Accepted Manuscript

Title: Interactive Evolutionary Colour Assignment and Proportioning for Camouflage Design with K-Means and Genetic Algorithm: A Case Study of Nigeria's Landscape

Authors: Taofikat Obasekore, Hammed Obasekore, Bo-Yeong Kang, Lee Jongseok

Manuscript number**: PCCC-2401-1279**

To appear in: Progresss in Color, Colorants and Coatings

Received: 21 January 2024

Final Revised: 18 April 2024

Accepted: 21 April 2024

Please cite this article as:

Taofikat Obasekore, Hammed Obasekore, Bo-Yeong Kang, Lee Jongseok, Interactive Evolutionary Colour Assignment and Proportioning for Camouflage Design with K-Means and Genetic Algorithm: A Case Study of Nigeria's Landscape, Prog. Color, Colorants, Coat., 17 (2024) XX-XXX.

Interactive Evolutionary Colour Assignment and Proportioning for Camouflage Design with K-Means and Genetic Algorithm: A Case Study of Nigeria's Landscape

T. Obasekore [1], H. Obasekore [2], B.Y. Kang [2], L. Jongseok [1*]

[1] Department of Fashion Design, Kyungpook National University, Sangju 37224, Republic of Korea

[2] Department of Robot and Smart System Engineering, Kyungpook National University, Daegu 41566, Republic of Korea

*Corresponding author: email: color@knu.ac.kr

Abstract

Natural camouflage, seamlessly blending animals with their surroundings, remains challenging for artificial counterparts. Some animals exhibit near-permanent camouflaging, a product of decades of genetic evolution with their environment. At the same time, chameleons and octopuses achieve the ideal desired instantaneous camouflaging, unlike the heuristic-based approach of the artificial camouflage design. To attain similar perfection seen in animals, an evolutionary approach to artificial camouflage pattern development is necessary. Developing nations, primarily adopting the camouflage patterns of their more developed counterparts, may find themselves at a disadvantage. This study proposes a Genetic Algorithm (GA)- -based approach to aid designers in developing countries in crafting effective camouflage. By parameterising heuristic development as a procedural texturing problem and evolving colour assignments iteratively, this approach aims to emulate the evolutionary process seen in nature. Using the K-means algorithm, genes are initialised based on background image colours, exploring factorial combinations to achieve optimal camouflage. With a maximum of 100 iterations and interactive feedback, the method addresses Nigeria's specific case and offers a faster development solution than developed nations' approaches. This evolutionary approach could revolutionise artificial camouflage development worldwide.

Keywords: Camouflage, Colour assignment, Genetic Algorithm, K-Mean, Interactive Design, Procedural Texture.

1. Introduction

Researchers have defined camouflaging as using disruptive contour on a target to blend it with the background, making it harder to detect or hit [1, 2]. This idea of camouflaging occurs effortlessly in nature. It has evolved for some animals, mainly to prey on other animals or avoid being preyed upon. Depending on the type of animal, the frequency of blending with the background environment varies from permanent to seasonal to weeks to minutes to instantaneous. Thus contributing to one of the natural strategies for animal survival in their habitat [3]. This blending with the background without being detected resonates with the military's desire for stealthiness. During and after the Second World War, various camouflage schemes were used for aircraft and ground vehicles in different theatres of war [4]. Beyond the world wars, camouflage patterns have achieved a successful track record in modern warfare.

Despite the successes, the ideal desirable military camouflage is yet to be achieved [5], for instance, the chameleon and octopus's adaptive skin patterns and colours, the abstract desire for camouflage that makes total disappearance. As such, several countries have continuously invested heavily in research and development toward realising patterns suited for their geographical location. Many developing countries fall short in this by only adopting the camouflage patterns of either their colonial master or other developed countries, as seen in the case of Nigeria's military personnel wore the British desert Disruptive Pattern Material (DPM) camouflage in 2011 in the Darfur region of Sudan, and vertical lizard camouflage pattern influenced by the French and Portuguese designs during the Nigerian Civil War [6, 7]. These adopted camouflage patterns are necessarily not optimal for the country. However, the scientific resources, rigorous development process, and perhaps others might be the inhibiting factor. Given the current rise in world temperature, which is resulting in a shift in geographical features, there are speculations that previous military camouflages might become ineffective, thus necessitating the rapid development of new military camouflages worldwide. Over the years, the military camouflage research and development process has evolved to the following [8].

(1) the collection of earthly samples of background environments that are widespread across regions,
(2) the extraction of n numbers of dominating colours via laboratory processes,
(3) designing of seamless contours
(4) assigning the extracted colours to the contours
(5) testing designed camouflage patterns against backgrounds
(6) getting human eye-tracking feedback.

Recent advances in image processing and computer vision have presented the possibility of extracting both high- and low-level features from images like scenes and colour spectrums [9, 10]. This has led researchers to opt for collecting quality images and nondestructively extracting desired features using various algorithms [5], contrary to collecting earthly samples and using a laboratory extraction process afterwards, specifically, colourimeter, as demonstrated in [8]. The most obvious advantage of using images instead is the reusability and the non-destructiveness of the process. The other advantages include the speed of achieving desired results; in the presence of good computing resources, image-based samples achieve results faster compared to earthly samples in the laboratory and, likewise, the ability to extract a spectrum of colours at once, which will require several measurements for a colourimeter to obtain a representative colour value [11].

