HALFTONING WITH MULTI-AGENT DEEP REINFORCEMENT LEARNING

Haitian Jiang1, Dongliang Xiong1, Xiaowen Jiang1, Aiguo Yin2, Li Ding3, Kai Huang1

1Zhejiang University, Hangzhou, China
2Pantum Electronics Co., Ltd., Zhuhai, China
3Apex Microelectronics Co., Ltd., Zhuhai, China

ABSTRACT

Deep neural networks have recently succeeded in digital halftoning using vanilla convolutional layers with high parallelism. However, existing deep methods fail to generate halftones with a satisfying blue-noise property and require complex training schemes. In this paper, we propose a halftoning method based on multi-agent deep reinforcement learning, called HALFTONERS, which learns a shared policy to generate high-quality halftone images. Specifically, we view the decision of each binary pixel value as an action of a virtual agent, whose policy is trained by a low-variance policy gradient. Moreover, the blue-noise property is achieved by a novel anisotropy suppressing loss function. Experiments show that our halftoning method produces high-quality halftones while staying relatively fast.

Index Terms— Halftoning, deep learning, multi-agent reinforcement learning, blue noise

1. INTRODUCTION

Digital halftoning is the technique that converts continuous-tone images into images whose pixel values are limited to only two levels. Thanks to the low-pass filtering property of our human visual system (HVS), a halftone image can be perceived as its continuous-tone counterpart from a sufficient distance. According to the dot clustering style, halftoning images can be classified into clustered-dot and dispersed-dot. Besides, there are two types of halftone textures: periodic and aperiodic [1]. In this paper, we focus on generating halftones with dispersed-dot and aperiodic features, as this kind of halftone pattern gives the best visual pleasure, which is termed as the “blue-noise” property [2].

Three kinds of traditional halftoning techniques are widely used. Ordered dithering methods [3, 4] quantify pixels by comparing the continuous-tone image with a predefined dithering array. Those methods have high parallelism due to the pixel-wise processing style but sacrifice the halftone quality. On the other hand, error diffusion methods [5, 6, 7, 8] follow a simple principle: keep the local tone unchanged. At each step, a pixel is quantized and the quantization error is diffused to adjacent unprocessed pixels. Even though unpleasant visual artifacts are introduced, error diffusion keeps a good balance between quality and efficiency. Finally, search-based methods [9, 10] treat halftoning as an optimization problem. Optimization algorithms such as greedy search or simulated annealing provide pretty good halftone results, while the searching time cost is usually unacceptable on real-time devices. Considering the halftone quality and the efficiency comprehensively, error diffusion may be the most reasonable method in practical situations, but there is still room for improvement.

Deep neural network (DNN) has recently brought new developments to halftoning. Kim et al. [11] proposed the idea of rescreening, which dithers an image with another reference halftone image. They first pointed out that a native convolutional neural network (CNN) cannot create position-dependent properties in halftones, and overcame this problem by embedding the position information to the input tensor. Several works [12, 13] viewed halftoning as an unsupervised image translation problem, and tackled it via generative adversarial networks. However, they all considered only the periodic pattern. Xia et al. [14] proposed a reversible deep halftoning solution, which introduces a dynamically sampled Gaussian noise map to enable a CNN to synthesize aperiodic halftone patterns. Although the result halftones make it easier to recover color and details, they fail to achieve a strong blue-noise property, and their two-stage training scheme is...
complex and unstable.

In this paper, we propose HALFTONERS, a novel deep halftoning method generating halftones with high quality as well as good blue-noise property. Similar to search-based methods, HALFTONERS tries to maximize a image quality objective metric, but with a parameterized policy. Specifically, we treat halftoning as a multi-agent reinforcement learning (MARL) problem [15, 16], where pixels in the output halftone image are viewed as the actions chosen by virtual agents, as illustrated in Figure 1. The agents’ shared policy is parameterized by a CNN, which has high parallelism and can be accelerated by GPUs or NPUs. Motivated by the counterfactually multi-agent (COMA) method [15], we estimate one agent’s performance measure by accumulating each action’s global reward times the corresponding probability, while sampling other agents’ actions. Thanks to the one-step generating process and the low-dimensional action space, our policy gradient has a lower variance, which leads to a stable training process. Moreover, to achieve the blue-noise property, we design a loss function which suppresses the anisotropy in the frequency domain. HALFTONERS is simple to implement, both in terms of training and testing. Extensive experiments demonstrate the significance of our proposed method in comparison to previous works.

