

Adaptive electronic camouflage using texture synthesis

Narek Pezeshkian*, Joseph D. Neff

Space and Naval Warfare Systems Center, Pacific
San Diego, CA 92152

ABSTRACT

Camouflaged robots and leave-behind surveillance sensors are desirable in information, surveillance and reconnaissance operations to minimize the chances of detection by the enemy. Today's camouflaging techniques involve nets and painted patterns that are fixed in color and geometry, limiting their use to specific environments; a fact illustrated by numerous changes in military uniforms designed to fit the latest operating environment. Furthermore, nets are bulky and can interfere with the operation or use of a robot or leave-behind sensor. A more effective technique is to automatically adapt surface patterns and colors to match the environment, as is done by several species in nature. This can lead to the development of new and more effective robotic behaviors in surveillance missions and stealth operations. This biologically-inspired adaptive camouflage can be achieved by a) sampling the environment with a camera, b) synthesizing a camouflage image, and c) reproducing it on color electronic paper – a thin low-power reflective display – that is part of the outer enclosure surface of the robot or device. The focus of this paper is on the work performed for the first two steps of the process. Color-camouflage-synthesis is achieved via modifications made to a gray-level texture-synthesis method that makes use of gray-level co-occurrence matrices. Statistic equality in color-proportion is achieved with the use of conditional probability constraints.

Keywords: adaptive camouflage, color texture synthesis, GLCM, spin-flip, robot, unmanned system

1. INTRODUCTION

One of the most basic instincts in nature is to hide and not be seen. This gives predators the element of surprise when hunting for prey, significantly increasing the success-rate of catching food, ensuring their survival until the next hunt. For prey, on the other hand, camouflaging significantly increases the chances of evading predators, ensuring their survival until the next encounter. Many animals (predator or prey) must be able to hide in relatively open fields because a shelter is not always available. To do so, nature provides camouflaging as a means of blending in with the background and not be seen. Because camouflaging has proven to be extremely effective, most species possess some form of camouflage. Examples are shown in Figure 1.



Figure 1. A mixed variety of camouflage patterns found in moths (left), reptiles (center), mammals (right).

*narek.pezeshkian@navy.mil

The effectiveness of animal camouflage appears to be dependent on the diversity of its natural habitat. The camouflage of an animal may be highly adapted and extremely effective if the general background of its habitat has little diversity. In contrast, the camouflage of an animal living in a diverse habitat area may be more effective in some environments and less so in others. There are a few species, however, that possess the ability to change or adapt their camouflage based upon the environment. Octopuses and cuttlefish are two such examples and are shown in Figure 2.



Figure 2. Examples of adaptive camouflaging capability of cuttlefish (left two images) and octopus (right two images).

Because nature has proven that camouflaged animals can effectively hide or attack, it is no accident that the military has followed suit with a rich history of camouflaging ships, aircraft, vehicles, uniforms, weapons, and many other devices (Figure 3). Military camouflage colors and patterns have evolved throughout history to improve their effectiveness, with each variant designed for a specific environment. As a result, the camouflage pattern is only effective in areas where the local background remains relatively unchanged, much like the habitat area of most species in nature. For a military system that is capable of operating in a wide-variety of environments, its camouflage must adapt accordingly. The lack of this capability forces the military to periodically alter camouflage patterns and colors to fit the theater of operation. A testament to this fact is the changing military uniforms that are designed to fit the latest operating environment.



Figure 3. Fixed camouflage pattern examples used by the military (from left to right) for arctic, riverine, mountainous, woodland, and generic green or tan environments.

Ground troops are not the only military assets in need of camouflaging. Robots have become valuable assets for the military and their use in wartime operations has grown considerably over the past several years. As robots continue to evolve technologically, they will be used in more complex missions. For example, a robot may be used for information, surveillance, and reconnaissance (ISR) missions and be required to frequently change its observation location. Since the local background of one location may be drastically different than that of another, the ISR robot must be able to adapt its camouflage accordingly. If using leave-behind sensors scattered to monitor a wide area the local background for each leave-behind sensor may vary considerably. The operational environments can change as well, even if their local background has few variations. For example, woodland, urban, arctic, and desert environments have drastically different

backgrounds from one another. In all cases, the ISR robot and leave-behind sensor must adapt their camouflage patterns accordingly to reduce their visual signature and increase survivability.

The remainder of the paper is structured as follows: section 2 gives background information regarding attempts to achieve invisibility and adaptive camouflaging; section 3 will discuss the proposed approach taken to achieve adaptive camouflaging; section 4 will describe the color-texture-synthesis technique and experimental results, and section 5 will discuss future work.

