See the Difference: Direct Pre-Image Reconstruction and Pose Estimation by Differentiating HOG

Wei-Chen Chiu   Mario Fritz

Max Planck Institute for Informatics, Saarbrücken, Germany

{walon,mfritz}@mpi-inf.mpg.de

Abstract

The Histogram of Oriented Gradient (HOG) descriptor has led to many advances in computer vision over the last decade and is still part of many state of the art approaches. We realize that the associated feature computation is piecewise differentiable and therefore many pipelines which build on HOG can be made differentiable. This lends to advanced introspection as well as opportunities for end-to-end optimization. We present our implementation of ∇HOG based on the auto-differentiation toolbox Chumpy [18] *and show applications to pre-image visualization and pose estimation which extends the existing differentiable renderer* OpenDR [19] *pipeline. Both applications improve on the respective state-of-the-art HOG approaches.*

1. Introduction

Since the original presentation of the Histogram of Oriented Gradients (HOG) descriptor [4] it has seen many use cases beyond its initial target application to pedestrain detection. Most prominently it is a core building block of the widely used Deformable Part Model (DPM) object class detector [9] and exemplar models [23] which both on their own have seen many follow-up approaches. Most recently, HOG-based approaches have repeatedly shown good generalization performance to rendered [1] and artistic images [2], while such type of generalizations are non-trivial to achieve in recently very successful deep learning models in vision [24].

As all feature representations also HOG seek a reduction of information in order to arrive at a more compact representation of the visual input that is more robust to nuisances such as noise and illumination. It is specified as a mapping of an image into the HOG space. The resulting representation is then typically further used in classification or matching approaches to solve computer vision tasks.

While HOG is only defined as a feed-forward computation and introduces an information bottleneck, sometimes
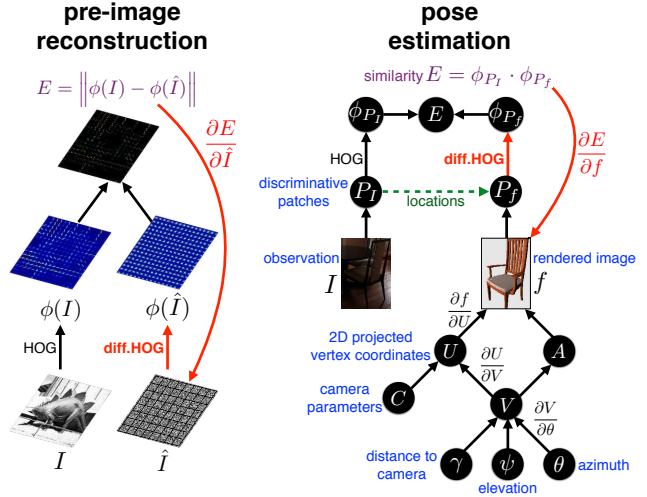

Figure 1: We exploit the piecewise differentiability of the popular HOG descriptor for end-to-end optimization. The figure shows applications on the pre-image reconstruction given HOG features as well as the pose estimation task based on the same idea.

we desire to invert this pipeline for further analysis. E.g. previous work has tried visualize HOG features by solving an pre-image problem [28, 13]. Given a HOG representation of an unobserved input image, the approaches try to estimate an image that produces the same HOG representation and is close to the original image. This has been addressed by sampling approach and approximation of the HOG computation in order to circumvent the problem of the non-invertible HOG computation. Another example, is pose estimation based on 3D models [31, 1, 25, 27] that exploits renderings of 3D models in order to learn a pose prediction model. Here the HOG computation is followed up by a Deformable Part Model [9] or simplified versions that related to the Exemplar Model [23]. Typically, these methods employ sampling based approaches in order to render discrete view-points that are then used in a learning-based scheme

to match to images.

In our work, we investigate directly computing the gradient of the HOG representation which then can be used for end-to-end optimization of the input w.r.t. the desired output. For the visualization via pre-image estimation, we observe the HOG representation and compute the gradient w.r.t. the raw pixels of the input image. For pose estimation we consider the whole pose scoring pipeline of [1] that constitutes a model with multiple parts and a classifier on top of the HOG representation. Here we show how to directly maximize the pose scoring function by computing the gradient w.r.t. the pose parameters. In contrast to the previous approach, we do not reply on pre-rendering views exhaustively and our pose estimation error is therefore not limited by the initial sampling.

We compare to previous works on HOG visualizations and HOG-based pose estimation using rendered images. By using our approach of end-to-end optimization via differentiation of the HOG computation, we improve over the state of the art on both tasks.

