ISSN: 2664-9640

An automatic digital camouflage pattern generation method based on texture structure

Congli Li, Chengjun Xu*, Chao Xu, Zhe Wei

Ordnance engineering department, army artillery air defense force college, Hefei 230000, China

* 1503708908@qq.com

Abstract

It is of great practical value to automatically generate camo image based on natural environment image. Aiming at the problems that the main generation methods only pay attention to color extraction and mixing, but ignore the image structure, and the shape of camouflage patches is too artificial and lacks randomness, an automatic camo generation method based on the fusion of source image texture and structure is proposed. There are two steps: background texture generation and foreground structure generation. In the background texture generation, the main color of the image is extracted using the K-means algorithm in the LAB space, and the color closest to the color difference is matched in the given color library. Then the patch polygons are randomly generated according to the parameters set by the user, and then the dyeing algorithm is proposed to fill the color, and the background texture reflecting the color information of the source image is obtained. In the foreground structure generation, according to the color difference and area threshold set, a grid point region growth algorithm is proposed to obtain a series of regions. Then, the colors retrieved from the color library are filled successively to obtain the foreground patches reflecting the structural information of the source image. The front background is fused into the camouflage image. The experimental results show that the processing results of this method not only reflect the color details of the source image, but also have the consistency of macro structure. Moreover, the randomness of the algorithm is strong, so it has a certain practicability.

Keywords

Digital camouflage pattern; Fused source image; K-means algorithm; Grid point region growth algorithm; Image processing.

1. Introduction

With the development of reconnaissance technology and the large number of guided weapons, it has become increasingly popular for research on how to effectively camouflage targets and improve survivability. For visible spectral remote sensing reconnaissance, it usually adopts camouflage network to disguise the target and make it as consistent as possible with the background, so the design of camouflage pattern determines the effect of disguise.

The traditional camouflage design method requires manual colorimetry of the camouflage environment. Candidate colors are first selected in the color library, and then patches are manually drawn and colored. This method is highly subjective and inefficient, and unable to adapt to changes in the environment. Modern methods generate digital camouflage pattern based on natural environment image with high flexibility and timely, and the main steps include primary color extraction, patch drawing and color rendering. For primary color extraction, Yujun et al extracted background primary colors using K-means algorithm for color clustering

ISSN: 2664-9640

[1][2][3]; Yunchong Cai et al proposed a primary color extraction based on layer-by-layer fuzzy C-mean algorithm with pyramid structure [4]; Ying Xu used HSI color model to quantify the background color histogram by special quantization and selected the background primary colors using threshold method [5]. For patch generation, Qin Lei et al proposed a digital patch probability distribution model to quantify the distribution of digital camouflage units based on the principle of spatial color mixing [6]; Yunchong Cai et al proposed a fractal dimension estimation method based on fractal Brownian motion to extract the texture features of the background image and used an improved diamond-rhombus algorithm to generate digital camouflage pattern that match the background features [4]; Rongqiang Xu [7] and Shan Xu et al [8] established a speckle library for random loading during drawing; Bison [9] and Minghua Nie et al [10] used a spline function or Bessel function to fit the optimal patches; Rui Wu [11] proposed a digital camouflage unit segmentation method based on the blending principle. A depth-based learning is also applied in camouflage generation that can directly obtain camouflage patterns from background images, such as the CycleGAN-based method by Teng Xu [12].

Although scholars have proposed various solutions, there are still some shortcomings in the existing methods, which are reflected in: (1) most of the methods only focus on preserving the color characteristics of the image, and only a few methods extract the texture information of the image, but still ignore the structural information of the background, which will inevitably cause a certain degree of discontinuity when artifacting; (2) on patch generation, the methods are so artificial that they cannot reflect the dynamic characteristics of the real background, and although they can bring a good experience of subjective vision, they are not random enough; (3) almost all methods render only according to the extracted colors, but ignore the limited number of standard colors for use in production, so the camouflage effect is estimated to be on the optimistic side.

In this regard, this paper proposes a digital automatic camo pattern generation method based on based on the fusion of source image texture and structure and divides into two independent modules that synthesize background information related to the color of the source image and foreground information related to the structure, and finally fuse them. This method takes into account both the structural outline of the natural image and the color and texture as well as randomness and controllability, and uses the current standard colors for coloring, making practical use more flexible and efficient.