2. BACKGROUND

Research conducted to achieve invisibility and active camouflaging has seen some activity in the recent past¹. A common practice is to place a camera behind an object and display-panels in front of it that show background images captured by the camera. The illusion is that the object seems invisible when looking at its front, but this is only effective from a single vantage point. This approach requires knowledge of the observer location in order to calculate the proper size of the displayed image and to ensure the display panels are facing the observer.

To achieve invisibility outside the visible spectrum BAE Systems has developed a cloaking technology that works in the infrared (IR) band². Demonstrations have been performed on vehicles outfitted by hand-sized hexagonal “pixels” that can be heated or cooled to match the ambient temperature, making the vehicle blend into the IR background. Duke University has been able to demonstrate invisibility in the microwave band using metamaterials³. Microwave radiation approaching the metamaterial object flows around it and recombines on the opposite side with little disturbance, making the object appear as if it is not there. The invisibility, however, is limited to a 2D planar region and for a very narrow band of microwave frequencies. Neither of these two approaches is adaptable to the visible spectrum.

An adaptive camouflage system called CAMELEON⁴, developed by the Netherland’s Army collaboratively with Germany and Canada, targets the visible spectrum. An array of Polymer LEDs (PLEDs) is placed behind fixed color filters, with the individual PLED brightness level optimally adjusted with the aid of a camera to match the background. The use of electrochromic materials (tunable color filters) is considered, but such development appears to still be in the research phase. Drawbacks of this system from a man-portable robotic perspective are insufficient resolution, due to the relatively large size of an individual PLED, and continuous power consumption of the PLEDs, which makes this technology unsuitable for extended missions.

3. APPROACH

Attaining invisibility is not trivial and much research is still needed. A more practical interim approach is to make the camouflaged object difficult to distinguish from its surrounding environment in order to keep observers unaware of its presence. Because there is no assumption made about the location of the observer with respect to the camouflaged object, no attempt is made to project a background image on the front-side of the object. Instead, a synthesized camouflage pattern that is visually similar to the surrounding environment of the object is presented on display panels on its exterior surface. The display-panel technology is important, as it is required to faithfully reproduce colors and brightness levels capable of matching the local environment. There are new display-technologies being developed that look promising and will be discussed in the following section; however, what is of equal importance is knowing what to display on them. Texture synthesis is an important part of this concept and a critical component in the development of a future Adaptive Electronic Camouflage (AEC) system.

3.1 AEC overview

Adaptive camouflaging requires the appearance of the outer surface of a robot or a leave-behind sensor to change in color and pattern to match the surrounding environment. This can be achieved by taking an image of the local environment with the existing onboard camera of a robot, synthesizing a statistically equivalent texture image, and displaying the synthesized image on the exterior display panels that shroud the robot or leave-behind sensor. Electronic paper (e-paper) may be a potential solution for the display panel^{5, 6}. E-papers possess many characteristics that are similar to regular paper: For starters they are reflective in nature – the more light that shines upon them the greater will be the perceived brightness level, and vice versa. For example, if part of an AEC robot is under shade and the other under sunlight the part under shade will naturally appear darker than the part under sunlight. This eliminates the need for complex analysis of light and shadow in order to accordingly adjust the brightness level for various regions of a display panel. Secondly they are thin and flexible – the conformal nature of e-papers allow them to bend around corners,

which can be exploited to design exterior shells with minimal sharp edges, which look out of place in natural environments. Finally, they require zero power for image retention – this is critical for long-term ISR missions. The display only consumes a small amount of power to change the displayed image; otherwise, no power is consumed for image retention.

For a leave-behind sensor in need of camouflaging but not equipped with a camera, the camouflage pattern can be computed externally (by a user or delivery robot) and downloaded to it. If the leave-behind sensor is sufficiently small, it is possible to skip the texture-synthesis process and simply display the image taken from the environment. However, the image taken may be of insufficient resolution to entirely cover a larger object, requiring tiling, which generates repetitive patterns that will look unnatural.

3.2 Texture synthesis approach

A wide-variety of texture-synthesis methods can be found in the literature, and a comprehensive review is beyond the scope of this paper. However, patch-based⁷ synthesis and pixel-based⁸ synthesis are two well-known methods.

