Effectively Learning Moiré QR Code Decryption from Simulated Data

Yu Lu †,Hao Pan†*, Feitong Tan‡, Yi-Chao Chen†,Jiadi Yu†, Jinghai He♦, Guangtao Xue†*,
† Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
‡School of Computing Science, Simon Fraser University, Canada
♦Industrial Engineering and Operations Research Department, University of California, Berkeley, USA
Email:{yulu01,panh09,yichao,jiadiyu, xue-gt}@sjtu.edu.cn, feitongt@sfu.ca, jinghai_he@berkeley.edu,

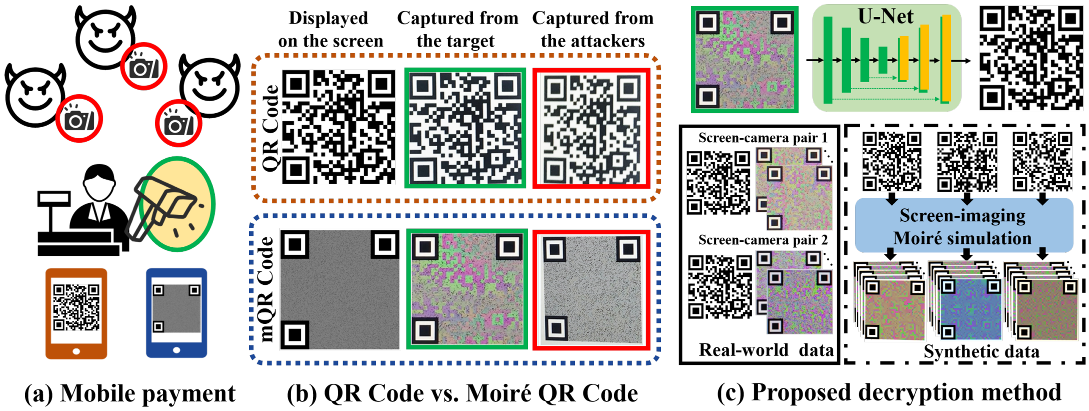


Fig. 1: (a) In mobile payment scenarios, the cashier is the target receiver and the surrounding attackers are the illegal receivers. (b) The traditional QR code image can be obtained by both the cashier and the attackers, while the Moiré QR code is highly secure and cannot be spied out by attackers. (c) A deep learning based method is proposed to achieve fast decryption of Moiré QR codes. Due to the fact that the Moiré pattern is highly sensitive to the relative pose of screen and camera as well as their hardware variability, a physical screen-imaging Moiré simulator is developed to augment the training data

***Abstract*—Moiré QR Code is a secure encrypted QR code system that can protect the user's QR code displayed on the screen from being accessed by attackers. However, conventional decryption methods based on image processing techniques suffer from intensive computation and significant decryption latency in practical mobile applications. In this work, we propose a deep learning-based Moiré QR code decryption framework and achieve an excellent decryption performance. Considering the sensitivity of the Moiré phenomenon, collecting training data in the real world is extremely labor and material intensive. To overcome this issue, we develop a physical screen-imaging Moiré simulation methodology to generate a synthetic dataset that covers the entire Moiré-visible area. Extensive experiments show that the proposed decryption network can achieve a low decryption latency (0.02 seconds) and a high decryption rate (98.8%), compared with the previous decryption method with decryption latency (5.4 seconds) and decryption rate (98.6%).**

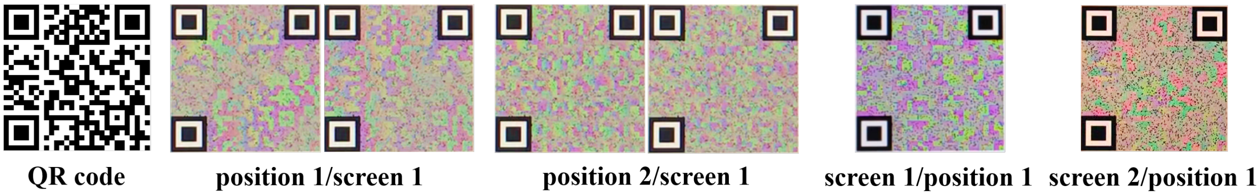
***Index Terms*—Secure QR code, Moiré pattern, Image-to-image translation, Simulated data**

I. Introduction

Quick Response (QR) code has become a widely used method in near-field communication due to its fast readability and the popularization of smartphones with built-in cameras. Unfortunately, traditional QR code systems are vulnerable to security risks in the form of Replay attacks [1]–[3] and Synchronized Token Lifting and Spending (STLS) attacks [4]. In these attack scenarios, attackers surreptitiously intercept images of victims' QR codes, which can lead to serious private information leakage, and financial losses [5], [6], etc. Some special physical materials are currently deployed to protect the contents (*i.e.*, QR code image) displayed on the screen by restricting the visible angle, *e.g.*, privacy filter (also called anti-peep film) [7]. However, these privacy filters are hardly useful if the attackers spy on the victim's smartphone screen from the same visible area (around 60°) facing the screen.

Several works utilize the Moiré phenomenon that existed in the interaction between the digital screens and cameras and proposed Moiré QR code (mQR code) system to improve the security of the standard QR code [8], [9]. In physics, Moiré is a visual geometrical design that occurs when one set of straight lines or curves is superposed simultaneously onto another set [10]. Pan *et al.* [8] proposed a phase modulation method to encrypt a standard QR code image into a camouflaging pattern (see Fig. 1(b)) with a similar spatial frequency to the Color Filter Array (CFA) in the camera. Only when the target receiver's camera is positioned in the Moiré-visible area, where the encrypted QR code image can be projected to the imaging plane with the similar spatial frequency to the CFA, the information of the original QR code can be

[*] Guangtao Xue and Hao Pan are the corresponding authors.

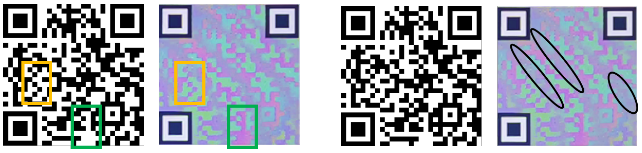


(a) iPhone 7Plus

(b) Xiaomi 11Pro

Fig. 2: (a) Examples of encrypted QR code images captured by iPhone 7 Plus in Moiré-visible positions. (b) Examples of encrypted code images captured with a Xiaomi 11Pro when displayed on different screens

revealed (see the green box in Fig. 1(b)). When the attacker's camera captures the image of the encrypted QR code outside the Moiré-visible area, a meaningless gray image is obtained (see the red box in Fig. 1(b)), which prevents information in the QR code image from leaking out.



