# Manifold Alignment for Person Independent Appearance-based Gaze Estimation

Timo Schneider, Boris Schauerte, and Rainer Stiefelhagen
Computer Vision for Human Computer Interaction Lab
Karlsruhe Institute of Technology, Germany
https://cvhci.anthropomatik.kit.edu

***Abstract*—We show that dually supervised manifold embedding can improve the performance of machine learning based person-independent and thus calibration-free gaze estimation. For this purpose, we perform a manifold embedding for each person in the training dataset and then learn a linear transformation that aligns the individual, person-dependent manifolds. We evaluate the effect of manifold alignment on the recently presented Columbia dataset, where we analyze the influence on 6 regression methods and 8 feature variants. Using manifold alignment, we are able to improve the person-independent gaze estimation performance by up to 31.2 % compared to the best approach without manifold alignment.**

## I. Introduction

Being able to determine where people look is one of the most important human abilities. It is essential for every infant's early (social) development (e.g., gaze following) and later everyday human-human interaction (e.g., mutual gaze and gaze aversion). Consequently, computer vision algorithms that estimate where people look are an important, enabling technology for natural human-computer and human-robot interaction or assistive technologies that aid in everyday communication. Apart from these human-centric applications, there are numerous other application areas such as, e.g., design and advertisement. Nowadays, there exists a number of solutions in the market (e.g., Tobii trackers [1]), but they typically rely on active sensing technologies (most commonly, infrared illuminators) or require manual calibration for every user.

We investigate how manifold alignment can assist passive, person-independent gaze estimation based on machine learning. Moreover, person independence also allows to omit a calibration procedure for new users. During manifold alignment, we first perform a dimensionality reduction for each person in the training dataset. Subsequently, we use the gaze directions and person identities to align the resulting manifolds across persons. Here, our approach is linear and results in an additional rotation of the Principal Component Analysis (PCA) results, which makes it computationally very efficient. We refer to a specific combination of a regression algorithm with a feature as a *configuration*. We show that manifold alignment can substantially improve the results, especially if the number of target dimensions is low. Compared to the best configuration without alignment, we achieve a statistically significant, relative performance improvement of $31.2\,\%$ with a combination of multi-level Histograms of Oriented Gradients (multi-level HOG or mHOG) features, a reduction to 16 dimensions, and nearest neighbors for regression. To validate our findings, we use statistical tests to ensure that the observed performance differences are in fact statistically significant. This way, we show that manifold alignment statistically significantly improves the performance of many configurations that consist of raw pixels, Discrete Cosine Transform (DCT), Local Binary Patterns (LBP), or Histograms of Oriented Gradients (HOG) as features and Gaussian Processes Regression (GPR), $k$ Nearest Neighbors (kNN), Regression Trees (Regtrees), Support Vector Regression (SVR), Relevance Vector Regression (RVR), or Spline Regression (Splines) for regression; especially, the further we reduce the number of dimensions.

## II. Related Work

Gaze estimation approaches are commonly divided into *feature-based* and *appearance-based* methods (see [2]). While feature-based approaches rely on the corneal reflection of infrared light, the pupil center (e.g., [3]), or iris contour (e.g., [4]), they often require high-resolution images.

For appearance-based approaches, eye images are treated as points in a high-dimensional space. Hence, there is no need to extract small dimensional features, allowing for promising results on consumer webcams [5]. Subsequently, the high-dimensional feature vectors are mapped onto 2D gaze directions, using a broad range of regression methods. Therefor, Baluja and Pomerleau [6] and Stiefelhagen et al. [7] propose an Artificial Neural Network, whose training, however, requires thousands of training samples. Tan et al. [8] construct an appearance manifold of training samples, specifying neighborhoods by label distances. The label of a new sample is induced by linear combination of the respective $k$ nearest neighbors. Williams et al. [9] propose sparse and semi-supervised Gaussian Process Regression ($S^3$GP), to account for unlabeled training data in order to reduce the calibration time and effort. Martinez et al. [10] calculate multi-level HOG features and compare RVR and SVR for regression. A further approach to deal with sparsely collected training samples is Adaptive Linear Regression (ALR) as presented by Lu et al. [5]. Lu et al. propose the low-dimensional *pixelsum* feature (the sum of pixel intensities in $3 \times 5$ blocks) and interpolate a new sample by the fewest training samples that best keep the linear combination. Unfortunately, especially with the high number of training samples as in our evaluation, ALR has a high computational complexity.

Despite the success of appearance-based methods, truly unconstrained gaze estimation still faces two key challenges: head motion and person independence, i.e. the lack of strong constraints and calibration. For introducing head pose, different approaches have been proposed such as, for example,

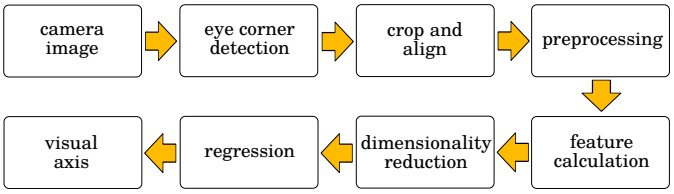



Fig. 1: Processing pipeline for appearance-based gaze estimation.