The design of seamless contours for the camouflage patterns lies in the heuristic creativity of the designer [8] on disruptive contour design. This is usually achieved with vector software such as Inkscape. Jong proposes extracting disruptive contour from high-resolution images of granite and pine trees, oak tree bark, and their colony using edge detection techniques in image processing software [8]. However, this manual heuristic design method is challenging to parametrise for algorithmic optimisation. Genetic algorithm (GA) is a popular algorithm that has been proven to optimise heuristics across multiple domains like robot path-planning and design [12-14]. Reynolds [14] proposed using procedural texture synthesis for camouflage design parameterisation and genetic programming (GP) as the optimisation technique. However, seamlessness is not considered for the camouflage generated, perhaps because there was no motive for transferring onto fabric.

Generally, GA is an example of evolutionary computation inspired by Darwinian selection, mutation, and crossover concepts. It is a metaphorical representation of the idea of genetic reproduction in biology. It begins with initialising the population of candidate solutions (often called individuals or chromosomes). Each individual represents a set of features relating to the problem. This individual evolves over successive generations to better fit a fitness function, resulting in a fitness value representing the solution's quality. The higher the fitness value, the higher the quality of the solution. The best individuals are selected to constitute new parents. The parents mate to generate offspring expected to bear better quality than their parents, thus suppressing the low-quality individuals. However, this generated offspring from the selected parents only has the characteristics of its parents and without changes, including the drawbacks in its parents. Crossover and mutation prevent offspring from being identical to their parent, achieved by injecting randomness during mating and gene levels. This continues until specific stopping criteria, such as acceptable solution or number iterations, are met. Figure 1 shows the complete flow chart that includes the sequence of intermediate processes.

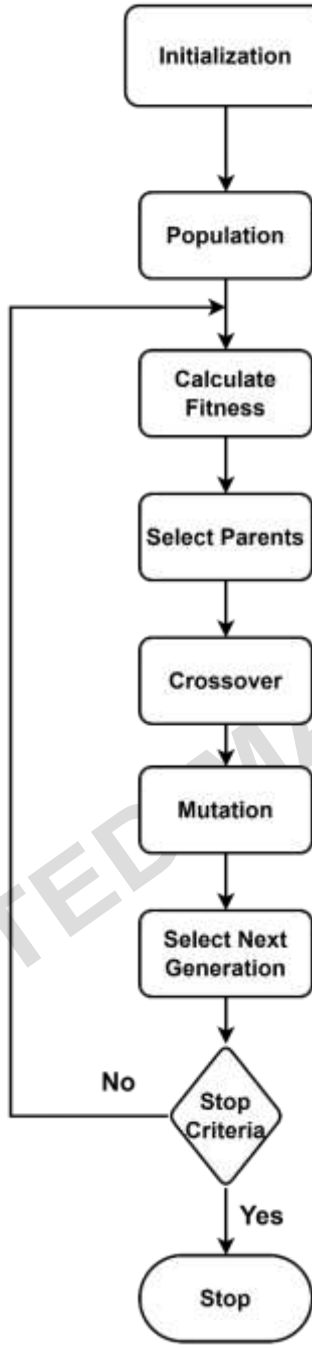


Figure 1. Flow chart illustrating the processes in GA.

Procedural texturing techniques are dynamic and parametric (with underlying mathematics and algorithms), which means that designers generate complex and adaptive patterns that can be mapped onto a shape as a texture. It often procedurally utilises various noise algorithms like Perlin, fractal, Voronoi, and Worley Noises in combination with mathematical operations to achieve a natural look at the resulting pattern [15] improved upon Perlin noise, providing smoother gradients and better continuity, enhancing the visual quality of procedural patterns. Lagae and Dutré, 2008 demonstrated the application of Voronoi patterns in disruptive applications [16]. Procedural texture can either be defined programmatically or visually. For instance, Reynolds [14] defines programmatic texture synthesis as modules of program snippets using Open BEAGLE [17], like the definition available in Three.js, a cross-browser JavaScript library. At the same time, applications like Blender (USA) provide programmatic and visual pallets to connect procedural texture elements as nodes and edges.

In computer graphics, seamlessness is usually handled digitally. However, aside from adding a seam to strategically locate the seamline, the textured shape is a significant determinant of the nature of the seam. For instance, a geometrically unwrapped open-ended cylinder will only amount to seamlessness in one axis (either horizontally or vertically). Whereas geometrically unwrapping torus results in seamlessness in all directions, one axis could be more stretched [18], as shown in Figure 2. However, Vincent [19] presents a unique formulation to achieve a square flat torus that remains undistorted when unwrapped. This work leverages the continuity of a flat torus to achieve seamlessness. That is, the procedural texturing is generated on a torus (with minimal distortion); therefore, the designer does not need to worry about the seamlessness of generated camouflage. Adopting this technique to achieve seamlessness is novel to this domain, as previous work tends to achieve seamlessness manually.

