A Phase Field Model for Continuous Clustering on Vector Fields

Harald Garcke, Tobias Preußer, Martin Rumpf,
Alexandru C. Telea, Ulrich Weikard, and Jarke J. van Wijk

Abstract—A new method for the simplification of flow fields is presented. It is based on continuous clustering. A well-known physical clustering model, the Cahn Hilliard model, which describes phase separation, is modified to reflect the properties of the data to be visualized. Clusters are defined implicitly as connected components of the positivity set of a density function. An evolution equation for this function is obtained as a suitable gradient flow of an underlying anisotropic energy functional. Here, time serves as the scale parameter. The evolution is characterized by a successive coarsening of patterns—the actual clustering—during which the underlying simulation data specifies preferable pattern boundaries. We introduce specific physical quantities in the simulation to control the shape, orientation and distribution of the clusters as a function of the underlying flow field. In addition, the model is expanded, involving elastic effects. In the early stages of the evolution shear layer type representation of the flow field can thereby be generated, whereas, for later stages, the distribution of clusters can be influenced. Furthermore, we incorporate upwind ideas to give the clusters an oriented drop-shaped appearance. Here, we discuss the applicability of this new type of approach mainly for flow fields, where the cluster energy penalizes cross streamline boundaries. However, the method also carries provisions for other fields as well. The clusters can be displayed directly as a flow texture. Alternatively, the clusters can be visualized by iconic representations, which are positioned by using a skeletonization algorithm.

Index Terms—Flow visualization, clustering, Cahn-Hilliard, multiscale, nonlinear diffusion, finite elements, skeletonization.

1 Introduction

Today, fast computing hardware and efficient numerical algorithms enable highly detailed and large scientific simulations which deliver enormous amounts of data. Various visualization strategies have been proposed to represent such data in an intuitively understandable way.

The larger and more complex the simulation results become, the stronger is the need for a suitable multiscale visualization approach. Simplified representations of the data, useful for seeing the global pattern, can be gradually refined for further insight. Moreover, different viewers need different representations. Numerical experts might want to see the raw data in full detail, technological experts might want to see certain features such as vortices, whereas management might need a simplified presentation.

Clustering, well-known from statistics, is such a multiscale approach. Data are grouped in successively larger sets of strong internal correlation. Many techniques are available for scattered and scalar data, e.g., based on wavelet or Fourier analysis [12], [28]. However, for vector data, only a few multiscale visualization methods are available. The most ubiquitous vector field simplification method is still regular subsampling, which is well-known for producing aliases (see, e.g., Fig. 1). Turk and Banks [25] use an energy minimizing approach to place equally distributed streamlines at a user prescribed resolution on the screen. Selected streamline drawings are furthermore considered by Jobard and Lefer [10].

Recently, two approaches for clustering vector data have been proposed. In both approaches, a hierarchical clustering tree is produced and the resulting clusters are visualized with arrows. Heckel et al. [9] start from scattered points with vector data. Initially, all points are stored in a single cluster, which is recursively split in a top-down manner. At each step, the cluster with the strongest discrepancy between streamlines generated by the original field and its approximation by the cluster is bisected with a plane, using principal component analysis. The resulting clusters are guaranteed to be convex as a result of this bisection approach. However, accurately representing complex fields with convex clusters may require a large cluster count.

Telea and Van Wijk [23] use a bottom-up approach. Initially, each data point is a cluster; next, these clusters are merged. In each step, the most similar clusters are merged, according to a measure of the difference in position and orientation of the vectors that represent the clusters. The cluster shapes are constrained only indirectly by adapting the weights of the various terms in the error measure. However, this method is sometimes sensitive to the mentioned weight tuning.

