J Math Imaging Vis (2017) 57:269–291
DOI 10.1007/s10851-016-0678-0

Graph Clustering, Variational Image Segmentation Methods and Hough Transform Scale Detection for Object Measurement in Images

Luca Calatroni[1] · **Yves van Gennip**[2] · **Carola-Bibiane Schönlieb**[3] · **Hannah M. Rowland**[4,5] · **Arjuna Flenner**[6]

Received: 25 February 2016 / Accepted: 11 July 2016 / Published online: 25 July 2016

Abstract We consider the problem of scale detection in images where a region of interest is present together with a measurement tool (e.g. a ruler). For the segmentation part, we focus on the graph-based method presented in Bertozzi and Flenner (Multiscale Model Simul 10(3):1090–1118, 2012) which reinterprets classical continuous Ginzburg–Landau minimisation models in a totally discrete framework. To overcome the numerical difficulties due to the large size of the images considered, we use matrix completion and splitting techniques. The scale on the measurement tool is detected via a Hough transform-based algorithm. The method is then applied to some measurement tasks arising in real-world applications such as zoology, medicine and archaeology.

Keywords Graph clustering · Discrete Ginzburg–Landau functional · Image segmentation · Scale detection · Hough transform

✉ Luca Calatroni
calatroni@dima.unige.it

Yves van Gennip
y.vangennip@nottingham.ac.uk

Carola-Bibiane Schönlieb
cbs31@cam.ac.uk

Hannah M. Rowland
hr325@cam.ac.uk

Arjuna Flenner
arjuna.flenner@navy.mil

1 MIDA Group, Dipartimento di Matematica, Universitá degli studi di Genova, Via Dodecaneso 35, 16146 Genova, Italy

2 School of Mathematical Sciences, The University of Nottingham, University Park, Nottingham NG7 2RD, UK

3 Department of Applied Mathematics and Theoretical Physics (DAMTP), University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, UK

4 Department of Zoology, University of Cambridge, Downing Street, Cambridge CB2 3EJ, UK

5 Institute of Zoology, Zoological Society of London, Regents Park, London NW1 4RY, UK

6 Image and Signal Processing Branch, NAVAIR, 1 Administrative Circle, China Lake, CA, USA

1 Introduction

Image segmentation denotes the task of partitioning an image in its constituent parts. Feature-based segmentation looks at distinctive characteristics (features) in the image, grouping similar pixels into clusters which are meaningful for the application at hand. Typical examples of features are based on greyscale/RGB intensity and texture. Mathematical methods for image segmentation are mainly formalised in terms of variational problems in which the segmented image is a minimiser of an energy. The most common image feature encoded in such energies is the magnitude of the image gradient, detecting regions (or contours) where sharp variations of the intensity values occur. Examples include the Mumford–Shah segmentation approach [51], the snakes and geodesic active contour models [16,38]. Moreover, in [18], Chan and Vese proposed an instance of the Mumford–Shah model for piecewise constant images whose energy is based on the mean greyvalues of the image inside and outside of the segmented region rather than the image gradient and hence does not require strong edges for segmentation. The Chan–Vese model has been extended for vector-valued images such as RGB images in [19]. Other image segmentation methods have been considered in [26,39]. They rely on the use of the total variation (TV) semi-norm [5], which is commonly used for image processing tasks due to its properties of simultaneous edge preservation and smoothing (see [55]).

The non-smoothness of most of the segmentation energies renders their numerical minimisation usually difficult. In the case of the Mumford–Shah segmentation model, the numerical realisation is additionally complicated by its dependency on the image function as well as the object contour. To overcome this, several regularisation methods and approximations have been proposed in the literature, e.g. [4,10,11,66] for Mumford–Shah segmentation. In the context of TV-based segmentation models, the Ginzburg–Landau functional has an important role. Originally considered for the modelling of physical phenomena such as phase transition and phase separation (cf. [12] for a survey on the topics), it is used in imaging for approximating the TV energy. Some examples of the use of this functional in the context of image processing are [24–26], which relate to previous works by Ambrosio and Tortorelli [4,5] on diffuse interface approximation models.

Such variational methods for image segmentation have been extensively studied from an analytical point of view, and the segmentation is usually robust and computationally efficient. However, variational image segmentation as described above still faces many problems in the presence of low contrast and the absence of clear boundaries separating regions. Their main drawback is that they are limited to image features which can be mathematically formalised (e.g. in terms of an image gradient) and encoded within a segmentation energy. In recent years, dictionary-based methods have become more and more popular in the image processing community, complementing more classical variational segmentation methods. By learning the distinctive features of the region to be segmented from examples provided by the user, these methods are able to segment the desired regions in the image correctly.

In this work, we consider the method proposed in [9,28,42,43] for image segmentation and labelling. This approach goes beyond the standard variational approach in two respects. Firstly, the model is set up in the purely discrete framework of graphs. This is rather unusual for variational models where one normally considers functionals and function spaces defined on subdomains of $\mathbb{R}^2$ in order to exploit properties and tools from convex and functional analysis and calculus of variations. Secondly, the new framework allows for more flexibility in terms of the features considered. Additional features such as texture, light intensity can be considered as well without encoding them in the function space or the regularity of the functions. Due to the possibly very large size of the image (nowadays of the order of megapixel for professional cameras) and the large number of features considered, the construction of the problem may be computationally expensive and often requires reduction techniques [27,52,53]. In several papers (see, for example, [33,59,61]), the segmentation problem was rephrased in the graph framework by means of the graph cut objective function. Follow-up works on the use of graph-based approaches are, for instance, [44,45] where an iterative application of heat diffusion and thresholding, also known as the Merriman–Bence–Osher (MBO) method [46], is discussed for binary image labelling, and [36] where the Mumford–Shah model is reinterpreted in a graph setting.

In this paper, we also address the problem of detection of objects with geometrical properties that are a priori known. An example is the detection of lines and circles. These objects can be identified by mapping them onto an auxiliary space where relevant geometrical properties (such as linear alignment and roundness) are represented as peaks of specific auxiliary functions. In this work, we use the Hough transform [35] to detect measurement tools (rulers, concentric circles of fixed radii) with the intent of providing quantitative, scale-independent measurements of the region segmented by one of the techniques described above. In this way, an absolute measurement of the region of interest in the image is possible, independent of the scale of the image, which could depend, for instance, on the distance of the objective to the camera.

We demonstrate the use of our method in the context of real-world applications in which segmentation and subsequent object measurement are crucial. Our main application is the measurement of the white forehead patch (blaze) of male pied flycatchers, which has been studied with regard to sexual selection in [58], see Fig. 1. The forehead patch is known to vary between individuals [40] and can be subject to both intra- [37] and intersexual [54] selection with pied flycatchers from Spain preferring males with large patches. Forehead patch size has been shown to signal male phenotypic quality through plasma oxidative stress and antioxidant capacity [50]. However, in all studies to date the measurements of patches have been inconsistent and generally inaccurate. For example, some studies have simply measured patch height [22], whereas Potti and Montalvo [54] assumed the shape to be a trapezium with area equal to $0.5(B+b)H$, B being the white patch width, b the bill width and H the height

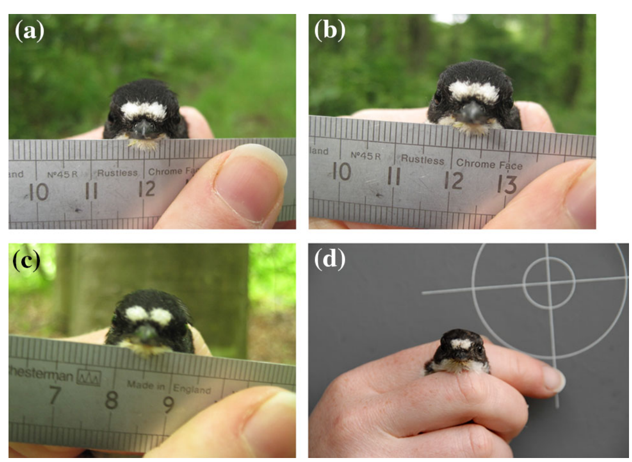


Fig. 1 Blaze segmentation and measurement problem: pictures are taken at different distances, thus requiring a measurement tool

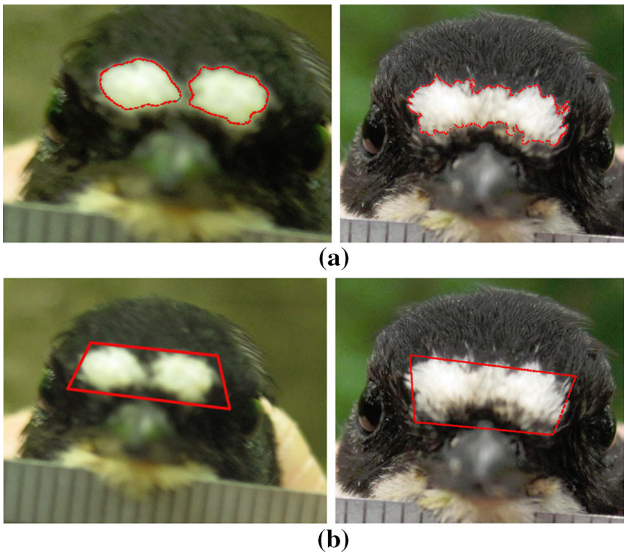


Fig. 2 Flycatcher blaze segmentation of Fig. 1b, c obtained either by using the 'magic-wand' tool of the ImageJ software, similarly as described by Moreno [50], or by trapezium fitting as suggested by Potti and Montalvo [54]. In the first case, the result is strongly user dependent; in the second one, the blaze area is overestimated. **a** Magic wand. **b** Trapezium fitting

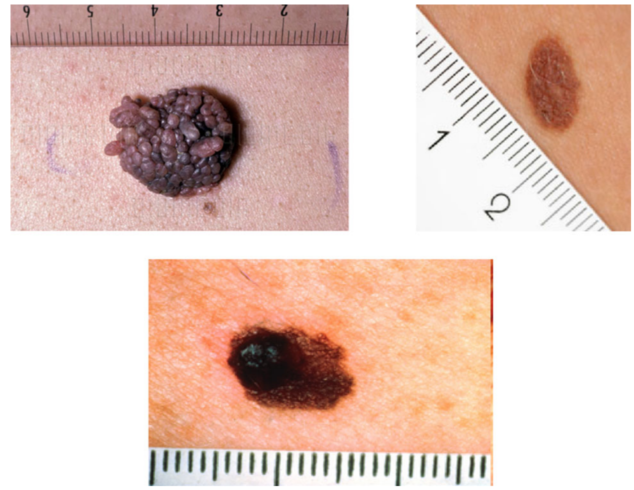


Fig. 3 The monitoring and measuring of moles is essential for the early diagnosis of melanoma. Normally, due to their small size, they can be measured by juxtaposing a small ruler with them

of the white patch. Morales et al. [49] measured the length and breadth of the forehead patch with callipers to the nearest 0.01 mm, and its size (mm^2) was calculated as the area of a rectangle. Other studies have measured the patches from photographs, e.g. Järvistö et al. [37] Ruuskanen et al. [57] and Sirkiä et al. [60] who photographed the forehead with a scale bar included in each picture, and measured the patch as the white area in mm^2 using ImageJ software [2]. But none of these three papers provide methods of how the measurement was actually achieved, e.g. whether patches were delineated or roughly estimated with a simple shape. Most recently, Moreno et al. [50] analysed digital photographs of forehead patches with Adobe PhotoShop CS version 11.0. relating the distance of 1 mm on the ruler to number of pixels and used this to estimate length. Zooming to 400 % and using the paintbrush tool with 100 % hardness and 25 % spacing, the authors delineate the patch and measure the area of the white areas on forehead. While this is the best measurement method to date, it still is subject to human measurement error and subjective assessment of patch boundaries. We report some segmentation results obtained by manual selection and polygon fitting in Fig. 2. In this manuscript, we use a mathematically robust approach to segment the blaze independently to provide an accurate measurement of forehead patch area.

A similar challenge can be encountered in medical applications monitoring and quantifying the evolution of skin moles for early diagnosis of melanoma (skin cancer). A normally user-dependent measurement of the mole is performed using a ruler located next to it. A picture is then taken and used for future comparisons and follow-up; see Fig. 3 and compare [1,17] for previous attempts of automatic detection of melanomas. For such an application, a systematic quantitative analysis is also required.[1]

In several other applications, the task of measuring objects directly from the image is encountered. These include zoological and behavioural studies arising in the animal world where detecting size, shape and possible symmetries of specific distinctive animal features can be useful, as well as, for instance, in archaeological digs where the measurement of finds is important for comparisons and classification [34].

Outline of the method We consider the image as a graph whose vertices are the image pixels. Similarity between pixels in terms of colour or texture features is modelled by a weight function defined on the set of vertices. Our method runs as follows. Firstly, using examples provided by the user (dictionaries) as well as matrix completion and operator splitting techniques, the segmentation of the region of interest is performed. In the graph framework, this corresponds to clustering together pixels having similar features. This is obtained by minimising on the graph the Ginzburg–Landau functional typically used in the continuum setting to describe diffuse interface problems. In order to provide quantitative measurements of the segmented region, a second detection step is then performed. The detection here aims to identify the distinctive geometrical features of the measurement tool (such as line

[1] Mole images from http://www.medicalprotection.org/uk/practice-matters-issue-3/skin-lesion-photography, ©Chassenet/Science Photo Library, http://en.wikipedia.org/wiki/Melanoma (public domain), http://www.diomedia.com/stock-photo-close-up-of-a-papillomatous-dermal-nevus-mole-a-raised-pigmented-skin-lesion-that-results-from-a-proliferation-of-benign-melanocytes-c-cid-image14515019.html, ©Phototake RM/ ISM.

alignment for rulers or circularity for circles) to get the scale on the measurement tool considered. The segmented region of interest can now be measured by simple comparisons, and quantitative measurements such as perimeter and area can be provided.

