Hierarchical Sliding Slice Regression for Vehicle Viewing Angle Estimation

Dan Yang, Yanlin Qian, Ke Chen, Eleni Berki, *Member, IEEE* and Joni-Kristian Kämäräinen, *Member, IEEE*

***Abstract*—We propose a novel Hierarchical Sliding Slice Regression (HSSR) which in a coarse-to-fine manner represents global circular target space with a number of ordinally localised and overlapping subspaces. Our method is particularly suitable for visual regression problems where the regression target is circular (e.g., car viewing angle) and visual similarity inconsistent over the target space (e.g., repetitive appearance). A good application example is the camera based car viewing angle estimation problem, where visual similarity of different views is highly inconsistent - front and back views and left and right side views are pair-wise similar, but appear at the far ends of the circular view angle space. In practice, the problem is even more complicated due to large visual variation of objects (e.g., different car models). We perform extensive experiments on the EPFL Multi-view Car and KITTI Datasets as well as the TUD Multi-view Pedestrians Dataset and achieve superior performance as compared to the state-of-the-art algorithms.**

***Keywords*—Visual regression, viewing angle estimation, hierarchical sliding slice regression, coarse classifiers, fine regressors.**

I. Introduction

The viewing angles (known as 'poses' or 'viewpoints') of surrounding vehicles implying the drivers' intentions provide important information for autonomous and assisting driving systems, intelligent transportation systems and surveillance in general. While LiDAR, time-of-flight and structured light sensors can provide 3D information, they can be expensive, unreliable outdoors and with limited working distance. In this work, we investigate single monocular camera based viewing angle estimation which can be installed to an affordable sensor setup. Passive vision systems in the terms of onboard front, side and back view cameras can provide a powerful and affordable alternative for viewpoint estimation. Considering the continuously-changing nature of the viewing angle, viewpoint estimation is usually formulated into a visual regression problem. In simple terms, given visual observations from images or video frames and corresponding vehicle viewpoints as inputs and outputs respectively, a regression function is trained and then applied to estimate poses in test images. A number of applications utilising estimated vehicle viewpoint have been proposed in the field of intelligent transportation [1], [2], [3], [4], [5], [6]. Pedestrian viewpoint estimation problem [7], [8], [9], [10] shares similar characteristics as vehicle viewing angle estimation.

Figure 1: Illustrative examples of the problem specific characteristics of vision-based car pose estimation. Top: visual differences between different models for the same viewing angle; middle: visual similarity of the neighbouring view angles for the same car; bottom: problem-specific circular similarity of flipping poses.

The problem remains active and yet challenging due to certain general and problem specific characteristics (Figure 1). The visual dissimilarity between different car types and models can be significant, and image distortions, such as illumination and perspective changes, make the problem challenging. Another challenge is that visual differences between similar viewing angles can be subtle (see the middle two rows of Figure 1) as compared to differences between two car models. Moreover, there is axial symmetry in similarity, for example between the front and back views and the left and right side views (shown in the bottom row of Figure 1) which represent the maximal difference in the output space (-180° vs $+180^\circ$) and make the problem highly non-linear. The above challenges make the global regression approaches fail and, therefore, a number of special circular regression methods have been proposed to achieve more robust performance and to address the inconsistent feature-pose relationship in estimating vehicle viewpoint [11], [12], [13], [14].

The vehicle viewpoint estimation needs to model the

This work was funded by Academy of Finland under the Grant No. 267581, 298700 and D2I SHOK project funded by Digile Oy and Nokia Technologies (Tampere, Finland). The authors wish to acknowledge CSC-IT Center for Science, Finland, for generous computational resources. (Corresponding author: K. Chen, email: ke.chen@tut.fi)

D. Yang, Y. Qian, K. Chen and J.-K. Kämäräinen are with the Laboratory of Signal Processing, Tampere University of Technology, Finland.

E. Berki is with the School of Information Science, University of Tampere, Finland.

problem-specific output space, *a global circular target space*, which is periodic from 0^o to 360^o, *e.g.* the distance between 15^o and 345^o should be shorter than that between 15^o and 60^o. In contrast to the related research work, we model the output space in a more advanced way by adopting a coarse-to-fine approach inspired by a number of hierarchical regression frameworks successfully used in non-circular regression problems [15], [16], [17], [18]. In particular, we assume that the global circular target space consists of a number of adjacent localised target groups (slices), which represent much stronger local correlation across neighbouring viewing angle targets as compared to weaker and inconsistent correlation of the full output space. The intuitive concept can be explained by the polygon approximation of the value of π, which incrementally approximates a circle via a number of line segments. As a result, the whole circular target space is made up of a number of linear localised subspaces as illustrated in Figure 2. Finally, in order to improve robustness, our design of local target groups borrows the concept of classic sliding windows to construct a number of overlapped sliding slices. Compared to hard group boundaries, the proposed sliding slice algorithm improves the robustness due to the introduction and optimisation of the size and stickness of the sliding pieces which become important method parameters that are optimised by cross-validation.

This paper concerns designing a novel two-layer regression framework, namely Hierarchical Sliding Slice Regression (HSSR), which consists of coarse classifiers to determine a main target group and target group optimised fine regressors to estimate viewing angles. For training each classifier, all samples within and outside a slice (the target group) are set to be positive and negative examples respectively, while only the samples belonging to the target piece are utilised to train its specific regressor. It is noteworthy that owing to the overlap defined by the *slice step* parameters the target spaces also overlap. Given a new testing image, imagery features are first fed into trained classifiers to determine the coarse target group and then the fine viewing angle is estimated with the trained regressor specific to the target group. Because of the introduction of the sliding-window concept into the hierarchical structure to both capture local target correlation and also improve the robustness against hard group boundaries, the proposed framework consistently achieves better performance in the experimental evaluation on the public EPFL multi-view Car benchmark dataset. We also evaluate the proposed method on a more realistic KITTI Vision Benchmark dataset, whose results verify that our method performs better than the state-of-the-art methods. TUD Multiview Pedestrians Dataset is employed to verify that our method is not limited to vehicles and also works effectively on other object pose estimation tasks.

