In N. Pillay, A. P. Engelbrecht, A. Abraham,
M. C. du Plessis, V. Snášel, A. K. Muda (Eds.):
Advances in Nature and Biologically Inspired Computing.
Advances in Intelligent Systems and Computing,
Vol. 419, 261–270, Springer, Cham, 2016.
The final publication is available at link.springer.com.

Modelling Image Processing with Discrete First-Order Swarms

Leif Bergerhoff and Joachim Weickert

Mathematical Image Analysis Group,
Faculty of Mathematics and Computer Science,
Campus E1.7, Saarland University, 66041 Saarbrücken, Germany
{bergerhoff,weickert}@mia.uni-saarland.de

Abstract. So far most applications of swarm behaviour in image analysis use swarms as models for *optimisation* tasks. In our paper, we follow a different philosophy and propose to exploit them as valuable tools for *modelling* image processing problems. To this end, we consider models of swarming that are individual-based and of first order. We show that a suitable adaptation of the potential forces allows us to model three classical image processing tasks: grey scale quantisation, contrast enhancement, and line detection. These proof-of-concept applications demonstrate that modelling image analysis tasks with swarms can be simple, intuitive, and highly flexible.

Keywords: swarms, image processing, dynamical systems, collective behaviour

1 Introduction

Understanding and simulating swarm behaviour continues to be an exciting interdisciplinary research area for more than six decades [2]. Numerous researchers have investigated models of swarming in such different fields as biology [4], computer science [16], mathematics [5], physics [22], and even philosophy [20]. For recent reviews and comparisons, we refer the reader to [7, 23] and the references therein. Generally, there exist two different model classes:

1. continuum / population-based / Eulerian / macroscopic models,
2. discrete / individual-based / Lagrangian / microscopic models.

Continuum models describe the evolution of a swarm's population density in space and time. These models provide a large scale description of general swarm attributes, but they cannot distinguish between individual swarm members.

Such a distinction requires discrete models that address each swarm member individually by characterising its position, velocity and other properties. Discrete models define simple rules that affect each individual. These rules are either based on the sociological behaviour of animals or – considering an artificial setup – motivated by a given task, which needs to be fulfilled. They control the attraction, repulsion, and orientation behaviour of the swarm members. Pairwise potentials model the effects between individuals; see e.g. [8] and the references therein for some commonly used potential functions. Integrating these rules into equations of motion describes the temporal evolution of the individuals. The literature distinguishes between first-order and second-order models [9]: First-order models use a set of equations that describe the velocities of each individual, while second-order models involve their acceleration.

Due to its heuristic character, it is common practice to apply discrete models to approximate solutions for difficult optimisation problems. Two well-known representatives are ant colony optimization (ACO) [3] and particle swarm optimization (PSO) [11]. By reducing problems to a pure *optimisation* task, ACO, PSO, and related models have already been applied numerous times in digital image analysis. However, apart from the idea of optimisation, discrete swarm methods have been used only rarely in order to *model* problems in the domain of image analysis. Notable exceptions deal with image halftoning [17], colour correction [19], segmentation [14], contour detection [12], boundary identification and tracking [15, 21], and the detection of fibre pathways [1]. Most of these modelling applications are fairly new and show convincing performance. However, all of these authors have focussed on a specific application. It seems that they have not been interested in exploiting the genericity behind the models of swarming.

Goals of our Paper. Motivated by these recent encouraging results, the goal of our article is to present novel applications of discrete first-order models of swarming in image analysis. To this end, we define behavioural rules for three fairly different image processing problems: grey scale quantisation, contrast enhancement, and line detection with the Hough transform. In all scenarios we use essentially the same model and modify only some of its features. This emphasises the versatility and genericity of models of swarming.

Paper Structure. Section 2 reviews the modelling of swarm behaviour in a discrete setup. We present our different behavioural rules and the corresponding potentials, and we discuss some model characteristics and a time discretisation. Section 3 adapts this modelling framework to three different applications in image processing and shows experimental results. Section 4 summarises our contributions and gives an outlook to future work.

