Phase-Shifting Coder: Predicting Accurate Orientation in Oriented Object Detection

Yi Yu[1,2], Feipeng Da[1,2,*]
[1]School of Automation, Southeast University, Nanjing, China
[2]Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing, China
{yuyi, dafp}@seu.edu.cn

Abstract

With the vigorous development of computer vision, oriented object detection has gradually been featured. In this paper, a novel differentiable angle coder named phase-shifting coder (PSC) is proposed to accurately predict the orientation of objects, along with a dual-frequency version (PSCD). By mapping the rotational periodicity of different cycles into the phase of different frequencies, we provide a unified framework for various periodic fuzzy problems caused by rotational symmetry in oriented object detection. Upon such a framework, common problems in oriented object detection such as boundary discontinuity and square-like problems are elegantly solved in a unified form. Visual analysis and experiments on three datasets prove the effectiveness and the potentiality of our approach. When facing scenarios requiring high-quality bounding boxes, the proposed methods are expected to give a competitive performance. The codes are publicly available at https://github.com/open-mmlab/mmrotate.

1. Introduction

As a fundamental task in computer vision, object detection has been extensively studied. Early researches are mainly focused on horizontal object detection [33], on the ground that objects in natural scenes are usually oriented upward due to gravity. However, in other domains such as aerial images [2, 12, 20, 23, 25], scene text [7, 10, 13, 14, 34], and industrial inspection [11, 19], oriented bounding boxes are considered more preferable. Upon requirements in these scenarios, oriented object detection has gradually been featured.

At present, several solutions around oriented object detection have been developed, among which the most intuitive way is to modify horizontal object detectors by adding an output channel to predict the orientation angle. Such a solution faces two problems:

1) Boundary problem [24]: Boundary discontinuity problem is often caused by angular periodicity. Assuming orientation $-\pi/2$ is equivalent to $\pi/2$, the network output is sometimes expected to be $-\pi/2$ and sometimes $\pi/2$ when facing the same input. Such a situation makes the network confused about in which way it should perform regression.

2) Square-like problem [27]: Square-like problem usually occurs when a square bounding box cannot be uniquely defined. Specifically, a square box should be equivalent to a 90° rotated one, but the regression loss between them is high due to the inconsistency of angle parameters. Such ambiguity can also seriously confuse the network.

A more comprehensive introduction to these problems can be found in previous researches [25, 27]. Also, several methods have been proposed to address these problems, which will be reviewed in Sec. 2.

Through rethinking the above problems, we find that they can inherently be unified as rotationally symmetric problems (boundary under 180° and square-like under 90° rotation), which is quite similar to the periodic fuzzy problem of the absolute phase acquisition [37] in optical measurement. Inspired by this, we come up with an idea to utilize phase-shifting coding, a technique widely used in optical measurement [36], for angle prediction in oriented object detection. The technique has the potential to solve both boundary discontinuity and square-like problems:

1) Phase-shifting encodes the measured distance (or parallax) into the periodic phase in optical measurement. The orientation angle can also be encoded into the periodic phase, and boundary discontinuity is thus inherently solved.

2) Phase-shifting also has the periodic fuzzy problem, which is similar to the square-like problem, and many solutions exist. For example, the dual-frequency phase-shifting

*Corresponding author is Feipeng Da. This work is supported by Special Project on Basic Research of Frontier Leading Technology of Jiangsu Province of China (BK20192004C).

technique solves the periodic fuzzy problem by mixing phases of different frequencies (also known as phase unwrapping [37]).

Motivations of this paper:

Based on the above analysis, we believe that the phase-shifting technique can be modified and adapted to oriented object detection. What is the principle of the phase-shifting angle coder? How to integrate this module into a deep neural network? Will this technique result in better performance? These questions are what this paper is for.

Contributions of this paper:

1) We are the first to utilize the phase-shifting coder to cope with the angle regression problem in the deep learning area. An integral and stable solution is elaborated on in this paper. Most importantly, the codes are well-written, publicly available, and with reproducible results.

2) The performance of the proposed methods is evaluated through extensive experiments. The experimental results are of high quality—All the listed results are retested on identical environments to ensure fair comparisons (instead of copied from other papers).

The rest of this paper is organized as follows:

Section 2 reviews the related methods around oriented object detection. Section 3 describes the principles of the phase-shifting coder in detail. Section 4 conducts experiments on several datasets to evaluate the performance of the proposed methods. Section 5 concludes the paper.

2. Related work

With datasets such as DOTA [20], HRSC [12], and ICDAR [14], extensive studies around oriented object detection have been carried out, and some representative ones are summarized in this section.

2.1. From horizontal to oriented

Many oriented object detection methods are based on horizontal object detection, which has been reviewed in the literature [8, 17, 33]. Hence, we only briefly introduce four representative frameworks:

1) ***Anchor-based.*** RetinaNet (2017) [9]: As a one-stage detector, RetinaNet uses Feature Pyramid Network (FPN) as the backbone, to which two subnetworks are attached, one for classification and the other for regression.

2) ***Anchor-free.*** FCOS (2019) [18]: By adding a centerness regression, this work presents an anchor-free one-stage detector, which avoids the complicated computation and hyper-parameters related to anchor boxes.

3) ***Point-based.*** RepPoint (2019) [31]: RepPoint is a representation of objects as a set of sample points, which learn to arrange themselves in a manner that bounds the spatial extent of an object and indicates significant local areas.

4) ***High-efficiency.*** YOLO (v5, 2021) [6]: Famous for its high speed and accuracy, YOLO divides images into a grid system. Each cell in the grid is responsible for detecting objects within itself.

