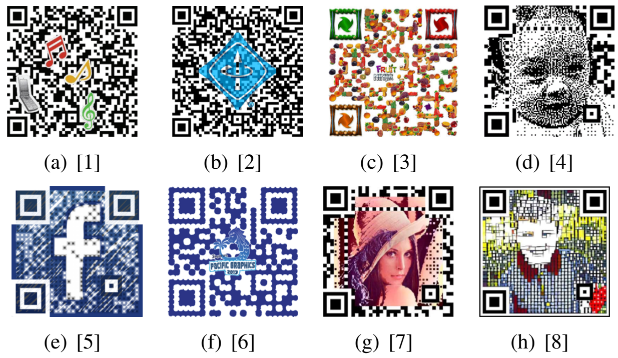
Visual-friendly Aesthetic QR Code Generation using Image Style Transfer

Tong Wu, Jiansheng Chen, Yiqing Huang, Yu Wang

***Abstract*—The quick response code, abbreviated as QR Code, is often presented as a two-dimensional matrix consisting of black and white modules that carry digital information. Visually, the vanilla QR Code is not friendly to human being since it is originally designed to be read by machines. Many efforts have been made in recent years to improve the visual appearance of the QR Code while maintaining its machine readability. Among all the proposals, the blending based methods aiming at achieving a balance between an aesthetic outlook and reading robustness have attracted most attentions. However, a common problem of existing blending based methods is that most dispersed data points are not successfully blended into the background, leading to unpleasant visual disturbances. Therefore, we propose a QR Code beautification algorithm that uses an intermedia images to help blending the QR Code and background image more naturally by reducing the scattered points while preserving the reading robustness at the same time. The intermedia image is produced using a modified neural style transfer network, which adopts a basic aesthetic QR Code as the content input and the background image as the style reference. We also introduce a self-defined mask allowing the user to select the region of interest in the background image of which the appearance will be preserved to the largest extent. Experimental results show that our method is able to stably produce QR Codes that are both machine readable and visual-friendly to human.**

***Index Terms*—QR Code beautification, Style Transfer, Image Fusion, Image Quality Assessment.**

I. Introduction


(a) [1] (b) [2] (c) [3] (d) [4]

(e) [5] (f) [6] (g) [7] (h) [8]

Fig. 1: Aesthetic QR Codes generated using previous methods.

With the increasing popularity of smartphones, the Quick Response (QR) code has been widely used in daily lives due to its convenience and robustness in delivering information. Vanilla QR Codes share similar and somewhat boring appearances since they mainly consist of seemingly irregularly arranged black and white modules. As shown in Fig. 1, efforts have been made to improve the aesthetics of QR Code for years. Early approaches [1], [2], [6], [9], [10] mainly focus on the insertion of small pieces of images, like logos or icons, into the QR Code. By utilizing the ability of QR Code to resist partial defects, these insertion based methods perform local modifications to the code while guaranteeing its readability. However, these modifications have to be small in size and are usually not sufficiently visually informative. To solve this problem, researchers began to study the approach to change the global appearance of the QR Code by blending the whole code into a background image [4], [5], [7], [8], [11]–[17]. These blending based methods beautify the QR Code by globally enriching the color and visual content. Besides the automatic methods mentioned above, aesthetic QR Codes can also be manually designed by revising the shape of modules and the overall style [3]. However, designers have to ensure the decodability by continuously doing tests with a QR reader throughout the whole designing process which can be very laboring and time-consuming.

A basic principle for any QR Code beautification method is to ensure that the original information encoded is not tampered and the generated aesthetic QR Code can be successfully scanned and correctly decoded by normal QR readers under most circumstances. In vanilla QR Codes, high contrast black and white square modules arranged in a lattice layout are used to achieve high reading robustness in real-life applications by reducing the impact of environmental factors such as illumination, camera angles and imaging noises during capturing. However, in QR Code beautification, the color, shape and layout of some of the modules have to be changed to accommodate the texture of the image to be inserted or blended. These modifications will inevitably reduce the reading robustness of the QR Code and therefore need to be carefully controlled. In the insertion based methods, such control is usually realized by straightly limiting the size and location of the inset. As such, most part of the original QR Code has to be kept unchanged, leading to relatively insufficient beautification effects. In the blending based methods, such control is achieved by constraining the relative brightness of different pixels in the background image. As such, more flexibility can be provided when making global changes to the QR code, leading to the increasing popularity of the blending based method in the research community. In this paper, we focus on the blending based automatic QR Code beautification.

The main challenge of the blending based QR beautifi-

The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing, China. E-mails: (wutong16@mails.tsinghua.edu.cn, jschenthu@mail.tsinghua.edu.cn, huang-yq17@mails.tsinghua.edu.cn, yu-wang@mail.tsinghua.edu.cn).

Manuscript received xxxx; revised xxxx.

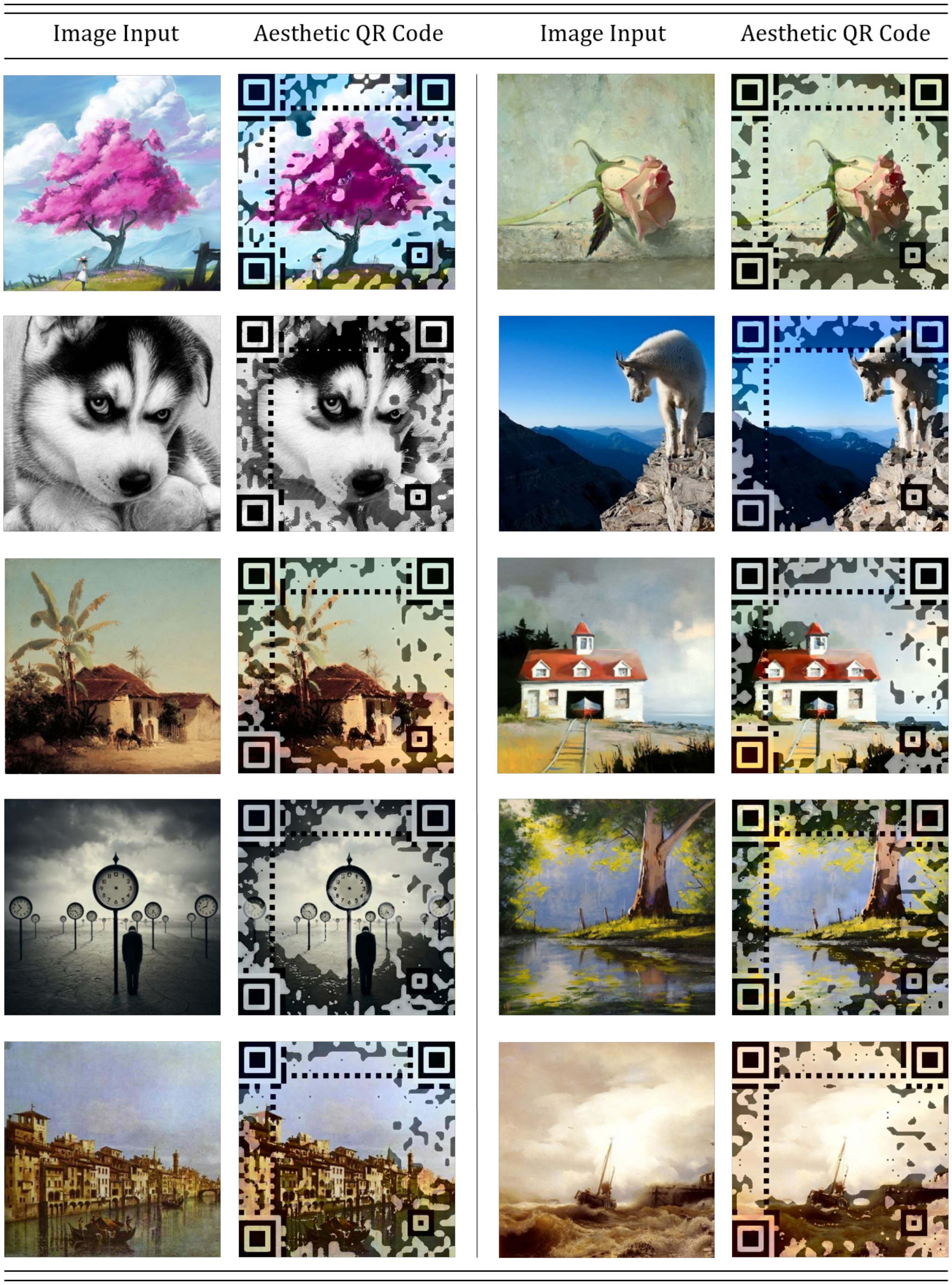


Fig. 2: Examples of aesthetic QR Codes generated using the proposed method. Zoom in to reduce the impact of the screen resolution on the reading robustness.

cation is to bridge the huge texture gap between a vanilla QR Code and the background image. The QR Code was initially designed to be suitable for information encoding and decoding by machines. The binarized block modules in the QR Code image are machine-friendly but not human-friendly. Background images, whether natural or artificial, are visual friendly to human beings. However, they cannot directly serve as the carrier of encoded information for QR decoders. An intuitive way of blending these two intrinsically different kinds of images while ensuring the decodability of the result is to modify the background image so that the relative intensity of certain pixel pairs comply with the QR Code image. However, this will introduce visually unpleasant block patterns to the background image, especially in the areas where the pixel intensity distributions of two images are highly inconsistent. To deal with this problem, R. Cox utilized the non-uniqueness of the QR Code and rearranged the modules of the QR Code to make it as close to the binarized background image as possible [17]. However, areas of inconsistency still exist in most cases and the generated results still contain a significant amount of scattered blocks which may cause visual discomfort.