(a) Color inversion phenomenon

(b) Blur phenomenon

Fig. 3: Issues of decryption Moiré code images

To extract the original QR code information correctly, in addition to capturing the Moiré QR code from the intended perspective (*i.e.*, in the Moiré-visible area), post-processing is required to convert the captured Moiré QR code to a standard QR code. The main reason is that in the scenarios where a user holds the smartphone camera and decodes the encrypted QR code, the imperfect match of the projected encrypted QR code and the CFA results in *blurred parts* and *color inversion* of the QR code blocks when captured by the camera. A demonstration of *color inversion* is shown in Fig. 3a. In the yellow box, black blocks of the original QR code are mapped to purple ones in the captured picture. However, in the green box, black blocks are mapped to green ones. Likewise, as shown in Fig. 3b, the portions boxed by the red lines illustrate the blur phenomenon. It is difficult to determine whether the colors in these portions are green or purple. . The decryption algorithm previously proposed in [8] must go through all the pixels on the captured image to determine the presence of blur and color inversion and relies on techniques to fuse information from multiple frames to identify the original QR code. This type of decryption method based on traditional image processing techniques and multiple frames suffers from a huge computational overhead and significant latency when deployed on mobile devices, which limits the use of Moiré QR code system in practice.

In this paper, we propose a deep learning based method for efficiently decrypting Moiré QR code (as shown in Fig. 1(c)). By employing the image-to-image translation technique, our proposed method can decode the embedded message of Moiré QR code images both faster and more accurately.

Although recent deep learning work has made great progress in supervised image-to-image translation task, the main concern is to ensure the robustness of the training data. The diversity and quantity of training data are the main determinants of the effectiveness and robustness of the deep learning based Moiré QR code decryption network. Unfortunately, it is almost unrealistic to collect real-world training data covering the entire Moiré-visible area which requires displaying the encrypted QR code images on the screens and capturing them with the cameras. One main reason is that the Moiré pattern in the captured images is very sensitive to the relative pose (*i.e.*, distance and rotation) between the screen and the camera (as shown in Fig. 2a). Therefore, to cover the entire Moiré-visible area, precise control of the relative pose between screen and camera is needed, which can vary with fine granularity (*e.g.*, $0.5mm$, $0.1°$). To obtain sufficient training data, the camera needs to be fixed at each pose to capture different QR code samples displayed on the screen. Therefore, the accuracy and stability of camera pose control are almost impossible to achieve by the user holding the smartphone. Another reason is the diversity of cameras and screens on the market. The camera modules built into smartphones have different hardware characteristics in terms of CFA filter quality, CMOS light sensitivity, etc. Likewise, different screens have a different color space due to the differences in LCD/OLED tubes, and display modes (*e.g.*, normal/night/low-blue). Therefore, the differences in camera and screen hardware also influence the Moiré QR code images (as in Fig. 2b).

To ensure the effectiveness of our deep learning based decryption model, in this paper, we propose a physical screen-imaging Moiré simulator to generate the Moiré QR code images when they are photographed in the entire Moiré-visible area. To accommodate the diversity of cameras and screens, we perform a data augmentation scheme on the simulated Moiré QR code images. The final robust synthetic dataset is used to train our decryption network.

The main contributions of our work are three-fold:

1) We propose a deep learning based Moiré code decryption method. It achieves a decryption latency of 0.02 seconds and a decryption rate of 98.8% for the Moiré QR code system. This reduces the average latency by 5.4 seconds compared with the multi-frame decryption method in [8].
2) We propose a screen-imaging Moiré simulation methodology that approximates the "physical transmission" (*i.e.*, the real screen and the subsequent image capture), and synthesize Moiré QR code images to improve the robustness of the training dataset. We also employ a data

augmentation scheme to cover the entire Moiré-visible area and physical screen-camera pairs.

3) We conduct extensive experiments to verify the effectiveness of the screen-imaging Moiré simulation. The results show that the model trained on the synthetic training dataset can achieve almost the same decryption performance as the one trained with the real-world dataset when limiting the types of the devices and screen-camera relative poses.

II. Related Work

A. Image Encryption

One related type of technique is information hiding, such as watermarking and stenography methods which inject given information into a cover media and hide its existence. Some related works propose the QR code hidden techniques to inject QR code images into videos or cover images, which obscures the existence of QR codes [11]–[13]. However, these techniques just make the QR code images unapparent to the human eyes, rather than the attackers' cameras; thus it cannot be applied to improve the security of the QR code system and prevent sneaking attacks. Image encryption is a technique that takes consecutive or random pixel bits of an image and modifies them collectively under specific rules, thereby leading to a complete set of new pixels, which differs from the original bits [14]. The image encryption methods are widely applied in digital image communication in public networks to prevent eavesdropping, illegal modification, duplication, etc. Nevertheless, these techniques are hardly applicable to the QR code systems. Because after the encrypted image is taken by the receiver (camera), serious problems such as distortion of RGB information at pixel level and blurring between pixel and pixel make it impossible to restore the original image information. Another problem is that the receiver end needs the decryption keys to retrieve the original image. However, in the QR code system, the screen and camera cannot be paired directly and the decryption keys are hardly transferred to the camera.

B. Moiré Phenomenon

There are considerable previous research efforts [15]–[17] dedicated to hiding images through Moiré patterns. Desmedt et al. [17] use Moiré patterns to secretly share information in realistic images. Lebanon et al. [15] exploit the superimposed grating patterns to create Moiré patterns of facial images which can be appreciated by humans. Tsai et al. [18] provide a novel pathway of Moiré art and visual decoding by superimposing grating images printed on separate transparencies. Hersch et al. [16] extend the understanding of the Moiré phenomenon and create moving Moiré components that can run up and down at different speeds and in various orientations when applying translation to the revealing layer. All the aforementioned approaches require two semi-transparent layers to overlap each other to reveal the hidden image. Distinguished from these works, Moiré QR code and [9], [19] exploit the nonlinear optical interaction between a camera (specifically the color filter array) and a camouflaging pattern to hide QR codes.

C. Image-to-image translation

Image-to-image translation techniques focus on learning a conditional image generation function that maps an input image from the source domain to a corresponding image in the target domain [20]. The generative adversarial networks (GAN) [21] and supervised learning [22]–[24] are extensively studied in image-to-image translations. Liang et al. [25] design a lightweight network for translating the low-frequency component with reduced resolution; they also propose a progressive masking strategy to efficiently refine the high-frequency images. Particularly, U-Net [26] wins the ISBI bioimage segmentation challenge. It consists of a contracting path that captures context and a symmetric expanding path that enables precise localization, which initiates a new route to complete the image-to-image translation task. This architecture can contribute to solving the problem of reconstruction from Moiré patterns to the original QR code.