Contribution We propose a self-contained programme combining automated detection and subsequent size measurement of objects in images where a measurement tool is present. Our approach is based on two powerful image analysis techniques in the literature: a graph segmentation approach which uses a discretised Ginzburg–Landau energy [9] for the detection of the object of interest and the Hough transform [35] for detecting the scale of the measurement tool. While these methods are state of the art, their combination for measuring object size in images proposed in this paper is new. Moreover, to our knowledge there is only little contribution in the literature that broach the issue of how the graph segmentation approach and the Hough transform are applied to specific problems [28,32,43]. Indeed, here we present these methodologies in detail, especially discussing important aspects in their practical implementation, and demonstrate the robust applicability of our programme for measuring the size of objects, showcasing its performance on several examples arising in zoology, medicine and archaeology. Namely, we first apply our combined model for the measurement of the blaze on the forehead of male pied flycatchers, for which we run a statistical analysis on the accuracy and predicted error in the measurement on a database of thirty images. State-of-the-art methods for such a task typically require the user to fit polygons inside or outside the blaze [54] or to segment the blaze by hand [50]. Similarly, the scale on the measurement tool is typically read from the image by manually measuring it on the ruler. With respect to medical applications, we apply our combined method for the segmentation and measurement of melanomas. Although efficient segmentation methods for automatised melanoma detection already exist in the literature (see, for example, [1,17]), to the knowledge of the authors, no previous methods providing their measurement by detecting the scale on the ruler placed next to them (see Fig. 3) exist. Conversely, in the case of archaeological applications, some models for the automatic detection of the measurement tool in the image exist [34], but no automatic methods are proposed for the segmentation of the region of interests. A free release of the MATLAB code used to compute the results will be made available after the zoological analysis of the pied flycatcher's data based on our segmentation, and measurement has been completed [14].

Organisation of the paper In Sect. 2, we present the mathematical ingredients used for the design of the graph-based segmentation technique used in [9,28,42,43]. They come from two different worlds: the framework of diffusion PDEs used for modelling phase transition/separation problems (see Sect. 2.1) and graph theory and clustering, see Sect. 2.2. In view of a detailed numerical explanation, we also recall a splitting technique and a popular matrix completion technique used in our problem to overcome the computational costs. In Sect. 3, we explain how the geometrical Hough transform is used to detect the scale in an image. Finally, Sect. 4 contains the numerical results obtained with our combined method applied to the problems described above. For completeness, we give some details on the Nyström matrix completion technique in 'Appendix 1' and a review of the Hough transform for line and circle detection in 'Appendix 1'.

2 Image Segmentation as Graph Clustering

We present in this section the mathematical background for the design of the Ginzburg–Landau (GL) graph segmentation algorithm introduced in [9]. There, the image segmentation problem is rephrased as a minimisation problem on a graph defined by features computed from the image. Compared to the methods above, the graph framework allows for more freedom in terms of the possible features used to describe the image, such as texture.

2.1 The Ginzburg–Landau Functional as Approximation of TV

In the following, we recall the main properties of the original continuum version of the GL functional explaining its importance in the context of image segmentation problems as well as the main concepts of graph theory which will be used for the segmentation modelling.

Several physical problems modelling phase transition and phase separation phenomena are built around the well-known GL functional:

$$GL(u) := \frac{\varepsilon}{2}\int_{\Omega} |\nabla u(x)|^2 \, \mathrm{d}x + \frac{1}{\varepsilon}\int_{\Omega} W(u(x)) \, \mathrm{d}x. \qquad (2.1)$$

The functional above is defined in the continuous setting. Here, Ω represents a open subset of $\mathbb{R}^d$, $d = 2, 3$, $u : \Omega \to \mathbb{R}$ is the density of a two-phase material, and $W(u)$ is a double-well potential, e.g. $W(u) = \frac{1}{4}(u^2 - 1)^2$. The two wells ± 1 of W correspond to the two phases of the material. The parameter $\varepsilon > 0$ is the spatial scale. Variational models built around this functional are also referred to as *diffuse interface* models because of the interface appearing between the two regions containing the phases (i.e. the two wells of W) due to the competition between the two terms of the functional (2.1). Nonetheless, some smoothness preventing u from having jumps between the two regions is ensured by the first regularisation term.

The use of the GL functional has become very popular in image processing due to its connections with the total variation (TV) semi-norm. In [47,48], for instance, Γ-convergence properties of (2.1) to the TV functional are shown. Thus, the GL functional is very often used as a quadratic approximation of total variation. Fast numerical schemes relying on these connections have been designed for many imaging problems, thus overcoming the issues related to non-smooth TV minimisation [5,18,26]. In image processing, the functional considered often is of the form

$$E(u) := \mathrm{GL}(u) + \lambda\, \phi(u, u_0), \tag{2.2}$$

where $\phi(u, u_0)$ is a fidelity term measuring the distance of the reconstructed image u to the given image u_0. Depending on the application, different data fidelities are employed. Typically, they are related to statistical and physical assumptions of the model considered. Standard examples of fidelity terms are $\phi(u, u_0) = \|u - u_0\|^d_{L^d(\Omega)}$, $d = 1, 2$. The parameter $\lambda > 0$ determines the influence of the data fit compared to the regularisation. Taking the L^2 gradient descent of (2.2), we get the following evolutionary PDE, known in the literature as the Allen–Cahn equation [3] with an additional forcing term due to the fidelity ϕ:

$$u_t = -\frac{\delta \mathrm{GL}}{\delta u} - \lambda \frac{\delta \phi}{\delta u} = \varepsilon \Delta u - \frac{1}{\varepsilon} W'(u) - \lambda \frac{\delta \phi}{\delta u}. \tag{2.3}$$

Steady states of Eq. (2.3) are the stationary points of the energy E in (2.2). Note that E is not convex so uniqueness is not guaranteed and, consequently, the long time behaviour for solutions of (2.3) will depend on the initial condition. The linear diffusion term weighted by ε appearing in (2.3) allows for fast solvers using, for instance, the fast Fourier transform (FFT) which translates the Laplace operator into a multiplication operator on the Fourier modes.

2.2 Towards the Modelling: The Graph Framework

In the following, we rely on the method presented in [9,42] for high-dimensional data classification on graphs, which has been applied to several imaging problems [28,43], showing good performance and robustness. We consider the problem of binary image segmentation where we want to partition a given image into two components where each component is a set of pixels (also called a cluster, or a class) and represents a certain object or group of objects. Typically, some a priori information describing the object(s) we want to extract is given and serves as initial input for the segmentation algorithm. For image labelling, in [9] two images are taken as input: the first one has been manually segmented in two classes and the objective is to automatically segment the second image using the information provided by the segmentation of the first one.

We revise in the following the main ingredients of the model considered and start from a quick review of concepts in graph theory. We represent a rectangular image with $S := N \times M$ pixels by the set $I := \{x = (x_1, x_2) \in \mathbb{Z}^2 : 0 \le x_1 \le N-1 \text{ and } 0 \le x_2 \le M-1\}$. For each $x \in I$, we define the image neighbourhood of x as the set

$$\mathcal{N}(x) := \{y \in I : |x_1 - y_1| \le \tau \text{ and } |x_2 - y_2| \le \tau\},$$

with $\tau \in \mathbb{N}$ fixed, i.e. $\mathcal{N}(x)$ contains the pixels in a $(2\tau+1) \times (2\tau+1)$ sized square centred at x. For some appropriate $K \in \mathbb{N}$, we associate with every pixel $x \in I$ a vector $z \in \mathbb{R}^K$ encoding selected characteristics of the neighbourhood $\mathcal{N}(x)$. These characteristics are related to the grey or RGB (red, green, blue) intensity values as well as the texture features of the neighbourhood. In Sect. 2.5, we will explain in more detail our feature vector construction. The map $\psi : I \to \mathbb{R}^K, x \mapsto z$ is called the feature function. For constructing the feature vectors in Sect. 2.5, it will be useful to associate a neighbourhood vector $\nu(x) := (x_j)_{j \in \mathcal{N}(x)} \in I^{(2\tau+1)\times(2\tau+1)}$ with each neighbourhood, such that the ordering of the x_j in $\nu(x)$ is consistent between pixels x, e.g. order the pixels from each square $\mathcal{N}(x)$ from left to right and top to bottom. The specific choice of ordering is not important, as long as it is consistent for each pixel neighbourhood.

Next we construct a simple weighted undirected graph $G = (V, E, w)$ whose vertices correspond to the pixels in I and with edges whose weights depend on the feature function ψ. Let V be a vertex set of cardinality S. To emphasise that each vertex in V corresponds to exactly one pixel in I, we will label the vertex corresponding to $x \in I$ by x as well. Let $w : V \times V \to \mathbb{R}$ be a symmetric and non-negative function, i.e. for each $x_i, x_j \in V$

$$w(x_i, x_j) = w(x_j, x_i), \qquad w(x_i, x_j) \ge 0. \tag{2.4}$$

We define the edge set E as the collection of all undirected edges connecting nodes x_i and x_j for which $w(x_i, w_j) > 0$ [20]. The function w restricted to $E \subset V \times V$ is then a positive edge weight function.

In our applications, we define w as

$$w(x_i, x_j) := \hat{w}(\psi(x_i), \psi(x_j)) = \hat{w}(z_i, z_j),$$

where $\hat{w} : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$ is a given function and ψ is the feature function.

In operator form, the *weight matrix* $W \in \mathbb{R}^{S \times S}$ is the non-negative symmetric matrix whose elements are $w_{i,j} = w(x_i, x_j)$. In the following, we will not distinguish between the two functions w and $\hat{w}$ and, with a little abuse of notation, we will write $w(z_i, z_j)$ for $\hat{w}(z_i, z_j)$.

Remark 1 Weight functions express the similarities between vertices and will be used in the following to partition V into clusters such that the sum of the edge weights between the clusters is small. There are many different mathematical approaches to attempt this partitioning. When formulated as a balanced cut minimisation, the problem is NP-complete [67], which inspired relaxations which are more amenable to computational approaches, many of which are closely related to spectral graph theory [59]. We refer the reader to [20] for a monograph on the topic. The method we use in this paper can be understood (at least in spirit, if not technically, [63,64]) as a nonlinear extension of the linear relaxed problems.

To solve the segmentation problem, we minimise a *discrete GL functional* (which is formulated in the graph setting, instead of the continuum setting), via a gradient descent method similar to the one described in Sect. 2.1. In particular, in this setting the Laplacian in (2.3) will be a (negative) normalised graph Laplacian. We will use the spectral decomposition of u with respect to the eigenfunctions of this Laplacian. In Sect. 2.4, we discuss the Nyström method, which allows us to quickly compute this decomposition, but first we introduce the graph Laplacian and graph GL functional.

The discrete operators We start from the definition of the differential operators in the graph framework.

For each vertex $x \in V$, we define the *degree* of x,

$$d : V \to \mathbb{R}, \quad d(x) := \sum_{y \in V} w(x, y).$$

In operator form, the diagonal *degree matrix* $D \in \mathbb{R}^{S \times S}$ is defined to have diagonal elements $d_{i,i} = d(x_i)$.

A subset A of the vertex set V is *connected* if any two vertices in A can be connected by a path (i.e. a sequence of vertices such that subsequent vertices are connected by an edge in E) such that all the vertices of the path are in A. A finite family of sets $A_1, \ldots, A_t$ is called a *partition* of the graph if $A_i \cap A_j = \emptyset$ for $i \neq j$ and $\bigcup_i A_i = V$.

We now have all the ingredients to define the *graph Laplacian*. Denoting by $\mathcal{V}$ the space of all the functions $V \to \mathbb{R}$, the graph Laplacian is the operator $L : \mathcal{V} \to \mathcal{V}$ such that:

$$Lu(x) = \sum_{y \in V} w(x, y)(u(x) - u(y)), \quad x \in V. \tag{2.5}$$

We are considering a finite graph of size S, so real-valued functions can be identified as vectors in $\mathbb{R}^S$. We can then write the graph Laplacian in matrix form as $L = D - W$ or elementwise as:

$$L(x, y) := \begin{cases} d(x), & \text{if } x = y, \\ -w(x, y), & \text{otherwise.} \end{cases} \tag{2.6}$$

It is worth mentioning (see Remark 2 below) that this graph Laplacian is a positive semi-definite operator. Note that by convention the sign of the discrete Laplacian is opposite to that of the (negative semi-definite) continuum Laplacian. The associated quadratic form of L is

$$Q(u, Lu) := \frac{1}{2} \sum_{x, y \in V} w(x, y)\,(u(x) - u(y))^2. \tag{2.7}$$

The quadratic form Q can be interpreted as the energy whose optimality condition corresponds to the vanishing of the graph Laplacian in (2.6).

Remark 2 The operator L has S non-negative real-valued eigenvalues $\{\lambda_i\}_{i=1}^S$ which satisfy: $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_S$. The eigenvector corresponding to λ_1 is the constant S-dimensional vector $\mathbf{1}_S$, see [67].

The operator in (2.5)–(2.6) is not the only graph Laplacian appearing in the literature. To set it apart from others, it is also referred to as the unnormalised or combinatorial graph Laplacian. Such operator can be related to the standard continuous differential one through non-local calculus [30]. More precisely, the eigenvectors of L converge to the eigenvectors of the standard Laplacian, but in the large sample size limit, a proper scaling of L is needed in order to guarantee stability of convergence to the continuum operator [9,43]. Hence, we consider in the following the normalisation of L given by the symmetric graph Laplacian

$$L_s := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}. \tag{2.8}$$

Clearly, the matrix L_s is symmetric. Other normalisations of L are possible, such as the random walk graph Laplacian (see [20,64,67]).

In [59, Section 5], a quick review on the connections between the use of the symmetric graph Laplacian (2.8) and spectral graph theory is given. Computing the eigenvalues of the normalised symmetric Laplacian corresponds to the computation of the generalised eigenvalues used to compute normalised graph cuts in a way that the standard graph Laplacian may fail to do, compare [20]. Typically, spectral clustering algorithms for binary segmentation base the partition of a connected graph on the eigenvector corresponding to the second eigenvalue of the normalised Laplacian, using, for example, k-means. For further details and a comparison with other methods, we refer the reader to [59] and to [9, Section 2.3] where a detailed explanation on the importance of the normalisation of the Laplacian is given.

