Autonomous Robots (2018) 42:1635–1650
https://doi.org/10.1007/s10514-018-9717-6

Distributed camouflage for swarm robotics and smart materials

Yang Li[1] · John Klingner[1] · Nikolaus Correll[1]

Received: 16 February 2017 / Accepted: 9 February 2018 / Published online: 21 February 2018

Abstract
We present distributed algorithms for a swarm of static particles to camouflage in an environment by generating colored patterns similar to those perceived in the environment, mimicking the camouflage systems used by cephalopods. We assume each particle to be equipped with sensing, computation, and local communication abilities. For pattern recognition, each particle measures local color and brightness information, exchanges this information with its neighbors, determines local match with a library of patterns, and finally performs a consensus algorithm. For pattern formation, particles implement a distributed variant of Turing's pattern generator. Together, these algorithms enable the swarm to obtain a high-level understanding of its environment and to quickly adapt its appearance to changing environments. All algorithms are evaluated on a swarm of 64 miniature robots ("Droplets") that can sense and change color, and exchange information using directed infrared communication. With required computation being minimal and communication exclusively local, the system serves as a blueprint for further miniaturization and work in biomimetic camouflage.

Keywords Smart materials · Swarm robotics · Autonomous robots · Artificial camouflage

1 Introduction

Camouflage is regularly employed in nature to hide from predators or to attract mates (Stevens and Merilaita 2009, 2011; Bradbury et al. 2011). Animals camouflage in a wide variety of ways, having evolved patterns or shapes suited to their environment. A few species stand out for their ability to change the color and texture of their skin to match their surroundings. This is observed in several distinct groups of animals, including reptiles, cephalopods, and flatfish (Hanlon and Messenger 1988; Stuart-Fox and Moussalli 2009). The underlying mechanisms of these natural camouflage systems are not yet fully understood, but we know that they must be driven by more than just the animals' visual system; evidence indicates that the octopus is color blind (Messenger 1977), yet they use different colors to adapt to their environment. Studies of cephalopods indicate a combination of local sensing and control (Ramirez and Oakley 2015) and higher-level selection of the appropriate pattern (Messenger 2001). The camouflage patterns we typically see in nature can be produced with only local coordination and self-organization (Meinhardt 1982), yet the underlying animal physiology is still not fully understood. We believe these efforts to understand biological camouflage might benefit from engineering a robotic model in the spirit of Webb (2001).

We wish to design an artificial camouflage system that can quickly adapt to a large variety of environments. We draw inspiration from the way cephalopods camouflage, which tightly integrates sensing, actuation (color change), neural computation, and communication (Yu et al. 2014). Our approach seeks to mimic this tight integration with a distributed artificial system. Whereas animals employ camouflage mostly for escaping predators, engineering applications range from enabling clandestine military operations to ubiquitous autonomous logistic solutions and environmental-monitoring applications that seamlessly blend into their environment. What a "good" camouflage pattern is, is still only poorly understood and can only be evaluated empirically using man-in-the-loop tests (Toet 2000; Peek et al.

This is one of several papers published in *Autonomous Robots* comprising the "Special Issue on Distributed Robotics: From Fundamentals to Applications".

✉ Yang Li
yang.li-4@colorado.edu

John Klingner
john.klingner@colorado.edu

Nikolaus Correll
nikolaus.correll@colorado.edu
http://correll.cs.colorado.edu/

[1] Department of Computer Science, University of Colorado, Boulder, USA

2006). For these applications and in nature, effective camouflage is not about perfectly matching the background, which strongly depends on the perspective of an observer. Rather, effective camouflage seeks to break up or conflict with a subject's outline or shadows (Stevens and Merilaita 2009).

In this paper, we aim to replicate the pattern matching ability of natural systems. We present a fully distributed approach, which we implement on a swarm of miniature robots called "Droplets" (Farrow et al. 2014), each equipped with the ability to sense and emit color as well as communicate with its local neighbors. The robots, which don't move, are conceptually related to the chromatophores (color-changing cells) in animals' skins. Distributed sensing and actuation for camouflage generation makes an implementation scalable in the resolution of the camouflage pattern, i.e. the density of chromatophores, or the size of the area being camouflaged. Making the system distributed also makes it more robust to the failure of individual robots. Furthermore, a distributed camouflage system could respond to local changes in the environment, in particular when deployed on non-trivial 3D surfaces. In the long run, we envision the algorithms presented here to run on particles small enough to be suspended in paint (Butera and Bove 2002).

Our approach consists of four phases: First, each particle senses the color in the immediate environment. Second, particles estimate their local pattern types. Third, the swarm consenses on the global pattern type and dominant colors with a consensus algorithm. Finally, the pattern is formed using a reaction-diffusion process. We note that pattern recognition and pattern formation are independent processes, and enable the distributed system to gain a higher-level understanding of the environment it is in, of which selecting an appropriate appearance is only one of many possible actions that a distributed pattern recognition system might enable.

As the Droplets are a general-purpose platform, some concessions are made in our algorithm that would not be necessary for more purpose-built hardware. In particular, the Droplets can not sense local color and emit light simultaneously, as the sensor is overwhelmed while the emitter is active. The "background" into which the swarm is trying to camouflage is projected on to the particles from above. To simplify the experimentation, the paper focuses on two-tone patterns. Finally, we arrange the particles in a grid pattern, which allows us to implement a discrete convolution operation and simplifies debugging the pattern at the resolution afforded by a swarm with a population on the order of tens of particles.

2 Related work

In engineering, there have been multiple attempts to achieve active camouflage using a combination of cameras and projection (Inami et al. 2003; Lin et al. 2009; Lasbury 2017). In these works, the background image is usually caught by a camera and projected on the front surface of an object. The surface is made of retro-reflective material to reflect the most light projected on it. In this way, an object can become virtually invisible from the viewpoint of an observer. Although such systems provide "perfect" camouflage, they are highly dependent on the observer's viewpoint. They are therefore impractical in scenarios where the position of the observer is unknown or where there are multiple observers.

A concept often confused with camouflage is that of "cloaking" using optical metamaterials, which allow to manipulate light waves in a way that allows them to be routed around an obstacle (Pendry et al. 2006). Albeit partial invisibility has been achieved even for larger objects (Smith 2014), existing approaches are usually limited to light of a specific wavelength.

Both approaches are distinctly different from active camouflage in animals, which neither simply reproduce an observed pattern nor employ cloaking techniques. Instead, animal camouflage consists of a combination of visual perception of the environment, decision making on an appropriate color-scheme and pattern, and distributed pattern formation in the skin.

2.1 Visual perception and recognition of color and pattern

Albeit how animals perceive their environment is not fully clear, the consensus is that the majority of the perception is done via the eyes and processed in the brain (Troscianko et al. 2009). The visual strategy cuttlefish use to select camouflage patterns and colors is fundamentally similar to human object recognition (Kelman et al. 2008), that is the cuttlefish actively processes features like edges (Zylinski et al. 2009), contrast, and size, to identify specific objects, and consequently decides upon a camouflage pattern (Stevens et al. 2006).

In order to understand what the best possible camouflage pattern is, it helps to understand the evolutionary origins of the mechanism. Minimizing the likelihood of detection is one of the major components for active camouflage. What kind of patterns have this property depends on the perception apparatus of a predator. Identification of an object, so-called figure-ground processing, has two stages. First, there is a low-level process whereby individual neurons detect the locations, polarity, and orientation of small edge segments. The second stage groups the local edge information to identify those edges that belong to a single object and reject others that belong to the background (Lamme 1995; Grossberg et al. 1997). Edge detection and edge grouping might be exploited in two ways by a prey animal to become less visible to predators. First, its coloration or markings might

make the small edge segments difficult to discern, if the animals body is of very similar color and brightness as the background; this way is called *background matching*. A second way of exploiting the mechanisms of edge processing is to disrupt the grouping of the small edge segments to form a coherent outline of a whole object (Troscianko et al. 2009); this way is called *disruptive coloration*. There is debate among researchers as to whether background matching or disruptive coloration have a more important role in camouflage, or whether it is the combination of the two that is most successful (Stuart-Fox and Moussalli 2009). In any case, these simple mechanisms are in stark contrast with displaying verbatim imagery from the environment, such as using projection-based methods or simply replaying local sensing information. Instead, such methods, depending on the viewing angle, might have the opposite effect and might stand out even more than a static camouflage pattern.

In this paper, we are not concerned with high-level reasoning for pattern selection, but initially focus on basic edge detection and color selection algorithms that can be implemented in a distributed architecture. There is only limited work on distributed pattern recognition in swarm robotic systems. For example, Otte (2016) demonstrates a distributed neural network architecture that can differentiate between different symbols projected on a swarm by evaluating local color gradients and consensus, a work that comes closest to that presented here.

2.2 Coloration and pattern formation

Once an appropriate color scheme and pattern has been selected, it needs to be generated. To implement color change, the literature considers two principle methods — morphological and physiological — which differ both in the mechanism and the speed with which the color change can be accomplished. Cephalopods and chameleons belong to the latter category and rely on neuromuscular organs known as chromatophores (Messenger 2001). Chromatophores are pigmented elements with color and brightness or luminance, and they are controlled by lobes in the brain so that cephalopods can adapt their appearance extremely rapidly (within milliseconds or seconds).

