EXPLICIT LEARNING OF FEATURE ORIENTATION ESTIMATION

Ji Dai, Junkang Zhang, Truong Nguyen

University of California San Diego

ABSTRACT

While many learning-driven algorithms for local feature detection and description have submerged during recent years. One key component in the pipeline, namely orientation estimation, still remains underdeveloped. Among all sorts of difficulties, the impracticality and tedium of finding a ”ground truth” feature orientation as a learning target is one big challenge. In this paper, we bypass this ”thinking trap” and propose an *unsupervised* scheme that *explicitly* trains a simple convolutional neural network to predict orientations for feature points. Together with a carefully designed loss term, the network manages to provide accurate orientation estimations. We further evaluate the capability of this estimator in two experiments: orientation estimation and feature matching. Results showed the proposed method outperforms other compared methods on multiple benchmark datasets. The pretrained model is publicly available.[1]

***Index Terms*—** Feature orientation, unsupervised learning, explicit learning, feature matching

1. INTRODUCTION

Local feature extraction is a powerful and omnipresent tool in computer vision, serving as the foundation of many algorithms and applications. A standard pipeline includes three consecutive procedures. First, at multiple scales, detect keypoints with potentially rich and distinguishable textural information around, typically corners or areas with significant gradient variations. Second, estimate the orientations of the keypoints using surrounding local geometric information, with dominant gradient as one example. These orientations will be used later to rectify the descriptions and ensure the extracted features to be rotational-invariant. Third, generate meaningful descriptions for the keypoints, usually based on patches centering at the keypoints with radii proportional to the scales mentioned in first step.

For many years, handcrafted feature detectors [1, 2, 3, 4] and descriptors [5, 3, 4, 6] dominated the field and achieved very decent results, with SIFT [3] being one of the most cited computer vision papers. Recently, as deep learning revisiting many computer vision problems, researchers started to investigate its power in local feature extraction. Both feature detectors [7] and descriptors [8, 9] have received performance gain by incorporating neural network into the solutions. Works using deep learning for the entire feature extraction pipeline [10, 11] also show their strength in the caveats of handcrafted algorithms such as wide baseline matching problem. However, as a major component in the pipeline, the orientation estimation hasn't received enough attention. Most learning algorithms either hope the descriptor network will learn to be rotational-invariant by itself with massive training data [8, 12], or they simple use a handcrafted method as guideline and train a regressor to approximate it [11]. In fact, we only find works from Yi et al. [13, 10] address this problem properly. We believe the difficult of finding a canonical orientation as ground truth is one obstacle that impedes the research.

In this paper, we propose an unsupervised scheme that averts this ”thinking trap” and explicitly trains a regression network to estimate the orientations for feature points. For each feature point, we first sample an upright patch surround it and ask network to estimate an orientation. We then sample another patch at the same location but apply rotation with some known angle, and ask the network to give another estimation. We supervise the network by comparing the difference between two estimations and this known angle. To deal with the periodicity property of angle, we carefully design a loss function which confines the range of output estimations.

We evaluate the proposed network on multiple benchmark dataset in two experiments: orientation estimation and feature matching. The proposed method manifests strong performances in both experiments compared with other handcrafted and learned methods.

2. RELATED WORKS

Feature extraction has been studied for many years. A comprehensive performance comparison of methods can be found in [14]. Here, we will focus solely on orientation estimation algorithms. We first offer a brief introduction of various handcrafted methods. We then shift to deep learning paradigm and cover two most related works by Yi et al.

2.1. Handcrafted Orientation Estimators

Most handcrafted methods count on finding a reliable dominant orientation within a patch surround the feature

[1] https://github.com/jidai-code/orientation_estimation

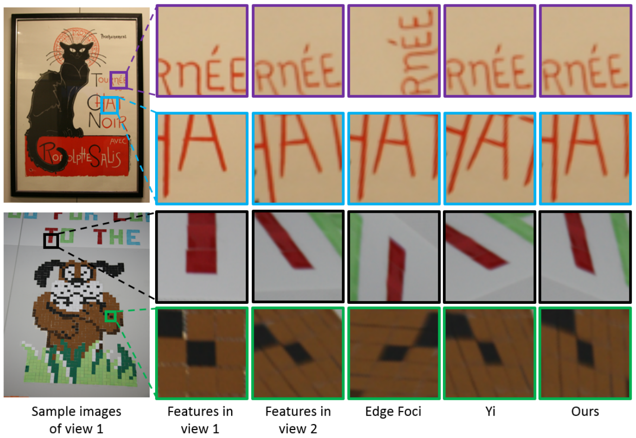

Fig. 1. Performance comparison of three orientation estimation methods. **Column 2** and **3** show the corresponding features from two different views. **Column 4, 5, 6** show the features from view 2 being rotated back according to the orientations estimated by Edge Foci [2], Yi [13], and our method. Ideally, the rotated features from view 2 should have same orientations as features from view 1 (**column 2**). As shown above, our method has the most consistent performance.