The texture synthesis method used on this effort is pixel-based and makes use of the gray-level co-occurrence matrix⁹ (GLCM). GLCMs provide second-order statistics (relationships between pairs of pixels) for a texture image. A brief review is given in section 4.1. The texture-synthesis method that makes use of GLCMs is the spin-flip algorithm¹⁰. The algorithm begins by computing the GLCMs, G_{IN} and G_{OUT} , of the input image (e.g., image of local environment) and output image (randomly generated), respectively. The algorithm randomly selects a pixel from the output image and cycles its gray-level through all values. For each value, G_{OUT} is updated. The value of the selected pixel is permanently changed to the gray-level value that minimizes the error between G_{IN} and G_{OUT} . Without selecting a pixel twice, this process repeats itself until all pixels are considered, which constitutes a single iteration. With each iteration, the error between G_{IN} and G_{OUT} decreases such that G_{OUT} becomes statistically equivalent to G_{IN} . This repetitive process can stop in one of two ways: 1) after completing a fixed number of iterations or 2) when reaching a predetermined minimum error between two consecutive iterations. Selecting a fixed number of iterations has been experimentally shown to be sufficient.

The texture synthesis results from this approach have been shown to have high correlation with the human perception of texture similarity¹⁰. Since the goal of the AEC system is to make observers unaware of the presence of the camouflaged object, the camouflage image must be visually similar to its surrounding environment as perceived by human observers. Therefore, the GLCM-based texture-synthesis technique is very well suited for this purpose.

4. COLOR TEXTURE SYNTHESIS

This section will provide a brief overview of the GLCM-based texture-synthesis approach and the modifications devised to produce color textures. Several example of color-texture synthesis will also be provided.

4.1 GLCM review

For a given texture, a GLCM provides second-order statistics, which captures the spatial relationship for a pair of pixels that are separated by a displacement vector $\mathbf{d} = [dx \ dy]$. For each displacement vector $\mathbf{d}_l = [dx_l \ dy_l]$, where $l = 1, 2, \dots, M$, a GLCM is calculated. Collectively the M GLCMs comprise the set $G = \{GLCM_1, GLCM_2, \dots, GLCM_M\}$. The cell value of $GLCM_l(i,j)$ is determined by counting the number of times the first pixel has value i and the second pixel, separated by \mathbf{d} , has value j . Examples of GLCMs for two specific displacement vectors are shown in Figure 4.

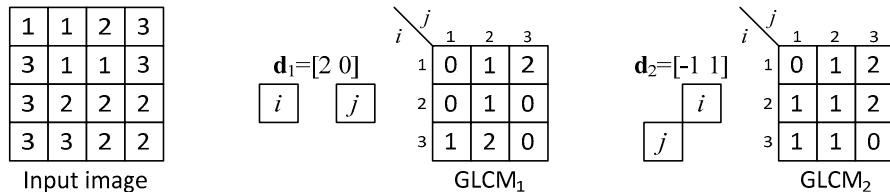


Figure 4. GLCM examples for displacements $\mathbf{d}_1 = [2 \ 0]$ and $\mathbf{d}_2 = [-1 \ 1]$.

Since the row and column indices of a GLCM correspond to the gray-level values of a pixel-pair in the input image, the GLCM will be of size $N \times N$, where N is the total number of gray-level values within the input image. In Figure 4, the GLCMs are of size 3 x 3 since the gray-level values of the input image range from 1 to 3. For a very simple image, like a

checkerboard pattern, where every other pixel alternates between the values 0 and 1, a single GLCM having displacement vector $\mathbf{d} = [1 \ 0]$, will completely describe the image. The GLCM in this case would be of size 2×2 with cells (0,0) and (1,1) containing the value zero, while cells (0,1) and (1,0) containing equal and non-zero values. For more complex and realistic textures, more displacement vectors must be considered, and a GLCM calculated for each in order to have a comprehensive statistical description for the texture. The collective displacement vectors form the set $D = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_M\}$, which defines the spatial neighborhood. For example, displacement vectors $\mathbf{d}_1 = [1 \ 0]$, $\mathbf{d}_2 = [1 \ 1]$, $\mathbf{d}_3 = [0 \ 1]$, and $\mathbf{d}_4 = [-1 \ 1]$ form the spatial neighborhood $D = \{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4\}$. Due to symmetry, there is no need to calculate $-\mathbf{d}_l$, where $l = 1 \dots 4$; only displacement vectors that vary over 180 degrees are considered. In this example, the size of D , as spanned by displacement vectors \mathbf{d}_l , is 2×3 .

4.2 Color texture synthesis

Clearly, GLCMs only capture the gray-level dependencies for pairs of pixels, and so color dependencies must also be taken into account. If color dependencies are not considered and the spin-flip algorithm applied to each channel (e.g., R, G, and B) independently, the combined synthesized channels will produce random colors that are not found in the input image, which results in a camouflage pattern that does not match the local environment.