The discrete GL functional Recalling (2.1)–(2.2) and (2.7), we define the discrete GL functional[2] as

$$\mathrm{GL}_d(u) := \frac{\varepsilon}{2}\, Q(u, L_s u) + \frac{1}{\varepsilon}\sum_{x\in V} W(u(x)) + \sum_{x\in V}\frac{\chi(x)}{2}(u(x)-u_0(x))^2. \tag{2.9}$$

Here u_0 represents the known training data provided by the user. As before, $W(u(x)) = \frac{1}{4}(u^2(x)-1)^2$ is the double-well potential. The function $\chi : V \to \{0, 1\}$ is the characteristic function of the subset of labelled vertices $V_{lab} \subset V$, i.e. $\chi = 1$ on V_{lab} and $\chi = 0$ on $V_{unlab} := V_{lab}^c$. Hence, the corresponding fidelity term enforces the fitting between u and u_0 in correspondence to the known labels on the set V_{lab}, while the labelling for the pixels in V_{unlab} is driven by the first two regularising terms in (2.9).

The corresponding ℓ^2 gradient flow for (2.9) reads

$$u_t = -\,\varepsilon\, L_s u - \frac{1}{\varepsilon}\sum_{x\in V}(u^3(x)-u(x)) - \sum_{x\in V}\chi(x)(u(x)-u_0(x)).$$

The idea is to design a semi-supervised learning (SSL) approach where a priori information for the set V_{lab} (i.e. cluster labels) is used to label the points in the set V_{unlab}. The comparison uses the weight function defined in (2.4) to build the graph by comparing the feature vectors at each point.

Remark 3 (*The weight function*) As pointed out in [9, Section 2.5], the main criteria driving the choice of the weight function are the desired outcome and the computational efforts required to diagonalise the corresponding matrix W. A common weight function is the Gaussian function, which, for $x, y \in V$, reads

$$w(x, y) = \exp(-\|\psi(x)-\psi(y)\|^2/\sigma^2), \quad \sigma > 0. \tag{2.10}$$

Note that this function is symmetric: $w(x, y) = w(y, x)$.

Several approaches to SSL using graph theory have been considered in the literature, compare [21,30]. The approach presented here adapts fast algorithms available for the efficient minimisation of the continuous GL functional to the minimisation of the discrete one in (2.9). In particular, to overcome the high computational costs, we present in the following an operator splitting scheme and a matrix completion technique applied to our problem.

[2] 'Discrete GL functional with a data fidelity term' would be a more accurate name, but we opt for brevity here.

2.3 Convex Splitting

Splitting methods are used in the study of PDEs. Here, we focus on convex splitting, which is used to numerically solve problems with a general gradient flow structure. Decomposing GL_d as

$$\mathrm{GL}_d = \mathrm{GL}_{1,d} - \mathrm{GL}_{2,d}$$

where both $\mathrm{GL}_{1,d}$ and $\mathrm{GL}_{2,d}$ are convex and denoting by U_n the spatial discretisation of $u(\cdot, n\Delta t)$, $\Delta t > 0$, $n \geq 0$, a semi-implicit discretisation for the steepest descent of GL_d reads

$$U_{n+1} - U_n = -\Delta t(\nabla_V \mathrm{GL}_{d,1}(U_{n+1}) - \nabla_V \mathrm{GL}_{d,2}(U_n)), \tag{2.11}$$

where ∇_V indicates formally the Fréchet derivative with respect to the metric in a Banach space V. The advantage of the convex splitting consists in treating the convex part implicitly in time and the concave part explicitly. Typically, nonlinearities are considered in the explicit part of the splitting and their instability is balanced by the effect of the implicit terms.

The terms $\mathrm{GL}_{d,1}$ and $\mathrm{GL}_{d,2}$ in (2.11) read in our case (cf. [9, Section 3.1])

$$\mathrm{GL}_{d,1}(u) := \frac{\varepsilon}{2}\, Q(u, L_s u) + \frac{C}{2}\sum_{x\in V} u^2(x), \tag{2.12a}$$

$$\mathrm{GL}_{d,2}(u) := -\frac{1}{4\varepsilon}\sum_{x\in V}(u^2(x)-1)^2 + \frac{C}{2}\sum_{x\in V} u^2(x) - \sum_{x\in V}\frac{\chi(x)}{2}(u(x)-u_0(x))^2, \tag{2.12b}$$

where the constant $C > 0$ has to be chosen large enough such that $\mathrm{GL}_{d,2}$ is convex for u around the wells of W. The differential operator contained in the implicit component of the splitting, $\mathrm{GL}_{d,1}$, is the symmetric graph Laplacian, which can be diagonalised quickly and inverted using Fourier transform methods. In [9, Section 3.1], more details of the splitting are presented. Writing out in detail the time-discretised scheme (2.11), we get, for every $n \geq 1$

$$\begin{aligned} U_{n+1}(x) - U_n(x) &= -\Delta t\,(\varepsilon\, L_s(U_{n+1}(x)) + C U_{n+1}(x)) \\ &\quad - \Delta t\left(-\frac{1}{\varepsilon}\left(U_n^3(x) - U_n(x)\right) + C\, U_n(x) \right. \\ &\quad \left. - \chi(x)\,(U_n(x) - U_0)\right), \quad x \in V. \end{aligned} \tag{2.13}$$

Here, U_0 denotes the training data, i.e. the known labels -1 and 1 assigned by the user to nodes in the subset $V_{lab} \subset V$. In our numerical experiments, we initialised the time-stepping (2.13) by taking

$$U_1(x) = \begin{cases} U_0(x), & \text{if } x \in V_{lab}, \\ 0, & \text{if } x \in V_{lab}^C. \end{cases} \tag{2.14}$$

Towards the numerical realisation The numerical strategy we intend to use is based on the following steps (see Sect. 2.5 for more details):

- At each time step $n\Delta t$, $n \geq 1$, consider at every point the spectral decomposition of U_n with respect to the eigenvectors v_k of the operator L_s as

$$U_n(x) = \sum_k \alpha_n^k(x) v_k(x), \quad x \in V \tag{2.15}$$

  with coefficients α_n. Similarly, use spectral decomposition in the $\{v_k\}$ basis for the other nonlinear quantities appearing in (2.13).
- Having fixed the basis of eigenfunctions, the numerical approximation in the next time step U_{n+1} is computed by determining the new coefficients α_{n+1}^k in (2.15) for every k through convex splitting (2.13).

The only possible bottleneck of this strategy is the computation of the eigenvectors v_k of the operator L_s, which, in practice, can be computationally costly for large and non-sparse matrices W. To mitigate this potential problem, we use the Nyström extension (Sect. 2.4).

2.4 Matrix Completion via Nyström Extension

Following the detailed discussion in [9, Section 3.2], we present here the Nyström technique for matrix completion [53] used in previous works by the graph theory community [7,27] and applied later to several imaging problems [44,45,52]. In our problem, the Nyström extension is used to find an approximation of the eigenvectors v_k of the operator L_s. We will freely switch between the representation of eigenvectors (or eigenfunctions) as a real-valued functions on the vertex set V and as a vectors in $\mathbb{R}^S$.

Consider a fully connected graph with vertices V and the set of corresponding feature vectors $\psi(V) = \{z_i\}_{i=1}^S$. A vector v is an eigenvector of the operator L_s in (2.8) with eigenvalue λ if and only if v is an eigenvector of the operator $D^{-1/2}WD^{-1/2}$ with eigenvalue $1-\lambda$, since

$$L_s v = v - D^{-1/2}WD^{-1/2}v = \lambda v \quad \Longleftrightarrow$$
$$D^{-1/2}WD^{-1/2}v = (1-\lambda)v. \tag{2.16}$$

Thus, finding the spectral decomposition of L_s boils down to diagonalising the operator $D^{-1/2}WD^{-1/2}$. This is not obviously easier, as the matrix W, despite being non-negative and symmetric, may be large and non-sparse, so the computation of its spectrum may be computationally hard. Here, however, we take advantage of the Nyström extension. Given the eigenvalue problem

$$\text{find } \theta \in \mathbb{R} \text{ and } v : V \to \mathbb{R}, v \neq 0 \quad \text{s. t.}$$
$$\sum_{x \in V} w(x, y)\, v(x) = \theta v(y), \tag{2.17}$$

for every point $y \in V$, we approximate the sum on the left-hand side using a standard quadrature rule where the interpolation points are chosen by randomly selecting a subset of L points from the set V and the interpolation weights are chosen correspondingly. The Nyström extension for (2.17) then approximates (2.17) by

$$\sum_{i=1}^L w(y, x_i)v(x_i) \approx \sum_{x \in V} w(y, x)v(x) = \theta v(y), \tag{2.18}$$

where $X := \{x_i\}_{i=1}^L \subset V$ is a set of randomly chosen vertices. The set X defines a partition of V into X and $Y := X^c$. In (2.18), we approximate the value $v(y)$, for an eigenvector v of W and $y \in Y$, only knowing the values $v(x_i)$, $i = 1, \ldots, L$, by solving the linear problem

$$\sum_{i=1}^L w(y, x_i)v(x_i) = \theta v(y). \tag{2.19}$$

With this method, we can approximate the values of an eigenvector v of W, corresponding to the eigenvalue θ, in the *whole* set of points V using its values in the subset X and solving the interpolated eigenvalue equation above. Generally, this is not as immediate as it sounds since the eigenvectors of W are not known in advance; however, by choosing $y = x_j$, $j = 1, \ldots, L$, in (2.19), we find an eigenvalue problem for the known matrix with entries $w(x_j, x_i)$, which is a much smaller matrix than the full matrix W:

$$\sum_{i=1}^L w(x_j, x_i)v(x_i) = \theta v(x_j). \tag{2.20}$$

If L is small enough such that this eigenvalue problem can be solved, then θ and $v(x_i)$, $i = 1, \ldots, L$, can be computed, which in turn can be substituted back into (2.19) to find an approximation to $v(y)$, for any $y \in V$. In short, we approximate the eigenvectors in (2.17) by extensions of the eigenvectors in (2.20), using the extension equation (2.19), and we approximate the eigenvalues in (2.17) by the eigenvalues from (2.20). The main Nyström assumption is that these approximated eigenvectors and eigenvalues approximately diagonalise W. For further details on the Nyström method, we refer the reader to 'Appendix 1' where a description of the method is given in matrix notation.

2.5 Pseudocode

We present here the pseudocode combining all the different steps described above for the realisation of the GL minimisation. We recall that ε is the scale parameter of the GL functional (2.9), σ is the variance used in the Gaussian similarity function (2.10), C is the convex splitting parameter in (2.12a)–(2.12b), and L is the number of sample points in (2.18).

Algorithm 1 GL-minimisation with Nyström extension for image segmentation

PARAMETERS: $L \ll S, \sigma, \varepsilon, C$.
select L random points and build the set $X \subset V$
get a partition $V = X \cup Y,\ Y := X^c$
determine features and edge weights of X and Y using (2.10) and build W_{XX} and W_{XY}
Nyström extension to compute normalised matrix of eigenvectors of W and get eigenvalues-eigenvectors of W $(\hat{\lambda}_i, v_i)$
output ← eigenvalues-eigenvectors $(1-\hat{\lambda}_i, v_i)$ of L_s used as GL minimisation input
convex splitting for GL minimisation through Fourier transform methods, as described in Sect. 2.3
output ← the binary segmentation.

We will now give further details. First, we randomly select L pixels from I. As described in Sect. 2.2, we now create a vertex set $V \cong I$, which we partition into a set X, consisting of the vertices corresponding to the L randomly chosen pixels, and a set $Y := V \backslash X$. We now compute the feature vectors of each vertex in V. If I is a greyscale image, we can represent features by an intensity map $f : V \to \mathbb{R}$. If I is an RGB colour image instead, we use a vector-valued (red, green, and blue) intensity map $f : V \to \mathbb{R}^3$ of the form $f(x) = (f_R(x), f_G(x), f_B(x))$. We mirror the boundary to define neighbourhoods also on the image edges. The feature function $\psi : V \to \mathbb{R}^K$ concatenates the intensity values in the neighbourhood $\nu(x)$ of a pixel into a vector: $\psi(x) := (f(\nu_1(x)), \ldots, f(\nu_{\tilde{\tau}}(x)))^{\mathrm{T}}$, where $\nu(x) = (\nu_1(x), \ldots, \nu_{\tilde{\tau}}(x)) \in \mathbb{R}^{\tilde{\tau}}$ is the neighbourhood vector of $x \in V$ defined in Sect. 2.2 and $\tilde{\tau} = (2\tau+1)^2$, the size of the neighbourhood of x. Note that $K = \tilde{\tau}$ if I is a greyscale image and $K = 3\tilde{\tau}$ if I is an RGB colour image.

Additional features can be considered, such as texture, for instance. For instance, we consider the eight MR8 filter responses [65] as texture features on a greyscale image and choose the function $t : V \to \mathbb{R}^8$ as $t(x) = (\mathrm{MR8}_1(x), \ldots, \mathrm{MR8}_8(x))$. Hence, the feature function ψ is now defined as $\psi(x) := (t(\nu_1(x)), \ldots, t(\nu_{\tilde{\tau}}(x)))^{\mathrm{T}}$, where $\nu(x)$ and $\tilde{\tau}$ are defined as above. Here, $K = 8\tilde{\tau}$. Of course, a combination of colour and texture features can be considered as well by considering ψ defined as $\psi(x) := (f(\nu_1(x)), t(\nu_1(x)), \ldots, f(\nu_{\tilde{\tau}}(x)), t(\nu_{\tilde{\tau}}(x)))$ for every x in V. In this case, when dealing with RGB colour images, the dimension of the feature vector is therefore $K = 11\tilde{\tau}$.

Using (2.10), the Nyström extension can be performed for approximating the eigenvectors and eigenvalues of W as described in Sect. 2.4 and in 'Appendix 1', which are then used to compute the eigenvectors $\{v_k\}$ of L_s and corresponding eigenvalues $\{\lambda_k\}$, compare (2.16). Recalling (2.15), those eigenvectors are used as basis functions for U_n, the numerical approximation of u in the nth iteration of the GL minimisation. Considering (2.13) and writing the nonlinear quantities appearing in terms of $\{v_k\}$ similarly as in (2.15), we have for $x \in V$

$$(U_n(x))^3 = \sum_k \beta_n^k(x)$$

$$v_k(x), \quad \chi(x)\left(U^n(x) - u_0(x)\right) = \sum_k \gamma_n^k(x)\, v_k(x).$$

The computation of U in the next iteration reduces to finding the coefficients α_{n+1}^k in the expression

$$U_{n+1}(x) = \sum_k \alpha_{n+1}^k(x) v_k(x), \quad x \in V,$$

in terms of β_n^k, γ_n^k and the other parameters involved, that is the scale parameter ε in (2.9), the parameter $C > 0$ appearing in the splitting (2.12) and the time step Δt. Using (2.13), we compute α_{n+1}^k simply as

$$\alpha_{n+1}^k = \mathcal{D}_k^{-1}\left(\left(1 + \frac{\Delta t}{\varepsilon} + C\Delta t\right)\alpha_n^k - \frac{\Delta t}{\varepsilon}\beta_n^k - \Delta t\, \gamma_n^k\right),$$

where $\mathcal{D}_k$ is defined as $\mathcal{D}_k := 1 + \Delta t(\varepsilon\lambda_k + C)$.

