Imitation Camouflage Synthesis based on Shallow Neural Network

Cai Xiuxia
Xi'an University of Post and Telecommunications
Zhang Pin (njlgd123@163.com)
Amy Engineering University
Shuaibin Du
Xi'an University of Post and Telecommunications

Research Article

Keywords: Camouflage, Deep Learning, Feature, Painting

Posted Date: February 13th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2567866/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Multimedia Systems on August 9th, 2023. See the published version at https://doi.org/10.1007/s00530-023-01149-z.

Imitation Camouflage Synthesis based on Shallow Neural Network

Cai Xiuxia[1,1], Zhang Pin[2,2*†] and Du Shuaibin[1,1]

[1]School of Electronic Engineering, Xi'an University of Post and Telecommunications, Changan, Xi'an, 710121, Shannxi, China.
[2*]National Key Laboratory of Lightning Protection and Electromagnetic Camouflage, Amy Engineering University, Houbiaoying, Nanjing, 210007, Jiangsu, China.

*Corresponding author(s). E-mail(s): njlgd123@163.com;
Contributing authors: haiqian822@163.com; 1637134380@qq.com;
†These authors contributed equally to this work.

Abstract

Deep learning technology has been widely used in the military field, which have achieved great success. The traditional method for painting camouflage either using the background information or the artificial pattern. None of the traditional methods can both consider the background information and camouflage rules. In this paper, A new automatic camouflage generation framework is proposed. A method for generating camouflage pattern is designed. The imitation camouflage pattern is synthesized from the features of both background and artificial pattern. In our method, the texture feature of both background and traditional pattern patches are extracted from the feature maps of shallow neural network (SNN). Based on the feature maps, statistic information of second order differential and mean subtracted contrast normalized coefficients for texture and color is extracted. By iterating to optimize the imitation camouflage to be generated, the statistical information of the imitation camouflage can approximate the characteristic statistical information of the background and pattern. The new generated camouflage pattern can contain the color and texture information of background; besides, it can maintain the traditional patch camouflage criteria. Our approach makes camouflage painting more flexible and allows the target to better infuse into the background. And our method is designed for the preparation of painting camouflage.

Keywords: Camouflage, Deep Learning, Feature, Painting

1 Introduction

Camouflage technology is a kind of technical measure taken to conceal oneself and confuse the enemy. As an important part of military combat support, camouflage technology has been paid close attention to and studied by scholars [1]. The camouflage includes concealing true targets, setting false targets, carrying out feints, spreading false information and blocking information [2]. Its main purpose is to conceal itself, deceive and confuse the enemy. Camouflage is a main way to deal with investigation and the important content of combat support.

Camouflage is a very comprehensive interdisciplinary subject. It includes subjects of optics, electricity, acoustics, physics, electricity, acoustics, thermal science,chemistry, biology, botany, bionics, materials science and so on[3]. In view of the characteristics of high-tech reconnaissance, modern camouflage technology is mainly a variety of engineering measures to reduce the difference of reflection or radiation energy between the target and the background in optical, thermal infrared, radio wave and other aspects [4]. According to the working spectrum range of the high-tech reconnaissance equipment, the camouflage can be divided into anti-optical detection camouflage, anti-thermal infrared detection camouflage, anti-radar reconnaissance camouflage and anti-acoustic detection camouflage [5]. At present, various stealth weapons are mainly utilized to prevent radar reconnaissance, and to deal with visible light reconnaissance [6]. In this paper, our work focuses on the research of anti-optical detection camouflage.

Under modern conditions, due to the rapid development and wide application of deep learning technology and precision-guided weapons, the capability of battlefield reconnaissance, surveillance, capture and attack has been improved unprecedentedly [7], which poses an increasingly serious threat to the battlefield survivability of ground targets (mobile targets and fixed targets such as weapons and equipment, field fortifications and protective works) [8]. Meanwhile, the technology development for camouflage based on deep learning is a little backward than surveillance. In this paper, we proposed a new method for generating camouflage pattern. At first, feature maps of the background are extracted from convolution layers. If the traditional patch or digital camouflage style needs to be preserved, feature maps of the style also need to be extracted.Besides, mean subtracted contrast normalized coefficients are extracted from the textures of background and pattern model. Then based on the statistic feature information, the similarity measure function is designed. At last, we generate the camouflage pattern by optimizing the similarity measure function. The optimization aim is to find the best camouflage pattern to minimize the similarity measure function.