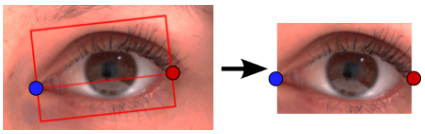


Fig. 2: Cropping and aligning eye patches to $60 \times 40$ images. The eye corners are mapped onto the horizontally aligned points $[2\ 25]$ and $[59\ 25]$, leaving more space for the upper eye part.

incrementally gathering a huge amount of training data [11], subsequent compensation of estimation biases [12], synthesizing data for unseen head poses [13], or backprojection of the face texture to a frontal model based on a RGB-D camera [14].

In this work, we focus on a frontal head pose. Existing approaches are mostly evaluated by the gaze estimation error per person, i.e. these methods require an active calibration step. However, many gaze applications certainly lack the possibility of manual calibration or user cooperation. Therefor, Noris et al. [15] use a head-mounted camera for studies of childrens' visual behavior in unconstrained environments. They apply a PCA to embed the retrieved eye images to 24 dimensions, then map the low-dimensional features to gaze coordinates using SVR and GPR, respectively, utilizing the training samples of 22 different persons. Furthermore, Mora et al. [16] extend their earlier head pose free approach [14] by person independence. They automatically choose the trained models of two persons that are most similar to the test person, i.e. that best reconstruct the test samples, and subsequently use ALR to map them to gaze directions.

In matters of person independence, head pose estimation faces similar challenges as gaze estimation. To determine an unknown – i.e., not part of the training data – person's head orientation is challenging, because the change in appearance due to a person's individual face characteristics is can be bigger than the change due to pose. There has been considerable research on this topic, especially in the field of *manifold embedding* and *manifold alignment*. Balasubramanian et al. [17] embed head images to a low-dimensional representation, by biasing the distance of samples with the corresponding poses. This ensures that images with similar labels are also closer together on the resulting manifold. Yan et al. [18] therefor propose Synchronized Submanifold Embedding (SSE). They utilize pose and identity to build individual submanifolds as a set of simplices. Afterward, these submanifolds are aligned by minimizing pose distances between the simplices of different persons and maximizing intra-submanifold variances. We transfer this concept to gaze estimation (see Sec. III-B) and adapt SSE for our purpose. In Yan et al.'s approach, due to the labels being 3D head poses, the label space is divided into tetrahedrons (3-simplices). These simplices are found by connecting each sample to the three corresponding nearest neighbors within the same submanifold. We instead have 2D gaze labels and find the simplices in label space, similarly to Tan et al. [8], by applying a Delaunay triangulation, which additionally guarantees the manifold's convexity. Furthermore, Yan et al. add an indicator function to the minimization term to omit synchronizing point pairs that are too far apart. For us, this would be the case if the sample of person $p$ could not be reconstructed in the manifold of person $p'$, i.e., it lies outside the manifolds's convex hull. However, since in our evaluation scenario the manifolds overlap, the border samples of person $p'$ in turn can be reconstructed within the manifold of person $p$, leading to a correct alignment at the manifold borders.

## III. Method

### A. Gaze Estimation Pipeline

*1) Overview:* The typical steps that are necessary to perform gaze estimation based on machine learning techniques are shown in Fig. 1 (cf. [10]). First, we have to extract the image patch that depicts the eye based on the eye corner detection. For each eye, we extract a $60 \times 40$ (width $\times$ height) image patch. Each patch is aligned by projecting the detected inner corner location onto point $[2\ 25]$ and the outer corner onto $[59\ 25]$ (see Fig. 2). Then, we perform a histogram equalization as a preprocessing step. On the basis of the normalized image patch, we calculate the image features. We do this for each eye independently and subsequently concatenate both feature vectors into a single vector that describes both eyes. The feature vector then serves as input to the regression algorithm, which finally predicts—or learns to predict—the gaze direction.

*2) Features:* Several features have been used to predict the gaze direction. We consider the following features: The *raw* feature is the vector of the eye patch's pixel intensities. Lu et al. [5] propose the 15-dimensional *pixelsum* feature, i.e., the sum of pixels in $5 \times 3$ blocks on the image. The DCT feature has been successfully used for, e.g., face recognition [19]. First the image is divided into cells, then the DCT coefficients for each cell are concatenated in zig-zag order. The LBP operator is a common texture descriptor [20] that compares each pixel to its 8 neighbors and encodes the binary results in one byte. Here, the image is divided into cells, then histograms of LBP features per cell are concatenated. Finally, HOG [21] uses histograms of gradient orientations, concatenated over different image blocks. Each block is divided into a grid of cells, in which the gradients are counted and normalized by the remaining cells in the same block. We use signed gradients on uniformly spaced blocks, each consisting of $2 \times 2$ cells. Furthermore, HOG features with differently subdivided blocks can be concatenated to form the multi-level HOG feature. We use signed gradients, $2 \times 2$ cells per block, and $1 \times 2$, $3 \times 1$, $3 \times 2$ and $6 \times 4$ blocks, referring to Martinez et al. [10], who optimized the block combinations for the task of gaze estimation.