Instead of mimicking the background, most animals rely on a few simple families of patterns of speckles and stripes, which can be classified into uniform, mottled, and disruptive (Hanlon 2007). Uniform patterns exhibit little to no contrast, mottle patterns exhibit small-to-medium scale light and dark patches, whereas disruptive patterns have large-scale light and dark components of multiple shapes, orientations, scales, and contrasts.

Interestingly, such patterns are ubiquitous in nature, not limited to camouflage patterns, but can also be observed in sand dunes or seashells, and can all be described with very similar mathematical tools (Meinhardt 2009). One of the best-known theoretical models used to explain self-regulated pattern formation is the Turing or reaction-diffusion (RD) model, which has first been presented by Turing (1952). Simulation studies of the RD model have shown that this system can replicate most biological patterns (Kondo and Miura 2010). Turing's basic idea is that "the mutual interaction of elements results in spontaneous pattern formation". Hereafter Meinhardt and Gierer (2000) proposed that a Turing pattern can be formed by only involving two different feedback mechanisms—a short-range positive feedback and a long-range negative feedback. The positive and negative feedback loops are also called activator and inhibitor. A local activator-inhibitor model for pattern formation is proposed in (Young 1984) (Young's model) based on evidence that the inter-cellular interaction is local. We use Young's model in our camouflage system as we can easily apply the model in a distributed way.

2.3 Materials for camouflage coloration

There exist multiple attempts to create artificial chromatophores. Rossiter et al. (2012) presented a soft and compliant artificial chromatophore that mimics the contrasting mechanisms of sacculus expansion employed by cephalopod chromatophores using electroactive polymer artificial muscles. Wang et al. (2014) demonstrated a soft material system that can exhibit a wide variety of fluorescent patterns under the control of electric fields. Morin et al. (2012) described a soft machine for camouflage and display, which can change color and contrast by pumping in and out a colored liquid through microfluidic channels. Chou et al. (2015) reported a stretchable electronic skin (e-skin) that can change color through varying the applied pressure. The artificial skin presented in (Fishman et al. 2015) is made of a sheet of electroactive dielectric elastomer painted with colored electrodes and can generate dynamic patterns. However, very few works articulate the systems challenges that require not only a mechanism for local color changes, but also local sensing and computation (Yu et al. 2014), or investigate the ability to co-locate simple signal processing with the sensors themselves (Fekete et al. 2009). This paper aims at filling this gap, and we are using RGB light to simulate chromatophores and exclusively focus on an algorithmic, sensing and communication aspect.

3 Distributed camouflage system

We assume an arbitrary arrangement of robots on a manifold that allows the robots to communicate locally. The only

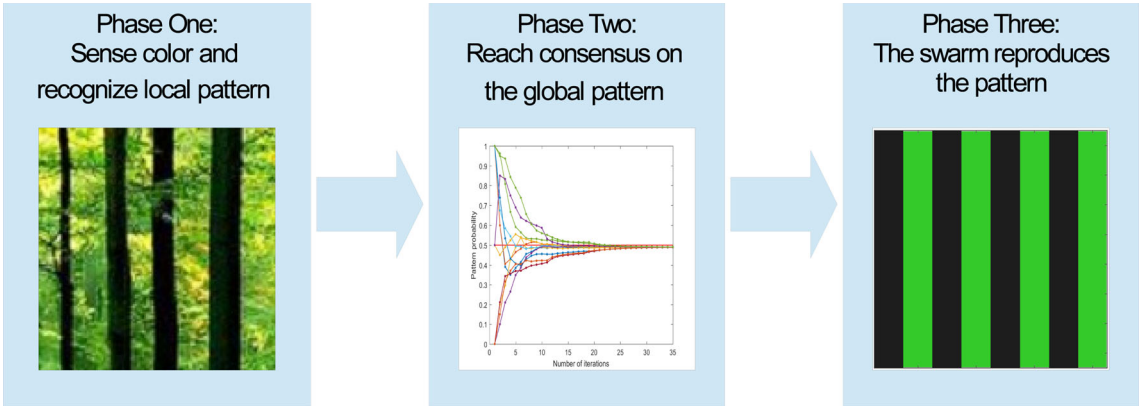

Fig. 1 Pipeline of the distributed camouflage algorithm (Color figure online)

restriction we place on this arrangement is that the robots' communication graph is connected and undirected. We treat the robots' environment, in which they are trying to camouflage, as projected on to that plane, and assume that each robot can measure the color projected on to it and communicate with its local neighbors.

The proposed system can be summarized as follows (see the pipeline of the system in Fig. 1). First, the robots use a weighted-average consensus algorithm (described further in Sect. 3.1) to determine the two dominant colors. Then, a modified Laplacian of Gaussian (LoG) filter is used to estimate the local pattern type, and the same consensus algorithm is used to decide the global pattern. Finally, the robots collaborate to generate an appropriate pattern using the two dominant colors. We begin by describing the consensus algorithm in detail.

3.1 Weighted-average consensus algorithm

The goal of the weighted-average consensus algorithm is to reach an agreement on some values among distributed robots. Let ν denote a global value vector which contains the values to be agreed on, and ν_i for each robot R_i's individual value vector. Note that values of ν and ν_i are real numbers. Moreover, $\Lambda(i)$ denotes the set of neighbors of robot R_i and $d_i = |\Lambda(i)|$ denotes the degree (number of neighbors) of robot R_i.

Pseudocode for this algorithm is provided in Algorithm 1, which is executed in each robot. At each step, each robot updates its ν_i to be the weighted-average of its own and that of its neighbors'. This step is repeated many times, allowing information to diffuse through the graph. Since the weighted-average procedure just uses local information, each iteration takes the same amount of time regardless of the number of nodes in the swarm.

The weighted-average calculation uses Metropolis weights, defined as:

Algorithm 1 Distributed Average Consensus Algorithm

1: ***Initialization***:
2: ν_i as described elsewhere
3: ***Iterations***:
4: **for** each iteration **do**
5: Broadcast ν_i and d_i
6: $\nu_i^* = 0$
7: **for** $R_j \in \Lambda(i)$ **do**
8: Receive ν_j and d_j
9: Calculate $W_{i,j}$ using Metropolis Weights
10: $\nu_i^* = \nu_i^* + W_{i,j}\nu_j$
11: **end for**
12: $\nu_i = \nu_i^* + W_{i,i}\nu_i$
13: **end for**

$$W_{i,j} = \begin{cases} \dfrac{1}{1 + \max(|\Lambda(i)|, |\Lambda(j)|)} & \text{if } R_j \in \Lambda(i) \\ 1 - \displaystyle\sum_{R_k \in \Lambda(i)} W_{i,k} & \text{if } i = j \end{cases} \quad (1)$$

The Metropolis weights are well suited for distributed algorithms, since weight-calculation requires only local knowledge. Further, it is proved in (Xiao et al. 2005) that "Metropolis weights guarantee convergence of the average consensus provided that the infinitely occurring communication graphs are jointly connected". If the union of a set of graphs with common vertex set is a connected graph, the set of graphs is called 'jointly connected'. It requires a weak condition on the long-term connectivity of the graph that it allows links to fail temporarily or even permanently as long as the surviving links make a connected graph. The number of iterations necessary for convergence is determined experimentally.

3.2 Dominant colors detection

Two-tone patterns are the simplest for animals to generate, and dominate in nature, so the first task of our distributed camouflage system is to detect the two dominant colors in

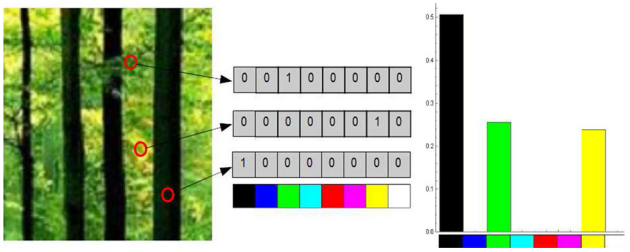

Fig. 2 Illustration of how to detect dominant colors with the palette based method. First, robots measure the environment colors (left image); Second, colors are mapped to colors in the palette (middle image); Then the robots initialize vectors accordingly, which will converge to a histogram (right image) to decide dominant colors (Color figure online)

the environment. Usually the most common way to find dominant colors in an image is clustering methods like k-means (Ying 2007) and fuzzy c-means (Bian et al. 2010). However, these clustering methods assume that all the colors in the image are known beforehand, an assumption we cannot make in a distributed system. Inspired by color-space reduction techniques from image compression, we propose a color-palette based method. First, by uniformly sampling the color space, we establish a color palette **CP** — consisting of a map of red, green, and blue (RGB) intensity value vectors.

Each robot measures the full RGB-space color of its immediate environment, and chooses the color from our restricted palette with the closest Euclidean distance (in RGB space) to the sensed color. The robot then initializes a vector the size of the color palette, with all values initialized to 0 except for its sensed color, initialized to 1, and starts participating in a weighted-average consensus algorithm with that vector (and the vectors from all other robots). This algorithm is described in details in Sect. 3.1.

Consensus is achieved when the vector converges (in practice, after a fixed number of iterations), the resulting vector will represent the frequency distribution of colors in the image. Figure 2 shows an example using a color palette with eight colors (black, blue, green, cyan, red, magenta, yellow, and white). We assume the colors with the two highest frequencies to be the two dominant colors, and use c^{A} and c^{B} to denote them.

