A Real-Time System for Head Tracking and Pose Estimation

Zengyin Zhang[1], Minyoung Kim[1], Fernando de la Torre[1], and Wende Zhang[2]

[1] Robotics Institute, Carnegie Mellon University
[2] Electrical & Controls Integration Lab, General Motors R&D

Abstract. Driver's visual attention provides important clues about his/her activities and awareness. To monitor driver's awareness, this paper proposes a real-time person-independent head tracking and pose estimation system using a monochromatic camera. The tracking and head-pose estimation tasks are formulated as regression problems. Three regression methods are proposed: (i) individual mapping on images for head tracking, (ii) direct mapping to subspace for head tracking, which predicts a subspace from one sample, and (iii) semantic piecewise regression for head-pose estimation. The approaches are evaluated on standard databases, and on several videos collected in vehicle environments.

1 Introduction

Monitoring driver's activity can greatly reduce the number of accidents by detecting situations, such as lack of attention. There exist several methods to potentially characterize driver's behavior [1]:

1. Fitness-for-duty technologies. These methods are based on performing tests by the driver to evaluate his/her capacity.
2. Mathematical models of alertness dynamics joined with ambulatory technologies. This approach involves the application of mathematical models that predict operator awareness/performance at different times based on circadian circles and related temporal antecedents of fatigue or distraction.
3. Vehicle-based performance technologies. Additional hardware is added to the transportation system to control the operator (i.e. lane deviation, steering, speed variability, etc.).
4. In-vehicle, on-line technologies. Technologies in this category seek to record some bio-behavioral dimension(s) of an operator, such as features of the eyes, face, head, heart, brain electrical activity and reaction time.

We focus on a non-invasive monitoring technologies to track driver's head and estimate his/her head orientation, because these are important clues to indicate driver's alertness. We strove for a systems that met the following criteria: (a) the system should be able to track head location and estimate head pose from a single camera, and (b) it must work invariant to different drivers and driving environments. The commercial available softwares of head tracking and pose

K.N. Kutulakos (Ed.): ECCV 2010 Workshops, Part I, LNCS 6553, pp. 329–341, 2012.

estimation work well in controlled environments (office space) but not in uncontrolled environments (driving under various lighting conditions). In this paper, we present a real-time system that can track head position and estimate head pose from a monocular camera. The main contribution is to formulate the head tracking and pose estimation problems as regression problems. Three regression methods are proposed: (i) individual mapping on images, (ii) direct mapping to subspace, which predicts a subspace from one sample for head tracking across pose, and (iii) semantic piecewise regression for head-pose estimation.

The rest of the paper is organized as follows: Section 2 reviews previous work on face tracking and head pose estimation. Section 3 reviews literature on ridge regression, reduced-rank regression and support vector regression (SVR). Section 4 summarizes the overall system. Section 5 describes the head tracking module. Section 6 discusses the pose estimation module. Experimental results are shown in Section 7, following by the summary in Section 8.

2 Previous Work

2.1 Head Tracking

Since the early work of Sirovich and Kirby [2], parameterizing the human face using Principal Component Analysis (PCA) [3] and the successful eigenfaces of Turk and Pentland [4], many computer vision researchers have used subspace techniques to construct linear models of optical flow, shape or gray value for tracking [5,6], detection and recognition [4,7]. The modeling power of subspace techniques is especially useful when applied to visual data, because there is a need for dimensionality reduction given a large number of features. In particular, PCA is a common statistical model to build model of face variation across pose, illumination and expression. Furthermore, the target, i.e., a driver, does not change during the tracking. Therefore, a person-specific subspace model usually outperforms a general one. The challenge is how to build a person-specific subspace model for an arbitrary person. Ross et al. [8] proposed a method to efficiently update a low dimensional subspace of the object to track, and successfully applied to tracking faces across challenging conditions. However, since this is self-learning or unsupervised learning, it is potentially unable to judge whether the current decision is correct or not in a credible manner, which is indeed the main cause of tracking drift. In our driving scenario, most of the appearance variations come due to changes in illumination and pose. A well built subspace that represents these variations should perform better than incrementally updating methods. Two regression methods are proposed in this paper to build such a subspace for head tracking.