3 Hough Transform for Scale Detection

In order to detect objects in an image with specific, a priori specified shapes, in the following we will make use of the Hough transform. For our purposes, we will focus in particular on straight lines detection (for which the Hough transform was originally introduced and considered [35]) and circles [23]. Other applications of this transformation for more general curves exist as well. In [8,41], the Hough transform is used in the context of astronomical and medical images for a specific class of curves (*Lamet* and *Watt* curves). In [32], applications to cellular mitosis are presented. There, the Hough transform recognises the cells (as circular/elliptical objects) and tracks them in the process of cellular division. For more details on the use of the Hough transform for line and circle detection, we refer the interested reader to 'Appendix 3'.

Numerical strategy Hough transform methods for edge detection are usually applied to binary images. Therefore, we start by using the classical Canny method for edge detection [15] in which we replace the original preliminary Gaussian filtering by an edge-preserving total variation smoothing [55] which has the advantage of removing noise while preserving edges. This step will result in a binary image for the most prominent edges in the image. Having decided which geometrical shape we are interested in (and, as such, its general parametric representation), the corresponding parameter space is subdivided into accumulator arrays (cells) whose dimension depends on the dimension of the parameter space itself (2D in the case of straight lines, 3D in the case of circles). Each accumulator array groups a range of parameter values. The accumulator array is initialised to 0 and incremented every time an object in the parameter space passes through the cell. In this way, one looks for the peaks over the set of accumulator arrays as they indicate a high value of intersecting objects for a specific cell. In other words, they are indicators of potential objects having the specific geometrical shape we are interested in.

3.1 Pseudocode

Numerically, dealing with the Hough transform consists of looking for peaks of the accumulator arrays in the parameter space onto which the original image is mapped. We use the MATLAB routines `hough`, `houghpeaks` and `houghlines` for straight lines detection and `imfindcircles` for circle detection. The accuracy and the number of detections for such routines can be tuned by some parameters, such as, for instance, the maximum number of peaks one wants to consider, obj_{max}, or the array peak threshold, $thresh$, i.e. the minimum number of elements for an accumulator array to be considered a peak. The user also has to specify an initial range of pixel values $[s_{min}, s_{max}]$ as a very rough approximation of the measurement scale. Namely, in the case of line detection this determines a minimum/maximum *spacing* between lines, whereas for circle detection this serves as a rough approximation of the range of values for the circles' radii. This rough approximation may be given, for example, from average data which the user knows a priori. We explain this with some examples in Sect. 4. Accuracy of the detection algorithm is tuned by a parameter acc. In case of linear objects, this corresponds to choosing the maximum number of pixels between two line segments to consider them as one single line, whereas for circle detection this corresponds to the circularity of an object to be considered a circle.

Algorithm 2 Hough transform for lines and circles detection

1: PARAMETERS: $[s_{min}, s_{max}]$, obj_{max}, acc, $thresh$
2: preprocessing: TV-Canny edge detection
3: compute the Hough transform of the edge image
4: set up detection accuracy, depending on acc, and use $[s_{min}, s_{max}]$ as rough initial guess
5: determine at most obj_{max} peaks in the parameter space, thresholding using $thresh$
6: output ← peaks in the parameter space, corresponding to objects of interest in the original image

4 Method, Numerical Results and Applications

We report in this section the numerical results obtained by the combination of the methods presented for the detection and quantitative measurement of objects in an image.

To avoid confusion, we will distinguish in the following between two different meanings of scale. Namely, by image scale we denote the proportion between the real dimensions (length, width) of objects in the image and their corresponding dimensions quantified in pixel count. Dealing with measurement tools, we talk about measurement scale to intend the ratio between a fixed unit of measure (mm or cm) marked the measurement tool considered and the correspondent number of pixels on the image.

4.1 Male Pied Flycatcher'S Blaze Segmentation

Here we present the numerical results obtained by applying Algorithms 1 and 2 to the male pied flycatcher blaze segmentation and measurement problem described in the introduction. Our image database consists of 32 images of individuals from a particular population of flycatchers. Images are 3648 × 2736 pixels and have been taken by a Canon 350D camera with Canon zoom lens EFD 18–55 mm, see Fig. 1. In each image, one of two types of measurement tool is present: a standard ruler or a surface on which two concentric circles of fixed diameter (1 cm the inner one, 3 cm the outer one) are drawn. In the following, we will refer to these tools as *linear* and *circular* ruler, respectively. Here, the measurement scale corresponds to the distance between ruler marks for linear rulers and to the radius of the inner circles for circular rulers.

Figure 1 shows clearly that the scale of the images in the database may vary significantly because of the different positioning of the camera in front of the flycatcher. In order to study possible correlations between the dimensions (i.e. perimeter, area) of the blaze and significant behavioural factors, the task then is to segment the blaze and detect automatically the scale of the image considered to provide scale-independent measurements.

Parameter choice for Algorithm 1 The GL segmentation method exploits similarities and differences between pixels in terms of RGB intensities and texture within their neighbourhood. In our image database, these similarities and differences are very distinctive and will guide the segmentation step. Recalling Sect. 2.5, we note that some parameters need to be tuned for the graph GL minimisation. Those are the number L of Nyström points, the variance σ of the similarity function (2.8), the GL parameter ε and the parameter C for the convex splitting (2.12). However, in our numerical experiments we had to tune these parameters only once. Namely, regarding the choice of L for both the head and blaze segmentation, we used values not bigger than 5 % of the total size of the image considered. The variance appearing in the similarity function (2.10) was set to $\sigma^2 = 20$, and the weighting parameter ε was chosen as $\varepsilon = 0.01$ (a smaller choice would create numerical instabilities), and we set the convexity parameter $C = 25$ or larger in order to guarantee the convexity of the functional appearing in (2.12b).

Parameter choice for Algorithm 2 We briefly comment also on the choice of the parameters for the Hough transform, that is Algorithm 2. Depending on the type of measurement tool considered (linear or circular ruler), different parameter selection methods are considered. In the case of linear rulers: for the longest line detection (i.e. ruler edge identification) the parameters obj_{max} and $thresh$ were set to 1 and to 85 % of the maximum value of the Hough transform matrix, respectively; for the detection of the ruler notches, the same parameters were chosen as $obj_{max} = 500$ and $thresh = 20\,\%$ of the maximum value of the Hough transform matrix were used. As discussed in 3.1, the range $[s_{min}, s_{max}]$ was chosen based on previously collected average data on the head diameter. In particular, after the head detection step, the number of pixels corresponding to the diameter of the head was automatically computed by means of the option `EquivDiam` of the MATLAB routine `regionprops` and compared with the average measurement of 1.51 cm provided by available databases on pied flycatcher. In this way, an initial, rough estimate of the ruler scale is found and used to determine a *spacing parameter* s and the interval $[s_{min}, s_{max}]$ by setting $s_{min} = s/2$ and $s_{max} = 2s$. This range serves as a *suppression* neighbourhood: once a peak in the Hough transform matrix (i.e. a line or a circle) is identified, starting from it the successive peaks found outside this range are set to 0 (i.e. possible lines or circles within this interval are discarded), while only the ones inside the range (typically, the following line/circle we want to detect) are kept. For our problem, this corresponds to identifying as candidates for ruler notches only the lines away from each other at least s_{min} and at most s_{max} from the peak which has been previously identified. Analogously, the same can be done with circular rulers, where we recall the inner/outer radii are 1 and 3 cm, respectively. In this case, $obj_{max} = 2$ since the circular ruler is made of only two concentric circles.

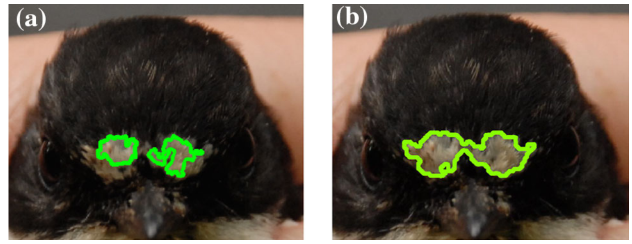


Fig. 4 Blaze segmentation results computed by using Chan–Vese model [18] and GL minimisation (Algorithm 1). The dependence of the Chan–Vese model on the initial condition and its sensitivity to the model parameters may result in inaccurate detections, while the GL approach provides more reliable segmentation results. **a** Chan–Vese segmentation. **b** GL segmentation

Comparison with Chan–Vese model Due to the very irregular contours and the fine texture on the flycatcher's forehead, standard variational segmentation methods such as Canny edge detection or Mumford–Shah models [4,10,11,51] are not suitable for our task, as preliminary tests showed. Chan–Vese [18] is not suitable either, mainly because of the small scale detection limits, the dependence on the initial condition and the parameter sensitivity which may prevent us from an automatic and accurate segmentation of the tiny, yet characteristic feathers composing the blaze. In particular, the optimal parameters μ and ν appearing in the Chan–Vese functional and a sufficiently accurate initial condition need to be chosen typically by trial and error for every image at hand.

For comparison, we report in Fig. 4 the blaze segmentation results obtained by using Chan–Vese model (see [18,19]) and our graph-based method which will be described in more detail in the following.

4.1.1 Detailed Description of the Method

We divide our task into different steps:

1. For a given unsegmented image, we detect the head of the pied flycatcher through a comparison with a user-prepared dictionary (see Fig. 6) using GL segmentation Algorithm 1. Further computations are restricted to the head only.
2. Starting from the reduced image, a second step similar to Step 1 is now performed for the segmentation of the blaze, using again Algorithm 1. A dictionary of blazes is used an extended set of features is considered.
3. A refinement step is now performed in order to reduce the outliers detected in the process of segmentation.

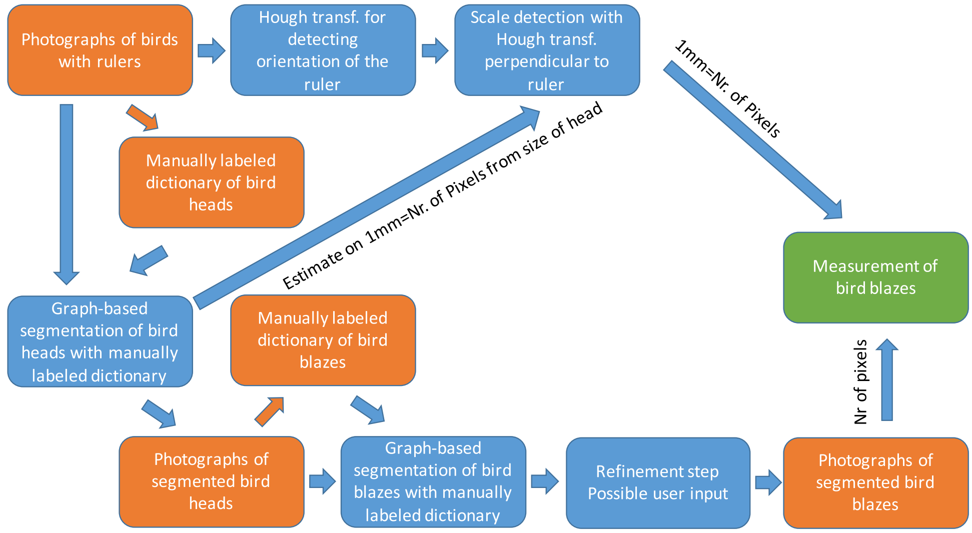


Fig. 5 The diagram describes the different steps of the segmentation/measurement procedure. Boxes requiring the user input are coloured *orange*, while the ones where the automatic segmentation/measurement steps are performed are coloured *blue*. The final objective is coloured *green* (Color figure online)

4. We use the Hough transform-based Algorithm 2 to detect in the image objects with a priori known geometrical shape (lines for linear rulers, circles for circular rulers) for the computation of the measurement scale.
5. The final step is the measurement of the values we are interested in (i.e. the perimeter of the blaze, its area and the width and height of the different blaze components). In the case of linear rulers, our results are given up to some error (due to the uncertainty in the detection of the measurement scale computed as average between ruler marks distances).

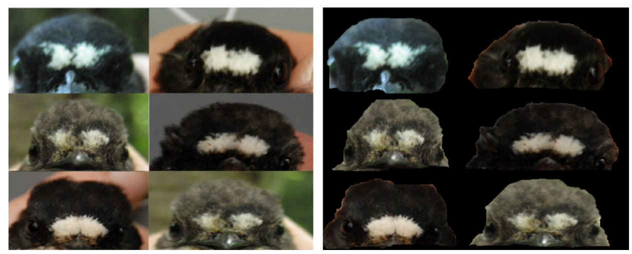


Fig. 6 Training dictionary for head detection: the heads are manually selected by the user and separated from the background. Then, the corresponding regions are labelled with 1, while the background is labelled by -1

Figure 5 shows a diagram which outlines the workflow of our method. In order to establish relations with behavioural and biological data confirming or contradicting the initial assumption of correlation between blaze size and higher attractiveness presented in the introduction [54], we have implemented a user ready programme for the quantitative analysis and measurements of the size of the bird blazes which is currently used by the Department of Zoology of the University of Cambridge. The results of this study will be the topic of a forthcoming paper [14].

In the following, we give more details about each step.

Step 1: Head detection We consider unlabelled images in the database and compare each of them with a dictionary of previously labelled images, see Fig. 6. The training regions (i.e. the heads) are labelled with a value 1, the background with value -1. Unlabelled regions are initialised with value 0.

The main computational difficulties in this step are due to size of the images considered. This may affect the performance of the algorithm as in order to apply the Nyström completion technique described in Sect. 2.4 one has to choose an adequate number of points whose features will approximate well the whole matrix. The larger and the more heterogeneous the image is, the larger will be the number of points needed to produce a sensible approximation. We circumvent this issue noticing that at this stage of the algorithm, we only need a rough detection of the head which will be used in the following for the accurate segmentation step. Thus, downscaling the image to a lower resolution (in our practice, reducing the resolution by ten times the original one) allows us to use a small number of Nyström sample points (typically 150–200) to produce an accurate result.