By adding an output channel to predict the orientation of each object, these horizontal detectors can comfortably be applied to oriented object detection, usually termed as *Rotated RetinaNet*, *Rotated FCOS*, *Rotated RepPoint*, and *Rotated YOLO* [35].

2.2. Rotation-invariant detectors

Based on the above frameworks, some studies use additional modules to cope with rotation and improve performance. Some representative ones are as follows:

1) RoI Transformer (2019) [1]: The core idea is to apply spatial transformations on Regions of Interest (RoIs) and learn the transformation parameters under the supervision of oriented bounding box (OBB) annotations.

2) ReDet (2021) [4]: Rotation-equivariant Detector (ReDet) is proposed in this work to explicitly encode rotation equivariance and rotation invariance. A rotation-equivariant network is used to accurately predict the orientation, and upon that, rotation-invariant features can be extracted through rotation-invariant RoI align.

3) S2ANet (2021) [3]: Single-shot alignment network (S2ANet) consists of two modules: a feature alignment module (FAM) and an oriented detection module (ODM). The FAM can generate high-quality anchors, and the ODM adopts active rotating filters to encode the orientation information and produce orientation-invariant features.

4) R3Det (2021) [26]: An end-to-end refined one-stage rotation detector is proposed for fast and accurate object detection by using a progressive regression approach from coarse to fine granularity, followed by a feature refinement module to further improve detection performance.

5) Oriented R-CNN (2021) [21]: This work proposes a general two-stage oriented detector with promising accuracy and efficiency. In the first stage, an oriented Region Proposal Network (oriented RPN) directly generates high-quality oriented proposals in a nearly cost-free manner. The second stage is oriented R-CNN head for refining oriented Regions of Interest (oriented RoIs) and recognizing them.

Most of the above methods enhance the network architecture and improve the rotation invariance of the output features. With the progress in related fields, some researchers found that to further improve the performance, two other problems need to be concerned: *boundary discontinuity* and *square-like* [24, 27].

2.3. Boundary and square-like problems

Since these two problems were pinpointed, many techniques have been proposed to solve part of them, which can be divided into three categories:

1) ***Smooth loss function.*** SCRDet (2019) [28]: IoU-smooth L1 loss is proposed to smooth the boundary loss

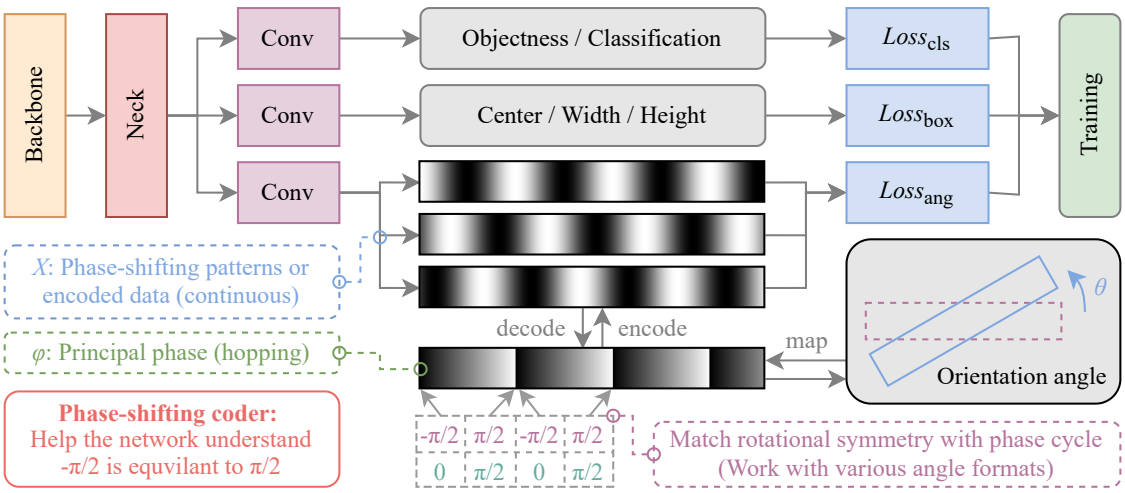


Figure 1. Overall flowchart of the phase-shifting coder integrated into a deep neural network.

jump; RSDet (2021) [16]: Similarly, a modified version using modulated loss is proposed in this work.

2) ***Angle coder.*** CSL (2020) [24]: Circular smooth label (CSL) technique converts angle regression to classification to handle the periodicity of the orientation angle and increase the error tolerance to adjacent angles; Upon CSL, DCL [22] further solves the square-like problem.

3) ***Gaussian distribution.*** GWD (2021) [27]: Regression loss based on Gaussian Wasserstein distance (GWD) is proposed, where the rotated bounding box is converted to a 2D Gaussian distribution to calculate the regression loss; KLD (2021) [29]: The Kullback-Leibler Divergence (KLD) between the Gaussian distributions is calculated as the regression loss; KFIoU (2022) [30]: Kalman filter is adopted to mimic the mechanism of Skew Intersection over Union (SkewIoU) by its definition, which requires less hyper-parameter tuning than GWD and KLD.

These methods consider the problems from different perspectives and each has its pros and cons. For example, SCRDet and RSDet are designed to alleviate the impact of the problems, instead of theoretically solving them; CSL is simple and stable, but not able to solve the square-like problem, and its performance could be greatly affected by hyper-parameters; GWD and KLD solve both problems elegantly, but their prediction is relatively inaccurate, resulting in high mAP_{50} but low mAP_{75} performance.

Based on the above analysis, much progress has been made in the field of oriented object detection, but related problems have not yet been completely solved.

3. Method

In this section, we will first introduce the encoding and decoding procedure of the phase-shifting coder. Afterward, an enhanced version dual-frequency phase-shifting coder will be introduced. Readers are referred to literature [36] for the basic principles of the phase-shifting technique.