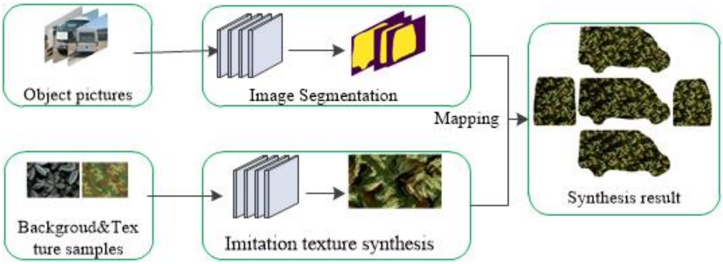

Fig. 1 The framework of imitation camouflage.

In section 2, the previous related works are introduced. Section 3 is the method illustrating how to generate camouflage pattern based on CNN feature maps. In order to verify the validity of our method, we make the experiment in section 4, and experiment analysis is included in this section. Our works are concluded in section 5, and some future researches are given in this section.

2 Related work

With the development of deep learning, there are more and more works about using this technology to develop the camouflage techniques. John G. Fennell [9] proposed to use deep neural network to learn the coloration of one picture, and render the target with the coloration of the background in the picture. This method does not consider the texture of background well, and is not good at large area camouflage. Yufei Chen [10] exploited the image scaling procedure to attack the YOLO-V3. Laszlo Talas [11] proposed to use Generative Adversarial Networks to generated the new texture for camouflage, which does not consider how to make the texture can be compatible with traditional camouflage pattern. In this study, we use the shallow neural network to learn the information of both texture and coloration of the background.The structure of deep Neural Network mainly contains input layer, convolution layer, pooling layer, rule layer, and output layer. We can extract feature information from these layers. Based on these information, new imitation camouflage pattern can be generated.

3 Our method

3.1 The framework of imitation Camouflage synthesis

The framework of our method is shown in Fig 1. We first segment the objects from the photos. The object is repainted using our synthesized imitation camouflage. The shallow neural network is used for extracting the features of background and pattern model. The details for feature extraction method is introduced in section 3.2. By mapping the synthesized imitation camouflage to the object surface, we can finish the camouflage designing work.

3.2 Feature extraction

On the battlefield, no matter in the front, depth or rear, all the un-camouflaged or poorly camouflaged targets are easy to be found by the enemy's optical, radar and infrared detection. Once the target is found and identified, there is a risk of being destroyed by the attack system. The task of camouflage is to systematically implement concealment and deception. Concealment is to make the object invisible or indistinguishable at a certain distance; it is divided into covert target, reducing the salience of target and changing the shape of target. The covert goal is to eliminate the various exposure signs of the target so that it is not detected [12]. All kinds of targets should be concealed when conditions are available. The aim of reducing the significance of targets is to reduce the significance of target exposure symptoms in the background and make it difficult to detect. Usually this is implemented when the context of covert targets is available. Changing the shape of the target is to change the original observation shape of the target, which makes the target difficult to identify or be mistaken as losing military value, so that the target will not be noticed by the enemy reconnaissance. Changing shape is usually done when the target is difficult to conceal or is less salient than camouflage.

In order to generate the camouflage pattern similar to the background, we should first lean the feature of background. The traditional feature extraction methods, such as SIFT [13], cannot express the global feature very well. The CNN feature maps can show the global feature in many channels. So we use convolution neural network to extract the feature maps. The applied structure of the convolution network is VGG-19 [14].

The second order differential is calculated from the feature maps. It can reflect the correlation of different filtering results in the same position of the texture, which is a reflection of characteristic relationship of different filters. Based on the feature maps, the second order differential is calculated in the following way:

$$G_{ij}^{sl} = \sum_k {F_{ki}}^l {F_{kj}}^l \tag{1}$$

Where l is the index of neural-net layers; i and j are indexes of feature maps; $G^l \in R^{N_l \times M_l}$, $N_l$is the numbers of feature maps in layer l , $M_l$is the size of feature map, k is the index of pixels in feature map. The second order differential is also a reflection of correlation of different pixels in the same position of different feature maps with the same filter.

$$G_{ij}^{fl} = \sum_k {F_{ik}}^l {F_{jk}}^l \tag{2}$$

i and j are the index of elements in feature map; k is the index of feature maps. We find that$G_{ij}^{sl} = (G_{ij}^{fl})^T$,so we let

$$G^l = G^{sl} = (G^{fl})^T \tag{3}$$

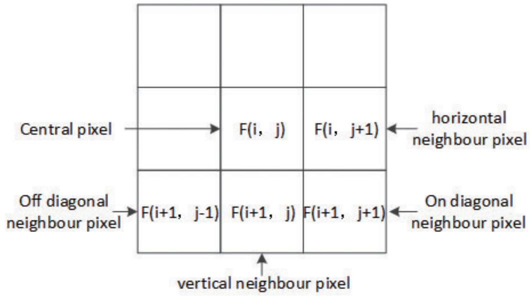

Fig. 2 Four directions MSCN coefficients

We also utilize the mean subtracted contrast normalized (MSCN) coefficients to statistic the information of feature maps, not only the feature maps themselves. The mean subtracted contrast normalized (MSCN) coefficient is calculated in the following way:

$$\hat{F}(i,j) = \frac{F(i,j) - \mu(i,j)}{\sigma(i,j) + C} \tag{4}$$

Where$\mu(i,j) = \sum_k \sum_l w_{k,l} F_{k,l}(i,j), \sigma(i,j) = \sqrt{\sum_k w_k (F_k(i,j) - F(i,j))^2}$ C is a constant which avoid the denominator to be zero.