Since the disturbing block patterns cannot be completely avoided, we argue that aesthetics of the beautified QR Code can be improved by reshaping and recoloring these blocks so that they can be blended into the background image more naturally. We therefore propose to transfer the style of the background image to the vanilla QR Code so that the shape and color of the modules are changed to resemble that of the objects in the background image. As such, the styles of different areas in the generated aesthetic QR Code become more visually consistent, leading to higher visual satisfaction. We also introduce a preference mask to allow the user to define the region of interest in the background image, which will remain unchanged to the largest extent during blending for realizing personalized aesthetic QR Code. An iterative fine-tuning process is also proposed to automatically adjust the parameters controlling the reading robustness of the final output. Fig. 2 shows examples of the proposed visual friendly aesthetic QR Code with different kinds of background images. They can be successfully decoded by normal QR readers directly from the screen.

Evaluating the decodability and reading robustness of the beautified QR Code can be performed by empirical experiments. However, evaluating the aesthetics of the QR beautification result is not easy considering that aesthetic tastes are intrinsically subjective. A possible metric is the similarity between the background image and the beautified QR Code. Nevertheless, high image similarity does not necessarily indicate high visual satisfaction. In this work, we seek for a natural overall look of the blending result by suppressing visual disturbances induced by the scattered points or blocks. Based on the results of user study, we find that higher intensity continuity and style consistency in the blending results are in general making the aesthetic QR Code more visual-friendly.

II. Related Work

1) QR Code Beautification: Inserting small-sized images like logos or decorations into the original QR Code was a major approach in early works. These methods mainly take advantage of the error-correction capability of the Reed-Solomon (RS) code, which means that readers do not have to capture every single module in a QR Code correctly to successfully decode it. S. Ono et al. [1] proposed a system that formulated the task of finding the most appropriate positions, scales, and angles for elements to be inset as an optimization problem. Lin et al. [6] proposed an error-aware warping technique to deform the elements inset, and they also reshaped the regular shape of the modules into binary exemplars. These methods, however, inevitably introduce bit errors, because the original codes are partly replaced by insertions directly. To avoid such bit errors, Baharav et al. [2] proposed a method that leaves completely intact the error correction capacity of the code by changing the transparency of the images inserted. Nevertheless, this kind of compromise affects the appearance of the inserted images quite obviously.

A common problem shared by the insertion based methods is that only a very small part of the QR Code can be modified. Therefore, the global visual appearance of the generated QR Code remains similar to a vanilla one. To improve the visual significance of the image to be added and to make a global change to the appearance of the QR Code, blending based methods have been introduced in recent years. Following the idea of halftoning, Chu et al. [4] proposed to subdivide each module into 3×3 sub-modules and bound the module's intensity to the center sub-module. As such, the gray-scale values of all the other eight sub-modules can be changed freely according to the binarized background image to be blended. However, since the sub-module is the smallest unit, the resolution of the generated QR Code is very low. R. Cox utilized the error correction properties of the RS Code and proposed an efficient QR Code beautification algorithm called Q^{Art} based on Gauss-Jordan elimination [17]. QR Codes generated by Q^{Art} share a roughly similar appearance with the binarized background image. Q^{Art} is now adopted by most of the blending based methods as the first step before further processing. Lin et al. [7] achieved a relatively high visual quality in the central area of the image with a specific rendering strategy. However, the edges of the generated QR Code remain the same as the original QR Code, leading to visual dissatisfaction. Xu et al. [8] proposed to apply the technique of neural style transfer to QR beautification. The process of style transfer is executed after the generation of the baseline aesthetic QR Code to provide users with more choices of personality.

In most previous blending based methods, part of squared modules or blocks still scatter in the generated QR Codes, making them less visually friendly. Inspired by Xu's work [8], we argue that these visually annoying squared modules or blocks can actually be better fused into the background image by applying the style transfer process during the initial blending stage.

2) Neural Style Transfer: With the prevailing of deep neural networks, the task of image style transfer has gone through extensive investigations during the past several years. Gatys et al. [18] proposed a neural style transfer method leveraging pre-trained deep convolutional networks such as VGG-19 [19]. Their main idea is to decompose and recombine the content and style from two images by matching the Gram matrix statistics of deep features using the method of optimization. Johnson et al. [20] utilized perceptual loss functions to train feed-forward networks for real-time texture transfer tasks in place of the original time-consuming iterative optimization. Recent works have included semantic segmentation as an optional guidance [21], [22] to partly avoid the edge-distortions during style transfer. More recently, Liao et al. [23] proposed the Deep Image Analogy for visual attribute transfer. They assume that the input image pairs have similar semantic structures. Thus, by combining the technology of dense image correspondence in deep feature space and neural style transfer, a prominent visual effect at the pixel level can be achieved.

III. Proposed Method

A vanilla QR Code is composed of black and white square modules that contain bit information of the embedded message. Given a background image I, the beautification of a

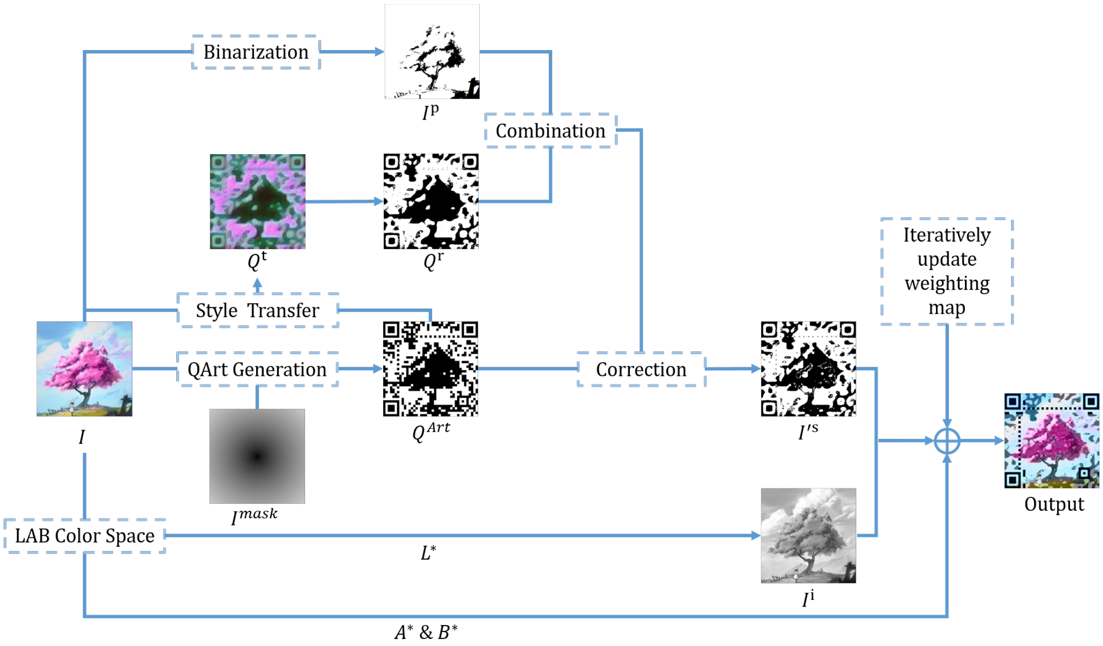


Fig. 3: An overview of the proposed method.

QR code can be generally formulated as an optimization problem, which aims to minimize the visual contrast between I and the noise-like black/white modules while maintaining an acceptable decoding rate. Our proposed QR beautification process mainly consists of the following stages. Firstly, a baseline QR Code with robust decodability but relatively poor visual quality, named Q^{Art}, is generated using the Gauss-Jordan elimination procedure proposed in [17]. Secondly, a style transfer network is applied on Q^{Art} to generate a stylized QR, named Q^t. Finally, Q^t and I are synthesized into a non-robust artistic QR Code through a specific rendering procedure. The parameters used in the rendering procedure are then iteratively updated to generate the final robust and visual friendly aesthetic QR Code Q^{VF}. An overview of our proposed method is shown in Fig. 3.

A. Q^{Art} Code Generation

In this stage, we adopt the method proposed by R. Cox [17] to generate a decodable binary QR Code Q^{Art}. Ideally, the output of this stage should have a similar outline as the binary image of I, named I^p. A QR Code contains a Reed-Solomon (RS) error-correction code and it has been proved in [17] that by utilizing certain properties of the RS code, the binary patterns of the square modules can be modified through a series of Gauss-Jordan elimination operations without hurting its error-correction capacity. Such a mechanism indicates that one can take control over the binary patterns of some modules so as to force them into forming a desiring shape. Nevertheless, the number of controllable modules is limited.