3.3 Pattern descriptors calculation

Once each robot has determined the two dominant colors, it needs to estimate the local projected pattern. The most common patterns in nature are stripes and mottled patterns. We assume three patterns: horizontal stripes, vertical stripes and mottle. To identify different patterns, it is straightforward to compare the color changes in horizontal direction and vertical direction. For example, in a horizontal pattern the color changes much more along the vertical direction than the horizontal direction. Here we employ a Laplacian of Gaussian (LoG) filter, which is commonly used in image processing (Jain et al. 1995). The Gaussian component of this filter helps to filter out noise, whereas the Laplacian component detects zero-crossings in the second derivative. To derive this filter, we start with the probability distribution function of a standard bivariate Gaussian with zero-mean and variance σ^2:

$$f(x, y) = \frac{1}{2\pi\sigma^2}\mathbf{e}^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

The Laplacian of this function is given by

$$\nabla^2 f = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \tag{3}$$

$$= \frac{1}{2\pi\sigma^6}\mathbf{e}^{-\frac{x^2+y^2}{2\sigma^2}}\left(x^2 + y^2 - 2\sigma^2\right) \tag{4}$$

We separate $\nabla^2 f$ into two components such that $\nabla^2 f(x, y) = \mathbf{LoG_x}(x, y) + \mathbf{LoG_y}(x, y)$ where

$$\mathbf{LoG_x}(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} = \frac{1}{2\pi\sigma^6}\mathbf{e}^{-\frac{x^2+y^2}{2\sigma^2}}(x^2 - \sigma^2) \tag{5}$$

$$\mathbf{LoG_y}(x, y) = \frac{\partial^2 f(x, y)}{\partial y^2} = \frac{1}{2\pi\sigma^6}\mathbf{e}^{-\frac{x^2+y^2}{2\sigma^2}}(y^2 - \sigma^2) \tag{6}$$

This separation allows us to identify edges in the x and y directions separately.

If we consider the projected environment as an image, then the colors sensed by arbitrarily distributed robots in that environment represent a sampling of that image. A standard LoG filtering would convolve the LoG kernel with the full known image such that each filtered pixel's value is the weighted sum of the region around it, with weights given by the LoG kernel.

Similarly, in our proposed algorithm, we produce an approximate LoG filter by using the LoG function to weigh the colors sampled by nearby robots. Specifically, assume that each robot R_i periodically sends a message consisting of x_i, y_i, c_i, where x_i, y_i represent the respective x, y positions of R_i, and c_i represents the color sensed by R_i. Note that $c_i \in \mathbf{CP}$, the reduced color space. Since the robot is interested specifically in patterns with the dominant colors, it now further reduces the color space, mapping **CP** according to:

$$\mathbf{reduce} : \mathbf{CP} \rightarrow \{0, \tfrac{1}{2}, 1\} \tag{7}$$

$$\mathbf{reduce}(c) = \begin{cases} 1 & \text{if } \|c - c^{\mathrm{A}}\| < \|c - c^{\mathrm{B}}\| \\ 0 & \text{if } \|c - c^{\mathrm{A}}\| > \|c - c^{\mathrm{B}}\| \\ \frac{1}{2} & \text{if } \|c - c^{\mathrm{A}}\| = \|c - c^{\mathrm{B}}\| \end{cases} \tag{8}$$

where, for example, $\|c - c^{\mathrm{A}}\|$ is the Euclidean distance between color c and dominant color c^{A}. Finally, each robot R_i can compute:

$$\nabla X_i = \sum_{R_j \in \Lambda(R_i)} \mathbf{reduce}(c_j) \times \mathbf{LoG_x}(x_j, y_j) \tag{9}$$

$$\nabla Y_i = \sum_{R_j \in \Lambda(R_i)} \mathbf{reduce}(c_j) \times \mathbf{LoG_y}(x_j, y_j) \tag{10}$$

where $\Lambda(R_i)$ is used to describe the set of all robots R_j such that R_i receives communications from R_j. These values $\{\nabla X_i, \nabla Y_i\}$ are used to determine the most likely local pattern.

Each robot R_i maintains a pattern-probability array $p_i = \{p_i^h, p_i^v, p_i^m\}$ to keep track of each node's estimated pattern, where $\{p^h, p^v, p^m\}$ represent the probability of a horizontally striped, vertically striped, or mottled pattern, respectively. One pattern type is selected and initialized with a probability of 1, and the other probabilities are set to 0. Each robot R_i performs this selection using $\{\nabla X_i, \nabla Y_i\}$ according to:

$$p_i = \{p_i^h, p_i^v, p_i^m\} = \begin{cases} \{1, 0, 0\} & \text{if } |\nabla Y_i| - |\nabla X_i| > T \\ \{0, 1, 0\} & \text{if } |\nabla X_i| - |\nabla Y_i| > T \\ \{0, 0, 1\} & \text{otherwise} \end{cases} \tag{11}$$

Note that the pattern estimation method described here assumes that patterns in the image are of unit width. This method can be adjusted for patterns of arbitrary width by simply scaling the inputs:

$$\mathbf{LoG_x}\left(\frac{x}{w}, \frac{y}{w}\right) \qquad \mathbf{LoG_y}\left(\frac{x}{w}, \frac{y}{w}\right) \tag{12}$$

Some control over effective pattern width can also be exerted by adjusting σ for $\mathbf{LoG_x}$ and $\mathbf{LoG_y}$, but we found parameter selection to be simpler when holding σ fixed at 0.5 and only adjusting the width parameter, w, in Equation 12. One solution for patterns at an arbitrary orientation (angled stripes, for example) would simply be to perform a coordinate transform (rotation matrix) to each (x_j, y_j) before computing $\mathbf{LoG_x}$ and $\mathbf{LoG_y}$ in Equations 9 and 10 . In practice, this would require the pattern descriptor multiple times, once for each desired width and for each orientation. The same is the case for detecting patterns of different sizes, which is the equivalent of a spectral analysis of spatial frequencies.

3.4 Consensus on the global pattern

Once each robot has determined the most likely local pattern (i.e., computed $p = \{p^h, p^v, p^m\}$), they need to achieve consensus on the global pattern, as described in Sect. 3.1. Once the robots have converged, the largest value in p represents the most likely global pattern. For example, $p^h > p^v$ and $p^h > p^m$ indicates that the most likely global pattern is horizontal stripes. After the consensus algorithm has converged, each robot R_i's p_i matches the global p. Thus, each robot has agreed on and is aware of the most likely global pattern, as well as the two dominant colors. Now that a global pattern has been selected, the robots need to generate an appropriate camouflage pattern.

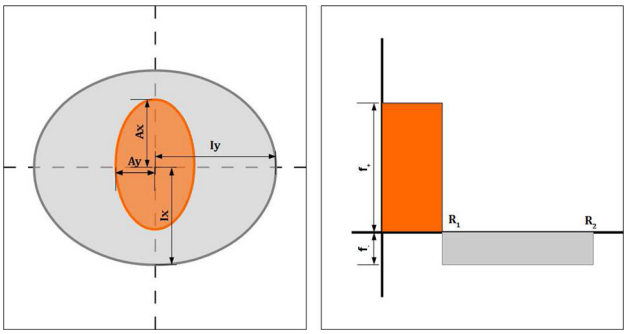

Fig. 3 Illustration of local activator-inhibitor model: on the left, the activator region (orange) is defined by A_x and A_y while the inhibitor region (gray) is defined by I_x and I_y; on the right, f_+ and f_- are the two field values. R_1 is related to A_x and A_y and R_2 is related to I_x and I_y (Color figure online)

3.5 Pattern formation using local activator-inhibitor model

In this section, we describe the distributed pattern formation algorithm to generate a proper pattern to match the environment.

We build upon the pattern formation algorithm presented by Young (1984), a local activator-inhibitor model built following Turing's work in this area (Turing 1952). In this model, each robot is either 'On' or 'Off' and can generate two kinds of morphogens: activator morphogen and inhibitor morphogen. Together, these form a "morphogenetic field". Note that the activator region should be inside of the inhibitor (see left of Fig. 3). The robots in the activator morphogen contribute to stimulate change for nearby 'On' robots, and robots in the inhibitor morphogen contribute to stimulate change for nearby 'Off' robots. 'On' and 'Off' can then be mapped to the two dominant colors.

During each step of this algorithm, each robot changes its 'Off'/'On' status based on the combined effect of all robots in nearby morphogenetic fields. To be specific, a 'strength' is calculated with each 'On' robot contributing a positive factor (f_+) if in the activator region or contributing a negative factor (f_-) if outside the activator region and inside the inhibitor region. The robot then changes its state to 'On' if the strength is greater than zero, and to 'Off' otherwise. This step

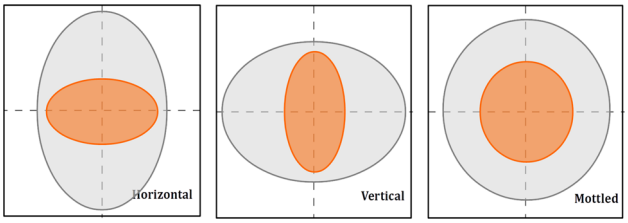

Fig. 4 Activator (orange) and Inhibitor (gray) Regions for horizontal, vertical and mottle patterns (Color figure online)

is repeated until the states of all robots converge to a stable pattern. Young (1984) observes that convergence typically takes around five steps. It is consistent with our observation in the simulations.