Here, we propose a continuous clustering method based on a phase separation model which leads to a diffusion problem. The main difference to the other approaches is that no Boolean merging or splitting decisions have to be made. Instead, a suitable diffusion process continuously

- *H. Garcke, T. Preußer, M. Rumpf, and U. Weikard are with the Institute for Applied Mathematics, University of Bonn, Wegelerstrasse 6, 53115 Bonn, Germany. E-mail: {harald, tpreuss, rumpf, wkd}@iam.uni-bonn.de.*
- *A. Telea and J. van Wijk are with the Department of Mathematics and Computer Science, Eindhoven University of Technology, Den Dolech 2, 5600 MB, Eindhoven, The Netherlands.*
  E-mail: {alext, vanwijk}@win.tue.nl.

Manuscript received 30 Mar. 2001; accepted 7 May 2001.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number 114133.

1077-2626/01/$10.00 © 2001 IEEE

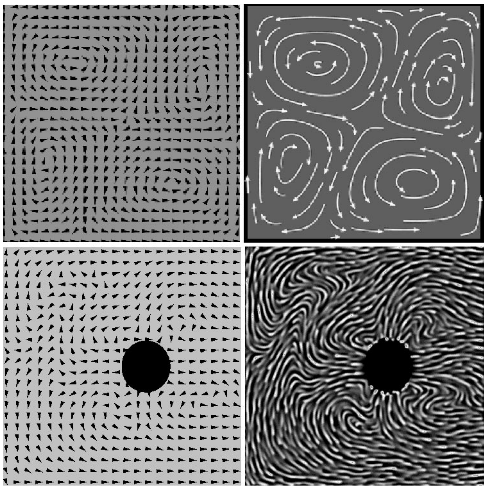


Fig. 1. Vector field visualization: hedgehog plots (left), proposed clustering methods (right).

enhances strong correlations in the cluster sets. In contrast to Heckel's method, the clusters that we generate are not necessarily convex. Hence, curved flow fields can be represented more effectively. Fig. 1 shows two vector fields visualized with the classical hedgehog plot (left) and with two variants of our method (right).

Our approach is motivated by a well-known physical model for phase separation in binary alloys which can be understood as a clustering of material in order to decrease the free energy of the physical system. As a major application, we consider clustering on flow fields. The method is related to multiscale image processing methodology which leads to second order parabolic equations, whereas our model here will be a fourth oder problem. Perona and Malik [17] have introduced a continuous diffusion model which allows the denoising of images together with edge enhancing. The recovery of lower dimensional structures in images is analyzed by Weickert [27], who introduced an anisotropic nonlinear diffusion method where the diffusion matrix depends on the so-called structure tensor of the image. Preußer and Rumpf presented an efficient implementation for large scale image data [19] and used an anisotropic diffusion approach for flow visualization [20].

In detail, the aims of our method are:

- to extract a collection of nicely shaped subsets of the physical domain, where each of them is being characterized by a strong correlation in the underlying physical data and all together are supposed to cover an approximately fixed fraction of the domain,
- to consider not only one such representation, but a scale of them, ranging from fine granularity in the subdivision to very few and coarse cluster sets.

This multiscale should enable the exploration of complicated simulation data and the visual perception of correlations in such data sets at different resolutions. In our model, the clusters will be represented implicitly by a scalar function evolving in time. In addition, we expand the model, incorporating a contribution to the energy due to elastic effects. We are thereby able to influence the distribution of the particles and, for small evolution times—not yet in the range of the actual clustering results—we obtain images which show shear layer type patterns. Furthermore, we use an upwind idea to obtain drop-shaped particles which clearly outline the flow field direction.

Concerning the graphical representation, we could straightforwardly use a color coded representation of this function on the physical domain as a texture. In the last decade, a variety of such texturing methods have been presented for flow visualization. We mention here the spot noise technique by de Leeuw and Van Wijk [5], the line integral convolution method by Cabral and Leedom [2], several improvements and modifications of this method [29], [7], [21], and the already mentioned nonlinear anisotropic diffusion method [20]. As an alternative to the above, we use the actual clustering as a precomputing step and pipe the output into an iconic representation approach. Thus, the distinct subsets at any scale are represented by suitable graphical icons. This allows a further reduction of graphically represented data while maintaining and strengthening the informational content.