Because image functions are generally piece-wise smooth aside from sparse edge discontinuities [15]. So, the structure relations among the neighbour pixels are important for texture information. There is high correlation between surrounding pixels. This structure relation is calculated by using the pairwise products of neighboring mean subtracted contrast normalized coefficients along four orientations. It is illustrated in Fig.2. The four directions are horizontal (H), vertical (V), main-diagonal (D1) and secondary diagonal (D2).

The four direction MSCN coefficients pairwise products calculated in the following way:

$$\hat{F}_h(i,j) = F^{MSCN}(i,j)F^{MSCN}(i,j+1) \tag{5}$$
$$\hat{F}_v(i,j) = F^{MSCN}(i,j)F^{MSCN}(i+1,j) \tag{6}$$
$$\hat{F}_{d1}(i,j) = F^{MSCN}(i,j)F^{MSCN}(i+1,j+1) \tag{7}$$
$$\hat{F}_{d2}(i,j) = F^{MSCN}(i,j)F^{MSCN}(i+1,j-1) \tag{8}$$

3.3 Similarity function design

In this section, the second order differential and MSCN coefficients are calculated based on feature maps. There are statistic information for background and camouflage pattern. Our aim is to generate new camouflage pattern based on these pieces of feature information. So matching function which can reflect the similarity between new camouflage and the background is designed. When the texture of traditional camouflage pattern is important, a matching function which can reflect the similarity between new camouflage and the texture of traditional camouflage pattern is needed to be designed.$\vec{b}$ is set to be the background image, and $\vec{t}$ is set be the target camouflage to be synthesized. So the simulate function of the similarity between target camouflage pattern and background is:

$$\begin{aligned} E_{b,l} &= \frac{1}{8{N_l}^2{M^2}_l}\sum_{i,j}((B_{ij}^{sl}-T_{ij}^{sl})^2+(B_{ij}^{fl}-T_{ij}^{fl})^2) \\ &= \frac{1}{8{N_l}^2{M^2}_l}\sum_{i,j}((B_{ij}^{sl}-T_{ij}^{sl})^2+((B_{ij}^{sl})^T-(T_{ij}^{sl})^T)^2) \\ &= \frac{1}{8{N_l}^2{M^2}_l}\sum_{i,j}((B_{ij}^{l}-T_{ij}^{l})^2+((B_{ij}^{l})^T-(T_{ij}^{l})^T)^2) \\ &= \frac{1}{4{N_l}^2{M^2}_l}\sum_{i,j}(B_{ij}^{l}-T_{ij}^{l})^2 \end{aligned} \tag{9}$$

$B$and$T$are the second order differential corresponding to the feature map $\vec{b}$ and $\vec{t}$. The total loss for the all layer loss function is:

$$\ell_b(\vec{b},\vec{t}) = \sum_{l=0}^{L} w_{b,l}E_{b,l} \tag{10}$$

Where $w_{b,l}$ are weights. The smaller the value of the similarity function the more similar the target pattern is to the background. $\vec{s}$ is set to be the traditional camouflage pattern, and $\vec{t}$ is set to be the target camouflage which is to be generated. The simulate function of the four direction MSCN coefficients between target and traditional camouflage pattern is :

$$\begin{aligned} E_{s,l} = &\lambda\left(F_s^l(i,j)-F_t^l(i,j)\right)+\lambda_v\left(F_s^{lv}(i,j)-F_t^{lv}(i,j)\right) \\ &+\lambda_h\left(F_s^{lh}(i,j)-F_t^{lh}(i,j)\right)+\lambda_{d1}\left(F_s^{ld_1}(i,j)-F_t^{ld_1}(i,j)\right) \\ &+\lambda_{d2}\left(F_s^{ld_2}(i,j)-F_t^{ld_2}(i,j)\right) \end{aligned} \tag{11}$$