2. Related Work

The HOG feature representation is widely used in many computer vision based applications. Despite its popularity, its appearance in the objective functions usually makes the optimization problem hard to operate where it is treated as a non-differentiable function [12, 32]. How to invert the the feature descriptor to inspect its original observation invokes a line of works of feature inversion and feature visualization (pre-image) problem. There are plenty of works on this interesting topic. Given the HOG features of a test image, Vondrick et al. [28] tried in their baseline to optimize the objective with HOG involved by the numerical derivatives but failed to get reasonable results, thus in their proposed method the inversion is done by learning a paired dictionary of features and the corresponding images. Weinzaepfel et al. [30] attempted to reconstruct the pre-image of the given SIFT descriptors [21] based on nearest neighbor search in a huge database of images for patches with the closet descriptors. Kato et al. [13] study the problem of pre-image estimation of the bag-of-words features and they rely on a large-scale database to optimize the spatial arrangement of visual words. Although these and other related works provide different ways to approximately illustrate the characteristic of the image features, we nearly have not seen the work directly addressing the differentiable form of the feature extraction procedure. In contrast, our approach contributes to make the differentiation of HOG descriptor practical such that it can be easily plugged into the computer vision pipeline to enable direct end-to-end optimization and extension to hybrid MCMC schemes [15, 16]. One most relevant work to ours is from Mahendran et al. [22], which inverts feature descriptors (HOG [9], SIFT [21],

and CNNs [14]) for a direct analysis of the visual information contained in representations, where HOG and SIFT are implemented by Convolutional Neural Networks (CNNs). However, their approach contains an approximation to the orientation binning stage of HOG/SIFT and includes two strong natural image priors in the objective function with some parameters need to be estimated from training set. Instead in our work, we do not have any approximation in the HOG pipeline and no training is needed.

Despite deep-learning based features are in fashion these years, there are plenty of applications using HOG, in particular the Examplar LDA [11] for the pose estimation task with rendered/CAD data [1, 17, 3]. In [6], slightly-modified SIFT (gradient-histogram-based as HOG) can beat CNNs in feature matching task. In this paper, we specifically demonstrate the application of our ∇HOG on the pose estimation problem for aligning 3D CAD models to the objects on 2D real images, we briefly review some recent research works here. Lim et al. [17] assume the accurate 3D CAD model of the target object is given, based on the discretized space of poses for initialization they estimate the poses from the correspondences of LDA patches between the real image and the rendered image of CAD model. Aubry et al. [1] create a large dataset of CAD models of chair objects, with rendering each CAD model from a large set of viewpoints they train the classifiers of discriminative exemplar patches in order to find the alignment between the chair object on the 2D image and the most similar CAD model of the certain rendering pose. In additional to the discrete pose estimation scheme as [1], there has been works on continuous pose estimation [26, 3, 25]. For instance, Pepik et al. [25] train a continuous viewpoint regressor and also the RCNN-based [10] key-point detectors which are used to localize the key-points on 2D images in an object class specific fashion, with the correspondences between the key-points on the 2D image and 3D CAD model, they estimate the pose of the target object. However, for these current state-of-the-art approaches most of them need to collect plenty of data to train the discriminative visual element detectors or key-point detectors for the matching, or to render many images of CAD models of various viewpoints in advance. Instead, our proposed method manages to combine the ∇HOG based exemplar LDA model with the approximate differentiable renderer from [19] which enable us to have directly end-to-end optimization for the pose parameters of the CAD model in alignment with the target object on the real images.

3. ∇HOG

Here we describe how we achieve the derivative of the HOG descriptor. In the original HOG computation, there are few sequential key-components, including 1) computing gradients, 2) weighted vote into spatial and orientation cells, and 3) contrast normalization over overlapping spatial

blocks. In our implementation we follow the same procedure. For each part we argue for piecewise differentiability. The differentiability of the whole pipeline follows from the chain rule of differentiation. Figure 2 shows an overview of the computations involved in the HOG feature computation pipeline which we describe in details in the following.

3.1. Gradients Computation

If a color image $I \in \mathbb{R}^{w \times h \times 3}$ is given, we first compute its gray-level image:

$$I_{\text{gray}} = I(:,:,0)*0.299 + I(:,:,1)*0.587 + I(:,:,2)*0.114 \tag{1}$$

Then we follow the best setting for gradient computation as in Dalal's approach [4], to apply the discrete derivative 1–D $[-1, 0, 1]$ masks on both horizontal and vertical directions without Gaussian smoothing. We denote the gradient maps on horizontal and vertical directions as G_x and G_y, while the magnitude $\|\nabla\|$ and direction Θ of gradients can be computed by:

$$\begin{aligned} \|\nabla\| &= \sqrt{G_x^2 + G_y^2} \\ \Theta &= \arctan(G_y, G_x) \end{aligned} \tag{2}$$

Note that here we use unsigned orientations such that the numerical range of the elements in $\|\nabla\| \in [0, 180]$. The $L2$ norm is denoted by $\|\cdot\|$ through this paper for consistency.