To include color dependencies, it is necessary to incorporate inter-channel and intra-channel pixel dependencies. The former is simply the gray-level dependencies produced by the GLCMs of the spatial neighborhood. The latter is produced by a set of conditional probability matrices (see modification #1). The color-texture-synthesis process, which is a modified version of the gray-level texture-synthesis process, is applied to each channel. Whether the channels are the traditional R, G, and B color channels, Hue (H), Saturation (S), and Luminosity (L) channels, or some other triplet, the process is the same. The channels are designated as A , B , and C to maintain independence from channel representation. The color-texture-synthesis process begins by generating a random output image for channel A that is iteratively modified for a fixed number of iterations. A random output image for channel B is generated that is dependent upon the final synthesized image for channel A , and iteratively modified for a fixed number of iterations. Finally, a random output image for channel C is generated that is depended upon the final synthesized images for channels A and B , and iteratively modified for a fixed number of iterations. The combined channels A , B , and C form the final synthesized image.

Three modifications are made to the gray-level texture-synthesis process, which are required for producing color textures, as described below.

Modification #1 – The first modification is to develop conditional probabilities for channels B and C instead of using histograms. The required probabilities are $P_A(i)$, $P_B(j | i)$, $P_C(k | i,j)$, where A , B , and C are the H, L, and S channels, respectively. The rationale behind the ordering of the channels is explained in section 4.3, however, as far as the texture-synthesis process is concerned, it does not matter. The variable i , j , and k are the gray-level pixel value for channels A , B , and C , respectively.

Figure 5 illustrates the intra-channel dependencies in the form of conditional probability trees. Suppose the input image (of the local environment) is of size $X \times Y$ (e.g., 4×4) and converted from RGB-space into HSL-space with $A = H$, $B = L$, and $C = S$. Furthermore, suppose that the gray-level pixel value in any channel A , B , or C can take on only one of three values. The conditional probability matrices in Figure 5 are generated as follows: For channel A , count all the pixels with value $H1$ and divide by the total number of pixels $T_A = XY$ to obtain p_1 . Do the same for values $H2$ and $H3$ to obtain p_2 and p_3 , respectively. Together the p_i values form the probability vector $P_A(i)$ or the histogram for channel A . For the next channel, count every pixel in channel B when the corresponding pixel location in channel A has value $H1$, to obtain the sum $T_{B|H1}$. Next, count all the pixels with value $L1$ and divide by $T_{B|H1}$ to obtain $p_{1,1}$. Then count all the pixels with value $L2$ and divide by $T_{B|H1}$ to obtain $p_{1,2}$. Repeat for pixel value $L3$.

To determine $T_{B|H2}$, count every pixel in channel B that occurs when the corresponding pixel location in channel A has value $H2$. To obtain $p_{2,1}$, divide the total number of pixels with value $L1$ by $T_{B|H2}$. Repeat for all remaining values to determine $p_{i,j}$ to form the conditional probability matrix $P_B(j | i)$. Finally, count every pixel in channel C when the corresponding pixel location in channel A has value $H1$ and the corresponding pixel location in channel B has value $L1$, to obtain the sum $T_{C|H1,L1}$. Next, count all the pixels with value $S1$ and divide by $T_{C|H1,L1}$ to obtain $p_{1,1,1}$. Repeat this process to determine $T_{C|H1,L2}$, $T_{C|H1,L3}$, and all the remaining sums needed to calculate $p_{1,2,1}$, $p_{1,3,1}$, and all the remaining probabilities $p_{i,j,k}$ that collectively form the conditional probability matrix $P_C(k | i,j)$.