Active Appearance Models (e.g. [9,10]) have been popular techniques to track and decouple rigid and non-rigid motion. AAMs use a combination of appearance and shape models, and formulate the tracking problem as a non-linear optimization with respect to the piecewise affine parameters. However, we use a low-resolution camera that makes AAMs not practical in our setting. Another popular set of techniques for head tracking are based on skin color (e.g. [11]).

However, color constancy is still an open research problem. To avoid the affect of color shifting in uncontrolled environment, skin-color-based techniques are not selected in our system.

2.2 Pose Estimation

Head pose estimation refers to the problem of estimating *pitch, roll*, and *yaw* angles from images. It is a relatively unexplored problem in comparison with head tracking.

There are several categories of methods to solve this problem, see [12] for a full review. Appearance template-based methods [13,14] compare an input image with a set of training samples that have been previously labeled with angles. These approaches are sensitive to the specific measure of similarity between samples, and a large number of samples are required to achieve a good precision. A second category of methods are the ones based on discriminative models [15] that classify images into several discrete poses. Non-linear regression approaches, such as Support Vector Regression (SVR) [16,24] and locally-linear projection [17] are able to estimate continuous measures of angles in head pose estimation. Also, AAMs [10,18] can be used to estimate head pose, however they are often trapped into local minima.

In our driving scenario, we aim to get numerical estimation of the head pose under various illumination conditions. Pose changes are non-linear in the appearance space, and non-linear regression methods are typically more appropriate to model this non-linearity. In this paper, we propose a local extension of SVR that outperforms standard regression methods such as kernel ridge regression and SVR.

3 Background

This section reviews previous work on Ridge Regression, Reduced-Rank Regression [19] and Support Vector Regression [20], which are applied or extended in further sections.

3.1 Ridge Regression

Let $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ be n training samples, where $\mathbf{x}_i \in \Re^{p\times 1}$ is a p-dimensional input sample and $\mathbf{y}_i \in \Re^{d\times 1}$ is the corresponding d-dimensional output sample. In the linear[1] regression, the prediction function has a linear form, i.e., $\mathbf{f}(\mathbf{x}) = \mathbf{W}^T\mathbf{x}$, where $\mathbf{W} \in \Re^{p\times d}$ is the matrix to be learned. We use the matrix notation for input, i.e., $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$, which is of size $(p\times n)$. Similarly, $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n]$ is of size $(d \times n)$. Following the regularized empirical risk minimization framework, we minimize:

$$\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{W}^T\mathbf{X}\|_F^2 + \lambda\|\mathbf{W}\|_F^2 \tag{1}$$

[1] The non-linear extension with kernel tricks is straightforward.

where λ controls the degree of regularization. (1) is often called the ridge regression, which admits a closed form solution: $\mathbf{W} = (\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{Y}^T$.

3.2 Reduced-Rank Regression

The reduced-rank regression (RRR) model, introduced by Anderson [19] in the early 1950s, has attracted much attention and has been successfully applied in many fields including computer vision and machine learning. The RRR learns a mapping between input $\mathbf{x}$ and output $\mathbf{y}$ by minimizing:

$$\min_{\mathbf{A},\mathbf{C}} \sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{A}\mathbf{C}^T\mathbf{x}_i\|_2^2 \tag{2}$$

where $\mathbf{A} \in \Re^{d\times q}$ and $\mathbf{C} \in \Re^{p\times q}$. One may guess that $\mathbf{A}$ and $\mathbf{C}$ can be obtained from singular value decomposition (SVD) of the learned $\mathbf{W}$ from (1). However, as shown in [21], this often yields inferior results to simultaneously optimizing $\mathbf{A}$ and $\mathbf{C}$ from the least square optimization. It is known that the least square solution has no local minima as it reduces to learning the canonical correlation analysis (CCA) [22] embedding on $\mathbf{x}$ (to learn $\mathbf{C}$) followed by the least square regression estimation for $\mathbf{A}$ from embedding of $\mathbf{x}$, i.e., $\mathbf{C}^T\mathbf{x}$ to $\mathbf{y}$.