Differentiability: The conversion to gray as well as the derivative computation via linear filtering are linear operations and therefore differentiable. arctan is differentiable in $\mathbb{R}$ and the gradient magnitude $\|\nabla\|$ is differentiable due to the chaining of the differentiable squaring function and the square root over values in $\mathbb{R}^+$.

3.2. Weighted vote into spatial and orientation cells

After we have the magnitude and direction of the gradients, we proceed to do the weighted vote of gradients into spatial and orientation cells which provides the fundamental nonlinearity of the HOG representation. The cells are the local spatial regions where we accumulated the local statistics of gradients by the histogram binning of their orientations. Assume we divide the image region into $N_w^c \times N_h^c$ cells of size $c_w \times c_h$, for each pixel located within the cell we compute the weighted vote of its gradient orientation to an orientation histogram (In our setting we use the same setting as Dalal's to have the histogram of 9 bins spaced over $0° - 180°$ which ignores the sign of the gradients).

Normally for each cell its orientation histogram is represented in a 1–D vector of length B (number of bins), but this operation will miss the positions of the pixels which contribute to the histogram. This does not lead to a formulation that allows for derivation of the HOG representation

with respect to different pixel positions. Our main observation here is to view each orientation binning as a filter f_b^o applied to each location in the gradient map. We store the filtered results in $F^o \in \mathbb{R}^{w \times h \times B}$. Analogously, the pixel-wise orientational filters $\{f_b^o\}_{b=1 \cdots B}$ are chosen to follow the bi-linear interpolation scheme of the gradients in neighboring orientational bins:

$$\begin{aligned} f_b^o(\Theta) &= \text{clip}_{min=0}^{max=1}(1 - |\Theta - \mu_b| \times \frac{B}{180}) & (3) \\ F_b^o &= \|\nabla\| \odot f_b^o(\Theta), \quad \forall b \in 1 \cdots B & (4) \end{aligned}$$

where μ_b is the central value of orientation degree for filter f_b^o, $\text{clip}_{min=0}^{max=1}$ function clamps the numerical range between 1 and 0, and $\odot$ is an element-wise multiplication. (Note that for the first filter f_1^o we also take care of the numerical range. See the visualization shown in Figure 2.)

We have the F^o for orientational binning, we then apply spatial binning for each cell. Here as in the Dalal's method, to reduce the aliasing, for each pixel location it will contribute to its 4 neighboring cells proportional to the distances to the centers of those cells, in another word, the votes are interpolated bilinearly. Following the similar trick as in orientational binning, by creating a $2c_w \times 2c_h$ bilinear filter f^s where its maximum value 1 is in the center with decreasing values toward four corners to minimum value 0, as shown in Figure 2, we convolve it with all F_b^o to get the spatial filtered results F_b^s:

$$F_b^s = F_b^o * f^s, \quad \forall b \in 1 \cdots B \tag{5}$$

then the spatial binning for cells can be easily fetched from:

$$F_b^s(x, y | x \in \mathcal{X}, y \in \mathcal{Y}), \quad \forall b \in 1 \cdots B \tag{6}$$

where $(\mathcal{X}, \mathcal{Y})$ are the (x, y) coordinates of the centers for all cells.

Note that till here when you concatenate $v = \{F_b^s(x, y | x \in \mathcal{X}, y \in \mathcal{Y})\}_{b=1 \cdots B}$ then actually we get exactly the same representation as from original HOG approach.

Differentiability By re-representing the data, we have shown that the histograming and voting procedure of the HOG approach can be viewed as linear filtering operations followed up by a summation. Both steps are differentiable.