The selection strategy of the controllable modules varies in different methods. For example, it can be performed by setting a priority for each module in the QR Code. Previous works decide the priorities by focusing on different factors of the image, such as local contrast [17], spacial structure [7] and global feature of the image [8]. In this work, we simplify the selection strategy by allowing the users to provide a user-defined mask I^{mask} to roughly depict their preference for different regions of the image. I^{mask} is a gray-scale image with the same size as I. Its pixel values are regularized to $[0, 1]$, where a value closer to 0 indicates a higher preference by the user, or a higher selection priority. If a binary mask image is provided, we blur it to generate continuously changing pixel values to ensure more flexibility. The average pixel values of I^{mask} are calculated inside each module and then sorted to form a module-level priority order. The module is then sequentially changed based on the rule proposed in [17] to produce Q^{Art}. When no user preference mask is provided, our method adopts a Gaussian kernel shaped mask by default, as is shown in Fig. 3.

B. Image Style Transfer

It can be observed from Fig. 3 that the object in the center part of the baseline artistic QR Code Q^{Art} has an appearance similar to that in I^p due to the priority map defined in the default I^{mask}. However, because of the limitation on the number of controllable modules, the visual quality of Q^{Art} is still unsatisfactory in regions with low priority. Therefore, in this stage, we propose to apply neural style transfer to Q^{Art} to further enhance its global appearance.

We adopt the style transfer method proposed in [18] with a slight modification to the original loss function. The key idea behind this method is to iteratively optimize the input image to match a desired CNN feature distributions. Specifically, a pre-trained VGG-16 [19] model is used as the backbone network to extract the features from an *input image*, a *content target image* and a *style target image*. Then a feature reconstruction loss and a style reconstruction loss are calculated utilizing the image features. Through jointly minimizing the losses, the input image I^x is iteratively updated through backpropagation, and finally, it will gain both the content of the *content target image* and the style of the *style target image*.

In this work, the Q^{Art} is used as both the *input image* and the *content target image*. The background image I is used as the *style target image*. As the target images are fixed while the input image will be gradually changed, we rename Q^{Art} as I^* when it is used as the *input image*.

Let $F^l_{m,n}(x)$ be the feature representation of an image x after layer l. The subscripts m and n represent the activation of the m^{th} filter at position n. Then the feature reconstruction loss L^c of layer l can be formulated as Eq. 1.

$$L^c(Q^{Art}, I^*, l) = \frac{1}{2}\sum_{m,n}(F^l_{m,n}(I^*) - F^l_{m,n}(Q^{Art}))^2 \quad (1)$$

In [18], the image style is represented by a Gram Matrix, which is defined as the inner product of vectorized feature maps. Let $G^l_{m,k}(x) = \sum_n F^l_{m,n}F^l_{k,n}$ which represents the Gram Matrix between the m^{th} and k^{th} feature map of x for layer l, then the style reconstruction loss L^s of layer l can be formulated as Eq. 2, in which N_l and M_l represent the lengths of the vectorized feature maps.

$$L^s(I, I^*, l) = \frac{1}{4N_l^2M_l^2}\sum_{m,k}(G^l_{m,k}(I^*) - G^l_{m,k}(I))^2 \quad (2)$$

To enhance the readability of the style transferred image Q^t as well as to accelerate the procedure, we add a new term called accuracy reconstruction loss, or L^a, along with an accuracy target image I^a. It has been proved in [10] that when scanning a QR Code, a decoder generally focuses on the pixels inside the central area of each module. So on one hand, in order to achieve a high reading robustness, pixel values of a stylized QR inside the central area of each module should be as close as possible to the binary values of the same position. On the other hand, the region outside the central part can as similar to the background image as possible for achieving high visual satisfaction. In order to make the output of the style transfer process to meet these goals, a pixel-wise supervision is introduced based on I^a. To contrive I^a, we roughly adopt a circle with a radius of 3 pixels as the central area boundary. Thus I^a is synthesized by taking I and modifying the pixel values in central area of each module to the binary values of the same position of Q^{Art}, as shown in Fig. 4. As such, L^a can be defined as the mean squared error (MSE) between the input image I^* and the accuracy target image I^a as in Eq. 3, in which subscripts x and y represent the pixel positions. It should be noticed that L^a should be calculated ahead of all the layers.

$$L^a(I^a, I^*) = \frac{1}{2}\sum_{x,y}(I^*_{x,y} - I^a_{x,y})^2 \quad (3)$$

For simplicity, in calculating the content reconstruction loss, we only use the feature map output from the layer $conv4\_1$, namely l_c; and in calculating the style reconstruction loss, we use the output from the following convolution layers: $conv1\_1$, $conv2\_1$, $conv3\_1$, $conv4\_1$ and $conv5\_1$, namely l_s. As such, the overall loss function is defined in Eq. 4, in which ω_l are weighting coefficients.

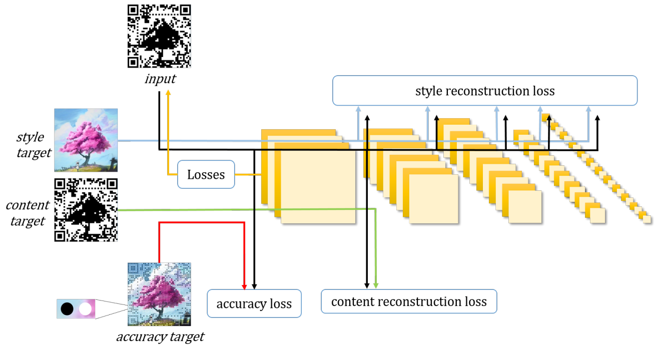


Fig. 4: Framework of the style-transfer network. The accuracy target is a combination of the Q^{Art} code and the original background image. the Accuracy-loss is directly calculated while the other losses are extracted after selected convolution layers.

$$\begin{aligned} L(Q^{Art}, I, I^a, I^*) &= w_c L^c(Q^{Art}, I^*, l_c) \\ &+ w_s \sum_{l \in l_s} \omega_l L^s(I, I^*, l) + w_a L^a(I^a, I^*) \end{aligned} \quad (4)$$

The overall architectures of the style transfer network is illustrated in Fig. 4. Visually, the resultant stylized QR Code Q^t keeps a similar structure of the original QR and is stylized with the texture of I. The binary image of the stylized QR is named Q^r in Fig. 3.

It should be emphasized that unlike a typical output of a neural style transfer network, Q^t does not has a clear visual content and a distinguishable artistic style. This is due to the fact that the content target image Q^{Art} itself is a visually meaningless machine code. In fact, Q^t should be seen as a transition image from the machine code to a natural image.

C. The Rendering Procedure

The rendering procedure leverages the resultant images from the previous stages to synthesize the final output Q^{VF}. In this stage, most of the operations are applied to the L channel of the LAB color space. The main goal of this stage is to generate a gray-scale image that is both as decodable as Q^{Art} and as visual friendly as I^i, which is the gray-scale image of I. Finally, a colored Q^{VF} image will be generated simply by channel merging. The inputs to this stage include the binarized background image I^p, the binarized stylized QR Code Q^r, the baseline QR Code Q^{Art} and the gray-scale image of the background I^i. A two-step process is designed to accomplish the desired rendering procedure. In the first step, a basic rendering process is performed through the weighted sum of images in the L channel of the LAB color space. In the second step, an iterative updating procedure is used to adjust the weight maps used in the last step. Finally, simply by merging the color channels, we can generate a colored Q^{VF}.

1) Basic Rendering Step: In this stage, the two binary images I^p and Q^r are blended to generate an intermediate result I'^s. For better understanding, we adopt i and j as subscripts for the operations at module level, and x and y for operations at pixel level. Strictly speaking, all the operations are applied to each pixel in the images. By module level, we actually mean that the operations inside the same module are identical.

The blending operation is performed by calculating a weighted sum of I^p and Q^r as is shown in Eq. 5, in which β is a non-negative scalar.

$$I^s_{x,y} = \beta \times W^s_{x,y} \times Q^r_{x,y} + (1 - \beta \times W^s_{x,y}) \times I^p_{x,y} \tag{5}$$

To decide the weighting coefficients W^s, two major factors are taken into account. Firstly, the resultant image I^s needs to keep a module-level consistency with Q^{Art} to ensure decodability. Since Q^r shares a similar structure with Q^{Art}, we measure the difference between I^p and Q^{Art} by the absolute difference of their module-wise average pixel values, $W^n = |\overline{I^p}_{i,j} - Q^{Art}_{i,j}|$. Secondly, a low distortion w.r.t. I^p is highly desired, especially in the area with higher priority defined by I^{mask}. So by leveraging W^n and I^{mask}, the definition for W^s can be formulated as Eq. 6.

$$W^s_{x,y} = \gamma \times I^{mask}_{x,y} + (1 - \gamma) \times W^n_{x,y} \tag{6}$$

As we have mentioned above, when scanning a QR Code, the decoder generally focuses on the pixels inside the central area of each module. The specific size of the crucial central part of practice is influenced by the application condition and algorithms used by different decoders. Therefore, we adopt a 3×3 block out of the 9×9 module as the central area. However, the central consistency may be violated when applying Eq. 5. This means that average pixel values in the central area are not consistent with that in the original Q^{Art}. To solve this problem, we further take a correction step on I^s to enhance the central consistency. For each module, pixels in the central area are reset to the binary values as in the same position of Q^{Art}, and the corrected image is renamed as I'^s.