point. The ways of finding such dominant orientation vary from methods to methods. SIFT [3] uses histograms of gradient orientations for the entire patch to decide the dominant orientation. 3D-SIFT [15] inherits this core idea and extends it to 3D field. SURF [4] applies guassian weights to the wavelet responses in horizontal and vertical directions, and from which, determines the orientation for the patch. ORB [16] introduces a simple but effective measure, called *intensity centroid*. It uses image moments to compute the center of mass as well as the main orientation. Generally these handcrafted orientation estimators are reliable. However, they are prone to giving inaccurate estimations under noisy conditions.

2.2. Learning Based Orientation Estimators

Yi et al. [13] propose to train a feature orientation estimator within the pipeline of feature matching. They first pick some existing feature detector and descriptor, then they substitute the orientation estimation part with their network. Next, they use this entire pipeline to detect and match features between a pair of images. The orientation network is trained using the loss related to matching accuracy. To further support this training scheme, they propose a new activation function named Generalized Hinging Hyperplanes (GHH). Comparing to other activation functions like ReLU or PReLU [17], the authors claim GHH to preserve more gradients throughout this long pipeline. They extend the idea in [10], where they proposed an end-to-end network combining detector, orientation estimator and descriptor together. According to Yi, in the previous training scheme, some existing description methods cannot backpropagate enough gradients, leading to performance drop. By designing the entire pipeline with neural network, they show further improvement in matching accuracy.

3. METHOD

3.1. Challenges in Orientation Estimation

As described previously, explicitly learn to estimate feature orientation is challenging. In fact, even the notion of feature orientation might not be a valid concept per se, as it is hard to assign a canonical pose to a feature point. One possible approach is to pick one handcrafted estimator as mentioned in Sec. 2.1 and train a network to approximate it. Yet, it is debatable whether local structural information captured by handcrafted kernels is a good representation for orientation. Especially when we are dealing with large baseline problems, the accuracy will degrade dramatically due to substantial viewpoint and illumination change.

Another approach is to train the orientation implicitly like [13]. However, as described in Sec. 2.2, this approach doesn't work for all descriptors. The performance also degrades when the testing descriptor is different from the one used in training.

In this paper, we introduce a scheme that explicitly trains the network to predict orientations of feature points. The proposed method doesn't rely on any handcrafted kernels and is perfectly compatible with all descriptors.

3.2. Unsupervised Learning Scheme

As illustrated in Fig. 2, given a training image I, we first use Edge Foci detector [2] to extract top 1000 feature points with highest response. Each feature point comes with a corresponding coordinate (x, y) in image and a radius r. We then extract an upright 32×32 patch P_1 centering at $I(x, y)$ with size r. Usually $r \neq 32$, and this process needs up/downsampling. Next, we sample another 32×32 patch P_2, again centering at $I(x, y)$ with size r, but with a random rotation $\theta_{\text{gt}} \in (-\pi, \pi]$. P_1, P_2 are passed to the network G and output two predicted angles $G(P_1) = \theta_1, G(P_2) = \theta_2$. We train the network G by comparing the difference between $\theta_2 - \theta_1$ and θ_{gt}.

