RoPaSS: Robust Watermarking for Partial Screen-Shooting Scenarios

Zehua Ma[1], Han Fang[2*], Xi Yang[3], Kejiang Chen[1*], Weiming Zhang[1, 4]

[1]Anhui Province Key Laboratory of Digital Security, University of Science and Technology of China
[2]National University of Singapore
[3]Jinan University
[4]Institute of Hefei High Dimensional Data Ltd.

Abstract

Screen-shooting robust watermarking is an effective means of preventing screen content leakage from unauthorized camera shooting, as it can trace the leaked source through the watermark extraction thereby providing an effective deterrent. However, current screen-shooting resilient watermarking schemes rely on the image's contours to synchronize and then extract the watermark. While in practical applications, it's common for only a portion of the image to be captured, resulting in a limited performance of the previous watermarking schemes. To address this problem, we propose the RoPaSS: a **ro**bust watermarking scheme for **pa**rtial **s**creen-**s**hooting scenarios, which effectively constructs symmetric characteristics on the embedding watermark to handle the sticky resynchronization issue. Specifically, RoPaSS consists of a watermark encoder, a decoder, and three estimators, which are trained in two stages. In the first training stage, RoPaSS integrates the flipping operation into the watermark encoder and decoder training to increase the redundancy of watermark messages and artificially guide the generation of symmetric watermarks. In the second stage, estimators utilize the watermark symmetry as an additional reference to estimate the restoration parameters to resynchronize the partially captured watermarked image. Experiments have demonstrated the excellent performance of RoPaSS in partial screen-shooting traceability, with extraction accuracy of above 93% in frontal shooting and above 86% in 30° shooting even if only 50% of the image content is captured.

Introduction

In recent years, there has been a rapid growth in screen usage (Min et al. 2021), owing to the advent of the digital information age and the increasing adoption of paperless offices (Briscoe 2022). This has led to a significant increase in the display of important data and sensitive information on a wide range of screens, including personal computing devices, collaborative workspaces, and various digital platforms. Meanwhile, the widespread use of high-resolution cameras in small smart devices, such as mobile phones, has heightened the risk of screen content leakage. A typical threat scenario is that the confidential drawings displayed on a screen are easily captured unauthorized and further leaked.

*Corresponding Authors.

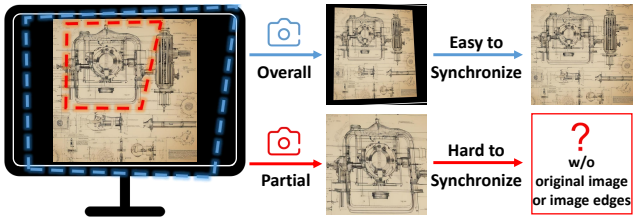

Figure 1: Schematic diagrams of the screen-shooting process (blue) and the partial screen-shooting process (red).

Digital watermarking technology (Ma et al. 2021; Wang et al. 2024; Zhang et al. 2024; Fang et al. 2024) serves as a common solution for tracing the source of leakage. It embeds imperceptible markers, such as identification information, into digital content to authenticate its copyright and trace its source. Robustness is a key property of digital image watermarks, ensuring traceability despite distortions. To ensure robustness against screen-shooting distortion, many researchers have proposed various screen-shooting resilient watermarking schemes (SSR watermark) (Fang et al. 2018, 2022, 2023; Ge et al. 2023). These schemes primarily address scenarios where full watermarked images are captured. But real-life scenarios often capture only partial content, reducing their effectiveness.

The primary reason is that most existing SSR watermark methods necessitate synchronization of the captured image before extraction, where the synchronization reference is typically the image contours or screen edges. However, in the partial screen-shooting scenario, there are not enough references to synchronize the captured image. Such a desynchronization distortion poses challenges in watermark extraction; see the bottom row of Fig. 1.

It should be noted that the existing watermark synchronization methods (Ma et al. 2021; He et al. 2023) cannot adapt well to the screen-shooting scenario since they can only recover the base transform on low-distortion images. In addition to desynchronization, the screen-shooting process also introduces strong pixel distortions such as illumination distortion and moiré distortion (Fang et al. 2022), which also seriously affect the survivability of the watermark.

Therefore, the core of realizing a watermarking scheme applicable to partial screen-shooting scenarios lies in de-

signing a watermarking mechanism that is robust to screen-shooting distortion and at the same time can realize the synchronization based on the partially captured image.

To achieve this goal, we propose the RoPaSS: a **ro**bust watermarking scheme for **pa**rtial **s**creen-**s**hooting. When training the watermarking encoder, RoPaSS introduces the flipping operation to generate symmetric watermark. Subsequently, using these symmetries as a reference, the homography estimation network in RoPaSS can estimate the image restoration parameters to achieve synchronization for partially captured images. Meanwhile, we incorporate a partial screen-shooting noise layer throughout the training process to enhance the robustness of network modules.

The main contributions of this paper are listed as follows:

1). We focus on a practical and advanced property for screen-shooting resilient watermarking, which is partially screen-shooting robustness, and provides an in-depth analysis of challenges and solutions for the scenario.