3.3 Support Vector Regression (SVR)

The goal of Support Vector Regression is to find a linear function[2], $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$, which has at most ε deviation from the actual output y_i for all training data. This is also called ε-SVR [20]. The SVR is formulated within the maximum margin framework and minimizes:

$$\begin{aligned} \min_{\mathbf{w},b,\xi,\xi^*} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum_i(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - \mathbf{w}^T\mathbf{x}_i - b \le \varepsilon + \xi_i \\ & \mathbf{w}^T\mathbf{x}_i + b - y_i \le \varepsilon + \xi_i^* \\ & \xi_i \ge 0, \quad \xi_i^* \ge 0 \end{aligned} \tag{3}$$

where $\mathbf{w}$ and b are the parameters of SVR. $\{\xi_i\}$ and $\{\xi_i^*\}$ are slack variables. ε and C are given constants. SVR typically is robust to outliers and maximizes the margin in the regression.

4 Overall System

This section describes the design of our real-time head tracking and pose estimation system for driver's alertness. The system scenario is within a vehicle with a camera mounted at a fixed position. Videos are captured with an analog camera.

[2] The non-linear extension with kernel tricks is straightforward in its dual form.

To deal with local illumination changes, the frame image is first normalized using histogram equalization. Tracking in this scenario is quite challenging because there are large variations in illumination and pose changes. One of the major challenges is that the driver's facial pose can vary abruptly and dramatically.

In this context, we first used the OpenCV face detector to detect the frontal face. Then, a person-specific appearance-based subspace tracker [5,8] was applied which is robust to noise and large variation in appearance. Specifically, the appearance model of the tracker is a subspace that can capture the variability of the target appearance, which is combined with efficient searching strategies to yield the head tracker. Sampling-based particle filtering [23] method is a typical searching strategy in tracking tasks. In this paper, we also use particle filter method to estimate motion parameters. Instead of incremental updating subspace, we learn a subspace directly from the frontal image. This is more efficient as we do not need to update the subspace during the tracking procedure. After the head tracking procedure, we can crop the head region from the image frame. Then, a regression based approach is applied to estimate head pose of the driver, so we can give continuous measurements. In this regression framework of head pose estimation, Histogram of Oriented Gradients (HOG), which is robust to noise and illumination changes, is one of the most widely used features [24,25]. We choose one extension of HOG called pyramid histogram of oriented gradients (PHOG) as features.

The flowchart is shown in Fig. 1. This system is composed of three modules: initialization, head tracking and pose estimation. First, we detect a frontal face. Then, we learn a person-specific subspace via pose changes for the tracker. Rest of the frame images are processed by head tracking module and pose estimation module. The following two sections describe details of head tracking and pose estimation modules.

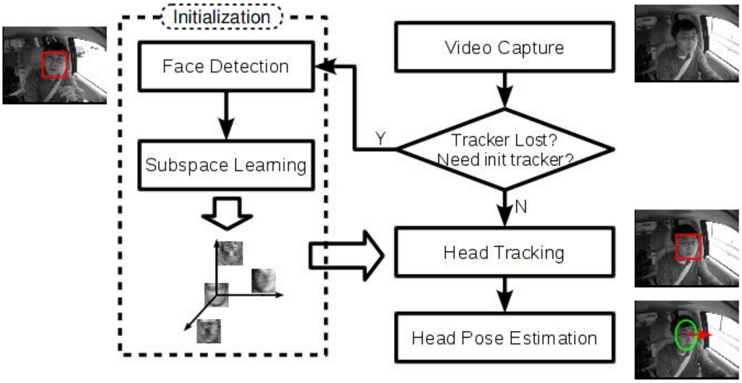

Fig. 1. Flowchart of Overall System

5 Head Tracking

This section describes the module of head tracking. The head tracking module is based on a subspace tracker [5,8]. Observe that the most common head motion for a driver is to move the head from profile to profile looking. To track the head across pose, we build a person-specific subspace of pose from a frontal image.

5.1 Subspace Learning

We describe two methods that learn the mapping between one sample and a subspace: Individual Mapping on Images (IMI) and Direct Mapping to Subspaces (DMS). Assume that there are images of different poses belong to n subjects. Each subject i has $K+1$ images (denoted by $\mathbf{x}_i^s$ of different poses, $s \in \{0, 1, \cdots, K\}$). The state $s = 0$ is reserved for indicating the reference state. i.e., frontal face.