For this first step, we use as features simply the RGB intensities and proceed as described in Sect. 2.5. Once the head is detected, the resulting image is upscaled again to its original resolution. The solutions computed for the images in Fig. 1 are presented in Fig. 7.

Step 2: Blaze segmentation We consider now the reduced image from which we want to extract the flycatcher's blaze. Again, a dictionary of different blazes is manually created by the user (see Fig. 8). Again, training regions (the blazes) are labelled with value 1 and the black feathers in the background with value -1. As before, unlabelled regions are initialised

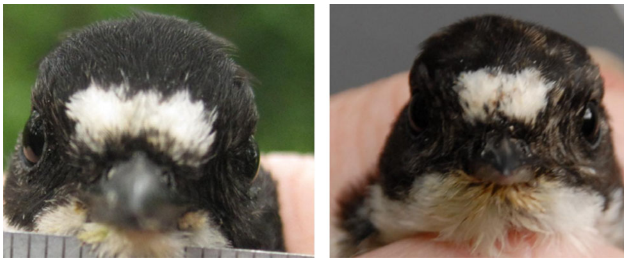


Fig. 7 Head detection from images in Fig. 1 using dictionary in Fig. 6

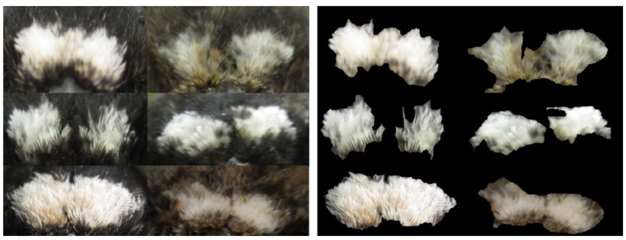


Fig. 8 Training dictionary for blaze segmentation. As in Fig. 7, blazes are manually selected by the user and labelled with 1, while black feathers on the background are labelled with −1

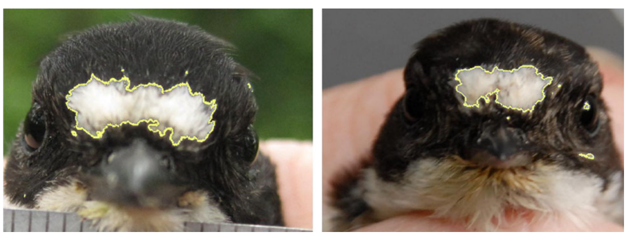


Fig. 9 Blaze segmentation

with value 0. At this stage, RGB intensities alone are not enough to differentiate the blazes from the background consistently in a large number of bird images, due to the colour difference between different blazes. For this step, an additional feature to be considered is the *texture* of the blaze. For this purpose, we use the MR8 texture features presented in [65] and proceed as detailed in Sect. 2.5. For 3×3 neighbourhoods, the feature vector for each pixel will be an element in $\mathbb{R}^{99}$, see Sect. 2.5. The Ginzburg–Landau minimisation provides the segmentation results shown in Fig. 9.

Step 3: Segmentation refinement This step uses very simple morphological operations in order to remove false detections obtained after Step 2. These can occur due to the choice of colour–texture-based features used to compute the feature vectors in Step 2. For instance, when looking at Fig. 9 (right) we observe that some bits on the left pied flycatcher's cheek have been detected as they exhibit similar texture properties as the ones on the blaze. In order to prevent this, our software asks the user to confirm whether the segmentation result provided is the expected one or if there are additional unwanted regions detected. If that is the case, using the MATLAB routine `bwconncomp` we label all the connected components segmented in the previous step, discarding among them all the ones whose area is smaller than a fixed percentage (we use 10 %) of the largest detected component (presumably, the blaze). This works well in practice, see Fig. 10. If the user is not satisfied, he or she can remove manually the unwanted regions. Figure 11 shows some blaze segmentation results after the refinement step.

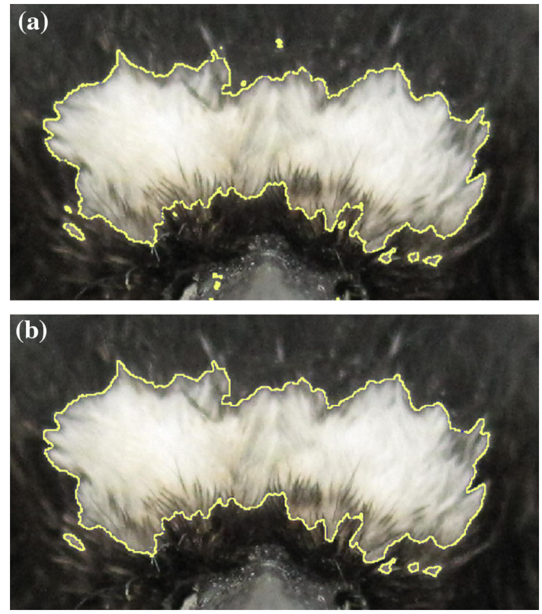


Fig. 10 Example of segmentation refinement. **a** Before refinement. **b** After refinement

Remark 4 (*Robustness to noise*) In order to reproduce the more realistic situation of images suffering from noise, we artificially added Gaussian noise with zero mean and different variances to some of the images in our database and performed the three analysis steps of our method. We report in Fig. 12, the results corresponding to two noise variances ($\sigma_1^2 = 0.02, \sigma_2^2 = 0.05$). The presence of noise influences both the head and blaze segmentation only slightly; the combination of RGB and texture features extracted in the neighbourhood of each point combined with the comparison to the dictionary makes the algorithm robust to noise and allows for an accurate blaze segmentation even in the noisy case.

Remark 5 (*Comparison with MBO segmentation*) We compare the blaze segmentation results obtained by minimising the discrete GL functional with the ones obtained using the segmentation algorithm considered in [44] as a variant of the classical Merriman–Bence–Osher (MBO) scheme [46]. More details on the connections between this approach and the GL minimisation as well as some insights into its numeri-

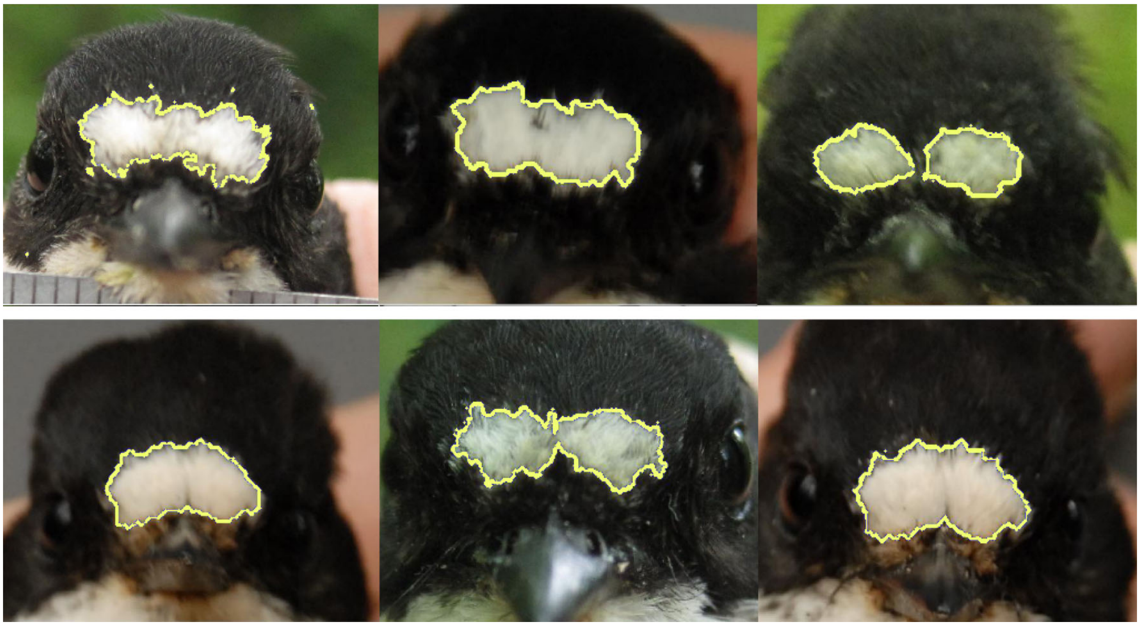


Fig. 11 Segmentation results after refinement step

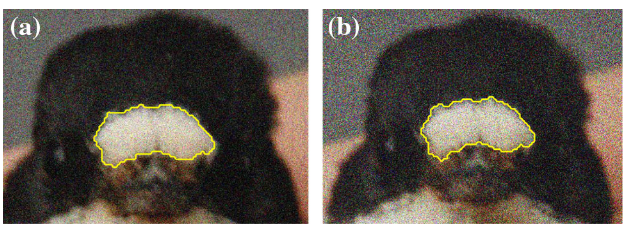


Fig. 12 Robustness to noise oscillations of GL minimisation for binary segmentation. Images have been artificially corrupted with Gaussian noise with zero mean and different variances. **a** $\sigma_1^2 = 0.02$. **b** $\sigma_1^2 = 0.05$

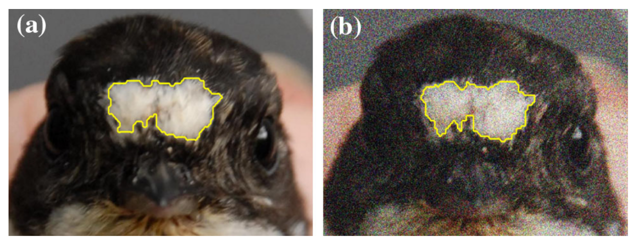


Fig. 13 Blaze segmentation results obtained by the MBO segmentation algorithm described in [44], after refinement step. Robustness to noise is observed in this case as well. In both numerical tests, the diffusion time is chosen as $\tau = 0.005$. **a** MBO result. **b** MBO result, $\sigma^2 = 0.05$

cal realisation are given in 'Appendix 2'. Following faithfully what is described in Sects. 2.2 and 2.4 for the graph and the operator construction step, respectively, we implemented the MBO segmentation algorithm following [44, Section 2]. We remark that the MBO method has the advantage of eliminating the dependence on the interface parameter ε of the GL functional by means of a combination of heat diffusion and a thresholding step. Instead of ε, the heat diffusion time τ needs to be chosen. In our numerical implementation, we used $\tau = 0.005$. Since no convex splitting strategies are required in this case, due to the absence of the non-convex double-well term, standard Fourier transform methods are used to solve the resulting time-stepping scheme. In Fig. 13, we report the blaze segmentation results obtained after applying a refinement step similar to the one described above: we note that a segmentation result comparable to the ones shown in Fig. 11 is obtained. Moreover, robustness to noise is observed also in this case. In terms of computational times, we observed that the replacement of the GL minimisation step with the MBO one did not affect significantly the speed of the segmentation algorithm.

Step 4: Measurement scale detection The images in our database are divided into two groups: the first is characterised by the presence of linear rulers, whereas the second contains circular rulers (Fig. 1). We thus need to use the Hough transform-based Algorithm 2 to detect lines or circles, respectively. The user is then required to tell the software which objects he or she wants to detect. In both cases, in order to avoid false detections (such as 'aligned' objects erroneously detected as lines, or circle-like objects wrongly considered as circles, see Fig. 14), a good candidate for a rough, sensible approximation of the measurement scale is needed as described in Sect. 3.1. In order to get this, we proceed as follows: after detecting the head as in Step 1, we use the option `EquivDiam` of the MATLAB routine

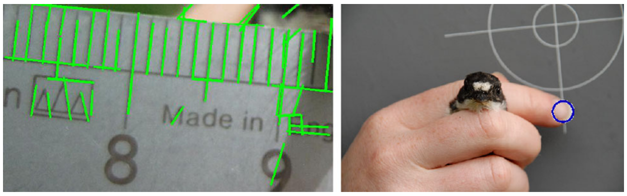


Fig. 14 Shadows, blur, noise or other objects in the image may disturb the detection

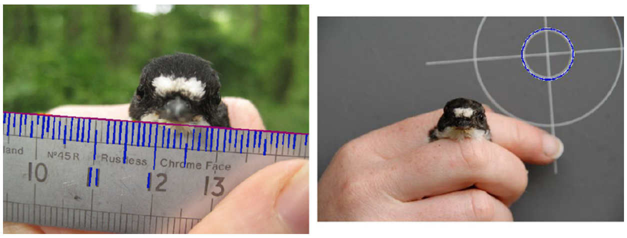


Fig. 15 Hough transform used for detecting geometrical objects. *Left lines* detection using MATLAB routines `houghlines`, `houghpeaks`. *Right circle* detection using MATLAB routine `imfindcircles`

`regionprops` to detect the diameter of the head region (in pixels). We then compare such measurement with pre-collected average measurements of head diameters of male pied flycatchers of a similar population (in *cm*), thus obtaining an initial approximation of the measurement scale. In the case of images containing linear rulers, this will serve as a spacing parameter s for the algorithm. In other words, only lines distant at least s pixels from each other will be considered. In the case of circular rulers, the same rough approximation will serve similarly as an indication of the range of values in which the Hough transform-based MATLAB function `imfindcircles` will look for circles' radii. For linear ruler images, the algorithm will look only for parallel lines aligned with a prescribed direction. We set this direction as the one perpendicular to the longest line in the image (since the expectation is that this longest line is the edge of the ruler). Results of this step are shown in Fig. 15.

Outliers removal for linear rulers The scale detection step described above may miss some lines on the ruler. This can be due to an oversmoothing in the denoising step, to high threshold values for edge detection or also to the choice of a large spacing parameter. Furthermore, as we can see from Figs. 1 and 15, we can reasonably assume that the ruler lies on a plane, but its bending can distort some distances between lines. Moreover, few other false line detections can occur (like the number 11 marked on the ruler main body in Fig. 15). To exclude these cases, we compute the distance (in pixels) between all the consecutive lines detected and eliminate possible outliers using the standard interquartile range (IQR) formula [62] for outliers' removal. Indicating by Q_1 and Q_3 the lower quartile and the third quartile, an outlier is every element not contained in the interval $[Q_1 - 1.5 * (Q_3 - Q_1), Q_3 + 1.5 * (Q_3 - Q_1)]$. Finally, we compute the empirical mean, variance and standard deviation (SD) of the values within this range, thus getting a final indication of the scale of the ruler together with an indicator of the precision of the method.

