All together now: Simultaneous Object Detection and Continuous Pose Estimation using a Hough Forest with Probabilistic Locally Enhanced Voting

Carolina Redondo-Cabrera[1]
carolina.redondoc@alu.uah.es
Roberto López-Sastre[1]
robertoj.lopez@uah.es
Tinne Tuytelaars[2]
Tinne.Tuytelaars@esat.kuleuven.be

[1] University of Alcalá
GRAM
Alcalá de Henares, ES

[2] K.U. Leuven,
ESAT-PSI, iMINDS
Leuven, BE

Abstract

Simultaneous object detection and pose estimation is a challenging task in computer vision. In this paper, we tackle the problem using Hough Forests. Unlike most methods in the literature, we focus on the problem of *continuous* pose estimation. Moreover, we aim for a probabilistic output. We first introduce a new pose purity criterion for splitting a node during the forest training. Second, we propose the concept of *Probabilistic Locally Enhanced Voting* (PLEV), a novel regression strategy which consists in modulating the regression with a kernel density estimation to consolidate the votes in a local region near the maxima detected in the Hough space. And third, we propose a pose-based back-projection strategy to improve the bounding box estimation. With these three additions, we show that our Hough Forest can achieve state-of-the-art results without needing 3D CAD models. We present a quite versatile method, showing results for different categories (cars as well as faces) and for different modalities (RGB as well as depth images).

1 Introduction

Object category detection has received a lot of attention over the last decades and a lot of progress has been realized. Recently, several approaches have gone one step further proposing solutions for the problem of simultaneous object category detection and pose estimation. That is also the problem we are addressing in this paper.

We start from the observation that current methods suffer a number of shortcomings. First, most methods still rely on coarse quantizations of the poses for multi-view object detection [12, 14, 17, 19], while only few approaches take into account that pose estimation of object categories is ultimately a continuous problem [4, 6, 11, 21]. Second, state-of-the-art approaches [14, 16, 21] achieve remarkable performance when leveraging existing 3D CAD models for the object class of interest. However, such models are not always available or easily constructible. Third, methods usually output a single combined confidence score

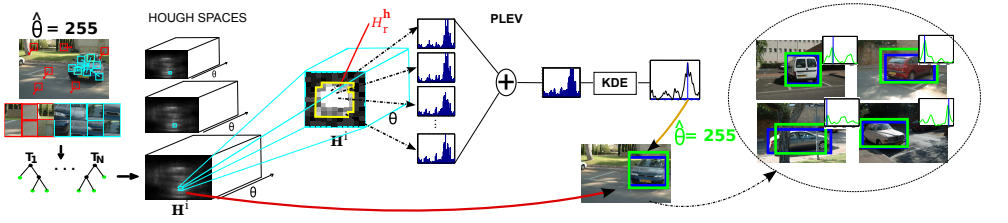


Figure 1: Our approach is able to jointly estimate the localization and the continuous pose of objects. To this end, we follow a HF regression voting in conjunction with our PLEV strategy, to integrate votes from a local region in the Hough space near the detected modes.

that reflects both the confidence in detection as well as pose estimation. Ideally, these two should be decoupled, and for the pose a full probability density function over the continuous pose space is desirable.

We propose a new approach (see Figure 1) which *jointly* solves both tasks, providing detection hypotheses and probabilistic estimates of their *continuous* pose. We draw inspiration from recent work on Hough Forests (HF) for object detection [7]. Here, we introduce a new formulation for the regression to be performed with HF, incorporating an uncertainty criterion for the continuous pose of the categories. This uncertainty in pose is decoupled from the traditional localization uncertainty [7], which allows us to randomly choose between them during the HF learning. The resulting HF can effectively locate objects and estimate their pose, by pooling votes from Random Forest (RF) regressors.

However, the extension of the Hough space to cover also the pose regression turns out to be suboptimal. The main reason is that the pose voting is very noisy, as we have experimentally observed, especially for views with shared appearance. Instead, we propose to first localize the object, and then estimate its pose. For this second step, a novel regression strategy is introduced, named *Probabilistic Locally Enhanced Voting* (PLEV), which consists in modulating the regression with a kernel density estimation (KDE) to consolidate all the votes in a *local* Hough region near the maxima detected in the Hough space. With the PLEV, our HF can cope with the uncertainty of the pose estimation votes. The output of our model is in the form of a probability density function (PDF) for the pose estimation. This is especially useful when fusing information from multiple sources. As a case in point, we show how to exploit the temporal continuity in a video sequence to obtain more accurate pose estimation by fusing information from multiple frames. Having a PDF also allows to deal with symmetry in a principled way. We finally propose to integrate a novel pose-based backprojection (BP) strategy to boost the bounding box (BB) estimation using the pose cues.