III. Training with Moiré QR Simulator

To overcome the difficulties in collecting real-world data, we develop a Moiré QR code simulator that approximates real-word Moiré patterns. Its simulated images are then used to train the decryption model. The whole pipeline of the training procedure is shown in Fig. 4. It consists of four main procedures: **(i)** Generate the encrypted QR code I_E from the standard QR code I_S. **(ii)** Synthesize the Moiré QR code I_M by simulating the real-world Moiré pattern caused by physically displaying of the screen and imaging of the camera. **(iii)** Augment the color to account for the diversity in hardware settings of the different camera-screen pairs. **(iv)** Use a U-Net to decrypt the augmented Moiré QR code images $\hat{I}_M$ into the original QR code, and enforce the losses to minimize the difference.

A. QR Code Encryption and Its Secure Characteristics

Given a standard QR code, we apply the encryption algorithm proposed in [8] to convert the original QR code image into an encrypted one. As shown in the first and second figures in Fig. 6, the black and white blocks in the standard QR code are modulated as special patterns, *i.e.*, the alternating black and white *screen pixel*s with different sequences. Assuming a pinhole camera model, the encrypted QR code image (I_E) is projected through the lens onto the imaging plane in the camera, and we define the projected image as I'_E. If the *screen pixel* size in I'_E perfectly matches the image sensor pixel size[1] of the camera, the Moiré pattern is perfectly presented as the standard QR code (see Fig. 5b), and we define the perfect-match position as P_{pm}. This perfect-match position can also be formulated as the following equation:

$$\frac{L_S}{L_{CFA}} = \frac{D}{f} \tag{1}$$

[1] CFA is a tiny color filter over the image sensor (imaging plane) to capture color information [27], and each red/green/blue filter unit in the CFA has the same size as the single image pixel sensor in the imaging plane.

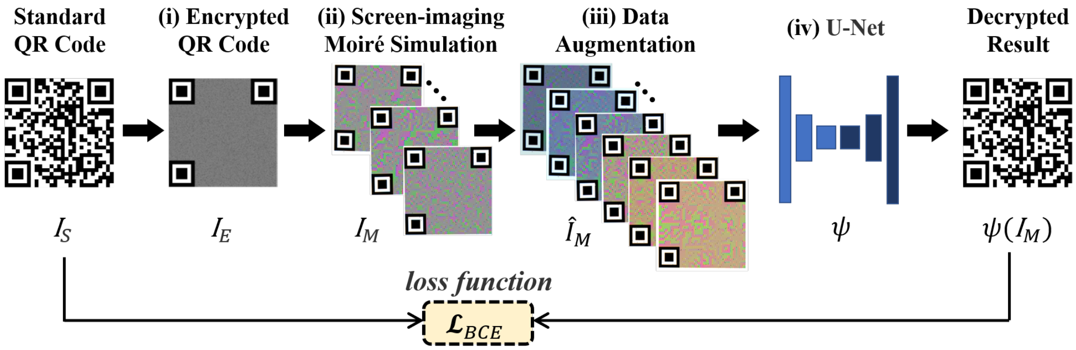
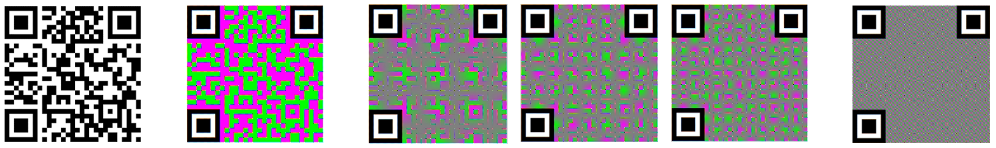


Fig. 4: Pipeline of the training process of deep learning based decryption model



(a) Original QR code

(b) At the perfect position P_{pm}

(c) In the Moiré-visible area. Left to right are getting further away from P_{pm}

(d) Out of moiré-visible area A_{mv}

Fig. 5: (a) Standard QR code image. (b)-(d) Moiré patterns of the encrypted QR code images when captured by the camera at different positions from the screen

where L_S is the length of the physical screen pixel (*i.e.*, the size of each LCD tube), L_{CFA} is the length of each color filter unit (the same as the length of the image pixel sensor), D is the distance between camera and screen that displays the encrypted QR code, and f is the focal length of the camera. We define the relative position between the camera and the screen that perfectly satisfies the Eq. 1 as the P_{pm}.

In practical scenarios, even if it's impossible for users to place the camera at the perfectly matched position, the Moiré pattern that contains the original QR code information is still visible (see Fig. 5c) when the camera is near the P_{pm} (*i.e.*, the *screen pixel* size in I'_E appreciates the image sensor pixel size to be matched). Therefore, we define the relative positions of the screen and the camera that approximately satisfies the Eq. 1 as the Moiré-visible area (*i.e.*, the legal QR code receiving area for the target), and denote it as A_{mv}. When I_E outside A_{mv}, the pixel information in the projected image I'_E cannot be captured by the image sensor due to the limited sampling rate. Therefore, the attackers in the vicinity(*i.e.*, far away from A_{mv}) can only get a gray image (see Fig. 5d) when they capture the encrypted QR code image, and cannot sniff the original QR code information.

B. Screen-imaging Moiré Simulation

The second stage is to simulate the Moiré pattern generated when a digital camera captures the encrypted QR code displayed on the screen. The procedures of our screen-imaging Moiré simulation system are illustrated in Fig. 6. Four essential procedures are: perspective transformation, Bayer CFA sampling, interpolation (also called demosaicing), and noise adding. In the following part, we will describe the details in the designs of the simulator.

1) Perspective Transformation: We simulate the image information captured on image plane before the Bayer CFA by employing a pinhole camera model to project an encrypted QR code image I_E in 3D space onto the camera plane. For simplification purposes, we use the camera coordinate system and set the screen pixel size and the image pixel sensor size (*i.e.*, the size of each color filter in the CFA) to be 1. To ensure the Moiré pattern is available, the encrypted QR code image is placed at the focal point in the simulation. As mentioned in Sec. III-A, the camera usually cannot be placed in the perfect-match position P_{pm}. Moreover, the vibration perturbations caused by the user's hand holding the smartphone result in various Moiré patterns. To generate a robust dataset incorporating all possible situations, we randomly perturb the translation (T_x, T_y, and T_z) and rotation of the encrypted QR code to simulate the entire Moiré QR code patterns captured by a camera in the whole Moiré-visible area A_{mv}.

2) Bayer CFA Sampling: After the encrypted QR code image is projected to the imaging plane through a random homography, the CFA filters out the color information, and an image sensor captures the light intensity of the encrypted QR code. We simulate the CFA sampling process and calculate the luminance by computing the overlapped information between each color unit in the CFA and each pixel in the encrypted QR code. For example, if a red color unit covers 80% of a white pixel and 20% of a black pixel in the encrypted QR code, then the luminance obtained by the image pixel sensor for that red color unit is 0.8.