2). We propose a watermarking encoder-decoder framework that incorporates the flipping operation to artificially guide the generation of symmetric watermarks, as the basis for subsequent synchronization method.

3). We propose a synchronization method based on symmetry and homography transformation estimation for partial screen-shooting watermarks, where the design of the serial estimator can effectively improve synchronization accuracy.

4). Various experimental results demonstrate the good performance of the proposed scheme against the partial screen-shooting distortions.

Related Work

Screen-shooting resilient watermarking stands as one of the most challenging yet intriguing fields within watermark technology due to its intricate process. Fang et al. (Fang et al. 2018) delved into the analysis of screen-shooting distortion and introduced a method based on Scale-Invariant Feature Transform (SIFT) and Discrete Cosine Transform (DCT) to enhance screen-shooting robustness. StegaStamp (Tancik, Mildenhall, and Ng 2020) introduced a novel approach that models the physical world photography noise in a differentiable manner. This noise layer is incorporated into an end-to-end training process, which jointly trains an encoder for watermark embedding and a decoder for extraction. Building upon this, Fang et al. (Fang et al. 2022) proposed PIMoG, a deep learning-based, screen-shooting resilient watermarking scheme that achieves state-of-the-art performance. The cornerstone of PIMoG is its introduction of an effective differentiable screen-shooting noise layer, which more accurately enhances the simulation of complex distortions encountered in screen photography. Nonetheless, all these methods face difficulties when dealing with partial screen-shooting, where only a fragment of the watermarked image is captured. Recently, Hidden Code (Jia et al. 2022) has been proposed, which can locate and extract watermarked sub-images within a larger image. Nonetheless, the effectiveness of its location is associated with the integrity of the captured image.

Homography transformation is a global projective mapping between images from different perspectives and has been widely used in computer vision tasks like image stitching (DeTone, Malisiewicz, and Rabinovich 2016), gigapixel photography (Shao et al. 2021), and multispectral image fusion (Ying, Shen, and Cao 2021). In this paper, the homography estimation network is used to estimate the synchronization parameters for watermark extraction.

Method

The overall schematic diagram of the watermark framework is shown in Fig. 2. Considering that the output of the synchronization networks significantly affects the decoding results, it is difficult for networks to achieve an optimal balance by end-to-end training. In contrast, a two-stage training framework is adopted to alleviate this issue. In stage 1, the watermark encoder Enc and decoder Dec are trained. Then, in stage 2, three serial estimators are trained for watermark synchronization. The watermark estimator Est_w and symmetry estimator Est_s are used to estimate the symmetry of the captured image more accurately. The estimated symmetry and the original symmetry are fed into the homography estimator Est_h to estimate the synchronization parameters.

At the training stages, the homography transformation restoration is processed using dashed lines of the corresponding color. In the inference stage, the watermark extraction follows the process marked by the blue dashed line.

Flipping Operation

As shown in Fig. 3, in the flipping composite operation, the initial sub-block 'b' is first flipped along the vertical, horizontal, and diagonal axis to generate sub-blocks 'd', 'p', and 'q'. Then they are composed to generate the symmetric block. In this example, the generated symmetric block contains $4 \times 4 = 16$ sub-blocks, thus it has $3 \times 3 = 9$ symmetric centers, located at the intersection of every four sub-blocks. Then, the flipping concatenate operation can be regarded as the reverse implementation of the flipping composite.

Encoder

The encoder generates the watermark sub-block, which is subsequently used in the symmetric watermark generation. The encoder has two inputs: the host image I_o and the watermark message M. The host image I_o is a 128×128 RGB image. Before being fed into the encoder, it is transformed into a $(16 \times 3) \times 32 \times 32$ tensor by the flipping concatenate operation, as shown in Fig. 3, providing the awareness of the image pixels at all symmetric positions for the encoder. The watermark message M is a bit sequence of length L. In the encoder, M is transformed into a 1×1024 vector through a fully connected layer, then reshaped into a $1 \times 32 \times 32$ tensor. After that, it passes through $2\times$ConvINRelu blocks, generating a $64 \times 32 \times 32$ watermark tensor. The input host image tensor is transformed into a $64 \times 32 \times 32$ tensor through $4\times$ConvINRelu blocks and then concatenated with the watermark tensor. Then, the concatenated tensor passes through $4\times$ConvINRelu blocks and is concatenated again with the watermark tensor. Finally, through $6\times$ConvINRelu blocks, a $3 \times 32 \times 32$ watermark sub-block is generated.