The joint location and pose estimator we obtain has several advantages compared to existing alternatives. Building on HF, our method is quite generic. In section 3, we show results on cars as well as faces, and using RGB as well as depth images as input. As a HF based approach with simple features, it is efficient and fast. Being a voting-based scheme, it is intrinsically robust to occlusions. While many state-of-the-art approaches achieve remarkable performance when training data is plentiful (*e.g.* leveraging existing 3D CAD models for the object class of interest [14, 21]), our approach is simple in the sense that we are able to learn the model directly from annotated images. This makes the method suitable for a wide range of categories. Lastly, thanks to our PLEV strategy, we obtain a probabilistic output score, allowing easy integration as a building block in a larger probabilistic framework. Our

extension to video-based pose estimation shows how to leverage the temporal continuity in video, even though poses may change from frame to frame.

The rest of the paper is organized as follows. In the next section, a review of related work is given. Section 2 introduces the proposed model, *i.e.* the Regression HF with PLEV. Section 3 presents the experimental results. We conclude in Section 4.

1.1 Related Works

Pose estimation of object categories is a growing research field. We find two types of strategies: 2D [9, 12, 13, 17, 19] and 3D [14, 21]. Within the 2D group, [17] represents an object category as a collection of view-invariant regions linked by transformations that capture the relative change of pose among parts. This model is extended into a generative approach in [19]. [13] uses a SVM classifier trained for each discrete pose with spatial pyramids of histograms. And in [12], a semi-latent approach is followed to train a Deformable Part Model [5], where the components correspond to discrete viewpoints.

3D methods have been mostly proposed within the context of continuous pose estimation of object categories [14, 21]. These works leverage 3D CAD models to generate synthetic training data. While CAD models may be widely available for some classes (*e.g.* cars), they may be less abundant or less realistic for other classes (*e.g.* faces). Also, these works apply a coarse to fine quantization of the viewpoints. We believe this dramatically increases the complexity of the models, which is in contrast to our compact HF based approach.

Other examples of continuous pose estimation are [6, 11, 20]. In [20], a regression approach is employed on the whole image, projecting all features on a smaller dimensional manifold. In contrast, we use local patches, as in [6]. While our method is a HF based strategy, in [6] they propose to learn regression functions from local descriptors of the same patch collected under different viewpoints. Moreover, we simultaneously detect the object and estimate the pose, which they don't. Finally, the method in [11] builds a class model (for detection and pose estimation) by merging 3D shapes of objects, obtained by applying a structure from motion reconstruction on the training data. This clearly limits the applicability of the model to datasets where such reconstruction is possible. Moreover, the model feeds the detections to a SVM classifier to refine their location and improve their pose estimates. Ours is a closed solution, fully integrated in the HF framework.

HF based solutions have also been proposed [4, 7, 10]. In [7] a HF is learned using discrete pose annotations in the training data. The pose estimation is considered as a *classification* problem. This is in contrast to our formulation, where: i) the pose estimation is treated as a *regression* problem; ii) classification and regression coexist, such that multiple classes can in principle be integrated. In [10], a HF based approach to predict the pose of interior body joints, from depth images, is described. The method uses a Gaussian Parzen density estimator to aggregate the votes for each body joint. Note that our approach is, in contrast to [10], more compact: i) our PLEV strategy *directly* uses votes in the Hough space, neither mean-shift nor depth adjusted weights strategies are needed as in [10]; ii) the PLEV casts the *final* pose estimation for the object, so the kinematic tracker used in [10] is not needed either.

The PLEV is inspired by [16], where *regions* of the Hough space are optimized to serve as seed regions for finding an optimal segmentation of an object. However, the idea of our PLEV is more simple: to capture the uncertainty of the pose estimation votes by considering a local Hough *region*. That is, no optimization is needed to define the regions, we simply gather the votes from the neighborhood of the detected object location in the Hough space

to feed the PLEV strategy, which is novel and essential for good results, as clearly shown in the experiments.

Probably most similar to our work is [4], where HF are used to estimate the head pose estimation in depth images. In their approach the uncertainty for both the localization and the pose estimation are integrated in the same regression measure. However, we keep them separated, defining specific uncertainty measures for each particular task, which are randomly chosen during the HF learning. We also incorporate the PLEV strategy to the HF framework. These aspects allow us to go beyond the particular application of head pose estimation from depth images. Actually, our experimental evaluation shows that we are able to obtain similar results for the problem of head pose estimation *simply* from RGB images (*i.e.*, without using depth), which we consider an important contribution of our work.