3.3. Contrast normalization

In the original procedure of Dalal's HOG descriptor, contrast normalization is performed on every local region of size 3×3 cells, which they call *blocks*. As many recent applications that we are interested in [1, 2, 13, 28, 9] do not use blocks, we do not consider them either in our implementation. While this step is possible to incorporate, it would also lead to increased computational costs due to
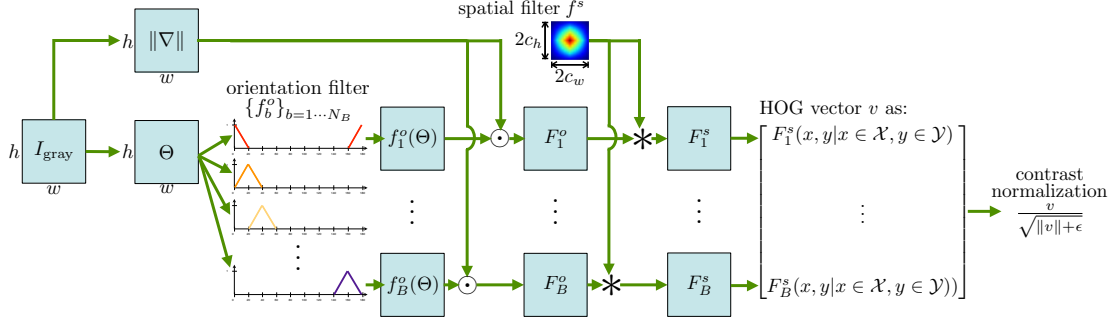
Figure 2: Visualization of the implementation procedure for our ∇HOG method.

multiple representation of the same cell. We instead only use the global normalization by using the robust $L2\text{-}norm$. Given the HOG representation v from previous steps, the global contrast normalization can be written as:

$$v_{\text{normalized}} = \frac{v}{\sqrt{\|v\| + \epsilon}} \qquad (7)$$

where ϵ is a small positive constant.

Differentiability: This is a chain of differentiable functions and therefore the whole expression is differentiable.

Difference to Original HOG While there is a large diversity of HOG implementations available by now, we summarize the two main difference to the standard one as proposed in [4]: First, the original HOG compute the the gradients on different color channels and apply the maximum operator on the magnitudes over all channels to get the gradient map. In our implementation we simply first transform the color image into gray scale and compute the gradient map directly. Second, we do not include the local contrast normalization for every overlapping spatial blocks. But we do include the global, robust $L2$ normalization.

3.4. Implementation

In the above equations (Eq. 1, 2, 3, 5, 7) all the operations are (piecewise-) differentiable (summation, multiplication, division, square, square root, arc-tangent, clip), with the use of the chain rule, our overall HOG implementation is differentiable on each pixel position. Overall, this is not surprising as visual feature representations are designed to vary smoothly w.r.t. to small changes in the image. We have implemented this version of the HOG descriptor by using the Python-based auto-differentiation package *Chumpy* [18], which evaluates an expression and its derivatives with respect to its inputs. The package and our extension integrate with the recently proposed Approximate Differentiable Renderer OpenDR [19]. We will make our implementation publicly available in the near future.

4. Experimental Results

4.1. Reconstruction from HOG descriptors

We evaluate our proposed ∇HOG method on the image reconstruction task based on the feature descriptors. We are interested in this task since it provides a way to visualize the information carried by the feature descriptors and open the opportunity to examine the feature descriptor itself instead of based on the performance measures of certain tasks as proxies. There is already prior work on this problem. [13, 28, 5] focus on different feature representations such as Bag-of-Visual-Words (BoVW), Histogram of Orientated Gradients (HOG), and Local Binary Descriptors (LBDs). However, state-of-the-art approaches typically need to use large-scale image bases for learning the reconstruction.

Objective As we have derived the gradient of the HOG feature w.r.t. the input, we can – given a feature vector – directly optimize for the reconstruction of original image without any additional data needed. To define the problem more formally, let $I \in \mathbb{R}^{X \times Y}$ be an image and its HOG representation as $\phi(I)$, we optimize to find the reconstructed image $\hat{I}$ whose HOG features $\phi(\hat{I})$ have the minimum euclidean distance E to $\phi(I)$:

$$\hat{I} = \operatorname*{argmin}_{\hat{I} \in \mathbb{R}^{X \times Y}} E$$
$$= \operatorname*{argmin}_{\hat{I} \in \mathbb{R}^{X \times Y}} \left\| \phi(I) - \phi(\hat{I}) \right\| \qquad (8)$$

The option to approach the problem in this way was mentioned in [28], however there was no result achieved as numerical differentiation is very computational expensive in this setting. Direct optimization is facilitated for the first time using our ∇HOG approach.