$\lambda\lambda_v\lambda_{d1}\lambda_{d2}$and λ_{d2} are weights. The total loss for the all-layer loss function is:

$$\ell_s(\vec{s},\vec{t}) = \sum_{l=0}^{L} w_{s,l}E_{s,l} \tag{12}$$

Where $w_{s,l}$ are weights. The smaller the value of the loss function the more similar the texture of target pattern is to the traditional camouflage pattern. When the designed camouflage pattern need to contain the traditional blocks pattern and the background information at the same time, the following loss function is designed:

$$L_{total} = \alpha\ell_b(\vec{b},\vec{t}) + \beta\ell_s(\vec{s},\vec{t}) \tag{13}$$

α and β are weights. If $\alpha > \beta$, the designed camouflage pattern will be more similar to the background information. And if $\alpha < \beta$,the designed camouflage pattern will retain more information of the traditional camouflage pattern. When the newly generated camouflage does not need to retain the traditional camouflage pattern, β can be set to zero. The smaller the value of L_{total} the more similar the target pattern is to the background. That is we are going to find the best $\vec{t}$ to let

$$\min\{L_{total}\} = \min_{\vec{t}}\{\alpha\ell_b(\vec{b},\vec{t}) + \beta\ell_s(\vec{s},\vec{t})\} \tag{14}$$

3.4 Camouflage pattern synthesis

The camouflage synthesized by quantized in similarity function. That means to find the best $\vec{t}$ to let the similarity function smallest. This is a convex optimization problem. Here, Large-scale bound-constrained optimization is a limited-memory(L-BFGS) iterative method, which is used to get the target camouflage. L-BFGS algorithm for solving large nonlinear optimization problems is developed from Quasi-Newton method [16]. The classic optimization problem algorithms includes gradient descent method [17], Newton method [18], conjugate gradient method [19], Quasi-Newton method [16] and so on. All the methods mentioned before are through calculating Hessian matrix or gradient descent to find the optimal result. The Quasi-Newton method can to be represented in the following way:

$$\begin{aligned} f(x) = {} & f(x_{i+1}) + (x - x_{i+1})^T \nabla f(x_{i+1}) \\ & + \tfrac{1}{2}(x - x_{i+1})^T H_{i+1}(x - x_{i+1}) \\ & + o(x - x_{i+1}) \end{aligned} \tag{15}$$

x_{i+1} is one real value, and H_{i+1} is the Hessian matrix in x_{i+1}. The following formula can be obtained by differentiating Formula (15).

$$\nabla f(x) \approx \nabla f(x_{i+1}) + H_{i+1}(x - x_{i+1}) \tag{16}$$

If we let $x = x_i$,the following formula can be got from the formula (16):

$$H_{i+1}^{-1}(\nabla f(x_{i+1}) - \nabla f(x_i)) \approx (x_{i+1} - x_i) \tag{17}$$

Let $\mathrm{B}_{i+1} = H_{i+1}^{-1}$, the B_i can be calculated for the optimist result. Let $t_i = \nabla f(x_{i+1}) - \nabla f(x_i)$ and $s_i = x_{i+1} - x_i$, the k latest values are saved. The L-BFGS calculated in the following way:

$$\begin{aligned} P_i = {} & V_{i-1}^T P_{i-1} V_{i-1} + r_{i-1} s_{i-1} s_{i-1}^T \\ & \vdots \\ & = (V_{i-1}^T \cdots V_{i-m}^T) P_i^0 (V_{i-m} \cdots V_{i-1}) + \\ & r_{i-m}(V_{i-1}^T \cdots V_{i-m+1}^T) s_{i-m} s_{i-m}^T (V_{i-m+1} \cdots V_{i-1}) + \\ & \cdots + r_{i-1} s_{i-1} s_{i-}^T, \end{aligned} \tag{18}$$

Where $r_i = s_{i-1}^T t_{i-1} / t_{i-1}^T t_{i-1}$ is the learning rate; $P_i^0 = r_i I$ is the Initial value, I is the identity Matrix. Our designed loss function L_{total} is in the following way:

$$\frac{\partial \mathrm{L}}{\partial \vec{t}} = \alpha \sum_{l=0}^{L} w_{b,l} \frac{\partial E_{b,l}}{\partial F_{cij}^{l}} + \beta \sum_{l=0}^{L} w_{s,l} \frac{\partial E_{s,l}}{\partial F_{ij}^{MSCN}} \tag{19}$$