2. Methods

The flow of the method is shown in Figure 1. The user selects any rectangular region of an image containing the environment to be camouflaged as the source image, and then enters a processing network with two branches. In the background generation module, according to the collected color dictionary, the colors in the source image are clustered and compared with the standard colors to produce candidate primary colors; then the region splitting and color filling are performed according to the parameters set by the user, such as polygon perimeter, splitting depth and dye search depth, to obtain a background pattern reflecting the color and texture information of the source image. In the foreground generation module, the grid points are constructed, and the source image is regionally grown according to the color difference and area threshold parameters set by the user, and then matched with the dictionary to obtain a foreground pattern that reflects the structure in pieces. The foreground and background are fused and pixelated to obtain a final digital camouflage pattern that can be used to camouflage in the source image area.

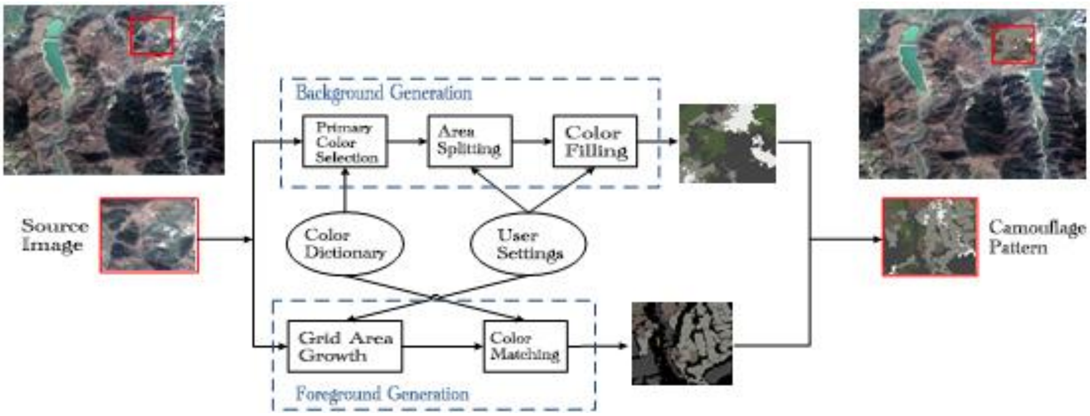


Figure 1: General block diagram of the method

2.1. Background generation

Humans are sensitive to color information, and the selection of colors must match the color distribution of the camouflaged area. In addition, in order to reflect the cluttered nature environment, a whole area needs to be randomly divided into blocks according to certain rules, filled with candidate colors randomly, and ensuring that adjacent blocks have different colors as much as possible. Therefore, the processing of the module is divided into three steps.

2.1.1. Primary color selection

Kmeans clustering algorithm is used to extract the color center of the camouflage area, and then search the most similar color in the dictionary as the candidate primary color.

(1) Collect the LAB space values of 30 standard camouflage colors of GJB90 and save them as a color dictionary $\{(L_i, A_i, B_i)\}_{i=0}^{29}$;

(2) Convert the source image from RGB space to LAB space and take out the L, A, B values of each pixel in the source image to form a sample set $\{(l_j, a_j, b_j)\}_j$;

(3) Clustering to 5 centers $\{(l_k^c, a_k^c, b_k^c)\}_{k=0}^{4}$ on $\{(l_j, a_j, b_j)\}_j$ with the K-means algorithm and a distance metric of Euclidean distance;

For each center k, the color with the closest Euclidean distance is searched on the dictionary and taken out, while the color is removed from the dictionary, and the color taken out is the candidate color $\{(l_k^s, a_k^s, b_k^s)\}_{k=0}^{4}$.

2.1.2. Region splitting

The image region is considered as a polygon, which is composed of vertices in turn, and the whole region can be partitioned into a controlled number of patches by randomly splitting the polygon and recursively.

(1) Initialization: the array P consisting of the four vertices (x, y) of the original image, the split depth d, the minimum polygon perimeter $c_{\min}$, the array Ps consisting of polygons is set to empty, followed by steps (2) to (4) noted as split (P, d);

(2) If the current perimeter of P is less than $c_{\min}$, or d = 0, then P is added to Ps;

(3) if not, calculate the lengths of each side of P and choose the 2 longest ones and take a point on them respectively;

(4) The new two points and the old points form 2 new polygons Pa and Pb, respectively, with d = d-1;

(5) Calculate split (Pa, d) and split (Pb, d), respectively, until the recursion stops;

(6) Finally, randomly displace the polygon elements in Ps.