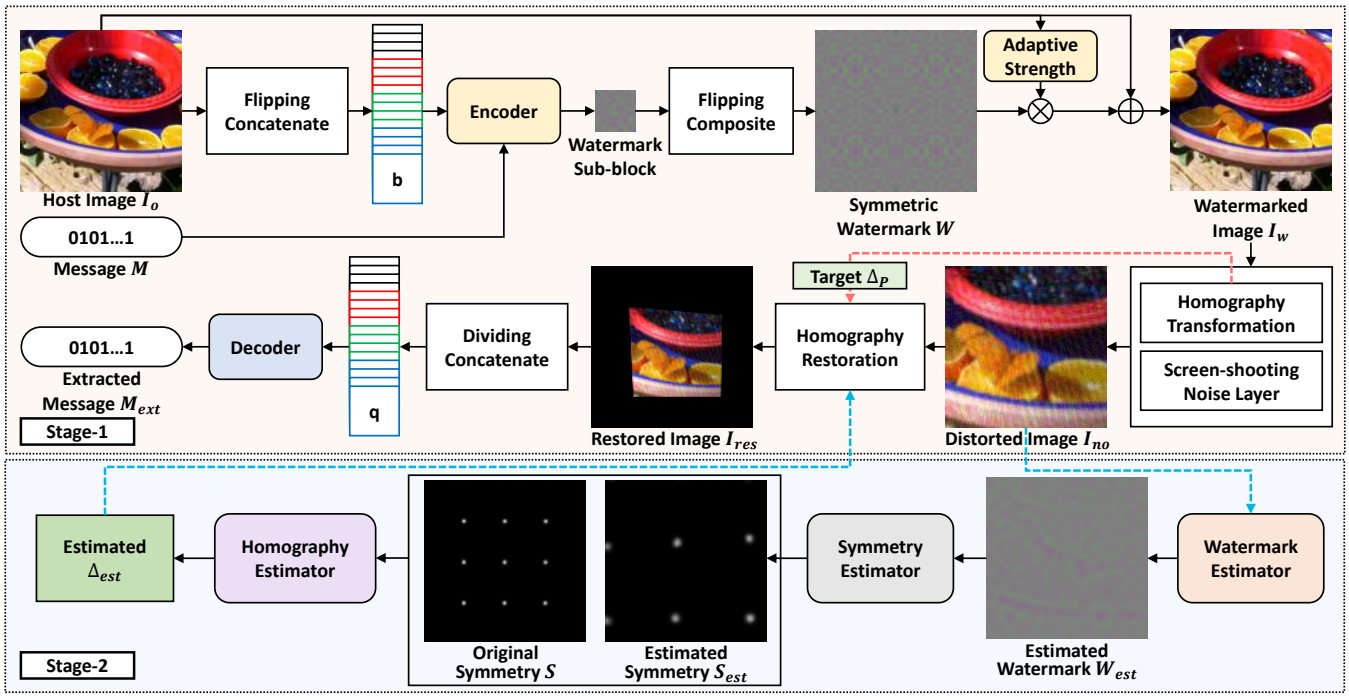

Figure 2: The framework of the proposed method. It contains five main network models: watermark encoder, decoder, watermark estimator, symmetry estimator, and homography estimator. These models are trained in two stages to respectively realize robustness to image processing distortion and desynchronization distortion during the partial screen-shooting process. Details of the flipping operation are shown in Fig. 3.

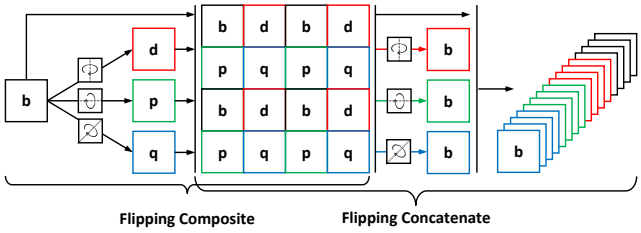

Figure 3: Schematic diagram of flipping composite and concatenate operations. Symbols 'bdpq' denote the state of the sub-blocks, and colors denote the positions.

Symmetric Watermark Embedding

The watermark sub-block generated by the encoder is adopted as the initial block for the flipping composite operation to generate the symmetric watermark W. Meanwhile, we use an adaptive strength matrix α to improve the watermark robustness. Specifically, α is generated by passing the host image I_o through 2×ConvINRelu blocks, which has the same size as the symmetric watermark W. Finally, the watermarked image I_w is generated as follows:

$$I_w = I_o + \alpha \times W. \tag{1}$$

Then, the loss function of the watermarking process is:

$$\mathcal{L}_I = MSE(I_o, I_w). \tag{2}$$

Partial Screen-shooting Noise Layer

In this paper, we categorize the distortions caused by the partial screen-shooting process into two types. The first type, image processing distortion, alters pixel values, while the second type, desynchronization distortion, affects pixel positions. We simulate the former based on the screen-shooting noise layer proposed in PIMoG (Fang et al. 2022), which is currently the best simulation of this distortion.

Then, the desynchronization distortion can be modeled as homography transformation, also known as projective transformation or perspective transformation, which maps points in one plane to the corresponding points in another plane in a different viewpoint. In this paper, coordinates set of the four original corner points of the watermarked image is denoted as P_0. Then, Δ_P is regarded as the offset of P_0 undergoing the homography transformation with a transformation matrix H. The transformed coordinates set is $P' = P_0 + \Delta_P$. Considering that H has eight unknown variables, equations can be formed using the coordinates of four P_0 and P' to solve for H. In the sequel, we use $H_{\{P_A \to P_B\}}$ to denote the homography transformation that maps P_A to P_B.