An overview of our approach is shown in Figure 1. We compute derivatives $\frac{\partial E}{\partial i_{x,y}}$ with respect to the intensity values $i_{x,y}$ of all the pixel positions (x, y) on $\hat{I}$ via auto-differentiation. By gradient-based optimization we are able

to find a local minimum of E and corresponding reconstructed image $\hat{I}$. In order to regularize our estimation, we introduce a smoothness prior that penalizes gray value changes of adjacent pixels. Intuitively, this encourages propagation of information into areas without strong edges for which no signal from the HOG features is available.

$$\hat{I} = \underset{\hat{I} \in \mathbb{R}^{X \times Y}}{\operatorname{argmin}} \left\| \phi(I) - \phi(\hat{I}) \right\| + \xi \sum_{p,q \in \mathcal{N}} \| i_p - i_q \| \qquad (9)$$

where $p, q \in \mathcal{N}$ means that pixel p and q are neighbors, and ξ is the weight for the smoothness term which we usually set to a big number as 10^2 in our experiments. Although we give a high weight for the smoothness term, it will only play a key role in the first few iterations of the optimization procedure then the euclidean distance E will dominate to find the local minimum.

The evaluation is based on the image reconstruction dataset proposed in [13] which contains 101 images for all the categories from Caltech 101 dataset [8] and all have a resolution of 128×128. We compare our method with few state-of-the-art baselines on image reconstruction from feature descriptions: the **BoVW** method from [13], the **HOGgles** method from [28], also **CNN-HOG** and **CNN-HOGb**(CNN-HOG with bilinear orientation assignments) from [22].

Note that our ∇HOG described in Section 3 is based on **Dalal's**-type HOG[4], while for HOGgles/CNN-HOG/CNN-HOGb baselines they are using **UoCTTI**-type HOG[9] which additionally considers directed gradients. To have a fair comparison, we also implement UoCTTI HOG under our proposed framework.

We propose two additional variants for reconstruction that exploit multi-scale information as shown in Figure 3.

∇HOG multi-scale We use the single scale HOG descriptor as input, but we first reconstruct $\hat{I}_{\frac{1}{s}}$ with s times smaller resolution than I (the cell size for $\phi(\hat{I}_{\frac{1}{s}})$ is $\frac{1}{\sqrt{s}}$ of the original one used for $\phi(I)$, $s \in \{4, 16, 64\}$ in our experimental setting.). After few iterations of updates in optimization process, we up-sample $\hat{I}_{\frac{1}{s}}$ to higher resolution and continue the reconstruction procedure. These steps are repeated until we reach the initial resolution of I.

∇HOG multi-scale-more We use the multi scale HOG vectors of the original image I as the input. For the reconstruction on different scale s, the corresponding HOG descriptor $\phi(I_{\frac{1}{s}})$ extracted on the same scale is used in the euclidean distance E, as shown in Figure 3(b). As additional HOG descriptors are computed from the original image at different scales, we use more information than in the original setup and therefore the results of the "multi-scale-more" approach cannot be directly compared to prior works.

(a) ∇HOG multi-scale   (b) ∇HOG multi-scale-more
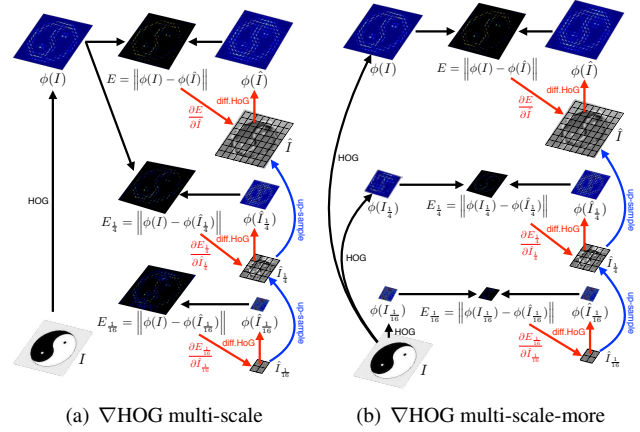
Figure 3: Visualizations of variants of the proposed method for the task of image reconstruction from feature descriptors.

The optimization is done based on the nonlinear optimization using Powell's dogleg method [20] which is implemented in *Chumpy* [18]. Example results of the multi scale approaches can be seen in Table 1.

	test	1/64	1/16	1/4	1/1
(a)					
(b)					

	test	iter-0	iter-1	iter-2	iter-final
(c)					
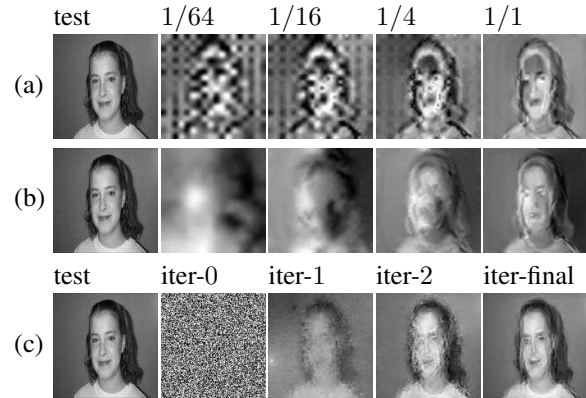
Table 1: Example results for (a)(b) ∇**HOG multi-scale** and ∇**HOG multi-scale-more** in which both are based on Dalal-HOG[4]; and (c) for ∇**HOG** on UoCTTI-HOG[9].