2) Iterative Updating Rendering Step: The blended result I'^s generated in the above step is already an aesthetic QR Code. Compared with the baseline Q^{Art}, I'^s is visually more similar to the background image owing to the image style transfer operation. In this stage, the gray-scale background image I^i is utilized to further improve the visual friendliness by restoring more details.

Specifically, we'll calculate another weighted sum leveraging I'^s and I^i to get the final gray-scale output I^r, as shown in Eq. 7. The weighting coefficient W^r is initialized as all-one matrix, and will be adjusted in an iterative manner.

$$I^r_{x,y} = I^{mask} \times W^r_{x,y} \times I'^s_{x,y} + (1 - I^{mask} \times W^r_{x,y}) \times I^i_{x,y} \tag{7}$$

The gray-scale image I^r is the L channel representation, by stacking it with the A and B channels extracted from I, the colored aesthetic QR Code Q^{VF} can be achieved. However, what's worth noting is that the correction process is applied before the weighted sum operation with I^i. The specific order allows a more natural gray-scale resultant image with very few 0 and 1 valued pixels. However, it may also harm the central consistency to some extent. One possible solution to this problem is to adaptively change W^r in Eq. 7 through an iterative updating process.

In order to locate the positions where inconsistency occurs, we simulate the thresholding process inside a QR decoder and apply it to I^r. The thresholding result Q^{read} is achieved according to Eq. 8, in which t_h and t_l are two preset thresholds. Notation $center$ means that the averaging is performed in the central area for each module.

$$Q^{read}_{i,j} = \begin{cases} 1, & \overline{I^r}_{center,i,j} > t_h \\ 0, & \overline{I^r}_{center,i,j} < t_l \\ 0.5, & else \end{cases} \tag{8}$$

Thresholds t_h and t_l in Eq. 8 are manually set and are usually adjusted to adapt to different environmental conditions. Their values significantly affect the reading robustness, which will be further discussed in the experiments. We have also tried other strategies like local thresholding proposed in [12], where the threshold is defined as the average luminance in a sliding window of 5×5 modules. However, this policy does not show obvious superiority in our case. In fact, the thresholding strategy varies for different QR decoders [24], [25]. The reading robustness relies heavily on the light condition of the environment. Using two fixed valued threshold is a generally applicable stratagem.

The inconsistency between Q^{Art} and Q^{read} can be calculated by the absolute difference of their module-wise average pixel value, $W^e_{i,j} = |\overline{Q^{read}}_{i,j} - Q^{Art}_{i,j}|$. An error score, S^e, which is a real number, is represented as the global sum of pixel values of the W^e as in Eq. 9.

$$S^e = \overline{W^e_{i,j}} \tag{9}$$

Apart from the accuracy score, a visual score is also calculated to measure the difference between the output and background in Eq. 10, where I^{mask} is used to depict regionally different importance.

$$S^v = \overline{(|Q^r_{x,y} - I^i_{x,y}| \times (1 - I^{mask}_{x,y}))} \tag{10}$$

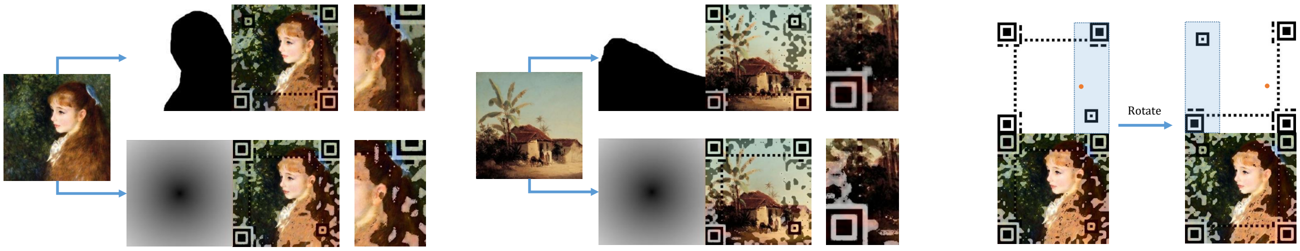
$$W^r = W^r + \delta_1 \times S^e \times W^e - \delta_2 \times S^v \times (1 - W^e) \tag{11}$$

By utilizing S^e and W^e, weighting coefficients W^r can be iteratively updated using Algorithm 1, in which a single step is defined in Eq. 11. The iteration ends until S^e is lower than an pre-defined error tolerance, of which the value is closely related to the error correction level of the QR Code.

Algorithm 1 The iterations of the fine tuning step

```
Input: $W^r$;
Output: final $W^r$ and $I^r$;
  Initialize Error Map and Losses;
  while $S^e > errortolerance$ do ;
    Update $W_e$;
    Calculate $S^e$ and $S^v$;
    Update $W^r$;
    Update $I^r$;
  end while
```

The above procedure realizes local adaptation of W^r so as to fix the problem of inconsistency. Finally, W^r can be used



(a) Examples of preference mask application

(b) An example of Q^{Art} rotation

Fig. 5: (a) Two examples of the application of preference mask. Once I^{mask} is provided, the algorithm assures a lower pollution to areas with higher priority. (b) How the rotation strategy works. The orange point indicates the center of the mask and the blue area is where the message bits are placed.

to rebuild I^r and Q^{VF}, an aesthetic QR Code with both high reading robustness and appealing visual appearance.

IV. Experimental Results

A. Implementation Details

In the experiments, the default values of the parameters are set as follows. The number of modules contained in a general QR Code as well as Q^{Art} is decided by its *version*, V: for each row and column, there are $N_m = 4 \times V + 17$ modules. We adopted $V = 6$ in all the experiments for it's a common practice for ordinary QR Codes. Considering that each module contains 9×9 pixels, the length of side of Q^{Art} is $N_m \times 9 = 369$. As a results, all the background images are resized to 369×369 before further processing. We generate QR Codes with the URL *https://www.bing.com/* for most of our experiments except for those in Sec. IV-G.

Experiments involving decoding success rate completed using a 14-inch screen with a resolution of 1920×1080, and the physical size of the testing QR images are resized to around 3.5×3.5 inches (≈ 9 cm) on the screen.

For the style transferring stage, we use $w_c = 1, w_s = 1e6, w_a = 200$ for the three losses; and in the basic rendering step, we fix $\beta = 1.4$ and $\gamma = 0.6$ to achieve a balance between visual appearance and robustness. In the iterative updating rendering step, we set the two threshold parameters in Eq. 8 to be $t_h = 170$ and $t_l = 90$. And the step sizes in Eq. 11 are set to be $\delta_1 = 0.015$ and $\delta_2 = 3e^{-7}$, with the error tolerance set as 5, considering the iteration speed and output resolution.

B. User-defined Preference Mask

As mentioned above, we incorporate a user-defined mask I^{mask} to indicate the user's preferences for different regions in the background image. Regions with high preference are expected to have a higher priority to remain unpolluted and visually pleasant. By default, our method uses a center-emphasized mask considering that for most cases the central part is the most salient region of a natural image. Nevertheless, users are also allowed to customize the mask. Fig. 5(a) shows the generated aesthetic QR Codes when different masks are used. It can be clearly observed that user preferred regions can be well protected through deliberately customizing the shape of the mask.

In the Q^{Art} generation method we adopt, the modules where message bits are placed are not changeable, which means that Q^{Art} can't resemble I^p well in that area. An example of this unchangeable message area is illustrated in light blue color in Fig. 5(b). When this area heavily overlaps with the user-defined mask, the generation result may be visually unsatisfactory. For example, the hair area in the lower left image of Fig. 5(b) is polluted by many isolated QR modules. We utilize the QR rotation to solve this problem. Specifically, given a user-defined mask, we first calculate its center of gravity. And then we rotate the QR Code so that the message area is placed away from the ROI center, as is visualized in the right side of Fig. 5(b).

C. Decoding Robustness

When decoding the generated Q^{VF}, the success rate may be affected by the values of the two thresholds t_h and t_l defined in Eq. 8. The reading robustness of Q^{VF} towards high dynamic illumination variations can be improved when decreasing t_l and increasing t_h simultaneously. In the extreme case when $t_l \to 0$ and $t_h \to 255$, empirical experimental results show that the reading success rate can be nearly 100% for a large number of randomly selected background images. However, too small t_l or too large t_h will hurt the visual friendliness of the generated Q^{VF}, as will be discussed later. Actually, such an extreme thresholding scheme may not be necessary for practical uses. With a specific background image under a certain environmental condition, users may manually adjust t_h and t_l to achieve a balance between the reading robustness and visual friendliness. Furthermore, the decoding success rate may vary with the device, the decoding software, or the scanning condition. We therefore empirically fix t_l and t_h to 90 and 170, which is suitable for most daily use conditions. In the following subsections, we first empirically test the reading robustness of Q^{VF} with different devices, software, and scanning angles, then effects of the threshold values on the visual appearance of Q^{VF} are demonstrated.