The ingredients of our continuous clustering strategy are as follows:

- We formulate an *evolution problem* for a function which implicitly describes the set of clusters. The evolution can be interpreted as the gradient flow with respect to an appropriate energy.
- There are *two major energy contributions*. The first one leads to *the nucleation of cluster sets* on the physical domain. The second one gives rise to a *successive coarsening of the clusters*.
- Depending on the underlying physical data, *surface segments are weighted depending on their location and orientation*. That is, surfaces are considerably penalized if they are oriented in cross direction to the correlation; otherwise, their energy contribution is kept small. Several energy components can be defined to constrain the clusters' shapes in relation to various quantities in the flow dataset.
- On any scale, a skeletonization method is used to reduce the informational content of the cluster sets to their essence, which is to be further visualized.
- Finally, geometric icons are selected to represent the extracted skeleton information graphically, e.g., with arrows in the case of vector data.

Let us emphasize that the actual physical data enter the clustering method only via the anisotropic energy. Moreover, the evolved function is solely used to define the cluster sets without any further physical meaning.

As application, we mainly consider flow fields, where the concept of correlation along streamlines is near at hand. Nevertheless, the methodology is not restricted to flow visualization and is thus presented here for more general data.

The organization of the paper is as follows: In Section 2, we outline the physical model of phase separation in binary alloys which motivates this work. Section 3 extends this model by taking into account physical elasticity. In Section 4, we expand this model and interpret it in terms of a multiscale cluster analysis. In Section 5, we describe how

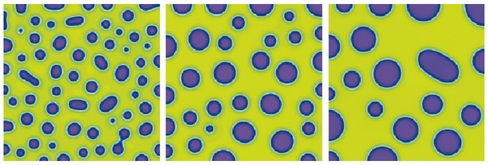


Fig. 2. Three timesteps of the original Cahn-Hilliard evolution.

the cluster shape can be controlled to visualize the direction of a vector field. Section 6 discusses how anisotropic elasticity can be used to visualize shear layer type patterns. A finite element discretization is described in Section 7. In Section 8, the skeletonization approach is outlined and we discuss graphical icons in case of vector data. Finally, in Section 9, we discuss the results and, in Section 10, we draw conclusions.

2 Reviewing a Physical Clustering Model

Before we discuss our model of continuous clustering on simulation data we will review in this section a physical model for clustering in metallic alloys, which goes back to Cahn and Hilliard [3]. The Cahn-Hilliard model was introduced to describe phase separation and coarsening in binary alloys. Phase separation occurs when a uniform mixture of the alloy is quenched below a certain critical temperature beneath which the uniform mixture becomes unstable. As a result, a very fine microstructure of two spatially separated phases with different concentrations develops. In later stages of the evolution, on a much slower time scale than the initial phase separation, the structures become coarser: either by merging of particles or by growing of bigger particles at the cost of smaller ones. This coarsening can be understood as a clustering, where the system mainly tries to decrease the surface energy of the particles, which leads to coarser and coarser structures during the evolution. In the basic Cahn-Hilliard model, this surface energy is isotropic. There are no preferred directions of the interfaces. Hence, the particles tend to be ball-shaped (cf. Fig. 2).

We now briefly outline the basic ideas of the Cahn-Hilliard model. For more details we refer to the review papers by Elliott [6] and Novick-Cohen [15]. The Cahn-Hilliard model is based on a Ginzburg-Landau free energy, which is a functional in terms of the concentration difference ρ of the two material components. The Ginzburg-Landau free energy E is defined to be

$$E(\rho) := \int_\Omega \left\{ \Psi(\rho) + \frac{\gamma}{2} |\nabla \rho|^2 \right\},$$

where Ω is a bounded domain. The first term in the free energy, $\Psi(\rho)$, is the chemical energy density and typically has a double well form. In this paper, we take

$$\Psi(\rho) = \frac{1}{4}\left(\rho^2 - \beta^2\right)^2$$

with a constant $\beta \in (0, 1]$ (cf. Fig. 3). We note that the system is locally in one of the two phases if the value of ρ is close to one of the two minima $\pm\beta$ of Ψ. The diffusion equation for the concentration ρ is given by