2 Discrete Modelling of Swarm Behaviour

Basic Notations and Definitions. We consider a set $S = \{A_i \,|\, i = 1, ..., N\}$, called *swarm*, which is composed of N *agents* A_i. In the following, we use the terms agent, *particle*, and *individual* interchangeably. By $\boldsymbol{x_i} \in \mathbb{R}^d$ we denote the position of an individual A_i, and $\boldsymbol{v_i} \in \mathbb{R}^d$ describes its velocity. Both the particle position and its velocity are functions over time $t \in [0, \infty)$:

$$\boldsymbol{x_i} = \boldsymbol{x_i}\,(t)\,, \qquad \boldsymbol{v_i} = \boldsymbol{v_i}\,(t)\,. \tag{1}$$

If the agents are intended to have a limited field of perception, many discrete models such as [16] make use of a disk-shaped neighbourhood with radius δ. For an agent A_i, it is given by

$$\mathcal{N}_{i,\delta}\,(t) \;=\; \{A_j \in S \,|\, j \neq i,\, |\boldsymbol{x_i} - \boldsymbol{x_j}| \leq \delta\} \tag{2}$$

where $|\,.\,|$ denotes the Euclidean norm. If the neighbourhoods $\mathcal{N}_{i,\delta}\,(t)$ contain all swarm mates A_j for all times t, a model is said to be *global*. Otherwise, it is called *local*.

Potential Energies and Forces. To describe a desired collective behaviour, discrete models of swarming define update rules for the positions and velocities of their individuals. These rules include effects based on attractive, repulsive, and orientating behaviour among agents [4, 5, 16, 22], as well as on the environment [11], or a combination of both [8, 17]. In our paper, we restrict ourselves to the treatment of attraction and repulsion among the agents.

The influence of the swarm mates on an agent A_i is described by a pairwise function $W : \mathbb{R}^d \to \mathbb{R}$ that denotes the *potential energy*. The total potential energy of the swarm is given by

$$E_{pot}\,(S) \;=\; \frac{1}{2} \sum_{A_i \in S} \; \sum_{A_j \in S \setminus \{A_i\}} W(\boldsymbol{x_i} - \boldsymbol{x_j}). \tag{3}$$

Our potential functions model either attraction,

$$W_a(\boldsymbol{x_i} - \boldsymbol{x_j}) \;=\; \frac{1}{2} \cdot |\boldsymbol{x_i} - \boldsymbol{x_j}|^2\,, \tag{4}$$

or repulsion,

$$W_r(\boldsymbol{x_i} - \boldsymbol{x_j}) \;=\; \frac{c^2}{2} \cdot \exp\left(-\frac{|\boldsymbol{x_i} - \boldsymbol{x_j}|^2}{c^2}\right) \qquad (c \neq 0), \tag{5}$$

where c serves as a spatial scale of repulsion. If we compute the partial derivative w.r.t. $\boldsymbol{x_i}$, we arrive at the *potential forces* $-\boldsymbol{\nabla}_{\boldsymbol{x_i}} W : \mathbb{R}^d \to \mathbb{R}^d$ that act on the agent A_i:

$$-\boldsymbol{\nabla}_{\boldsymbol{x_i}} W_a(\boldsymbol{x_i} - \boldsymbol{x_j}) \;=\; -\,(\boldsymbol{x_i} - \boldsymbol{x_j})\,, \tag{6}$$

$$-\boldsymbol{\nabla}_{\boldsymbol{x_i}} W_r(\boldsymbol{x_i} - \boldsymbol{x_j}) \;=\; \exp\left(-\frac{|\boldsymbol{x_i} - \boldsymbol{x_j}|^2}{c^2}\right) \cdot (\boldsymbol{x_i} - \boldsymbol{x_j}). \tag{7}$$

For further details on potential functions and their use in discrete models of swarming, we refer the reader to [8].