*3) Regression Algorithms:* Naturally, a wide range of algorithms can be used for appearance-based gaze estimation as well. We consider the following regression methods: kNN for regression [22] linearly weights the labels of the $k$ nearest neighbors of a new sample (we use the inverse distance) to derive the new sample's label. Multivariate adaptive re-

gression splines [23] learn a model of the weighted sum of basis functions, while the number of basis functions and the corresponding parameters are automatically determined using the available data. A Regression tree [24] is a decision tree with binary splits and respective training samples at each leaf. A new sample is then propagated through the tree and the label is induced by the labels of the resulting leaf (e.g., the mean label). GPR [25] uses Gaussian processes to predict a probability distribution for a new sample, providing a mean value and variance. SVR [26] non-linearly maps the data to a high-dimensional feature space (we use a polynomial kernel) and subsequently performs linear regression in this space. The related RVR [27] uses Bayesian inference to find the weights for regression, avoiding the manual setting of the SVR's hyperparameters. Furthermore, the found solution is relatively sparse and therefore has a good generalization performance.

### B. Synchronized Delaunay Submanifold Embedding

Our approach, Synchronized Delaunay Submanifold Embedding (SDSE), is like SSE a dually supervised embedding method, i.e., it uses label and identity information. In our case, the labels are the 2D gaze directions (i.e., azimuth and elevation angle), whereas the person defines the identity. The training samples from person $p$ are given as $X_p = (x_1^p, x_2^p, \ldots, x_{n_p}^p)$, where $n_p$ is the number of training samples for person $p$ and features $x_i^p \in \mathbb{R}^m$. The corresponding gaze labels are $\Theta_p = (\theta_1^p, \theta_2^p, \ldots, \theta_{n_p}^p)$, where $\theta_i^p \in \mathbb{R}^2$. The overall number of training samples is $N = \sum_{p=1}^{P} n_p$, where $P$ is the number of persons in the training set.

First, we build an individual submanifold for each person. Since the available gaze directions of different persons in general are not the same, the manifolds of different persons cannot be directly aligned. Therefor, we construct a simplicial 2-complex of samples from person $p$, by applying a Delaunay triangulation to the corresponding gaze labels $\Theta_p$. A $k$-simplex is the convex hull of $k+1$ affinely independent points in a space of $k$ dimensions or higher. For example, a 2-simplex is a triangle connecting three points in 2D. A homogeneous *simplicial $k$-complex* $S$ is a set of $k$-simplices, that satisfy the conditions that any face of a simplex is again in $S$ and the intersection of any two simplices is a face of these simplices. In other words, the $k-$simplices have to be smoothly connected. Thus, instead of using the Euclidean distances between feature vectors $x_i^p$, we find the corresponding sample neighbors by gaze label distances. This way, we obtain continuous submanifolds that are fully defined inside the convex hull of $\Theta_p$, which is guaranteed by the Delaunay triangulation.

Now, arbitrary points on a submanifold can be reconstructed using locally linear interpolation. Let $T_p$ be the total number of simplices of person $p$, and $t^i = (t_1^i, t_2^i, \ldots, t_{(k+1)}^i)$, $i = 1, \ldots, T_p$, contain the $k+1$ sample indices of the $i$-th $k$-simplex of person $p$. Assuming local linearity, we can interpolate a point $x_{\text{new}}$ inside the simplex $(x_{t_1^i}^p, x_{t_2^i}^p, \ldots, x_{t_{(k+1)}^i}^p)$ with its surrounding neighbors

$$x_{\text{new}} = \sum_{j=1}^{k+1} w_j x_{t_j^i}^p \ . \tag{1}$$

The interpolated label $\theta_{\text{new}}$ is defined by weighting the neighbors' labels in the same way

$$\theta_{\text{new}} = \sum_{j=1}^{k+1} w_j \theta_{t_j^i}^p \ . \tag{2}$$

For each feature $x_i^p$ from person $p$, the reconstructed equivalent point on the manifold from person $p'$ is calculated using the corresponding simplex and weights

$$(\tilde{w}, \tilde{t}) = \arg\min_{w,t} \left\| \theta_i^p - \sum_{j=0}^{k+1} w_j \theta_{t_j}^{p'} \right\|^2 \tag{3}$$

so that the corresponding feature is reconstructed as

$$y(x_i^p, p') = \sum_{j=1}^{k+1} \tilde{w}_j x_{\tilde{t}_j}^{p'} \ . \tag{4}$$

For our purpose, i.e., for each label of person $p$, we find the corresponding triangle from person $p'$, calculate the barycentric coordinates as weights, and finally interpolate a new feature for $p'$. This way, we get samples with same labels on each submanifold, which subsequently can be aligned in several ways. One possible approach is a linear transformation that uses a projection matrix $W \in \mathbb{R}^{m \times d}$ with $d \ll m$ so that

$$y_i = W^T x_i \quad , \tag{5}$$

where $y_i$ is the $d$-dimensional mapping of feature $x_i$. Here, the projection matrix $W$ can be chosen to minimize the distances between samples with same label of different persons

$$S_{\text{syn}}(W) = \sum_{p=1}^{P} \sum_{i=1}^{n_p} \sum_{p' \neq p} \left\| W^T x_i^p - W^T y(x_i^p, p') \right\|^2 \ . \tag{6}$$