If $F_{cij}^{l} > 0$,$F_{\mathrm{t}ij}^{MSCN} \leq 0$

$$\begin{aligned} \frac{\partial \mathrm{L}}{\partial \vec{t}} &= \alpha \sum_{l=0}^{L} w_{s,l} (\frac{1}{{N_l}^2 {M^2}_l} ((B^l)^T (B^l - T^l))_{ij}) \\ &-\beta [\frac{F_{tij}^{MSCN}}{\sigma_{ij}} (\lambda + \lambda_h F_{t,i,j+1}^{MSCN} + \lambda_v F_{t,i+1,j}^{MSCN} \\ &+\lambda_{d1} F_{t,i+1,j+1}^{MSCN} + \lambda_{d2} F_{t,i+1,j-1}^{MSCN}) + F_{tij}^{MSCN} \\ &(\lambda + \lambda_h \frac{F_{t,i,j+1}^{MSCN}}{\sigma_{s,i,j+1}} + \lambda_v \frac{F_{t,i+1,j}^{MSCN}}{\sigma_{t,i+1,j}} \\ &+\lambda_{d1} \frac{F_{t,i+1,j+1}^{MSCN}}{\sigma_{t,i+1,j+1}} + \lambda_{d2} \frac{F_{t,i+11,j-1}^{MSCN}}{\sigma_{t,i+11,j-1}})] \end{aligned} \tag{20}$$

If $F_{cij}^{l} \leq 0$,$F_{\mathrm{t}ij}^{MSCN} \leq 0$

$$\begin{aligned} \frac{\partial \mathrm{L}}{\partial \vec{t}} &= \beta [\frac{F_{sij}^{MSCN}}{\sigma_{sij}} (\lambda + \lambda_h F_{t,i,j+1}^{MSCN} + \lambda_v F_{t,i+1,j}^{MSCN} + \\ &\lambda_{d1} F_{t,i+1,j+1}^{MSCN} + \lambda_{d2} F_{t,i+1,j-1}^{MSCN}) + F_{tij}^{MSCN} (\lambda + \\ &\lambda_h \frac{F_{t,i,j+1}^{MSCN}}{\sigma_{s,i,j+1}} + \lambda_v \frac{F_{t,i+1,j}^{MSCN}}{\sigma_{t,i+1,j}} + \lambda_{d1} \frac{F_{t,i+1,j+1}^{MSCN}}{\sigma_{t,i+1,j+1}} + \lambda_{d2} \frac{F_{t,i+11,j-1}^{MSCN}}{\sigma_{t,i+11,j-1}})] \end{aligned} \tag{21}$$

If $F_{cij}^{l} > 0$,$F_{\mathrm{t}ij}^{MSCN} > 0$

$$\frac{\partial \mathrm{L}}{\partial \vec{t}} = \alpha (\frac{1}{{N_l}^2 {M^2}_l} ((B^l)^T (B^l - T^l))_{ij}) \tag{22}$$

If $F_{cij}^{l} \leq 0$,$F_{\mathrm{t}ij}^{MSCN} > 0$

$$\frac{\partial \mathrm{L}}{\partial \vec{t}} = 0 \tag{23}$$

Where ${B_{ij}}^l$ is the second order differential value of background in position (i, j) . ${T_{ij}}^l$ is the second order differential value of camouflage pattern to be generated in position (i, j). $F_{tij}^{MSCN}$is the MSCN coefficients in in position (i, j) for the camouflage to be generated.

In equation (18), we set $r_i = 1/((\partial L_{total}/\partial \vec{t}_{i+1} - \partial L_{total}/\partial \vec{t}_i)^T U_i)$ to be the learning rate, where $U_i = L_{total}(\vec{t}_{i+1}) - L_{total}(\vec{t}_i)$. The latest 20 $\partial L_{total}/\partial \vec{t}$ and $L_{total}(\vec{t}_i)$ are saved. So in equation (18) has the following formula:

$$V_i = I - r_i (\frac{\partial L_{total}}{\partial \vec{t}_{i+1}} - \frac{\partial L_{total}}{\partial \vec{t}_i}) P_i^T \tag{24}$$

The L-BFGS iteration process continues until either the maximum number of iteration steps or the value of P converges. The $\vec{t}$ corresponding to the last iteration is the final goal result. The vector $\vec{t}$ is transformed into an image, where $\vec{t}$ is the final generated camouflage pattern image. The example of feature

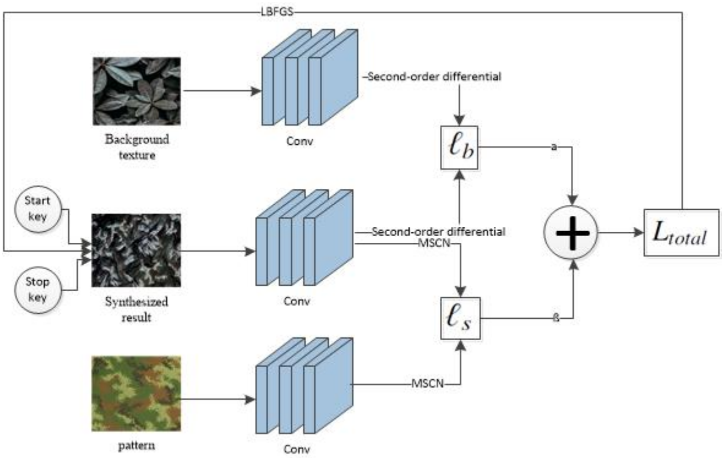

Fig. 3 Camouflage pattern synthesis flowchart

maps extraction is shown in Fig. 3. Where the start and stop keys are controlled by the iteration setting.