First-Order Models. First-order models are based on the (physically simplified [10]) assumption that the particle velocity $\boldsymbol{v_i}$ can be expressed in terms of the potential forces $-\boldsymbol{\nabla}_{\boldsymbol{x_i}} W$:

$$\frac{d\boldsymbol{x_i}(t)}{dt} \;=\; \boldsymbol{v_i}(t) \;=\; -\sum_{A_j \in S\setminus\{A_i\}} \boldsymbol{\nabla}_{\boldsymbol{x_i}} W(\boldsymbol{x_i}(t) - \boldsymbol{x_j}(t)) \qquad \forall A_i \in S, \tag{8}$$

where we assume that we know the initial state at time $t = 0$.

Time Discretisation. Since we cannot expect to find an analytical solution to the dynamical system (8), we have to approximate it numerically on the computer. This requires to discretise it in time.

Let $\tau > 0$ denote some time step size, and let $t_k := k\tau$. Moreover, we abbreviate $\boldsymbol{x_i}(t_k)$ by $\boldsymbol{x_i^k}$. The simplest time discretisation of Equation 8 approximates the time derivative by its forward difference:

$$\frac{d\boldsymbol{x_i^k}}{dt} \;\approx\; \frac{\boldsymbol{x_i^{k+1}} - \boldsymbol{x_i^k}}{\tau}\;. \tag{9}$$

This turns (8) into the following explicit update scheme:

$$\boldsymbol{x_i^{k+1}} \;=\; \boldsymbol{x_i^k} \;-\; \tau \cdot \sum_{A_j \in S\setminus\{A_i\}} \boldsymbol{\nabla}_{\boldsymbol{x_i^k}} W\left(\boldsymbol{x_i^k} - \boldsymbol{x_j^k}\right) \qquad (k = 0, 1, ...) \tag{10}$$

with some appropriate initialisation $\boldsymbol{x_i^0}$ for all $A_i \in S$.
If we restrict the interactions of agent A_i to its δ-neighbourhood $\mathcal{N}_{i,\delta}(t_k)$ from (2), we can replace (10) by the local update rule

$$\boldsymbol{x_i^{k+1}} \;=\; \boldsymbol{x_i^k} \;-\; \tau \cdot \sum_{A_j \in \mathcal{N}_{i,\delta}^k} \boldsymbol{\nabla}_{\boldsymbol{x_i^k}} W(\boldsymbol{x_i^k} - \boldsymbol{x_j^k}). \tag{11}$$

It is well-known from the theory of numerical methods for differential equations that such explicit schemes may require a fairly small time step size τ in order to be stable [13], in particular if the right hand side fluctuates strongly w.r.t. its argument.[1]

The experiments below make use of the explicit schemes (10) and (11) with the potential forces $\boldsymbol{\nabla}_{\boldsymbol{x_i^k}} W$ given by (6) or (7).

[1] If this becomes too time-consuming, one can also consider more efficient, so-called implicit schemes [13]. However, they require to solve linear or nonlinear systems of equations.

3 Application to Image Processing Problems

Let us now apply our discrete first-order model of swarming to three different image processing problems: grey scale quantisation, contrast enhancement, and line detection. This requires to interpret the specific image processing problem in terms of swarming agents, and to specify our potential forces in an appropriate way.

In our application scenarios, we consider a *digital greyscale image* f that is discrete in its domain and its codomain. The domain contains n_x equally spaced pixels in x-direction and n_y pixels in y-direction. The grey value range is given by the set $\{0, ..., 255\}$, which results from a bytewise encoding:

$$f : \{1, ..., n_x\} \times \{1, ..., n_y\} \;\rightarrow\; \{0, ..., 255\}. \tag{12}$$

For such a two-dimensional greyscale image f, its *histogram* $h[f](k)$ counts how often each grey value $k \in \{0, ..., 255\}$ is attained. Thus, $h[f]$ is a one-dimensional function from $\{0, ..., 255\}$ to $\mathbb{N}_0$.