In the training, we adjust the random generation range of Δ_P to move the image corners outwards, as shown in Fig. 4, to simulate the partial screen-shooting distortion. Then, the homography transformation in the distorted image I_{no} will be restored according to the estimated Δ_{est} as follows:

$$\begin{aligned} H_{res} &= H_{\{P_0 + \Delta_{est} \to P_0\}}, \\ I_{res} &= \phi(I_{no}, H_{res}), \end{aligned} \tag{3}$$

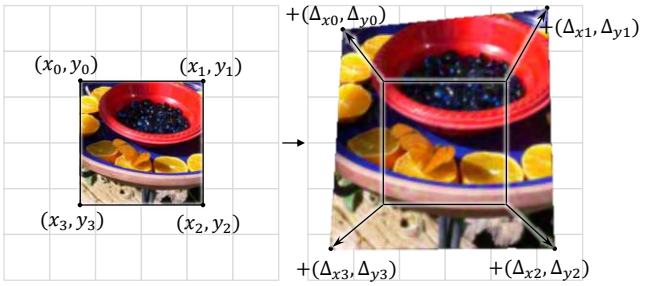

Figure 4: Partial screen-shooting simulation via homography transformation. Δ_P extends the image corners outward, resulting in a partial image captured from the original coordinates (the black square in the center of the right figure).

where I_{res} denotes the restored image and the function $\phi(I, H)$ transforms its input image with homography transformation H. In the training stage 1, we set $\Delta_{est} = \Delta_P$ to correctly re-synchronize the distorted watermarked image. In training stage 2 and inference stage, Δ_{est} is estimated by homography estimator from partial screen-shooting image.

Decoder

The restored image I_{res} is divided into multiple non-overlapping blocks with the same size as the watermark sub-block. These sub-blocks are directly concatenated together as input of the decoder. Thus, the decoder extracts the watermark message from all four states of the watermark sub-blocks. The architecture of decoder consists of five ConvIN-Relu blocks, nine "residual" blocks (He et al. 2016), and one fully connected layer, where the downsample operation is carried out one time in the "residual" blocks. The objective of the decoder is to minimize the difference between the extracted message and the original one:

$$\mathcal{L}_M = MSE(M, M_{ext}). \tag{4}$$

Watermark Estimator

The goal of the watermark estimator is to estimate the watermark component W_{est} from I_{no} to improve the accuracy of the following symmetry estimator. We employ U-Net (Ronneberger, Fischer, and Brox 2015) as the network architecture and the corresponding loss function can be written as:

$$\begin{aligned} \mathcal{L}_{EW} &= MSE(W_H, W_{est}), \\ &= MSE(\phi(I_w - I_o, H_{\{P_0 \to P_0 + \Delta_P\}}), W_{est}), \end{aligned} \tag{5}$$

where W_H is the watermark residual that undergoes the same homography transformation as I_w.

Symmetry Estimator

We define the symmetry map S to reflect the symmetry magnitude of one image. Ideally, the symmetry magnitude will have a local maximum at the symmetry center of one image. Thus, the original symmetry map S of the watermark W can be obtained by setting 1 on an all-0 map at the 9 positions corresponding to 9 symmetric centers of W; see Fig. 2. To avoid overfitting, S are Gaussian smoothed to achieve label smoothing. The goal of the symmetry estimator Est_s is to calculate the symmetry map from the estimated watermark W_{est}, which is one input of the subsequent homography estimator. The proposed Est_s architecture is based on the HRNet (Wang et al. 2020), yet removes the downsampling operation. The loss function is as follows:

$$\begin{aligned} \mathcal{L}_{ES} &= BCE(S_H, S_{est}), \\ &= BCE(\phi(S, H_{\{P_0 \to P_0 + \Delta_P\}}), S_{est}), \end{aligned} \tag{6}$$

where S_H is the original symmetry map S that undergoes the same homography transformation as I_w.

Homography Estimator

Finally, we employ HomoNet (DeTone, Malisiewicz, and Rabinovich 2016) as the homography estimator Est_h, which receives both estimated symmetry map S_{est} and the original S and outputs the estimated four-point offset Δ_{est}. The corresponding loss function are:

$$\mathcal{L}_{EH} = L_1(\Delta_P, \Delta_{est}). \tag{7}$$

Training Strategy

In stage 1, the training loss is as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_I + \lambda_2 \mathcal{L}_M, \tag{8}$$

where λ_1 and λ_2 are set as 400 and 1 by default. In stage 2, the encoder and decoder are fixed and the training loss is:

$$\mathcal{L} = \mathcal{L}_{EW} + \mathcal{L}_{ES} + \mathcal{L}_{EH}. \tag{9}$$

Considering the influence of these modules on each other, the above models will be trained sequentially.

Experimental Results

Implementation Details

In the training stage, we randomly select 10000 images from the COCO dataset (Lin et al. 2014) as the training set. Homography transformation and restoration are implemented using Kornia (Riba et al. 2020). The proposed RoPaSS is implemented by PyTorch (Collobert, Kavukcuoglu, and Farabet 2011) and executed on NVIDIA RTX A6000.Training images are randomly cropped and resized to $128 \times 128 \times 3$ pixels, and the watermark message is randomly generated with length $L = 30$. The batch size is 64.