As shown in the 4^{th} part in Fig. 6, when the black and white pixels of the encrypted QR code are arranged in the same pattern as the green filter in the Bayer CFA , the filtered color of each block will be 'purely green' or 'purple (red and blue)'. However, the perspective transformation will cause the black and white pixels in the encrypted QR code not be aligned with CFA, and this non-alignment further leads to inconsistencies

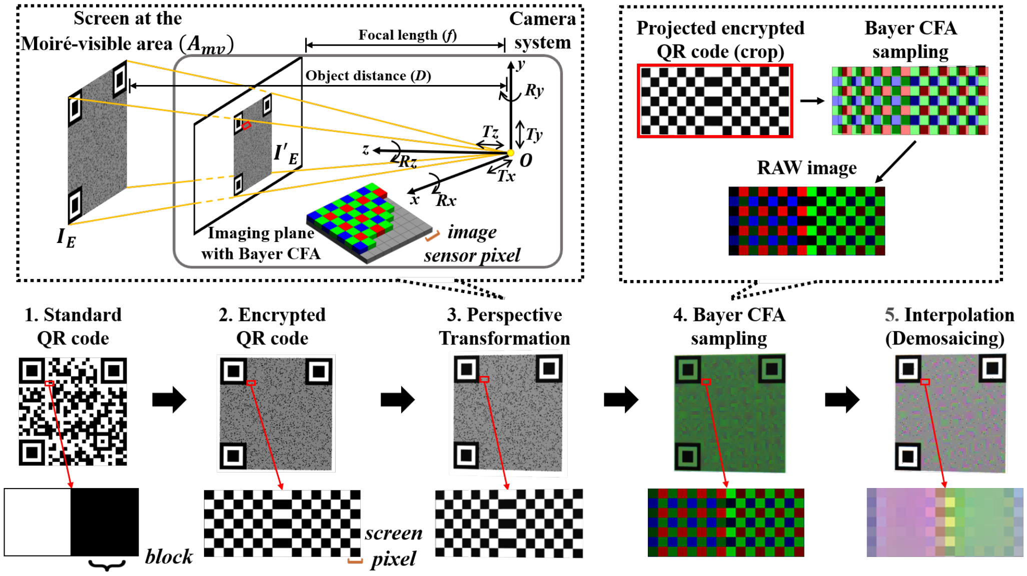


Fig. 6: Pipeline of encryption and screen-imaging Moiré simulation from the standard QR code

of colors in lightness (*i.e.*, blurring) and an inverse of Moiré color. The blurring can be easily seen in Fig. 5c, and the Moiré color inverse can be observed by comparing Figs. 5a with 5c. The blocks of the same initial color are shown to be chromatically different in the three Moiré QR code images captured at different positions. Therefore, the non-alignment between I'_E and CFA explains why a camera captures different Moiré patterns at different positions in the Moiré-visible area. Besides, the captured Moiré pattern is very sensitive to the relative position of the camera to the screen, since even small positional changes (*i.e.*, at the pixel size level of the image sensor) influence the captured Moiré pattern dramatically.

3) Interpolation: Interpolation (also known as demosaicing) algorithms are used to reconstruct a complete RGB image from an incomplete color sampling of a CFA (*i.e.*, RAW image). For example, one basic method is the bilinear interpolation, which considers the nearest 2×2 neighborhood of known pixel values around the computed position of an unknown pixel. This method takes the weighted average of these 4 pixels as the final interpolated value. However, one disadvantage is that it may smooth sharp and high-frequency textures. Other complex interpolation algorithms such as Smooth-hue Interpolation [28], High-Quality Linear Filter [29], Gradient-Based Threshold-Free [30], and Adams-Hamilton's interpolation [31], Variable Number of Gradients [32], Pixel Grouping [33], and Adaptive Homogeneity-Directed [34] have been applied to smartphone photography in order to obtain more detailed and clear resolution. Considering the fact that Moiré QR images captured by cameras do not have sharp or high-frequency textures and their colors vary smoothly, the spatial features of the Moiré phenomenon can be preserved once a reasonable interpolation algorithm is chosen. In the experiment, we use the bilinear interpolation algorithm and some mainstream linear interpolation algorithms for screen-imaging Moiré simulation respectively. The simulated results shown in Fig. 7 also confirm that the choice of interpolation algorithm does not have a substantial impact on our simulation. To speed up the calculation, we choose bilinear interpolation as the interpolation algorithm in our pipeline. The interpolated results are also very similar to the ones taken by smartphones with complex interpolation algorithms.

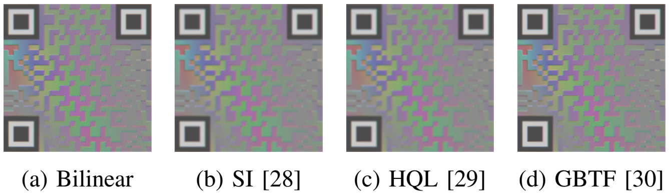

(a) Bilinear (b) SI [28] (c) HQL [29] (d) GBTF [30]

Fig. 7: Impacts of different interpolation algorithms

C. Data Augmentation

We design a data augmentation scheme to overcome the issues derived from the diversity of cameras and screens on the market. In detail, we approximate different color representations between the screen-camera pairs with a series of random affine color transformations (constant across the whole simulated Moiré QR images from Sec. III-B) as follows:

- Saturation: convert the simulated image from RGB to HLS, change the value of the S channel by a random multiplier $m \sim U[0.5, 1.5]$; then convert it back to the RGB color space.
- Brightness and contrast: adjust the brightness and contrast according to the formula: $g(i, j) = \alpha f(i, j) + \beta$, where $\alpha \sim U[0.6, 1.4]$, $\beta \sim U[-0.3, 0.3]$, $f(i, j)$ is the original pixel value, and $g(i, j)$ is the output pixel value.
- Color temperature: add random color offsets s_r, s_g to the Red/Green channels and subtract a random offset s_b to the Blue channel with $s_r \sim U[-0.4, 0.4]$, $s_g \sim U[-0.4, 0.4]$, $s_b \sim U[-0.4, 0.4]$, $s_r s_g \geq 0$ and $s_r s_b \leq 0$.