3.1 Grey Scale Quantisation

The discretisation of the codomain of an image is called *quantisation*. Obviously, the number of different greyscales in an image determines how expensive it is to store them: While 256 different values require a full byte, 8 values can be encoded already with 3 bits. Since humans cannot distinguish many greyscales, one can compress image data without severe visual degradations by reducing the number of quantisation levels.

To design a model of swarming for obtaining a coarser quantisation of some digital greyscale image f, we proceed as follows. We consider its histogram $h[f]$ and identify some histogram value $h[f](n) = c_n$ with c_n agents sharing the same position $x_i = n$. Thus, we have a one-dimensional model of swarming. Note that multiple agents that share the same position have to undergo the same joint motion. This reduces the computational complexity in a substantial way: The computational effort becomes proportional to the number of greyscales instead of the number of pixels.

In order to cluster multiple quantisation levels into a single level, we use the linear attraction force from (6). As we will see below, it makes sense to localise the interaction to a δ-neighbourhood, which requires the update scheme (11).

In our quantisation experiments we have chosen $\tau = 10^{-5}$. This leads to a stable steady state solution after at most $4 \cdot 10^4$ iterations. For a 512×512 image, this can be accomplished in far less than one minute on a single core of a standard PC. Fig. 1 illustrates the effect of our model of swarming for different δ values. We observe that increasing δ reduces the number q of quantisation levels. Interestingly there seems to be an almost inverse relation, such that $2\delta q$ is roughly equal to the length of the original greyscale interval (255 in our case).

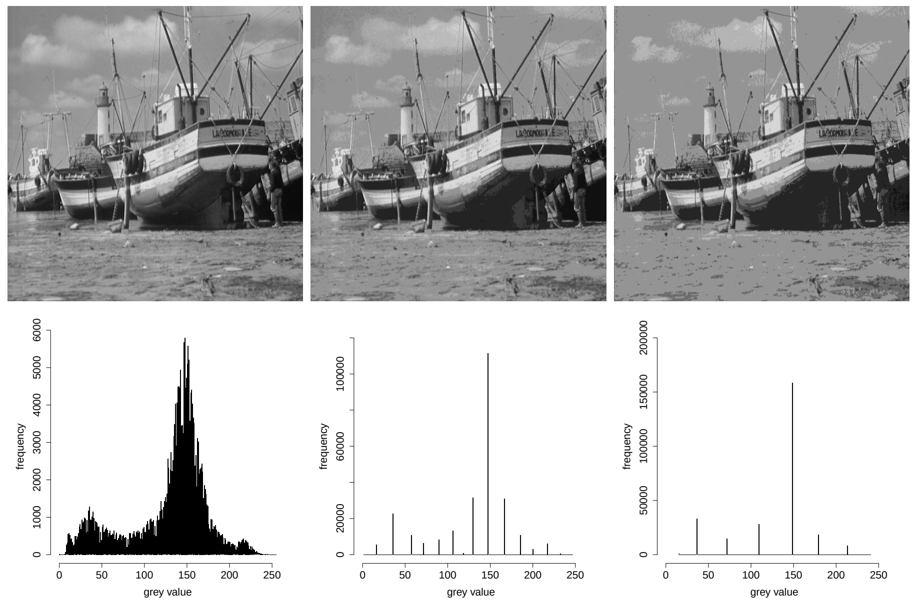

Fig. 1. Swarm-based image quantisation. **Top, from left to right: (a)** Original image from [18], 512×512 pixels, $q = 255$ greyscales. **(b)** Swarm-based quantisation with $\delta = 8$, yielding $q = 16$ greyscales. **(c)** $\delta = 16$, $q = 8$ greyscales. **Bottom, from left to right: (d)–(f)** Corresponding histograms.