Individual Mapping on Images (IMI). In this method, we learn a regression function for each state that maps the reference sample $\mathbf{x}^0$ to the s-state sample $\mathbf{x}^s$. This enables us to estimate the subspace for a new subject using PCA with the generated images from the learned regressors. Formally, we form the training data for regression as: $\{(\mathbf{x}_i^0, \mathbf{x}_i^s)\}_{i=1}^n$, for $s = 1, \cdots, K$. In the reduced-rank model we solve:

$$\min_{\{\mathbf{A}_s\},\{\mathbf{C}_s\}} \sum_{s=1}^{K} \sum_{i=1}^{n} \|\mathbf{x}_i^s - \mathbf{A}_s \mathbf{C}_s^T \mathbf{x}_i^0\|_2^2 \tag{4}$$

Where $\mathbf{A}_s \in \Re^{p\times l}$ and $\mathbf{C}_s \in \Re^{p\times l}$ (for $s = 1, \cdots, K$) are the parameters of the model, and l ($\ll p$) is the reduced-rank dimension. Once we solve Equation (4), given an input image of state 0 of an unseen subject $*$, $\mathbf{x}_*^0$, the synthesized sample at state s is: $\mathbf{x}_*^s = \mathbf{A}_s \mathbf{C}_s^T \mathbf{x}_*^0$. Therefore, we generate samples for all possible states, $\mathbf{x}_*^s$ for $s = 1, \cdots, K$, from which we can build a person-specific subspace for the subject $*$ via PCA, afterwards.

Direct Mapping to Subspace (DMS). DMS learns a direct mapping between the reference sample ($\mathbf{x}_0$) and the subspace built with all samples at different states ($\mathbf{x}^s$, for $s = 0, 1, \cdots, K$). In this setting, the training data can be formed as $\{(\mathbf{x}_i^0, \text{vec}(\boldsymbol{\mu}_i, \mathbf{B}_i))\}_{i=1}^n$, where the output $\text{vec}(\boldsymbol{\mu}_i, \mathbf{B}_i)$ is the subspace of subject i. $\boldsymbol{\mu}_i$ is the mean vector and $\mathbf{B}_i$ is the subspace. Here, $\text{vec}(\cdot)$ is the operator to transform vectors and matrices to a long vector. Typically, $\text{vec}(\boldsymbol{\mu}_i, \mathbf{B}_i)$ is learned via PCA from the training samples $\{\mathbf{x}_i^s\}_{s=0}^K$. Similar to the IMI approach, as the output consists of many variables (e.g., mean and eigenvectors), we apply reduced-rank regression.

5.2 Subspace Tracking

Once the person-specific pose subspace ($\mathbf{B}$) has been estimated, the tracking is done with a particle filtering algorithm [23] that minimizes:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \left(\min_{\mathbf{c}} \|I(\mathbf{M}(\mathbf{x};\mathbf{p})) - \mathbf{B}\mathbf{c}\|_2^2 \right) \tag{5}$$

where $\mathbf{B}$ denotes the person-specific subspace previous identified. $\mathbf{p}$ is the parameter of similarity transform, $\mathbf{p} = (u, v, r, s)$, where u, v indicate the center of tracking box; r indicates the rotation angle and s indicates the scale factor. $\mathbf{M}(\cdot)$ is the similarity transformation function. $I(\mathbf{M}(\mathbf{x};\mathbf{p}))$ is the warped image. For instance, the pixel at coordinate $\mathbf{x}$ of the warped image has the gray intensity value of input image I at coordinate, $\mathbf{M}(\mathbf{x};\mathbf{p})$. Coefficient $\mathbf{c}$ minimizes the least square error between the warped image under the given motion parameter $\mathbf{p}$ and the reconstructed one, which is computed with a closed form solution: $\mathbf{c} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T I(\mathbf{M}(\mathbf{x};\mathbf{p}))$. The motion parameters, $\mathbf{p}$'s, are generated as particles. The objective function (5) infers to identify the best particle which has the minimum reconstruction error.

6 Pose Estimation

This section describes the pose estimation module. Since the yaw angle is the most dominant head motion for a driving situation, we focus on applying regression approaches to estimate the yaw angle for the driver's head motion.