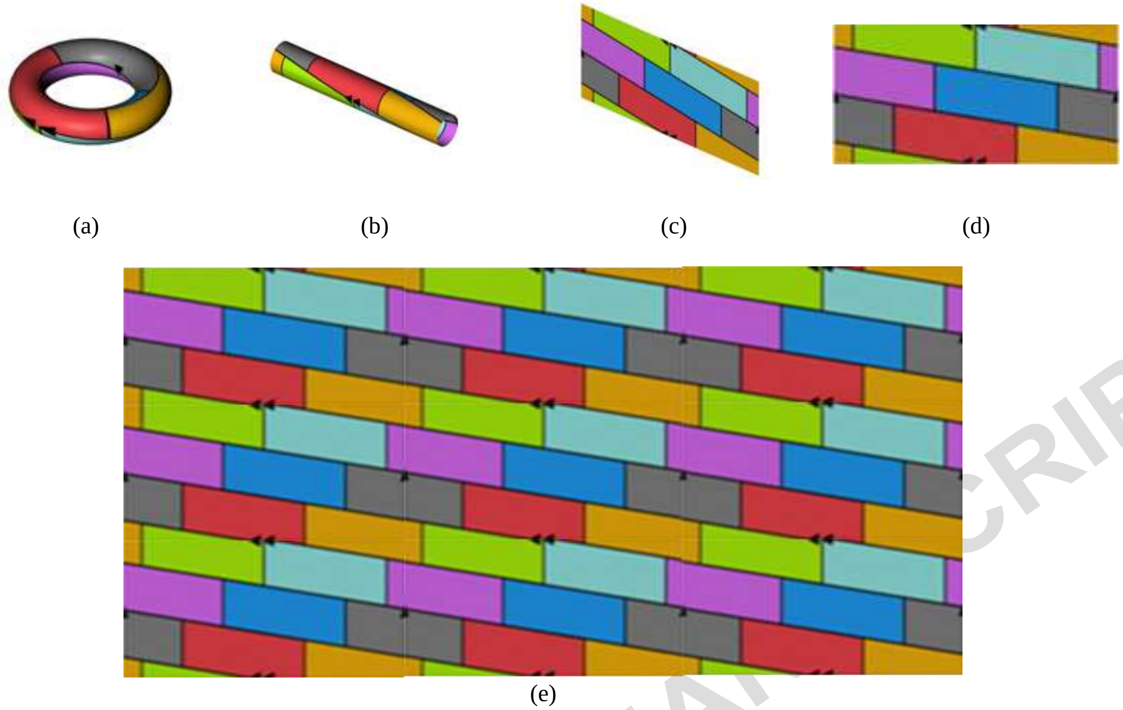


Figure 2. illustrating the unwrapping procedure to elucidate how it translates to seamlessness.

Like other processes, assigning colours to the camouflage contours has often been a designer heuristic. Unfortunately, the unforgiving part is the delayed feedback on whether the resulting colour assignment is perfectly concealed in the background, as there are k factorial ways (k!) to assign k colours. A near-natural way to assign k colours while giving the designer some control and active feedback is an interactive evolution proposed in [14]. The interactiveness is formulated as an interface that overlays each of the generated camouflage patterns (with assigned colours), called populations, against the representative background for the designer to either rate or filter based on blending. In [14], the interaction proposed is a series of mouse clicks to mark the five (5) most conspicuous populations for deletion out of the ten (10) presented. The properties (procedural textures), called genes that make up the marked and the remaining surviving populations, are then crossed over and mutated to reproduce new offspring that replace lower-fitness individuals. These newly generated populations constitute the new evolution. The cycle of interaction and evolution continues until the resulting camouflage pattern achieves a certain degree of blending. Typically, in [14], this satisfactory concealment was observed after 1000 evolutions, that is, 5,000 mouse clicks.

Therefore, the following is our new proposed approach that replaces the earlier conventional approaches.

(1) Collecting high-resolution images of representative background

(2) Algorithmic extraction of low-level features

(3) Parameterized designing of seamless contours

(4) Evolutional-based assignment of features to the contours

(5) Realtime interactive camouflage pattern design testing against the background

Therefore, this work proposes an assistive mechanism that is GA-based to evolve camouflage design colour assignment and proportioning as a designer designs interactively. The assistive pipeline's composition starts with features extracted from high-resolution images of the desired representative background to blend using the K-Means algorithm and interactively evolve using a GA parametrised as a procedural texturing problem. This work is specifically novel for the end-to-end interactive camouflage design, resulting in a seamless design that can be directly printed on fabric.

This paper is divided into four sections. Section 1 begins with the introduction of the motivation behind the research. The research methodology and the materials are introduced in Section 2, which briefly describes the data source, the feature extraction, procedural texturing, algorithms and the experimental setup. Section 3 discusses the results of the dominant colour extraction with the K-means algorithm, the resulting camouflage design from GA, and the designed camouflage's printing. Finally, Section 4 presents the conclusions and recommends possible directions for future studies.

2. Experimental

This section details the implementation of the proposed replacement to the earlier conventional approaches mentioned. Starting with the collection of representative samples from Nigeria’s environment. Then, a machine learning algorithm was used to extract five (5) dominating colours from each sample. Afterwards, a detail of the procedural synthesis-based camouflage generation was generated with Blender software, followed by the torus-based seamlessness and, finally, the GA-based evolution of camouflage design colour assignment and proportioning.