3.1. Phase-shifting coder (PSC)

The overall flowchart of the phase-shifting coder integrated into a deep neural network is illustrated in Fig. 1. Taking the "long edge 90" angle definition as an example for illustration, symbols can be defined as follows:

- θ: Orientation angle, in range $[-\pi/2, \pi/2)$
- φ: Principal phase, in range $[-\pi, \pi)$
- N_{step}: The number of phase-shifting steps
- X: Encoded data, $X = \{x_n \mid n = 1, 2, \cdots, N_{\text{step}}\}$

Mapping: The cycle of sin or cos is 2π, whereas a rectangle box is identical to itself when rotated by π, thus a mapping is required to match them, as follows:

$$\varphi = 2\theta \tag{1}$$

Encoding: The formula of encoding φ into X can be described as:

$$x_n = \cos\left(\varphi + \frac{2n\pi}{N_{\text{step}}}\right) \tag{2}$$

where $n = 1, 2, \cdots, N_{\text{step}}$.

To simplify the subsequent description, Eq. (2) is also denoted as $X = f_{\text{enc}}(\varphi)$.

Decoding: The formula of decoding φ from X can be described as:

$$\varphi = -\arctan \frac{\sum_{n=1}^{N_{\text{step}}} x_n \sin\left(\frac{2n\pi}{N_{\text{step}}}\right)}{\sum_{n=1}^{N_{\text{step}}} x_n \cos\left(\frac{2n\pi}{N_{\text{step}}}\right)} \tag{3}$$

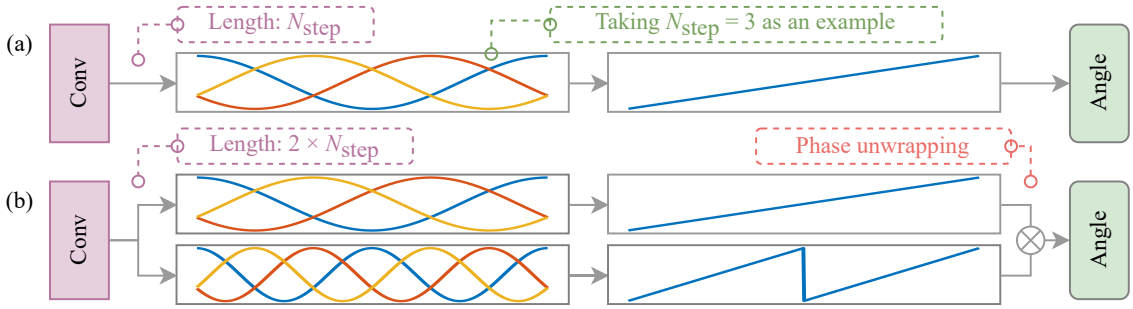


Figure 2. Inference flow of (a) single-frequency PSC and (b) dual-frequency PSC.

The arctan in the formula should be implemented by the arctan2 function so that its output is in the range $(-\pi, \pi]$. Equation (3) is also denoted as $\varphi = f_{\text{dec}}(X)$.

3.2. Dual-frequency phase-shifting coder (PSCD)

Through rethinking the boundary problem and the square-like problem, we believe these two problems can be inherently unified. If a bounding box is equivalent to itself under 180° rotation, the boundary problem occurs, but if they are equivalent under 90° rotation, the square-like problem occurs. Both cases are periodic fuzzy problems but of different cycles.

Therefore, to solve both boundary discontinuity and square-like problems, an additional phase is required to establish the dual-frequency phase-shifting coder. The difference between the basic phase-shifting coder and the dual-frequency one is illustrated in Fig. 2.

Additional symbols used in dual-frequency phase-shifting coder can be defined as follows:

- φ_1: Phase of the first frequency, in range $[-\pi, \pi)$
- φ_2: Phase of the second frequency, in range $[-2\pi, 2\pi)$
- φ: Final principal phase, in range $[-\pi, \pi)$
- X_1: Data encoded from the phase of the first frequency, $X_1 = \{x_n \mid n = 1, 2, \cdots, N_{\text{step}}\}$
- X_2: Data encoded from the phase of the second frequency, $X_2 = \{x_n \mid n = 1, 2, \cdots, N_{\text{step}}\}$
- X: Final encoded data, with coding length $2 \times N_{\text{step}}$, $X = \{X_1, X_2\}$

Mapping: In dual-frequency PSC, two principal phases are mapped from angle θ during the training process:

$$\begin{cases} \varphi_1 = 2\theta \\ \varphi_2 = 4\theta \end{cases} \tag{4}$$

The output orientation angle is mapped from the final principal phase during the inference process:

$$\theta = \frac{1}{2}\varphi \tag{5}$$

Encoding: Similar to Eq. (2), the formula of encoding φ_1 and φ_2 into X_1 and X_2 can be described as:

$$\begin{cases} X_1 = f_{\text{enc}}(\varphi_1) \\ X_2 = f_{\text{enc}}(\varphi_2) \end{cases} \tag{6}$$

Decoding: Similar to Eq. (3), the formula of decoding φ_1 and φ_2 from X_1 and X_2 can be described as:

$$\begin{cases} \varphi_1 = f_{\text{dec}}(X_1) \\ \varphi_2 = f_{\text{dec}}(X_2) \end{cases} \tag{7}$$

Unwrapping: The network outputs two principal phases during inference: φ_1 as the absolute phase and φ_2 as the wrapped phase. We need to mix them to obtain the final phase (also known as phase unwrapping). To this end, we first calculate the inner product between φ_1 and φ_2 by:

$$\delta = \cos\varphi_1 \cos\frac{\varphi_2}{2} + \sin\varphi_1 \sin\frac{\varphi_2}{2} \tag{8}$$

Afterward, φ_2 is unwrapped according to δ, so that the two phases can be automatically mixed to obtain the final phase φ, as follows:

$$\varphi = \begin{cases} \pi + \frac{\varphi_2}{2}, & \text{if } \delta < 0 \\ \frac{\varphi_2}{2}, & \text{else} \end{cases} \tag{9}$$

It should be noted that the above formula is a simplified version for higher clarity. In fact, after added by π, φ could be out of the range $[-\pi, \pi)$. In such case, φ needs to be subtracted by 2π, otherwise the output angle could be outside the definition range.