The following describes a piecewise extension of SVR for accurate pose estimation. [24] showed the performance of SVR in head pose estimation tasks. Generally, piecewise regression strategy is a practical extension. The trade-off is that one needs careful effort in segmenting data to avoid overfitting. Typical piecewise regression divides the input space into several subspaces, and learns a linear or non-linear function to predict outputs for each input subspace. Due to the high dimensionality of input space, it is difficult to determine how many segments we should have and where to segment the space. However, recall that output values are typically more meaningful than the corresponding samples in input space. Classification in output space is much easier and more intuitive than in input space. For instance, humans can easily distinguish the head poses by its yaw angle, but more difficult to separate them by facial details. Piecewise Support Vector Regression with segmentation strategy in output space would be a better solution. Unfortunately, compared to the traditional piecewise regression approaches, the challenge of piecewise regression in the output space is to determine which regression function should be chosen for prediction. We propose the semantic piecewise regression which overcomes this problem when the output values are bounded.

As we described in Section 3.3, SVR is a well formulated maximum marginal method to learn the regression between inputs and outputs. We assume that the output value is bounded, i.e., $y_i \in [L, U]$. Instead of learning a regressor over all output space, we want to learn a function over a smaller interval, $[L_k, U_k] \subseteq [L, U]$. We can divide the inputs into three sets, $\mathcal{V}_k^-$, $\mathcal{V}_k$ and $\mathcal{V}_k^+$. $\mathcal{V}_k$ contains the inputs that have outputs in the interval, i.e., $\mathbf{x}_k^i \in \mathcal{V}_k, y_k^i \in [L_k, U_k]$. $\mathcal{V}_k^-$ contains the inputs that have output values smaller than the lower bound of interval, i.e., $\mathbf{x}_k^{u^-} \in \mathcal{V}_k^-, y_k^{u^-} < L_k$ and $\mathcal{V}_k^+$ contains the rest. We wish our regression function is only able to predict the relationship on $\mathcal{V}_k$, i.e., the outputs will be between L_k and U_k. To avoid ambiguity, we add another two constraints: all outputs on $\mathcal{V}_k^-$ are less than L_k, and all outputs on $\mathcal{V}_k^+$ are greater than U_k. In this case, we can easily recognize whether the regressor is acceptable of a given input by its corresponding output. Acceptable if it is within the interval $[L_k, U_k]$, unacceptable if not. The objective function can be formulated within maximum margin framework as:

$$\min_{\mathbf{w},b,\xi,\xi^*,\eta} \quad \frac{1}{2}||\mathbf{w}||^2 + C\left(\sum_i(\xi_i + \xi_i^*) + \sum_u \eta_u\right) \tag{6}$$

$$\begin{aligned} \text{subject to} \quad & y_i - \mathbf{w}^T\mathbf{x}_i - b \leq \varepsilon + \xi_i \\ & \mathbf{w}^T\mathbf{x}_i + b - y_i \leq \varepsilon + \xi_i^* \\ & S(y_u)[\mathbf{w}^T\mathbf{x}_u + b - \frac{1}{2}(L_k + U_k)] \geq \frac{1}{2}(U_k - L_k) - \eta_u \qquad (7) \\ & \xi_i \geq 0, \quad \xi_i^* \geq 0, \quad \eta_u \geq 0 \end{aligned}$$

where $\mathbf{x}_i \in \mathcal{V}_k$ and $\mathbf{x}_u \in \mathcal{V}_k^- \cup \mathcal{V}_k^+$. $S(y_u)$ equals 1 if $y_u > U_k$, -1 if $S(y_u) < L_k$. Inequality (7) forces the outputs of $\mathbf{x}_u$ to be outside of $[L_k, U_k]$, i.e., the distance to the interval's center is greater than interval's half size. This is inspired by avoiding confusion when a test sample is applied in several regression functions. Incorrect regression functions will return values beyond their effective ranges, while the correct regression function will return a valid value within its range.

7 Experimental Results

We conducted experiments on predicting a person-specific facial pose subspace from a frontal image and estimated the head pose on both standard databases and collected driving videos.