We build a background image dataset by selecting a hundred images of different kinds from the Internet. These images cover common subjects and contents including landscape, portrait, oil painting and cartoon. As for the information to be encoded, we use the URL 'https://www.bing.com/' in most

Mobile Phone	App	Success Rate
Huawei Mates	Wechat	96%
	QR reader	96%
	Alipay	96%
	QR Droid	96%
Huawei Honor8	Wechat	92%
	QR reader	92%
	Alipay	92%
	QR Droid	90%
Iphone 6s	Wechat	96%
	QR reader	96%
	Alipay	96%
	QR Droid	96%
Meizu M15	Wechat	94%
	QR reader	94%
	Alipay	94%
	QR Droid	94%

TABLE I: Decoding success rate using different devices and software

of the following experiments, except for the experiment in Sec. IV-G in which we focus on how the length of encoded information affect the performance. Over 95% of the images in the background image dataset are successfully used to generate the Q^{VF}; and in Sec. IV-H we'll give a brief discussion on how fail cases may occur and how they can be partly solved.

1) Free form QR decoding: We select 100 generated Q^{VF} as the test set for the free form QR decoding experiments, which are all performed based on screen display. The definition for a successful free form QR reading try is that, the volunteer doing the test can move the mobile device freely in order to find a proper distance and angle to simulate the real-life case. If the reader gets the correct information within a time limitation of 3 seconds, it's counted as a successful attempt. The statistic results for different devices and software are shown in Table I.

2) The effect of scanning angles: We define the scanning angle as the angle between the normal direction of the QR Code plane and the optical axis of the camera. In this experiment, we fix the QR Code plane and rotate the camera to change the scanning angle. The change of decoding success rate with different scanning angles is shown in Fig. 6. It can be observed that the scanning angle has a prominent influence on the decoding success rate when its value is larger than 30 degrees.

3) Visual impacts of the threshold values: Thresholds t_l and t_h in Eq. 8 simulate the thresholding policy inside a normal QR decoder. In terms of the visual effect of the generated aesthetic QR Code, the t_l usually has a greater effect on the darker part of the output image, while t_h mainly affects the brighter part, as shown in Fig. 7. Together with the error tolerance mentioned in Algorithm **??**, the two thresholds also decide the reading robustness of the final Q^{VF}. If the t_l and t_h are set close enough to 0 and 255 respectively, the resultant Q^{VF} will be enforced to a binary image, of which high reading robustness is achieved by seriously sacrificing the visual friendliness.

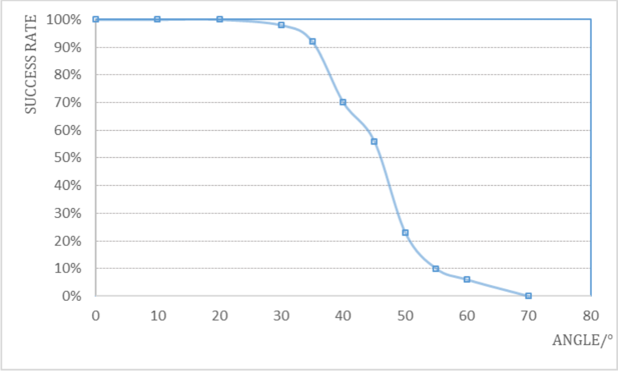


Fig. 6: Success rate effected by scanning angle

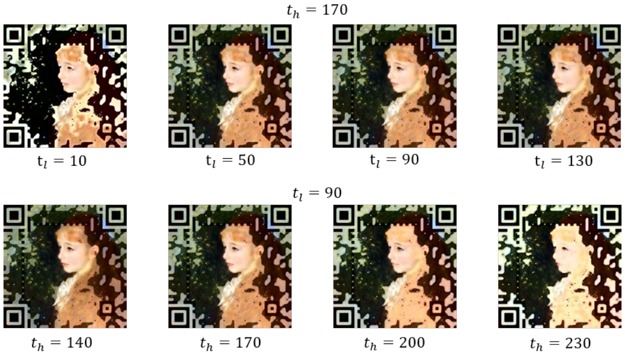


Fig. 7: Visual effects when using different thresholds.

D. Rendering Parameters β and γ

As described above, several weighting maps are leveraged to calculate the weighted sum of different intermediate resultant images in the rendering stage. The major task of this stage is to achieve a balance between two intrinsically contradictory targets, namely maintaining the reading robustness and protecting the visual appearances. This is actually achieved by selecting proper values for the two weighting parameters, namely β in Eq. 5 and γ in Eq. 6.

1) The effect of β: Parameter β decides the weighting bias of Q^r when combining it with I^p to generate I^s. The larger the value of β is, the more I^s will assemble Q^r as well as Q^{Art}, leading to higher reading robustness. Sample generated QR Codes using different β values are shown in Fig. 9. It can be observed that as the value of β increases, more scattered points are hidden into the blended image. This is because that the scattered points are introduced through the correction procedure based on Q^{Art}, and once I^s itself matches the module distribution of Q^{Art} well, fewer scattered points will be introduced. When $\beta = 0$, all the scattered points are exposed as shown in the upper left image in Fig. 8. Although the whole image looks well, it's reading robustness is relatively low. Actually, in order to be correctly decoded, the scattered points in the image need to be bigger and more conspicuous, which will in turn hurt the visual appearance. We also test the decoding success rate with different β values and the result is shown in Fig. 9.

2) The effect γ: On calculating the weight map W_s in Eq. 6, two images I^{mask} and W_n are used. I^{mask} emphasizes that in regions of interest, W_s should be endowed with higher values to improve visual friendliness. And W_n emphasizes that in modules where the average brightness of I^p and Q^{Art} differs a lot, W_s should be endowed with a lower weight to avoid the risk of losing accuracy. Parameter γ, of which the value ranges from 0 to 1, is used to balance between them. Experimental results on how the visual appearances of the generated results are influenced by γ are shown in Fig. 10. Globally, the visual appearance does not change much. However, some scattered points are enlarged into small pieces of continuous areas when γ increases, leading to enhancements in the decoding robustness.

E. Image Adjustment

Adjusting the image may affect the quality of the final generation results. We study the adjustments on the image contrast. As proved in [7], conducting contrast enhancement in the LAB color space usually results in better visual quality than in the RGB color space for most of the low-contrast images. The contrast adjustment is performed on the L channel in the LAB color space as shown in Equation 12, in which notations i, o and ave mean input, output and average value respectively.

$$I^o_{x,y} = I^{ave} + a_l \times (I^i_{x,y} - I^{ave}) \tag{12}$$

We find that after contrast adjustment, besides the obvious changes on the appearance, as shown in Fig. 11, the decoding robustness of the outcome is also influenced. A higher contrast generally results in a better decoding ability. This is probably due to the local thresholding strategy [12] adopted by some of the QR readers like the Zxing library [24].

F. The Shape of Points

In the correction procedure in the basic rendering step, scattered points are introduced. The selection of the point shape also affects the overall visual quality of the generation result. In previous work [6], a variety of shapes has been tried. To pursue a minimum visual disturbance and maximum readability, we mainly follow the center principle proposed in [10] and adopt the center $1/3$ square area of the module as the key area. We have also tried other choices like the Gaussian and the stellate brightness distribution for each module. Definitions of these different kinds of point shapes are shown in Eq. 13 to 15 . Some of the generated QR Codes using different point shapes are shown in Fig. 12. Experiments show that the square point generally achieves the highest reading robustness and the most satisfying visual appearances.

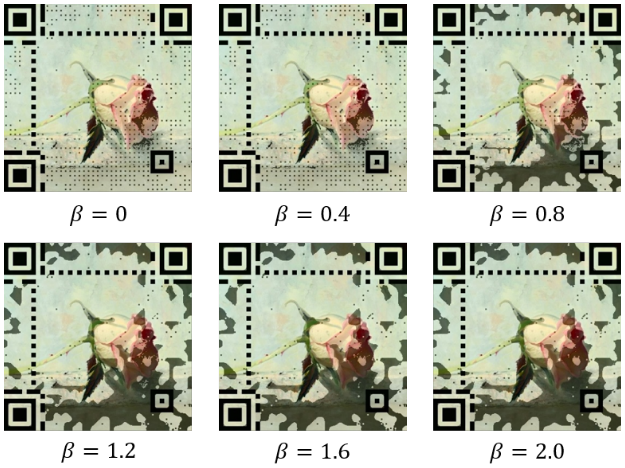


Fig. 8: As the value of β increases, the scattered points are better hidden into the blended image, along with the increase of decoding robustness.