In this framework, different types of patterns are represented with differently shaped activator and inhibitor regions. The regions for each pattern are shown in Fig. 4.

Details of the algorithm applied on each robot in a distributed system are shown in Algorithm 2. Each robot stores a pattern type $p^* \in \{p^h, p^v, p^m\}$, field strength factors f_+ and f_-, and iteration times T. After several iterations, robots shall achieve a stable state to form a static pattern.

Algorithm 2 Distributed Pattern Formation

1: ***Initialization***:
2: Randomly set the status 'On'/'Off'
3: Define activator morphogen according to p^*
4: Define inhibitor morphogen according to p^*
5: ***Iterations***:
6: **for** $t = 1 : T$ **do**
7: Count the number of 'On' robots in activator morphogen: N^+
8: Count the number of 'On' robots in inhibitor morphogen: N^-
9: Compute the strength: $ss = f_+ * N^+ + f_- * N^-$
10: **if** $ss > 0$ **then**
11: Turn the robot 'On'
12: **else if** $ss < 0$ **then**
13: Turn the robot 'Off'
14: **else**
15: Don't change the robot's status
16: **end if**
17: **end for**

4 Distributed camouflage on grid graphs

So far, all algorithms have been developed for arbitrary arrangements of robots. In the interest of simplifying localization, we arrange the physical robots on a grid without loss of generality. Within a grid graph, each robot can easily identify its four neighbors (i.e., the robots to the north, south, east, and west of it). Once a robot's neighbors have been identified, the algorithm can proceed with all the same steps described above: dominant colors detection, pattern determination, and

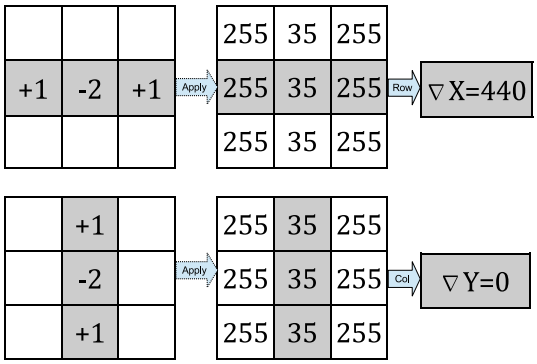

Fig. 5 Illustration of applying the discrete $\mathbf{LoG_x}$ and $\mathbf{LoG_y}$

pattern formation. The same consensus algorithm can be used for dominant color selection, however, the remaining two steps merit some adjustments for this grid-based framework.

Before further explaining these steps, note that the robots' measured colors still represent a sampling of the projected environment as described in Sect. 3.3. Furthermore, since the sampling is now restricted to a grid, the collective robots' sensed colors can be thought of as a low-resolution copy of the projected environment, where each robot functions as a pixel.

4.1 Pattern descriptors

With the robots restricted to a grid, LoG filtering can now be performed using a discrete kernel as is done in image processing. This kernel is shown in Fig. 5 and represents a special case of $\mathbf{LoG_x}$ and $\mathbf{LoG_y}$ as defined in Sect. 3.3.

With $\sigma = \frac{1}{2}$,

$$\mathbf{LoG_x}\begin{pmatrix} (-1,1) & (0,1) & (1,1) \\ (-1,0) & (0,0) & (1,0) \\ (-1,-1) & (0,-1) & (1,-1) \end{pmatrix} = \begin{pmatrix} 0.14 & -0.34 & 0.14 \\ 1.03 & -2.55 & 1.03 \\ 0.14 & -0.34 & 0.14 \end{pmatrix} \quad (13)$$

which is approximately equal to the discrete kernels from image processing shown in Fig. 5. The differences come from rounding for simplicity and adjusting to ensure that $\mathbf{LoG_x}$ still sums to zero over its full domain.

Note that if the grid of robots is viewed as an image with each robot as a pixel, these two pattern descriptors are simply the value the pixel would have after each of the two convolutions. These second-order derivatives — ∇X and ∇Y — can now be used to calculate the most probable local pattern in the same way as described in Sect. 3.3

To be more specific, with c_M denoting a robot's color and c_T, c_R, c_B, and c_L denoting the colors of the robot's top, right, bottom, and left neighbors, ∇X and ∇Y are given by:

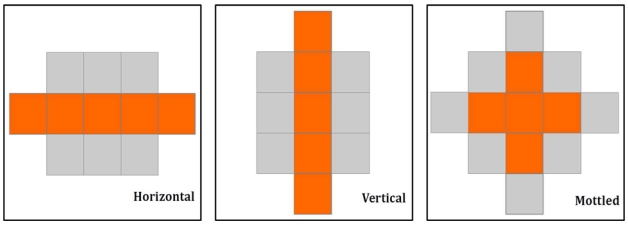

Fig. 6 Activator (orange) and inhibitor (gray) regions for each of the three patterns (Color figure online)

$$\nabla X = c_L + c_R - 2c_M$$
$$\nabla Y = c_T + c_B - 2c_M$$

And each robot R_i can then use these values to initialize p_i, as described in Sect. 3.3.

4.2 Pattern generator

When the robot configuration is reduced to a grid, each robot can use a simplified, discrete form of the activator and inhibitor regions described in Sect. 3.5. The sizes of these regions, shown in Fig. 6 are selected such that each robot only requires information from robots within two hops of it.

5 Simulation results

5.1 Dominant colors detection

We implemented the dominant color detection algorithm introduced in Sect. 3.2 on a centralized system for testing. In the experiments, the size of palette used is 138.[1] In Fig. 7, we show the palette-based method is able to find first two dominant colors in different images. Images in the left column are the original ones size of 128 × 128; images in the middle column are the "color map" obtained by mapping each pixel to the palette; images in the right column are the part of histogram we get. We can see that the first two dominant colors chosen are similar to those that we would choose on the original images.

5.2 Pattern formation on grid graphs

The two kinds of morphogens—activator morphogen and inhibitor morphogen — are defined by two ellipses; for horizontal major axis, it is defined as

[1] http://www.rapidtables.com/web/color/RGB_Color.htm.

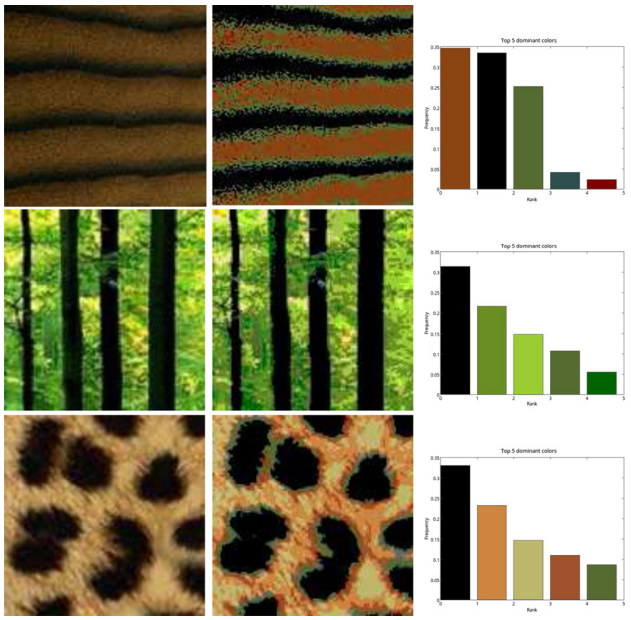

Fig. 7 Decide the two dominant colors in the pattern. Original images (left column); "color map" obtained after mapping each pixel to the palette (middle column); Histogram of "color map" (right column) (Color figure online)

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} \leq 1, a < b,$$

and for the vertical axis it is defined as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} \leq 1, a > b.$$

To form stripe patterns, the activator should be polarized parallel to and the inhibitor perpendicular to the stripes, whereas we define both activator and inhibitor as circles to generate mottled patterns. The parameter setup for different patterns is defined as follows:

- For horizontal patterns, the ellipse parameters of the activator morphogen and inhibitor morphogen are $a = 1, b = 2.5$ and $a = 3, b = 2.5$.
- For vertical patterns, the ellipse parameters of the activator morphogen and inhibitor morphogen are $a = 2.5, b = 1$ and $a = 2.5, b = 3$.
- For mottle patterns, the ellipse parameters of the activator morphogen and inhibitor morphogen are $a = 1.5, b = 1.5$ and $a = 2.5, b = 2.5$.