3.5 Color matching

The last step for designing imitation camouflage is color mapping. In our designed camouflage synthesising method, we can decide whether to apply the background color of the environment or the traditional camouflage pattern according to the battlefield requirements.

For example, if we want to map the color from pixel x_0 to x_t , the following equation can be used to express the linear relationship between them:

$$x_t = Ax_0 + b \tag{25}$$

when we set x_t be the pixel in the image to be generated, x_0 is the pixel in background image or the traditional camouflage pattern. Because color is a three dimensional vector, so A is a $3 * 3$ mapping matrix. b is a three dimensional vector. The next step is to calculate A and b . Image color mean and covariance between two images are also satisfied the equation (25). So we have the following two equations:

$$\mu_t = A\mu_0 + b \tag{26}$$

$$A\Sigma_t A^T = \Sigma_o \tag{27}$$

μ_0 and μ_t are the color means. They are calculated in the following way:

$$\mu = \sum_i x_i / N \tag{28}$$

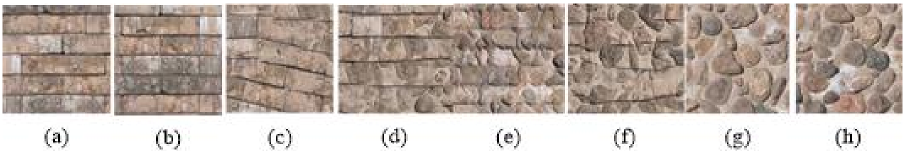

Fig. 4 Examples of synthesized camouflage pattern

$$\Sigma = \sum_i (x_i - \mu)(x_i - \mu)^T / N \tag{29}$$

x_i is the pixel, and N is the image size. The Cholesky decomposition of variance matrix [20] for equation (27) is in the following way:

$$\begin{aligned} & A\Sigma_t A^T = \Sigma_o \\ & \Rightarrow AL_t L_t^T A^T = L_o L_o^T \\ & \Rightarrow AL_t = L_o \\ & \Rightarrow A = L_o L_t^{-1} \end{aligned} \tag{30}$$

$L_t L_t^T$ and $L_o L_o^T$ are the Cholesky decomposition of variance matrix. The result (30) is took into equation (26), Σ_o and Σ_t then the result of b can be obtained. According to A and b , the color from one image to another is mapped.

4 Experimental results and analysis

Table 1 SSIM for Figure 4

Texture	Fig4.(a)	Fig4.(h)
Fig4.(c)	0.4421	0.1376
Fig4.(d)	0.3235	0.1499
Fig4.(e)	0.2200	0.1695
Fig4.(f)	0.1900	0.2063
Fig4.(g)	0.1010	0.3847

In our experiments, the feature maps are extracted from the shallow convolution network.α and β are the key parameters. Here we first show the influence to the synthesized result by changing of α and β. In Fig.4, Fig.4(a) and Fig.4(h) are two exemplars. Fig.4(b)-(g) are synthesized images with the difference parameters. In Fig.4(b), $\alpha = 1.0$ and $\beta = 0.0$. In Fig.4(c), $\alpha = 0.8$ and $\beta = 0.2$. In Fig.4(d), $\alpha = 0.6$ and $\beta = 0.4$. In Fig.4(e), $\alpha = 0.4$ and $\beta = 0.6$. In Fig.4(f), $\alpha = 0.2$ and $\beta = 0.8$. In Fig.4(g), $\alpha = 0.0$ and $\beta = 1.0$. Table 1 show the similarity of synthesized results between exemplars. From the Fig.4 and table 1 we can find that the change of SSIM conforms to the visual change effect of the composite image in Fig.4. But we should note that even if only one sample is used to synthesize the image(Fig.4(b) and Fig.4(g)), the SSIM value between the synthesized result and the exemplar is not very