However, we want to enhance the separability of different labels, and also avoid the trivial solution to project each sample to the same point. Consequently, $W$ should additionally maximize the distances between different sample pairs of one person

$$S_{\text{sep}}(W) = \sum_{p=1}^{P} \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \left\| W^T x_i^p - W^T x_j^p \right\|^2 \ . \tag{7}$$

To simultaneously maximize $S_{\text{sep}}$ and minimize $S_{\text{syn}}$, we derive $W$ as

$$W = \arg\max_W \frac{S_{\text{sep}}(W)}{S_{\text{syn}}(W)} = \arg\max_W \frac{\text{Tr}(W^T S_1 W)}{\text{Tr}(W^T S_2 W)} \tag{8}$$

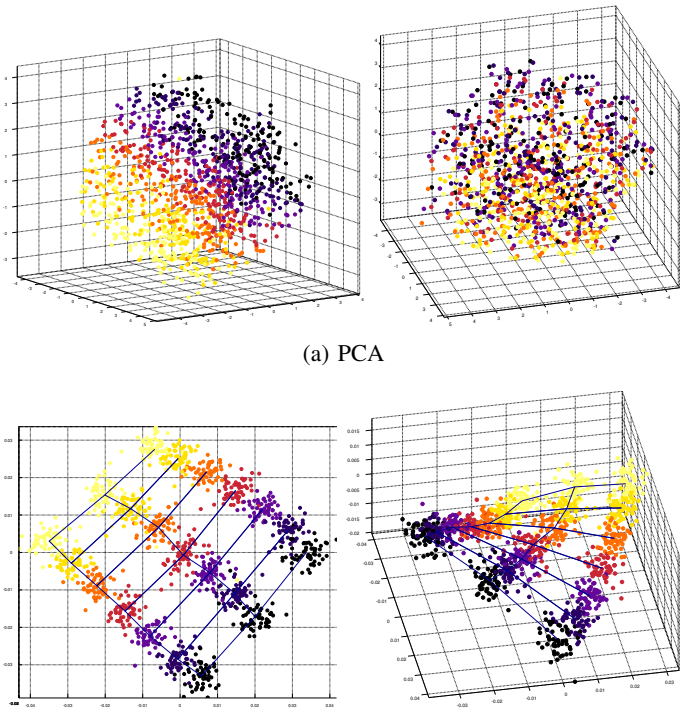
with

$$S_1 = \sum_{p=1}^{P} \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} \left(x_i^p - x_j^p\right) \left(x_i^p - x_j^p\right)^T \tag{9}$$

$$S_2 = \sum_{p=1}^{P} \sum_{i=1}^{n_p} \sum_{p' \neq p} \left(x_i^p - y(x_i^p, p')\right) \left(x_i^p - y(x_i^p, p')\right)^T \ . \tag{10}$$

This can be transformed into the form

$$W = \arg\max_W \ \text{Tr}\left[(W^T S_2 W)^{-1} (W^T S_1 W)\right] \tag{11}$$



(a) PCA

(b) SDSE

Fig. 3: Illustration of the (a) PCA and (b) SDSE embeddings that show samples of 56 people looking in $7 \times 3$ gaze directions, reduced to 3 dimensions. The embeddings are based on mHOG+LBP as feature vectors, because it is the basis for the best results in our evaluation (see Tab. II and Tab. III). Each embedded feature vector is color-coded by its corresponding yaw (i.e., pan or azimuth) label. For SDSE we display a grid between the mean embedded points for each label, which clearly corresponds to the $7 \times 3$ gaze directions.

which we solve with the generalized eigenvalue decomposition method, i.e.,

$$S_1 \cdot W = S_2 \cdot W \cdot L \ , \tag{12}$$

where the columns of $W$ are eigenvectors, and $L$ is a diagonal matrix with the corresponding eigenvalues in descending order as diagonal elements.

## IV. Evaluation

### A. Dataset

The Columbia gaze dataset was published recently by Smith et al. [28]. It contains high-resolution images of 56 subjects that look at 21 ($7 \times 3$) gaze points for a frontal head pose. We chose the Columbia dataset for our evaluation, because it provides a reasonable amount of subjects and gaze directions. Although the Gi4E dataset [29] contains nearly twice the amount of subjects, we decided to not evaluate on it, because it only provides samples for 11 points on a $4 \times 3$ grid, which introduces a considerable bias.

We rely on the automatic eye corner detections that were provided to us by Smith et al. [28]. To detect the eye corners, Smith et al. [28] used a commercial system by Omron [30].

### B. Procedure and Measures

We follow an 8-fold cross-validation evaluation procedure, which means that for every fold we have 7 test persons and 49 training persons. Since the persons in the Columbia dataset are not organized according to any particular feature or criteria, we simply use the persons' numerical identifiers to create the folds. Accordingly, the training persons of the $k$-th fold are identified by $P_k = \{[7(k-1)+1], \ldots, 7k\}$.