2. PROPOSED METHOD

In this section, we first present our MARL model for halftoning. Then an anisotropy suppressing loss is introduced to achieve the blue-noise property. Finally, we provide a simple and stable training algorithm.

2.1. MARL Model

Given a continuous-tone image \( c \in C \), a search-based halftoning method generates a halftone image \( h \in \{0, 1\}^N \) by minimizing the visual error \( E(h, c) \), where \( E(\cdot) \) is the error metric (e.g. MSE, SSIM [17]) and \( N \) is the number of pixels. Binary pixels \( h_a \in \{0, 1\} \) compose \( h \), in which \( a \in A \equiv \{1, \ldots, N\} \) identifies the pixel index. Now we formulate halftoning as a fully cooperative multi-agent reinforcement learning problem (see Figure 1 for an overview).

State. The environment state is defined as \( s \equiv (c, z) \), where \( z \) is the dynamically sampled Gaussian noise map enabling CNNs to dither constant grayness images [14].

Agents. A virtual agent with the shared parameterized policy \( \pi \) is constructed at each pixel position, and there are \( N \) agents in total. After observing the state \( s \) and communicating with other agents during the CNN forward pass, each agent gives its probabilities of actions:

\[
P(h_a|c, z; \theta) \approx \pi_a(h_a|c, z; \theta),
\]

where \( \theta \) denotes the policy parameters. The joint policy is defined as \( \pi \equiv [\pi_1, \ldots, \pi_N] \). We believe that the Gaussian noise map as a prior latent variable offers an image-agnostic dependence source that enables the independent generation of halftone pixels:

\[
\pi(h|c, z; \theta) = \prod_a \pi_a(h_a|c, z; \theta).
\]

Reward. The global reward function is defined as:

\[
R(h, c) \equiv -E(h, c).
\]

To keep the notion simple, we leave it implicit that \( \pi \) and \( R \) are functions of \( c, z, \theta \), and the gradient is with respect to \( \theta \).

Policy Gradient. The goal of our MARL model is to maximize the expected global reward:

\[
J(\theta) = \mathbb{E}_{c,z} \left[ \mathbb{E}_{h \sim \pi}[R(h, c)] \right].
\]

We minimize the negative expected reward with a low-variance policy gradient estimation:

\[
\nabla L_{MARL}(\theta) = -\nabla J(\theta) = -\mathbb{E}_{c,z} \left[ \mathbb{E}_{h' \sim \pi} \left[ \sum_a \nabla \pi_a(h_a) R(h_a, h'_a) \right] \right].
\]

where \( h_{-a} \equiv [h_1, \ldots, h_{a-1}, h_{a+1}, \ldots, h_N] \) and \( h' \) denotes the sampled joint actions. This is motivated by the COMA method [15], which assesses one agent’s advantage while keeping the other agents’ sampled actions fixed. But what’s different is: we directly sum over an agent’s all actions, rather than estimating it with its one time sampling like REINFORCE [18]. It is possible due to the low-dimensional action space (only 2). The total calculation amount is also acceptable, as an agent only affects a limited area. Note that the Markov decision process (MDP) in our RL problem has only one step, so there is no need to design another NN to predict the accumulated reward as in actor-critic methods. The proof of Equation (5) is given as follows.