3.3. General form of mapping

Similarly, the ambiguity caused by "triangle-like" or "pentagon-like" objects can also be solved, as well as the 360° orientation problem (for example, predicting the heading direction of the airplane), by extending the mapping between phase and angle to a more general form:

$$\varphi = k\theta \tag{10}$$

where $k = 2\pi/s$, assuming objects to be detected are symmetric under the rotation of s radians, as shown in Fig. 3.

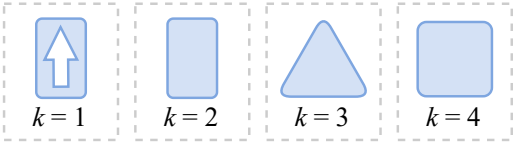


Figure 3. Objects with different rotational periodicity.

PSC is applicable in all of these cases, while the multi-frequency strategy can be used when different types exist simultaneously. In other words, by mapping the rotational periodicity of different cycles into the phase of different frequencies, we provide a unified framework for various rotational symmetry problems in oriented object detection.

3.4. Loss function

The oriented bounding box is represented by five parameters (x, y, w, h, θ), denoting the box's center coordinates, width, height, and angle, respectively. As an angle-coder-based method, PSC only involves the regression of θ.

According to Eq. (2), the encoded data (phase-shifting patterns) are in the range $[-1, 1]$. To make the training more stable, we also transform the output features with:

$$X_{\text{Pred}} = 2 \times \text{sigmoid}(X_{\text{Feat}}) - 1 \tag{11}$$

where X_{Feat} is the output features of the convolution layer, and X_{Pred} is the predicted encoded data in range $[-1, 1]$.

Afterward, the loss of the angle branch can be calculated with L1 loss:

$$L_{\text{ang}} = |X_{\text{GT}} - X_{\text{Pred}}| \tag{12}$$

where X_{GT} is the ground truth phase-shifting patterns encoded from the orientation angle of annotated boxes.

Finally, the overall loss can be expressed as:

$$L = w_1 L_{\text{cls}} + w_2 L_{\text{box}} + w_3 L_{\text{ang}} \tag{13}$$

where $w_1 L_{\text{cls}}$ and $w_2 L_{\text{box}}$ are the weighted loss of classification and box regression branches defined by the backbone detector, and w_3 is set to $0.2 w_1$ by default.

It should be noted that Eq. (13) describes only the general situation, and there are also some special cases, such as FCOS with center-ness loss.

4. Experiments

With the help of PyTorch [15], ultralytics/yolov5 [6], and MMRotate [35] tool kits, experiments are carried out to evaluate the performance of the proposed methods. To compare with existing literature, we choose mean average precision (mAP) as the major metric. The computing infrastructure is as follows: CPU: Intel i9-12900K, GPU: Nvidia RTX3080, OS: Windows 10, PyTorch: 1.10.1, ultralytics/yolov5: 6.0, MMRotate: 0.3.2.

4.1. Datasets and benchmarks

DOTA [20]: DOTA is comprised of 2,806 large aerial images—1,411 for training, 937 for validation, and 458 for testing. The dataset is annotated using 15 categories with 188,282 instances in total. The categories are defined as: Plane (PL), Baseball Diamond (BD), Bridge (BR), Ground Field Track (GTF), Small Vehicle (SV), Large Vehicle (LV), Ship (SH), Tennis Court (TC), Basketball Court (BC), Storage Tank (ST), Soccer-Ball Field (SBF), Roundabout (RA), Harbor (HA), Swimming Pool (SP), and Helicopter (HC). We follow the standard preprocessing procedure in MMRotate—The high-resolution images are split into 1024×1024 patches with an overlap of 200 pixels for training, and during inference, the detection results of all patches are merged to evaluate the performance.

HRSC [12]: As a ship detection dataset, HRSC contains ship instances both on the sea and inshore, with arbitrary orientation. The training, validation, and testing set include 436, 181, and 444 images respectively. We use the preprocessing provided by MMRotate, where the images are scaled to 800×800 for training and testing.

OCDPCB [1]**:** OCDPCB is a dataset for oriented component detection in printed circuit boards aimed at automated optical inspection. The dataset consists of 636 images, of which 445 images are used for training and 191 for testing. The resolution of the images is 1280×1280.

4.2. Ablation study

Ablation study of hyper-parameters:

In most existing methods such as CSL, GWD, and KLD, hyper-parameters could highly affect the performance. Worse still, the best parameters vary in different scenarios and datasets, which require laborious tuning.