II. Related Work

In the fields of intelligent transportation and visual surveillance, vehicle viewing angle estimation plays an important role in the success on vehicle recognition and tracking, but is difficult to perform with typical vehicle onboard sensors such as radar and laser rangefinder. Recent development of

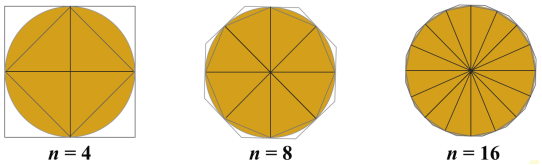


Figure 2: Intuitive concept of the proposed HSSR.

onboard car camera systems allows vision-based monitoring and detection which are essential for autonomous driving. The car view point problem was originally cast as a classification problem [12], where the 360^o pose space was quantized into a number of bins and a multi-class classifier was trained to learn the mapping from imagery feature space to the discrete angle range classes. However, the main limitation of the classification approach is that the class labels are explicitly made independent, which omits the natural latent continuously-changing correlation across pose class labels. For example, the closer the pose class labels of a pair of images are, the more visually similar these two images are and, therefore, learning of the classes benefits from examples in both classes. In the light of this, it is more meaningful to cast car pose estimation as a continuous regression problem [11], [13], [14], [19]. In visual regression, the goal is to learn a mapping from the imagery feature space to a continuous real-valued output space of viewing angles. Most of the existing regression methods for vehicle viewpoint direction incorporated manifold locality into either explicit feature representation [11], [13] or implicit regression model training [14]. In [11], a supervised manifold learning based algorithm was proposed to capture the mapping relation between one point in the local manifold to the target. Fenzi *et al.* [13] ensembled a set of generative models using local features to predict object pose. In [19], a probabilistic framework was proposed to incorporate generative regression on feature and a matching graph to discover geometry consistence across pose. Hara and Chellappa [14] introduced novel regression forests to implicitly model manifold locality in the pose space. Alternatively, another group of existing methods [20], [21], [22], [23], [24], [25] aimed to simultaneously cope with object detection and pose estimation, which is out of our scope about improving regression learning to exploit local manifold in target space.

However, these methods [11], [13], [14], [19] omit the special characteristics of the problem due to the repetitive visual structure of vehicles and circular output space. Our work was inspired by the recent hierarchical methods in visual regression [15], [16], [17], [18]. Dantone *et al.* [15] designed a hierarchical regression framework consisting of a set of weak local part regressors and a strong global regressor for estimating human gesture. Liu *et al.* [18] divided the whole target space into a set of overlapped groups and fused the prediction of regressors for different groups, which is different from our coarse-to-fine style in the proposed algorithm. Recent work [16], [17] shared a similar approach to our method in the form of coarse-to-fine hierarchical structure. However, their methods designed for non-circular regression problems divided the target space into pre-defined groups and then trained group-

specific regression, which required prior knowledge and was sensitive to grouping. On contrast, our method introduces the concept of sliding slice to relax the requirement of precise grouping and each training sample is included into multiple groups ($\geqslant 2$) in the circular pose space (data sharing). As a result, pose locality is discovered and exploited in the proposed method to improve the robustness against inconsistent feature-target mapping.

In the proposed algorithm, we take the best of the two worlds: hard class boundaries of the pre-classifier allow local correlations to be captured in the terms of class boundaries, and local regression helps to exploit consistent imagery features in regression. Moreover, overlapping target groups remove the effect of hard boundaries since each true angle is not near the boundary in more than one target group. Both our classifier and regression methods are trained using the examples in sliding slice manner over the circular target space of pose angles.

The novel contributions of our work are three-fold:

- Inspired by the concept of polygon approximation and hierarchical decision making, a hard coarse decision classifier is proposed as the first stage of visual regression for vehicle viewing angle estimation - nonlinear global circular pose changes are modelled via a number of piecewise overlapped linear models that model pose locally.
- A second stage fine regressor of visual imagery features is trained omitting boundaries between the hard "pose slices" by a sliding-slice approach that avoids treating samples near hard boundaries as extreme cases - in the next slice the same samples are near the centre line. This improves regression accuracy and robustness as compared to the hard boundaries.
- We provide an extensive experimental evaluation on multiple public benchmarks and report superior performance over state-of-the-art.

III. Methodology

Given p-dimensional imagery feature representation $\boldsymbol{x} \in \mathbb{R}^p$ and a scalar-valued viewing angle $y \in \mathbb{R}$, the input and output training pairs for the proposed two-stage hierarchical regression consist of $\{\boldsymbol{x}_i, y_i\}^N,\ i = 1, 2, \ldots, N$, where N denotes the number of training samples. As shown in Figure 3, the whole pipeline of the proposed approach consists of two steps which are 1) a set of coarse classifiers and 2) corresponding fine regressors.

- In the first step, the whole (circular) label space is quantised into a number of circular overlapping slices, and we train a strong classifier for each slice by using samples within the viewing angle subspace (slice) as positive examples and the remaining as negative examples.
- In the second step, fine regressors for each slice group are trained. The sliding slice subspaces help to better exploit output label correlations than the hard boundaries.