**Proof.**

\[
\nabla L_{MARL}(\theta) = -\nabla J(\theta) = -\mathbb{E}_{c,z} \left[ \sum_h \nabla \pi(h) R(h) \right]
\]

\[
\sum_h \nabla \pi(h) R(h)
\]

\[
= \sum_h \sum_a \nabla \pi_a(h_a) \pi_{-a}(h_{-a}) R(h_a, h_{-a})
\]

\[
= \sum_a \sum_{h_{-a}} \nabla \pi_a(h_a) \pi_{-a}(h_{-a}) R(h_a, h_{-a}) \sum_{h_{-a}} \pi_{-a}(h_{-a})
\]

\[
= \sum_a \sum_{h_{-a}} \nabla \pi_a(h_a) \pi_{-a}(h_{-a}) \sum_{h_{-a}} \pi_{-a}(h_{-a}) R(h_a, h_{-a})
\]

\[
= \sum_a \pi(h'_a) \sum_{h_{-a}} \nabla \pi_a(h_a) R(h_a, h'_{-a})
\]

\[
= \mathbb{E}_{h' \sim \pi} \left[ \sum_a \nabla \pi_a(h_a) R(h_a, h'_{-a}) \right]
\]
Although the proposed MARL model enables training a CNN to output non-differentiable discrete images, the desired blue-noise property [2] is not explicitly guaranteed. Even worse, the convolution operation makes it easy to create the stripe artifacts (shown in Figure 2 (a)), as all agents with the same policy tend to enhance the orientation preference. According to [1], the blue-noise halftone of a constant grayness image should keep the following spectral characteristics: (a) little low-frequency spectral components; (b) high-frequency spectral region; (c) a peak at the blue-noise principal frequency \( f_b \); (d) the anisotropy should be minimized to -10db at all frequency. Xia et al. [14] proposed to penalize the low frequency. However, we believe that the excessive anisotropy is the key problem to be solved.

Given the definition of power spectral estimation and radially averaged power spectrum density (RAPSD):

\[
\hat{P}(f) \approx \frac{1}{N} |\text{DFT}(h)|^2
\]

(8)

\[
P(f) = \frac{1}{n(r(f)))} \sum_{f \in r(f)} \hat{P}(f),
\]

(9)

where \( n(r(f))) \) is the number of discrete frequency samples in an annular ring with width \( \Delta_f = 1 \) around \( f_r \), the anisotropy is defined as:

\[
A(f_r) = \frac{1}{n(r(f)))} \sum_{f \in r(f)} \frac{(\hat{P}(f) - P(f_r))^2}{P^2(f_r)}.
\]

(10)

To achieve the blue-noise property, we propose an anisotropy suppressing loss function:

\[
\mathcal{L}_{AS}(\theta) = \mathbb{E}_{c,h} \left[ (\hat{P}_\theta(f) - P(f))^2 \right],
\]

(11)

which minimizes the numerator in Equation (10). We could treat \( A(f_r) \) as part of the reward in the MARL model, but the power spectrum item would result in dense gradient computations. Here we take the differentiable probabilities \( \pi(h) \) as a proxy for \( h \) to estimate the power spectrum in Equation (8), since it can already capture the anisotropy of \( h \). As the spectral analysis only make sense for dithered constant grayness images, we optimize this loss on a batch of uniformly sampled \( c_g \) in every training iteration following [14]. As shown in Figure 2 (b), the artifact vanishes as a result of trained HALFTONERS.

2.3. Training of HALFTONERS

Now we describe the complete training procedure. To generate halftones with the tone consistency as well as structural similarity, we define the error metric \( E \) as:

\[
E(h, c) = \text{MSE}(G(h), G(c)) - \omega_{SSIM}(G(h, c)),
\]

(12)

where \( G(\cdot) \) is the HVS low-pass filter and \( \omega_{SSIM}(\cdot) = 0.006 \). Following [19] we adopt the Näsänen HVS model [20] with a moderate \( scale = 2000 \). We train the model from scratch optimizing the combined loss function:

\[
\mathcal{L}_{total} = \mathcal{L}_{MARL} + \omega_{A}\mathcal{L}_{AS},
\]

(13)

where hyper-parameter \( \omega_{A} = 0.002 \) is set empirically.

In the testing phase, agents give their discrete actions through the simple argmax operation (see Figure 1). One may consider a more complex halftone error metric since the assessing computation only exists in the training phase.