2 HF with Probabilistic Locally Enhanced Voting

RF [1] are a powerful tool for classification and regression problems. A typical RF is an ensemble classifier consisting of a set of randomized decision trees. During training, a binary weak classifier is learned for each non-leaf node. At runtime, test samples are passed through the trees, and the output is computed by averaging the distributions learned at the reached leaf-nodes. Hough Forests (HF) [7] are a generalization of the Hough transform within the RF framework. The randomized trees are trained to learn a mapping from sampled d-dimensional features to their corresponding votes in a Hough space $\mathcal{H} \in \mathbb{R}^H$. We build on the work of Gall *et al.* [7] and introduce a novel HF formulation for the challenging problem of simultaneous object detection and continuous pose estimation.

2.1 Training

In our HF $\mathcal{F}$, we aggregate a set of T binary decision trees $\mathcal{T}_t(P_i) : \mathcal{P} \rightarrow \mathcal{H}$, where $\mathcal{P} \subset \mathbb{R}^d$ is the d-dimensional feature space and $\mathcal{H} \subset \mathbb{R}^h$ describes the Hough space where the hypotheses are encoded. This Hough space lets us recover hypotheses for the location and the continuous pose of the object at multiple scales. Each object hypothesis $\mathbf{h} \in \mathcal{H}$ can be defined as $\mathbf{h} = (x_h, y_h, \theta_h, s_h)$, where x_h and y_h encode the position of the object, $\theta_h \in \mathbb{R}^p$ represents the continuous pose, and s_h identifies the scale. Note that our formulation can integrate different pose definitions for the objects (*e.g.* the viewpoint angle $\theta_h \in \mathbb{R}$, a combination of azimuth and zenith $\theta_h \in \mathbb{R}^2$, etc.).

We learn the HF $\mathcal{F}$ from a set of sampled image patches $P_i = \{(\mathcal{I}_i, c_i, d_i, \theta_i)\}$, of size $R \times R$. $\mathcal{I}_i = \{I_i^1, I_i^2, ..., I_i^C\}$ encodes the appearance of a training image, with I_i^j the appearance of the j^{th} channel. $c_i \in \{0,1\}$ is the class label (1 and 0 for features extracted from foreground and background patches, respectively). $d_i = (x_i, y_i)$ encodes the relative 2D location of the object center to the sampled patch. θ_i defines the continuous pose of the object. Training a decision tree involves recursively splitting each node such that the training data in newly created child nodes is more pure according to the class label (c_i), the relative 2D location (d_i) or the pose (θ_i). Each tree is grown until some stopping criterion is reached, *e.g.* the maximum tree depth or the minimum number of patches at a node.

Following the original HF formulation, at each node, a split function based on the patch appearance is learned. Our split function $f(\mathcal{I}_i; \tau_1, \tau_2, \{R_r\}_{r=1}^4)$ is characterized by the following parameters: the appearance channel specified by $\tau_1 \in \{1, 2, ..., C\}$, four asymmetric

rectangles defined within the patch $\{R_r\}_{r=1}^4$, and a threshold $\tau_2 \in \mathbb{R}$ for the difference of average values of the rectangular areas. We then define

$$f(\mathcal{I}_i;\tau_1,\tau_2,\{R_r\}_{r=1}^4) = \begin{cases} 0 \text{ if } f_a(\mathcal{I}_i;\tau_1,\{R_r\}_{r=1}^4) < \tau_2, \\ 1 \text{ otherwise.} \end{cases} \quad (1)$$

with $f_a(\mathcal{I}_i;\tau_1,\{R_r\}_{r=1}^4) = |R_1|^{-1} \sum_{q\in R_1} I_i^{\tau_1}(q) - \sum_{r=2}^{4} \left(|R_r|^{-1} \sum_{q\in R_r} I_i^{\tau_1}(q) \right)$. This is similar to the binary test used with depth images in [4], except that we use four regions instead of two to allow more flexibility.

For splitting a set of patches S into $\{S^{left}, S^{right}\}$ at tree node k, we first generate a pool of tests with random values for $\{\tau_1,\tau_2,\{R_r\}_{r=1}^4\}$. Then, we select the test which maximally reduces uncertainty in the class labels, or in the relative 2D location, or in the pose of the patches. So, we define three different impurity measures, $\mathcal{M}_*(\mathcal{S})$, which are chosen randomly during training. The impurity of the class labels is measured as in [2] by

$$\mathcal{M}_c(S) = H(S) - \sum_{child\in(left,right)} \frac{S^{child}}{S} H(S^{child}), \quad (2)$$

where $H(S)$ is the entropy given by $H(S) = -\sum_{c=0}^{1} p(c|S)\log(p(c|S))$, and $p(c|S)$ indicates the empirical distribution extracted from the training points within the set S.