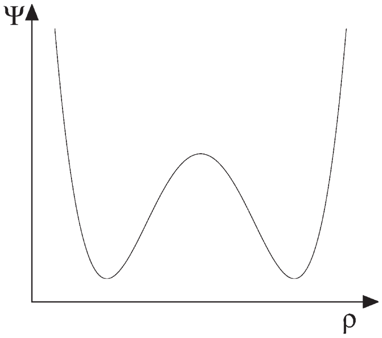


Fig. 3. Chemical energy as function of concentration.

$$\frac{\partial \rho}{\partial t} = \Delta w$$

on $\mathbb{R}^+ \times \Omega$. In the equation above, we denote by w the local chemical potential difference which is given as the variational derivative $\frac{\delta E}{\delta \rho}$ of E with respect to ρ (cf. Section 4). Thus, we obtain

$$w = -\gamma \Delta \rho + \Psi'(\rho).$$

The system has to be supplemented with boundary and initial conditions. Here, we request $\frac{\partial}{\partial \nu} w = \frac{\partial}{\partial \nu} \rho = 0$, where ν is the outer normal on $\partial\Omega$ and $\rho(0, \cdot) = \rho_0(\cdot)$ for some initial concentration distribution ρ_0. We remark that, with these boundary conditions, mass is conserved and that the Ginzburg-Landau free energy is a Lyapunov functional, i.e., we have

$$\frac{d}{dt} \int_\Omega \rho(x, t) dx = 0 \qquad \text{and} \qquad \frac{d}{dt} E(\rho(t)) \leq 0.$$

Starting with a random perturbation of a constant state $\bar{\rho}_0$, which has values in the unstable concave part of Ψ, we observe the following: In the beginning, the chemical energy decreases rapidly, whereas the gradient energy increases. This is due to the fact that, during phase separation, ρ attains values which are, at large portions of the domain, close to the minima of the chemical energy Ψ. Since regions of different phases are separated by transition zones with large gradients of ρ, the gradient energy increases during phase separation. In the second stage of the evolution—the actual clustering—when the structures become coarser, the total amount of transition zones decreases. Correspondingly, the amount of gradient energy becomes smaller again.

3 Including Elasticity

Elastic stresses play an important role during phase separation in most alloys. Such stresses arise from an elastic energy which takes into account elastic interactions due to different crystal structures. The free energy E can be defined as a function of the concentration ρ and the displacement field $\mathbf{u}$, as follows:

$$E(\rho, \mathbf{u}) := \int_\Omega \left\{ \psi(\rho) + \frac{\gamma}{2} |\nabla \rho|^2 + W(\rho, \mathbf{u}) \right\} dx.$$

The third term accounts for an energy contributions due to elasticity. We consider linear elasticity and

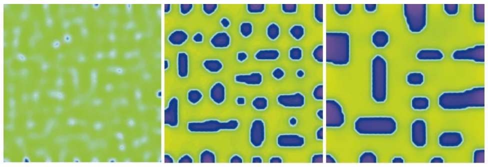


Fig. 4. Three timesteps of a Cahn-Hilliard process including anisotropic elasticity.

obtain $W(\rho,\mathbf{u}) := (\mathcal{E}(\mathbf{u}) - \bar{\mathcal{E}}(\rho)) : \mathcal{C}(\mathcal{E}(\mathbf{u}) - \bar{\mathcal{E}}(\rho))$, where $\mathcal{E}(\mathbf{u}) := \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^t\right)$. Here, $\mathcal{C}$ is the possibly anisotropic elasticity tensor and the term $\bar{\mathcal{E}}(\rho)$ is the stress free strain at a concentration ρ given in the simplest model by $\bar{\mathcal{E}}(\rho) = e\,\rho\,\mathrm{Id}$ with a material constant e. The product $A : B$ of two $d \times d$ matrices A, B is defined to be $\sum_{i,j=1}^{d} A_{ij}B_{ij}$.