2.1. Sample Collection

We propose collecting high-resolution images rather than earthly samples like tree bark, leaves, and soil from the representative areas. This study draws a case study on the Nigerian landscape; therefore, the representative areas are the geographical distribution of Nigeria. Geographically, Nigeria is in the West Africa region (between latitudes 4.1027° and 13.7994° N and longitudes 2.7991° and 15.1005° E) with 36 states widespread across the six (6) geopolitical zones, namely, North-Central, North-West, North-East, South-South, South-West, and South-East. Climatically, she is divided into Sahel, Savanna, Tropical Rainforest, and Coastal, starting from the North to the South, as shown in Figure 3. We collect high-resolution images across the zones mentioned above. Specifically, these selected locations are the potential military drill sites across the geopolitical zones in Nigeria. For instance, the National Youth Service Corps (NYSC) Campsites are military-controlled camps across 36 states. All Nigerian graduates below the age of thirty (30) years have been on strategic deployment into any of the states since 1973 to engage in three (3) weeks of military drill as an avenue to facilitate the reconciliation, reconstruction, and rebuilding of the nation after the civil war [20, 21]. Some of these high-resolution images are sourced from these campsites and beyond. Depending on the region, the images span from tree features like roots, barks, branches, and leaves to grasses of varying colours. Figure 4 shows a few samples with homogeneous features.

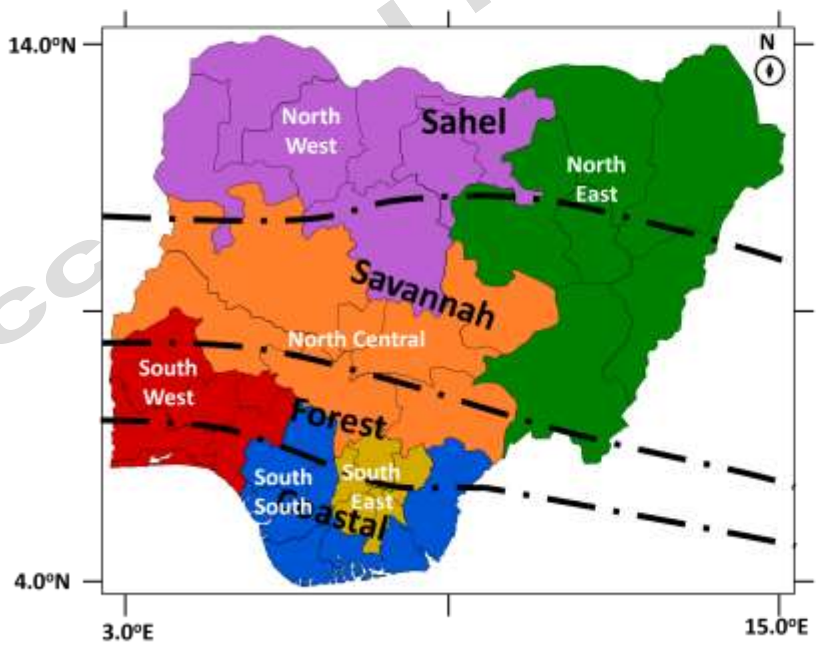


Figure 3: Map of Nigeria's geopolitical and climatic distribution.

Figure 4: Samples of sourced high-resolution images that spans from tree features like roots, barks, branches, and leaves, to grasses.

2.2. Feature extraction and extraction of dominating colours

Replacing earthly samples with high-resolution images negates the need for experimental laboratory procedures for extracting colour constituents, which was a significant overhead due to the need to await the result. However, we extract low-level features from each high-resolution image as an alternative. RGB channels were vectorised for each image, as illustrated in Figure 5, and clustered into K dominant colours using an unsupervised machine learning algorithm called K-Means, as shown in Figure 5. The K-means algorithm works by grouping similar data points into a K-predetermined number of clusters, where each cluster is represented by its centre, called a centroid. In this context, given a background image of height (H), width (W) and Channel (C), in this case for coloured (Red, Green, and Blue) image C =1, 2 and 3, the resulting vectorised image feature per channel is a H*W vector. For example, the input-coloured background image in Figure 5 is 3072 × 4080, and the resulting vector per channel is a [12,533,760 x 3]. The K-means problem is to group this feature into K clusters. For instance, to design a 5-colour military camouflage, the K = 5; therefore, the result is a spectrum of 5 dominant colours in terms of the RGB values and frequency of occurrence, as shown in Figures 6 and 7.