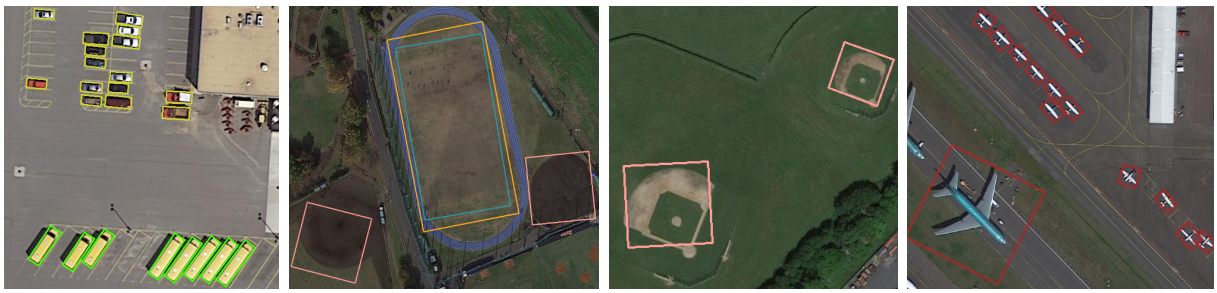
PSC is quite different—The only hyper-parameter N_{step} is an integer greater than or equal to 3, so we evaluate several N_{step} values most commonly used in the phase-shifting technique, and the results, which are obtained on RetinaNet with dual-frequency PSC, are shown in Tab. 1.

Metrics	$N_{\text{step}} = 3$	$N_{\text{step}} = 4$	$N_{\text{step}} = 5$
DOTA (mAP_{50})	**71.09**	70.96	71.01
DOTA (mAP_{75})	41.17	**41.82**	41.53
DOTA ($\text{mAP}_{50:95}$)	41.25	**41.51**	41.13
HRSC (mAP_{50})	**85.53**	85.49	85.46
HRSC (mAP_{75})	59.57	**59.64**	59.54
HRSC ($\text{mAP}_{50:95}$)	53.20	**53.22**	53.12

Table 1. Performance under different N_{step} value.

According to the experimental results, this parameter has quite a limited impact on the results. Although $N_{\text{step}} = 4$

[1]The dataset is available at https://yuyi1005.github.io.



(a) Detection results of PSC: Boundary discontinuity free, accurate, and unquantized

(b) Detection results of PSCD: Square-like problem free and more accurate

Figure 4. Visual comparison between single-frequency PSC and dual-frequency PSC.

shows better mAP_{75} performance, it also impairs mAP_{50} performance. On the whole, larger N_{step} will not bring significant benefits. Taking the computational complexity into consideration, we would recommend $N_{step} = 3$.

Once $N_{step} = 3$ is determined, PSC has no adjustable hyper-parameters.

Ablation study of dual-frequency:

We provide an intuitive comparison between single-frequency PSC and dual-frequency PSC to verify the effectiveness of the dual-frequency module and help researchers decide which one to choose.

The visual comparison displayed in Fig. 4 quite conforms with our theory, proving that the dual-frequency strategy can work as expected and solve both boundary discontinuity and square-like problems in a unified way. And thus, the dual-frequency strategy is highly recommended in scenarios containing square-like objects.

4.3. Experimental settings

Group settings:

PSC can work with various detectors and backbones. Thus, we select three state-of-the-art backbones for experiments: FCOS (anchor-free), RetinaNet (anchor-based), and YOLO (high-efficiency). In the experiments, FCOS and RetinaNet are set to be based on ResNet-50 [5] (denoted as R-50), and YOLO includes two configurations: YOLOv5s and YOLOv5m. The parameter volume of the four models is about FCOS (R-50): 32M, RetinaNet (R-50): 36M, YOLOv5s: 7M, and YOLOv5m: 21M.

We set up five experimental groups for DOTA dataset, and two for HRSC or OCDPCB datasets (as shown in Tabs. 2 to 4). Both the backbone and the data augmentation are identical within each group to make the comparisons fair. Also, SWA [32] and multi-scale testing are not adopted. Networks based on FCOS and RetinaNet are trained by 12 epochs on DOTA and 72 epochs on HRSC and OCDPCB, whereas the YOLO-based groups are trained by 120 epochs. The learning rate is initially set to 1e-3 and finally reduced to 1e-5. Readers are referred to the configuration files in our codes for all the particulars.

Baseline methods:

As an angle-coder-based method for boundary discontinuity and square-like problems, we take the following two most relevant methods as the baseline for comparison.

(1) CSL [24]: The most widely used angle coder converting the angle regression to classification to solve boundary discontinuity problem.

(2) KLD [29]: The state-of-the-art method solving both boundary discontinuity and square-like problems based on Gaussian distribution.

4.4. Results and analysis

The quantitative results on DOTA, HRSC, and OCDPCB datasets are demonstrated in Tabs. 2 to 4. The top two results within each group are labeled in bold red and blue.

▼ Comparisons on DOTA dataset:

1) ***With KLD.*** Both PSC and PSCD show advantages over existing methods in many groups of experiments. Specifically, PSC is on a par with KLD in mAP_{50}, but significantly better than KLD in mAP_{75}. Such improvement