Thus, the resulting diffusion equation introduced in Section 2 has to be modified. We obtain, for the chemical potential,

$$w = -\gamma\Delta\rho + \psi'(\rho) - \mathcal{S} : \bar{\mathcal{E}}'(\rho) \qquad \text{in} \quad \Omega_T, \tag{3.1}$$

where

$$\mathcal{S} = \mathcal{C}(\mathcal{E}(\mathbf{u}) - \bar{\mathcal{E}}(\rho))$$

is the stress tensor. Fig. 4 shows three timesteps from a diffusion process that takes anisotropic elasticity into account.

Since the relaxation into mechanical equilibrium occurs on a time scale that is fast compared to the time scale at which diffusion takes place, we assume quasistatic equilibrium for the deformation. Hence, we obtain the mechanical equilibrium condition $\operatorname{div}\mathcal{S} = 0$.

4 A Multiscale Clustering Approach

The aim of this section is to derive a continuous clustering model mainly on flow data. Motivated by the Cahn-Hilliard model for phase separation and particle coarsening (cf. Section 2), we introduce a cluster mapping, $u : \mathbb{R}_0^+ \times \Omega \to \mathbb{R}$, which will be the solution of an appropriate evolution problem. Time will thereby serve as the scale parameter leading from fine cluster granularity to successively coarser clusters. For fixed time t, our definition of the set of clusters $\mathcal{C}(t)$ is founded on the function u by

$$\mathcal{C}(t) = \{x \mid u(t,x) \geq 0\}.$$

This set splits up into the actual clusters

$$\mathcal{C}(t) = \bigcup_i \mathcal{C}_i(t),$$

where $\{\mathcal{C}_i(t)\}_i$ are the connected components of $\mathcal{C}(t)$.

Now, we study the evolution problem which controls the quantity u. We suppose this evolution to be a suitable clustering model, if, for the induced $\mathcal{C}(t)$:

- The number of clusters generically decreases in time,
- The shape of the cluster components strongly corresponds to correlations in the data field,
- The volume fraction covered by $\mathcal{C}(t)$ is approximately constant in t, i.e., $\frac{|\mathcal{C}(t)|}{|\Omega|} \approx \Theta$ for $\Theta \in (0,1)$.

We pick up the physical Cahn-Hilliard model and consider a double well separation potential $\Psi(u)$ and define a separation energy $E_s = \int_\Omega e_s(u)\mathrm{d}x$ with energy density $e_s(u) = \Psi(u)$. Under all u with $\int_\Omega u\mathrm{d}x = \bar{u}_0 = const.$, the energy E_s is minimal if u attains only the values $\pm\beta$.

This leads to a binary decomposition of the domain into two parts, where one part corresponds to $\{x \mid u(x) = \beta\}$.

The set $\{x \mid u(x) = \beta\}$, however, can have many connected components and may even be very unstructured. Furthermore, there is no mechanism which enforces a successive coarsening and, thus, a true multiscale of clusters. Therefore, we want to introduce a term penalizing the occurrence of many disconnected cluster components with high interfacial area. Motivated by the Cahn-Hilliard theory of phase transition, we choose a gradient energy $E_\partial = \int_\Omega e_\partial \mathrm{d}x$ with local energy density e_∂ that penalizes rapid spatial variations of u.

In order to have the flexibility to choose an anisotropic and inhomogeneous gradient energy, an appropriate definition of an interfacial energy density is given by

$$e_\partial(\nabla u) = \frac{\gamma}{2} A\nabla u \cdot \nabla u,$$

where "$\cdot$" denotes the scalar product in $\mathbb{R}^n$, γ is a scaling coefficient, and $A \in \mathbb{R}^{n\times n}$ is some symmetric positive definite matrix that may depend on the space variable and other quantities involved.