In experiments, we use images from the USC-SIPI dataset (USC-SIPI 2022) as host images. PSNR and SSIM evaluate visual quality, while extraction accuracy (ACC) measures robustness. We compare RoPaSS with one traditional watermarking scheme (SSRW (Fang et al. 2018)) and three deep learning-based schemes (StegaStamp (Tancik, Mildenhall, and Ng 2020), PIMoG (Fang et al. 2022), and Hidden Code (Jia et al. 2022)), all exhibiting screenshot robustness. After adjusting the embedding strategy, the watermark embedding density in all schemes is less than or equal to RoPaSS. The default capturing device is Redmi K40, and the display is BenQ PD2705U. All watermarked images are captured under the same conditions in the comparison experiments.

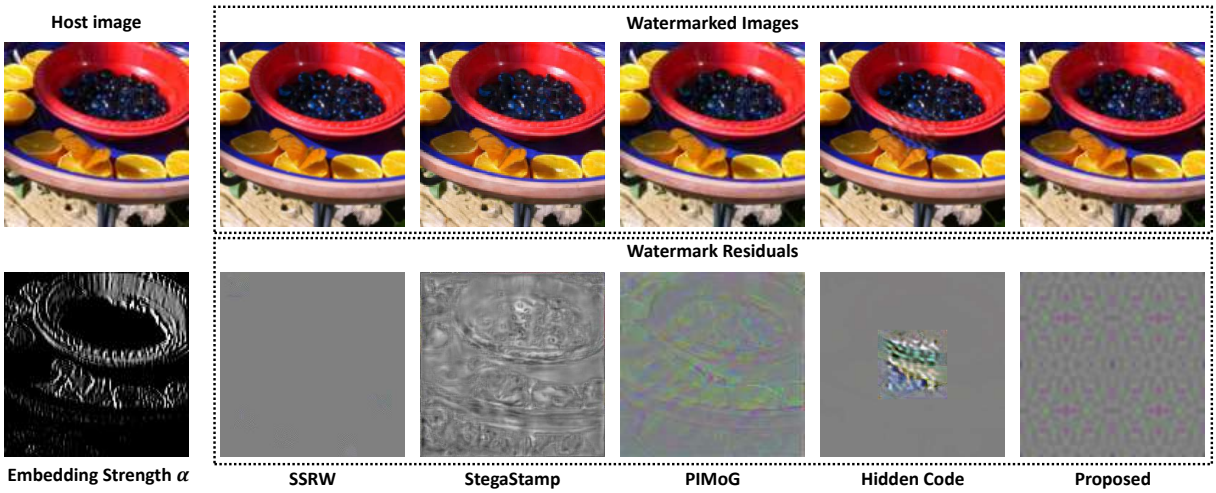

Figure 5: Visual examples of watermarked images I_w and watermark residuals of the proposed watermarking scheme and the compared schemes. The left side displays the host image I_o and the proposed adaptive embedding strength α as references. The watermark residual is calculated by $(I_w - I_o) \times 2 + 128$ and the strength matrix α is normalized for better observation.

Methods	SSRW	StegaStamp	PIMoG	Hidden Code	Proposed
PSNR (dB)	35.674	35.274	34.892	32.243	**35.902**
SSIM	**0.9816**	0.9721	0.9531	0.9711	0.9744

Table 1: The PSNR and SSIM values of each method.

Visual Quality

In this section, we compare the visual quality of the proposed watermarking scheme with comparison schemes, in which these watermarked images will be used in the following screen-shooting experiments. Based on the source code and pre-trained models released, we adjusted the embedding strength of part comparison schemes to keep the visual quality at the same level. As shown in Table 1, the proposed scheme has the highest PSNR value of 35.902 dB and the second highest SSIM value of 0.9744. When calculating the SSIM value, the size of all watermarked images is resized to 512×512 to exclude the image size effect.

Fig. 5 shows some visual examples of watermarked images. The proposed RoPaSS has similar overall visual quality as StegaStamp and Hidden Code. And the proposed adaptive embedding model achieves adaptive embedding strength α to a certain degree. It has slightly improved the visual quality (around 1% LPIPS decrease) and improved the decoding accuracy (around 1.4%). Yet the overall visual is still dominated by symmetry due to the need for symmetry in the synchronization process.

Screen-shooting Robustness

In this section, we evaluate the screen-shooting robustness of the proposed watermarking scheme and compare it with comparison schemes. Specifically, we refer to the experimental settings of PIMoG, and all images are captured completely and synchronized correctly.

distance (cm)	20	30	40	50	60
SSRW	95.70%	94.14%	93.56%	89.06%	83.40%
StegaStamp	96.63%	95.55%	95.38%	93.50%	92.63%
PIMoG	97.92%	97.92%	96.25%	95.42%	94.17%
Hidden Code	97.83%	97.32%	96.81%	95.41%	**96.94%**
Proposed	**98.33%**	**97.92%**	**97.08%**	**96.25%**	95.00%

Table 2: Extraction ACC at varying shooting distances.