In the testing phase, an unseen image is first classified into a coarse angle group (slice generation in Section III-A), and then a corresponding regressor is used to provide a real-valued

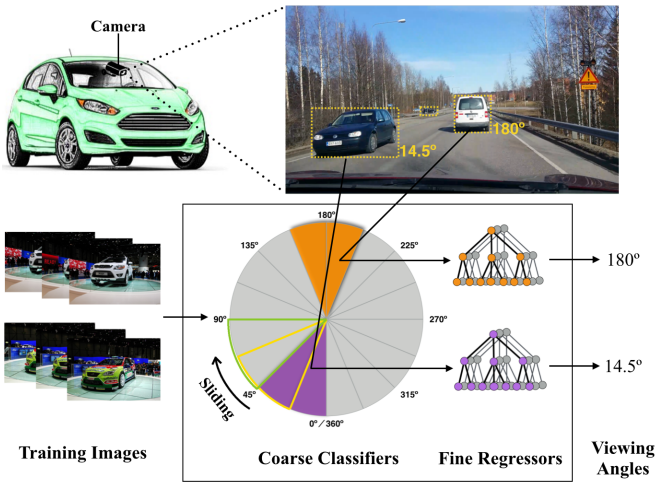


Figure 3: The overall workflow of our approach for estimating car pose. The estimation procedure is hierarchical in two steps where coarse grouping is first achieved via classification and then secondly fine-grained estimate via regression.

angle estimate. In our framework, the traditional single stage regression,

$$f : \boldsymbol{x} \rightarrow y,\ \ \boldsymbol{x} \in \mathbb{R}^p,\ \ y \in \mathbb{R}$$

is replaced with a two-step regression

$$f : \boldsymbol{x} \underset{f_1}{\rightarrow} \bar{\boldsymbol{y}} \underset{f_2}{\rightarrow} y$$

where $\bar{\boldsymbol{y}}$ defines the coarse angle space (slice) and instead of the single mapping f we need to construct two mappings f_1 and f_2, where f_1 is the coarse classifier (Section III-B) and f_2 a fine regressor (Section III-C). Note that f_2 depends on $\bar{\boldsymbol{y}}$ and its input is $\boldsymbol{x}$, i.e. it operates on the original feature space $f_2 = f_2(\boldsymbol{x}; \bar{\boldsymbol{y}})$.

A. Circular Slice Construction

For learning a robust regressor for continuous value estimation, a number of coarse-to-fine hierarchical regression approaches have been proposed [15], [16], [17], [18]. The results from either coarse regressors [15] or coarse classifiers [16], [17], [18] have positive effect on the fine regressor performance. Similarly, in our approach, the choice of constructing coarse angle groups is important to robustify our two-stage regression. On one hand, if the coarse slices are too fine and dense, examples become too ambiguous and classifier performance degrades. On the other hand, if the slices are too sparse, learning a good regressor becomes more difficult due to inconsistency in examples. Clearly, the slice size is an essential parameter for successful regression. However, the traditional approach is to use non-overlapping slices that uniformly span the output space. In this work, we adopt the sliding window approach and allow overlap of the slices by defining another parameter, *slice step*, that defines the amount of overlap (see Figure 3). Basically, this has only positive effect on the performance and the only disadvantage is that more

computation(e.g., more pre-classifiers in the one-vs-all SVM setting) is needed.

In the light of this, we devise strategies to determine and construct the coarse angle groups, slices, and their overlap, slice step, which are experimentally studied in Section IV. There are two parameters to be defined: the slice step and the slice size. We define the slice step value as proportional to the slice size, e.g., $\{1/4\times, 1/2\times, 3/4\times, 1\times\}$ where $1\times$ produces non-overlapping slices. The slice size has a strong effect on the success of regression, and therefore it should be optimally selected over a set of suitable values, e.g., $\{45°, 90°, 180°\}$. We experimentally study the effects of these parameters in the experimental part of our work (Tables II and III). It is noteworthy, that despite the fact that for simplicity we use uniform sampling over the circular space in this work, also non-uniform slice sizes and slice steps could be used to better cope with non-uniform data distributions. This could be achieved, for example, using spectral clustering on feature similarity space [26] or traditional vector quantization in the output space, but these are out of the scope in this work.

B. Coarse Classifier

Given the training pairs $\{\boldsymbol{x}_i, y_i\}^N$, $i = 1, 2, \ldots, N$, and the coarse angle groups (circular overlapping slices) $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_J\}$ with J denoting the total number of slices. The first step of our hierarchical regression is to estimate the correct angle group $\bar{\mathcal{G}}$ using a supervised trained classifier. For this purpose, we introduce a new set of binary output variables consisting of $\bar{y}_i^j$ which is 1 if the specific sample $\boldsymbol{x}_i$ belongs to the group $\mathcal{G}_j$ and 0 otherwise:

$$\bar{y}_i^j = \begin{cases} 1 & \text{if } y_i \in \mathcal{G}_j \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

$j = 1, \ldots, J$ classifiers are trained with all training samples $\{\boldsymbol{x}_i, \bar{y}_i^j\}_{i=1}^N$. We adopt the highly successful Support Vector Machine (SVM) classifier. In the experiments we adopt RBF-kernel SVM using libSVM [27]. The SVM target function for our case is

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i^N \xi_i \\ \text{s.t. } & \bar{y}_i^j(\boldsymbol{w}\cdot\Phi(\boldsymbol{x}_i) - b) \geqslant 1 - \xi_i, \\ & \xi_i \geqslant 0 \ , \end{aligned} \tag{2}$$

where $\boldsymbol{w}$ and b are the weight vector and bias to be optimised, and $\boldsymbol{\xi}$ consists of slack variables. $\|\cdot\|$ denote the Euclidean norm. $K(\boldsymbol{x}_w, \boldsymbol{x}_h) = \Phi(\boldsymbol{x}_w)\Phi(\boldsymbol{x}_h)$ is the kernel function to project low-dimensional input $\boldsymbol{x}$ to a high-dimensional kernel space. Trade-off parameter C is used to balance the regularised term and loss term, which needs data-specific tuning such as n-fold cross-validation. The object function and inequality constraints of Support Vector Machine, a convex optimisation problem, can be transformed into an equality-constrained dual problem with Lagrange multipliers. Based on the Karush-Kuhn-Tucker Conditions [28], the gradient of object function in the dual problem of SVM is enforced to zero, which can thus obtain the optimised $\boldsymbol{w}$ and b.