To test how the parameters influence the patterns formed, we set $f_+ = 1$ and change f_-. In Fig. 8, we show the simulation results of three patterns formed by Young's model with different f_- values. The plane of the simulation is 30 × 30 units and 2700 robots are positioned on the plane, that is, 3 robots per square unit. For example, the areas of activator

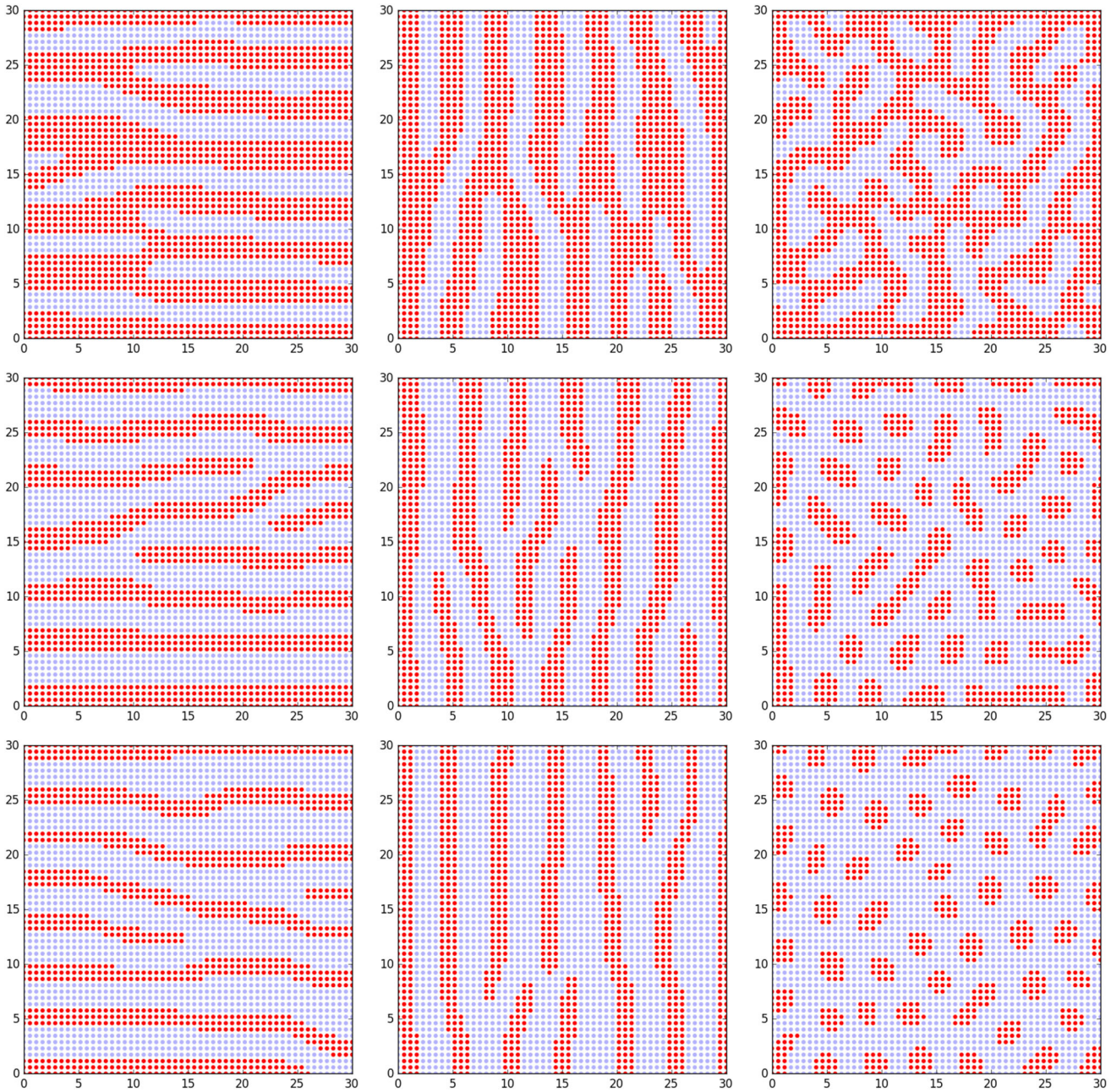

Fig. 8 Simulation results of field factors influencing the pattern formation. For horizontal patterns (left column), parameters of the activator morphogen and inhibitor morphogen are $a = 1, b = 2.5$ and $a = 3, b = 2.5, f_+ = 1, f_- = -0.5, -0.7, -0.9$ from top to bottom. For vertical patterns (middle column), parameters of the activator morphogen and inhibitor morphogen are $a = 2.5, b = 1$ and $a = 2.5, b = 3, f_+ = 1, f_- = -0.5, -0.7, -0.9$ from top to bottom. For mottle patterns (right column), parameters of the activator morphogen and inhibitor morphogen are $a = 1.5, b = 1.5$ and $a = 2.5, b = 2.5, f_- = -0.6, -0.7, -0.9$ from top to bottom. The 'On' robots are red while 'Off' robots are blue (Color figure online)

morphogen and inhibitor morphogen for horizontal patterns are $\pi \times 1 \times 2.5 \approx 7.9$ and $\pi \times 3 \times 2.5 - \pi \times 1 \times 2.5 \approx 15.7$, respectively, so there are approximately 24 robots contributing to the activator and 47 robots contributing to the inhibitor. Given a value of 'strength', the two morphogens interact with each other and reach an equilibrium state by forming a stable pattern. We can see that the red stripes in the stripe patterns get thinner as we increase $|f_-|$, which means increasing the strength of inhibitor morphogen, while the dots in the mottle patterns get clearer and smaller. In other words, the field strength factor f_- controls how much the stripes or dots of 'On' robots push away from each other. Increasing the value

of $|f_-|$ makes the stripes and dots push hard and faraway from each other, making stripes disconnected and dots isolated.

5.3 Distributed camouflage on grid graphs

We simulated the distributed camouflage system introduced above. By presenting some simulated results here, we hope to demonstrate that the algorithm's functionality and add clarity to the above explanation. We run tests with three images, one for each of the pattern types. Each image is 128×128 pixels and gray scale. We simulate 64 (8×8) robots.

Note that this grid of 8×8 robots is in many ways analogous to the sensor of a digital camera, albeit a camera with only 8×8 sensors and thus with very low resolution. If one were to recapture our test images with such a low resolution camera, many different pixels in the test image would contribute to the camera's output, resulting in a very blurry image. We therefore down sample the input image by taking the average of 16×16 pixel blocks. This blurred image is used as the color sensed by each robot for selecting the most likely pattern. For pattern formation, the initial 'On'/'Off' state is determined by making the blurred image binary (i.e. white and black). Figure 9 shows the entire process for the three input images.

Once the robots calculate the local pattern based on their sensed color and that of their neighbors, they need to achieve consensus on the global pattern. As has been discussed in Sect. 3.1, convergence of this value is guaranteed by enforcing a large number of iterations.

Next, the pattern generator described above is used (Sect. 3.5) with the activator and inhibitor regions seen in Fig. 6. The activator field value of $f_+ = 1$ was used, as suggested in (Young 1984). The inhibitor field value, f_-, is a parameter which gives rough control over what proportion of the robots are 'On' in the final pattern. We found that $f_- = -0.75$ gave qualitatively good results for all three of the pattern generators.

Finally, we start pattern formation with each robot's initial state to be 'On' or 'Off' status based on the sensed value. If the value is less than 127 we set it black, otherwise we set it white. The pattern generator runs for ten iterations. Robots on the image boundary use a reflection of their neighbors. For example, a robot on the top row, would count its bottom neighbor twice, as the top row is empty.

To further test the simulated algorithm, we added a simple noise model. For measurement error, instead of always assigning the appropriate color to a robot based on its position, we assign a uniformly random color with probability ρ_{meas}. For communication error, at each step in the algorithm where information from a neighbor is shared with a robot for a calculation (including the step where a robot's neighbors are calculated in the first place), a robot does not share this information with probability ρ_{comm}.

For a quantitative measure of the effects of error, we calculated the total absolute difference between the final generated pattern in the presence of error, and the final generated pattern without any error (as visible in the bottom row of Fig. 9). These results are charted in Fig. 10. Note that, with the 8×8 images used, a purely random image should give us a difference of 32, on average. The algorithm seems quite robust to errors of up to $0.15 - 0.2$. After these thresholds, the error increases sharply. (Results shown here are for the forest image, with other images yielding similar results.) Qualitatively, we observe that even as errors started to appear, many of the resulting patterns still looked 'good', i.e., still had prominent vertical stripes. The main determining factor as the probability of error increased seemed to be in the global pattern detection. If the correct pattern (vertical stripes in this case) is selected, the resulting pattern will fit well even with large errors. Correct pattern selection grows increasingly infrequent, however.

6 Hardware implementation on grid graphs

To validate the proposed distributed camouflage system (described in Sect. 4) and to understand the sorts of errors that real hardware introduces, we implemented the system described above on a swarm of "Droplets" (Farrow et al. 2014; Klingner et al. 2014). The Droplets are an open-source platform, with source code and manufacturing information available online. Each Droplet is roughly cylindrical with a radius of 2.2 cm and a height of 2.4 cm. The Droplets use an Atmel xMega128A3U micro-controller, and receive power via their legs through a floor with alternating strips of $+5V$ and GND. Each Droplet has six infrared emitters, receivers, and sensors, which are used for communication and for the range and bearing system (Farrow et al. 2014). The top of each board has sensors to detect the color and brightness of ambient light, and an RGB LED. Each droplet has a 16-bit unique ID. The measurements of color sensors are not accurate enough to distinguish close RGB values, in this hardware implementation we don't add the dominant color algorithm.

In our implementation, each Droplet maintains an array of neighbor's IDs. Messages are labeled with phase flags and attached with Droplets' IDs. The Droplets are synchronized using a firefly synchronization algorithm (Mirollo and Strogatz 1990; Werner-Allen et al. 2005). A simple TDMA protocol is used with 37 slots, each 271ms long. Each frame is thus 10s long. Each robot is assigned a slot based on its unique ID modulo 37. The number of slots (37) was chosen to be large enough such that the probability of two adjacent robots sharing a slot is low, but small enough that the algorithm runs quickly.