7.1 Subspace Pose Prediction

CMU PIE data set [27] is composed of 41,368 images from 68 people. The face of each person is taken under 13 different poses, 43 different illumination conditions, and 4 different expressions. The images are labeled with these states. We used a subset of these images by taking 720 images of 60 subjects with 12 different poses with the same expression and illumination conditions. Each face image is cropped into a tight bounding box using the ground-truth facial landmark points which are also provided by the data set. The images of all poses of five subjects are shown in Fig. 2. All the images are with the same expression and the same

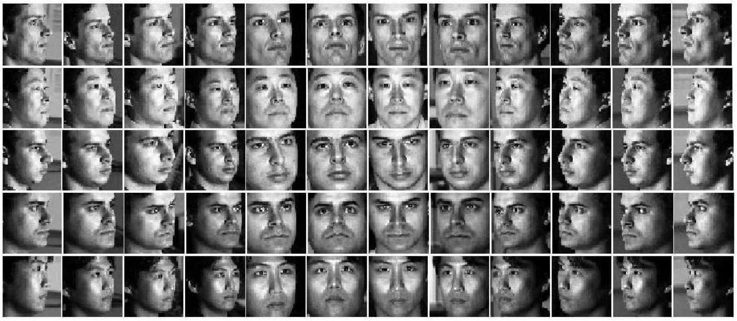

Fig. 2. Different pose images for first five subjects in the CMU PIE data set [27].The images are arranged in a way that each row corresponds to a particular subject while each column represents a specic state (pose in this case).

illumination condition. We normalized the images into the same size (48 × 48). For each subject, we estimated a PCA subspace with a fixed dimension. Then we split the data randomly into 50/10 training/testing subjects. By revealing only a single frontal image for each subject in the test fold, we predicted the subspaces of the test subjects. To measure the quality of subspace alignment (between the estimated subspace and the ground-truth subspace obtained from real samples), we used three quantitative error metrics: (i) the smallest principal angle [22], (ii) the sum of the squared cosine angles between basis vectors, and (iii) the subspace distance defined in [28]. More specifically, to assess the distance between two subspaces $\mathbf{B}_1 = [\mathbf{b}_1^{(1)}, \cdots, \mathbf{b}_q^{(1)}]$ and $\mathbf{B}_2 = [\mathbf{b}_1^{(2)}, \cdots, \mathbf{b}_q^{(2)}]$, we performed the SVD decomposition: $\mathbf{B}_1 = \mathbf{U}_1\boldsymbol{\Sigma}_1\mathbf{V}_1^T$ and $\mathbf{B}_2 = \mathbf{U}_2\boldsymbol{\Sigma}_2\mathbf{V}_2^T$. Then the sum of the squared cosine angle errors between two subspaces can be defined as:

$$d_1(\mathbf{B}_1, \mathbf{B}_2) = 1 - \frac{1}{q}\sum_{j=1}^{q}\left(\mathbf{u}_j^{(1)^T}\mathbf{u}_j^{(2)}\right) \tag{8}$$

where $\mathbf{u}_j^{(m)}$ is the j^{th} column of $\mathbf{U}_m$ for $m = 1, 2$. The subspace distance is defined in [28], which has the following formula:

$$d_2(\mathbf{B}_1, \mathbf{B}_2) = \sqrt{\frac{1}{2}|\mathrm{tr}\left(\mathbf{B}_1\mathbf{B}_1^T - \mathbf{B}_2\mathbf{B}_2^T\right)|} \tag{9}$$

Note that for all three measures, smaller numbers indicate better performance. We performed our four subspace regression approaches. The test errors are shown in Table 1 (a). In this experiment, we set the subject subspace dimension $q = 4$, the state (pose) subspace dimension $r = 5$. As shown in the Table 1(a), IMI always outperforms DMS approach.

7.2 Face Tracking across Pose

In this section, we show how the person-specific subspaces built from one sample can be effectively applied to the problem of face tracking across pose.

Table 1. (a) Subspace prediction errors for pose. PA=principal angle, d_1=cosine angle, d_2=subspace distance. (b) Results of Face tracking.

(a) Pose Subspace Errors

Pose	IMI	DMS
PA	**0.4740**	0.9940
d_1	**0.2514**	0.5854
d_2	**0.5902**	0.8784

(b) Tracking RMS Errors

ATM	42.99
IVT [8]	38.41
IMI	**38.16**
DMS	40.04

We recorded videos for about 2 minutes at 25 fps rates. To compare the performance, we designed and implemented three trackers. (1) The first one is a basic template matching tracker. We maintained a template image model for the face target. The template model is updated at every frame by computing a weighted average of current template and the tracked images, hence named as Adaptive Template Matching (ATM). (2) The recent Incremental Visual Tracker (IVT) [8] is involved in this experiment. IVT updates the subspace model using the previously tracked images. IVT is essentially identical to ATM except that ATM maintains only the mean of subspace. (3) We designed a tracker system that incorporates the subspace estimated by our subspace regression approaches into the particle filtering framework. The training data used for the subspace regression are obtained from the CMU PIE data set [27], where we used 50 subjects with 12 different poses (other conditions such as illumination and expression are not considered here).