It is worth mentioning here that coarse classifiers are not limited to SVM, other classifier such as random forests [29], [30] and Logistic regression [31] can also be employed. We adopt SVM in our framework because of its benchmarking role in pattern recognition and stable performance in a number of relevant tasks [12], [32], [33], [34].

C. Fine Slice-Specific Regressor

After coarse group classification, fine regressor for each angle group $\mathcal{G}_j, j = 1, 2, \cdots J$ is trained to learn regression functions $f_j(\boldsymbol{x})$ that minimise the loss function $L(f_j(\boldsymbol{x}_k), y_k)$ $\forall \langle \boldsymbol{x}_k, y_k \rangle \mid y_k \in \mathcal{G}_j$ between the prediction $f_j(\boldsymbol{x}_k)$ and y_k. The object function for the jth regressor is the simple sum of squared loss

$$\min \quad \sum_{k=1}^{N} \|y_k - f_j(\boldsymbol{x}_k)\|^2. \tag{3}$$

Regression forest [35] is a popular regression method with high computational efficiency and robustness. It learns an ensemble of decorrelated regression trees by randomly selecting the training samples and features. In the training stage, each tree is grown independently with binary splitting strategy adopted in each node, *i.e.* each node can have two child nodes. To cope with the limitation of the standard binary splitting method leading to less efficient tree model in reducing the empirical loss, K-clusters Regression Forests (KRF) [14] employ a more flexible splitting method that can allow each node to have more than two child nodes. Motivated by the strong performance of the K-clusters regression forest in vehicle viewpoint estimation in [14], we adopt the following loss function for the jth regressor as:

$$L(f_j(\boldsymbol{x}_k), y_k) = 1 - \cos(y_k - f_j(\boldsymbol{x}_k)) \in [0, 2] \tag{4}$$

for training fine regressors in the second step of our framework. Each node of K-clustering regression forests partitions the output space into K clusters $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_K\}$, and the cluster labels are used to divide the input space into K disjoint subspaces $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_K\}$. It is worth mentioning here, clusters in $\mathcal{T}$ are determined without considering the input space. Let us assume that $\mathcal{T}^* = \{\mathcal{T}_1^*, \mathcal{T}_2^*, \cdots, \mathcal{T}_K^*\}$ are the optimised clusters and $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_K\}$ denoting a set of constant estimates associated with each subspace, object function in (3) can thus be written as

$$\mathcal{T}^* = \underset{\mathcal{T}}{\operatorname{argmin}} \quad \sum_{k=1}^{K} \sum_{i \in \mathcal{T}_k} 1 - \cos(y_i^j - \mathcal{A}_k), \tag{5}$$

where $\mathcal{T}_k = \{i : 1 - \cos(y_i^j - \mathcal{A}_k) \leqslant 1 - \cos(y_i^j - \mathcal{A}_l), \forall 1 \leqslant l \leqslant K\}$. Given $\mathcal{T}^*$, the problem is cast as a multi-class classification problem, *i.e.* partitioning K disjoint input subspace $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_K\}$ to preserve $\mathcal{T}$ can be equivalent to training samples $\boldsymbol{x}$ and their class labels $\{1, 2, \cdots, K\}$. For sample splitting at each node of K-clusters regression forests, Support Vector Machine with (2) is employed again. However,

different from coarse classifiers using RBF-kernel, we adopt linear kernel here for higher efficiency. Since determining the parameters of K, the size of clusters adopted in each node splitting, the straightforward choice is to tune via n-fold cross-validation. Alternatively, in [14], adaptive determination of the flexible number of child nodes for each node, namely Adaptive K-clusters Regression Forests (AKRF) was proposed by measuring Bayesian Information Criterion (BIC) [36], [37] to select the size of clusters K.

IV. Experiments

A. Datasets and Settings

Figure 4: Examples from the EPFL Multi-view Car Dataset (top) and the KITTI Vision Benchmark Suit (bottom).

The EPFL Multi-view Car Dataset is adopted for our experiments, which consists of 20 image sequences of 20 car types and has 2299 images of cars rotating in various directions. Example images from the dataset are shown in Figure 4. With manually-annotated bounding boxes, the foregrounds of images are cropped and resized into 64×64 image patches, from which multi-scale HOG (Histogram of Oriented Gradients) features [39] are extracted. For each image patch, 2×2 cell blocks and the cell size 8,16 and 32 are used and for each cell, 9 orientation bins are employed

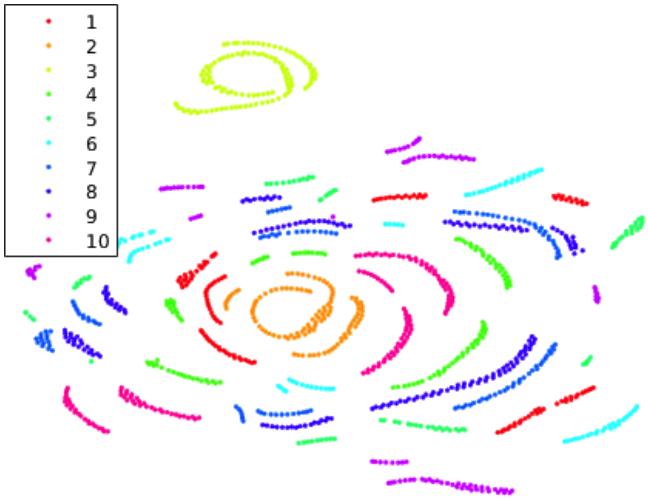


Figure 5: HOG feature visualization for the first 10 sequences in EPFL Multiview Car Dataset with t-SNE mapping to 2D.