In the following, we will call the set $\partial\{x \mid u(x) = 0\}$ the interface. The orientation of the interface can be described by the normal to the interface, which, in the case that $\nabla u \neq 0$, is given by

$$\nu = \frac{\nabla u}{\|\nabla u\|}.$$

We remark that the interface between the set of positive and negative values of u is perpendicular to ν. For $A = \mathrm{Id}$, all gradients of u and, hence, all interfaces are penalized equally independent of their orientation. With respect to our clustering intention, we consider an anisotropic energy density which strongly depends on the orientation of the local interface and, thereby, on the direction of ∇u.

Let us assume $v : \Omega \to \mathbb{R}^n$ to be some vector field on the domain Ω. Typically, such a field induces a flow on Ω with streamlines which are the solution of the ordinary differential equation $\dot{x} = v(x)$. Now, a natural clustering should emphasize the coherence along the induced streamlines. Thus, cross streamline interfaces have to be penalized significantly by the gradient energy. We choose

$$A := B(v)^T \begin{pmatrix} 1 & 0 \\ 0 & \alpha(\|v\|)\mathrm{Id}_{n-1} \end{pmatrix} B(v),$$

where Id_{n-1} is the identity mapping in $\mathbb{R}^{n-1}$ and, for given $r \in \mathbb{R}^n$, the mapping $B(r) \in SO(n)$ is a coordinate rotation with $B(r)r = \|r\|e_1$. Since interfaces that cross streamlines have to have larger energy, we choose a positive α with $\alpha \leq 1$.

Now, we define the first variation of the energy

$$w = \frac{\delta E}{\delta u},$$

which is defined on Ω by

$$\int_\Omega \frac{\delta E}{\delta u}\zeta \mathrm{d}x := \frac{d}{d\epsilon}E(u+\epsilon\zeta)\Big|_{\epsilon=0}.$$

We obtain $w = \Psi'(u) - \gamma\,\mathrm{div}(A\nabla u)$.

Let us assume that the evolution of the cluster mapping u is governed by diffusion, where the corresponding flux linearly depends on the negative gradient of the first variation of energy. As the simplest model, we choose $\partial_t u - \Delta w = 0$, and end up with the following fourth order parabolic problem: *Find a continuous cluster mapping* $u : \mathbb{R}_0^+ \times \Omega \to \mathbb{R}$ *such that*

$$\partial_t u - \Delta w = 0 \tag{4.1}$$

$$w = \Psi'(u) - \gamma\,\mathrm{div}(A\nabla u) \tag{4.2}$$

with boundary conditions $\frac{\partial}{\partial\nu}u = \frac{\partial}{\partial\nu}w = 0$ *and prescribed initial data* $u(0,\cdot) = u_0(\cdot)$.

This modified Cahn-Hilliard equation can be interpreted as the H^{-1} gradient flow for the energy E (see [15] for a discussion of this fact in the case of the standard Cahn-Hilliard equation). In particular, we immediately obtain the Lyapunov property

$$\partial_t E(u) \leq 0. \tag{4.3}$$

This energy decay is in fundamental accordance to the desired successive pattern coarsening in the evolution. After an initial short period of phase separation, it is mainly the interfacial energy contribution which is successively reduced. Furthermore, as in the case of the standard Cahn-Hilliard equation, we obtain that $\int_\Omega u(x,t)dx$ is constant in time, which corresponds to the approximate volume conservation of the generated scale of cluster sets.

In general, it does not make sense to consider certain initial data if no a priori information on the clustering is known. As initial data u_0, we thus choose a constant value $\bar{u}_0$ plus some small random noise. The constant $\bar{u}_0$ depends on the volume fraction Θ of the domain later on being covered by the clusters, i.e., by the sets $\{x \mid u(t,x) \geq 0\}$. Therefore, we choose

$$\bar{u}_0 = \Theta\beta - (1-\Theta)\beta.$$

Starting with a random perturbation of this constant first, cluster patterns will grow very rapidly without any prescribed location and orientation. This is in order to decrease $E_s = \int_\Omega \Psi(u)$, which forces the solution to obtain values close to $\pm\beta$ in most of the domain Ω. After this process, the clusters orient themselves in an anisotropic way to decrease the amount of the anisotropic gradient energy E_∂. In addition, the cluster becomes coarser and coarser due to the fact that smaller particles shrink and larger ones grow. We remark that, in particular, one observes that a large particle that is surrounded by smaller ones grows at the expense of the smaller ones. This implies that, as time evolves, locally, only the main features of the clusters will be kept.