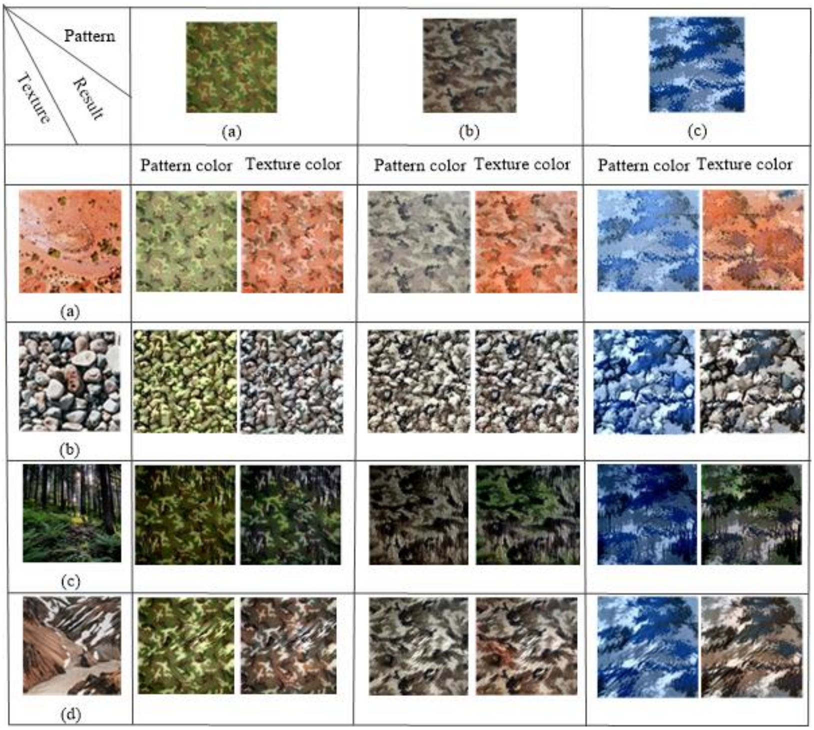

Fig. 5 Examples of synthesized camouflage pattern

high (smaller than 0.5). That mean the synthesized result is very similar to the exemplar (SSIM value is not too small) but not same with it.

Table 2 SSIM for Figure 5

Texture	Pattern(a)	Pattern(b)	Pattern(c)
SSIM-Pattern	0.4308	0.4976	0.5201
SSIM-Texture(a)	0.0648	0.0581	0.0531
SSIM-Pattern	0.5503	0.6335	0.6210
SSIM-Texture(b)	0.0736	0.0705	0.0711
SSIM-Pattern	0.3984	0.4815	0.5262
SSIM-Texture(c)	0.0626	0.0518	0.0650
SSIM-Pattern	0.4391	0.5837	0.6495
SSIM-Texture(d)	0.0493	0.0589	0.0616

To show the effectiveness of our algorithm, the digital camouflage pattern (It is commonly used in military affairs) is selected as the traditional pattern. Besides, we present a number of results in Fig.5, which are synthesized on the basis of different background images as learning samples. In Fig.5, columns of "Pattern color" are the synthesized results save the color of the traditional pattern. The columns of "Texture color" are the synthesized results color is mapped from the of the background image. In this group of experiments, we

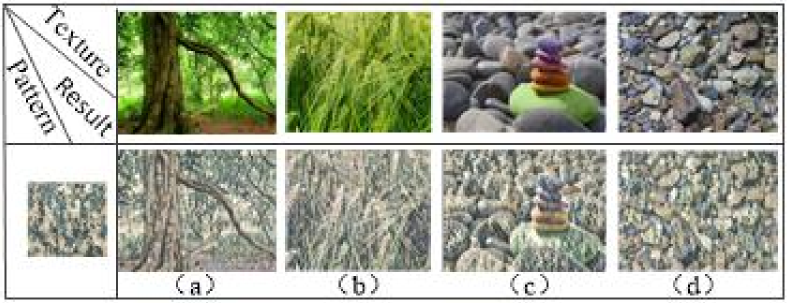

Fig. 6 Examples of synthesized camouflage pattern

set $\alpha = 0.2$ and $\beta = 0.8$. It is because we want to show you the imitation camouflage can present the background texture effect to a certain extent on the basis of presenting the traditional digital camouflage visual effect as a whole. Table 2 shows the SSIM results for Fig.5. From this table, we can find that the SSIM value between synthesized result and camouflage pattern is greater than that between synthesized result and background texture. This is well conform to our logic algorithm setting.

Fig.6 is another group of experiments. In this group of experiments, we set $\alpha = 0.9$ and $\beta = 0.1$. Here we hope that the synthesized imitation camouflage can save the most background textures. We let the traditional digital camouflage pattern appear on the synthesized imitation camouflage without highly changing the background texture. Table 3 shows the SSIM results for Fig.6. From this table, we can find that the SSIM value between synthesized result and background texture is greater than that between synthesized result and camouflage pattern. These scalar quantity values can be consistent with the logic algorithm in this paper.