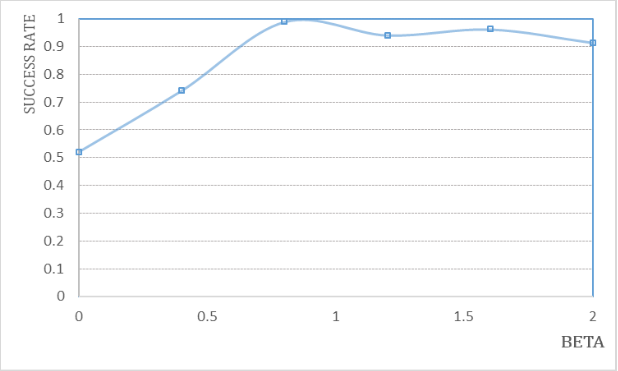


Fig. 9: Reading success rate for different β values.

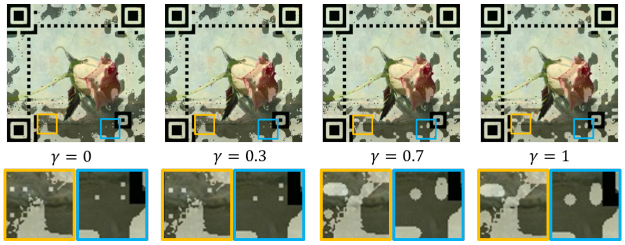


Fig. 10: Sample generated QR Codes for different γ values. Local details are shown in the second row.

$$S^{square}_{x,y} = \begin{cases} 1, & \text{central area} \\ 0, & \text{else} \end{cases} \tag{13}$$

$$S^{gaussian}_{x,y} = \exp - \frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2} \tag{14}$$

$$S^{star}_{x,y} = \exp - \frac{(|x - x_c| + |y - y_c|)^2}{2\sigma^2} \tag{15}$$

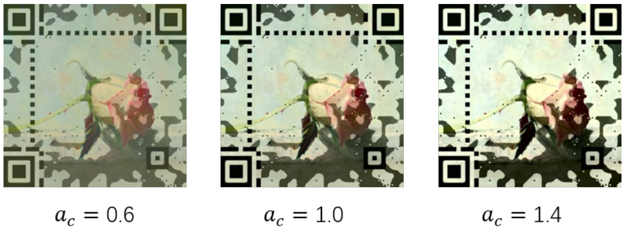


Fig. 11: The visual effect of adjustment on image contrast.

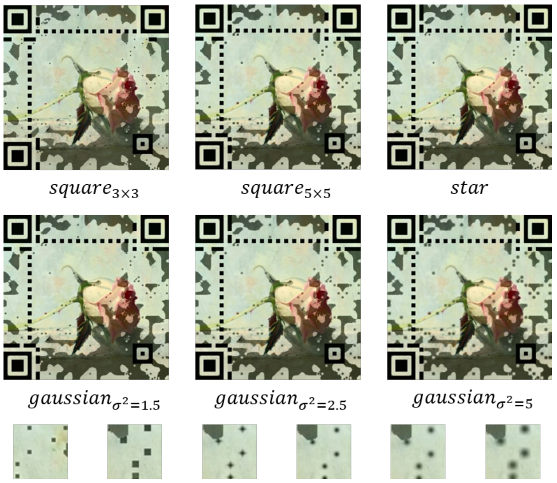


Fig. 12: The visual effect of different shapes of points

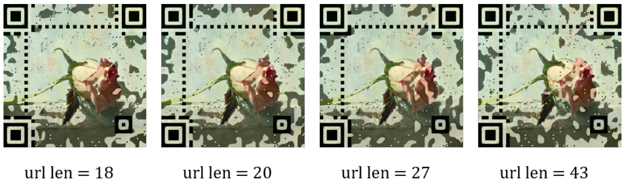


Fig. 13: Q^{VF} generated with different data length.

G. Length of the Encoded Data

The data length to be encoded may affect the visual appearance of the generated QR Code due to its direct relationship with the number of controllable modules in Q^{Art} generation. The longer the data bit stream is, the less flexible the space is left for Q^{Art} to assemble the background, which will finally hurt the visual quality of Q^{VF}. We test several URLs with different length as the encoded information and render them into the same background image as is shown in Fig. 13. It can be seen that when the URL length increases more scattered points appear, especially in the right side of the resultant image, where the data bit stream is placed.

H. Fail Cases Analysis

The proposed method sometimes fails at the stage where the baseline Q^{Art} is being generated. By generation failure, we mean that Q^{Art} does not resemble the binary image of the background well enough, and the final resultant image has an intolerably low visual quality although it may still be decodable, as shown in Fig. 14. Experimental results show that in most of the fail cases, the background image usually have low contrast so that I^p does not have a clear shape to guide the proper generation of Q^{Art}. Another common cause of failure is that the background image has a too high or too low global brightness. Such images generally violate the natural QR brightness distribution with both black and white parts, leading to difficulty in applying QR beautification. However, standard image enhancement methods work well for some of them. We simply add a standard image-enhance procedure, e.g. *histogram equalization*, before the rendering stage, and it improves the visual appearance for most of the fail cases, one of which is shown in the first row of Fig. 14. While for the extreme case like the one in the second row of Fig. 14, although slight improvement can be observed, the result is still not visually satisfying. Practically, such an image may seldom be selected as QR background considering the low visual information it contains.

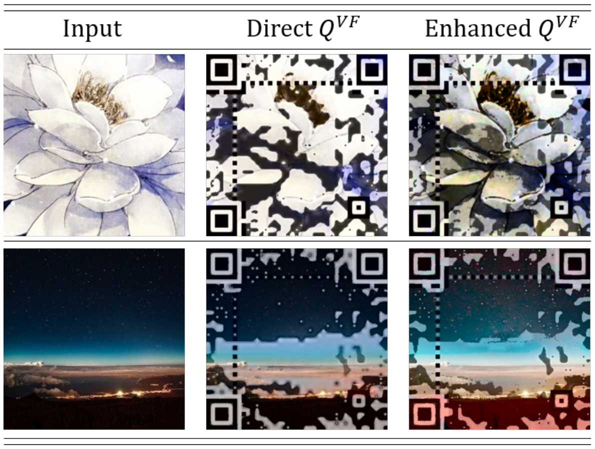


Fig. 14: The second columns shows the fail cases with unpleasant appearance, and the first two examples in the third column show how image-enhance strategy help refine the results. While it won't work for some extreme cases like the last example in column 3.

I. Visual Quality Assessments

Visual quality comparisons between our proposal and several previous works are shown in Fig. 22. The visual results corresponding to [7] are re-implemented by us and are therefore slightly different from that in the original paper. For fairness, we choose not to include this method in the following comparison experiments. Another consideration for us to omit [7] is that a significant amount of original QR blocks are kept by this method, leading to quite obvious visual disturbances. We use two objective metrics, MSSIM and Sobel, to evaluate the similarity and visual satisfaction, respectively. We also carry out a user study to conduct a subjective assessment. Finally, we justify the appropriateness of the two objective metrics by analyzing the correlation between objective and subjective assessments.

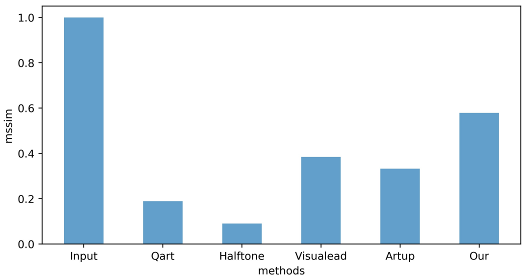
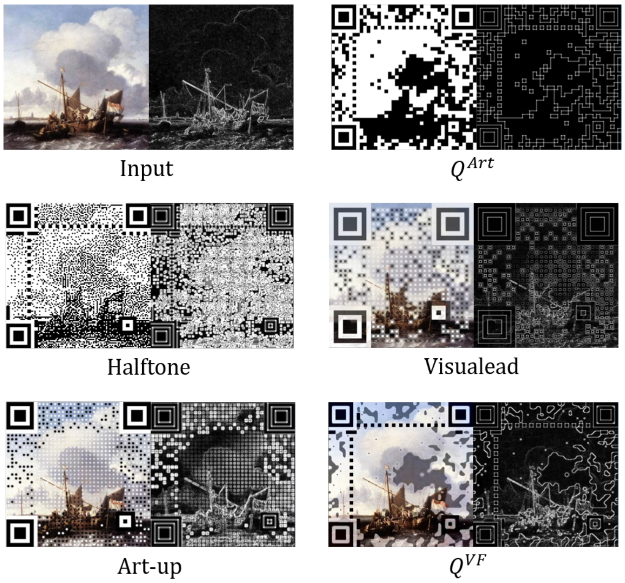


Fig. 15: MSSIM results.


Input

Q^{Art}

Halftone

Visualead

Art-up

Q^{VF}

Fig. 16: Edge Maps by different works show the capacity of our method to reduce the distribution of scattered points from an overall view.