This suggests that our model of swarming clusters the grey scales into q bins of approximately[2] the same size 2δ. Note that the interval length 2δ is the diameter of the neighbourhood $\mathcal{N}_{i,\delta}$. Hence, the model of swarming can be interpreted and handled in a very intuitive way.

3.2 Contrast Enhancement

The contrast of an image is characterised by the modulus of the difference between the greyvalues of neighbouring pixels. For recognising interesting image structures, their contrast should be sufficiently high. This may require some preprocessing that enhances the image contrast.

Let us now adapt our model of swarming to this application. To this end, it is sufficient to find a mapping of the greyvalues that yields a better contrast. As before, we consider the histogram $h[f]$ of the image f, and we assign c_n agents to a grey value n if $h[f](n) = c_n$. However, since we want to increase the global contrast this time, we use the global explicit scheme (10), and we equip

[2] It is clear from the structure of our approach and the experiments that the quantisation levels depend on the actual image histogram and are not necessarily equidistant.

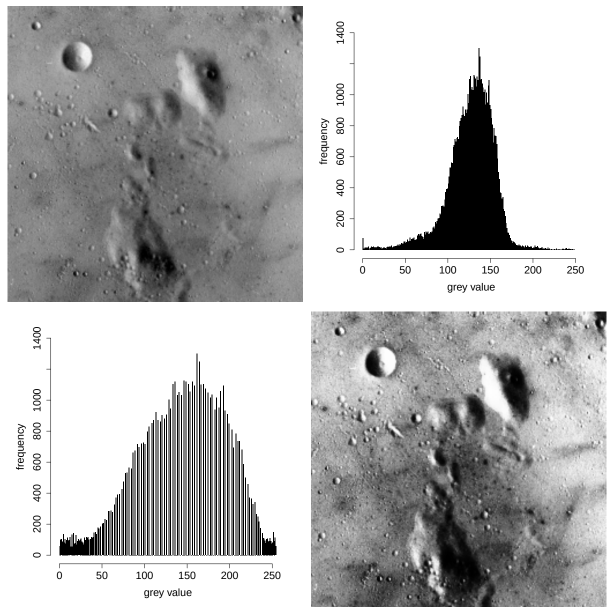

Fig. 2. Swarm-based contrast enhancement. **(a) Top left:** Moon surface image from [18], 256×256 pixels. **(b) Top right:** Its histogram. **(c) Bottom left:** After swarm-based histogram enhancement with $c = 1$, and $2 \cdot 10^6$ iterations with step size $\tau = 10^{-3}$. **(d) Bottom right:** Enhanced image using the grey values from the transformed histogram.

it with the repulsion forces from (7). Moreover, we employ reflecting boundary conditions to prevent that agents leave the admissible greyscale range $[0, 255]$. For $t \to \infty$, the swarm converges to a steady state distribution, where the grade of contrast enhancement grows with the repulsion parameter c. Once the histogram is enhanced, one simply replaces the grey values of the image by their transformed values.

Figure 2 illustrates this procedure, where the evolution reaches a steady state. We observe a clear visual contrast improvement of the test image.This is also confirmed quantitatively by its standard deviation, which has increased from 27.74 to 56.86.

3.3 Line Detection

Our third application scenario for models of swarming is concerned with another important image processing problem, the detection of lines. Our goal is to improve a classical method which is based on the so-called Hough transform [6].

The basic idea behind line detection with the Hough transform is as follows. For some greyscale image f, one searches for locations that may lie on significant lines by computing the gradient magnitude $|\boldsymbol{\nabla} f|$. For a digital image, this requires finite difference approximations. A location is significant if its gradient magnitude exceeds a certain threshold T_g. In a next step, the line candidate pixels vote for all lines that pass through them. All lines through a pixel (x, y) satisfy the normal representation

$$\rho \;=\; x \cdot \cos\theta + y \cdot \sin\theta, \tag{13}$$

where θ denotes the angle between the line normal and the x-axis, and ρ is the distance to the origin. Thus, a candidate point is mapped to a trigonometric curve $\rho(\theta)$ in the Hough space (θ, ρ). If n candidate points lie on a line with parameters $(\tilde{\theta}, \tilde{\rho})$, then their corresponding n trigonometric curves in Hough space intersect in $(\tilde{\theta}, \tilde{\rho})$. Therefore, one can find lines in the input image f by searching for clustering points in its Hough space: One discretises the Hough space (θ, ρ), and each trigonometric curve votes for all cells that it crosses. The cells with the most votes characterise the most significant lines in the original image. Typically one finds these clustering points by applying a threshold T_a on the votes in Hough space.