D. Deep Moiré QR Code Decryption

In this section, we propose a deep learning based decryption method. We use the notation Ψ to represent the function that transfers the I_M into the original QR code: $I_O = \Psi(I_M)$. Considering that the decryption method needs to cope with QR codes of different sizes, we use a U-Net [26] "encoder-decoder" style architecture for image-to-image translation. The encoder has one $Conv - ReLU$ layer with 64 4×4 spatial filters and seven $Conv - BN - ReLU$ layers with $[128, 256, 512, 512, 512, 512, 512]$ 4×4 spatial filters where $Conv$ downsamples by a factor of 2 and $ReLU$ is leaky with slope 0.2. The decoder has seven $Conv - BN - Dropout - ReLU$ layers with a dropout rate of 50% where $Conv$ upsamples by a factor of 2 and $ReLU$ is not leaky. After the last but one layer in the decoder, an eventual convolution is applied to map to RGB 3 channels, followed by a $Tanh$ function. U-Net can handle different input image sizes $N \times N$ (as long as $N = 2^k, k \in$ N+), and output the images with the same size $N \times N$. Accordingly, Moiré QR code images are also resized to the $2^{\lfloor \log_2 w+1 \rfloor} \times 2^{\lfloor \log_2 w+1 \rfloor}$, where w is the width of Moiré QR code images cropped from the captured image. The loss function is set as follows:

$$\mathcal{L}_{BCE}(\Psi) = \frac{1}{N^2}\sum_{i,j=0}^{N-1} -w_{ij}[I_{N_{ij}} log I_{O_{ij}} + (1 - I_{N_{ij}}) log(1 - I_{O_{ij}})] \tag{2}$$

where i, j the index of each pixel of the input and output images and w_{ij} is the weight given to the loss of each pixel.

E. Post-processing

Moreover, a decrypted image output from the deep Moiré decryption network needs to be post-processed to make the final result consistent with the original QR code and improve the decoding efficiency of the QR code reader. The post-processing process is described as follows:

1) Determine the block size (*i.e.*, the number of pixels occupied in each block) of the QR code according to its three position patterns in the corners, which are fixed as 7 blocks.
2) Segment the QR code by blocks, according to the number of pixels occupied by each block.
3) Binarize each block based on the mean value of all pixels in the box, unify each block of the QR codes into white (*i.e.*, $RGB(255, 255, 255)$) or black (*i.e.*, $RGB(0, 0, 0)$).

IV. Evaluation of Effectiveness of Moiré Simulation

A. Experiment Setup

We randomly generate 1000 messages (*e.g.*, each message consists of English letters and numbers) and use versions from 1 to 5 (*i.e.*, side length of QR symbol from 21×21 to 37×37) and level "M" error correction capability (*i.e.*, 15% data recovery) to generate the corresponding 1000 original QR code images.

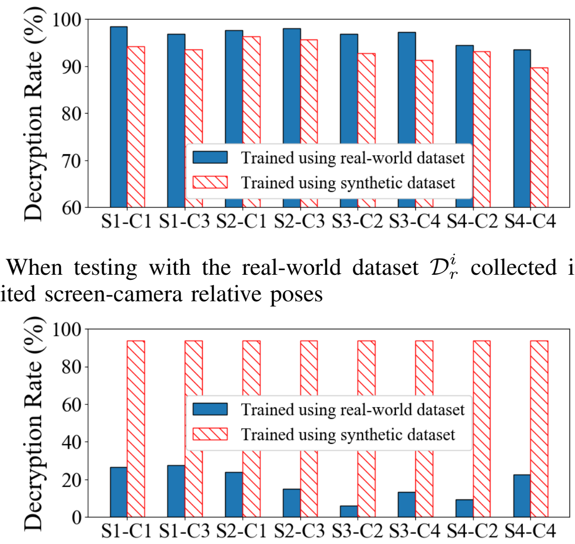
1) Synthetic Training Dataset: For the training dataset, we select 800 original QR code images to simulate the moiré QR code images. For each original QR code, we generate 100 Moiré patterns by varying the perspective transformation parameters (*i.e.*, T_x, T_y, T_z, and R_x, R_y, R_z in Fig. 6) for each standard QR code. In the data augmentation stage, we perform 10 random color transformations on each of the Moiré QR codes generated above. Thus, the synthesized $800 \times 100 \times 10$ Moiré QR images and their corresponding original images QR code images are treated as a training dataset that is fed into the U-Net based decryption network for training the model $\mathcal{M}_s$.

2) Real-world Test Dataset: For the test dataset, we encrypt the remaining 200 QR code images as encrypted QR code images and display them on the digital screens. Then, we use the cameras to capture a real test dataset (called $\mathcal{D}_s$) to evaluate the model $\mathcal{M}_s$, note that during the data collection, the test cameras are randomly placed throughout 3D space, both inside and outside of the Moiré-visible area. The test screens in this experiment are: iPhone 7Plus (S_1), iPhone 11 (S_2), Samsung S7 (S_3), and Huawei P40 (S_4); and the test cameras are: iPhone 6 (C_1), iPhone XS (C_2), Huawei P20Pro (C_3), Xiaomi 10Pro (C_4). During the test data collection, the test screen is set to automatic mode (*i.e.*, the brightness and the display mode are changed independently according to the environment), and the test camera is also set to the automatic mode when capturing pictures (*i.e.*, the camera parameters, such as the exposure time, ISO, and the white balance, are adjusted automatically).

B. Real-world vs. Moiré Simulation

Here, we first evaluate the effectiveness of the synthetic dataset derived from the screen-imaging Moiré simulation. Each experimental screen-camera pair is fixed with tripods, and the screen sequentially displays the 1000 encrypted QR code images while the camera captures the corresponding Moiré images. We change the relative position and rotation of the camera and the screen five times and repeat the data collection steps described above. 800 QR code images and the corresponding captured Moiré QR code images are treated as a real-world dataset, and we use them to train the deep Moiré QR decryption model as $\mathcal{M}_r$. The remaining 200 QR code images and the corresponding captured Moiré QR code images are the test dataset as $\mathcal{D}_r$.

We compare the decryption performance of the two deep decryption models: $\mathcal{M}_s$ trained using the synthetic dataset, and $\mathcal{M}_r^i$ trained using the real-world dataset collected under the constrained relative poses of the i^{th} screen-camera pair. The real-world test dataset $\mathcal{D}_r^i$ of each screen-camera pair is used to evaluate the performance of the above two models, and the corresponding experimental results are shown in Fig. 8a. We find that $\mathcal{M}_s$ achieves a decryption rate almost equal to that of $\mathcal{M}_r^i$, that is, the model trained using synthetic data can reveal the original QR code well when decrypting the real-world Moiré QR code.



(a) When testing with the real-world dataset $\mathcal{D}_r^i$ collected in the limited screen-camera relative poses

(b) When testing with the real-world dataset $\mathcal{D}_s$ collected in the entire Moiré-visible area

Fig. 8: Decryption rates of two decryption models, one trained using a limited real-world dataset and the other trained using a synthetic dataset

Then, for each screen-camera pair, we use the test dataset $\mathcal{D}_s$ from Sec. IV-A2 to evaluate the above two decryption models $\mathcal{M}_r$ and $\mathcal{M}_s$. The corresponding experimental results shown in Fig. 8b confirm that the model $\mathcal{M}_r$ trained using the limited real-world dataset cannot perform well when it encounters the unknown Moiré QR codes that are not included in the training dataset. However, the model $\mathcal{M}_s$ trained using the synthetic dataset still performs well because the synthetic dataset generated from the Moiré simulation and data augmentation covers the entire Moiré-visible area and various screen and camera hardware. In summary, this is further evidence of the importance and effectiveness of the screen-imaging Moiré simulation.