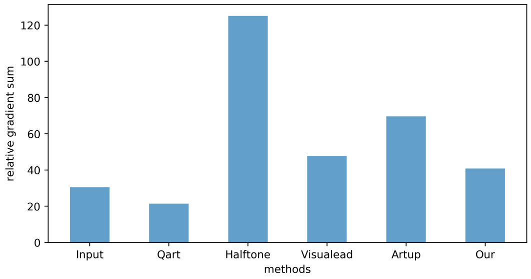


Fig. 17: Sum of image gradient after sobel operator.

1) MSSIM: We adopt the MSSIM (Mean Structure Similarity Index Measure) proposed in [26] as a quantitative image quality assessment metric to measure the similarity between the aesthetic QR Code and corresponding background image. We resize the QR code images of all the compared methods to the same size as Q^{Art}, i.e. 369×369 pixels. MSSIM is then computed by averaging the results calculating from a 9×9 pixel window sliding over the whole image. It should be noted that the whole image including function codes are taken into consideration. We adopt such a strategy mainly based on two considerations. First, although functional patterns are directly kept by some of the existing methods, we still think they are effective components of the QR code and should be taken into account in QR beautification. Second, in some of the methods, certain functional patterns are blended into the background and are therefore difficult to be separated out. We have also tested larger window sizes up to 41×41 pixels. The results are basically the same. The quantitative results are shown in Fig. 15, in which our method shows its ability to maintain relatively higher similarity to the original input background image.

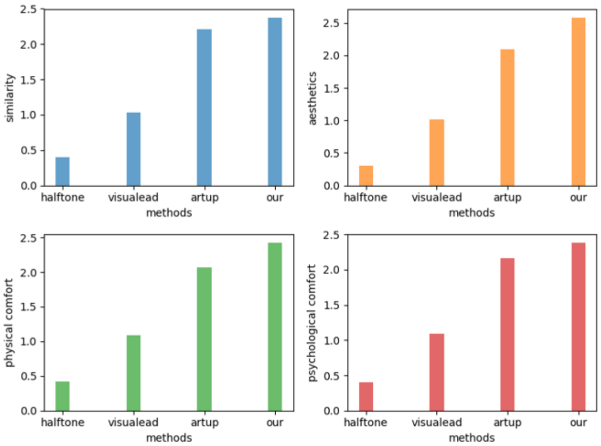


Fig. 18: User study from four subjective assessments: similarity, aesthetics, physical comfort and psychological comfort

2) Sobel: As mentioned before, we consider the continuity of an image as an important factor to indicate how well it resembles a natural image, To quantify this, we apply the Sobel operator [27] to the QR codes to generate edges maps as shown in Fig. 16. It can be observed that compared to other methods, our proposal introduces much less artifact edges. We calculate the summed brightness of the edge map, which is also the sum of the image gradient of the QR images. The quantitative results in Fig. 17 roughly depict the discontinuity of the images. Generally, higher value of the gradient sum indicates more artifact edges which may lead to physical and psychological tiredness when looking at the QR code. Among all compared methods, Q^{Art} has the lowest value of the gradient sum. However, this is achieved by significantly sacrificing the details of the background image. Our method successfully avoids artifact edges by reducing the number of scattered points, leading to a relatively low average gradient value.

3) User Study: Following [7], we invite 50 volunteers, aging from twenties to fifties, to take part in our qualitative questionnaire. Four subjective metrics are evaluated in the questionnaire: similarity, aesthetics, physical comfort and psychological comfort. when the volunteers stare at the image for over 5 seconds. Each individual is asked to stare at each QR image for over 5 seconds and then rank the four comparing beautified QR results regarding the four metrics, respectively. The ranking are then converted into numeric scores for fair comparison. Specifically, for each metric, we can get a ranking result of the four comparing methods. The 1^{st}, 2^{nd}, and 3^{rd} places gain 3/4, 2/4, and 1/4 points, respectively. We average the results coming from all the individuals. As shown in Fig. 18, the results indicate the superiority of our method in all the four subjective metrics.

We further test the correlations between objective and subjective metrics to justify if the proposed objective quality metrics, MSSIM and Sobel, are appropriate. In Fig. 19, we provide scatter plots between MSSIM and similarity assessment; and between Sobel and aesthetics assessment, as shown in Fig. 19. The results show a positive correlation for the

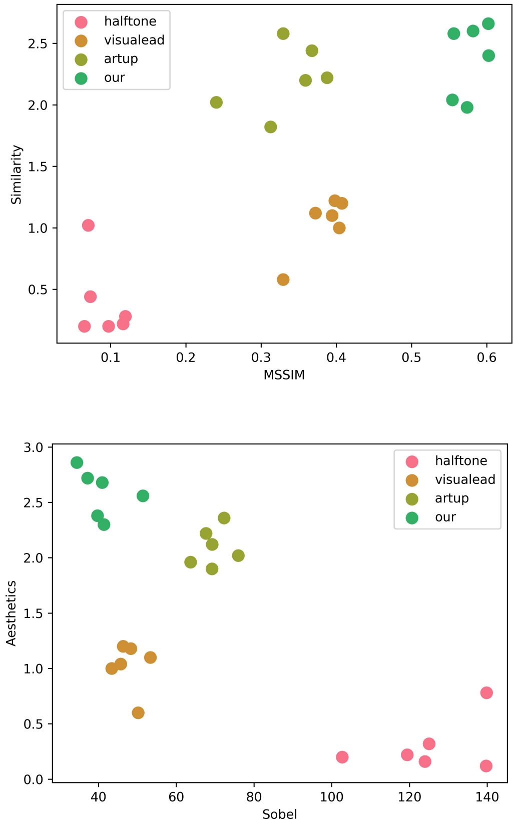


Fig. 19: Correlation between objective metrics and subjective assessments.

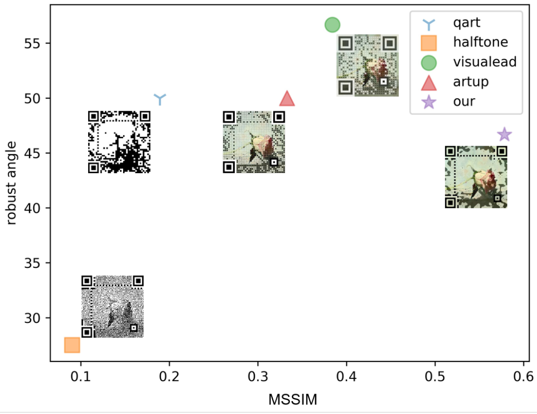


Fig. 20: Joint comparison of robustness and visual similarity.

former and a negative correlation for the latter, indicating that the aforementioned objective evaluations reflect human feelings quite reasonably.

J. Joint evaluation of robustness and similarity

To compare with previous works on both decoding robustness and visual appearance, we draw a 2-D map where the X-axis represents the MSSIM metric and the Y-axis represents the maximum of angles between the cell-phone and QR normal directions when the QR code still readable. The reason why we choose this metric is that most of the methods including us has a relatively high decoding success rate (95%) under ideal conditions, so it's hard to compare with each other especially when the sample scale is small. Considering that the decoding success rate is sensitive to camera angles, we use the maximum decodable angle as a metric of robustness. Concretely, we rotate the camera around the QR with a fixed distance of 30cm, as the angle between the camera optical axis and QR norm vector gets larger, it gets harder to be decoded, and we record the maximum decodable angle as a measurement of robustness. The results can be seen in Fig 20, which shows that our method achieves the highest MSSIM, and a moderate maximum decodable angle. The results is reasonable as visualead [5] and artup [16] is using binary and larger areas for central part of each module, while our center points are small and are not binarized. Enlarging the center area from 3×3 to 5×5 as in Sec. IV-F or changing thresholds as in IV-C3 may enhance our method.

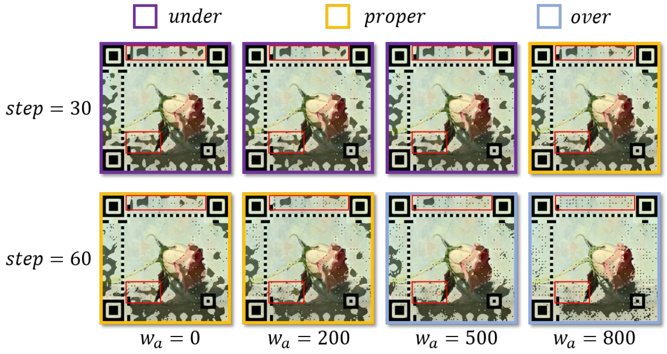


Fig. 21: Purple means under-transferred where the block-like structures are still conspicuous as in the red boxes; orange means properly transferred with smooth patterns; blue means over transferred and the dark-light regions are away from Q^{Art} black-white distribution. Higher weight for L_a accelerates the style transferring procedure.

K. Effect of accuracy reconstruction loss

The reconstruction loss L_a helps to improve the reading robustness as well as speeding up the style transfer process. Comparatively speaking, the acceleration merit is more obvious as shown in Fig. 21. Without L_a, it takes 60 iterations of backpropagation so that Q^{Art}'s block-like structures are properly eliminated. However, when the accuracy loss is added, the number of backpropagation iterations can be reduced to 30 to produce the basically the same results, as can be seen in Fig 21. Considering that the style transfer process is the most time-consuming part of the pipeline, the introduction of the accuracy loss can greatly reduce the processing time of the whole procedure.