The impurity of the relative 2D location of the patches, as in [7], is defined by

$$\mathcal{M}_d(S) = \sum_{child\in(left,right)} \sum_{j:c_j=1} ||d_j - \frac{1}{|S^{child}|} \sum_{k:c_k=1} d_k||^2. \quad (3)$$

We also introduce a novel regression purity criterion to measure the uncertainty in pose:

$$\mathcal{M}_p(S) = \sum_{child\in(left,right)} \sum_{j:c_j=1} \left(\frac{min\{(||\boldsymbol{\theta}_j - \boldsymbol{\theta}_A||), 360^\circ - (||\boldsymbol{\theta}_j - \boldsymbol{\theta}_A||)\}}{180^\circ} \right)^2, \quad (4)$$

where $\boldsymbol{\theta_A}$ is the viewpoint angle average over all foreground patches in the set S^{child} and it is computed taking the cyclic nature of viewpoint angles into account.

2.2 HF Regression with PLEV

During testing, patches sampled from the test image traverse the trees and cast votes to the Hough space $\mathcal{H} \in \mathbb{R}^H$ based on the location and pose distributions stored in the leaves. The forest-based estimate is then computed by aggregating votes from different patches. Following a standard HF regression approach [7], votes are accumulated in an additive way into the Hough space. Then, local maxima are found, *e.g.* using mean-shift. Here we introduce an alternative procedure, the PLEV (see Figure 1).

The PLEV starts, as in a standard HF voting strategy, by collecting the votes in our multidimensional Hough space $\mathcal{H} \subset \mathbb{R}^{2+p}$, where p is the number of angles that defined the continuous pose. We then resize the test image by a set of scale factors $\{s_1,s_2,\ldots,s_S\}$, and compute the corresponding Hough voting spaces $\{\mathcal{H}^1,\mathcal{H}^2,\ldots,\mathcal{H}^S\}$, where $\mathcal{H}^i \in \mathbb{R}^{2+p}$. Then, these Hough spaces are stacked and scaled, so the maxima can be jointly localized at multiple scales.

For a particular scale s_i, we first project all votes on the (x,y) subspace of $\mathcal{H}^i$, and recover the object center hypothesis $\hat{\mathbf{h}}_\mathbf{d} = (\hat{x},\hat{y})$ where the maximum is (see Fig. 1). The continuous

pose $\hat{\theta}$ might be estimated recovering the maximum along the pose dimension of $\mathcal{H}^i$. However, while for the object center the estimation based on the maximum works reasonably well, the pose needs to be refined. This is because Hough voting spaces are noisy, especially along the pose dimensions. So, inspired by [16], the idea of PLEV is to cope with this uncertainty by considering a local Hough region, rather than a single Hough maximum, for the regression of the pose.

We first build a local Hough region $H_r^{\hat{\mathbf{h}}_\mathbf{d}} \subset \mathcal{H}^i$ for each detection hypothesis $\hat{\mathbf{h}}_\mathbf{d}$. We consider to be in the defined local region only those voting positions $v_i \in H_r^{\hat{\mathbf{h}}_\mathbf{d}}$ which receive at least one vote and are spatially close to the detected maximum. Then, PLEV aggregates all *pose* votes received within $H_r^{\hat{\mathbf{h}}_\mathbf{d}}$, obtaining the distribution of the poses $g_r^{\hat{\mathbf{h}}_\mathbf{d}}$ in the Hough region, which can be computed as,

$$g_r^{\hat{\mathbf{h}}_\mathbf{d}} = \sum_{v_i \in H_r^{\hat{\mathbf{h}}_\mathbf{d}}} \left(\sum_{L_j \to v_i} \frac{p(c=1|L_j)}{|L_j|} p(\theta|L_j, v_i) \right), \tag{5}$$

where $p(c=1|L_j)$ and $|L_j|$ encode the foreground likelihood and the number of patches in leaf L_j, respectively, and $p(\theta|L_j, v_i)$ is the distribution of poses associated to the patches in leaf L_j which cast a vote in $H_r^{\hat{\mathbf{h}}_\mathbf{d}}$, which we denote as $L_j \to v_i$. Then, a Gaussian KDE is performed on $g_r^{\hat{\mathbf{h}}_\mathbf{d}}$ in order to obtain a smooth probability density function for the pose estimation,