Step 5: Measurement Once the measurement scale has been detected, it is easy to get all the required measurements. We are interested in the perimeter, the area of the blaze and also the height and width of the whole blaze component. For linear rulers, due to the error committed in the scale detection step, these values present some uncertainty and variability (see above). In Table 1, we show the results of numerical tests on a sample of 30 images with linear rulers. For every image in the sample, we compute the standard deviation (SD) error and report in the table the minimum, maximum and average SD error over the single ones compute, together with the relative standard deviation (RSD) which gives a percentage indication of the error committed.

$$RSD := \frac{\sigma}{\bar{X}} \cdot 100,$$

where σ is the sample SD and $\bar{X}$ is the sample mean of measurements. We observe a minimum and maximum SD of 4.00 and 10.67 pixels, respectively, which compared to the dimension of the original image (3648 × 2736 pixels) suggests a reasonable precision. This is confirmed by the average SD value over the sample which is found to be 6.81 pixels. In percentage, the average error over the sample is 11.99 %. For circular rulers, we observed in all our experiments that an initial approximation of the range of values for the circle radius (see Step 4 above) results in a robust and typically outlier-free detection of the circular ruler and consequently in an accurate measurement of its radius; the only possible cause of variability and error is its bending.

Uncertainty in the measurements of lengths and areas is calculated with standard formulas in propagation of errors.

Despite these variabilities, our method is a flexible and semi-supervised approach for this type of problem. Further tests on the whole set of images and improvements on its accuracy are a matter of future research. The analysis of the resulting data measurements for the particular problem of flycatchers' blaze segmentation will be the topic of the forthcoming paper [14].

We compare in Table 2 the use of our combined approach and the use of the manual *line tool* of the ImageJ software for the measurement of the blaze area. Namely, we measured in Fig. 1b and in Fig. 1c the ruler scale by means of the ImageJ line tool by considering, for each image, two different 3-cm sections of the ruler; we then measured manually the number

Table 1 Precision of the measurement scale detection for linear rulers on a sample of 30 images

SD min	SD max	Mean SD	RSD min	RSD max	Mean RSD
4.01 pixels	10.67 pixels	6.81 pixels	6.59 %	17.36 %	11.99 %

The minimum, maximum and average standard deviation (SD) error together with the corresponding relative standard deviation (RSD) errors are reported

Table 2 Comparison between ruler scale detection by using manual IMAGEJ line tool and our Hough Transform (HT) method with corresponding measurements of the segmented blaze area obtained by using ImageJ 'magic-wand' (MW) tool [50], trapezium fitting (Trap.) [54] (see also Fig. 2) and the graph Ginzburg–Landau (GL) minimisation

	Scale (#$pixels$ = 1 mm)		Blaze area (pixel count)			Blaze area (mm^2)		
	Manual	HT (Ours)	MW	Trap.	GL (Ours)	MW	Trap.	GL (Ours)
Fig. 1b	70.2504	72.551	85,026	117,415	84,831	17.2288	23.7917	16.1164
Fig. 1c	71.863	71.8367	101,730	146,751	121,360	19.6980	28.4165	23.517

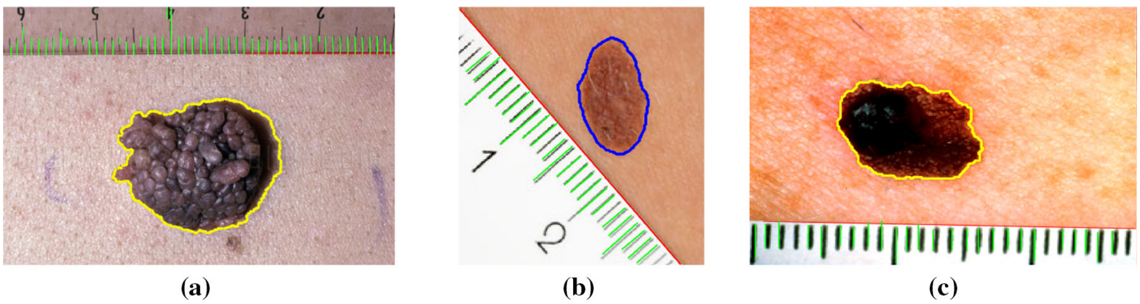


Fig. 16 Moles' detection using GL Algorithm 1 (**a**), the Chan–Vese model [18] (**b**, **c**), and measurement scale detection by Hough transform (Algorithm 2)

of pixels contained in each, divided each measurement by 30 and averaged the two results to obtain an estimate of the ruler scale (i.e. the number of pixels crossed by a 1-mm horizontal or vertical line segment). We then measured the area of the blaze after segmenting it by means of the 'magic-wand' [50] ImageJ tool and trapezium fitting [54] (see Fig. 2). The results are reported in Table 2 both as number of image pixels inside the blaze and in mm^2, where this second value has been calculated using the measurement scale detected as described above. We then repeated such measurements using our fully automated Hough transform method for ruler scale detection, reporting as above the measurements of the blaze area computed both as number of image pixels and in mm^2. We observe a good level of accuracy of our combined method (see also Table 1) with respect to the 'magic-wand' manual approach of Moreno [50], while, unsurprisingly, the blaze measurements obtained by pure trapezium fitting as proposed by Potti and Montalvo in [54] tend to overestimate the area of the blaze.

4.2 Moles Monitoring for Melanoma Diagnosis and Staging

In this section, we focus on another application of the scale detection Algorithm 2 in the context of melanoma (skin cancer) monitoring, see Fig. 3. Early signs of melanoma are sudden changes in existing moles and are encoded in the mnemonic ABCD rule. They are **A**symmetry, irregular **B**orders, variegated **C**olour and **D**iameter.[3] In the following, we focus on the D sign.

Due to their dimensions and their irregular shapes, moles are often very hard to measure. Typically, a common dermatological practice consists in positioning a ruler under the mole and then taking a picture with a professional camera. Sudden changes in the evolution of the mole are then observed by comparison between different pictures taken over time. Hence, their quantitative measurement may be an indication of a malignant evolution

In the following examples reported in Fig. 16, we use the graph segmentation approach described in algorithm 1 where texture characteristic regions are present (see Fig. 16a) and the Chan–Vese model [18] for images characterised by homogeneity of the mole and skin regions and the regularity of mole boundaries (Fig. 16b, c). For the numerical implementation, we use the freely available online IPOL Chan–Vese segmentation code [29]. Let us point out here that previous works using variational models for accurate melanoma seg-

[3] *Prevention: ABCD's of Melanoma* American Melanoma Foundation, http://www.melanomafoundation.org/prevention/abcd.htm.

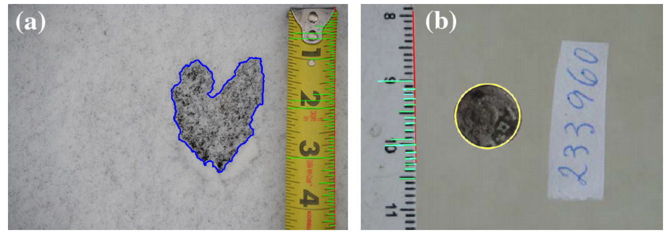


Fig. 17 The measurement scale has been detected only in a portion of the figure for the sake of reading clarity. **a** White-tailed deer tracks measurement. **b** Coin measurement, image taken from [34]

mentation already exist in the literature, see [1,17], but in those no measurement technique is considered.

4.3 Other Applications: Animal Tracks and Archaeological Finds' Measurement

We conclude this section presenting some other applications for the combined segmentation and scale detection models presented above.

The first application is the identification and classification of animals living in a given area through their soil, snow and mud footprints. Their quantitative measurement is also interesting in the study of the age and size a of a particular animal species. As in the problems above, such measurement very often reduces to a very inaccurate measurement performed with a ruler placed next to the footprint image. In Fig. 17a,[4] our combined method is applied for the measurement of a white-tailed deer footprint.

As a final application, we focus on archaeology. In many archaeological finds, objects need to be measured for comparisons and historical studies [34]. Figure 17b shows the application of our method to coin measurements. Due to its circular shape, for this image a combined Hough transform method for circle and line detection has been used. The example image is taken from [34] where the authors propose a gradient threshold-based method combined with a Fourier transform approach. Despite being quite efficient for the particular applications considered, such approach relies in practice on the good experimental setting in which the image is taken: almost noise-free images and very regular objects with sharp boundaries (mainly coins) and homogeneous backgrounds are considered. Furthermore, results are reported only for rulers with vertical orientation and no bending.

5 Conclusions

In this paper, we consider image segmentation applications involving measurement of a region's size, which has applications in several disciplines. For example, zoologists may be interested in quantitative measurements of some parts of the body of an animal, such as distinctive regions characterised by specific colours and texture, or in animal tracks to differentiate between individuals in the species. In medical applications, quantifying an evolving, possibly malignant mass (for instance, skin melanoma) is crucial for an early diagnosis and treatment. In archaeology, finds need to be measured and classified. In all these applications, often a common measurement tool is juxtaposed to the region of interest and its measurement is simply read directly from the image. This practice is typically inaccurate and imprecise, due to the conditions in which pictures are taken. There may be noise corrupting the image, the object to be measured may be hard to distinguish, and the measurement tool can be misplaced and far from the object to measure. Moreover, the scale of the image depends on the image itself due to the varying distance from the camera of the ruler and objects to measure.

The method presented (based on [9]) consists of a semi-supervised approach which, by training the algorithm with some examples provided by the user, extracts relevant features from the training image (such as RGB intensities, texture) and uses them to detect similar regions in the unknown image. Mathematically, this translates into the minimisation of the discrete Ginzburg–Landau functional defined on graphs. To overcome the computational issues due to the size of the data, Nyström matrix completion techniques are used and for the design of an efficient numerical scheme, convex splitting is applied. The measurement scale detection task is performed by using the Hough transform, a geometrical transformation which is capable of detecting objects with a priori known geometrical shapes (like lines on a ruler or circles with fixed diameter). Once the measurement scale is detected, all the measurements are converted into a unit of measure which is not image dependent.

Our method represents a systematic and reliable combination of segmentation approaches applied to several real-world image quantification tasks. The use of dictionaries, moreover, allows for flexibility as, whenever needed, the training database can be updated. With respect to recent developments [68] in the fields of data mining for the analysis of big data, where predictions are often performed using training sets and clustering, our approach represents an interesting alternative to standard machine learning (such as k-means) algorithms.

[4] Image from http://mamajoules.blogspot.co.uk/2015/01/a-naturalists-thoughts-on-animal-tracks.html.

Acknowledgements Many thanks to Colm Caulfield who has introduced HMR and the bird segmentation problem to us mathematicians and to Andrea Bertozzi for her very useful comments on the manuscript. LC acknowledges support from the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/H023348/1 for the University of Cambridge Centre for Doctoral Training, the Cambridge Centre for Analysis (CCA). CBS acknowledges support from EPSRC grants Nr. EP/J009539/1 and EP/M00483X/1. AF was supported under ONR grant N0001415WX01350. HMR was supported by research grants from the Royal Society Nr. RG64240 and the British Ecology Society Nr. BES2322, a Junior Research Fellowship from Churchill college. HMR is currently supported by an Institute Research Fellowship at the Institute of Zoology, by the Department of Zoology at the University of Cambridge, and Churchill College, Cambridge.

6 Appendix 1: The Nyström Extension

With respect to the eigenvalue problem formulation (2.19) and (2.20), we revise in this section the Nyström extension [53] in a matrix form.

Let us define first the submatrices $W_{XX} \in \mathbb{R}^L \times \mathbb{R}^L$ and $W_{XY} \in \mathbb{R}^L \times \mathbb{R}^{S-L}$ as

$$W_{XX} = \begin{pmatrix} w(x_1, x_1) & \cdots & w(x_1, x_L) \\ \vdots & \ddots & \vdots \\ w(x_L, x_1) & \cdots & w(x_L, x_L) \end{pmatrix},$$
$$W_{XY} = \begin{pmatrix} w(x_1, y_1) & \cdots & w(x_1, y_{S-L}) \\ \vdots & \ddots & \vdots \\ w(x_L, y_1) & \cdots & w(x_L, y_{S-L}) \end{pmatrix}. \tag{6.1}$$

Analogous definitions hold for W_{YY} and W_{YX}. Each of these matrices represents the submatrix having as elements the weights between the points in X, Y or between the two sets. With this notation, the whole matrix $W \in \mathbb{R}^S \times \mathbb{R}^S$ can be written in block form as

$$W = \begin{pmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YY} \end{pmatrix}, \qquad W_{YX} = W_{XY}^{\mathrm{T}}.$$

Similarly, vectors $v \in \mathbb{R}^S$ can be written as $v = (v_X^{\mathrm{T}}\, v_Y^{\mathrm{T}})^{\mathrm{T}}$. We focus on the spectral decomposition of the first block of W, that is W_{XX}. Since this matrix is symmetric, calling Θ_X the matrix $\Theta_X = \mathrm{diag}(\theta_1, \ldots, \theta_L)$ containing the eigenvalues of W_{XX}, then by the spectral theorem $W_{XX} = V_X \Theta_X V_X^{\mathrm{T}}$ (compare with (2.20)), with V_X being the orthogonal matrix having as columns the eigenvectors of W_{XX}. Writing (2.19) for $y \in Y$, in operator form, we obtain V_Y as

$$V_Y = W_{YX} V_X \Theta_X^{-1}.$$

The approximated eigenvectors of W can then be written in matrix form as

$$V = \begin{pmatrix} V_X \\ W_{YX} V_X \Theta_X^{-1} \end{pmatrix}. \tag{6.2}$$

Let us observe that

$$\begin{aligned} V \Theta_X V^{\mathrm{T}} &= \begin{pmatrix} V_X \\ W_{YX} V_X \Theta_X^{-1} \end{pmatrix} \Theta_X \begin{bmatrix} V_X^{\mathrm{T}} & (W_{YX} V_X \Theta_X^{-1})^{\mathrm{T}} \end{bmatrix} \\ &= \begin{pmatrix} V_X \Theta_X V_X^{\mathrm{T}} & W_{XY} \\ W_{YX} & W_{YX} W_{XX}^{-1} W_{XY} \end{pmatrix} \\ &= \begin{pmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YX} W_{XX}^{-1} W_{XY} \end{pmatrix} \approx W. \end{aligned} \tag{6.3}$$