Results In order to quantify the performance of image reconstruction, different metrics have been proposed in prior works. For instance, in [13] the mean squared error of raw pixels is utilized, while in [28] the cross-correlation is chosen to compare the similarity between the reconstructed image and the original one. In addition to using cross-correlation as the metric for qualitative evaluation, we also investigate different choices used by the research works on the problem of image quality assessment (IQA), includ-

ing mutual information and **Structural Similarity (SSIM)** [29]. In particular, mutual information measures the mutual dependencies between images hence gives another metric for similarities, while SSIM measures the degradation of structural information for the distorted/reconstructed image from the original one, under the assumption that human visual perception is adapted to discriminate the structural information from the image.

We report the performance numbers from all the metrics in Table 2. The proposed method using UoCTTI-type HOG outperforms the state-of-the-art baselines by a large margins for all metrics. Visually inspected, our proposed method can reconstruct many details in the images and also give accurate estimate on gray-scale values if using UoCTTI HOG. Please note again, our method does not need any additional data for training while in baselines training is necessary.

4.2. Pose estimation

We also evaluate our ∇HOG approach on a pose estimation task where 3D CAD models have to be aligned to objects in 2D images. We build on openDR [19] which is an approximate differentiable renderer. It parameterizes the forward graphics model f based on vertices locations V, per-vertex brightness A and camera parameters C, which is shown on the left part of Figure 5, where U is for the $2D$ projected vertex coordinate position. Based on the auto-differentiation techniques, openDR provides a way to derive the derivatives of the rendered image observation with respect to the parameters in the rendering pipeline.

Approach We extend openDR in the following ways as illustrated in Figure 5: **1)** We parameterize the vertices locations V of CAD models by three parameters: azimuth θ, elevation ψ, and distance to the camera γ; **2)** During the pose estimation procedure, as in [1], the matching between the objects on real images and the rendered images from the CAD models are addressed by the similarities between the HOG descriptors of the visual discriminative elements extracted from them. The detailed procedure of extracting visual discriminative elements is discussed in [1]. In our method, we use our ∇HOG method $\phi(P_f)$ for the image patches P_f which have the same regions as the visual elements P_I extracted from the test image I, and the similarity between the P_f and P_I is the dot product between HOG descriptors $\phi(P_I)$ of P_I and $\phi(P_f)$. As shown in the right part of Figure 5 this similarity can be traversed back to the pose parameters $\{\theta, \psi, \gamma\}$ and the derivatives of the similarity with respect to the pose parameters can be again computed by the auto-differentiation, our method can directly optimize to maximize the similarity to estimate the poses.

Setup We follow the same experimental setting as [1], where we test on the images annotated with no-occlusion,
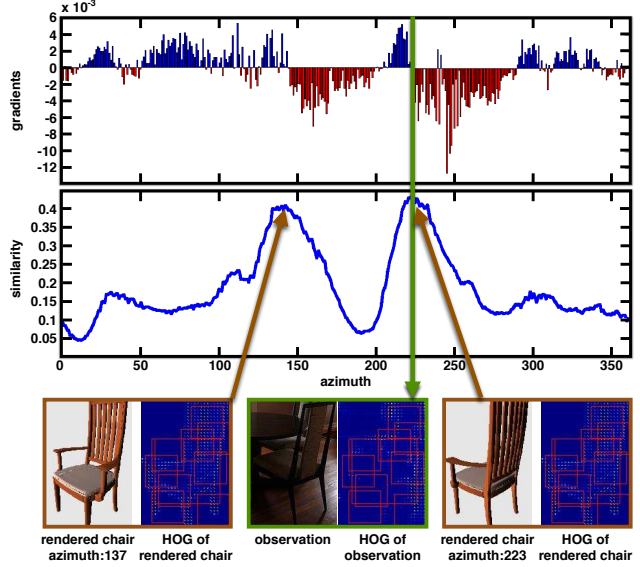
Figure 4: Visualization of the similarity and its gradients w.r.t azimuth. The red boxes on the HOG representations are the visual discriminative patches.

no-truncation and not-difficult of the chairs validation set on PASCAL VOC 2012 dataset [7], therefore in total 247 chairs from 179 images are used for the evaluation. To purely focus on evaluation of the pose estimation, we extract the object images based on their bounding boxes annotation, and resize them to have at least length of 100 pixels on the shortest side of image size.