Table 3 SSIM for Figure 6

synthesized result	Fig5.(a)	Fig5.(b)	Fig5.(c)	Fig5.(d)
SSIM-Pattern	0.0547	0.0440	0.0568	0.0607
SSIM-Texure	0.4404	0.4463	0.4262	0.6380

5 Conclusions

In this paper, we propose a new method for generating imitation camouflage pattern. Second order differential of feature maps and MSCN coefficients are used for extract the features of background texture and camouflage pattern. The color of background texture or camouflage pattern can be mapped to synthesized camouflage. Besides, our method can flexibly control the similarity between synthesized imitation camouflage and background texture by adjusting the weight values. Our approach makes camouflage painting more flexible.

The experiment shows that the quantity values in experiment can be consistent with the logic algorithm.

Acknowledgments. The authors would like to express sincere gratitude to National Natural Science Foundation of Shaanxi Province (Grant No. 2020SF-377) for providing fund for conducting experiments, and also the National Key Laboratory of Lightning Protection and Electromagnetic Camouflage for its support.

References

[1] Xin, Y.A., Wdx, A., Qi, J.A., Ling, L.A., Wnz, B., Jyt, A., Hao, X.B.: Research on extraction and reproduction of deformation camouflage spot based on generative adversarial network model - sciencedirect. Defence Technology **16**(3), 9 (2020)

[2] Stevens, M, Merilaita, S: Animal camouflage: current issues and new perspectives. Philos Trans R Soc B Biol Sci (2009)

[3] Song, X., Pan, J., Zhang, X., Chen, C., Huang, D.: Bionics-based optimization of step-climbing gait in a novel mini-rhex robot. Journal of Bionic Engineering **19**(3), 13 (2022)

[4] Li, L., Li, H., Kou, G., Yang, D., Hu, W., Peng, J., Li, S.: Dynamic camouflage characteristics of a thermal infrared film inspired by honeycomb structure. Journal of Bionic Engineering **19**(2), 13 (2022)

[5] Wei, F.U.: Reconnaisance and anti-reconnaissance of aerospace navigation in high technology war. MODERN DEFENCE TECHNOLOGY (2000)

[6] Zhu, R., Wang, J., Jiang, J., Cuilian, X.U., Liu, C., Jia, Y., Sui, S., Zhang, Z., Liu, T., Chu, Z.: Machine-learning-empowered multispectral metafilm with reduced radar cross section,low infrared emissivity,and visible transparency. Photonics Research (010-005) (2022)

[7] Leira, F.S., Helgesen, H.H., Johansen, T.A., Fossen, T.I.: Object detection, recognition, and tracking from uavs using a thermal camera. Journal of Field Robotics (2020)

[8] Allen, M.A., Flynn, M.E., Machain, C.M.: Us global military deployments, 1950?c2020*:. Conflict Management and Peace Science **39**(3), 351–370 (2022)

[9] Fennell, J., Talas, L., Baddeley, R., Cuthill, I., Scott-Samuel, N.: The camouflage machine: Optimising protective colouration using deep learning with genetic algorithms (2020)

[10] Chen, Y., Shen, C., Wang, C., Xiao, Q., Li, K., Chen, Y.: Scaling camouflage: Content disguising attack against computer vision applications. IEEE Transactions on Dependable and Secure Computing **PP**(99), 1–1

[11] Talas, L., Fennell, J.G., Kjernsmo, K., Cuthill, I.C., Scott-Samuel, N.E., Baddeley, R.J.: Evolving optimum camouflage with generative adversarial networks. Cold Spring Harbor Laboratory (2018)

[12] Yang, X., Xu, W.D., Jia, Q., Liu, J.: Mf-cfi: a fused evaluation index for camouflage patterns based on human visual perception. Defence Technology **17**(5), 7 (2021)

[13] Ke, Y.: Pca-sift : A more distinctive representation for local image descriptors. Proc. CVPR Int. Conf. on Computer Vision and Pattern Recognition, 2004 (2004)

[14] Long, Jonathan, Shelhamer, Evan, Darrell, Trevor: Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)

[15] Jun-Feng, L.I., Zhang, Z.X., Shen, J.M., Automation, D.O., University, S.T.: No-reference image quality assessment based on luminance statistics. Journal of Optoelectronics??laser **27**(10), 1101–1110 (2016)

[16] Wedderburn, R.: Quasi-likelihood functions, generalized linear models, and the gauss??newton method. Biometrika **61**(3), 439–447 (1974)

[17] Lcun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[18] Qi, L., Jie, S.: A nonsmooth version of newton method. Mathematical Programming **58**(1-3), 353–367 (1993)

[19] Dai, Y.H., Li, L.D.: On restart procedures for the conjugate gradient method. Numerical Algorithms (2004)

[20] Yu, H., Chung, C.Y., Wong, K.P., Lee, H.W., Zhang, J.H.: Probabilistic load flow evaluation with hybrid latin hypercube sampling and cholesky decomposition. IEEE Transactions on Power Systems **24**(2), 661–667 (2009)