While this sounds nice in theory, in practice it is not easy to find appropriate thresholds that avoid false negatives and false positives. Also the bin size of the discrete Hough space is problematic: If the discretisation is too fine, it is unlikely that many votes will fall in the same cell. If it is too coarse, the line parameters are prone to imprecisions.

As a remedy, we propose the following procedure. First we consider a relatively fine discretsation in Hough space and threshold the votes. Afterwards we process the surviving votes with a swarm-based clustering. To this end, we set up n agents at every position (ρ, θ) that received n votes. Note that in contrast to our clustering for quantisation – which took place in the one-dimensional histogram space – this is a two-dimensional clustering. In analogy to the quantisation setting, we use the linear attraction force (6) within the localised update scheme (11), and compute its steady state.

Fig. 3 shows how this works in a real-world setting. We observe that the classical Hough transform suffers from the fact that lines in the image cluster in several adjacent cells in Hough space. As a consequence, we obtain a bundle of almost parallel lines instead of a single line. Our swarm-based clustering in Hough space is well-suited to solve this problem, since votes from the neighbours move towards the local centroids. In this way they sharpen the clusters and avoid multiple almost parallel lines.

4 Conclusions

Our paper shows that discrete first-order models of swarming have a high potential in image processing that goes far beyond classical applications as tools for difficult *optimisation* tasks: By means of three proof-of-concept applications

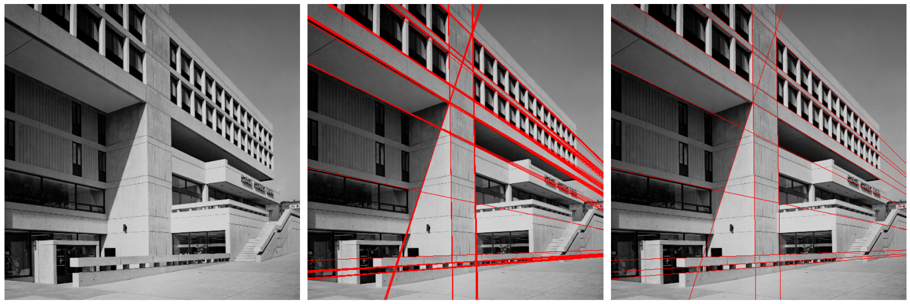

Fig. 3. Swarm-based line detection. **(a) Left:** Test image, 512×512 pixels. **(b) Middle:** 71 lines detected with the Hough transform. ($T_g = 19$, $T_a = 244$). **(c) Right:** 13 lines detected with the Hough transform with swarm-based postprocessing ($T_g = 19$, $T_a = 244$, $\delta = 5$, $\tau = 10^{-4}$, 300 iterations).

we have demonstrated their usefulness as powerful *modelling* methods. The fact that these applications serve fairly different goals underlines the genericity of the swarm-based paradigm: It is a highly versatile framework that can be adapted in an intuitive way to a broad spectrum of problems.

In our ongoing work, we intend to study more efficient numerical algorithms, evaluate different ways to incorporate neighbourhood information, equip our models with more problem-specific features, and compare them to non-swarm based approaches. Last but not least, we will also study other models of swarming and apply them to further problems in the broad area of visual computing.

Acknowledgement. Our research activities have been supported financially by the *Deutsche Forschungsgemeinschaft (DFG)* through a *Gottfried Wilhelm Leibniz Prize*. This is gratefully acknowledged.