Distance Test In this section, we test the effect of shooting distance on watermarking performance. The watermarked images are displayed on the screen and captured with mobile phones at different shooting distances ranging from 20cm to 60cm in 10cm steps. As shown in Table 2, the proposed watermarking scheme maintains an accuracy higher than 95% at all shooting distances. At distances from 20 to 50cm, the proposed scheme has the highest extraction accuracy, which is above 96% for all. At longer distances of 60cm, the extraction accuracy of the proposed scheme is slightly lower than Hidden Code, but still the second-highest.

The performance degradation of the proposed scheme at long distances may result from pixel fusion during long-distance shooting, leading to the loss of watermark details.

Angle Test In this section, we test the effect of shooting angle on watermarking performance. The shooting distance is fixed to 30cm, and then the watermarked image displayed on the screen is captured using a mobile phone at different angles. The shooting angle ranges from left 40° to right 40° and up 40° to down 40° with the intervals of 10°.

The extraction results are shown in Table 3. The proposed watermarking scheme has the best extraction accuracy at most of the shooting angles and achieves an extraction accuracy higher than 96% at all tested shooting angles. The experimental results indicate that the proposed watermark-

Angles (°)	Left 40/Up 40	Left 30/Up 30	Left 20/Up 20	Left 10/Up 10	Right 10/Down 10	Right 20/Down 20	Right 30/Down 30	Right 40/Down 40
SSRW	95.90%/96.09%	94.73%/93.95%	95.31%/95.31%	94.14%/95.70%	94.73%/95.51%	94.73%/95.90%	93.95%/94.53%	95.70%/96.09%
StegaStamp	95.88%/94.13%	96.00%/95.50%	94.50%/94.50%	94.38%/93.75%	95.75%/95.63%	94.50%/94.25%	94.88%/95.13%	95.75%/94.75%
PIMoG	97.50%/96.25%	**97.92%**/96.67%	95.83%/96.67%	95.83%/96.67%	96.67%/97.92%	97.92%/**97.92%**	**97.92%**/95.83%	97.92%%/**97.92%**
Hidden Code	96.56%/96.56%	97.19%/97.07%	95.41%/96.68%	94.64%/96.68%	95.28%/95.28%	98.34%/95.41%	95.92%/96.30%	95.92%/97.45%
Proposed	**97.92%/97.08%**	96.25%/**97.50%**	**97.50%/97.08%**	**96.67%/97.50%**	**97.08%/98.33%**	**98.75%**/97.08%	96.67%/**97.92%**	**98.75%**/97.08%

Table 3: Extraction ACC with different shooting angles.

Devices	iPhone 15 Pro	iPhone 14 Pro	HUAWEI Mate 60	Redmi K40
BenQ	100%	99.58%	99.58%	98.33%
Lenovo	100%	99.58%	100%	98.75%
RedmiBook	100%	99.17%	100%	100%

Table 4: Extraction ACC with different shooting devices.

Percentage	100%	90%	80%	70%
Hidden Code	95.66%	47.70%	57.78%	49.36%
Proposed	**96.25%**	**97.08%**	**95.42%**	**94.17%**
Percentage	60%	50%	40%	30%
Hidden Code	45.03%	52.42	44.90%	51.40%
Proposed	**92.92%**	**93.58%**	**91.92%**	**85.58%**

Table 5: Extraction ACC with varying shooting percentages.

ing scheme can handle most of the screen-shooing leakage scenarios, as attackers usually shoot at smaller angles to ensure the readability of the image content.

Device Test This section conducts experiments with various devices to assess the universality of the proposed watermarking scheme. Specifically, we added three shooting devices: iPhone 15 Pro, iPhone 14 Pro and HUAWEI Mate 60, and two display devices: Lenovo ThinkVision T22i-10 and the laptop screen of RedmiBook Pro 15. The shooting distance is set at about 20cm. The results are presented in Table 4, with some device names abbreviated for clarity.

It can be seen that the proposed watermarking scheme achieves more than 98% extraction accuracy for all tested shooting devices, proving its excellent device universality. Notably, all devices outperform the default Redmi K40, with extraction accuracy exceeding 99%. This is likely due to the superior imaging quality of the other devices, highlighting that advancements in smart device imaging enhance the traceability performance of our watermarking scheme.

Partial Screen-shooting Robustness

In this section, the robustness of the proposed watermarking scheme for the partial screen-shooting process is evaluated, and most of the experimental setups are consistent with previous section. The difference is mainly in the screen-shooting conditions. We adjust the imaging frame of the shooting device to capture square images, which are downsampled to the appropriate size as input to the extraction process **without** any manual calibration or synchronization. Thus, for wrongly synchronized images, the decoder still performs decoding and obtains an extraction result that is close to a random guess with about 50% accuracy.

Screen-shooting Percentage Test Here, we test the effect of screen-shooting percentage, the percentage of watermarked image pixels in partially captured images, on extraction accuracy. Considering the complexity of the screen-shooting process, it is difficult to determine the screen-shooting percentage precisely before the capture, but it can still be approximated. The field of view (FOV) is fixed for each shooting device for the default setting, which makes the captured image size s approximately proportional to the shooting distance d. Let d_0 represent the distance required for capturing the full watermarked image, with the length of the complete image side denoted as s_0. To capture an image with a pixel ratio of $n\%$, we should capture the image with a side length of $\sqrt{n\%} \cdot s_0$ at the distance $\sqrt{n\%} \cdot d_0$. To make the distances corresponding to different screen-shooting percentages more distinguishable, it is better to set a large d_0. Therefore, we display the watermarked image full screen and measure the corresponding d_0=30cm. Then, the mapping between screen-shooting percentage and distance can be calculated. For example, to capture a watermarked image with a 50% pixel percentage, the shooting distance should be $\sqrt{0.5} \times 30$ =21.21cm in current settings.