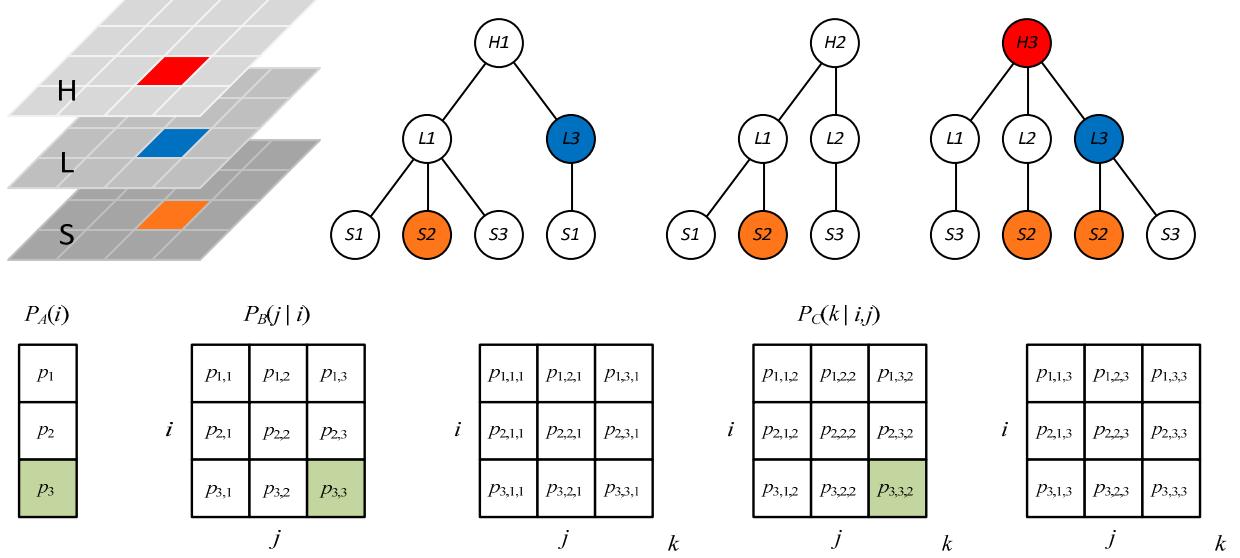


Figure 5. A sample input image shows the corresponding pixel locations for the combination $H3$ (red), $L3$ (blue), and $S2$ (orange), that occur with probability $p_{3,3,2}$ (highlighted in $P_C(k | i, j)$) in channels A , B , and C . Similarly, the combination $H3$ and $L3$ occur with probability $p_{3,3}$ (highlighted in $P_B(j | i)$) in channels A and B . Finally, the value $H3$ occurs with probability p_3 (highlighted in $P_A(i)$) in channel A .

Modification #2 – The second modification ensures that the color proportions in the final synthesized image are statistically equivalent to the color proportions in the input image. For example, if one color occurs more often than another color in the input image, the same will be true in the synthesized image. This is achieved by making use of the probabilities discussed in modification #1.

The standard gray-level texture-synthesis process requires an initial random image that is generated based on the histogram of the input image, which ensures that the gray-level proportions are maintained in the final synthesized image. For color-texture-synthesis, an initial random image is also required, but for channels B and C the conditional probabilities $P_B(j | i)$ and $P_C(k | i, j)$ are used instead of the histograms $P_B(j)$ and $P_C(k)$. Each pixel value in the initial random image of channels B and C is drawn from the conditional probabilities $P_B(j | i)$ and $P_C(k | i, j)$, respectively. This ensures that as the random image is iteratively modified, the color proportions are maintained. The initial random image for channel A , however, only requires the histogram $P_A(i)$, because channel A is completely independent.

Modification #3 – The third modification ensures that the final synthesized image only contains colors that appear in the input image. In other words, the final camouflage image will not contain colors that do not appear in the local environment. As discussed in section 3.2, the spin-flip algorithm cycles the gray-level values of the pixel under consideration to determine which gray-level produces the minimum error between the sets G_{IN} and G_{OUT} . It is clear from Figure 5 that some combinations of H , L , and S do not occur in the input image. For example, the combinations $H1-L2$ and $H2-L2-S1$ do not exist in the input image; their probabilities $p_{1,2}$ and $p_{2,2,1}$ are equal to zero. Therefore, the spin-flip algorithm must not consider such combinations while cycling through pixel values. To enforce this restriction, the spin-flip algorithm only cycles through pixel values that have a probability $P_B(j | i) > 0$ and $P_C(k | i, j) > 0$. Consider Figure 5 and note that pixel value $L2$ does not occur in combination with pixel value $H1$. As a result, the spin-flip algorithm only cycles through values $L1$ and $L3$ for a pixel in channel B that has a corresponding pixel in channel A with value $H1$. Similarly, for a pixel in channel C that has corresponding pixels in channels A and B equal to $H2$ and $L1$, respectively, the spin-flip algorithm only considers pixel values $S1$ and $S2$.

4.3 Results

Several parameters must be specified before the texture-synthesis process can begin. They are: size of the spatial neighborhood, number of iterations, number of gray-levels, and size of the output image. The size of the spatial neighborhood defines the extent of the displacement vector \mathbf{d} . A larger spatial neighborhood will result in a synthesized image that better captures the features of the input image. The number of iterations determines the amount of error

between the sets G_{IN} and G_{OUT} . The error is reduced with increasing number of iterations, but with diminishing gains. The number of gray-levels determines the size of the GLCMs and the shades of gray used to produce the synthesized image. The last parameter specifies the size of the synthesized image. An increase in value for any of these parameters will result in an exponential increase in synthesis time. In order to help reduce the synthesis time and improve the quality of the synthesized textures, Copeland et al.¹⁰ suggest using a multi-resolution technique (to allow for a reduction in the size of the spatial neighborhood), reducing the number of gray-levels (they use 8 gray-levels computed via K -means clustering), and using weighted error calculations. The color-texture-synthesis process incorporates these techniques.