2.1.3. Color filling

According to the color selected in 2.1.1, it is randomly filled in 2.1.2, and the colors of adjacent areas are different as far as possible. Therefore, a heuristic search strategy is used. First, the neighborhood of each polygon is obtained, then each polygon is traversed, its neighborhood is searched, and the adjacent colors of the neighborhood are searched and sorted to determine the next recursive position. When the complexity is small, this can ensure that the blocks in the homogeneous color region are destroyed to the greatest extent.

(1) initialization: candidate colors $\left\{\left(l_k^s, a_k^s, b_k^s\right)\right\}_{k=0}^{4}$, polygon array Ps, search depth b;

(2) iterate through all polygons in Ps, with the color of each set to its ordinal number;

(3) iterate through each polygon and obtain the neighborhood index of each polygon by selecting the color of the points outside the polygon, as follows:

① Starting from the first vertex of the polygon, set the current vertex as v1, the next vertex as v2 and the previous one as v3. The offsets xoff and yoff in x and y directions are initialized to 0;

② Judgment:

if v2.x < v1.x (horizontal coordinate), xoff+=1;

if v3.y > v1.y (vertical coordinate), yoff -= 1;

if v3.x > v1.x, xoff -= 1;

if v2.y > v1.y, yoff -= 1; if v2.y > v1.y, yoff -= 1;

if v2.x > v1.x, xoff -= 1;

if v3.y < v1.y, yoff += 1;

if v3.x < v1.x, xoff += 1; if v3.x < v1.x, xoff += 1;

if v2.y < v1.y, yoff += 1;

③ Take (v1.x+ xoff, v1. y+ yoff) as the coordinate point, and if it is within the image range, take out the pixel value (index) of that point and put it into that polygon neighborhood sequence;

④ Turn to the next vertex and go back to ① until all vertices are traversed.

(4) assigning a color to each polygon, as follows:

①Take a color at random from the candidate colors as 1 color index assigned to the current polygon;

②If b>0, iterate through all the neighbors of the polygon, and if the neighbor is not assigned a color, it is used as 1 candidate, and then search whether the color of its (candidate color) neighbors is the same as the color of the polygon, and record the number of consistent ones. Choose the candidate with the highest number, b=b-1, and recursively use it as the new polygon until the end;

③ Move to the next polygon and go back to ① until all polygons are traversed.

(5) Fill the whole image with the color of k=0 in $\left\{\left(l_k^s, a_k^s, b_k^s\right)\right\}_{k=0}^{4}$, and then fill it according to the color index of each polygon.

2.2. Foreground generation

The background generation algorithms in the previous section can provide highly irregular patches and blended colors that can approach the camouflaged environment in terms of texture details, yet how to simulate the realism of the environment in terms of scale structure has rarely been studied. The foreground generation algorithm proposed in this paper is based on the assumption of region similarity, using uniformly distributed grid points as seeds, which are grown and merged one by one, and then matched and filled with the color library. The algorithm is optimized in this paper so that it can be done within a grid cycle.

Foreground generation algorithm (set the image width and height as W and H respectively):

ISSN: 2664-9640

(1) Initialization: grid point index i=0, the grown region marker $M_{old} = [0]_{W\times H}$, the generated foreground image $I_F = [0]_{W\times H}$, the color difference threshold T_c , the region area ratio threshold T_r , the candidate color $\{(l_k^s, a_k^s, b_k^s)\}_{k=0}^{4}$;

(2) The region marker M is obtained by growing the source image, grid point i, T_c and T_r. The specific algorithm is:

①initialization: seed is the coordinate of grid point i, a and b , seedlist=[seed], $M = [0]_{W\times H}$;

②when seedlist is empty, then exit and return M, otherwise execute subsequent statements;

③seedlist the 0th element sd out of the column;

④take out the neighboring pixel p of sd, if p is not out of the image boundary, calculate the color difference (LAB Euclidean distance) D between sd and p pixels, if $D < T_c$ and $M[p] == 0$, then M[p]=1, put p into the tail of the seedlist. This is repeated until all neighboring pixels are traversed.

⑤ If the ratio of the number of M is 1 to the total number of pixels is greater than T_r , then exit and return M, otherwise go back to ②.

(3) The average color of M as 1 position is used to retrieve the most similar among the candidate colors and used to color these positions, resulting in I_{temp} , update.

$$
\begin{aligned}
& M_{old} += M - M * M_{old} \\
& I_F += I_{temp} \\
& \text{i+=1}
\end{aligned} \tag{1}
$$

(4) Go back to step (2) until all grid points have been traversed.