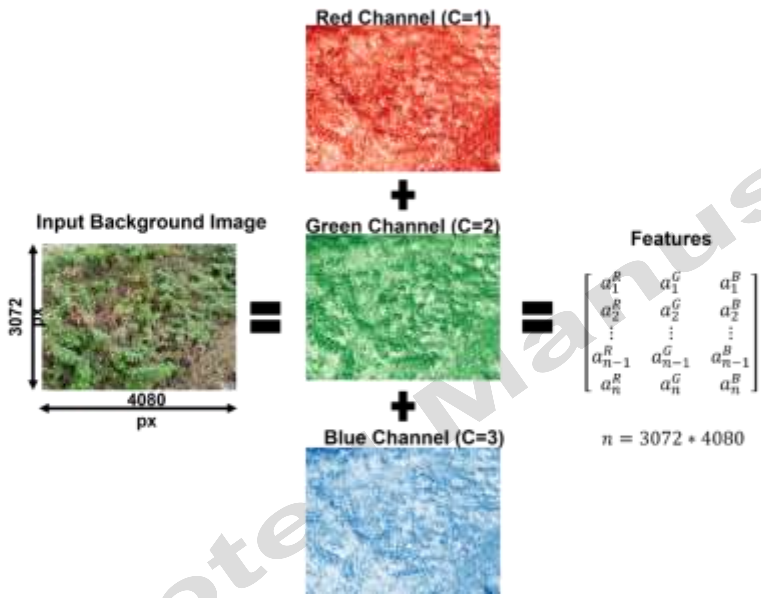


Figure 5. Illustration of the splitting of the RGB and vectorisation of the input background image.

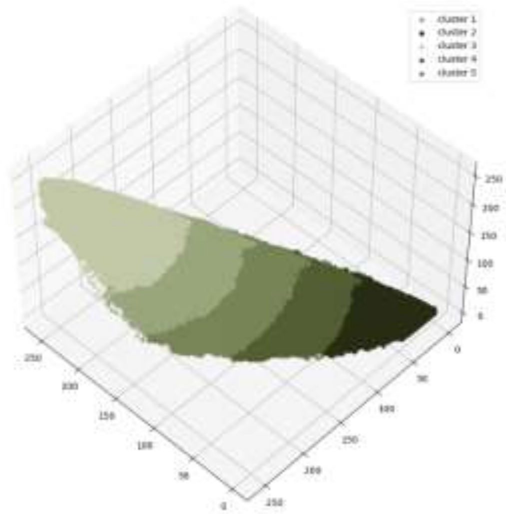


Figure 6. Clustering the vectorised feature into dominant colours using K-Means (K = 5).

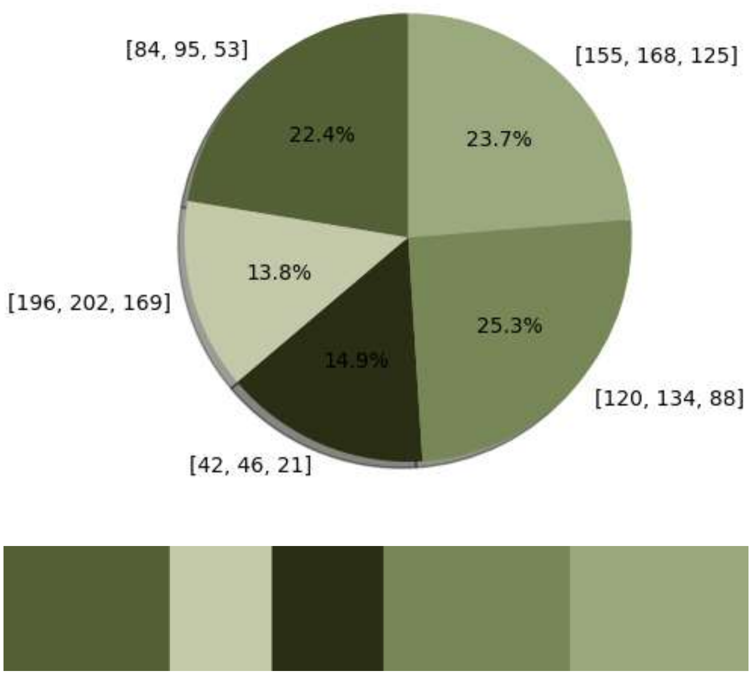
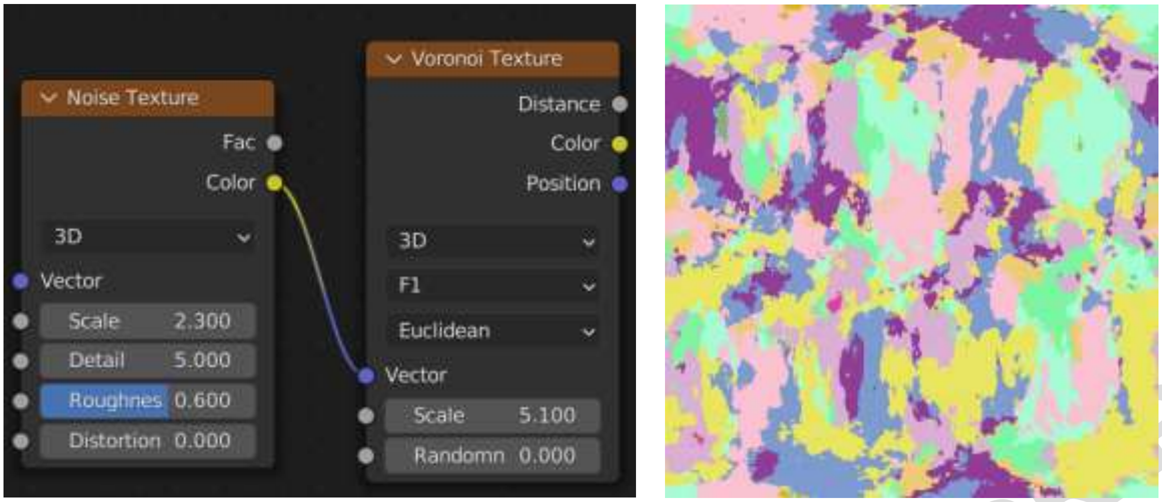