Therefore, Nyström extension can be interpreted as the approximation $W \approx V \Theta_X V^{\mathrm{T}}$, under the approximation of W_{YY} given by $W_{YY} \approx W_{YX} W_{XX}^{-1} W_{XY}$. The quality of the approximation of the full W is quantified by the norm of the Schur complement $\| W_{YY} - W_{YX} W_{XX}^{-1} W_{XY} \|$, see [27]

Recalling the definition of the symmetric graph Laplacian L_s given by (2.8) and the relation between the spectral decomposition of W and the one of W in (2.16), we observe that a normalisation step now needs to be computed for obtaining the spectral decomposition of L_s. Defining $\mathbf{1}_L$ as the L-dimensional vector consisting of ones and $\mathbf{1}_{S-L}$ analogously, we use (6.3) and start computing the non-negative vector $d = (d_X^{\mathrm{T}} d_Y^{\mathrm{T}})^{\mathrm{T}}$ by

$$\begin{aligned} \begin{pmatrix} d_X \\ d_Y \end{pmatrix} &= \begin{pmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YX} W_{XX}^{-1} W_{XY} \end{pmatrix} \begin{pmatrix} \mathbf{1}_L \\ \mathbf{1}_{S-L} \end{pmatrix} \\ &= \begin{pmatrix} W_{XX} \mathbf{1}_L + W_{XY} \mathbf{1}_{S-L} \\ W_{YX} \mathbf{1}_L + W_{YX} W_{XX}^{-1} W_{XY} \mathbf{1}_{S-L} \end{pmatrix}. \end{aligned} \tag{6.4}$$

Therefore, the matrices W_{XX} and W_{XY} can be normalised simply by considering:

$$\begin{aligned} \hat{W}_{XX} &= W_{XX} . \Big/ \left(\sqrt{d_X} \otimes \sqrt{d_X}^{\mathrm{T}} \right), \\ \hat{W}_{XY} &= W_{XY} . \Big/ \left(\sqrt{d_Y} \otimes \sqrt{d_Y}^{\mathrm{T}} \right), \end{aligned} \tag{6.5}$$

where the division is intended elementwise and $\otimes$ is the standard vector tensor product.

A further step of normalisation is now needed since the approximated eigenvectors of W, i.e. the columns of the matrix V in (6.2), may not be orthogonal. Such normalisation may be obtained by using auxiliary unitary matrices. We refer the reader to [9, Section 3.2] for more details on this.

Once these additional normalisation steps are completed, we then get a spectral decomposition of W in terms of its eigenvalues $\hat{\lambda}_i$ and the corresponding normalised eigenvectors v_i, $i = 1, \ldots, S$. Therefore, recalling (2.16), the spectral decomposition of L_s is given in terms of the eigenvalue $1 - \hat{\lambda}_i$ and eigenvectors v_i.

7 Appendix 2: The MBO Scheme for Image Segmentation

As previously commented in Sect. 2.1, by taking the L^2 gradient descent of the Ginzburg–Landau functional defined in (2.1), one gets the well-known Allen–Cahn equation [3]:

$$u_t = \varepsilon \Delta u - \frac{1}{\varepsilon} W'(u), \tag{7.1}$$

which has often been studied for the modelling of several phase transition and separation problems and for the study of mean curvature flow (see, for example, [12]). In the limit $\epsilon \to 0$, solutions consist of two phases corresponding to the wells of W. In [56], it is shown that for rescaled solutions of Eq. (7.1), the interface between these phases evolves according to mean curvature flow. In [46], Merriman, Bence and Osher propose an alternative approach (later named MBO scheme) which, by using threshold dynamics, approximates the mean curvature flow of the interface at discrete times. As proved rigorously in [6], for small values of the interface parameter ε, the MBO scheme can then be used to solve Eq. (7.1) numerically.

In [44], the authors propose a variant of the MBO scheme as an alternative way to (approximately) minimise the graph GL functional with fidelity term, (2.9). Recalling the graph framework introduced in Sect. 2.2, the MBO segmentation starts from an initialisation U_1 given by (2.14) and computes for every $n \geq 1$ the new iterate U_{n+1} from U_n by applying sequentially the two following steps:

- *Step 1* (diffusion with forcing term): Starting from $U_n^1 = U_n$, solve for every $1 \leq k \leq K$ the discretised heat diffusion equation with fidelity term

$$\frac{U_n^{k+1} - U_n^k}{\tau} = -L_s\, U_n^{k+1} - \chi(x)(U_n^{k+1} - U_0), \tag{7.2}$$

  where $\tau := \frac{\Delta t}{K}$ is the heat diffusion time and K is the number of diffusion steps. Practically, τ has to be chosen small enough to approximate the motion by mean curvature and large enough to avoid *freezing* or *pinning*, which occurs when the diffusion time is so short that not enough mass diffuses along the edges of the network and the thresholding operation described in the following Step 2 leaves U_n unchanged.
- *Step 2* (thresholding): For every point x, set U_{n+1} as:

$$U_{n+1}(x) = \begin{cases} 1, & \text{if } U_n^K(x) \geq 0, \\ -1, & \text{if } U_n^K(x) < 0. \end{cases}$$

Numerically, (7.2) is solved at each diffusion time step $k\tau,\ k \geq 1$ by considering the spectral decomposition of U_n^k with respect to the eigenvectors of the operator L_s, similarly as in (2.15), and using classical Fourier transform methods to compute the new iterate U_n^{k+1}.

8 Appendix 3: The Hough Transform

The general idea behind the use of the Hough transform [23,35] is to map the ambient space to an auxiliary space, called the parameter space (as it is related to the parametric representation of the geometrical objects we are interested in). There, objects with specified shapes are easily recognisable as peaks of specific functions. Let us clarify these concepts with two examples.

Detecting line segments We start from the typical *slope-intercept form* of a line:

$$y = mx + b, \quad m, b, x, y \in \mathbb{R}. \tag{8.1}$$

Traditionally, the equation above is considered as a function of points with coordinates (x, y) satisfying equation (8.1) for *fixed* values of m and b. In other words, these values identify a specific straight line in the $x - y$ plane, cf. Fig. 18a. Rewriting (8.1) as $b = y - mx$ and keeping fixed the coordinates (x, y), we obtain a new equation of a straight line in the $m-b$ plane, cf. Fig. 18b, depicting the parameter space. If lines in the $m - b$ parameter space intersect, their sign-changed slopes (given by their x values) and m-intercepts (their y values) correspond to points lying on the same line in the $x - y$ plane. The (m, b) coordinates of the intersection point in parameter space specify the slope and x-intercept, respectively, of that line in the $x - y$ plane.

Hence, if we are given a black and white image in the $x-y$ plane, and for all coordinates (x, y) of black locations in the image, we draw the corresponding lines in the $m - b$ plane, intersection points of those lines will tell us which (x, y) locations in the image lie on the same line. Of course, any

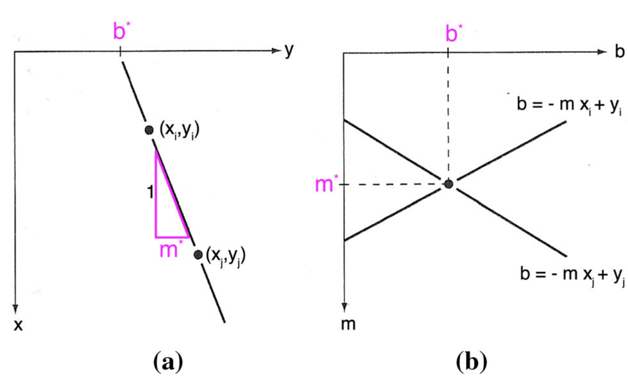


Fig. 18 Slope-intercept form, (8.1). Images edited from [31]. **a** $x - y$ plane. **b** $m - b$ plane

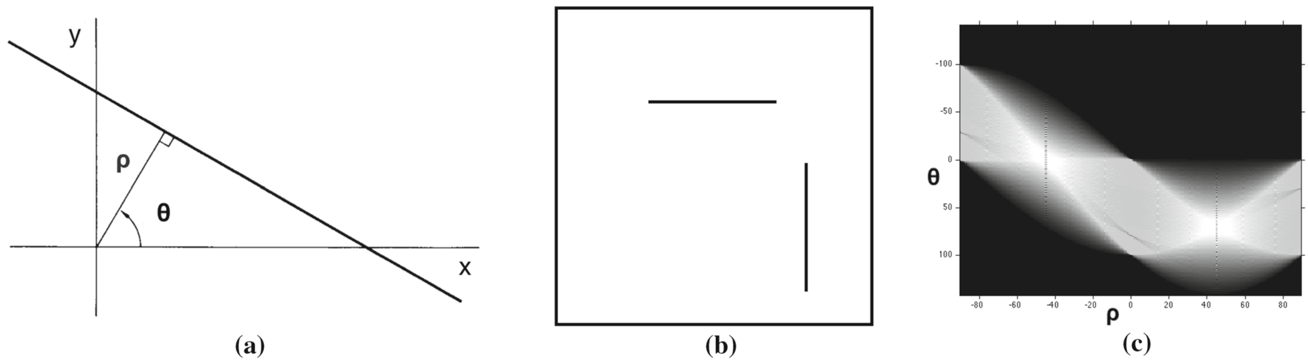


Fig. 19 Normal form: image and parameter spaces. **a** Normal parametrisation (8.2). **b** Binary image with two lines. **c** Parameter space

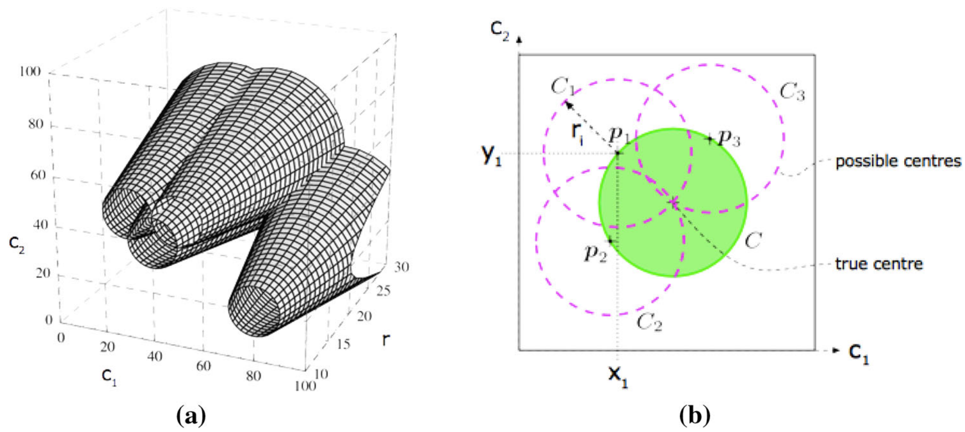


Fig. 20 Circular Hough transform. Images edited from [13]. **a** Cones in 3D parameter space. **b** Slice of 3D parameter space

two points lie on a line; thus, we are specifically interested in intersection points in the $m-b$ plane in which many different lines intersect, indicating the presence of an actual black line segment in the original image.

Drawbacks of this parametrisation are the need for an unbounded parameter space to describe near-vertical lines and the impossibility to describe a vertical line. One alternative is the *normal parametrisation* which views a straight line in $x-y$ space as the tangent line to a circle with radius ρ, touching the circle at angular coordinate θ, as illustrated in Fig. 19a, [23]. In $\rho-\theta$ parameter space, this leads to

$$\rho = x\cos\theta + y\sin\theta, \quad \theta \in [0,\pi]. \tag{8.2}$$

The objects in the parameter space are now sinusoidal curves, but again intersection points identify parameters for the points lying on the same straight line in the $x-y$ plane. Figure 19b, c shows a binary image with two black straight lines and the corresponding parameter space. The bright spots in the parameter space indicate a large number of intersections, thus identifying the two lines in the original image.

Detecting circles Analogously to what we did above, when looking for circular structures in a given image, we consider for $(x, y) \in \mathbb{R}^2$ the parametric representation of a circle,

$$r^2 = (x-c_1)^2 + (y-c_2)^2, \tag{8.3}$$

where $r > 0$ is the radius of the circle and $(c_1, c_2) \in \mathbb{R}^2$ are the coordinates of its centre. Every point (x, y) lying on the circle satisfies Eq. (8.3) for fixed r, c_1 and c_2. As before, we now consider Eq. (8.3) in the three-dimensional parameter space $c_1 - c_2 - r$ for fixed x and y. Here, the objects of interest are cone-shaped surfaces, as shown in Fig. 20a. Their intersection identifies the desired values of r, c_1 and c_2 in Eq. (8.3), see Fig. 20b.