C. Ablation Study

We evaluate the role of the different modules in the Moiré QR code simulator. The color filter array includes a variety of image sensor matrices, such as Bayer matrix (RGGB matrix), RGBW matrix and CYGM matrix. Considering that the image sensor matrix is entirely determined by the camera hardware, we do not analyze the effect of using different image sensor matrices in the simulator. Consequently, we evaluate the benefits of using different interpolation algorithms in the simulator and whether or not to use the data augmentation module.

1) interpolation algorithm: We utilize simulators with different interpolation algorithms(Bilinear Interpolation, Smooth-hue Interpolation [28], High-Quality Linear Filter [29], Gradient-Based Threshold-Free [30]) to generate synthesized datasets and feed them into the U-net based decryption network for training model M_{s_i}. Then we use the dataset $\mathcal{D}_s$ from Sec. IV-A2 to evaluate these decryption models $\mathcal{M}_{s_i}$. And the results shown in Fig. 13a suggest that these interpolation algorithms are well-adapted to our simulator and all provide a satisfactory decryption performance.

2) Data Augmentation: Then we evaluate the functionality of the data augmentation module. We utilize simulators with or without the data augmentation module to generate the synthesized datasets and obtain the trained models $\mathcal{M}_{s_j}$. Next, we use the same steps as above to test and get the results as shown in Fig. 13b. The results demonstrate that the data augmentation module is indeed an essential part of the simulator.

V. Evaluation of Deep Moiré QR Decryption

In this section, we comprehensively compare the decryption performance of the deep learning based decryption scheme proposed in this paper and the multi-frame decryption scheme proposed in [8] in the real-world data. We choose the iPhone 7Plus as the test screen and then alternately display 10 encrypted QR codes on it. Then the Huawei P20Pro is selected as the test camera and roughly fixed (by hand) at a certain position, *e.g.*, with the coarse grain of $1cm$ and 1°, and the Moiré QR code is recorded by video streaming. Every 20 frames of the Moiré images are input to a multi-frame decoding method to recover the original QR code information. One frame of the Moiré images is input to a deep Moiré method to produce a standard QR code, and the output of two consecutive frames is averaged as the final recovered QR code information. In the following parts, we compare the two decryption methods in terms of the decryption rate and the decryption latency.

A. Decryption Robustness in Moiré-visible Area

We first evaluate the influence of the offset distance against the P_{pm} (*i.e.*, perfect-match pose defined in Sec. III-A) on the decryption performance. During the experiment, the test camera is placed parallel to the screen, so that the angular deviations in the three axes are zero. We take the P_{pm} as the origin and the direction of the camera facing the screen as the z-axis to establish the coordinate system, then we change the offset distance (*i.e.*, T_x,T_y, and T_z) in three axes. The batch test is conducted every $1cm$ in the range $[-4cm, 4cm]$ of the z-axis. In each batch, 9^2 tests are conducted every $1cm$ in the range $[-4cm, 4cm]$ of $x-$axis and $y-$axis respectively. The decryption rate of the aforementioned two decryption methods is shown in Fig. 9. It is noteworthy that the closer to the P_{pm}, the higher the decryption rate. Due to the end-to-end training of the deep Moiré QR code decryption method, even if the offset distance exceeds $4cm$, the decryption rate reaches about 80%, which is significantly better than the multi-frame decryption.

Then, we evaluate the influence of the offset angle against the P_{pm} on the decryption performance. We fix the camera center at the P_{pm} and rotate R_x, R_y, R_z around θ_x-, θ_y- and θ_z-axis. For R_x, R_y, R_z, we perform a test every 2 degrees in the range of $[-5^{\circ}, 5^{\circ}]$ and conduct 7^3 tests totally. It can

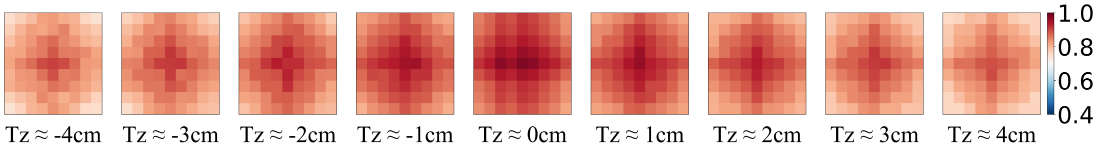


Fig. 9: Decryption heatmap of our proposed deep learning based decryption method. In each subfigure, the $z-$axis offset is fixed, and the offsets of $x-$ and $y-$axes are increased from $-4cm$ to $4cm$ with the center point being $0cm$

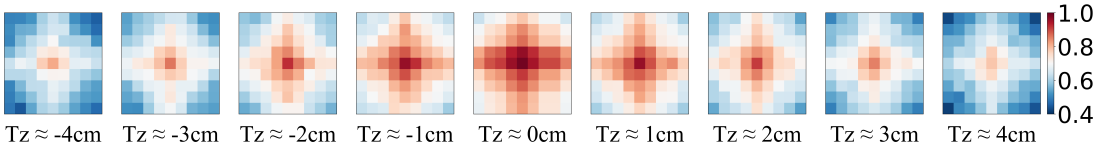


Fig. 10: Decryption heatmap of the multi-frame decryption method [8] where the offset distances on all three axes are in the range of $[-4cm, 4cm]$

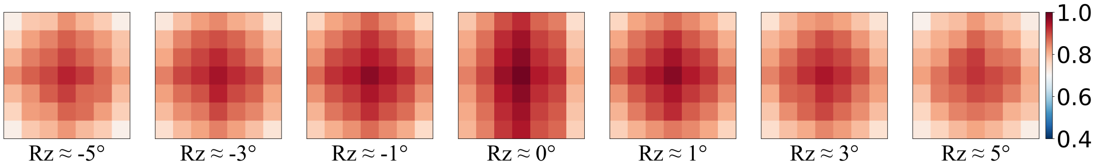


Fig. 11: Decryption heatmap of our proposed deep learning based decryption method. In each subfigure, the offset of θ_z-axis is fixed, and the offsets of θ_x- and θ_y-axes are increased from $-5°$ to $5°$ with the center point being $0°$