to compute the orientation histogram. The dimensionality of the final HOG feature is 2124. Two experiments are conducted according to two settings of data split, even split and leave-one-sequence-out, which have been adopted from the recent works [13], [14] and [12] respectively. More specifically, images in the first 10 sequences are used for training and remaining for testing in even split, while leave-one-sequence-out protocol utilises images in one sequence for testing and the rest for testing each time and repeat until all the sequences are tested. Evidently, the former even split protocol is more challenging in view of more sparse train samples, compared to the latter one. We visualise the features for the first 10 sequences in Figure 5 by using t-SNE [40]. This figure visualises the feature variation between different sequences since each sequence is apparently isolated with others. We compare our results against a number of state-of-the-art methods for car pose estimation. Besides Ozuysal *et al.* [12], the remaining methods cast car pose estimation into a regression problem. KPLS [41], SVR [28], BRF [14], KRF [14] and AKRF [14] employ the identical HOG features. Free parameters of KPLS, SVR, KRF and AKRF are tuned via leave-one-sequence-out cross-validation by following the original works. Mean absolute error (mae) to take the average of the absolute differences between predicted angles and the ground truth is employed to evaluate and compare the performance of our approach. In addition, following the previous works, the mae of 90%-percentile of the absolute errors and that of 95%-percentile are also reported as robust performance measures.

Considering of that samples of the EPFL Dataset are only collected indoor which can be easier than real road images, we also evaluate our method on the KITTI Vision Benchmark Suite [42] which consists of 7481 training images and 7518 test images. All the images were captured in rural areas and

Table I: Comparative evaluation of the state-of-the-art methods with the EPFL Multi-View Car dataset.

Methods	*even split* mae (90%)	mae (95%)	mae (100%)	Median	*leave-one-sequence-out* mae (90%)	mae (95%)	mae (100%)	Median
Torki *et al.* [11]	19.40°	26.70°	33.98°	–	23.13°	26.85°	34.90°	–
Fenzi *et al.* [13]	14.51°	22.83°	31.27°	–	14.41°	22.72°	31.16°	–
KPLS [14]	16.86°	21.20°	27.65°	–	–	–	–	–
SVR [14]	17.38°	22.70°	29.14°	–	–	–	–	–
BRF [14]	23.97°	30.95°	38.13°	–	–	–	–	–
Zhang *et al.* [20]	–	–	24.00°	–	–	–	–	–
Fenzi and Ostermann [19]	–	–	23.28°	–	–	–	–	–
AKRF-W [38]	7.42°	15.94°	24.06°	–	–	–	–	–
AKRF-VW [38]	6.76°	15.65°	23.81°	–	–	–	–	–
AKRF [14]	7.73°	16.18°	24.24°	–	15.74°	21.50°	27.42°	3.49°
KRF [14]	8.32°	16.76°	24.80°	–	11.16°	14.99°	20.18°	3.11°
$HSSR_{KRF}$ (ours)	**3.88°**	**11.98°**	**20.30°**	**3.36°**	**8.31°**	**10.90°**	**14.24°**	**2.55°**
Ozuysal *et al.* [12]	–	–	46.48°	–	–	–	–	–
Redondo-Cabrera *et al.* [22]	–	–	39.8°	7°	–	–	–	–
Teney and Piater [23]	–	–	34.7°	5.2°	–	–	–	–
Yang *et al.* [24]	–	–	24.1°	3.3°	–	–	–	–
He *et al.* [21]	–	–	15.8°	6.2°	–	–	–	–
Fenzi *et al.* [25]	–	–	13.6°	3.3°	–	–	–	–