2.3. Fusion and post-processing

After the foreground processing, the background is marked as:

$$B = [1]_{W\times H} - I_F \tag{2}$$

The final result is the sum of the foreground and background under each marker:

$$I_c = I_B * B + I_F * M_{old} \tag{3}$$

The final digital camouflage pattern is obtained by applying speckling and pixelation to I_c .

3. Experiment

A slice of the aerial image is used as the source image and processed by the method in part 2. Figure 2. gives the background, foreground, and the final pixelated camouflage pattern.

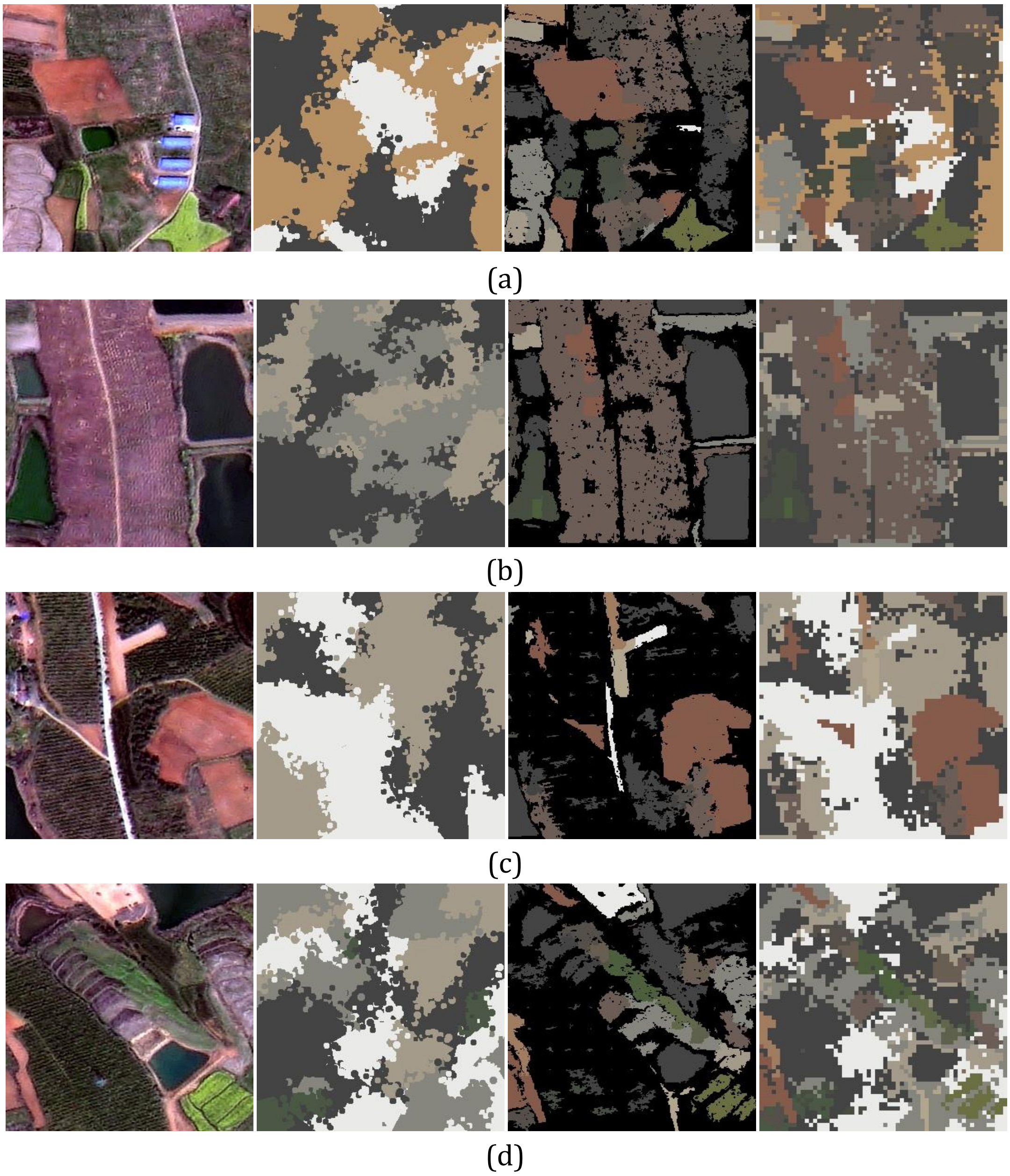


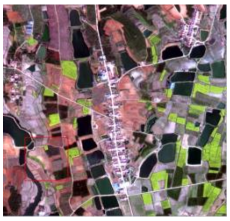
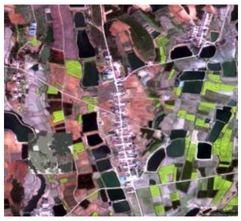
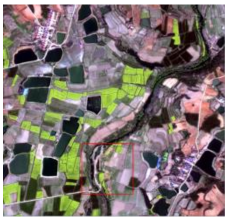
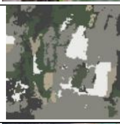
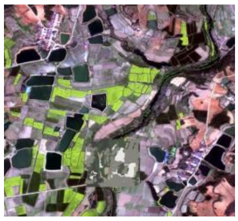
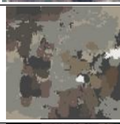
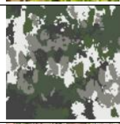
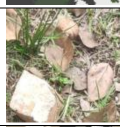
Figure 2: Results from the camouflage generation experiment (The source image, background texture, foreground structure and final result are shown from left to right)

As can be seen, the background generation algorithm module of this paper extracts the color information of the source image and simulates some details in reality with random patch textures, while the foreground generation algorithm reflects large areas of the same color of the source image, making the final result more similar to the source image.

To further demonstrate the artifactual effect of the method in this paper, the method is embedded in the software. Firstly, a panoramic image is selected, and the user is allowed to key a rectangular area in the system as the source image, which is processed by this method and then backfilled into the panoramic image, and the effect is shown in Table 1. Here not only remote sensing images are used, but also images of natural scenes taken by cell phones.

ISSN: 2664-9640

Table 1: Panorama camouflage experiment

Panorama (keying area is circled in red)	Source images and camouflage pattern	Backfill effect
	 	
	 	
	 	
	 	
	 	
	 	

As can be seen from the table, the camouflage pattern produced by this method fits better with the color and texture of the camouflage background, and achieves a better visual effect.

4. Conclusion