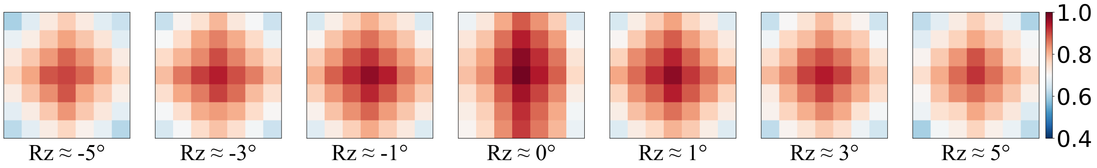
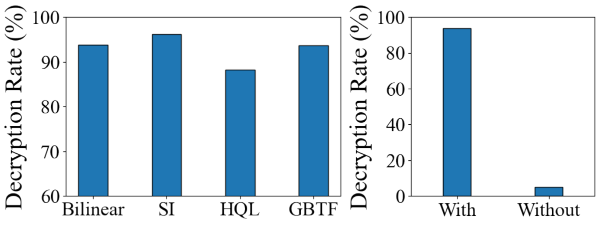


Fig. 12: Decryption heatmap of multi-frame decryption method [8] where the offset angles on all three axes are $[-5°, 5°]$



(a) simulators with different interpolation algorithms

(b) simulators with or without data augmentation

Fig. 13: Decryption rates of decryption models trained using synthetic datasets produced by different simulators

be seen that the decryption rate is highest at the P_{pm} and the deep Moiré QR decryption scheme has a significantly better performance than the multi-frame decryption method [8].

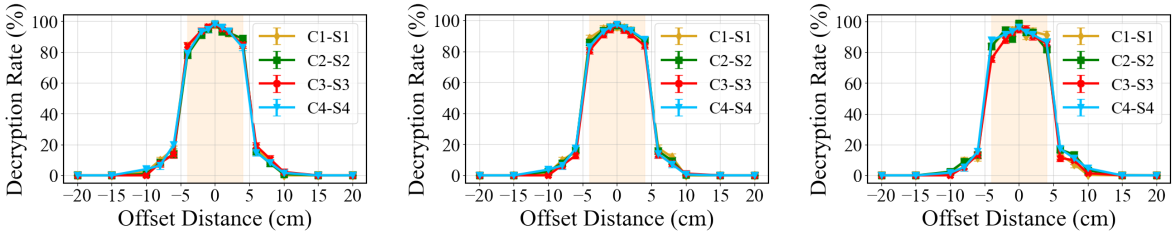
B. Secure Scanning Range

To determine the secure scanning range of the Moiré QR code system when using the deep Moiré decryption method, we performed experiments in 3D space and tested the decryption rate when the camera is offset from the screen by the perfect-match position point.

Considering that users will always turn their mobile device's cameras towards the screen in real-world scenarios, we change the position of the test cameras with different offset distances (*i.e.*, T_x, T_y, T_z) and keep them pointed towards the screens at all times. To obtain clearer visualization and better analysis of experimental results, we change the camera pose only in a single dimension and leave the remaining two dimensions unchanged. At each test position, we change the offset angle lightly (*i.e.*, R_x, R_y, and R_z follow a standard normal distribution $N(0, 1)$) and take the average of all the results in the position as the final result. As shown in Fig. 14, we note that the best decryption range (*i.e.*, the decryption rate is over 80%) in the $x-$ and $y-$ and $z-$axes are all in $[-4cm, 4cm]$ compared to the perfect-match point. And if the offset distance in a translation axis is above $10cm$, our proposed deep Moiré decryption method can't reveal the original QR code information. Therefore, the results demonstrate that the Moiré QR code system has high security in physical space. Even with an end-to-end supervised learning strategy, the decryption model has difficulty decoding the images captured outside the Moiré-visible area, which ensures secure communication of the QR code.

C. Impact of Devices

As shown in Fig. 14 and Fig. 8, we implement experiments with several screen-camera pairs. Although the different screen-camera pairs can make a difference in moiré QR code



(a) Offset distance T_x

(b) Offset distance T_y

(c) Offset distance T_z

Fig. 14: Decryption rate of the deep Moiré QR decryption for offset distances and angles on the $x-/y-/z-$axis compared to the perfect-match point

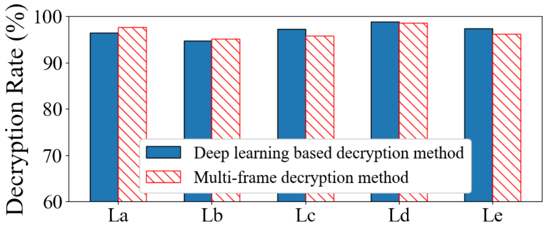


Fig. 15: Impact of lighting conditions on decryption methods. La: Outdoor at 8AM; Lb: Outdoor at 12AM; Lc: Outdoor at 11PM; Ld: Office; Le: Indoor with all lights off.

capture, all of these devices work in a similar manner and the results only have a negligible difference.

D. Impact of Environment/Ambient

We evaluate our proposed Moiré QR code decryption method under different ambient lighting conditions. Fig. 15 shows the decryption rate under different light conditions. We found that thanks to the adaptive adjustment of screen brightness, the impact of ambient environment illumination can be nearly ignored.

E. Overall Comparison

Finally, we test the consumption of the two decryption methods separately. A laptop equipped with an Intel i7-10875H 2.80GHz CPU with 32GB RAM is used for the entire offline decryption computation to test the memory overhead and decryption delay of two decryption methods. The entire decryption process is computed on the CPU. We denote the memory overhead as the average memory consumption of the entire decryption process. Moreover, we denote the decryption latency T_d as the sum of image acquisition time and Moiré code decryption time.

For the deep Moiré QR code decryption scheme, we assume that the model has been loaded in advance, and the model loading time is not included in the calculation of delay. T_d is calculated by the following formula: $T_d = \frac{N}{fps} + T_a \times N$, where N means the number of frames required by the decryption method, fps is the camera frame rate when recording videos, and T_a is the decryption time for a single frame.

TABLE I: Overall comparison of two decryption methods

	Multi-frame [8]	Deep learning based
Distance range	$[-2cm, 2cm]$	$[-4cm, 4cm]$
Angle range	$[-4°, 4°]$	$[-6°, 6°]$
Decryption rate	98.6% (11.3 frames)	98.8% (2 frames)
Decryption latency	5.4 ± 0.07s	0.02 ± 0.006s
RAM	27.4MB	224.2MB

The overall comparison is summarized in Tab. I. Based on the decryption rate results, further experiments show that the recommended input frames for the two decryption methods to achieve quite a high accuracy (close to 100%) are: 16 frames for the multi-frame decryption method and 3 frames for our proposed deep learning based decryption method. The most important is that the deep Moiré QR decryption can obtain negligible decryption latency ($0.02s$), while the multi-frame decryption method needs average 5.4 seconds to decode one encrypted QR code. Although the memory overhead of the deep Moiré QR decryption method is $224.2MB$, it is still acceptable for mainstream mobile devices. Therefore, we believe that our proposed deep Moiré QR decryption method can make the Moiré QR code system widely used in mobile application scenarios, which can improve the communication security for the standard QR code system with almost the same decryption latency.