3. EXPERIMENTS

In this section, we show the implementation details of HALFTONERS and compare both halftone image quality and processing efficiency with prior typical halftoning methods: ordered dithering (VAC [4]), error diffusion (Ostromoukhov [6]), search-based method (DVS [9]) and DNN-based method (Xia et al. [14]). We test all methods on 3,367 images randomly selected from 17,125 images in the VOC2012 dataset [21] following [14]. To take full advantage of the parallelism in some halftoning methods, we implement them on an NVIDIA GeForce RTX 2080Ti GPU with PyTorch 1.10.1.
CUDA 11.3 and cuDNN 8.2.0. For those serial methods, we implement them in C with GCC 6.3.0 (-O3 optimization) on an Intel Xeon E5-2650v4 (2.2 GHz) CPU.

**CNN Architecture & Training details.** We choose ResNet [22] with 16 residual blocks and 32 channels as the backbone CNN model and append a pixel-wise sigmoid layer to output the probability of action 1. The model is trained for 200,000 iterations on the remaining 13,758 images with the Adam optimizer [23]. As halftones mainly depend on the input tone, we augment training images by jittering the brightness randomly (ColorJitter(brightness=0.9) in torchvision). The batch size is set as 64 and the learning rate is adjusted from 3e-4 to 1e-5 with the cosine annealing schedule [24].

**Visual Comparison.** Figure 3 shows halftone samples made by our method and other reference works, whereas Figure 4 shows the gray ramp halftone examples. The structural details are preserved well in both Xia et al.’s and our results, while others suffer in the edge area. Ostromoukhov’s and Xia et al.’s methods introduce annoying artifacts in the flat area, which indicate bad blue-noise property, and it is tackled by the anisotropy suppressing loss function in our results.

**Quantitative Comparison.** The tone consistency is measured by PSNR between the HVS-filtered halftone and the HVS-filtered input continuous-tone. The structural similarity is measured by SSIM between the halftone and the continuous-tone. Table 1 shows the results. We can see that our method achieves a competitive PSNR score and the best SSIM score. According to the preference setting, one can trade SSIM for better PSNR by decreasing $\omega_s$ and vice versa.

**Learning Curves.** We compare the MSE training loss curves of [14] (train the first stage to convergence) and ours (adopting Gaussian HVS for fair comparison). As shown in Figure 5, our method converges faster and achieves better results.

**Processing Speed.** We evaluate the processing time of all methods on a 1200x1200 image (shown in Table 1). Our method outperforms [14] in terms of image quality while being 1.6x faster with only 0.79% parameters. It is possible to close the speed gap between our model and error diffusion approaches by designing a more efficient CNN or using compression techniques [25], and we leave it for future work.

4. CONCLUSION

In this work, we introduce HALFTONERS, a halftoning method based on multi-agent deep reinforcement learning. Blue-noise halftone results with high quality are generated thanks to the low-variance policy gradient and the anisotropy suppressing loss function. Additionally, our training procedure is more stable and simpler compared to existing deep halftoning methods. In the meantime, the high parallelism of CNN makes it possible to deploy our model on real-time devices. Promising future directions include designing more efficient NN architectures, constructing more complex HVS or device models, and expanding to multi-toning.

**Acknowledgments.** This work is supported by the National Key R&D Program of China (2021YFB2206200).

---

**Table 1.** Quantitative comparison with prior halftoning methods. “*” indicates this method is GPU-accelerated.

| Method       | PSNR | SSIM | Param. # | Time (ms) | GPU-accelerated |
|--------------|------|------|----------|-----------|-----------------|
| VAC (64) [4] | 29.19| 0.082| 4,096    | 0.04*     |                 |
| VAC (512) [4]| 29.29| 0.082| 262,144  | 0.04*     |                 |
| Ostromoukhov [6] | 33.07| 0.102| 384      | 18.64     |                 |
| DBS [9]     | 34.00| 0.084| -        | 3008.71   |                 |
| Xia et al. [14] | 29.31| 0.159| 37.8M    | 280.53*   |                 |
| Ours        | 31.32| 0.176| 298,945  | 171.50*   |                 |
5. REFERENCES

[1] D. L. Lau and G. R. Arce, Modern Digital Halftoning, CRC Press, 2008.