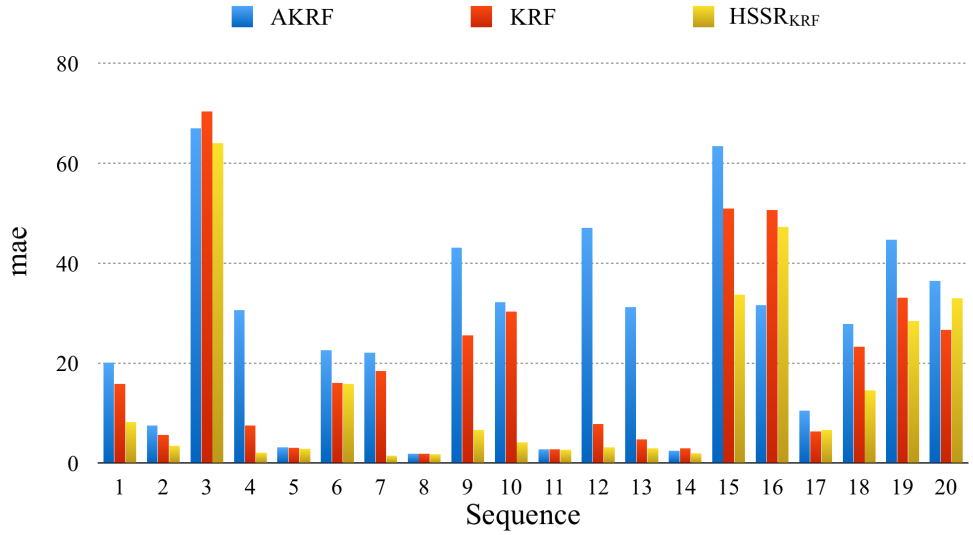


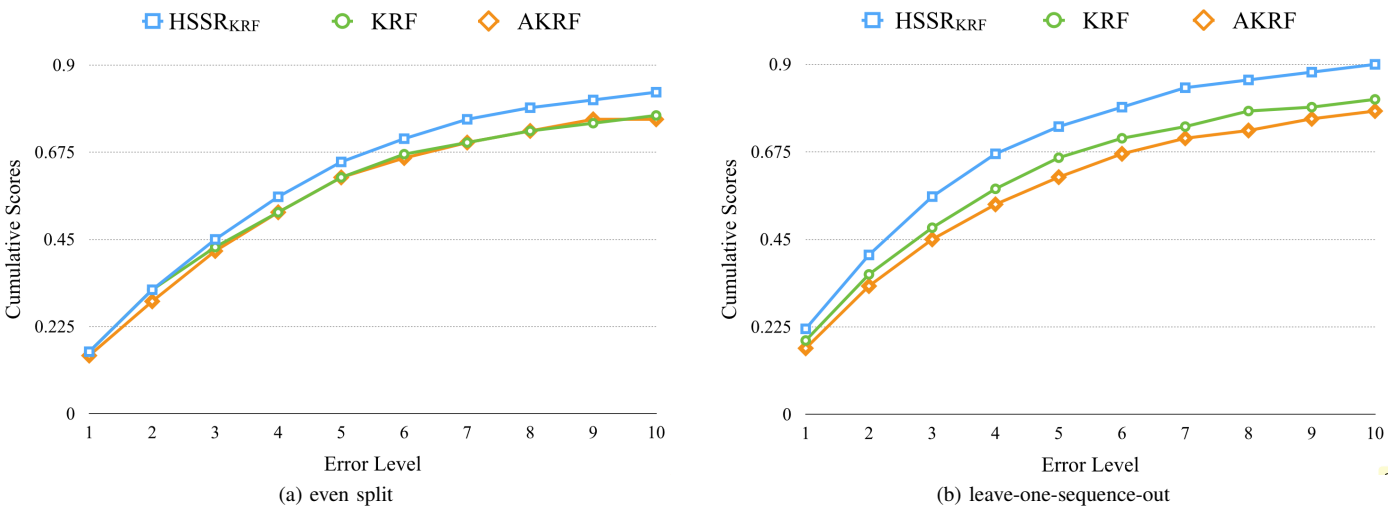
Figure 6: Comparison of $HSSR_{KRF}$ (ours), KRF and AKRF on mae (100%) for each sequence (leave-one-sequence-out).

highways, and there are different number of car instances in each image. We extracted the non-occluded car instances from the labelled training images and resized them into 64×64 image patches, from which HOG features are extracted as the same with previous feature extraction for EPFL Multiview Car Dataset. There are 13457 samples in total. We split those samples into two datasets: training dataset (6730 samples) and test dataset (6727 samples). The evaluation on this new formed dataset is measured in the same way as previous even split evaluation.

As a distinct visual regression task, we evaluate our method also on TUD Multiview Pedestrians Dataset [7] to predict pose of pedestrians. The dataset contains 5228 images of pedestrians and they are split into a training set of 4732 images, a validation set of 290 images and a testing set of 309 images. Continuous orientation labels are given in [38] by using Amazon Mechanical Turk. The label space of pedestrians is also a circular space from 0° to 360°. We followed the feature extraction process as [38]: color images are firstly converted into gray scale from which Multi-scale HOG features are extracted. PCA is used to reduce the feature dimensionality of each image into 2000.

B. Evaluation

Comparison with State-Of-The-Arts – The results in Table I verify that our method significantly outperforms all state-of-the-art methods with at least 16.25% marginal in reducing mae and 25.96% and 49.81% in reducing 95% mae and 90% mae, respectively, using the even split setting. Similar performance improvement is observed for the leave-one-sequence-out setting. Figure 6 illustrates the results for each



(a) even split (b) leave-one-sequence-out

Figure 7: Comparative cumulative scores. The higher the better.

Table II: Evaluation on the circular slice step size proportional to the $45°$ slice size.

	even split				*leave-one-sequence-out*			
Slice step	mae (90%)	mae (95%)	mae (100%)	median	mae (90%)	mae (95%)	mae (100%)	median
$1\times$ slice	7.88°	16.63°	24.73°	3.52°	10.75°	13.54°	17.34°	**2.35°**
$3/4\times$	4.30°	12.69°	20.98°	3.41°	10.21°	12.65°	16.14°	2.38°
$1/2\times$	4.11°	12.14°	20.45°	3.44°	8.98°	11.60°	15.52°	2.64°
$1/4\times$	4.97°	13.74°	21.98°	3.51°	8.81°	11.10°	14.63°	2.53°
Combination	**3.88°**	**11.98°**	**20.30°**	**3.36°**	**8.31°**	**10.90°**	**14.24°**	2.55°

Table III: Evaluation on the circular slice size.

	even split				*leave-one-sequence-out*			
Slice size	mae (90%)	mae (95%)	mae (100%)	median	mae (90%)	mae (95%)	mae (100%)	median
$180°$	4.51°	12.66°	20.93°	3.52°	9.65°	12.36°	16.90°	2.85°
$90°$	5.12°	14.00°	22.24°	3.51°	9.24°	11.83°	16.21°	2.66°
$45°$	**4.11°**	**12.14°**	**20.45°**	**3.44°**	8.98°	**11.60°**	15.52°	2.64°
$22.5°$	–	–	–	–	**8.55°**	11.71°	**15.39°**	**2.63°**

sequence separately and compares to our direct competitors KRF and AKRF. Evidently, the proposed method performs better in most of the sequences. By exploiting target locality, our method can mitigate the suffering from the flipping errors ($\approx$ $180°$, *e.g.* the sequences 3, 15 and 16). By adopting the cumulative scores introduced in Geng *et al.* [43], we visualise the results generated by our method (HSSR$_{KRF}$) and the other state-of-the-art methods in Figure 7. It is shown that the HSSR$_{KRF}$ approach significantly improves the accuracy with both splitting methods. Considering the identical HoG features and regressor adopted for HSSR$_{KRF}$ and KRF, notable improvement on performance can only be explained by our hierarchical design.