Figure 7. Illustration of spectrum of 5 dominant colours in term of the RGB values and frequency of occurrence in the input background image

2.3. Generation of camouflage

Aside from the need to draw camouflage contour, designers also ensure seamlessness. Rather than using vector software to achieve this manually, we adopt procedural texturing, thus enabling the parametrisation of the camouflage pattern design while still giving the human designer flexibility. In this work, parametrisation is achieved via tuning the parameters for noise and Voronoi texture nodes that evaluate a fractal Perlin and Worley noise, respectively. In this work, we leveraged Blender software's flexibility in designing procedural textures visually and programmatically. The desired pattern is carefully designed by visually tuning the parameters of the abovementioned noise algorithms. At the same time, the colour proportion and order of colour assignment are programmatically assigned as the genes of the GA. The details of the input-output for both noises are provided in Blender online documentation [22]. For example, setting the parameters in Figure 8a and feeding the colour – output of the noise texture node into vector – input of the Voronoi texture node and then rendering on torus shape results in the colour-segmented seamless patterns in Figure 8b.



(a) (b)

Figure 8. Connecting (a) colour – output of noise texture node into vector – input of Voronoi texture node to result in (b) colour-segmented patterns.

2.4. Genetic algorithm-based evolution of the camouflage

Even after the colour-segmented pattern has been carefully designed, the arrangement and the proportion of colours in the patterns are crucial. Manually tweaking this arrangement and proportion is simpler for a few colour combinations. However, as the number increases, the manual adjustment becomes forbiddingly unrealistic, thus requiring a more natural and optimal approach. For instance, for a two-colour camouflage, the possible ways of arrangement are two factorial (2), and the proportion can be varied infinitely in the range of 0 – 1 between the two colours. However, by discretising into a step of 0.1, 11 possible complementary values can be assigned between the two colours, thus making a maximum of 22 trials.

Nevertheless, for a typical five-colour camouflage, 120 ways of arrangement exist. A maximum of 1200 trials is possible when discretised in a similar version. Of course, discretisation is an approximation often accompanied by a problem of loss of details. In this work, the GA is employed to search and optimise the arrangement and proportion of each colour for the k-colours camouflage. Although GA can evolve the whole procedural texturing, the resulting camouflage pattern evolved in previous work [14] is entirely fitted for the given background image. Therefore, we opted to evolve only the texture node for colour ordering and proportioning while giving some control to the designer's expertise.

The GA population is initialised as the proportion of factorial of K number of colours clustered by the K-means algorithm from the background image. The GA then uses a given fitness function to evaluate each individual in this population. The lower-fitness individuals in the population are then replaced by new offspring arising from the crossover and mutation of the more viable individuals. This technically implies rearrangement and re-proportioning of the colours based on the better ones. Over time, the population is expected to contain the populations that improve fitness.

2.5. Interactive Evaluation of the Camouflage

The role of the fitness function in this work is played by a human observer who visually compares the quality of the evolved colour assigned to the camouflage pattern and how they best blend with the background. This is achieved via a web-based user interface (UI) where the top ten individuals are randomly overlaid on the background image, as shown in Figure 11. Immediately, the individuals are drawn, and incremental counting starts for each individual. The observer's task is to locate and click the patterns in the order of their conspicuousness. Once a particular individual is clicked, its corresponding incremental counting stops, thus translating to low fitness for the first clicked individual and highest fitness for the last clicked individual. Afterward, the observer clicks the evolve generation button to invoke the GA to use the new fitness to evolve the next individual through the earlier-mentioned crossover and mutation. The new top 10 individuals from the next generation is then presented to the observer through the UI for evaluation. This iterative process continues until a satisfactory blending is achieved.

Figure 9. The interactive user interface for the observer to evaluate the blending of the top 10 individuals.

2.6. Experimental setup

This work opted for a 5-colour camouflage pattern design. Therefore, the K = 5 for the K-Means algorithm. The selected high-resolution background images have varying dimensions with homogeneous content. The primary computational resource used for this experiment is an Intel(R) Corei7 @ 2.80GHz (RAM 16GB) running Windows 10. We utilised open-source software like Blender 3.0 for procedural texturing and Python 3.7 as the programming language. The Python-specific library for GA, K-Means, image processing, and web backend are the implementations provided by PyGAD [23], scikit-learn [24], OpenCV [25], and Flask [26], respectively. The parameter for the GA includes the fitness function, which is a function that computes the normalisation of the fitness provided by the human observer; the number of individuals in a population is set to 120, and the number of genes per individual is 5. On average, it took a maximum of 100 generations to achieve a satisfactory camouflage pattern.

3. Results and Discussion

This section presents the results of applying the described methods to the Nigerian environment. Initially, six high-resolution images were carefully selected, each representing a geopolitical zone, sourced from various NYSC camps and military barracks across the nation. Using the proposed k-means algorithm, five dominant colours were extracted from each image, resulting in 30 colours across all regions. These colours and their corresponding Red-Green-Blue (RGB) values and dominance proportions are detailed in Table 1.