The color-texture-synthesis process also requires specifying the triples and the ordering of the channels. Although they have no impact on the synthesis time, experimental results show that working in the HSL-space with an ordering of $A = H$, $B = L$, and $C = S$, produces somewhat better (more similar-looking) results than working in the RGB-space. This appears to be the result of the probability distribution for individual GLCMs in the set G_{IN} . Generally, for a vast majority of images taken from nature, the gray-levels of the R, G, and B channels are more uniformly distributed than that of the H-channel. Since the gray-levels of the H-channel represent color tones, and there are usually just a handful of color-tones in such images, the gray-level distribution of the H-channel will be highly non-uniform.

As a result, the probability distribution of the corresponding GLCM will be highly non-uniform. In addition, the low spatial frequency of color-tones in such images will result in highly non-uniform probability distributions in a greater number of GLCMs in the set G_{IN} . This non-uniformity is advantageous when using the weighted-error method because it will force the spin-flip algorithm to favor pixel values that will minimize the larger discrepancies (caused by the non-uniformities) between the sets G_{IN} and G_{OUT} . In other words, the spin-flip algorithm will choose pixel values in the output image that better represent (look more similar to) the input image.

The ordering of the L and S channels does not appear to make a significant contribution to texture similarity, possibly due to the increased uniformity in gray-levels as compared to the H channel. However, the probability distribution of the L-channel gray-level values is generally more non-uniform than that of the S-channel gray-levels, therefore, it is processed next, with $B = L$. Figure 6 shows an example of a synthesized color texture along with the error for each iteration and resolution.

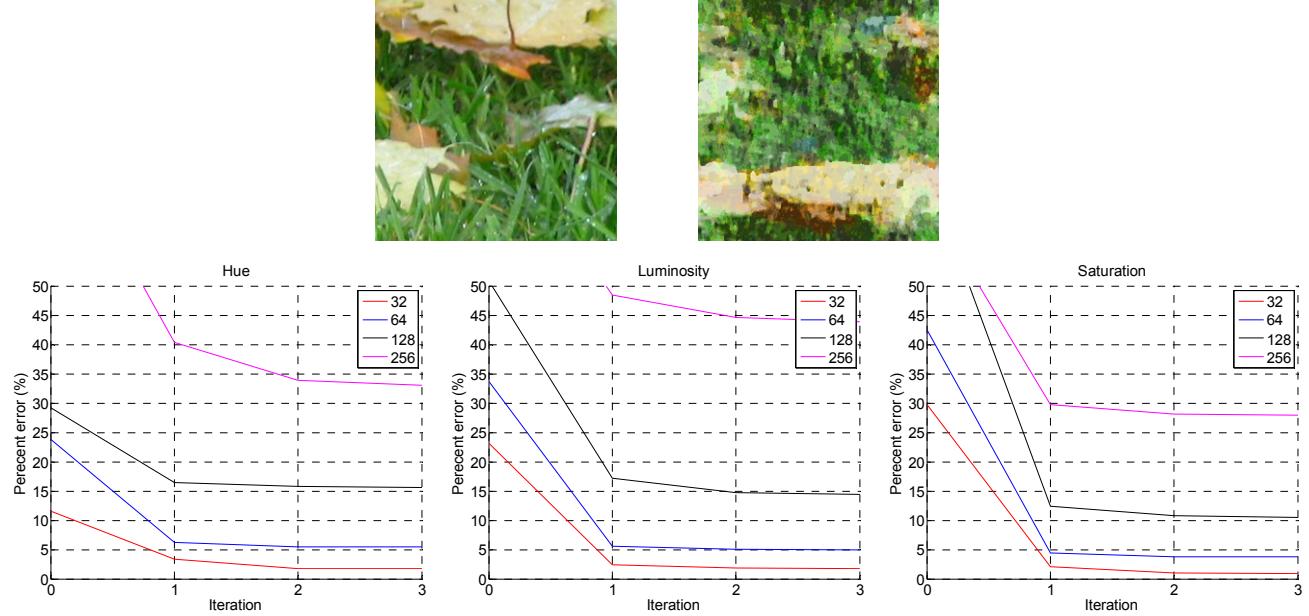


Figure 6. Input image (top left) and synthesized image (top right) using the multi-resolution technique with four levels. A spatial neighborhood of 11×21 was used for level 1 (image size 32×32), 9×17 for level 2 (image size 64×64), 7×13 for level 3 (image size 128×128), and 5×9 for level 4 (final image size 256×256). The error plots (bottom row) show the decreasing error between the sets G_{IN} and G_{OUT} with increasing iterations. The error increases with each level due to the increasing size of the synthesized image and D .