$$f_{g_r^{\hat{\mathbf{h}}_\mathbf{d}}}(\theta) = \frac{1}{|g_r^{\hat{\mathbf{h}}_\mathbf{d}}|} \sum_{\forall g_r^{\hat{\mathbf{h}}_\mathbf{d}}(i)} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{(\theta - g_r^{\hat{\mathbf{h}}_\mathbf{d}}(i))^2}{2h^2}\right), \tag{6}$$

with h the bandwidth and $|g_r^{\hat{\mathbf{h}}_\mathbf{d}}|$ the total number of voting positions considered. The final estimation for the pose $\hat{\theta}$ is obtained as $\arg\max_{\hat{\theta}} = f_{g_r^{\hat{\mathbf{h}}_\mathbf{d}}}(\theta)$. Therefore, the final object hypotheses can be defined as $\hat{\mathbf{h}} = (\hat{\mathbf{h}}_\mathbf{d}, \hat{\theta}, s_i)$.

2.3 Pose-based backprojection for BB estimation

For the BB estimation, we follow a modified backprojection (BP) strategy inspired by [8]. For each object hypothesis $\hat{\mathbf{h}}$, we start by backprojecting to the image the largest BB of the training images. Within this initial BB, image patches are densely collected and passed again through the trees. To compute the BP mask, every time a patch at position y, *i.e.* $P(y)$, votes for $\hat{\mathbf{h}}$, we compute its contribution weight $\omega(P(y), \hat{\mathbf{h}})$ as follows,

$$\omega(P(y), \hat{\mathbf{h}}) = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{p(c=1|L_t(P(y)))}{|L_t(P(y), c=1)|} \sum_{L_t(P(y), c=1)} K(\hat{\mathbf{h}}, d, \theta, y) \right), \tag{7}$$

where $L_t(P(y))$ is the leaf reached by patch $P(y)$, and, $p(c=1|L_t(P(y)))$ and $|L_t(P(y), c=1)|$ encode the foreground likelihood and the number of foreground patches in leaf $L_t(P(y))$, respectively. In the second term, we propose to modify the contribution weight, using the relative 2D locations d and poses θ of the patches in leaf $L_t(P(y))$. We penalize patches that vote not only for different object locations, as in [8], but also for different poses. For doing

so, we define $K(\hat{\mathbf{h}}, d, \theta, y)$ as follows,

$$K(\hat{\mathbf{h}}, d, \theta, y) = \exp\left(-\sqrt{\frac{1}{\lambda}||d - \frac{u(y-\hat{h}_d)}{s_i}||^2 + \left(\frac{min\{||\hat{\theta}-\theta||, 360^\circ - (||\hat{\theta}-\theta||)\}}{180^\circ}\right)^2}\right), \quad (8)$$

where λ is a normalized parameter equal to $(u-R)^2$, u is the normalized training size of the BBs, and R is the patch size. The first and second terms of Eq. 8 consider the object localization and pose estimation errors, respectively. To obtain the bounding box, the mask is thresholded to estimate the tightest bounding box encompassing the binary mask. As in [8], the threshold is defined by $\frac{1}{2} max_y(\omega(P(y), \hat{\mathbf{h}}))$.

3 Experiments

3.1 Experimental Setup

We build our approach starting from our own implementation of [7]. Positive examples are cropped and rescaled to the same size, chosen so that the largest BB dimension is $u = 100$ pixels. Negative examples are extracted from PASCAL VOC 2007 [3], except for the Biwi Kinect Head Pose dataset (see details below). Patch size R is fixed to 30×30 pixels. The trees are trained with 20 positive and 20 negative patches randomly extracted from each training example. In each node 20.000 binary tests are considered. Forests have 15 trees with a maximum depth of 20. For RGB images, we use the same features as in [7], while for depth images simply the depth values are used (following [4]). For the PLEV we consider a neighborhood of 11×11 pixels. We follow the KDE implemented in [18], and tune the smoothing parameters and bandwidths following a leave-one-out strategy, where a single training object model is used for validation.

We evaluate the object detection performance using Average Precision (AP) as in [3]. For continuous pose estimation, we use, the Median Angular Error [11], and the Mean Absolute Error [13]. For discrete viewpoint classification, the Pose Estimation Average Precision (PEAP) and the Mean Precision of Pose Estimation (MPPE) proposed in [12] are used.

