Cascades trained for that specific subspaces are obtained. After a next OE the full SVR estimation is used to obtain the final most accurate pose estimation. This estimation over the last remaining patches (typically less than 1%) is used to decide which specific trained full SVM stage (iii) will be applied next. A final OE and a reduction of clusters (stages (iv) and (v)) are applied.

2) Tracking: The PDF consisting of the measurement from the previous section is very accurate for the remaining positive samples, but also used for the rejected samples. The new sample distribution function is computed from the PDF as published for the CCT [13]. The gained frame rate depends on the used hardware, complexity of the background, number of used image pyramid scales and number of persons. Fig. 7 shows tracking results on few frames on a videostream in real-time with more than 15fps on a Core i7 with a webcam using a VGA 640x480 pixel resolution.

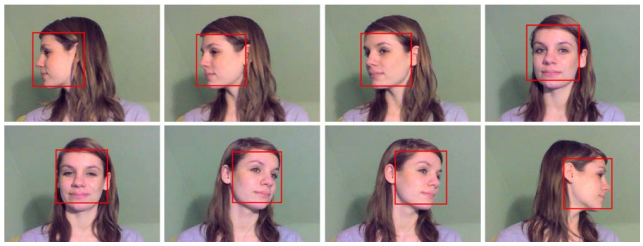

Figure 7. Example ERT tracking results on frames of a videostream.

IV. Conclusion

We were able to successfully develop a framework for real-time Support Vector Regression, based on previous work for support vector reduction. Where direct adaption was not possible, for example the cascade, we found new, similar solutions. We demonstrate this new Wavelet Reduced

Vector Regression approach on the task of head pose estimation of human faces. The proposed novelties make this approach real-time capable up to full profile view, improving our Cascaded Condensation Tracking. We showed a fast kernel evaluation based on Integral Images and a Cascaded Regression, which automatically uses the optimal number of WRVs. Using an Evolutionary Regression Tree, we were able to combine the classification and regression step of face detection and pose estimation. The tree uses a coarse-to-fine approach, such that it is capable of handling the huge space of faces and pose angles in real-time.

In future work, we wish to investigate the possibility of simultaneously estimating the three pose angles using regression for multi-dimensional labels [28]. We plan to combine the advantages of the analytical reduction of the polynomial kernel with the advantages of Integral Images and our Cascaded Regression.

ACKNOWLEDGMENTS

The authors would like to thank Michael Fischer for training WVMs and his work on WVM-trees. Also we would like to thank Andreas Forster, Thorsten Thies, and Klaus Luig for very valuable discussions and implementations.

REFERENCES

[1] H. Moon and M. Miller, "Estimating facial pose from a sparse representation," *Proc. IEEE Int'l Conf. Image Processing*, pp. 507 – 78, 2004.

[2] Y. Ma, Y. Konishi, K. Kinoshita, S. Lao, and M. Kawade, "Sparse bayesian regression for head pose estimation," *Proc. 18th Int'l Conf. Pattern Recognition*, pp. 507 – 510, 2006.

[3] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation," *Proc. 10th Int'l IEEE Conf. Intelligent Transportation Systems*, pp. 709 – 714, 2007.

[4] Y. Fu and T. Huang, "Graph embedded analysis for head pose estimation," *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 3 – 8, 2006.

[5] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607 – 626, 2009.

[6] V. Balasubramanian, J. Ye, and S. Panchanathan, "Biased manifold embedding: A framework for person-independent head pose estimation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.

[7] D. Huang, M. Storer, F. Torre, and H. Bischof, "Supervised local subspace learning for continuous head pose estimation," in *CVPR*, 2011, pp. 2921–2928.

[8] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," *ACM SIGGRAPH*, pp. 187 – 194, 1999.

[9] T. Vetter, "Synthesis of novel views from a single face image." *International Journal of Computer Vision*, vol. 28, no. 2, pp. 103 – 116, 1998.

[10] M. Rätsch, G. Teschke, S. Romdhani, and T. Vetter, "Wavelet frame accelerated reduced support vector machines," *IEEE Transactions on Image Processing*, vol. 17(12), 2008.

[11] S. Romdhani, P. Torr, B. Schölkopf, and A. Blake, "Computationally efficient face detection," in *Proceedings of the 8th International Conference on Computer Vision*, July 2001.

[12] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *IJCV*, 1998.

[13] M. Rätsch, C. Blumer, T. Vetter, and G. Teschke, "Efficient object tracking by condentional and cascaded image sensing," *Advanced Computing and Interfacing Systems, Elsevier Journal of Computer Standards & Interfaces*, 2010.

[14] H. Sahbi, D. Geman, and P. Perona., "A hierarchy of support vector machines for pattern detection," *Journal of Artificial Intelligence Research*, vol. 7, pp. 2087 – 2123, 2006.

[15] V. Vapnik, *The Nature of Statistical Learning Theory*. N.Y.: Springer, 1995.

[16] M. Osadchy, Y. LeCun, and M. Miller, "Synergistic face detection and pose estimation with energy-based models," *In Toward Category-Level Object Recognition*, pp. 196 – 206, 2006.

[17] A. J. Smola, "Regression estimation with support vector learning machines," Diplomarbeit, TU München, 1996.

[18] Y.-J. Lee, W. F. Hsieh, and C. M. Huang, "Machine for epsilon-insensitive regression," *IEEE Transactions on knowledge and data engineering*, vol. 17(5), 2005.

[19] C. J. C. Burges, "Simplified support vector decision rules," in *13th Intl. Conf. on Machine Learning*, 1996, pp. 71–77.

[20] I. Kukenys and B. McCane, "Classifier cascades for support vector machines," in *Image and Vision Computing New Zealand (IEEE)*, Nov. 2008, pp. 1–6.

[21] A. Marconato, M. Gubian, A. Boni, and D. Petri, "Support vector machines for system identification," *DIT Technical Report; 07-023*, Mai 2007.

[22] 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 Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000 – 1017, 1999.

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

[24] F. Crow, "Summed-area tables for texture mapping," *In Proc. of SIGGRAPH*, vol. 18(3), pp. 207 – 212, 1984.

[25] W. Gao, B. Cao, S. Shan, D. Zhou, X. Zhang, and D. Zhao, "The cas-peal large-scale chinese face database and evaluation protocols," Joint Research and Development Laboratory, CAS, Tech. Rep., 2004.

[26] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker, "Multi-pie," in *Proceedings of the Eighth IEEE Int'l Conference on Automatic Face and Gesture Recognition*, 2008.

[27] J. A. Black, M. Gargesha, K. Kahol, P. Kuchi, and S. Panchanathan, "A framework for performance evaluation of face recognition algorithms," 2002.

[28] G. Guo and G. Mu, "Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression," *(Hebei University of Technology) CVPR*, 2011.