The baseline [1] is applied on the chair images to search over a chair CAD database of 1393 models which includes the rendered images from 62 different poses relative to camera for each of them, then to detect the chairs, match the styles of the chairs, and simultaneously recover their poses based on rendered images. We select the most confident detection for each chair together with the estimated pose.

We apply our proposed method on pose estimation by using the elevation and azimuth estimates of [1] as a initialization of pose, and add few more initializations for azimuth (8 equidistantly distribute over $360°$). We use gradient descent method with momentum term for optimization in order to optimize for the azimuth parameter and interleave iterations in which we additionally optimize for the distance to camera. In Figure 4 we visualize an example of the similarity between the chair object on the real image and the CAD model on the rendered image, as well as its gradients w.r.t azimuth θ (full $360°$). We can see how gradients change related to different local maximums and the corresponding poses of the CAD model.

Method		cross correlation	mutual information	structural similarity (SSIM) [29]
	BoVW [13]	0.287	1.182	0.252
UoCTTI HOG [9]	HOGgles [28]	0.409	1.497	0.271
	CNN-HOG [22]	0.632	1.211	0.381
	CNN-HOGb [22]	0.657	1.597	0.387
	our ∇HOG (single scale)	**0.760**	**1.908**	**0.433**
Dalal's HOG [4]	our ∇HOG (single scale)	0.170	1.464	0.301
	our ∇HOG (multi-scale: $\frac{1}{64}$)	0.058	1.444	0.121
	our ∇HOG (multi-scale: $\frac{1}{16}$)	0.076	1.470	0.147
	our ∇HOG (multi-scale: $\frac{1}{4}$)	0.108	1.458	0.221
	our ∇HOG (multi-scale: $\frac{1}{1}$)	0.147	1.478	0.293
	our ∇HOG (multi-scale-more: $\frac{1}{64}$)	0.147	1.458	0.251
	our ∇HOG (multi-scale-more: $\frac{1}{16}$)	0.191	1.502	0.291
	our ∇HOG (multi-scale-more: $\frac{1}{4}$)	0.220	1.565	0.320
	our ∇HOG (multi-scale-more: $\frac{1}{1}$)	**0.236**	**1.582**	**0.338**

Table 2: Comparison on the performance of reconstruction from feature descriptors.
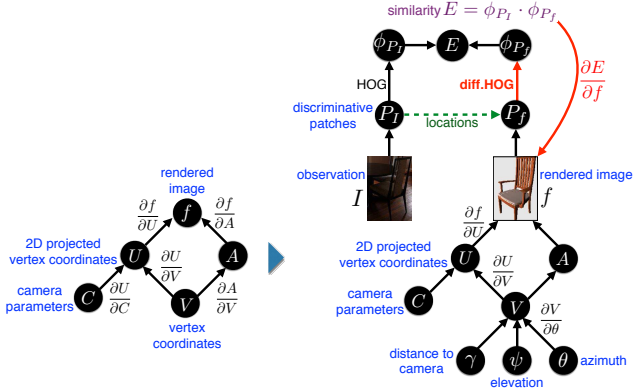
Figure 5: (left) The differentiable rendering procedure from openDR [19]. (right) The visualization of our model for pose estimation.

Results In order to quantify our performance on pose estimation task, we use the continuous 3D pose annotations from PASCAL3D+ dataset [31]. Following the same evaluation scheme, the view-point estimation is considered to be correct if its estimated viewpoint label is within the same interval of the discrete viewpoint space as the ground-truth annotation, or its difference with ground-truth in continuous viewpoint space is lower than a threshold. We evaluate the performance based on various settings of the intervals and thresholds in viewpoint space: $\{4\,\text{views}/90°, 8\,\text{views}/45°, 16\,\text{views}/22.5°, 24\,\text{views}/15°\}$. In Table 3 we report the performance numbers for Aubry's baseline and our proposed approach. We are outperforming the previous best performance up to 10% points on the coarse and fine measures. Some example results which

show improvements of the baseline method are shown in Table 4.

Discussion One advantage of our proposed method is that we are able to parameterize the vertexes coordinates of the CAD models by the same pose parameters as used in [1], then the differentiable rendering procedure provided by openDR [19] and our ∇HOG representations enable us to directly compute the derivatives of the similarity with respect to the pose parameters, and optimize for continuous pose parameters. In another word, for the proposed approach we do not need to discretize the parameters as [1] and do not need to render images from many poses in advance for the alignment procedure either.