We first evaluate our approach on the *Weizmann Cars Viewpoint* (WCV) dataset [11]. It contains 1539 images of 22 different cars divided into 3 sets. Following the setup described in [11], we use one set for testing and the other two for training. At test time, 12 scales (from 1.25 to 2.35) are used. Next, we present results using the *EPFL Multi-view car dataset* (EPFL) [13]. It contains around 2000 images, belonging to 20 different car models. We follow the experimental setups proposed in [13] and [21] using 8 scales (from 1.1 to 2.5) at test time. We also report results on the *Biwi Kinect Head Pose Database* (Biwi) [4]. This dataset contains over 15K images of 20 people. We use the same setup as in [4] employing 4 scales (from 0.8 to 1.1) at runtime. We finally evaluate our approach on the discrete pose estimation problem, using the cars of the *3D Object categories dataset* [17] and the setup presented in [17]. At test time, 16 scales (from 1.35 to 2.85) are used.

3.2 Effect of our contributions on pose estimation

We first evaluate the impact of *each* of our contributions using the WCV dataset. We start from a baseline system where a standard HF is trained, and then gradually add our contributions one-by-one. For the baseline system, we follow our approach in Section 2 and train a

Table 1: Analysis on WCV dataset.

Method	Median Angular Error	Mean Absolute Error
Baseline	18°	58.6°
Baseline + pose uncertainty	12°	42.4°
Baseline + PLEV	**7°**	33.4°
Baseline + pose uncertainty + PLEV	**7°**	27.5°
Baseline + pose uncertainty + PLEV + BP	**7°**	**25.8°**

Table 2: Results on WCV dataset.

	Glasner *et al.* [8]		Our approach	
	azimuth	elevation	azimuth	elevation
Mean	36.44°	8.66°	**25.8°**	**3.2°**
Median	12.25°	5.41°	**7°**	**3°**
Std	55.32°	8.18°	**52.5°**	**3°**

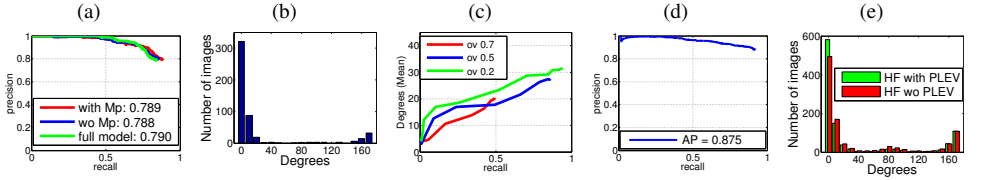


Figure 2: (a), (b) and (c) Analysis on WCV dataset. (a) Detection results. HF trained with (red curve) and without (blue curve) the pose uncertainty, and our full model (green curve). (b) Azimuth Error Histogram. (c) Azimuth mean error vs recall for different PASCAL VOC overlap criteria. (d) and (e) Analysis on EPFL dataset. (d) Precision-recall curve. (e) Angular Error Histograms.

HF where *only* the class-labels and the offset uncertainties (Eq. 2 and 3) are used. Neither the pose uncertainty (Eq. 4) nor PLEV and nor pose-based BP masks are used. The estimated hypotheses are recovered directly from the maxima of the Hough voting space and the BBs are estimated by taking the average BB of the training examples. Table 1 shows the results.

Pose uncertainty. Results in Figure 2(a) compare the precision recall curves for object detection with and without the pose uncertainty. Both curves are very similar. This reveals that incorporating the representation of the pose into the model does not overly impact the detection, while the pose estimation results (Table 1) improve drastically: more than 15° for the mean and 6° for the median.

PLEV. Next we evaluate the performance of our HF combined only with the PLEV. Note that the pose uncertainty is not used in this experiment. Results in Table 1 confirm the adequateness of the proposed Hough region based strategy for regression. Introducing PLEV reduces the mean error by more than 25°.

Pose uncertainty & PLEV. When both the pose uncertainty and the PLEV are integrated into our HF, the mean decreases to 27.5°, while the median is reduced from 18° to 7°.

All together now. Integrating the proposed pose-based BP with the rest of contributions into our HF, the mean decreases to 25.8°, while the detection slightly improves (see also Fig. 2(a)), showing that the right pose estimation helps the object detection.

3.3 Detection and continuous viewpoint estimation results

WCV dataset. We achieve an AP of 79% (see Fig. 2(a) green curve). We are the first ones reporting detection results on this dataset. Table 2 shows that our model outperforms the state-of-the-art in pose estimation. Figure 2(b) shows a histogram of the azimuth error. The small peak around 180 degrees is caused by the similarity of opposite views. Figure 2(c) shows the trade-off between azimuth mean error and recall. We generate these curves using the confidence of the pose estimation as the score to rank the predictions, and vary the recall for different PASCAL VOC overlap criteria. The higher the confidence of our pose estimation, the lower the mean error. If we relax the overlap criterion (from 0.7 to 0.2), the pose estimation performance also decreases. This reveals a nice property of our system: the more precise the object localization, the better the pose estimation. Qualitative results for

Table 3: Results on Biwi Kinect Head Pose Database.