Slice Construction – Given a fixed slice size of $45°$, Table II compares varying slice step values. The results indicate that the combination of all different step sizes performs best. Besides the combination strategy, the best results for even splitting are achieved using the $1/2\times$ slice step (half overlap) while for the leave-one-sequence-out setting the $1/4\times$ step performs the best (three quarters overlap). Evidently, all overlapping strategies (*i.e.* $3/4\times$, $1/2\times$, $1/4\times$, and combination) show higher accuracy than the non-overlapping slice spacing ($1\times$ slice) which verifies our sliding slice strategy.

Evaluation on Varying Size of Sliding Window – We also evaluate different sizes of the slices by half slice step in our method and the results are shown in Table III. Among them, $45°$ achieved the best results. Notably, HSSR is superior to state-of-the-art with all slice sizes (cf. Table I).

Evaluation on Automatic Detected Cars – KRF, AKRF and HSSR are evaluated on testing image with automatically detected bounding box. The state-of-the-art YOLO detector

Table IV: Results for YOLO detected bounding boxes (EPFL test) and HOG features. The average running times to process one image with and without YOLO detection are also reported. Computer with Intel Core i7-4790 CPU and 16 GB of memory is employed to conduct experiments.

Methods	mae (90%)	*even split* mae (95%)	mae (100%)	median	runtime	detect +runtime
AKRF [14]	21.96°	29.31°	36.61°	9.47°	14ms	53ms
KRF [14]	21.31°	28.46°	35.80°	9.64°	13ms	52ms
$HSSR_{KRF}$ (ours)	**17.61°**	**24.25°**	**31.80°**	**8.13°**	19ms	58ms

Table V: Evaluation on the KITTI Car Dataset.

Methods	mae (90%)	*even split* mae (95%)	mae (100%)	median
AKRF [14]	1.36°	2.69°	9.12°	0.83°
KRF [14]	1.28°	2.30°	8.38°	0.90°
$HSSR_{KRF}$ (ours)	**0.98°**	**1.28°**	**5.06°**	**0.77°**

Figure 8: Examples from TUD Multiview Pedestrians Dataset: (top) successful cases and (bottom) failure cases. The largest errors are often due to poor illumination, background clutter or motion blur.

[44] is employed to detect cars and provide bounding boxes. HOG features of detected cars are extracted and then fed to trained models to predict the viewing angles. As shown in Table IV our method performs best in reducing 11.2% mae compared to KRF. For the average running time, our method is comparable with KRF and AKRF, especially when detecting time is included (final column).

Evaluation on KITTI Benchmark – We adopt KITTI Benchmark to verify that our method can tackle with car pose prediction in realistic environment. 45° sized sliding window and half slice step are employed in the evaluation. The results in Table V show that our method outperforms KRF and AKRF with 39.62% marginal in reducing mae and 44.35% and 23.44% in reducing 95% mae and 90% mae.

Evaluation on TUD Multiview Pedestrians Dataset – Experiment on TUD dataset is conducted to predict pose of pedestrians whose label space can also be considered as circular. 90° sized sliding window and half slice step are adopted. As shown in Table VI, our method performs better than the AKRF-VW [38] by reducing 6.3% mae. HSSR obtains the best result in Accuracy-45° (ratio of predicted errors within 45°) and the result 66.7% in Accuracy-22.5° which is comparable with the best 68.6%. Some successful and failed examples are shown in Figure 8.

Table VI: Evaluation on the TUD Multiview Pedestrians Dataset.

Methods	Accuracy-22.5°	Accuracy-45°	mae	median
AKRF [38]	63.1%	76.1%	36.1°	–
AKRF-W [38]	65.7%	76.1%	35.9°	–
AKRF-VW [38]	**68.6%**	78.0%	34.7°	–
KRF	62.1%	77.3%	35.2°	15.8°
$HSSR_{KRF}$ (ours)	66.7%	**81.5%**	**32.5°**	**12.5°**

V. Conclusion

In this paper a two-step, coarse-to-fine, hierarchical approach is proposed for visual regression where one-vs-all SVM classifiers are used to find coarse regression groups and group-specific regressors are used to provide an accurate estimate. Our application is vehicle viewing angle estimation where axial symmetry brings additional challenges for regression. In this case, we form the groups as circular overlapping slices and demonstrated how this approach leads to state-of-the-art accuracy on public benchmark datasets. In our future work, we will extend the novel approach to other similar circular visual regression problems and study the effect of non-uniform slices and overlaps to further improve the approach.

References

[1] J. Nilsson, J. Fredriksson, and A. C. Odblom, "Reliable vehicle pose estimation using vision and a single-track model," *IEEE Transaction on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2630 – 2643, 2014.

[2] A. D. Sappa, F. Dornaika, D. Ponsa, D. Gerónimo, and A. López, "An efficient approach to onboard stereo vision system pose estimation," *IEEE Transaction on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 476 – 490, 2008.

[3] H. He, Z. Shao, and J. Tan, "Recognition of car makes and models from a single traffic-camera image," *IEEE Transaction on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3182 – 3192, 2015.