The gaze directions in the Columbia dataset are given as yaw and pitch angles. Accordingly, since we estimate the yaw and pitch angle, we can calculate the angular yaw and pitch estimation error. To derive a single evaluation criterion, we calculate the joint angular error which is the norm of the yaw and pitch error. In the following, we focus on the joint angular error as our target evaluation measure and perform all quantitative comparisons—including statistical tests—with respect to the joint angular error (reported in $^\circ$).

We perform statistical significance tests to determine whether or not the performance differences are significant. Therefor, we record each algorithms prediction for every image and use the joint error as input data for the statistical tests. We rely on three pairwise, two-sample t-tests to categorize the results: First, we perform a two-tailed test to check whether the compared errors come from distributions with different means (i.e., $\mathcal{H}_=$: "means are equal"). Analogously, second, we perform a left-tailed test to check whether an algorithm's error distribution's mode is greater (i.e., $\mathcal{H}_>$: "mean is greater") and, third, a right-tailed test to check whether an algorithm's error distribution's mode is lower (i.e., $\mathcal{H}_<$: "mean is lower"). All tests are performed at a confidence level of $95\%$, i.e., $\alpha = 5\%$.

To simplify the presentation and discussion, we group the test results into five classes: "better" means that the hypotheses of equal and worse mean error were rejected. "better or equal" means that only the hypothesis of a worse mean error could be rejected. "probably equal" means that no hypothesis could be rejected. "equal or lower" means that the hypothesis of a better mean error was rejected. "lower" means that the hypotheses of equal and better mean error were rejected.

### C. Results

On its own, the (multi-level) HOG feature leads to the best performance compared to pixelsums, DCT, the raw image patch as well as LBP alone, see Tab. I. This supports Martinez et al.'s [10] approach, who demonstrated that the combination of SVR or RVR and mHOG leads to a very good performance. However, in our case, RVR performs worse than SVR and, if we include LBP in the feature vector, SVR even provides the best performance without manifold alignment reduction.

In general, as can be seen in Tab. I, II and III, we achieve the best results with a combination of (multi-level) HOG and LBP as basis features. This comes at no surprise, because in recent years the combination of HOG and LBP has been reported as good choice of appearance-based feature for various applications such as, for example, large scale image classification [31] or pedestrian detection [32]. This is most likely caused by the fact that HOG and LBP are complementary features where HOG focuses on shape information while LBP focuses on texture information. We would like to note that in our experience HOG alone performs well, if the image patches of the eye are almost perfectly aligned. However, in case of noisy eye corner detections the performance improves considerably when incorporating LBP features.

The quantitative evaluation results without SDSE are

| ID | raw | pixelsum* | DCT | LBP | HOG | HOG+LBP | mHOG | mHOG+LBP |
|---|---|---|---|---|---|---|---|---|
| **GPR** | 5.59 | 5.20 | 5.71 | 4.59 | 4.13 | 4.23 | 4.19 | 4.12 |
| **kNN** | 4.96 | 5.04 | 5.13 | 5.33 | 4.75 | 4.56 | 4.56 | 4.55 |
| **Regtree** | 5.98 | 5.86 | 6.99 | 6.86 | 5.50 | 6.14 | 5.82 | 5.78 |
| **RVR** | 10.90 | 10.90 | 6.24 | 5.92 | 24.70 | 5.59 | 4.19 | 4.03 |
| **Splines** | 5.41 | 5.01 | 5.84 | 5.11 | 5.00 | 4.87 | 4.81 | 4.69 |
| **SVR** | 4.71 | 17.00 | 4.79 | 4.87 | 4.60 | 4.28 | 3.55 | **3.53** |

TABLE I: Joint error (in °) of different configurations. (*): Since the pixelsum feature only has 15 dimensions, we do not perform dimensionality reduction and only provide the result as a further baseline.

| SDSE256 | raw | DCT | LBP | HOG | HOG+LBP | mHOG | mHOG+LBP |
|---|---|---|---|---|---|---|---|
| **GPR** | 5.34 (4%) | 5.62 (2%) | 4.52 (2%) | 4.85 (-17%) | 3.97 (6%) | 3.76 (10%) | 3.74 (9%) |
| **kNN** | 5.35 (-8%) | 5.30 (-3%) | 4.81 (10%) | 4.56 (4%) | 3.89 (15%) | 3.52 (23%) | 3.49 (23%) |
| **Regtree** | 4.62 (23%) | 4.80 (31%) | 3.84 (44%) | 4.62 (16%) | **2.92 (52%)** | 3.86 (34%) | 3.97 (31%) |
| **Splines** | 4.68 (13%) | 4.97 (15%) | 3.86 (24%) | 4.53 (9%) | 3.29 (32%) | 3.87 (20%) | 3.85 (18%) |
| **RVR** | 5.58 (49%) | 6.05 (3%) | 4.59 (22%) | 5.29 (79%) | 4.18 (25%) | 3.98 (5%) | 3.96 (2%) |
| **SVR** | 6.53 (-39%) | 6.35 (-33%) | 7.49 (-54%) | 5.01 (-9%) | 5.19 (-21%) | 4.02 (-13%) | 4.00 (-13%) |