Altogether, we obtain a scale $u(t,\cdot)$ of cluster mappings and induced cluster sets $\mathcal{C}(t)$. They represent a successively coarser representation of simulation data and continuously enhance coherences in the underlying simulation data set, where the cluster set $\mathcal{C}(t)$ will cover a volume of approximate size $\Theta|\Omega|$. As already mentioned, the multiscale property comes along with the decay of the energy $E(u)$ (see (4.3)).

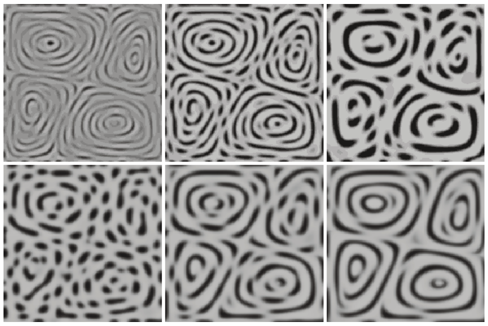


Fig. 5. Continuous clustering of a vector field: time evolution (upper row), effect of increasing anisotropy (lower row). The computation is based on a grid of resolution 257^2.

To summarize, the vector field that is to be represented defines the anisotropy of the energy and therefore governs the diffusion process of u. Roughly speaking, the vector field determines in which direction an interface between phases is relatively "cheap" (from an energy point of view). As the energy is minimized during the evolution, the interfaces will move in such a way that there are mostly "cheap" interfaces (i.e., interfaces where ∇u is roughly perpendicular to v, which means that the interface is roughly parallel to v). So, for any particle, most of its boundary will be aligned with the vector field v. If not too large, the particles themselves will be aligned to the vector field. Fig. 5 shows the result of this process and the influence of increasing anisotropy in the surface energy term.

5 Control of Particle Shape

The results presented in Fig. 5 are quite similar to the various flow texture methods known in the literature [5], [2], [29], [7], [21]. Given a good scale choice, such images could be used on their own to give insight in a 2D flow field. However, such images convey only the orientation, but not also the direction, of a flow field. This limitation has been recognized by Wegenkittl et al. [26], who have presented several methods to emphasize direction in LIC algorithms. One method to convey directional information via iconification is further presented in Section 8. In this section, we discuss an alternative method that adds directional information directly into the continuous clustering model.

Our aim is to create interfaces that are asymmetric with respect to the flow field's direction. We divide such an interface into two regions, depending on the angle between the interface's surface normal and the flow field, i.e., the quantity $\nabla u \cdot \frac{v}{|v|}$ (see also Fig. 7). In the *front half*, defined by $\nabla u \cdot \frac{v}{|v|} < 0$, the energy is as described in the previous section. In the *back half*, defined by $\nabla u \cdot \frac{v}{|v|} > 0$, we define an additional shape energy term, E_s, as a function of the angle between the local gradient ∇u and the flow field v:

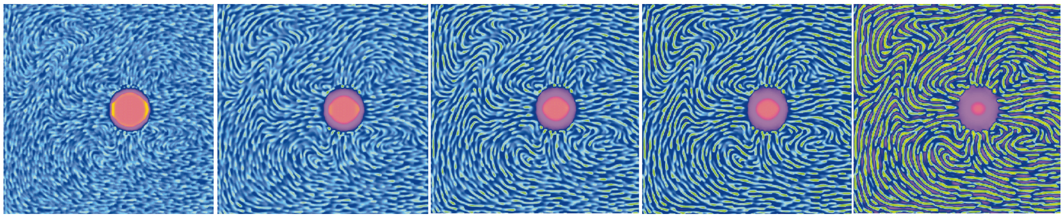


Fig. 6. Vector field visualization with control on particle shape (different time scales).