V. Conclusions

In this work, we propose a new method for generating aesthetic QR Codes with both pleasant visual appearance and

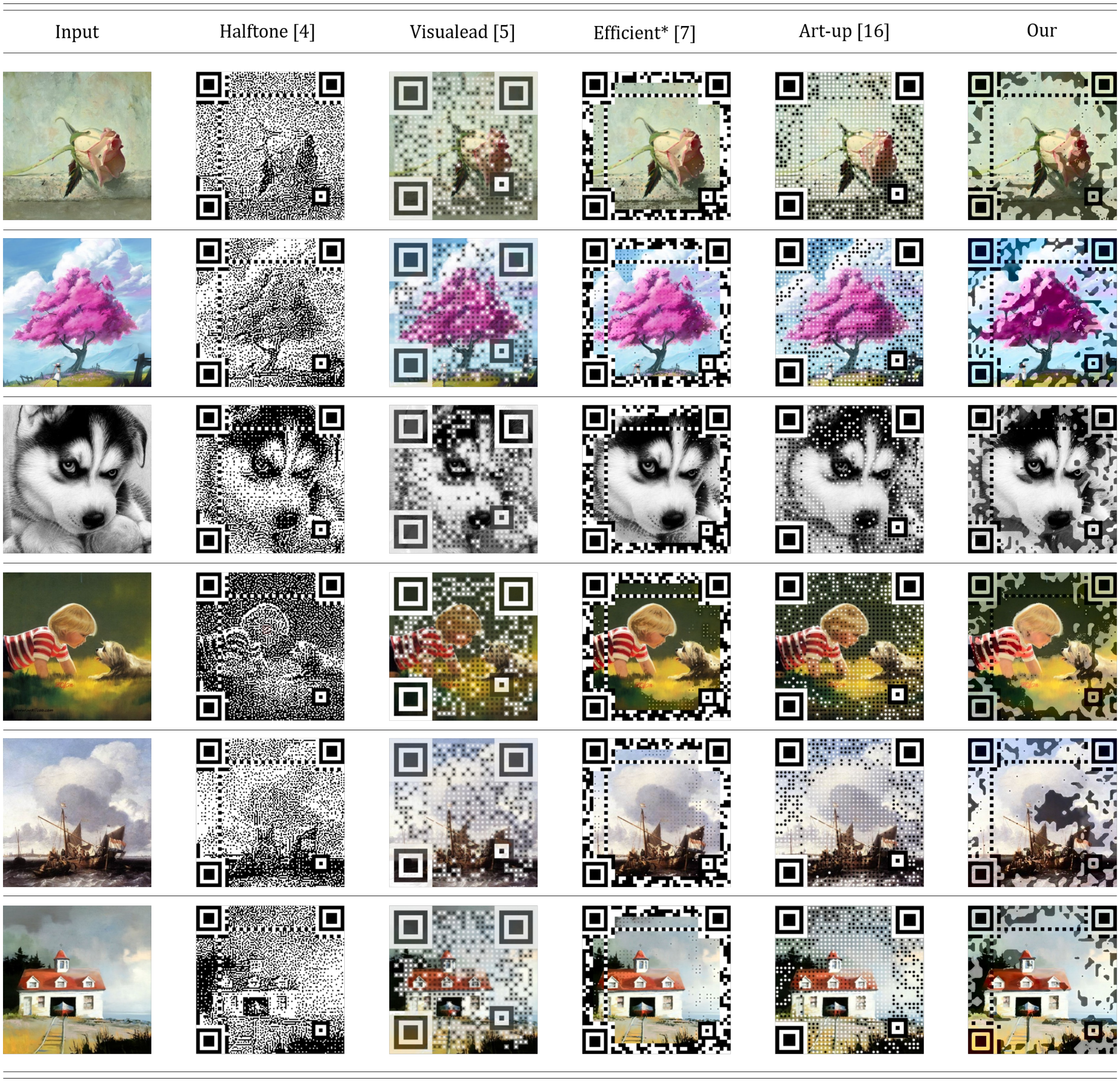


Fig. 22: Visual comparison of aesthetic QR Codes generated using different methods. Mark * means that the results of [7] are re-produced by ourselves.

decoding robustness. The multi-stage procedure first adopts the advanced Q^{Art} generation method as the baseline, and then leverages the application of neutral style transfer to significantly reduce the number of scattered points which may hurt the visual quality, allowing the result to resemble natural images with less artificial traces. Finally, by iteratively updating the weighting maps used in the rendering stage, proper values are assigned to parameters used in the whole procedure for to achieve a balance between robustness and visual appearance.

The procedure proposed in the paper still has some drawbacks to be improved in the future. Firstly, the basic QR code structure is still perceptible in the resultant images. A promising future direction might be to hide the whole QR information into the background image. Secondly, we take a Gaussian kernel shaped mask by default currently, which can be improved by automatically detecting the salience area of the background image and generate a mask accordingly.

References

[1] S. Ono, K. Morinaga, and S. Nakayama, “Animated two-dimensional barcode generation using optimization algorithms - redesign of formulation, operator, and quality evaluation.” *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 13, pp. 245–254, Jan. 2009.

[2] Z. Baharav and R. Kakarala, “Visually significant QR codes: Image blending and statistical analysis,” in *IEEE International Conference on Multimedia and Expo*, Jul. 2013, pp. 1–6.

[3] A.Falcon, “40 gorgeous QR code artworks that rock,” http://www.hongkiat.com/blog/qr-code-artworks/, accessed Dec, 2018.

[4] H.-K. Chu, C.-S. Chang, R.-R. Lee, and N. J. Mitra, "Halftone QR codes," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 217:1–217:8, Nov. 2013.

[5] U. P. N. Aliva and F. Itamar., "Visualead," http://www.visualead.com/, accessed Dec, 2018.

[6] Y.-S. Lin, S.-J. Luo, and B.-Y. Chen, "Artistic QR code embellishment," *Computer Graphics Forum*, vol. 32, no. 7, pp. 137–146, 2013.

[7] S. S. Lin, M. C. Hu, C. H. Lee, and T. Y. Lee, "Efficient QR code beautification with high quality visual content," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1515–1524, 2015.

[8] M. Xu, H. Su, Y. Li, X. Li, J. Liao, J. Niu, P. Lv, and B. Zhou, "Stylize aesthetic QR code," *IEEE Transactions on Multimedia*, vol. 21, pp. 1960–1970, 2018.

[9] T. Wakahara and N. Yamamoto, "Image processing of 2-dimensional barcode," in *International Conference on Network-based Information Systems*, Sept. 2011, pp. 484–490.

[10] D. Samretwit and T. Wakahara, "Measurement of reading characteristics of multiplexed image in QR code," in *Third International Conference on Intelligent Networking and Collaborative Systems*, Nov. 2011, pp. 552–557.

[11] Y. H. Lin, Y. P. Chang, and J. L. Wu, "Appearance-based QR code beautifier," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2198–2207, 2013.

[12] G. J. Garateguy, G. R. Arce, D. L. Lau, and O. P. Villarreal, "QR images: optimized image embedding in QR codes." *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2842–2853, 2014.

[13] Y. Zhang, S. Deng, Z. Liu, and Y. Wang, "Aesthetic QR codes based on two-stage image blending," in *International Conference on Multimedia Modeling*, 2015, pp. 183–194.

[14] M. Kuribayashi and M. Morii, "Enrichment of visual appearance of aesthetic QR code," in *International Workshop on Digital Watermarking*, 2015.

[15] ——, "Aesthetic QR code based on modified systematic encoding function," *IEICE Transactions on Information and Systems*, vol. 100, no. 1, pp. 42–51, 2017.

[16] M. Xu, Q. Li, J. Niu, X. Liu, and Z. Bing, "Art-up: A novel method for generating scanning-robust aesthetic QR codes," *arXiv preprint arXiv:1803.02280*, 2018.

[17] R. Cox., "Qart," http://research.swtch.com/qart/, accessed Dec, 2018.

[18] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *IEEE Conference Computer Vision and Pattern Recognition*, 2016.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, 2016, pp. 694–711.

[21] A. J. Champandard, "Semantic style transfer and turning two-bit doodles into fine artworks," *arXiv preprint arXiv:1603.01768*, 2016.

[22] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," *IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017.

[23] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, "Visual attribute transfer through deep image analogy," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 115, Jul 2017.

[24] S. Owen, "Zxing, multi-format 1d/2d barcode image processinglibrary with clients for android, java and c++," https://code.google.com/p/zxing/, accessed Dec, 2018.

[25] "Zbar bar code reader," http://zbar.sourceforge.net/index.html, accessed Dec, 2018.

[26] W. Zhou, B. Alan Conrad, S. Hamid Rahim, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[27] R. O. Duda and P. E. Hart, "Pattern classification and scene analysis," *IEEE Transactions on Automatic Control*, vol. 19, no. 4, pp. 462–463, 2003.