| SDSE64 | raw | DCT | LBP | HOG | HOG+LBP | mHOG | mHOG+LBP |
|---|---|---|---|---|---|---|---|
| **GPR** | 5.19 (7%) | 5.45 (5%) | 4.29 (7%) | 4.83 (-17%) | 3.66 (14%) | 3.56 (15%) | 3.54 (14%) |
| **kNN** | 4.67 (6%) | 4.95 (4%) | 4.21 (21%) | 3.86 (19%) | 3.17 (30%) | 2.94 (36%) | 2.90 (36%) |
| **Regtree** | 4.61 (23%) | 4.84 (31%) | 3.91 (43%) | 4.55 (17%) | **2.86 (53%)** | 3.87 (34%) | 3.93 (32%) |
| **Splines** | 4.69 (13%) | 5.02 (14%) | 3.87 (24%) | 4.54 (9%) | 3.29 (33%) | 3.88 (19%) | 3.82 (19%) |
| **RVR** | 5.59 (48%) | 6.08 (3%) | 4.61 (22%) | 5.29 (79%) | 4.19 (25%) | 3.98 (5%) | 3.96 (2%) |
| **SVR** | 6.54 (-39%) | 6.36 (-33%) | 7.49 (-54%) | 5.02 (-9%) | 5.21 (-22%) | 4.01 (-13%) | 4.00 (-13%) |

| SDSE16 | raw | DCT | LBP | HOG | HOG+LBP | mHOG | mHOG+LBP |
|---|---|---|---|---|---|---|---|
| **GPR** | 4.74 (15%) | 5.15 (10%) | 4.06 (12%) | 4.61 (-12%) | 3.40 (20%) | 3.27 (22%) | 3.24 (21%) |
| **kNN** | 4.40 (11%) | 4.81 (6%) | 3.77 (29%) | 3.64 (23%) | 2.84 (38%) | 2.71 (41%) | **2.69 (41%)** |
| **Regtree** | 4.61 (23%) | 4.85 (31%) | 3.87 (44%) | 4.38 (20%) | 2.89 (53%) | 3.81 (35%) | 3.88 (33%) |
| **Splines** | 4.67 (14%) | 4.99 (15%) | 3.85 (25%) | 4.39 (12%) | 3.25 (33%) | 3.88 (19%) | 3.84 (18%) |
| **RVR** | 5.60 (48%) | 6.08 (3%) | 5.60 (5%) | 5.31 (79%) | 4.19 (25%) | 3.99 (5%) | 3.96 (2%) |
| **SVR** | 6.54 (-39%) | 6.36 (-33%) | 7.49 (-54%) | 5.03 (-9%) | 5.21 (-22%) | 4.02 (-13%) | 4.00 (-13%) |

TABLE II: Joint error (in °) using SDSE to reduce the dimensions to 256, 64, and 16, respectively, in comparison with the joint error of Tab. I. Cell colors represent the five t-test classes, see Sec. IV-B: better, better or equal, probably equal, equal or lower, and lower . This table is best viewed in color.

shown in Tab. I and the results with SDSE are presented in Tab. II. Furthermore, since SDSE represents a linear transformation of the PCA features, in Tab. III we present the results that were achieved with PCA (without applying SDSE) to serve as further baseline (Fig. 3). As can be seen in Tab. II, compared to the results achieved without dimensionality reduction (see Tab. I) SDSE leads to a substantial performance benefit for nearly all configurations, i.e., algorithm and feature combinations. Here, SVR is a clear exception and, interestingly, the highly related RVR does not exhibit the same problems. The cause for this phenomenon could either lie in the implementations[1], the choice or number of support vectors, or that the optimal SVR parameters differ substantially after an SDSE has been applied to the features. Even for cases where the statistical tests do not indicate that SDSE is clearly better, the performance benefits alone are still considerable (e.g., Tab. II, 16 dimensions, DCT+kNN $+6\,\%$ or mHOG+RVR $+5\,\%$). Interestingly, the results for SDSE improve as we lower the number of target dimensions, which in many cases differs from the behavior of PCA. As is illustrated by the cell coloring of Tab. III, as we lower the number of target dimensions, the performance of PCA compared to SDSE becomes either worse or equal in the majority of cases. Again, we would like to note that even in cases where the tests do not indicate that SDSE is clearly better, the plain joint error differences are often still considerable, i.e., PCA alone is nonetheless inferior (see, e.g., Tab. III, 16 dimensions, DCT+kNN $-6\,\%$ or Splines+HOG $-6\,\%$).

[1]We rely on libSVM [33] to implement SVR and the Pattern Recognition Toolbox for Matlab (PRT) [34] to implement RVR. Please note that we issued a bug report to New Folder Consulting, i.e., the company behind PRT, because the occasionally occurring problems of RVR (see, e.g., Tab. III, 16 dimensions, RVR+LBP) are caused by PRT's RVR implementation and behavior.