The error is obtained using the sum of absolute difference between the sets G_{IN} and G_{OUT} .

$$error(\%) = 100 \frac{\sum_{w} \sum_{v} \sum_{u} |G_{IN}(u, v, w) - G_{OUT}(u, v, w)|}{\sum_{w} \sum_{v} \sum_{u} G_{IN}(u, v, w)}$$

The indices u and v correspond to the size of an individual GLCM and vary from 0 to $N-1$. The index w corresponds to the number of GLCMs and varies from 1 to M . The error metric is only meaningful in evaluating the performance of the texture-synthesis process by comparing synthesized images of the same size, same triplet, and same spatial neighborhood D . Variations in D and image size will cause variations in the number of GLCMs and probability distributions, respectively, which will result in error values that cannot be compared. Comparing the error between the R and H channels, for example, is also meaningless; therefore, the triplets must also be the same.

Figure 7 shows the effects of modification #2. The columns from left to right correspond to channels $A = H$, $B = L$, and $C = S$. The top row is the input image, the middle row the initial output image created randomly, and the bottom row the final synthesized image. The initial image for channel A is entirely random because channel A is independent of any other channel. However, the initial images for channels B and C are clearly not randomly distributed and depend on the final synthesized image of the previous channel. Modification #2 allows the color proportions to be maintained in the final synthesized image.

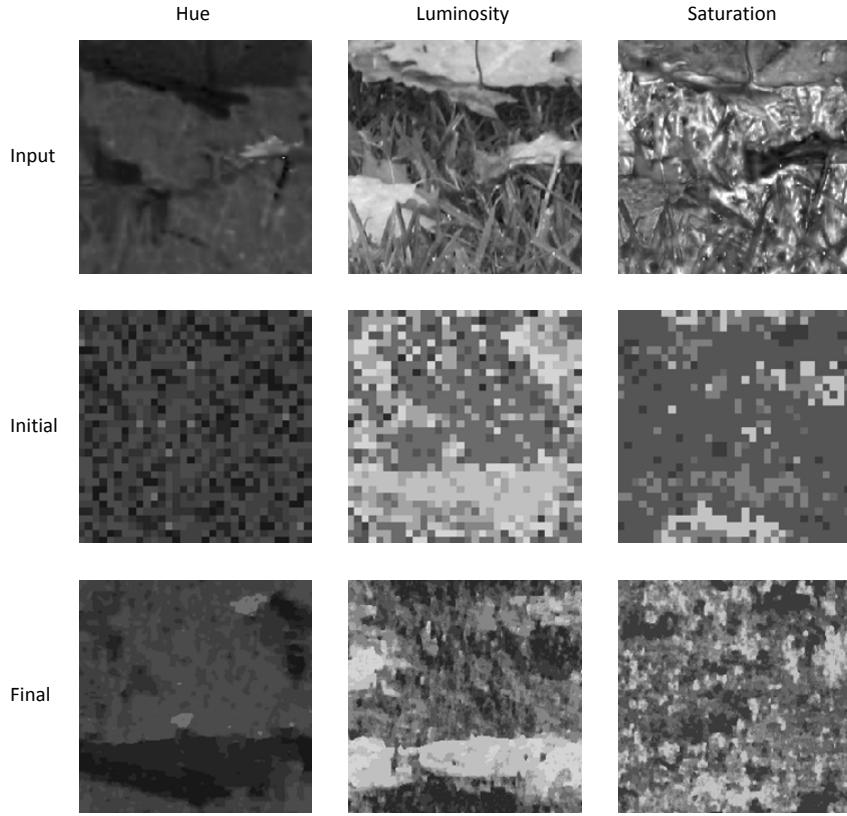


Figure 7. Individual channels $A = H$, $B = L$, and $C = S$. Top row is the original image, middle row the initial image, and bottom row the synthesized image. Initial image for channels B and C shows non-random distribution of pixel values resulting from modification #2. The low resolution of the initial output image (32×32) is due to using the multi-resolution technique.

Figure 8 shows the results of several more synthesized images placed alongside their corresponding input image. Although a synthesized image is not entirely identical to the corresponding input image, its appearance is very similar. This *similarity* is maintained whether the input image is stochastic in nature or has some level or repetitive features,

which supports the fact that the GLCM-based texture-synthesis process produces results that have a high correlation with the human perception of texture similarity.