[4] C.-C. R. Wang and J.-J. J. Lien, "Automatic vehicle detection using local features – a statistical approach," *IEEE Transaction on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 83 – 96, 2008.

[5] T. Chen, R. Wang, B. Dai, D. Liu, and J. Song, "Likelihood-field-model-based dynamic vehicle detection and tracking for self-driving," *IEEE Transaction on Intelligent Transportation Systems*, vol. pp, no. 99, pp. 1 – 17, 2016.

[6] S. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath, "Collision sensing by stereo vision and radar sensor fusion," *IEEE Transaction on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 606 – 614, 2009.

[7] M. Andriluka, S. Roth, and B. Schiele, "Monocular 3d pose estimation and tracking by detection," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 623–630.

[8] D. Baltieri, R. Vezzani, and R. Cucchiara, "People orientation recognition by mixtures of wrapped distributions on random trees," *European Conference on Computer Vision (ECCV)*, pp. 270–283, 2012.

[9] T. Gandhi and M. M. Trivedi, "Image based estimation of pedestrian orientation for improving path prediction," in *Intelligent Vehicles Symposium*. IEEE, 2008, pp. 506–511.

[10] C. Chen, A. Heili, and J.-M. Odobez, "Combined estimation of location and body pose in surveillance video," in *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*. IEEE, 2011, pp. 5–10.

[11] M. Torki and A. Elgammal, "Regression from local features for viewpoint and pose estimation," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2603–2610.

[12] M. Ozuysal, V. Lepetit, and P. Fua, "Pose estimation for category specific multiview object localization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 778–785.

[13] M. Fenzi, L. Leal-Taixé, B. Rosenhahn, and J. Ostermann, "Class generative models based on feature regression for pose estimation of object categories," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[14] K. Hara and R. Chellappa, "Growing regression forests by classification: Applications to object pose estimation," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 552–567.

[15] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, "Body parts dependent joint regressors for human pose estimation in still images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2131–2143, 2014.

[16] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 2578–2585.

[17] J. Foytik and V. K. Asari, "A two-layer framework for piecewise linear manifold-based head pose estimation," *International Journal of Computer Vision*, vol. 101, no. 2, pp. 270–287, 2013.

[18] K.-H. Liu, S. Yan, and C.-C. J. Kuo, "Age estimation via grouping and decision fusion," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2408–2423, 2015.

[19] M. Fenzi and J. Ostermann, "Embedding geometry in generative models for pose estimation of object categories." in *British Machine Vision Conference (BMVC)*, vol. 1, no. 2, 2014, p. 3.

[20] H. Zhang, T. El-Gaaly, A. M. Elgammal, and Z. Jiang, "Joint object and pose recognition using homeomorphic manifold analysis." in *Association for the Advancement of Artificial Intelligence (AAAI)*, vol. 2, 2013, p. 5.

[21] K. He, L. Sigal, and S. Sclaroff, "Parameterizing object detectors in the continuous pose space," in *European Conference on Computer Vision*, 2014, pp. 450–465.

[22] C. Redondo-Cabrera, R. López-Sastre, and T. Tuytelaars, "All together now: Simultaneous object detection and continuous pose estimation using a hough forest with probabilistic locally enhanced voting," in *British Machine Vision Conference (BMVC)*, 2014.

[23] D. Teney and J. Piater, "Multiview feature distributions for object detection and continuous pose estimation," *Computer Vision and Image Understanding*, vol. 125, pp. 265–282, 2014.

[24] L. Yang, J. Liu, and X. Tang, "Object detection and viewpoint estimation with auto-masking neural network," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 441–455.

[25] M. Fenzi, L. Leal-Taixé, J. Ostermann, and T. Tuytelaars, "Continuous pose estimation with a spatial ensemble of fisher regressors," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1035–1043.

[26] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems (NIPS)*, 2004, pp. 1601–1608.

[27] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[28] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[29] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3d face analysis," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 437–458, 2013.

[30] C. Huang, X. Ding, and C. Fang, "Head pose estimation based on random forests for multiclass classification," in *20th International Conference on Pattern Recognition (ICPR)*. IEEE, 2010, pp. 934–937.

[31] A. DeMaris, "A tutorial in logistic regression," *Journal of Marriage and the Family*, pp. 956–968, 1995.

[32] K. Chen, S. Gong, and T. Xiang, "Human pose estimation using structural support vector machines," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2011, pp. 846–851.

[33] J. Orozco, S. Gong, and T. Xiang, "Head pose classification in crowded scenes," in *Proceedings of the British Machine Vision Conference (BMVC)*, vol. 1, 2009, p. 3.

[34] R. Muñoz-Salinas, E. Yeguas-Bolivar, A. Saffiotti, and R. Medina-Carnicer, "Multi-camera head pose estimation," *Machine Vision and Applications*, vol. 23, no. 3, pp. 479–490, 2012.

[35] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[36] R. L. Kashyap, "A Bayesian comparison of different classes of dynamic models using empirical data," *IEEE Transactions on Automatic Control*, vol. 22, no. 5, pp. 715–727, 1977.

[37] G. Schwarz *et al.*, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[38] K. Hara and R. Chellappa, "Growing regression tree forests by classification for continuous object pose estimation," *International Journal of Computer Vision*, pp. 1–21, 2016.

[39] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[40] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[41] R. Rosipal and L. J. Trejo, "Kernel partial least squares regression in reproducing kernel hilbert space," *The Journal of Machine Learning Research*, vol. 2, pp. 97–123, 2002.

[42] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 3354–3361.

[43] X. Geng, Z.-H. Zhou, and K. Smith-Miles, "Automatic age estimation based on facial aging patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2234–2240, 2007.

[44] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.