References

1. Aranda, R., Rivera, M., Ramirez-Manzanares, A.: A flocking based method for brain tractography. Medical Image Analysis 18(3), 515–530 (Apr 2014)
2. Clark, P.J., Evans, F.C.: Distance to nearest neighbor as a measure of spatial relationships in populations. Ecology 35(4), 445–453 (Oct 1954)
3. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by ant colonies. In: Proceedings of the First European Conference On Artificial Life. pp. 134–142. Paris, France (Dec 1991)
4. Couzin, I.D., Krause, J., James, R., Ruxton, G.D., Franks, N.R.: Collective memory and spatial sorting in animal groups. Journal of Theoretical Biology 218(1), 1–11 (Sep 2002)
5. Cucker, F., Smale, S.: Emergent behavior in flocks. IEEE Transactions on Automatic Control 52(5), 852–862 (May 2007)

6. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM 15(1), 11–15 (Jan 1972)
7. Fine, B.T., Shell, D.A.: Unifying microscopic flocking motion models for virtual, robotic, and biological flock members. Autonomous Robots 35(2-3), 195–219 (Oct 2013)
8. Gazi, V.: On Lagrangian dynamics based modeling of swarm behavior. Physica D: Nonlinear Phenomena 260, 159–175 (Oct 2013)
9. Gazi, V., Fidan, B.: Coordination and control of multi-agent dynamic systems: Models and approaches. In: Sahin, E., Spears, W.M., Winfield, A.F.T. (eds.) Swarm Robotics, Lecture Notes in Computer Science, vol. 4433, pp. 71–102. Springer, Berlin (2007)
10. Gazi, V., Passino, K.M.: Stability analysis of social foraging swarms. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34(1), 539–557 (Feb 2004)
11. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks. vol. 4, pp. 1942–1948. Perth, WA (Nov 1995)
12. Kirchmaier, U., Hawe, S., Diepold, K.: A swarm intelligence inspired algorithm for contour detection in images. Applied Soft Computing 13(6), 3118–3129 (Jun 2013)
13. LeVeque, R.J.: Finite Difference Methods for Ordinary and Partial Differential Equations. SIAM, Philadelphia (2007)
14. Liu, J., Tang, Y.Y.: Adaptive image segmentation with distributed behavior-based agents. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(6), 544–551 (Jun 1999)
15. Marthaler, D., Bertozzi, A.L.: Tracking environmental level sets with autonomous vehicles. In: Recent Developments in Cooperative Control and Optimization, Cooperative Systems, vol. 3, pp. 317–332. Springer, New York (2004)
16. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. ACM SIGGRAPH Computer Graphics 21(4), 25–34 (Aug 1987)
17. Schmaltz, C., Gwosdek, P., Weickert, J.: Electrostatic halftoning. Computer Graphics Forum 29(8), 2313–2327 (Dec 2010)
18. Signal and Image Processing Institute of the University of Southern California: The USC-SIPI image database. http://sipi.usc.edu/database/ (2015), last visited August 16, 2015
19. Simone, G., Audino, G., Farup, I., Albregtsen, F., Rizzi, A.: Termite retinex: a new implementation based on a colony of intelligent agents. Journal of Electronic Imaging 23(1), 013006 (2014)
20. Sumpter, D.J.T.: The principles of collective animal behaviour. Philosophical Transactions of the Royal Society B: Biological Sciences 361(1465), 5–22 (Nov 2005)
21. Triandaf, I., Schwartz, I.B.: A collective motion algorithm for tracking time-dependent boundaries. Mathematics and Computers in Simulation 70(4), 187–202 (Dec 2005)
22. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., Shochet, O.: Novel type of phase transition in a system of self-driven particles. Physical Review Letters 75, 1226–1229 (Aug 1995)
23. Vicsek, T., Zafeiris, A.: Collective motion. Physics Reports 517(3-4), 71–140 (Aug 2012)