Figure 8. Results of color texture synthesis. The left half of each texture shows the input image and the right half the synthesized image. Although the synthesized texture is not identical to the input image, the similarity is highly evident.

4.4 AEC simulation

To visualize the effectiveness of camouflaging an ISR robot using the color-texture-synthesis process, a simple 3D model of a fictional ISR robot was placed within various environments and its camouflage pattern changed according to the local background. The results are shown in Figure 9. The robot is designed to be as practical and realistic as possible. For example, the bezel of a real display panel is modeled as a gap between the virtual panels (gray patches in left image). For effective maneuverability in rough terrain, the robot is given tracks that are shrouded by display panels as much as possible without compromising mobility. Finally, the camera is placed inside an articulated arm to allow observation from higher vantage points, which is also beneficial for taking images of the local surroundings for camouflaging purposes. The camouflage patterns in Figure 9 were generated using the color-texture-synthesis process described in this paper. The input image was taken by raising the arm and pointing the camera down, as illustrated in the left image.



Figure 9. A virtual ISR robot using the AEC system in various simulated environments. An image is taken of the local environment (as illustrated in the left image) and supplied as the input image to the real color-texture-synthesis process to generate the camouflage pattern displayed on the exterior surface of the robot.

5. FUTURE DEVELOPMENT

The color-texture-synthesis process described in this paper has been shown to produce a texture that is similar in appearance to its corresponding input image. Texture similarity is critical in developing a convincing camouflage pattern for ISR robots and leave-behind sensors, and thus an important component of a future AEC system. Texture synthesis from a single input image, however, may be insufficient to capture the appearance of a particular local environment. Future work will consist of using multiple input images taken from various regions of a local area and combining them to produce a more convincing and comprehensive camouflage pattern. Ultimately, this approach must be tested on display panels that are suitable for use, but with the understanding of their brightness and color-gamut-range limitations.

For purely reflective displays like e-paper, the brightness-level will be limited by the material and design of the display. The limited color-saturation levels resulting from the limited color-gamut-range of the display will cause noticeable color mismatches between the environment and the displayed image. To address these limitations, it may be necessary to develop display-panels with enhanced brightness and color-saturation with a slight tradeoff in resolution. Color-correction methods may also be necessary to optimally match the displayed colors to those of the local environment. Finally, specular reflectance under sunlight may draw undesired attention to the camouflaged object. An outer-coating covering the display that can diffuse and scatter the light may be required. These limitations will be addressed as display technologies improve, helping to make the proposed camouflaging method more effective.

REFERENCES

- [1] McKee, K. W. and Tack, D. W., "Active Camouflage for Infantry Headwear Applications," Humansystems, Inc., on behalf of Defense Research and Development Canada – Toronto, (2007).
- [2] Corporate Communications, "ADAPTIV – A Cloak of Invisibility," BAE Systems, http://www.baesystems.com/magazine/BAES_019786/adaptiv--a-cloak-of-invisibility
- [3] Schurig, D., Mock, J. J., Justice, B. J., Cummer, S. A., Pendry, J. B., Starr, A. F., and Smith, D. R., "Metamaterial Electromagnetic Cloak at Microwave Frequencies," Scienceexpress, (2006).
- [4] <http://www.youtube.com/watch?v=zLdNeatXCvE>
- [5] Vermeulen, P., Feenstra, J., Giraldo, A., and Hampton, M. W., "Modeling of the Performance of Electrowetting Displays," Proc. SID Vol. 42 Issue 1, 1573-1576 (2011).
- [6] Kim, C. D., Yoo, J. S., Lee, J. K., Yoon, S. Y., Park, Y. I., "Full Color Flexible Displays on Thin Metal Foil with Reduced Bending Radius," Flexible Electronics & Displays Conference and Exhibition, (2009).
- [7] Efros, A. A., Freeman, W. T., "Image Quilting for Texture Synthesis and Transfer," Proc. SIGGRAPH '01, 341-346 (2001).
- [8] Wei, L., Levoy, M., "Fast Texture Synthesis using Tree-structured Vector Quantization," Proc. SIGGRAPH '00, 479-488, (2000).
- [9] Jain, R., Kasturi, R., Schunck, B. G., [Machine Vision], McGraw-Hill, Inc., 236-239 (1995).
- [10] Copeland, A. C., Ravichandran, G., Trivedi, M. M., "Texture synthesis using gray-level co-occurrence models: algorithms, experimental analysis, and psychophysical support," Opt. Eng. 40, 2655, (2001).