Table 1. The 30 extracted dominant colours.

S/N	Sample	Region	Colour-Index	R	G	B	RGB	% of occurrence per image
1	1	SW	K = 1	160	173	94		26.69
2			K = 2	25	28	6		16.30
3			K = 3	197	208	146		16.59
4			K = 4	78	87	25		18.42
5			K = 5	123	136	55		21.99
6	2	NE	K = 1	169	152	144		27.26
7			K = 2	52	38	31		8.91
8			K = 3	104	86	77		21.08
9			K = 4	138	120	111		28.91
10			K = 5	215	191	173		13.84
11	3	NC	K = 1	208	198	184		14.00
12			K = 2	153	129	110		31.49
13			K = 3	120	98	84		20.59
14			K = 4	179	160	140		27.05
15			K = 5	67	55	49		6.87
16	4	SE	K = 1	154	167	124		24.01
17			K = 2	41	45	21		14.55
18			K = 3	195	201	168		14.20
19			K = 4	119	133	87		25.21
20			K = 5	83	94	52		22.03
21	5	SS	K = 1	81	92	21		8.89
22			K = 2	239	241	236		28.21
23			K = 3	203	184	140		11.55
24			K = 4	166	161	79		22.40
25			K = 5	126	146	43		28.96
26	6	NW	K = 1	192	152	116		53.07

27			K = 2	101	82	53		7.68
28			K = 3	174	182	189		22.97
29			K = 4	142	119	94		9.39
30			K = 5	46	39	22		6.89

Analysis of the extracted colours revealed distinct trends among the regions. The South-West (SW), South-South (SS), and South-East (SE) regions predominantly exhibited shades of green, indicative of their coastal and tropical rainforest climates. Conversely, the Northeast (NE), North-Central (NC), and North-West (NW) regions were characterised by shades of brown, reflective of their desert and savannah landscapes, with occasional variations of white likely attributed to skylight dilution.

The time required for colour extraction varies based on image resolution and computational power. Table 2 illustrates the extraction times for images of different resolutions across two different PCs. Notably, our proposed approach extracted five dominant colours from a 3072 x 4080 RGB image in a maximum of 3 minutes and 16 seconds, significantly outperforming the 12-hour timeframe required by the colourimeter technique [8].

Table 2. Duration taken to extract 5 colours based on PC configuration and image resolution.

	Image Resolution (H x W x RGB)	Intel(R) Corei7 @ 2.80GHz (RAM 16GB) Windows 10	Intel(R) Xeon(R) @ 2.20GHz (RAM 12.6GB) Ubuntu 22.04.2 LTS
Sample 1	3072 x 4080	196.87 secs	105.78 secs
Sample 2	4080 x 3072	152.18 secs	103.22 secs

Sample 3	3072 x 4080	196.00 secs	128.85 secs
Sample 4	4080 x 3072	140.50 secs	105.12 secs
Sample 5	4080 x 3072	140.60 secs	121.54 secs

Additionally, achieving a satisfactory camouflage design from a background image involved a maximum of 100 iterations of GA-based evolution, with up to 5 hours spent per design. Figure 10a depicts a background image with dominant colours, while Table 3 presents the top 10 individuals from the initial generations. After 50 iterations and interactive feedback, Figure 10b showcases one of the resulting satisfactory camouflage designs, validated through application on a 3D army dress (Figure 11) and subsequent heat transfer printing onto fabric (Figures 12a and 12b).

The fabric printing process utilised EPSON SureColor SC-F540 and JC-7B Semi-automatic hydraulic double stations heat press machine. Following colour calibration with Pantone, the seamless camouflage design was printed onto paper and transferred onto the fabric using the heat press machine at 200°C for 60 seconds, resulting in the final printed camouflage design on fabric (Figure 12b).

(a) (b)

Figure 10: (a) Input background image and (b) Output camouflage design from satisfactory generation

Table 3: Sample of intermediate evolutional generation.

Generation 0 (Top 1 -10 individual)		Generation 1 (Top 1 -10 individual)	
			
			
			
			
			

Figure 11: Verifying the blending by applying the designed camouflage on a 3D military uniform with the source image as the background.

(a) (b)

Figure 12. Seamlessly tiled camouflage–like pattern generated

4. Conclusions

This study introduces a Genetic Algorithm (GA)-based approach tailored to support camouflage research and development in developing countries, offering a customizable alternative to existing patterns. Notably, it addresses the anticipated challenges of climate change, which may render current camouflage ineffective.

This approach accelerates development by utilizing high-resolution image samples from diverse environments, procedural texturing, and GA-assisted colour arrangement and proportioning. Analysis of six samples from Nigeria's geopolitical regions demonstrates promising outcomes, including the correlation between extracted colours and regional climatic features, expedited development compared to conventional methods, particularly in colour extraction, and the ability to generate GA-based camouflage designs within five hours. Moreover, to facilitate accessibility for developing countries, all techniques and software utilized are freely available, including the PyGAD library for GA, the scikit-learn library for K-means clustering, and Blender software for procedural texturing and rendering. This study emphasizes the importance of collecting undiluted, homogeneous image samples and underscores the necessity for rigorous field testing before deploying evolved camouflage designs in combat scenarios.