## V. Conclusion

We have shown that SDSE can substantially improve the performance of machine learning based person-independent and thereby calibration-free gaze estimation. To support this statement, we evaluated the influence that SDSE has on 6 regression algorithms in combination with 8 feature vector variations. This way, we were able to show that SDSE leads to statistically significant performance improvements in the majority of regression/feature combinations. Interestingly, in contrast to PCA, the joint gaze estimation error based on SDSE transformed features improves as we reduce the number of target dimensions. We achieved the best results with a target dimensionality of 16 elements. Furthermore, we have shown that the combination of HOG and LBP features lead to considerable performance improvements compared to LBP or even HOG alone. In short, SDSE was able to reduce the person-independent gaze estimation error by up to $31.2\,\%$ compared to the best feature/regression combination without SDSE.

As part of our future work, we want to investigate how SDSE can be applied in head pose free scenarios.


## Acknowledgment

We would like to thank Brian A. Smith for sharing the Omron eye corner detections with us. The work presented in this paper was supported by the Quaero Programme, funded by OSEO, French State agency for innovation.

| PCA256 | raw | | DCT | | LBP | | HOG | | HOG+LBP | | mHOG | | mHOG+LBP | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **GPR** | 5.58 | (-4%) | 5.19 | (8%) | 4.34 | (4%) | 4.06 | (16%) | 3.88 | (2%) | 3.70 | (1%) | 3.73 | (0%) |
| **kNN** | 4.91 | (8%) | 5.10 | (4%) | 5.29 | (-10%) | 4.76 | (-4%) | 4.58 | (-18%) | 4.58 | (-30%) | 4.58 | (-31%) |
| **Regtree** | 6.07 | (-31%) | 5.89 | (-23%) | 6.48 | (-69%) | 6.79 | (-47%) | 6.08 | (-108%) | 5.89 | (-53%) | 5.64 | (-42%) |
| **Splines** | 5.12 | (-9%) | 5.37 | (-8%) | 4.82 | (-25%) | 4.79 | (-6%) | 4.27 | (-30%) | 4.29 | (-11%) | 4.16 | (-8%) |
| **RVR** | 10.86 | (-95%) | 4.93 | (19%) | 4.59 | (0%) | 56.71 | (-972%) | 3.94 | (6%) | 3.68 | (8%) | **3.67** | **(7%)** |
| **SVR** | 5.64 | (14%) | 5.22 | (18%) | 4.30 | (43%) | 4.10 | (18%) | 3.88 | (25%) | 5.13 | (-28%) | 5.14 | (-29%) |

| PCA64 | raw | | DCT | | LBP | | HOG | | HOG+LBP | | mHOG | | mHOG+LBP | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **GPR** | 5.11 | (2%) | 4.68 | (14%) | 4.29 | (-0%) | 4.04 | (16%) | 3.89 | (-6%) | 3.78 | (-6%) | **3.77** | **(-7%)** |
| **kNN** | 4.95 | (-6%) | 5.05 | (-2%) | 5.18 | (-23%) | 4.74 | (-23%) | 4.51 | (-42%) | 4.58 | (-56%) | 4.60 | (-58%) |
| **Regtree** | 5.73 | (-24%) | 5.45 | (-12%) | 6.23 | (-59%) | 6.27 | (-38%) | 5.77 | (-101%) | 5.31 | (-37%) | 5.30 | (-35%) |
| **Splines** | 5.07 | (-8%) | 5.35 | (-7%) | 4.88 | (-26%) | 4.81 | (-6%) | 4.34 | (-32%) | 4.22 | (-9%) | 4.13 | (-8%) |
| **RVR** | 10.86 | (-94%) | 5.03 | (17%) | 4.93 | (-7%) | 27.41 | (-418%) | 4.68 | (-12%) | 4.23 | (-6%) | 4.21 | (-6%) |
| **SVR** | 5.88 | (10%) | 4.95 | (22%) | 4.37 | (42%) | 4.27 | (15%) | 4.05 | (22%) | 4.76 | (-19%) | 4.77 | (-19%) |

| PCA16 | raw | | DCT | | LBP | | HOG | | HOG+LBP | | mHOG | | mHOG+LBP | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **GPR** | 5.02 | (-6%) | 5.14 | (0%) | 4.72 | (-16%) | 4.75 | (-3%) | 4.31 | (-27%) | 4.17 | (-28%) | **4.17** | **(-29%)** |
| **kNN** | 4.84 | (-10%) | 5.10 | (-6%) | 5.52 | (-46%) | 4.81 | (-32%) | 4.69 | (-65%) | 4.40 | (-63%) | 4.38 | (-63%) |
| **Regtree** | 5.66 | (-23%) | 5.36 | (-10%) | 5.95 | (-54%) | 5.53 | (-26%) | 5.35 | (-85%) | 4.97 | (-31%) | 4.83 | (-25%) |
| **Splines** | 5.00 | (-7%) | 5.04 | (-1%) | 4.91 | (-27%) | 4.66 | (-6%) | 4.46 | (-37%) | 4.18 | (-8%) | 4.18 | (-9%) |
| **RVR** | 10.86 | (-94%) | 6.30 | (-4%) | 33.94 | (-506%) | 47.52 | (-795%) | 263.13 | (-6178%) | 4.67 | (-17%) | 4.68 | (-18%) |
| **SVR** | 16.11 | (-146%) | 9.20 | (-45%) | 4.80 | (36%) | 4.98 | (1%) | 4.56 | (12%) | 18.19 | (-353%) | 19.98 | (-400%) |