We enforced the same settings for all competing methods for fair comparison. The initial location of a face is obtained from the same face detector. For the tracking states, we used the axis-aligned bounding box representation, meaning that we kept track of two parameters: center position x and y. To provide quantitative tracking results, we manually labeled the face location for every 10th frame to form the ground-truth. The average root-mean-square (RMS) errors (in pixels) are shown in Table 1(b). Our IMI method achieves slightly better performance than IVT even though we only took into account the pose variation in the subspace learning. Besides, our approaches also avoid the computational overhead of updating the subspace at every frame since the subspace model is determined and fixed at the first frame, which is faster than IVT. Our head tracking system proceeds at 15 fps, while IVT achieves about 12 fps.

7.3 Pose Estimation

Experimental results of head pose estimation using the proposed regression method are discussed in this section. The experiments are performed on FacePix [29] database and recorded driving videos. The FacePix database includes images of faces from 30 subjects, which are taken from a wide range of precisely measure of pose angles with a granularity of 1°. Each subject has 181 images uniformly labeled from $-90°$ to $90°$. In the experiment, we only used the pose images. Some examples are shown in Fig. 3(a). The feature we extracted is PHOG

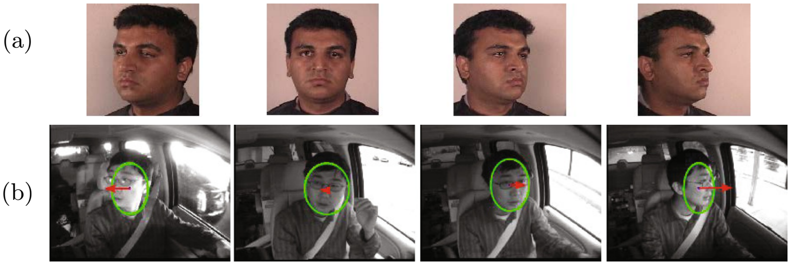

Fig. 3. The four column images (from left to right) correspond to head pose of −39°, −7°, 27°, and 47°, respectively. Row (a) shows example pose images of the first subject from the FacePix data set [29]. Row (b) shows example images of pose estimation result of driving video.

descriptor on a gray scale face image within 8 bins over three pyramid levels. The dimension of each feature for each image is 680. We randomly split the data into 25/5 training/testing subjects. The error is measured by the difference of estimated pose and ground truth pose in degrees.

We chose two classical regression methods with kernel, Ridge Regression and Support Vector Regression, as baseline systems. RBF function is selected as the kernel function for all regression methods. In this experiment, we trained 18 regressors over interval length of 30° in the pose angle space. The 18 intervals are uniformly distributed in the output space from −90° to 90°. Their overlapping between two sibling intervals is 20°. During the regression stage, we accepted only the outputs of the regressors with the values less than 10° from the corresponding center angles. Because of the overlaps, we may have multiple acceptable outputs. We simply used the averaged value of all acceptable outputs as the final prediction value. Table 2 shows the average error and standard deviation of all images of 5 testing subjects in degree. Our proposed Semantic Piecewise Regression (SP-Regression) outperforms the two classical regression methods, Ridge Regression (R-Regression) and Support Vector Regression (SV-Regression). We also tested the head pose regressors on recorded driving videos. Since it is difficult to label the ground truth of recorded video frames, we post several frames of pose estimation results in Fig. 3(b). Although our regression approach is slower than the two classical ones, we still achieved 30 fps for the head pose estimation.

Table 2. Yaw angle error of regression results on FacePix database

	R-Regression	SV-Regression	SP-Regression
Average Error	5.41°	8.20°	**4.35**°
Standard Deviation	4.08°	6.18°	**3.40**°

8 Conclusions

In this paper, we present a real-time head tracking and pose estimation system. The subspace-based head tracking approaches demonstrate the effectiveness and efficiency. we present a real-time head tracking and pose estimation system for driver alertness. The head tracking algorithm is based on subspace tracker. Instead of dynamic subspace updating, we propose two subspace learning methods to learn a subspace via pose changes after initialization, individual mapping on images and direct mapping to subspace. For the pose estimation algorithm, we propose a semantic piecewise regression which achieves better performance than classical ridge regression and support vector regression. Experimental results on standard databases and recorded videos show the effectiveness and efficiency of our system.