[2] Robert A. Ulichney, “Dithering with blue noise,” Proceedings of the IEEE, vol. 76, no. 1, pp. 56–79, 1988.

[3] Bryce E. Bayer, “An optimum method for two-level rendition of continuous tone pictures,” in IEEE International Conference on Communications, 1973, pp. 2611–2615.

[4] Robert A. Ulichney, “Void-and-cluster method for dither array generation,” in Human Vision, Visual Processing, and Digital Display IV, 1993, pp. 332–343.

[5] R. W. Floyd and L. Steinberg, “An adaptive algorithm for spatial grayscale,” Proceedings of the Society for Information Display, vol. 17, pp. 75–77, 1976.

[6] Victor Ostromoukhov, “A simple and efficient error-diffusion algorithm,” in Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, 2001, pp. 567–572.

[7] Jianghao Chang, Benoît Alain, and Victor Ostromoukhov, “Structure-aware error diffusion,” ACM Transactions on Graphics, vol. 28, no. 5, pp. 1–8, 2009.

[8] Yun-Fu Liu and Jing-Ming Guo, “Dot-diffused halftoning with improved homogeneity,” IEEE Transactions on Image Processing, vol. 24, no. 11, pp. 4581–4591, 2015.

[9] Mostafa Analoui and Jan P. Allebach, “Model-based halftoning using direct binary search,” in Human Vision, Visual Processing, and Digital Display III. International Society for Optics and Photonics, 1992, vol. 1666, pp. 96–108.

[10] Wai-Man Pang, Yingge Qu, Tien-Tsin Wong, Daniel Cohen-Or, and Pheng-Ann Heng, “Structure-aware halftoning,” ACM Transactions on Graphics, vol. 27, no. 3, pp. 1–8, 2008.

[11] Tae-Hoon Kim and Sang Il Park, “Deep context-aware descreening and rescreening of halftone images,” ACM Transactions on Graphics, vol. 37, no. 4, 2018.

[12] Jing-Ming Guo and S. Sankarasrinivasan, “H-gan: Deep learning model for halftoning and its reconstruction,” in IEEE International Conference on Consumer Electronics, 2020, pp. 1–2.

[13] Jianjin Gu and Li Li, “A new method of halftoning and inverse halftoning based on gan network,” in Advances in Artificial Intelligence and Security, 2021, pp. 119–131.

[14] Menghan Xia, Wenbo Hu, Xueting Liu, and Tien-Tsin Wong, “Deep halftoning with reversible binary pattern,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 14000–14009.

[15] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson, “Counterfactual multi-agent policy gradients,” in Proceedings of the AAAI Conference on Artificial Intelligence, 2018, vol. 32.

[16] Sven Gronauer and Klaus Diepold, “Multi-agent deep reinforcement learning: a survey,” Artificial Intelligence Review, pp. 1–49, 2021.

[17] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, 2004.

[18] Ronald J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” Machine learning, vol. 8, no. 3, pp. 229–256, 1992.

[19] Sang Ho Kim and Jan P. Allebach, “Impact of hvs models on model-based halftoning,” IEEE Transactions on Image Processing, vol. 11, no. 3, pp. 258–269, 2002.

[20] Risto Näsinen, “Visibility of halftone dot textures,” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-14, no. 6, pp. 920–924, 1984.

[21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge 2012 (voc2012) results,” http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[23] Diederik P. Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” in International Conference on Learning Representations, 2015.

[24] Ilya Loshchilov and Frank Hutter, “SGDR: stochastic gradient descent with warm restarts,” in International Conference on Learning Representations, 2017.

[25] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie, “Model compression and hardware acceleration for neural networks: A comprehensive survey,” Proceedings of the IEEE, vol. 108, no. 4, pp. 485–532, 2020.