Method		R	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP_{50}	mAP_{75}	$mAP_{50:95}$
FCOS (R-50)	Rotated		**89.18**	71.99	47.97	61.61	79.30	73.52	85.78	**90.90**	81.09	84.30	**59.57**	**62.69**	62.08	69.94	49.31	71.28	37.08	39.42
	+KLD		89.17	**75.42**	**49.41**	56.48	79.66	76.78	86.91	90.89	**83.52**	84.41	58.76	62.21	63.44	67.90	50.07	71.67	37.53	39.67
	+CSL		88.24	74.87	41.27	61.03	79.52	78.35	87.19	90.88	81.50	**84.53**	54.70	62.65	62.84	68.45	46.50	70.83	38.71	39.75
	+PSC		88.24	74.42	48.63	63.44	79.98	**80.76**	**87.59**	90.88	82.02	71.58	59.12	60.78	**65.78**	**71.21**	53.06	**71.83**	39.21	**40.42**
	+PSCD		88.04	73.95	48.83	**63.44**	**80.01**	80.75	87.58	90.88	81.69	67.23	58.70	60.26	65.67	71.11	**53.06**	71.41	**39.35**	40.36
FCOS (R-50)	+KLD	✓	89.05	**73.73**	**49.17**	57.86	**79.54**	**77.91**	87.41	90.87	**83.23**	82.42	58.97	61.83	62.91	72.69	59.44	72.47	37.84	39.95
	+CSL	✓	87.84	68.88	42.15	57.41	77.24	74.78	87.66	**90.89**	79.43	**84.34**	54.35	63.46	61.39	69.62	61.16	70.71	37.13	38.47
	+PSC	✓	**89.07**	73.60	48.91	**62.63**	75.24	77.71	**88.02**	90.85	82.86	69.85	**61.48**	**65.25**	65.68	**73.03**	**67.88**	72.80	38.83	41.10
	+PSCD	✓	89.06	73.61	49.03	62.34	75.18	77.69	88.00	90.85	82.63	72.97	**61.48**	64.20	**65.77**	72.88	**67.88**	**72.90**	**39.80**	**41.51**
RetinaNet (R-50)	Rotated		89.41	76.81	**40.88**	67.54	77.51	62.63	77.55	90.89	82.31	81.98	58.16	61.56	56.46	63.71	38.96	68.42	**42.03**	40.13
	+KLD		**89.50**	79.91	39.92	70.40	78.04	64.24	**82.79**	**90.90**	81.80	**83.02**	57.63	63.52	56.63	65.13	50.04	70.23	37.88	39.31
	+CSL		89.33	79.67	40.83	69.95	77.71	62.08	77.46	90.87	82.87	82.03	60.07	65.27	53.58	64.03	46.62	69.49	40.42	39.69
	+PSC		89.41	80.66	39.06	69.08	77.61	61.63	77.21	90.86	82.52	81.76	**60.98**	**66.20**	57.51	64.75	48.28	69.83	40.37	40.03
	+PSCD		89.32	**82.29**	37.92	**71.52**	**78.40**	**66.33**	78.01	90.89	**84.21**	80.63	60.22	64.73	**59.69**	**68.37**	**53.85**	**71.09**	41.17	**41.25**
YOLOv5s	+CSL	✓	89.26	84.53	51.36	60.69	80.70	84.74	88.41	90.68	**85.93**	87.59	59.62	65.18	74.53	81.71	**66.91**	76.79	46.85	45.63
	+PSC	✓	89.65	86.37	51.76	**63.42**	81.21	84.63	88.29	90.80	85.39	87.93	**61.00**	66.41	**75.01**	81.77	66.20	77.32	47.56	46.48
	+PSCD	✓	**89.70**	**86.68**	**52.47**	59.74	**81.57**	**85.14**	**88.44**	**90.83**	83.99	**88.40**	60.43	**69.04**	74.89	**83.30**	66.27	**77.40**	**51.50**	**48.27**
YOLOv5m	+CSL	✓	89.60	**86.39**	54.62	**62.04**	80.36	85.20	**88.40**	**90.80**	80.54	88.47	**64.16**	61.21	76.70	**83.13**	**67.91**	77.30	49.88	47.97
	+PSC	✓	89.85	85.93	**54.94**	61.56	81.89	85.47	88.37	90.73	**86.90**	88.79	63.90	68.92	76.82	82.83	63.25	78.01	50.50	48.60
	+PSCD	✓	**89.86**	86.02	54.94	62.02	**81.90**	**85.48**	88.39	90.73	86.90	**88.82**	63.94	**69.19**	**76.84**	82.75	63.24	**78.07**	**54.10**	**50.35**

Table 2. AP_{50} of each category and mAP on DOTA. Column "R" means using random rotation and random resize as augmentation.

in mAP_{75} is even more significant for PSCD, which outperforms KLD by 2.36 pp on average. Based on the above analysis, it can be concluded that PSC and PSCD have a similar recall rate to KLD, but the bounding boxes predicted by PSC and PSCD show higher Intersection over Union (IoU) and better quality.

2) ***With CSL.*** In comparisons with CSL, both PSC and PSCD are superior in most metrics. On average, PSCD outperforms CSL by 1.15 pp in mAP_{50}, and 2.59 pp in mAP_{75}. It can be seen from the results that the gap between PSC and CSL is even more significant when data augmentation is used. We believe this phenomenon can be theoretically explained—By observing the training log, we find that the angle loss of PSC is much lower than that of CSL. With lower angle loss, PSC leaves more margin for the network to fit those augmented data, allowing the network to pay more attention to the classification and the bounding box branches, and finally resulting in higher performance. Furthermore, PSC is fully differentiable with unquantized outputs, which can be useful in detecting tiny deviations. This feature usually cannot be reflected by the mAP value, but could be important for some applications.

3) ***Between PSC and PSCD.*** The mAP_{75} of PSCD is considerably higher than that of PSC on DOTA dataset, with the improvement reaching 3.94 pp on YOLOv5s and 3.60 pp on YOLOv5m. In experiments using FCOS and RetinaNet backbones, PSCD also increases the mAP_{75} by an average of 0.64 pp compared with PSC.

▼ Comparisons on HRSC dataset:

1) ***With CSL and KLD.*** From the experimental results, the mAP_{50} performance of different methods on HRSC dataset do not differ much, among which FCOS+PSC is the best, reaching 90.06 mAP_{50}. Whereas for mAP_{75}, there are noticeable differences among the methods—Although CSL achieves a decent mAP_{50}, its mAP_{75} performance is significantly lower than other methods, indicating that the quality of the bounding boxes detected by CSL is relatively poor, with IoU generally below 75%. When compared with the state-of-the-art method KLD, PSC also shows a significant advantage in mAP_{75}, with an improvement of 1.1 pp and 2.54 pp under FCOS and RetinaNet, respectively.