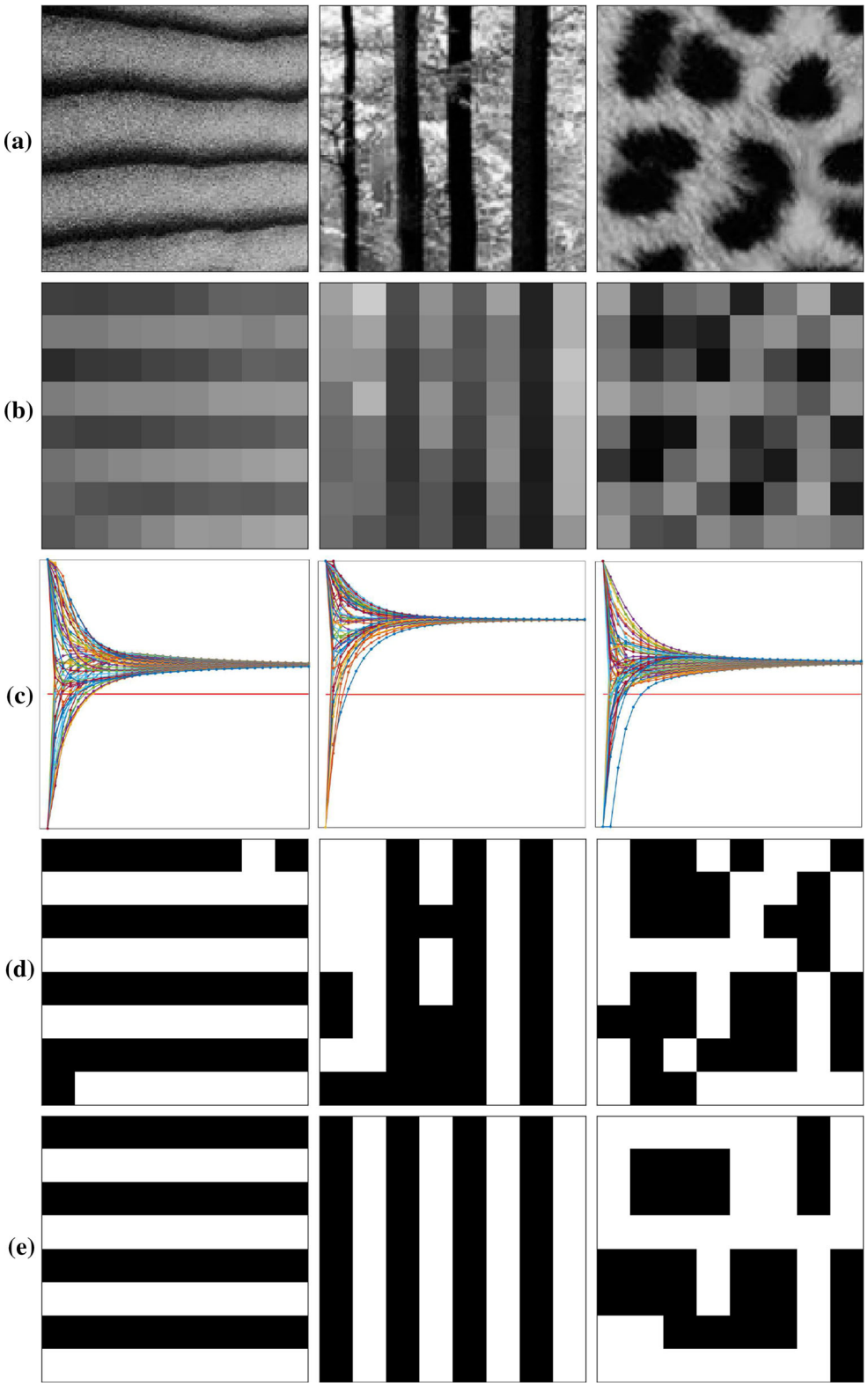

Fig. 9 The algorithm takes in the gray images (row **a**) and blurs them to images with 8×8 resolution (row **b**). These are the values sensed by each robot, and are used to calculate the pattern probabilities and choose the most probable local pattern. Row **c** shows consensus convergence for the most likely global pattern. For each of the charts in row **c**: the y axis shows pattern probability p from 0 to 1, and the x axis shows the number of steps taken from 0 to 35. Each robot maintains a probability vector $\{p^h, p^v, p^m\}$, however here we only plot the 'true' pattern probabilities for convenience. The red horizontal line marks $p = 0.5$. The blurred image from row **b** is converted to binary (row **d**) to get initial states for the pattern generator. Note that row **d** also represents the pattern that would be generated if each robot simply reproduced the color it sensed. Row **e** shows the resultant pattern (Color figure online)

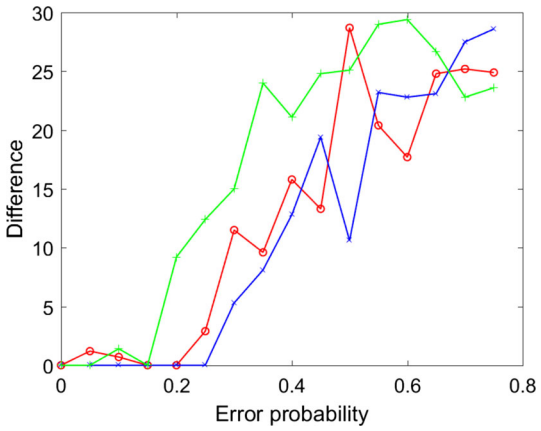

Fig. 10 The y-axis is the pixel difference from the 'correct' pattern and the x-axis is the error probability. The red line shows the effect of measurement errors (ρ_{meas}). The blue line shows the effect of communication errors (ρ_{comm}). The green line shows the effect of both measurement and communication errors ($\rho_{comm} = \rho_{meas}$). Each data point reflects the mean result over 10 trials of the forest image (Color figure online)

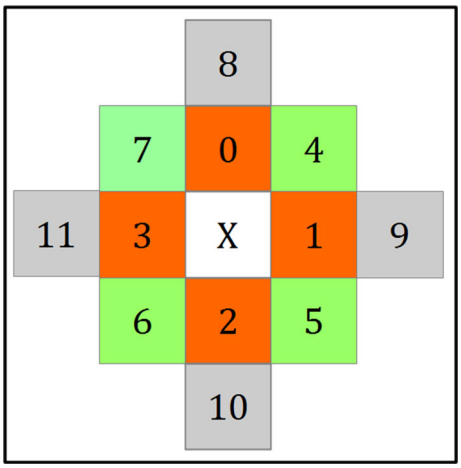

Fig. 11 Neighbor array. The orange neighbors(0–3) are used for pattern recognition; the green neighbors(4–7) are used in addition to the orange for pattern consensus. All pictured neighbors (orange, green and gray) are used for pattern formation (Color figure online)

In phase zero (neighbor identification), we initialize and configure the neighbor ID arrays which store neighboring Droplets' IDs. Range and bearing information is used to calculate positions for each Droplet's immediate neighbors, and neighbors of neighbors are learned by listening to the messages sent by neighboring Droplets. The positions of Droplets and their indices in the array are illustrated in Fig. 11. We allot 25 frames for phase zero to compute because the neighbor information is critical to the three phases.

In phase one (color sensing and recognition), each Droplet communicates the color it senses, and stores the colors its neighbors sense as learned through communications. Once this is complete, each (non-boundary) Droplet should know the ID and position of 12 neighbors, as well as those neighbors' sensed colors. With this information, each Droplet calculates an pattern probability array p as described in Sect. 3.3. This phase is allotted 10 frames to ensure each Droplet can finally receive its neighbors' color information.

In phase two (pattern consensus), each Droplet communicates its pattern probability array p and receives pattern probability arrays from its neighbors. At the end of each frame, each Droplet recalculates its current degree and updates pattern probability array according to the weighted-average consensus algorithm, as described in Sect. 3.1. Each 'step' of the consensus algorithm spans one frame. This phase is allotted 10 frames, which is enough for convergence of the global pattern.

In phase three (pattern formation), each Droplet communicates its intended color for the generated pattern and receives that information from neighboring Droplets. At the end of each frame, each Droplet updates its color in the generated pattern from corresponding Droplets. Each Droplet exchanges pattern color message with neighbors. At the end of each frame, each Droplet updates its pattern color according to the pattern formation algorithm described in Sect. 3.5. This phase is allotted 20 frames to achieve a stationary pattern.

7 Hardware results

The results of the hardware implementation of the distributed camouflage algorithm on desert-like, forest-like, and mottled white-black patterns are shown in Fig. 12. The results of the experiments are interesting because perfect horizontal and vertical patterns are maintained, and the mottled pattern is also formed despite the more-difficult-to-count failures in communication and color sensing. Several tests are done on each of the three patterns, the shown ones are representative as most of the patterns obtained are perfect.

The pattern probability convergence of 10 random sampled Droplets are shown in the middle row of Fig. 12. The weighted-average consensus algorithm is efficient as all of the Droplets achieve consensus on a global pattern within 5 iterations. To be specific, the initial value of each pattern probability plot is $0, 1$ or $1/3$ if the Droplet's color measurement is out of reasonable range. Different pattern-probability arrays might be obtained in different runs, but they always lead to the correct global patterns although there are Droplets that guessed incorrect, for example 4 for horizontal test, 3 for vertical pattern and 3 for mottled pattern. The final pattern probability array for the three patterns are $\{0.73, 0.04, 0.23\}$, $\{0.03, 0.63, 0.34\}$ and $\{0.32, 0.13, 0.55\}$. In other words, approximately 73%, 63%, and 55% of 64 Droplets make correct decision on the global pattern after convergence.