We evaluate the extraction accuracy of the proposed watermarking scheme across screen-shooting percentages ranging from 100% to 30%, with steps of 10%. It should be noted that variations in the shooting areas of watermarked images commonly cause extraction accuracy differences, especially in cases of partial screen-shooting with a small percentage. Thus, when the screen-shooting percentage is less than 60%, for each watermarked image, we capture it in the center and the four corners respectively, and take the average of the extraction accuracies of the five images.

The experimental results are shown in the Table 5. The proposed RoPaSS has an extraction accuracy of over 90% for screen-shooting percentages above 40%, demonstrating its excellent partial screen-shooting robustness, which is lacking in existing screen-shooting watermarking schemes. Among the comparison schemes, only Hidden Code is designed with a partial watermark localization strategy in the extraction process. Other methods require capturing the entire image and do not have synchronous capabilities for partially captured images, resulting in ACC lower than 53%. The extraction accuracy of Hidden Code reaches 95.66% on the uncorrected fully captured watermarked images. How-

Angles (°)	Left 40/Up 40	Left 30/Up 30	Left 20/Up 20	Left 10/Up 10	Right 10/Down 10	Right 20/Down 20	Right 30/Down 30	Right 40/Down 40
Accuracy	72.92%/75.83%	89.58%/87.50%	86.67%/87.50%	87.08%/89.58%	86.25%/87.92%	88.75%/89.17%	89.17%/87.08%	69.17%/77.50%

Table 6: Extraction ACC with different shooting angles in the partial screen-shooting process.

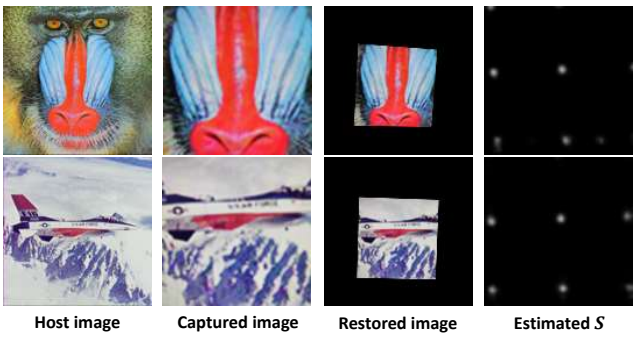

Figure 6: Synchronization examples.

Methods	StegaStamp	PIMoG	Proposed Enc & Dec
PSNR (dB)	28.230	30.346	31.674
ACC	57.36%	53.98%	59.94%

Table 7: Performance of end-to-end training.

Methods	w/o Est_s, Est_w	w/o Est_s	w/o Est_w	Proposed
PSNR (dB)	25.235	28.349	23.856	**35.902**
ACC	57.13%	51.75%	72.84%	**97.64%**

Table 8: Ablation study of synchronization method.

ever, this performance significantly deteriorates when the screen-shooting percentage decreases to 90%. The reason may be that the watermark localization network of Hidden Code was not trained in screen-shooting scenarios, resulting in watermark synchronization failures.

Fig. 6 shows some examples of the proposed method in the experiment of 30% screen-shooting percentages. These images have 100% watermark extraction accuracy. According to the restored images, their pixel percentages are 28.54% and 28.17% of the original image, confirming the rationality of the proposed experimental settings.

Angle Test We performed the angle test again under partial screen-shooting conditions. The shooting distance is fixed to the distance corresponding to 50% screen-shooting percentage, and other experimental settings are the same as Distance Test. The experimental results are shown in Table 6, that the proposed scheme can extract watermark messages from partially captured images with nearly 90% accuracy when the shooting angle is less than or equal to 30°. This indicates that the proposed watermarking scheme can adapt to more complex partial screen-shooting scenarios.

Ablation Study

Necessity of Synchronization Partial screen-shooting distortion complexly changes the value and position of watermarked image pixels, whose robustness is difficult to realize through end-to-end training with the corresponding noise layers. We select the comparative methods with trainable source code and end-to-end train them under the proposed partial screen-shooting noise layer with the same settings as RoPaSS. We also apply the same training setting to our proposed Enc and Dec. As shown in Table 7, without synchronization, end-to-end training only achieves watermarking accuracies below 60% under partial screen-shooting distortions, even under lower visual quality constraints.