3.3. Loss Function

The periodical property of angle (θ and any $\theta + 2k\pi$ are essentially same angle) needs special care when designing loss functions. As indicated in [13], simply letting alone periodicity in loss function will lead to multiple local minima. We overcome this hurdle by proposing a loss function with two terms: range loss $\mathcal{L}_r$ and prediction loss $\mathcal{L}_p$ (Eq. 1). λ is set

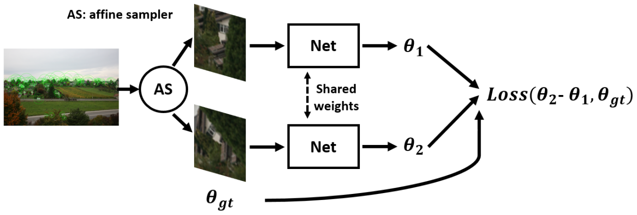

Fig. 2. Unsupervised training pipeline. We first detect features using Edge Foci detector. At each feature point, we sample two patches, one being upright and the other with some angle θ_{gt}. Network was supervised by comparing $\theta_2 - \theta_1$ and θ_{gt}.

to 10^{-1} in this paper.

$$\mathcal{L}(\theta_1, \theta_2, \theta_{gt}) = \mathcal{L}_r(\theta_1) + \mathcal{L}_r(\theta_2) + \lambda \mathcal{L}_p(\theta_2 - \theta_1, \theta_{gt}) \quad (1)$$

The range loss $\mathcal{L}_r$, as defined in Eq. 2, penalizes when $|\theta| > \pi$.

$$\mathcal{L}_r(\theta) = \max(|\theta| - \pi, 0) \quad (2)$$

The prediction loss $\mathcal{L}_p$ measures the distance between $\theta_2 - \theta_1$ and θ_{gt}. Since we constrain $|\theta| \leq \pi$, the range of $\theta_{\text{diff}} = \theta_2 - \theta_1$ will fall between -2π to 2π. Therefore, we first mod it to $[-\pi, \pi]$ as $\hat{\theta}_{\text{diff}}$. We also need to make sure that distance between $\hat{\theta}_{\text{diff}}$ and θ_{gt} is $< \pi$ (distance between 30° and 330° is not 300° but 60°). We present the detail of $\mathcal{L}_p$ in Eq. 3.

$$\mathcal{L}_p(\theta_{\text{diff}}, \theta_{gt}) = \min\left(|\hat{\theta}_{\text{diff}} - \theta_{\text{gt}}|, 2\pi - |\hat{\theta}_{\text{diff}} - \theta_{\text{gt}}|\right) \quad (3)$$

3.4. Network Architecture

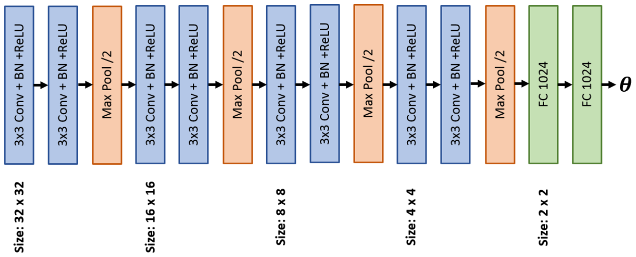

Fig. 3. Architecture of proposed network

We show the details of proposed orientation network in Fig. 3.

4. RESULTS

4.1. Datasets

We train the proposed network on HPatch dataset [18], and evaluate its performance on Edge Foci [2], Viewpoints [13], and Webcam [19] datasets. All these datasets serve as benchmarks for matching local features. They each contains a number of image sets of different contents. Each set includes multiple images capturing the same scene from different perspectives, lighting conditions, focal length, noise, etc. Each set also provides the homograph matrices for all combinations of image pairs. The homograph matrices are used to find ground truth when analyzing feature matching. We consider HPatch dataset [18] to be extremely comprehensive and fit for a network to learn local features from. Sizewise, it contains 116 sets of images, which is over 10 times larger than other datasets. It is also composed of images from very diverse categories including portraits, landscapes, indoors, planar objects (graffiti), cluttered space, etc.

4.2. Implementation Details

The network is trained with stochastic gradient descent optimizer [20]. The momentum is set to 0.9, and weight decay is set to 10^{-4}. The initial learning rate starts at 10^{-2}, and is divided by 10 every time the performance plateaus for 10 epochs. We stop training when learning rate is below 10^{-7}.

To further enrich the training data and approximate the real scenarios, when sampling the second patch P_2 with an angle θ_{gt}, we add a random translation, making P_2 centering at $I(x + \Delta_x, y + \Delta_y)$. The reasoning behind is that, even for the two corresponding feature points, they usually are not centered at exactly same location. We constrain the range of this random translation to be $\|(\Delta_x, \Delta_y)\|_2 \leq r/4$, where r is the radius of feature point.