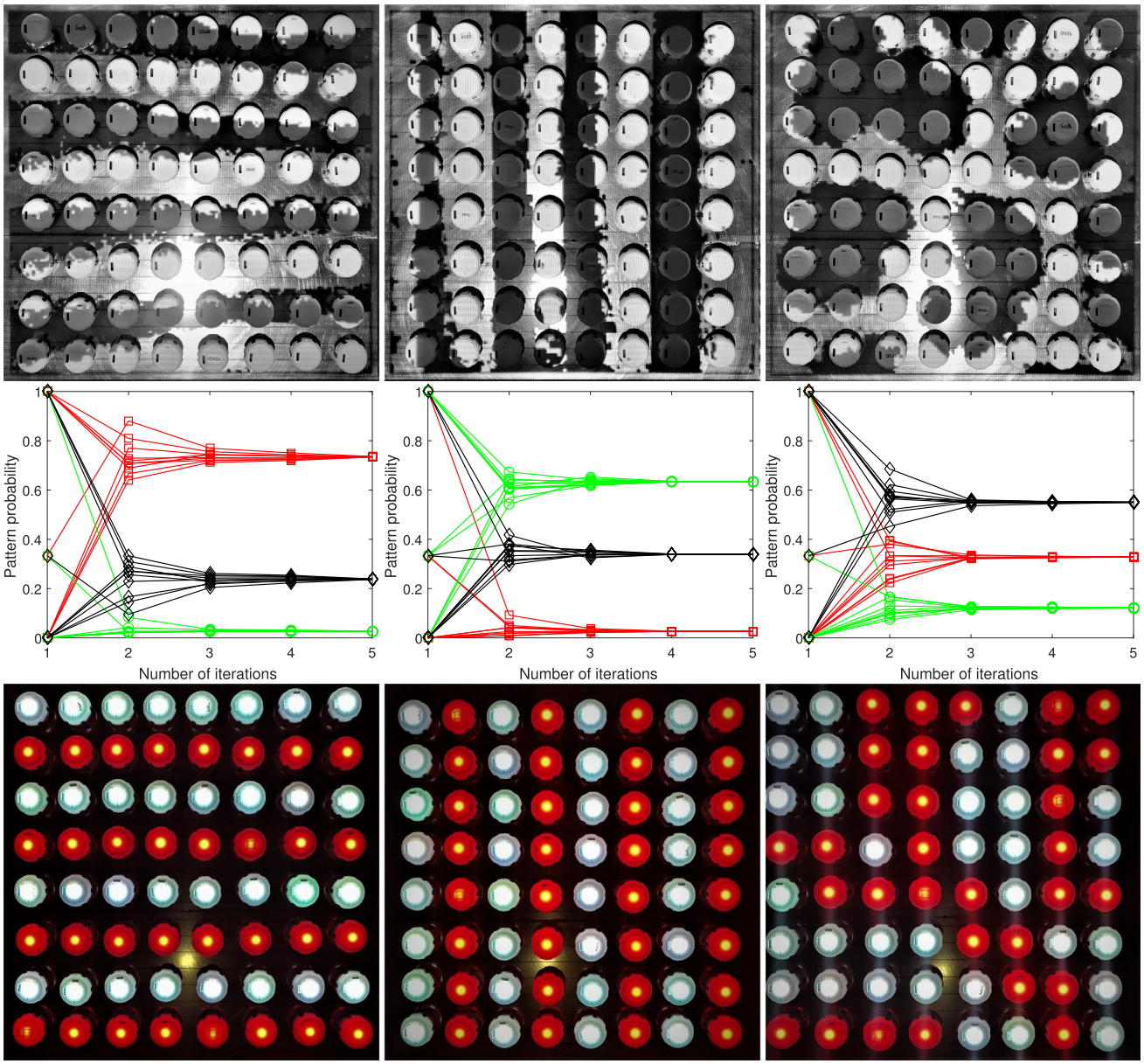

Fig. 12 Implementation of the distributed camouflage algorithm on Droplets. Desert-like pattern, forest-like pattern and mottled pattern are projected on the Droplets separately from left column to right column. From top row to bottom row are initial setup of Droplets with image projected (top), convergence plot of pattern probabilities of 10 randomly selected Droplets (middle) and final pattern formed (bottom). In the convergence plots, red □, green ○ and black ◇ represent the probabilities of horizontal, vertical and mottled patterns separately (Color figure online)

For all phases of this algorithm, the complexity of the computation performed by each Droplet is $\mathcal{O}(1)$; each Droplet only deals with information from a fixed, constant number of other Droplets. As is often the case for distributed algorithms, the complexity of the communication is less straightforward. Each Droplet communicates once per frame, so the total communication complexity of any given frame is $\mathcal{O}(N)$, where N is the total number of robots in the swarm. Consider the number of frames required for a given phase. For all phases except pattern consensus, information only needs to propagate through a fixed, constant number of robots, thus it remains fair to describe the overall communication complexity of these phases as $\mathcal{O}(N)$. The consensus phase, however, requires that information propagate through the entire swarm, thus the number of frames required for pattern consensus would have to increase as the size of the swarm increases. It is difficult to give a precise formulation of the rate of increase, as it depends on the connectivity

of the swarm's communication graph. When considering the communication complexity, it should be noted that although $\mathcal{O}(N)$ communications are required for some algorithm, it does not necessarily follow that the swarm will take $\mathcal{O}(N)$ time to complete. Each Droplet can only communicate to a finite range, so the number of communications, which can occur simultaneously, increases as the size of the swarm increases.

8 Discussion

We describe an end-to-end algorithmic system that processes environmental information to create an appropriate pattern and evaluate its performance on a swarm of 64 miniature robots. Albeit one might argue that simply "replaying" the recorded values might be sufficient to achieve similar results when considering results shown Fig. 12, the proposed approach is able to estimate characteristic patterns in complex scenes (Fig. 9), has been shown to be robust to sensing and communication noise, and allows the swarm as a whole to gain a high-level understanding of the scene it is in.

As communication on the Droplets is not perfectly reliable, the resultant patterns exhibit some random variations; they do not perfectly match simulation. Even these variations, however, will roughly follow the desired background pattern, seeming to bend or twist around the erroneous robot. In the future, we wish to test the algorithm on a more purpose-built hardware platform, which would allow for higher-resolution patterns, and extend the algorithm to include consensus on the dominant colors and patterns consisting of more than two colors.

We exclusively focus on distributed algorithms for pattern recognition and formation, which we believe to be complementary to centralized equivalents. We note, however, that the consensus steps are computationally much less efficient than the pattern formation algorithms when implemented on a distributed architecture. This is because consensus requires information to propagate through the entire system whereas pattern formation only requires access to information in the spatial vicinity on the order of the period of the repetitive pattern. Albeit consensus might be similarly limited, leading to the emergence of locally varying patterns, computation and communication are trade-offs that might shed light on the functioning of biological systems. For example, if pattern formation is indeed performed locally, we would expect nerve endings to radiate out from each chromatophore, covering an area that is proportional to the largest period of repetitive patterning an animal is able to display. Likewise, analysis of computational and communication complexity together with inspection of neural pathways might shed light on what information is communicated from an animal's brain to its skin.

It is also not clear what the role of light sensing nerves in the cuttlefish skin is. This paper performs pattern selection exclusively using such local sensors, which is definitely not the case in nature as the role of the brain in pattern selection is well understood. Local sensing might help, however, to switch between different patterns locally. In our current implementation, the entire system would converge to whatever the majority pattern is unless the range of consensus is limited. Performing experiments with mixed patterns that are exclusively perceived by the brain or the skin might shed light onto these mechanisms in the biological system, and in turn inform algorithm design.

We describe how the proposed method can be extended to recognize patterns of varying frequencies and orientations. For computational reasons, experiments are limited to only one frequency (assuming the period of all patterns to have unit length) and two orientations. We note that these computations are easily parallelize-able as they operate on the same neighborhood data.

9 Conclusion

We present a distributed camouflage system that mimics animal adaptive camouflage in a distributed way, from color sensing, pattern decision to pattern formation. In the system, a swarm of robots can sense their environment's color, recognize a local pattern, achieve consensus on a global pattern, and form a camouflage pattern consistent with the environment the robots are in. In our design, pattern descriptors are proposed for identifying local patterns. A weighted-average consensus algorithm is then used, allowing the swarm to converge to a global pattern. Finally, a pattern formation model, activator-inhibitor model, is applied to the swarm of robots to generate a pattern that matches the background. This is accomplished using local communication and simple mathematical operations.

We simulated the proposed algorithm on a couple of patterns from nature: a desert, a forest, and leopard skin. After completing the algorithm and successfully agreeing on a global pattern, the simulation results show that robots with erroneous color readings can correct themselves to match the global pattern. This is especially obvious for the horizontal and vertical patterns. We also tried to test the distributed algorithm by applying it on the Droplets swarm robotics platform. The result from the Droplets is promising since the robots can agree on the global pattern and show a proper matching pattern even if a few individual robots stop working.

We hope that the distributed camouflage system can provide a framework that serves as blueprint for researchers to further miniaturize the constituent components. In future work we are interested in further exploring potential avenues the proposed algorithms can suggest for the investigation of

biological systems. We are also interested in further reducing the size of the robots to be able to experiment with high-density deployments of larger numbers of robots, and relaxing some of the restrictions necessitated by the current generation of hardware.