VI. Conclusion

We present a novel decryption method based on deep learning that achieves a decryption latency of 0.02 seconds and a decryption rate of 98.8% for the Moiré QR code system. To generate a robust dataset for the training process, we propose a Moiré screen simulation method to synthesize Moiré QR code images. This simulator can generate images that cover the entire Moiré-visible area and physical camera-screen pairs. The extensive experiment is conducted to demonstrate the effectiveness of the proposed screen-imaging Moiré simulation and the excellent and robust performance of the deep learning based decryption model trained on the synthetic data. Deep Moiré QR decryption method can be easily deployed on mobile devices, and its outstanding decryption performance brings new vitality to the Moiré QR code system.

ACKNOWLEDGEMENT

This work is supported in part by NSFC (62072306, 61936015) and Program of Shanghai Academic Research Leader (20XD1402100).

References

[1] Harshith Keni, Montana Earle, and Manki Min. Product authentication using hash chains and printed qr codes. In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 319–324. IEEE, 2017.

[2] Vasileios Mavroeidis and Mathew Nicho. Quick response code secure: A cryptographically secure anti-phishing tool for qr code attacks. In *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*, pages 313–324. Springer, 2017.

[3] Siwon Sung, Joonghwan Lee, Jinmok Kim, Jongho Mun, Dongho Won, et al. Security analysis of mobile authentication using qr-codes. In *Computer Science & Information Technology-Computer Science Conference Proceedings*, 2015.

[4] Xiaolong Bai, Zhe Zhou, XiaoFeng Wang, Zhou Li, Xianghang Mi, Nan Zhang, Tongxin Li, Shi-Min Hu, and Kehuan Zhang. Picking up my tab: Understanding and mitigating synchronized token lifting and spending in mobile payment. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 593–608, 2017.

[5] Bridget O'Donnell. Steals money sneakily by scanning people's qr code, 2019.

[6] Katharina Krombholz, Peter Frühwirt, Peter Kieseberg, Ioannis Kapsalis, Markus Huber, and Edgar Weippl. Qr code security: A survey of attacks and challenges for usable security. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*, pages 79–90. Springer, 2014.

[7] Jun Wang and Wenqing Sun. A diffraction experiment with a peep-proof protection film of a cell phone using a spectrometer. *The Physics Teacher*, 57(4):268–268, 2019.

[8] Hao Pan, Yi-Chao Chen, Lanqing Yang, Guangtao Xue, Chuang-Wen You, and Xiaoyu Ji. mqrcode: Secure qr code using nonlinearity of spatial frequency in light. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–18, 2019.

[9] Hao Pan, Yi-Chao Chen, Guangtao Xue, Chuang-Wen Bing You, and Xiaoyu Ji. Secure qr code scheme using nonlinearity of spatial frequency. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pages 207–210, 2018.

[10] Li Junfei, Zhang Youqi, Wang Jianglong, Xiang Yang, Wu Zhipei, Ma Qinwei, and Ma Shaopeng. Formation mechanism and a universal period formula for the ccd moiré. *Optics Express*, 22(17):20914–20923, 2014.

[11] Itamar Friedman and Lihi Zelnik-Manor. Icon scanning: Towards next generation qr codes. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1137. IEEE, 2012.

[12] Di Xu, Colin Doutre, and Panos Nasiopoulos. Correction of clipped pixels in color images. *IEEE transactions on visualization and computer graphics*, 17(3):333–344, 2010.

[13] Ming Zhao, Wei Zhang, Zhile Wang, and Fugang Wang. Spatially adaptive image deblurring based on nonlocal means. In *2010 3rd International Congress on Image and Signal Processing*, volume 2, pages 853–858. IEEE, 2010.

[14] Manjit Kaur and Vijay Kumar. A comprehensive review on image encryption techniques. *Archives of Computational Methods in Engineering*, 27(1):15–43, 2020.

[15] Guy Lebanon and Alfred M Bruckstein. Variational approach to moiré pattern synthesis. *JOSA A*, 18(6):1371–1382, 2001.

[16] Roger David Hersch and Sylvain Chosson. Band moiré images. *ACM Transactions on Graphics (TOG)*, 23(3):239–247, 2004.

[17] Yvo Desmedt and Tri Van Le. Moiré cryptography. In *Proceedings of the 7th ACM conference on Computer and communications security*, pages 116–124, 2000.

[18] Pei-Hen Tsai and Yung-Yu Chuang. Target-driven moire pattern synthesis by phase modulation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1912–1919, 2013.

[19] Hao Pan, Yi-Chao Chen, and Guangtao Xue. Poster: Secure visible light communication via two-dimensional spatially aliased patterns. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, pages 51–52, 2019.

[20] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[22] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-domain correspondence learning for exemplar-based image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5143–5153, 2020.

[23] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5495–5505, 2021.

[24] Jianxin Lin, Yingce Xia, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Conditional image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5524–5532, 2018.

[25] Jie Liang, Hui Zeng, and Lei Zhang. High-resolution photorealistic image translation in real-time: A laplacian pyramid translation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9392–9400, 2021.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[27] Thomas Maschke. *Digitale kameratechnik: technik digitaler kameras in theorie und praxis*. Springer-Verlag, 2013.

[28] Rajeev Ramanath, Wesley E Snyder, Griff L Bilbro, and William A Sander. Demosaicking methods for bayer color arrays. *Journal of Electronic imaging*, 11(3):306–315, 2002.

[29] Henrique S Malvar, Li-wei He, and Ross Cutler. High-quality linear interpolation for demosaicing of bayer-patterned color images. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii–485. IEEE, 2004.

[30] Ibrahim Pekkucuksen and Yucel Altunbasak. Gradient based threshold free color filter array interpolation. In *2010 IEEE International Conference on Image Processing*, pages 137–140. IEEE, 2010.

[31] James E Adams Jr. Design of practical color filter array interpolation algorithms for digital cameras. In *Real-Time Imaging II*, volume 3028, pages 117–125. SPIE, 1997.

[32] Edward Chang, Shiufun Cheung, and Davis Y Pan. Color filter array recovery using a threshold-based variable number of gradients. In *Sensors, Cameras, and Applications for Digital Photography*, volume 3650, pages 36–43. International Society for Optics and Photonics, 1999.

[33] Chuan-kai Lin. Pixel grouping for color filter array demosaicing, 2003.

[34] Keigo Hirakawa and Thomas W Parks. Adaptive homogeneity-directed demosaicing algorithm. *Ieee transactions on image processing*, 14(3):360–369, 2005.