4.3. Experiment Setup

4.3.1. Orientation Estimation

This experiment is very similar to training and the goal is to evaluate the accuracy of orientation prediction. For each test image, we detect top 1000 features with Edge Foci detector. We then sample two patches P_1, P_2 using same procedure described in Sec. 3.2. We report the average error between $\theta_2 - \theta_1$ and θ_{gt} as performance metric in Table 1. We examine four orientation estimators in this experiment: the handcrafted estimators Edge Foci (EF) [2], SURF [4]; and the learning based method from Yi et al. [13], and ours. Note that in order to be fair to SURF [4] and Edge Foci [2], which are sensitive to keypoints locations. we didn't include random translation when sampling the second patch during testing.

We divide the range of random angle θ_{gt} into four partitions and see whether each method will degrade when θ_{gt} increases. Overall, the two learning based methods perform much better than handcrafted ones. Our method achieves best accuracy on all tested datasets. We also observe that the two learning based methods performed well as θ_{gt} increases.

4.3.2. Feature Matching

We replace the orientation estimator in traditional feature matching evaluation pipeline with ours and further demonstrate the power of the proposed network in this section. For

Viewpoints [13]	EF + VGG	EF + Daisy	Yi + VGG	Yi + Daisy	Ours + VGG	Ours + Daisy
Chatnoir	0.51	0.50	0.61	0.59	**0.63**	**0.61**
Duckhunt	0.38	0.20	0.48	0.33	0.45	0.31
Mario	0.24	0.15	0.31	0.20	**0.34**	**0.22**
Outside	0.39	0.32	0.49	0.41	**0.51**	0.41
Posters	0.57	0.55	0.65	0.59	0.65	0.57
Edge Foci [2]	EF + VGG	EF + Daisy	Yi + VGG	Yi + Daisy	Ours + VGG	Ours + Daisy
Notredame	0.21	0.17	0.32	0.27	**0.34**	**0.28**
PaintedLadies	0.03	0.02	0.05	0.05	0.04	0.03
Rushmore	0.09	0.06	0.12	0.07	**0.15**	**0.10**
Yosemite	0.04	0.03	0.06	0.04	**0.07**	**0.05**
Obama	0.12	0.12	0.19	0.18	**0.24**	**0.22**
Webcam [19]	EF + VGG	EF + Daisy	Yi + VGG	Yi + Daisy	Ours + VGG	Ours + Daisy
Chamonix	0.36	0.30	0.50	0.41	0.49	0.40
Courbevoie	0.30	0.26	0.41	0.37	**0.45**	**0.39**
Frankfurt	0.37	0.31	0.50	0.43	**0.51**	0.43
Mexico	0.31	0.21	0.39	0.31	**0.42**	**0.32**
Panorama	0.24	0.24	0.26	0.25	0.26	0.25
StLouis	0.19	0.13	0.29	0.21	0.27	0.21

Table 2. mAP for all 6 orientation estimator and feature descriptor combinations. Using learning based estimators receive constant gains the mAP, which perfectly aligns with orientation estimation results. Our method provides the most performance gains in the majority of image sets.

Viewpoints [13]	EF	SURF	Yi	Ours
$\lvert\theta_{\text{gt}}\rvert \leq 15^\circ$	5.82°	7.33°	3.90°	$\mathbf{2.48^\circ}$
$15^\circ < \lvert\theta_{\text{gt}}\rvert \leq 45^\circ$	11.99°	15.86°	5.47°	$\mathbf{3.94^\circ}$
$45^\circ < \lvert\theta_{\text{gt}}\rvert \leq 90^\circ$	12.27°	16.42°	5.49°	$\mathbf{3.98^\circ}$
$90^\circ < \lvert\theta_{\text{gt}}\rvert \leq 180^\circ$	14.31°	20.72°	5.38°	$\mathbf{4.11^\circ}$
Edge Foci [2]	EF	SURF	Yi	Ours
$\lvert\theta_{\text{gt}}\rvert \leq 15^\circ$	6.15°	7.72°	4.36°	$\mathbf{2.77^\circ}$
$15^\circ < \lvert\theta_{\text{gt}}\rvert \leq 45^\circ$	13.13°	16.83°	6.11°	$\mathbf{4.19^\circ}$
$45^\circ < \lvert\theta_{\text{gt}}\rvert \leq 90^\circ$	13.23°	17.99°	5.99°	$\mathbf{4.27^\circ}$
$90^\circ < \lvert\theta_{\text{gt}}\rvert \leq 180^\circ$	18.44°	22.84°	6.10°	$\mathbf{4.48^\circ}$
Webcam [19]	EF	SURF	Yi	Ours
$\lvert\theta_{\text{gt}}\rvert \leq 15^\circ$	5.99°	7.60°	4.45°	$\mathbf{2.86^\circ}$
$15^\circ < \lvert\theta_{\text{gt}}\rvert \leq 45^\circ$	12.60°	16.30°	6.04°	$\mathbf{4.15^\circ}$
$45^\circ < \lvert\theta_{\text{gt}}\rvert \leq 90^\circ$	12.77°	17.21°	5.98°	$\mathbf{4.24^\circ}$
$90^\circ < \lvert\theta_{\text{gt}}\rvert \leq 180^\circ$	18.40°	22.17°	6.01°	$\mathbf{4.49^\circ}$