Acknowledgements This research has been supported by NSF Grant #1150223 and by the Airforce Office of Scientific Research.

References

Bian, P., Jin, Y., & Zhang, Nr. (2010). Fuzzy c-means clustering based digital camouflage pattern design and its evaluation. In *Signal Processing (ICSP), 2010 IEEE 10th International Conference on, IEEE* (pp. 1017–1020).

Bradbury, J. W., & Vehrencamp, S. L., et al. (2011). *Principles of animal communication*. Sinauer Associates Sunderland.

Butera, W. J., & Bove, Jr. V. M. (2002). *Programming a paintable computer*. PhD thesis, Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences.

Chou, H. H., Nguyen, A., Chortos, A., To, J. W., Lu, C., Mei, J., Kurosawa, T., Bae, W. G., Tok, J. B. H., & Bao, Z. (2015). *A chameleon-inspired stretchable electronic skin with interactive colour changing controlled by tactile sensing*. Nature Communications 6.

Farrow, N., Klingner, J., Reishus, D., & Correll, N. (2014). Miniature six-channel range and bearing system: algorithm, analysis and experimental validation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (pp. 6180–6185).

Fekete, S. P., Fey, D., Komann, M., Kröller, A., Reichenbach, M., & Schmidt, C. (2009). Distributed vision with smart pixels. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry, ACM* (pp. 257–266).

Fishman, A., Rossiter, J., & Homer, M. (2015). Hiding the squid: Patterns in artificial cephalopod skin. *Journal of the Royal Society Interface*, *12*(108), 20150,281.

Grossberg, S., Mingolla, E., & Ross, W. D. (1997). Visual brain and visual perception: How does the cortex do perceptual grouping? *Trends in Neurosciences*, *20*(3), 106–111.

Hanlon, R. (2007). Cephalopod dynamic camouflage. *Current Biology*, *17*(11), R400–R404.

Hanlon, R. T., & Messenger, J. B. (1988). Adaptive coloration in young cuttlefish (sepia officinalis l.): The morphology and development of body patterns and their relation to behaviour. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *320*(1200), 437–487.

Inami, M., Kawakami, N., & Tachi, S. (2003). Optical camouflage using retro-reflective projection technology. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE Computer Society* (p. 348).

Jain, R., Kasturi, R., & Schunck, B. G. (1995). *Machine vision* (Vol. 5). New York: McGraw-Hill.

Kelman, E. J., Osorio, D., & Baddeley, R. J. (2008). A review of cuttlefish camouflage and object recognition and evidence for depth perception. *Journal of Experimental Biology*, *211*(11), 1757–1763.

Klingner, J., Kanakia, A., Farrow, N., Dustin, R., & Correll, N. (2014). A stick-slip omnidirectional drive-train for low-cost swarm robotics: Mechanism, calibration, and control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 846–851).

Kondo, S., & Miura, T. (2010). Reaction-diffusion model as a framework for understanding biological pattern formation. *Science*, *329*(5999), 1616–1620.

Lamme, V. A. (1995). The neurophysiology of figure-ground segregation in primary visual cortex. *Journal of Neuroscience*, *15*(2), 1605–1615.

Lasbury, M. E. (2017). Cloaking devices: Vanishing into thin space. In *The Realization of Star Trek Technologies*, Springer (pp. 35–66).

Lin, H. Y., Lie, W. N., & Wang, M. L. (2009). A framework of view-dependent planar scene active camouflage. *International Journal of Imaging Systems and Technology*, *19*(3), 167–174.

Meinhardt, H. (1982). *Models of biological pattern formation* (Vol. 6). London: Academic Press.

Meinhardt, H. (2009). *The algorithmic beauty of sea shells*. Berlin: Springer.

Meinhardt, H., & Gierer, A. (2000). Pattern formation by local self-activation and lateral inhibition. *Bioessays*, *22*(8), 753–760.

Messenger, J. B. (1977). Evidence that octopus is colour blind. *Journal of Experimental Biology*, *70*(1), 49–55.

Messenger, J. B. (2001). Cephalopod chromatophores: Neurobiology and natural history. *Biological Reviews*, *76*(4), 473–528.

Mirollo, R. E., & Strogatz, S. H. (1990). Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, *50*(6), 1645–1662.

Morin, S. A., Shepherd, R. F., Kwok, S. W., Stokes, A. A., Nemiroski, A., & Whitesides, G. M. (2012). Camouflage and display for soft machines. *Science*, *337*(6096), 828–832.

Otte, M. (2016). Collective cognition and sensing in robotic swarms via an emergent group-mind. In *International Symposium on Experimental Robotics*, Springer (pp. 829–840).

Peek, J. E., Hepfinger, L., Balma, R., Christopher, G., Fleuriet, J., Honke, T., Huebner, G., Mauer, E., Dotoli, P., & Ronconi, P. et al. (2006). *Guidelines for camouflage assessment using observers (instructions pour les evaluations de camouflage faisant appel a des observateurs)(cd-rom)*. Tech. rep., Nato Research and Technology Organization Neuilly-Sur-Seine (France).

Pendry, J. B., Schurig, D., & Smith, D. R. (2006). Controlling electromagnetic fields. *Science*, *312*(5781), 1780–1782.

Ramirez, M. D., & Oakley, T. H. (2015). Eye-independent, light-activated chromatophore expansion (lace) and expression of phototransduction genes in the skin of octopus bimaculoides. *Journal of Experimental Biology*, *218*(10), 1513–1520.

Rossiter, J., Yap, B., & Conn, A. (2012). Biomimetic chromatophores for camouflage and soft active surfaces. *Bioinspiration & Biomimetics*, *7*(3), 036,009.

Smith, D. R. (2014). A cloaking coating for murky media. *Science*, *345*(6195), 384–385.

Stevens, M., & Merilaita, S. (2009). Animal camouflage: Current issues and new perspectives. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1516), 423–427.

Stevens, M., & Merilaita, S. (2011). *Animal camouflage: Mechanisms and function*. Cambridge: Cambridge University Press.

Stevens, M., Cuthill, I. C., Windsor, A. M., & Walker, H. J. (2006). Disruptive contrast in animal camouflage. *Proceedings of the Royal Society of London B: Biological Sciences*, *273*(1600), 2433–2438.

Stuart-Fox, D., & Moussalli, A. (2009). Camouflage, communication and thermoregulation: Lessons from colour changing organisms. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1516), 463–470.

Toet, A. (2000). Technical evaluation report. In *Search and TargetAcquisition: NATO RTO Meeting Proceedings* (vol. 45).

Troscianko, T., Benton, C. P., Lovell, P. G., Tolhurst, D. J., & Pizlo, Z. (2009). Camouflage and visual perception. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *364*(1516), 449–461.

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *237*(641), 37–72.

Wang, Q., Gossweiler, G. R., Craig, S. L., & Zhao, X. (2014). *Cephalopod-inspired design of electro-mechano-chemically responsive elastomers for on-demand fluorescent patterning*. Nature Communications 5.

Webb, B. (2001). Can robots make good models of biological behaviour? *Behavioral and Brain Sciences*, *24*(06), 1033–1050.

Werner-Allen, G., Tewari, G., Patel, A., Welsh, M., & Nagpal, R. (2005). Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of the Third International Conference on Embedded Networked Sensor Systems, ACM* (pp. 142–153).

Xiao, L., Boyd, S., & Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on, IEEE* (pp. 63–70).

Ying, X. (2007). Camouflage color selection based on dominant color extraction. *Opto-Electronic Engineering*, *1*, 025.

Young, D. A. (1984). A local activator-inhibitor model of vertebrate skin patterns. *Mathematical Biosciences*, *72*(1), 51–58.

Yu, C., Li, Y., Zhang, X., Huang, X., Malyarchuk, V., Wang, S., et al. (2014). Adaptive optoelectronic camouflage systems with designs inspired by cephalopod skins. *Proceedings of the National Academy of Sciences*, *111*(36), 12,998–13,003.

Zylinski, S., Osorio, D., & Shohet, A. (2009). Perception of edges and visual texture in the camouflage of the common cuttlefish, sepia officinalis. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *364*(1516), 439–448.

Yang Li is a Ph.D. student of Computer Science at the University of Colorado at Boulder since 2015. He obtained BS and MS in Computer Science and Technology from East China Normal University in 2012 and 2015. Yang Li's research interests are swarm robotics and autonomous robots.

John Klingner has been a Ph.D. student at the Computer Science at University of Colorado Boulder since 2012. He obtained a bachelor's degree in Computer Science, Math, and Physics from Cornell College in 2012. John Klingner is the lead for the Droplets Swarm Robotics Platform (https://github.com/correlllab/cu-droplet) and his research interests are in the development of swarm robotics platforms, the algorithms that run on those platforms, and the interconnected relationship between the two.

Nikolaus Correll is an Associate Professor of Computer Science at the University of Colorado at Boulder. He obtained a master's degree in Electrical Engineering from the Swiss Federal Institute of Technology Zurich in 2003, a Ph.D. in Computer Science from EPFL in 2007, and was a postdoc at MIT CSAIL from 2007–2009. Nikolaus' research interests are materials that integrate sensing, actuation, computation, and communication with a focus on materials that make robots smart.