Our approach						
Images	Position Error (mm)	Direction Error (°)	Yaw (°)	Pitch (°)	Roll (°)	missed frames (%)
RGB	–	9.8±6.8	5.8°±5.9°	5.8°±4.8°	3.5°±3.4°	**2.4**
Depth	**7.18±12.1**	7.3±5.9	4.1°±6.9°	3.9°±4°	3.2°±3°	5
Early Fusion	–	7.1±4.9	4±4.9	3.7±4.2	3.1±2.9	5.2
Late Fusion	–	7.0±4.7	**3.7±4**	4.1±3.4	**2.6±2.6**	5.2
Fanelli *et al.* [8]						
Depth	12.2±22.8	**5.9±8.1**	3.8±6.5°	**3.5±5.8°**	5.4±6.0°	6.6

Table 4: Temporal continuity.

	WCV		EPFL	
	Mean Error	Median Error	Mean Error	Median Error
With	**22.5°**	7°	**29.7°**	8°
Without	25.8°	7°	39.8°	7°

Table 5: Results on EPFL.

	[8]	[4]	[1]	Our model
Mean Error	46.48°	33.98°	**31.27°**	39.8°
	[8]	[4] Lin	[4] Exp	Our model
Median Error				7°
8 bins	24.8°	11.1°	9.6°	
16 bins		6.9°	7.5°	
36 bins		**4.7°**	**4.7°**	

Table 6: Car results on [17].

2D Models	AP/MPPE	3D Models	AP/MPPE
[8]	96.0 / 89.0	[8]	99.2 / 85.3
[6]	**99.9 / 97.5**	[6]	99.6 / 95.8
Our model	89.0 / 90.2	[6]	99.9 / 97.1

this dataset are shown in Figure 3(a).

EPFL dataset. Our model yields an AP of 87% (see Fig. 2(d)), while median and mean error are 7° and 39.8°, respectively. Figure 2(e) shows how the PLEV improves the performance of the pose estimation (green bars), increasing the number of estimations with an error lower than 10°. Table 5 shows the comparison with the state-of-the-art. In terms of Mean Absolute Error, our model outperforms [13] but is not as good as [6, 20]. In terms of Median Angular Error, our model achieves 7°, which is 17° lower than [11]. In [14], 16 or more components are needed in their DPM to outperform our method. Note that [14], leverages existing 3D CAD models for the object class of interest. Also, their continuous pose estimation implies a coarse to fine quantization of the viewpoints. Our approach, just from images, jointly provides detection and continuous pose estimation in a regression framework.

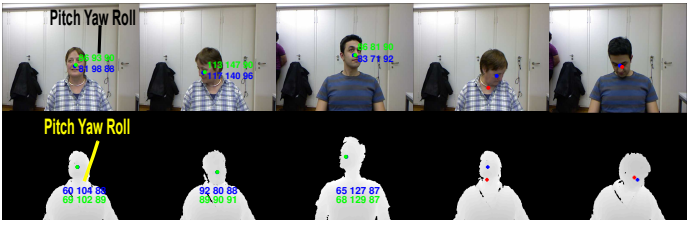
Biwi dataset. We train our model using as positives, the training patches extracted from the head, and as negatives, patches from other body parts. A detection is considered correct if the estimation is within 20 mm from the ground truth location when using depth images, or within the equivalent to 20 mm in pixels when RGB images are used. We report results in Table 3. For depth images, our errors for yaw and pitch are comparable to [4], while our estimation for the roll reduces the error by 2.2°. We report better results for the nose localization, and our missed frames ratio is lower than in [4]. When we remove the detections with lower confidence in pose, up to a missed frame ratio of 6.6% (same ratio as in [4]), we obtain an error for the pitch, yaw and roll of 3.7°, 3.7° and 3°, which outperforms the state-of-the-art. Using RGB images, the difference of the three angle errors, when compared to the results with depth images is lower than 2°. This RGB version reports better detection, reducing the missed frames ratio to 2.4%. That is, our model is able to obtain similar results for head pose estimation problem but simply from RGB images, which we consider an important contribution. Qualitative results for this dataset are shown in Figure 3(b).