Rationality of Synchronization Design We remove some of the estimators on the proposed synchronization framework and retrain stage 2 to verify the rationality of the current design. As shown in Table 8, the method without symmetry estimator Est_s and watermark estimator Est_w only uses the homography estimator Est_h to estimate the synchronization parameters from the partially captured image. This setting makes training difficult to converge with poor performance. The reason is that the proposed method embeds the watermark in the whole image to achieve robustness to the partial shooting of arbitrary image regions, whose pattern has no explicit spatial features or embedding boundaries for estimation. In the method without Est_s, Est_h struggles to learn the relationship between the original symmetry and the estimated watermark, and its training process is similarly difficult to converge. The method without Est_w has a relatively rational framework, but estimating symmetry directly from captured images is susceptible to screen-shooting distortion and image content, affecting subsequent synchronization and extraction accuracy (72.84%).

Our proposed synchronization method combines Est_w and Est_s to enhance the accuracy of symmetry estimation from the captured image, resulting in better watermark extraction performance with ACC = 97.64%.

Conclusion

RoPaSS is the first watermarking scheme, to our knowledge, that is robust against partial screen-shooting process. It enables synchronization and extraction of partially captured watermarked images through flipping operations for symmetry and estimation networks for synchronization parameters. We use a two-stage training strategy and train estimation networks sequentially for better convergence. Extensive experiments highlight RoPaSS's excellent robustness to partial captures. Future efforts will pursue better content adaptivity and long-distance shooting resilience.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant 62402469, 62072421, U2336206, 62472398, by Xiaomi Young Talents Program, and by Fundamental Research Funds for the Central Universities under Grant WK2100000041.

References

Briscoe, M. D. 2022. The paperless office twenty years later: Still a myth? *Sustainability: Science, Practice and Policy*, 18(1): 837–845.

Collobert, R.; Kavukcuoglu, K.; and Farabet, C. 2011. Torch7: A Matlab-like Environment for Machine Learning.

DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2016. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*.

Fang, H.; Chen, K.; Qiu, Y.; Liu, J.; Xu, K.; Fang, C.; Zhang, W.; and Chang, E.-C. 2023. DeNoL: A Few-Shot-Sample-Based Decoupling Noise Layer for Cross-channel Watermarking Robustness. In *Proceedings of the 31st ACM International Conference on Multimedia*, 7345–7353.

Fang, H.; Chen, K.; Qiu, Y.; Ma, Z.; Zhang, W.; and Chang, E.-C. 2024. DERO: Diffusion-Model-Erasure Robust Watermarking. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2973–2981.

Fang, H.; Jia, Z.; Ma, Z.; Chang, E.-C.; and Zhang, W. 2022. Pimog: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2267–2275.

Fang, H.; Zhang, W.; Zhou, H.; Cui, H.; and Yu, N. 2018. Screen-shooting resilient watermarking. *IEEE Transactions on Information Forensics and Security*, 14(6): 1403–1418.

Ge, S.; Fei, J.; Xia, Z.; Tong, Y.; Weng, J.; and Liu, J. 2023. A screen-shooting resilient document image watermarking scheme using deep neural network. *IET Image Processing*, 17(2): 323–336.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, M.; Wang, H.; Zhang, F.; and Xiang, Y. 2023. Exploring Accurate Invariants on Polar Harmonic Fourier Moments in Polar Coordinates for Robust Image Watermarking. *IEEE Transactions on Multimedia*.

Jia, J.; Gao, Z.; Zhu, D.; Min, X.; Zhai, G.; and Yang, X. 2022. Learning invisible markers for hidden codes in offline-to-online photography. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2273–2282.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Ma, Z.; Zhang, W.; Fang, H.; Dong, X.; Geng, L.; and Yu, N. 2021. Local geometric distortions resilient watermarking scheme based on symmetry. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(12): 4826–4839.

Min, X.; Gu, K.; Zhai, G.; Yang, X.; Zhang, W.; Le Callet, P.; and Chen, C. W. 2021. Screen content quality assessment: overview, benchmark, and beyond. *ACM Computing Surveys (CSUR)*, 54(9): 1–36.

Riba, E.; Mishkin, D.; Ponsa, D.; Rublee, E.; and Bradski, G. 2020. Kornia: an Open Source Differentiable Computer Vision Library for PyTorch. In *Winter Conference on Applications of Computer Vision*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, 234–241. Springer.

Shao, R.; Wu, G.; Zhou, Y.; Fu, Y.; Fang, L.; and Liu, Y. 2021. Localtrans: A multiscale local transformer network for cross-resolution homography estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 14890–14899.

Tancik, M.; Mildenhall, B.; and Ng, R. 2020. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2117–2126.

USC-SIPI. 2022. The USC-SIPI Image Database. Accessed: Mar. 2022 [Online]. Available: http://sipi.usc.edu/database/.

Wang, G.; Ma, Z.; Liu, C.; Yang, X.; Fang, H.; Zhang, W.; and Yu, N. 2024. MuST: Robust Image Watermarking for Multi-Source Tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 5364–5371.

Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. 2020. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10): 3349–3364.

Ying, J.; Shen, H.-L.; and Cao, S.-Y. 2021. Unaligned hyperspectral image fusion via registration and interpolation modeling. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–14.

Zhang, F.; Wang, H.; He, M.; and Xia, J. 2024. Robust Blind Symmetry-based Watermarking in the Frequency Domain Against Social Network Processing and Desynchronization Attacks. *IEEE Transactions on Circuits and Systems for Video Technology*.