References

[1] Hartley, L., Horberry, T., Mabbott, N., Krueger, G.: Review of fatigue detection and prediction technologies. Tech. Rep., Institute for Research in Safty and Transport (2000)

[2] Sirovich, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. J. Opt. Soc. Am. A 4, 519–524 (1987)

[3] Jolliffe, I.T.: Principal Component Analysis. Springer, New York (1986)

[4] Turk, M., Pentland, A.: Eigenfaces for recognition. Journal Cognitive Neuroscience 3, 71–86 (1991)

[5] Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of objects using view-based representation. IJCV 26, 63–84 (1998)

[6] Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)

[7] Bischof, H., Wildenauer, H., Leonardis, A.: Illumination insensitive recognition using eigenspaces. Computer Vision and Image Understanding 1, 86–104 (2004)

[8] Ross, D., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. IJCV 77, 125–141 (2007)

[9] Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. PAMI 23, 681–685 (2001)

[10] Matthews, I., Baker, S.: Active appearance models revisited. International Journal of Computer Vision 60, 135–164 (2004)

[11] Wang, C., Li, Z.: A new face tracking algorithm based on local binary pattern and skin color information (2008)

[12] Murphy-Chutorian, E., Trivedi, M.: Head pose estimation in computer vision: A survey. PAMI 31, 607–626 (2009)

[13] Ng, J., Gong, S.: Composite support vector machines for the detection of faces across views and pose estimation. Image and Vision Computing 20, 359–368 (2002)

[14] Kruger, N., Potzsch, M., von der Malsburg, C.: Determination of face position and pose with a learned representation based on labelled graphs. Image and Vision Computing 15, 665–673 (1997)

[15] Li, S., Fu, Q., Gu, L., Scholkopf, B., Cheng, Y., Zhang, H.: Kernel machine based learning for multi-view face detection and pose estimation. In: IEEE International Conference on Computer Vision, pp. 674–679 (2001)

[16] Li, Y., Gong, S., Liddell, H.: Support vector regression and classification based multi-view face detection and recognition. In: IEEE International Conference on Automatic Face and Gesture Recognition, pp. 300–305 (2000)

[17] Raytchev, B., Yoda, I., Sakaue, K.: Head pose estimation by nonlinear manifold learning. In: IEEE International Conference on Pattern Recognition, pp. 462–466 (2004)

[18] Zhu, Y., Fujimura, K.: Head pose estimation for driver monitoring. In: IEEE Intelligent Vehicle Symposium, pp. 501–506 (2004)

[19] Anderson, T.W.: An Introduction to Multivariate Statistical Analysis, 2nd edn. Wiley, New York (1984)

[20] Drucker, H., Burges, C.J., Kaufman, L., Chris, C.J., Kaufman, B.L., Smola, A., Vapnik, V.: Support vector regression machines. In: Advances in Neural Information Processing Systems (1997)

[21] Stoica, P., Viberg, M.: Maximum likelihood parameter and rank estimation in reduced-rank multivariate linear regressions. IEEE Trans. on Sig. Proc. 44, 3069–3078 (1996)

[22] Hotelling, H.: Relations between two sets of variates. Biometrika 28, 321–377 (1936)

[23] Isard, M., Blake, A.: Contour Tracking by Stochastic Propagation of Conditional Density. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1064, pp. 343–356. Springer, Heidelberg (1996)

[24] Murphy-Chutorian, E., Doshi, A., Trivedi, M.M.: Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In: Proc. of IEEE Intelligent Transportation Systems Conference (September 2007)

[25] Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)

[26] Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR 2007: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, pp. 401–408. ACM (2007)

[27] Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression (PIE) database. IEEE Trans. on PAMI 25, 1615–1618 (2003)

[28] Wang, L., Wang, X., Feng, J.: Intrapersonal subspace analysis with application to adaptive Bayesian face recognition. Pattern Recognition 38, 617–621 (2005)

[29] Little, G., Krishna, S., Black, J., Panchanathan, S.: A methodology for evaluating robustness of face recognition algorithms with respect to variations in pose and illumination angle. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 89–92 (2005)