We also propose two ways to fuse our RGB and depth HFs. An *early fusion*, where the votes are cast into the same Hough space, and PLEV uses the information of both RGB and depth votes, simultaneously. And the *late fusion* version, where the votes are cast into different Hough spaces. Then the estimated pose is obtained by averaging the PDFs given by the two separate PLEVs. Table 3 shows that the error can be further reduced by combining the RGB and depth forest. Results manifest that the *late fusion* works slightly better than the *early fusion* version.

3.4 Discrete viewpoint classification results

We show results for the car class using the 3D Object categories dataset [17]. The MPPE achieved by our approach is 90.2% (see Table 6), which is on par with several previous

(a) WCV



(b) Biwi

Figure 3: Qualitative results. GT in blue, estimations in green and wrong detections in red.

works. Although our model is a continuous viewpoint model, it achieves also good performance even when it is evaluated on a discrete viewpoint benchmark.

3.5 Exploiting temporal continuity

We end with an extension to video-based pose estimation. Our model casts PDFs as output for pose estimation. This is useful when fusing information from multiple consecutive frames in a video. WCV and the EPFL datasets offer a set of images for cars, which can be considered as video sequences. Given a video, we first estimate the pose for each frame. We define a temporal window of W frames. For a frame j, we aggregate the PDFs of the previous W frames. This aggregated PDF is smoothed to model the change in pose over time, which is assumed to be small in a video. This is fused with the PDF for frame j using a product based ensemble model [2]. This corrected PDF for frame j is incorporated in the temporal window for frame $j+1$. Two parameters are used: the window size W and the smoothing parameter. We do parameter tuning on a validation car instance in each dataset. Results for temporal window sizes of 1 for [11] and 10 for [13] are shown in Table 4. The mean error decreases more than 3° and 10° respectively.

4 Conclusion

We have proposed a new object detection and continuous pose estimation solution using HF. It can successfully detect objects, using depth or RGB images, while the pose is estimated with a probabilistic output using the PLEV. The method reports state-of-the-art results on 4 different datasets, while not relying on the availability of CAD models for training.

Acknowledgements. This work is supported by projects CSI (CCG2013/EXP-047), TIN2010-20845-C03-03, IPT-2012-0808-370000, TEC2013-45183-R and ERC Starting Grant COGNIMUND.

References

[1] L. Breiman. Random forests. In *Machine Learning*, 2001.

[2] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2–3):81–227, 2012.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC)challenge. *IJCV*, 88(2):303–338, 2010.

[4] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool. Random forests for real time 3d face analysis. *IJCV*, 101(3):437–458, 2013.

[5] P. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[6] M. Fenzi, L. Leal-Taixé, B. Rosenhahn, and J. Ostermann. Class generative models based on feature regression for pose estimation of object categories. *CVPR*, 2013.

[7] J. Gall, A. Yao, N. Razavi, L. van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. In *PAMI*, 2011.

[8] J. Gall, N. Razavi, and L. Van Gool. *An Introduction to Random Forests for Multi-class Object Detection*, chapter 11, pages 243–263. Springer, 2012.

[9] A. Ghodrati, M. Pedersoli, and T. Tuytelaars. Is 2D information enough for viewpoint estimation? In *BMVC*, 2014.

[10] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *ICCV*, 2011. ISBN 978-1-4577-1101-5.

[11] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and continuous pose estimation. *IVC*, 30(12):923–933, 2012.

[12] R. J. Lopez-Sastre, T. Tuytelaars, and S. Savarese. Deformable part models revisited: A performance evaluation for object category pose estimation. In *ICCV, 1st Workshop on CORP*, 2011.

[13] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009.

[14] B. Pepik, P. Gehler, M. Stark, and B. Schiele. $3D^2PM - 3D$ deformable part models. In *ECCV*, 2012.

[15] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3D geometry to deformable part models. In *CVPR*, 2012.

[16] H. Riemenschneider, S. Sternig, M. Donoser, P. M. Roth, and Horst Bischof. Hough regions for joining instance localization and segmentation. In *ECCV*, 2012.

[17] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007.

[18] H. Shimazaki and S. Shinomoto. Kernel bandwidth optimization in spike rate estimation. *J. Comput. Neurosci.*, 29(1-2):171–182, 2010.

[19] M. Sun, H. Su, S. Savarese, and L. Fei-Fei. A multi-view probabilistic model for 3D object classes. In *CVPR*, 2009.

[20] M. Torki and A. Elgammal. Regression from local features for viewpoint and pose estimation. *ICCV*, 2011.

[21] Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3D representations for object recognition and modeling. *PAMI*, 35:2608 – 2623, 2013.