	4 views	8 views	16 views	24 views
Aubry et al. [1]	47.33	35.39	20.16	15.23
our method	58.85	40.74	22.22	16.87

Table 3: Pose estimation results based on the groundtruth annotation from PASCAL3D+ [31].

5. Conclusions

We investigate the feature extraction pipeline of HOG descriptor and exploit its piecewise differentiability. Based on the implementation using auto-differentiation techniques, the derivatives of the HOG representation can be directly computed. We study two problems of image reconstruction from HOG features and HOG-based pose estimation while the direct end-to-end optimization becomes practical with our ∇HOG. We demonstrate that our ∇HOG-based approaches outperforms the state-of-the-art baselines

			HOGgles [28]	CNN-HOG [22]	CNN-HOGb [22]	Our ∇HOG (single-scale)	Our ∇HOG (single-scale)	Our ∇HOG (multi-scale)	Our ∇HOG (multi-scale-more)
Example	HOG	BOVW [13]	UoCTTI-HOG	UoCTTI-HOG	UoCTTI-HOG	UoCTTI-HOG	Dalal-HOG	Dalal-HOG	Dalal-HOG

Table 5: Example results for image reconstruction from feature descriptors.

for both problems. We have demonstrated that the approach can lead to improved introspection via visualizations and improved performance via direct optimization through a whole vision pipeline. Our implementation is integrated into an existing auto-differentiation package as well as the recently proposed Approximately Differentiable Renderer OpenDR [19] and is publicly available. Therefore it is easy to adopt to new tasks and is applicable to a range of end-to-end optimization problems.

6. Acknowledgement

References

[1] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014. 1, 2, 3, 6, 7, 8

Table 4: Example results for pose estimation.

test images									
Aubry et al. [1]									
our method									

[2] M. Aubry, B. C. Russell, and J. Sivic. Painting-to-3d model alignment via discriminative visual elements. *ACM Transactions on Graphics (TOG)*, 33(2):14, 2014. 1, 3

[3] C. B. Choy, M. Stark, S. Corbett-Davies, and S. Savarese. Enriching object detection with 2d-3d registration and continuous viewpoint estimation. In *CVPR*, June 2015. 2

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 3, 4, 5, 7

[5] E. d'Angelo, A. Alahi, and P. Vandergheynst. Beyond bits: Reconstructing images from local binary descriptors. In *ICPR*, 2012. 4

[6] J. Dong and S. Soatto. Domain-size pooling in local descriptors: Dsp-sift. In *CVPR*, June 2015. 2

[7] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 6

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVIU*, 2007. 5

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010. 1, 2, 3, 5, 7

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[11] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*. 2012. 2

[12] D. Huang, Y. Tian, and F. De la Torre. Local isomorphism to solve the pre-image problem in kernel methods. In *CVPR*, 2011. 2

[13] H. Kato and T. Harada. Image reconstruction from bag-of-visual-words. In *CVPR*, 2014. 1, 2, 3, 4, 5, 7, 8

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[15] T. D. Kulkarni, P. Kohli, J. B. Tenenbaum, and V. Mansinghka. Picture: A probabilistic programming language for scene perception. In *CVPR*, 2015. 2

[16] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum. Deep convolutional inverse graphics network. *arXiv:1503.03167 [cs.CV]*, 2015. 2

[17] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *ICCV*, 2013. 2

[18] M. Loper. Chumpy. https://github.com/mattloper/chumpy. 1, 4, 5

[19] M. M. Loper and M. J. Black. Opendr: An approximate differentiable renderer. In *ECCV*. 2014. 1, 2, 4, 6, 7, 8

[20] M. I. Lourakis and A. A. Argyros. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *ICCV*, 2005. 5

[21] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 2

[22] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015. 2, 5, 7, 8

[23] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 1

[24] X. Peng, B. Sun, K. Ali, and K. Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. *arXiv:1412.7122 [cs.CV]*, 2015. 1

[25] B. Pepik, M. Stark, P. Gehler, T. Ritschel, and B. Schiele. 3D Object Class Detection in the Wild. *arXiv:1503.05038 [cs.CV]*, 2015. 1, 2

[26] H. O. Song, M. Fritz, C. Gu, and T. Darrell. Visual grasp affordances from appearance-based cues. In *IEEE Workshop on Challenges and Opportunities in Robot Perception*, 2011. 2

[27] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC*, 2010. 1

[28] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. HOGgles: Visualizing Object Detection Features. In *CVPR*, 2013. 1, 2, 3, 4, 5, 7, 8

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 6, 7

[30] P. Weinzaepfel, H. Jégou, and P. Pérez. Reconstructing an image from its local descriptors. In *CVPR*, 2011. 2

[31] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014. 1, 7

[32] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013. 2