TABLE III: Joint error (in °) using PCA in comparison to the respective SDSE embedding from Tab. II. Cell colors represent the five t-test classes, see Sec. IV-B: better, better or equal, probably equal, equal or lower, and lower . This table is best viewed in color.

## REFERENCES


[1] Tobii Technology, www.tobii.com.

[2] D. W. Hansen and Q. Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze," *Trans. PAMI*, vol. 32, pp. 478–500, 2010.

[3] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation." *IEEE Transactions on Image Processing*, vol. 21, pp. 802–815, 2012.

[4] M. Reale, T. Hung, and L. Yin, "Viewing direction estimation based on 3D eyeball construction for HRI," in *CVPR Workshops*, 2010.

[5] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Inferring human gaze from appearance via adaptive linear regression," in *ICCV*, 2011.

[6] S. Baluja and D. Pomerleau, "Non-Intrusive Gaze Tracking Using Artificial Neural Networks." in *Advances in Neural Information Processing Systems*, 1994.

[7] R. Stiefelhagen, J. Yang, and A. Waibel, "Tracking Eyes and Monitoring Eye Gaze," in *Workshop on Perceptual User Interfaces*, 1997.

[8] K.-H. Tan, D. J. Kriegman, and N. Ahuja, "Appearance-based eye gaze estimation," in *IEEE Workshop on Applications of Computer Vision*, 2002.

[9] O. Williams, A. Blake, and R. Cipolla, "Sparse and Semi-supervised Visual Mapping with the S^3GP," in *CVPR*, 2006.

[10] F. Martinez, A. Carbone, and E. Pissaloux, "Gaze estimation using local features and non-linear regression," in *ICIP*, 2012.

[11] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, "An Incremental Learning Method for Unconstrained Gaze Estimation," in *ECCV*, 2008.

[12] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, "A Head Pose-free Approach for Appearance-based Gaze Estimation," *BMVC*, 2011.

[13] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Head pose-free appearance-based gaze sensing via eye image synthesis," in *ICPR*, 2012.

[14] K. A. F. Mora and J.-M. Odobez, "Gaze Estimation from Multimodal Kinect Data," in *CVPR Workshops*, 2012.

[15] B. Noris, K. Benmachiche, and A. Billard, "Calibration-Free Eye Gaze Direction Detection with Gaussian Processes," in *International Conference on Computer Vision Theory and Applications*, 2008.

[16] K. A. F. Mora and J.-M. Odobez, "Person indepedent 3D gaze estimation from remote RGB-D cameras," in *ICIP*, 2013.

[17] V. N. Balasubramanian, J. Ye, and S. Panchanathan, "Biased Manifold Embedding: A Framework for Person-Independent Head Pose Estimation," in *CVPR*, 2007.

[18] S. Yan, H. Wang, Y. Fu, X. Yan, and T. S. Huang, "Synchronized submanifold embedding for person-independent pose estimation and beyond." *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 202–210, 2009.

[19] H. K. Ekenel and R. Stiefelhagen, "Local Appearance based Face Recognition Using Discrete Cosine Transform," in *European Signal Processing Conference*, 2005.

[20] D.-C. He and L. Wang, "Texture Unit, Texture Spectrum And Texture Analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, no. 4, pp. 509–512, 1990.

[21] N. Dalal and W. Triggs, "Histograms of Oriented Gradients for Human Detection," in *CVPR*, 2004.

[22] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, vol. 46, pp. 175–185, 1992.

[23] J. H. Friedman, "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, vol. 19, pp. 1–67, 1991.

[24] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, 1984, vol. 19.

[25] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, 2006, vol. 14.

[26] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199–222, 2004.

[27] M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[28] B. A. Smith, Q. Yin, S. K. Feiner, and S. K. Nayar, "Gaze Locking: Passive Eye Contact Detection for Human-Object Interaction," in *ACM Symposium on User Interface Software and Technology*, 2013.

[29] V. Ponz, A. Villanueva, L. Sesma, M. Ariz, and R. Cabeza, "Topography-Based Detection of the Iris Centre Using Multiple-Resolution Images," *International Machine Vision and Image Processing Conference*, 2011.

[30] Omron, "OKAO Vision," www.omron.com/r_d/coretech/vision/okao.html.

[31] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: Fast feature extraction and svm training," in *CVPR*, 2011.

[32] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *ICCV*, 2009.

[33] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–39, 2011.

[34] New Folder Consulting, "Pattern Recognition Toolbox for Matlab," http://newfolder.github.io.