2) ***Between PSC and PSCD.*** The dual-frequency strategy plays a negative role in some of the experiments. Such results of dual-frequency PSC, a technique mainly aimed at the square-like problem, are quite in line with our expectation—Objects in ship detection datasets are unlikely to be "square-like". A similar phenomenon also occurs in OCDPCB dataset. These results provide guidance for researchers: PSC could be a better alternative than PSCD in scenarios containing rather few square-like objects.

	Method	mAP_{50}	mAP_{75}	$mAP_{50:95}$
FCOS (R-50)	Rotated	89.74	77.00	63.84
	+KLD	89.76	77.46	62.63
	+CSL	89.84	66.47	58.92
	+PSC	**90.06**	78.56	67.57
	+PSCD	89.91	**79.20**	**67.88**
RetinaNet (R-50)	Rotated	83.50	59.60	51.56
	+KLD	**85.85**	58.76	53.40
	+CSL	84.87	38.75	44.17
	+PSC	85.65	**61.30**	**54.14**
	+PSCD	85.53	59.57	53.20

Table 3. Detection accuracy on HRSC dataset.

	Method	mAP_{50}	mAP_{75}	$mAP_{50:95}$
FCOS (R-50)	Rotated	87.88	75.00	64.26
	+KLD	87.72	67.41	59.77
	+CSL	87.23	73.82	63.12
	+PSC	**88.87**	**75.72**	**64.85**
	+PSCD	87.49	75.48	64.18
RetinaNet (R-50)	Rotated	74.68	64.25	55.85
	+KLD	76.30	54.27	49.06
	+CSL	75.38	61.92	53.14
	+PSC	**77.35**	**65.61**	**57.58**
	+PSCD	75.77	64.24	55.70

Table 4. Detection accuracy on OCDPCB dataset.

▼ Comparisons on OCDPCB dataset:

Most oriented object detectors have been well-tuned on DOTA and HRSC datasets, but when adapted to a completely new dataset, they might give a performance far below expectation. Therefore, we choose a less common dataset and train the networks with the same hyper-parameters used in training HRSC.

Surprisingly, without tuning hyper-parameters (such as the smooth radius in CSL) specifically for OCDPCB dataset, CSL and KLD produce rather limited effects, with mAP_{50} slightly higher but mAP_{75} much lower under RetinaNet, and negative results under FCOS.

PSC and PSCD still work well in such circumstances, especially in mAP_{75}, where PSC outperforms KLD by 9.83 pp on average. Based on the above results, it can be concluded that PSC and PSCD can obtain high performance more easily than CSL and KLD when facing a new dataset.

4.5. Inference time

With images of 1024×1024 resolution from DOTA dataset, the average inference time and the model size are evaluated in Tab. 5, which demonstrate that PSC and PSCD are slightly slower than KLD but much faster than CSL.

Method	Time (ms)	Parameters
RetinaNet (R-50)	42.9 (base)	36.71M (base)
+KLD	43.0 **(+0.1)**	36.71M **(+0.00M)**
+CSL ($\omega = 4$)	46.3 **(+3.4)**	37.64M **(+0.93M)**
+CSL ($\omega = 1$)	50.5 **(+7.6)**	40.44M **(+3.73M)**
+PSC	43.1 **(+0.2)**	36.79M **(+0.08M)**
+PSCD	43.4 **(+0.5)**	36.88M **(+0.17M)**

Table 5. Comparisons on inference time and model size.

5. Conclusions

A novel differentiable angle coder for oriented object detection named phase-shifting coder (PSC) is proposed in this paper, which encodes the orientation angle into periodic phases to solve the boundary discontinuity problem. Upon PSC, an enhanced dual-frequency version (PSCD) mapping the rotational periodicity of different cycles into the phase of different frequencies is proposed to elegantly solve both boundary discontinuity and square-like problems.

Afterward, extensive experiments are carried out to evaluate the performance of PSC and PSCD, through which the following conclusions can be drawn:

1) Among angle-coder-based methods, PSC shows significant improvement in both mAP_{50} and mAP_{75} compared with the currently most widely used method CSL. On DOTA dataset, PSCD outperforms CSL by an average of 1.15 pp in mAP_{50} and 2.59 pp in mAP_{75}.

2) Compared with other state-of-the-art approaches, PSC is on a par with Gaussian-distribution-based method KLD in mAP_{50}, but significantly better than KLD in mAP_{75}. In particular, PSCD outperforms KLD by an average of 2.36 pp in mAP_{75} on DOTA dataset. When high-IoU bounding boxes are required, PSC and PSCD are expected to give a competitive performance.

3) The mAP_{75} of PSCD is considerably higher than that of PSC on DOTA dataset, with the improvement reaching 3.94 pp on YOLOv5s and 3.60 pp on YOLOv5m. However, on HRSC and OCDPCB datasets, the dual-frequency strategy plays a negative role, indicating that in scenarios containing rather few square-like objects, PSC could be a better alternative than PSCD.

4) Many existing approaches require different parameters on different datasets. PSC and PSCD, by contrast, do not use dataset-dependent hyper-parameter tuning, so they can be applied to new scenarios more comfortably.

In many scenarios, some objects require 360° oriented detection, while others are 180° or 90° rotationally symmetric. This is an extended form of the square-like problem and a common situation in real applications. In such a case, PSC can still be theoretically applicable, which will be further explored in our future work.