5. References

[1] Lin CJ, Chang CC, Lee YH. Evaluating camouflage design using eye movement data. Appl Ergon 2014; 45:714-23. https://doi.org/10.1016/J.APERGO.2013.09.012.

[2] Stevens M, Merilaita S. Defining disruptive coloration and distinguishing its

functions. Philos Trans R Soc B Biol Sci. 2008;364:481–8. https://doi.org/10.1098/RSTB.2008.0216.

[3] WALLACE AR. The Colours of Animals. Nat. 1890 421082 1890; 42:289-91. https://doi.org/10.1038/042289a0.

[4] Camouflage - Wikipedia 2022. https://en.wikipedia.org/wiki/Camouflage (accessed November 30, 2022).

[5] Xiao H, Qu Z, Lv M, Jiang Y, Wang C, Qin R. Fast Self-Adaptive Digital Camouflage Design Method Based on Deep Learning. Appl Sci 2020, Vol 10, Page 5284 2020;10:5284. https://doi.org/10.3390/APP10155284.

[6] Larson EH. Camouflage: Modern International Military Patterns. Pen \& Sword Books Limited; 2022.

[7] Borsarello JF, Palinckx W. Camouflage Uniforms of Asian and Middle Eastern Armies. Schiffer Publishing, Limited; 2004.

[8] Jong Seok L. 한국자연환경과융합한위장패턴디자인연구. 한국과학예술융합학회 2022; 40:305-17. https://doi.org/10.17548/KSAF.2022.12.30.305.

[9] Mutlag WK, Ali SK, Aydam ZM, Taher BH. Feature Extraction Methods: A Review. J Phys Conf Ser. 2020;1591:012028. https://doi.org/10.1088/1742-6596/1591/1/012028.

[10] Zhou X, Gong Y, Meng X-M, Hasan H, Shafri HZ, et al. Extraction of High-level and Low-level feature for classification of Image using Ridgelet and CNN based Image Classification. J Phys Conf Ser. 2021;1911:012019. https://doi.org/10.1088/1742-6596/1911/1/012019.

[11] Goñi SM, Salvadori VO. Color measurement: comparison of colorimeter vs. computer vision system. J Food Meas Charact. 2017;11:538-47. https://doi.org/10.1007/S11694-016-9421-1/FIGURES/8.

[12] Senniappan Karuppusamy N, Kang BY. Minimizing airtime by optimizing tool path in computer numerical control machine tools with application of A* and genetic algorithms. Adv Mech Eng. 2017;9. https://doi.org/10.1177/1687814017737448/ASSET/IMAGES/LARGE/10.1177_1687814017737448-FIG8.JPEG.

[13] Lee J, Kang BY, Kim DW. Fast genetic algorithm for robot path planning. Electron Lett. 2013;49:1449-51. https://doi.org/10.1049/EL.2013.3143.

[14] Reynolds C. Interactive Evolution of Camouflage. Artif Life 2011;17:123–36. https://doi.org/10.1162/ARTL_A_00023.

[15] Perlin K. An image synthesizer. ACM SIGGRAPH Comput Graph. 1985;19:287-96. https://doi.org/10.1145/325165.325247.

[16] Lagae A, Dutré P. A Comparison of Methods for Generating Poisson Disk Distributions. Comput Graph Forum. 2008;27:114-29. https://doi.org/10.1111/J.1467-8659.2007.01100.X.

[17] Gagné C, Parizeau M. Genericity in evolutionary computation software tools: principles and case study. Int J Artif Intell Tools. 2006; 15:173-94.

[18] Chen K-T. It’s a wrap! visualisations that wrap around cylindrical, toroidal, or spherical topologies 2022. https://doi.org/10.26180/20723092.v1.

[19] Borrelli V, Jabrane S, Lazarus F, Thibert B. Flat tori in three-dimensional space and convex integration. Proc Natl Acad Sci USA. 2012; 109:7218-23.

https://doi.org/10.1073/PNAS.1118478109/ASSET/418C20BF-C041-494E-9FF8-B178CA10B66D/ASSETS/GRAPHIC/PNAS.1118478109EQ63.GIF.

[20] Nwachukwu CE, Olufunmilayo EO, Chiroma GB, Okoye CF. Perception of National Youth Service Corps (NYSC) among corps medical doctors in Nigeria: a cross-sectional study. BMC Med Educ. 2023; 23:1-9. https://doi.org/10.1186/S12909-023-04135-1/TABLES/5.

[21] Marenin O. Implementing deployment policies in the national youth service corps of nigeria. Http://DxDoiOrg/101177/0010414090022004002 1990; 22:397-436. https://doi.org/10.1177/0010414090022004002.

[22] Community BO. Blender - a 3D modelling and rendering package. Blender Found 2018. https://docs.blender.org/manual/en/latest/index.html (accessed October 6, 2023).

[23] Gad AF. PyGAD: An Intuitive Genetic Algorithm Python Library. ArXiv 2021;abs/2106.0.

[24] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in {P}ython. J Mach Learn Res 2011; 12:2825-30.

[25] Bradski G. The OpenCV Library. Dr Dobb's J Softw Tools 2000.

[26] Grinberg M. Flask web development: developing web applications with python. "O'Reilly Media, Inc."; 2018.