In this paper, a new camouflage pattern generation method based on natural background is proposed for the problems that the main camouflage methods ignore the image structure and the design of camouflage patches lacks randomness. The method consists of a background generation algorithm and a foreground generation algorithm, incorporating detailed texture

ISSN: 2664-9640

and macrostructure information of the camouflaged scene, and achieving better results in subjective visualization and practicality.

References

[1] Yu Jun, Shuang Xiao. Design method of imitation digital camouflage [J]. Journal of Applied Sciences, 2012, 30(4):331-334.

[2] Gao Yifan, Wang Yong, Zhang Yue. Method of generating digital camouflage from ambient colors in visible light background [P]. Beijing: CN109741246A,2019-05-10.

[3] Wei YX, Wang Y, Wang YY, Gao YF. Method of extracting theme color from multiple images and generating camouflage under visible light background [P]. Beijing: CN110717952A, 2020-01-21.

[4] Cai Yunchao, Xue SQ, Zhou ZY, Huang Y. Research on digital camouflage pattern generation method based on fractal Brownian model[J]. Journal of Military Engineering,2016,37(01):186-192.

[5] Xu Ying. Camouflage camouflage color selection algorithm based on background representative color extraction [J]. Optical Engineering, 2007, 34(1):100-103, 144.

[6] Qin L, Hu JH, Fu TQ. A new digital camouflage generation method [J]. Optical Technology Applications, 2013, 28(5): 33-35, 45.

[7] Xu Rongqiang. Design and research of some key algorithms for digital camouflage [D]. Beijing University of Technology,2016.

[8] Xu Shan. Research on digital camouflage texture design and camouflage evaluation model[D]. Hefei University of Technology, 2015.

[9] Bi Sheng, Zheng Lin, Wu Yaowu, Gu Fangrui. Research on computer-aided design of camouflage patterns[J]. Journal of System Simulation,2013,25(09):2164-2170.

[10] Nie Minghua. Camouflage scheme generation technology and system for visible wavelengths[D]. Zhejiang University, 2013.

[11] Wu Rui. Research on digital camouflage design method[D]. Nanjing University of Aeronautics and Astronautics,2016.

[12] Teng X,Zhang F,Yang CH,Zhao XJ,Li B. A digital camouflage camouflage generation method based on cyclic consistent adversarial network[J]. Computer Applications,2020,40(02):566-570.