References

[1] Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2849–2858, 2019. 2

[2] Kun Fu, Zhonghan Chang, Yue Zhang, Guangluan Xu, Keshu Zhang, and Xian Sun. Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 161:294–308, 2020. 1

[3] Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022. 2

[4] Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2785–2794, 2021. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[6] Glenn Jocher, Alex Stoken, Ayush Chaurasia, Jirka Borovec, NanoCode012, TaoXie, Yonghye Kwon, Kalen Michael, Liu Changyu, Jiacong Fang, Abhiram V, Laughing, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Jebastin Nadar, imyhxy, Lorenzo Mammana, AlexWang1900, Cristi Fati, Diego Montes, Jan Hajek, Laurentiu Diaconu, Mai Thanh Minh, Marc, albinxavi, fatih, oleg, and wanghaoyang0106. ultralytics/yolov5: v6.0 - yolov5n 'nano' models, roboflow integration, tensorflow export, opencv dnn support, 2021. 2, 5

[7] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-Song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5909–5918, 2018. 1

[8] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 936–944, 2017. 2

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. 2

[10] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5676–5685, 2018. 1

[11] Yuekai Liu, Hongli Gao, Liang Guo, Aoping Qin, Canyu Cai, and Zhichao You. A data-flow oriented deep ensemble learning method for real-time surface defect inspection. *IEEE Transactions on Instrumentation and Measurement*, 69(7):4681–4691, 2020. 1

[12] Zikun Liu, Liu Yuan, Lubin Weng, and Yiping Yang. A high resolution optical satellite image dataset for ship recognition and some new baselines. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, volume 2, pages 324–331, 2017. 1, 2, 5

[13] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018. 1

[14] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, Wafa Khlif, Muhammad Muzzamil Luqman, Jean-Christophe Burie, Cheng-lin Liu, and Jean-Marc Ogier. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification - rrc-mlt. In *2017 14th IAPR International Conference on Document Analysis and Recognition*, volume 01, pages 1454–1459, 2017. 1, 2

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035, 2019. 5

[16] Wen Qian, Xue Yang, Silong Peng, Junchi Yan, and Yue Guo. Learning modulated loss for rotated object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2458–2466, 2021. 3

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 2

[18] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 9626–9635, 2019. 2

[19] Hongjin Wu, Ruoshan Lei, and Yibing Peng. Pcbnet: A lightweight convolutional neural network for defect inspection in surface mount technology. *IEEE Transactions on Instrumentation and Measurement*, 71:1–14, 2022. 1

[20] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018. 1, 2, 5

[21] Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented r-cnn for object detection. In *2021 IEEE/CVF International Conference on Computer Vision*, pages 3520–3529, 2021. 2

[22] Xue Yang, Liping Hou, Yue Zhou, Wentao Wang, and Junchi Yan. Dense label encoding for boundary discontinuity free rotation detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15814–15824, 2021. 3

[23] Xue Yang, Hao Sun, Kun Fu, Jirui Yang, Xian Sun, Menglong Yan, and Zhi Guo. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, 10(1), 2018. 1

[24] Xue Yang and Junchi Yan. Arbitrary-oriented object detection with circular smooth label. In *Computer Vision – ECCV 2020*, pages 677–694, 2020. 1, 2, 3, 6

[25] Xue Yang and Junchi Yan. On the arbitrary-oriented object detection: Classification based approaches revisited. *International Journal of Computer Vision*, 130:1340–1365, 2022. 1

[26] Xue Yang, Junchi Yan, Ziming Feng, and Tao He. R3det: Refined single-stage detector with feature refinement for rotating object. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3163–3171, 2021. 2

[27] Xue Yang, Junchi Yan, Ming Qi, Wentao Wang, Xiaopeng Zhang, and Tian Qi. Rethinking rotated object detection with gaussian wasserstein distance loss. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 11830–11841, 2021. 1, 2, 3

[28] Xue Yang, Jirui Yang, Junchi Yan, Yue Zhang, Tengfei Zhang, Zhi Guo, Xian Sun, and Kun Fu. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 8231–8240, 2019. 2

[29] Xue Yang, Xiaojiang Yang, Jirui Yang, Qi Ming, Wentao Wang, Qi Tian, and Junchi Yan. Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. In *Advances in Neural Information Processing Systems*, volume 34, pages 18381–18394, 2021. 3, 6

[30] Xue Yang, Yue Zhou, Gefan Zhang, Jirui Yang, Wentao Wang, Junchi Yan, Xiaopeng Zhang, and Qi Tian. The kfiou loss for rotated object detection. *arXiv preprint arXiv:2201.12558*, 2022. 3

[31] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 9656–9665, 2019. 2

[32] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sünderhauf. Swa object detection. *arXiv preprint arXiv:2012.12645*, 2020. 6

[33] Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019. 1, 2

[34] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: An efficient and accurate scene text detector. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2017. 1

[35] Yue Zhou, Xue Yang, Gefan Zhang, Jiabao Wang, Yanyi Liu, Liping Hou, Xue Jiang, Xingzhao Liu, Junchi Yan, Chengqi Lyu, Wenwei Zhang, and Kai Chen. Mmrotate: A rotated object detection benchmark using pytorch. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022. 2, 5

[36] Chao Zuo, Shijie Feng, Lei Huang, Tianyang Tao, Wei Yin, and Qian Chen. Phase shifting algorithms for fringe projection profilometry: A review. *Optics and Lasers in Engineering*, 109:23–59, 2018. 1, 3

[37] Chao Zuo, Lei Huang, Minliang Zhang, Qian Chen, and Anand Asundi. Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review. *Optics and Lasers in Engineering*, 85:84–103, 2016. 1, 2