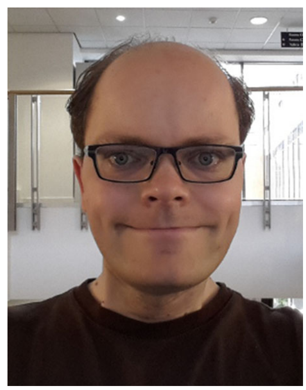
References

1. Abbas, Q., Fondon, I., Sarmiento, A., Emre Celebi, M.: An improved segmentation method for non-melanoma skin lesions using active contour model. In: Image Analysis and Recognition, Lecture Notes in Computer Science, pp. 193–200 (2014)
2. Abramoff, M.D., Magalhães, P.G., Ram, S.J.: Image processing with ImageJ. Biophotonics Int. **11**(7), 36–42 (2004)
3. Allen, S.M., Cahn, J.W.: A microscopic theory for the antiphase boundary motion and its application to antiphase domain coarsening. Acta Metall. **27**, 1085–1095 (1979)

4. Ambrosio, L., Tortorelli, V.M.: Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. Commun. Pure Appl. Math. **43**, 999–1036 (1990)
5. Ambrosio, L., Tortorelli, V.M.: On the approximation of free discontinuity problems. Boll. Un. Mat. Ital. B (7) **6**(1), 105–123 (1992)
6. Barles, G., Georgelin, C.: A simple proof of convergence for an approximation scheme for computing motions by mean curvature. SIAM J. Numer. Anal. **32**, 484–500 (1995)
7. Belongie, S., Fowlkes, C., Chung, F., Malik, J.: Partitioning with Indefinite Kernels Using the Nyström Extension. ECCV, Copenhagen (2002)
8. Beltrametti, M.C., Massone, A., Piana, M.: Hough transform of special classes of curves. SIAM J. Imaging Sci. **6**(1), 391–412 (2013)
9. Bertozzi, A.L., Flenner, A.: Diffuse interface models on graphs for classification of high dimensional data. Multiscale Model. Simul. **10**(3), 1090–1118 (2012)
10. Braides, A.: Approximation of Free-discontinuity Problems, vol. 1694 of Lecture Notes in Mathematics. Springer, Berlin (1998)
11. Braides, A., Dal Maso, G.: Non-local approximation of the Mumford–Shah functional. Calc. Var. Partial Differ. Equ. **5**(4), 293–322 (1997)
12. Brokate, M., Sprekels, J.: Hysteresis and Phase Transitions, Applied Mathematical Sciences, vol. 121. Springer, New York (1996)
13. Burger, W., Burge, M.J.: Digital Image Processing—An Algorithmic Introduction Using Java. Springer, Berlin (2009)
14. Calatroni, L., van Gennip, Y., Schönlieb, C.-B., Flenner, A., Coffey, P., Rowland, H.M.: Intraspecific variation in the head patches of male pied flycatchers (in preparation)
15. Canny, J.F.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(6), 679–698 (1986)
16. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. **22**(1), 61–79 (1997)
17. Cavalcanti, P.G., Scharcanski, J.: Macroscopic pigmented skin lesion segmentation and its influence on lesion classification and diagnosis. In: Color Medical Image Analysis, Lecture Notes in Computational Vision and Biomechanics, vol. 6. Springer, Berlin (2013)
18. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Trans. Image Process. **10**(2), 266–277 (2001)
19. Chan, T.F., Sanberg, B., Vese, L.A.: Active contours without edges for vector-valued images. J. Vis. Commun. Image Represent. **11**, 130–141 (2000)
20. Chung, F.R.K.: Spectral Graph Theory, volume 92 of CBMS Regional Conference Series in Mathematics. Published for the Conference Board of the Mathematical Sciences, Washington, DC (1997)
21. Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. Proc. Natl. Acad. Sci. **102**(71), 7426–7431 (2005)
22. Dale, S., Slagsvold, T., Lampe, H.M., Sætre, G.P.: Population divergence in sexual ornaments: the white forehead patch of Norwegian pied flycatchers is small and unsexy. Evolution. **53**, 1235–1246 (1999)
23. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM **15**, 11–15 (1972)
24. Esedoglu, S.: Blind deconvolution of bar code signals. Inverse Prob. **20**(1), 121–135 (2004)
25. Esedoglu, S., Shen, J.: Digital inpainting based on the Mumford–Shah–Euler image model. Eur. J. Appl. Math. **13**(4), 353–370 (2002)
26. Esedoglu, S., Tsai, Y.-H.R.: Threshold dynamics for the piecewise constant Mumford–Shah functional. J. Comput. Phys. **211**(1), 367–384 (2006)
27. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral grouping using the Nyström method. IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 214–225 (2004)
28. Garcia-Cardona, C., Merkurjev, E., Bertozzi, A.L., Flenner, A., Percus, A.G.: Multiclass data segmentation using diffuse interface methods on graphs. IEEE Trans. Pattern Anal. Mach. Intell. **36**(8), 1600–1613 (2014)
29. Getreuer, P.: Chan–Vese segmentation. Image Process. On Line **2**, 214–224 (2012)
30. Gilboa, G., Osher, S.: Nonlinear operators with applications to image processing. Multiscale Model. Simul. **7**(3), 1005–1028 (2008)
31. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall, London (2002)
32. Grah, J.: Methods for automatic mitosis detection and tracking in phase contrast images. M.Sc. thesis, University of Muenster (2014)
33. Guattery, S., Miller, G.L.: On the quality of spectral separators. SIAM J. Matrix Anal. Appl. **19**(3), 701–719 (1998)
34. Herrmann, M., Zambanin, S., Kampel, M.: Image-based measurement of ancient coins, making history interactive. In: Computer Applications and Quantitative Methods in Archaeology (CAA), Proceedings of the 37th International Conference, Archaeopress, Oxford, pp. 117–121 (2010)
35. Hough, P.V.C.: Method and means for recognizing complex patterns. US Patent 3,069,654 (1962)
36. Hu, H., Sunu, J., Bertozzi, A.L.: Multi-class graph Mumford–Shah model for plume detection using the MBO scheme. In: Tai X.-C., et al. (eds.)Proceedings of EMMCVPR Hong Kong 2015, Springer Lecture Notes in Computer Science, vol. 8932, pp. 209–222 (2015)
37. Järvistö, P.E., Laaksonen, T., Calhim, S.: Forehead patch size predicts the outcome of male–male competition in the pied flycatcher. Ethology **119**(8), 662–670 (2013)
38. Kaas, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vis. **1**(4), 321–331 (1988)
39. Kohn, R.V., Sternberg, P.: Local minimisers and singular perturbation. Proc. R. Soc. Edinb. Sect. A **111**(1–2), 69–84 (1989)
40. Lundberg, A., Alatalo, R.V.: The Pied Flycatcher. T & AD Poyser, London (1992)
41. Massone, A.M., Perasso, A., Campi, C., Beltrametti, M.C.: Profile detection in medical and astronomical images by means of the Hough transform of special classes of curves. J. Math. Imaging Vis. **51**(2), 296–310 (2015)
42. Merkujev, E., Bae, E., Bertozzi, A., Tai, X.-C.: Global binary optimization on graphs for classification of high-dimensional data. J. Math. Imaging Vis. **52**(3), 414–435 (2015)
43. Merkurjev, E., Garcia-Cardona, C., Bertozzi, A.L., Flenner, A., Percus, A.G.: Diffuse interface methods for multiclass segmentation of high-dimensional data. Appl. Math. Lett. **33**, 29–34 (2014)
44. Merkurjev, E., Kostic, T., Bertozzi, A.L.: An MBO scheme on graphs for segmentation and image processing. SIAM J. Imaging Sci. **6**(4), 1903–1930 (2013)
45. Merkurjev, E., Sunu, J., Bertozzi, A.L.: Graph MBO method for standoff detection in hyperspectral video. In: Proc. Int. Conf. Image Proc, Paris, pp. 689–693 (2014)
46. Merriman, B., Bence, J., Osher, S.: Diffusion generated motion by mean curvature, In: Proceedings of the Computational Crystal Growers Workshop, Providence, Rhode Island, pp. 79–83 (1992)
47. Modica, L., Mortola, S.: Il limite nella Γ-convergenza di una famiglia di funzionali ellittici. Boll. Unione Mat. Ital. V. Ser. **A 14**, 526–529 (1977)
48. Modica, L., Mortola, S.: Un esempio di Γ-convergenza. Boll. Unione Mat. Ital. V. Ser. **B 14**, 285–299 (1977)
49. Morales, J., Moreno, J., Merino, S., Sanz, J.J., Tomas, G., Arriero, E., et al.: Female ornaments in the Pied Flycatcher Ficedula hypoleuca: associations with age, health and reproductive success. Ibis **149**(2), 245–254 (2007)

50. Moreno, J., Velando, A., Ruiz-De-Castañeda, R., Cantarero, A., Gonzalez-Braojos, S., Redondo, A.: Plasma antioxidant capacity and oxidative damage in relation to male plumage ornamental traits in a montane Iberian Pied Flycatcher Ficedula hypoleuca population. Acta Ornithol. **46**(1), 65–70 (2011)
51. Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math. **42**, 577–685 (1989)
52. Naeini, M.M., Dutton, G., Rothley, K., Mori, G.: Action recognition of insects using spectral clustering. In: MVA 2007 IAPR Conference on Machine Vision Applications (2007)
53. Nyström, E.J.: Über die Praktische Auflösung von Linearen Integralgleichungen mit Anwendungen auf Randwertaufgaben der Potentialtheorie. Commentationes Physico-Mathematicae **4**(15), 1–52 (1928)
54. Potti, J., Montalvo, S.: Male arrival and female mate choice in Pied Flycatchers Ficedula Hypoleuca in central Spain. Ornis Scand. **22**(1), 45–54 (1991)
55. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D **60**, 259–268 (1992)
56. Rubinstein, J., Sternberg, P., Keller, J.B.: Fast reaction, slow diffusion, and curve shortening. SIAM J. Appl. Math. **49**, 116–133 (1989)
57. Ruuskanen, S., Lehikoinen, E., Nikinmaa, M., Siitari, H., Waser, W., Laaksonen, T.: Long-lasting effects of yolk androgens on phenotype in the pied flycatcher (*Ficedula hypoleuca*). Behav. Ecol. Sociobiol. **67**(3), 361–372 (2013)
58. Sætre, G.-P., Mourn, T., Bures, S., Kral, M., Adamjan, M., Moreno, J.: A sexually selected character displacement reinforces predating isolation. Nature **387**, 589–592 (1997)
59. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
60. Sirkiä, P.M., Laaksonen, T.: Distinguishing between male and territory quality: females choose multiple traits in the pied flycatcher. Anim. Behav. **78**(5), 1051–1060 (2009)
61. Stör, M., Wagner, F.: A simple min-cut algorithm. J. ACM **44**(4), 585–591 (1997)
62. Upton, G., Cook, I.: Understanding Statistics. Oxford University Press, Oxford (1997)
63. van Gennip, Y., Bertozzi, A.L.: Γ-convergence of graph Ginzburg–Landau functionals. Adv. Differ. Equ. **17**(11–12), 1115–1180 (2012)
64. van Gennip, Y., Guillen, N., Osting, B., Bertozzi, A.L.: Mean curvature, threshold dynamics, and phase field theory on finite graphs. Milan J. Math. **82**(1), 3–65 (2014)
65. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. Int. J. Comput. Vis. 62(61), 61–81 (2005)
66. Vese, L.A., Chan, T.F.: Reduced non-convex functional approximation for image restoration & segmentation. UCLA Report Cam 97–56 (1997)
67. von Luxburg, U.: A tutorial on spectral clustering. Technical Report No. TR-149, Max Planck Institute for Biological Cybernetics (2006)
68. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Elsevier, Amsterdam (2011)

Luca Calatroni completed his Master Degree in Applied Mathematics at the University of Pavia (Italy) in 2011. He completed his Ph.D. in 2015 under the supervision of Dr. Carola-Bibiane Schoenlieb at the Cambridge Centre for Analysis (CCA), a 4-year Doctoral Training Centre (DTC) at the University of Cambridge (UK) as a member of the Cambridge Image Analysis research group. Currently, he is working as an Experienced Researcher (ER) within the *Nano2fun* Marie-Curie FP7 ITN on image super-resolution problems for microscopy applications. Concurrently, he is also working as a postdoctoral fellow in the MIDA group at the University of Genova (Italy) on some ill-posed applied inverse problems. His mathematical interests lie in the fields of PDE and variational approaches for imaging problems, non-smooth optimisation and inverse problems regularisation.



Yves van Gennip received M.Sc. degrees in mathematics and theoretical physics from Utrecht University, in 2004, and a Ph.D. degree in mathematics from the Eindhoven University of Technology, in 2008. He was a PIMS Postdoctoral Fellow with Simon Fraser University from 2008 until 2010, when he joined the University of California at Los Angeles as an Assistant Adjunct Professor. Since 2013, he has been a Lecturer/Assistant Professor with the University of Nottingham. His research interests are in partial differential equations, variational methods, networks and applications to image processing, data analysis, and pattern formation.

Carola-Bibiane Schönlieb is a Reader in Applied and Computational Analysis, head of the Cambridge Image Analysis (CIA) group at the Department of Applied Mathematics and Theoretical Physics (DAMTP), University of Cambridge. Moreover, she is the Director of the Cantab Capital Institute for the Mathematics of Information, Co-Director of the PSRC Centre for Mathematical and Statistical Analysis of Multimodal Clinical Imaging, a Fellow of Jesus College, Cambridge, and co-leader of the IMAGES network. Carola obtained her degree in Mathematics at the University of Salzburg in 2004 and her Ph.D. in Mathematics at the University of Cambridge in 2009. After one year of postdoctoral activity at the University of Göttingen (Germany), she became a Lecturer in DAMTP, promoted to Reader in 2015. She is an internationally acknowledged researcher in applied and computational mathematics with particular focus on partial differential equations and variational methods for image analysis and image reconstruction, including work on biomedical imaging, remote sensing and image inpainting. She has active interdisciplinary collaborations with clinicians, biologists, physicists, art conservators, chemical engineers and plant scientists.

Hannah M. Rowland graduated from the University of Liverpool in 2007 with Ph.D. in evolutionary and behavioural ecology. From 2007 to 2010, she was a NERC Postdoctoral Research Associate at the University of Liverpool, and from 2010 to 2011, was a NERC Postdoctoral Research Associate at the University of Glasgow. Since 2011 she has been at the University of Cambridge: from 2011 to 2014 as a Junior Research Fellow at Churchill College and from 2014 as a Lecturer in Ecology and Evolution. In 2014 she joined the Institute of Zoology, at the Zoological Society of London, as a Research Fellow. She holds the IoZ Research Fellowship and Cambridge Lectureship concurrently. Her research interests are in the evolution and function of animal signals, and signal perception.

Arjuna Flenner received his Ph.D. in Physics at the University of Missouri–Columbia located in Columbia, MO, in the year 2004. His major emphasis was mathematical physics. His research interests at the Naval Air Weapons Center at China Lake include image processing, machine learning, statistical pattern recognition and computer vision. He won the 2013 Dr. Delores M. Etter award for Navy Scientist and Engineer, and the 2014 SIAM outstanding paper award for joint work with Andrea Bertozzi on Diffuse interface models on graphs. In his work, he investigates automated image understanding algorithms for advanced Naval capabilities. His major research areas are non-local operators, Geometric Diffusion, Graph Theory, nonparametric Bayesian analysis, and *a contrario* hypothesis testing methods. Furthermore, during his tenure at China Lake he established collaborative efforts with Sun Biomedical Technologies, Loma Linda University, UCLA, Duke, and NGA. In 1997–2001, he was a GAANN Fellow of US Department of Energy.