Table 1. Average error for orientation estimation. As shown in the table, two learning based methods outperform handcrafted ones by large margins. They also have relatively consistent performances for both large and small θ_{gt}. Our method has the lowest error in all benchmarks.

baseline, we use Edge Foci detector + Daisy [5] descriptor and Edge Foci detector + VGG [8] descriptor. These two descriptors each represents handcrafted and learning based methods. We substitute the orientation estimator in Edge Foci detector with Yi's method [13] and ours, yielding a total of 6 combinations. In detail, given a pair of images in the test dataset, we first detect top 1000 feature points with Edge Foci [2] detector. Ground truth matching pairs between the 2000 feature points in two images are found using the provided homograph matrix. Each feature point will then receive a description based on the orientation estimator and descriptor in the combination. We match these features by descriptions and check with ground truth matching. We report the mean Average Precision (mAP) for each combination, which is measured as the area under the precision-recall curve [14].

As illustrated in the Table 2, feature descriptors benefit a lot from both learning based estimators, which aligns with the results in Sec. 4.3.1. Surprisingly, our unsupervised method outperforms Yi's method in multiple image sets, considering Yi's method is trained under very similar scheme as this evaluation pipeline.

5. CONCLUSION

In this paper, we present an *unsupervised* scheme to *explicitly* train a network to estimate feature orientation. Both experiments in orientation estimation and feature matching show the proposed algorithm can provide accurate orientation estimation.

6. REFERENCES

[1] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison, "Kaze features," in *European Conference on Computer Vision*. Springer, 2012, pp. 214–227. 1

[2] C Lawrence Zitnick and Krishnan Ramnath, "Edge foci interest points," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 359–366. 1, 2, 3, 4

[3] David G Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Ieee, 1999, vol. 2, pp. 1150–1157. 1, 2

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417. 1, 2, 3

[5] Engin Tola, Vincent Lepetit, and Pascal Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 5, pp. 815–830, 2010. 1, 4

[6] C Lawrence Zitnick, "Binary coherent edge descriptors," in *European Conference on Computer Vision*. Springer, 2010, pp. 170–182. 1

[7] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443. 1

[8] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Learning local feature descriptors using convex optimisation.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1573–1585, 2014. 1, 4

[9] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3279–3286. 1

[10] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua, "Lift: Learned invariant feature transform," in *European Conference on Computer Vision*. Springer, 2016, pp. 467–483. 1, 2

[11] Hani Altwaijry, Andreas Veit, Serge J Belongie, and Cornell Tech, "Learning to detect and match keypoints with deep architectures.," in *BMVC*, 2016. 1

[12] Yurun Tian, Bin Fan, and Fuchao Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 661–669. 1

[13] Kwang Moo Yi, Yannick Verdie, Pascal Fua, and Vincent Lepetit, "Learning to assign orientations to feature points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 107–116. 1, 2, 3, 4

[14] Krystian Mikolajczyk and Cordelia Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005. 1, 4

[15] Stéphane Allaire, John J Kim, Stephen L Breen, David A Jaffray, and Vladimir Pekar, "Full orientation invariance and improved feature selectivity of 3d sift with application to medical image analysis," 2008. 2

[16] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, 2011, pp. 2564–2571. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034. 2

[18] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3852–3861. 3

[19] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit, "Tilde: A temporally invariant learned detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5279–5288. 3, 4

[20] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147. 3