$$E_s(u) = \int_\Omega q\frac{\gamma}{2}\max\left(\nabla u \cdot \frac{v}{|v|}, 0\right)^2.$$

The coefficient q defines the importance of this additional shape energy E_s in the global energy, thus the intensity of this effect in the final visualization. The shape energy thus penalizes only the back halves of the clusters, decreasing the value of u in these areas. Indeed, the clusters try to avoid back sides whose normals point in the upwind direction. If the function u is directly visualized by a color plot, the perceived effect suggests drops of fluid transported by the underlying vector field v. Fig. 8 shows this effect for a single, respectively, several particles, in a circular vector field.

Fig. 6 shows several time steps in the visualization of a more complex flow field. Fig. 9 illustrates the effect of the additional energy term E_s on the particle shape by showing a closeup for a timestep of the above sequence. The above images are similar to the fur-like textures for flow visualization presented in [16]: The flow field's direction is suggested by the fading away of the "tail" of the particles. However, whereas the cited method generates sharp, arrow-shaped particles that point in the direction of the flow field, we generate blunt, drop-like particles that point in the opposite direction.

Corresponding to the above additional energy term, the first variation of the energy w—which we have to build into our diffusion problem—is now defined by:

$$w = \Psi'(u) - \gamma\,\mathrm{div}(A\nabla u) - \gamma\mathrm{div}\tilde{v},$$

where

$$\tilde{v} := \begin{cases} q\left((\nabla u \cdot v)\frac{v}{|v|}\right), & \text{if } \nabla u \cdot v > 0, \\ 0, & \text{if } \nabla u \cdot v \le 0. \end{cases}$$

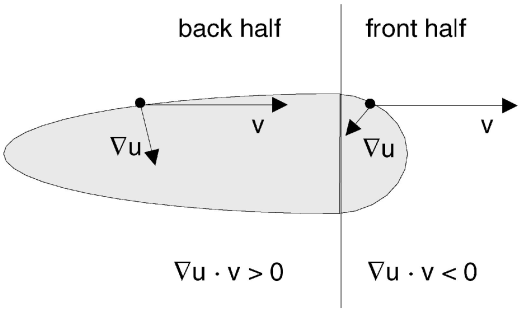


Fig. 7. Control of particle shape.

6 Stress-Driven Diffusion

Now, let us discuss the extension of our clustering model based on elastic stresses. We choose an anisotropic elasticity which is strongly correlated with the flow field. Hence, we define a modified strain

$$\mathcal{E}(\tilde{\mathbf{u}}) := B^T(v)\mathcal{E}(\tilde{\mathbf{u}})B(v),$$

a correspondingly modified stress:

$$\mathcal{S} = \mathcal{C}B^T(v)(\mathcal{E}(\mathbf{u}) - \bar{\mathcal{E}}(u))B(v),$$

and the appropriate modified elastic energy:

$$W(u, \mathbf{u}) = \left(B^T(v)\big(\mathcal{E}(\mathbf{u}) - \bar{\mathcal{E}}(u)\big)B(v)\right) : \mathcal{C}\left(B^T(v)\big(\mathcal{E}(\mathbf{u}) - \bar{\mathcal{E}}(u)\big)B(v)\right).$$

Again, the rotation $B(v)$ is used to transform the flow aligned coordinate system to the canonical coordinate frame. With the modifications presented above, the elasticity tensor is defined in such a way that the flow direction is the preferred stretching direction.

Incorporating anisotropic elasticity leads to interesting images already in the early clustering stages. These images are related to the shear zones of the flow field. For later times, the resulting clustering tries to avoid the crossing of high shear regions. Fig. 10 illustrates the above effects for the vector field discussed in Section 5.

It is sometimes convenient to choose u-dependent elasticity tensors $\mathcal{C}(u)$ in such a way that the particles have larger elasticity constants, i.e., the particles are harder than the surrounding matrix. In this setting, already in the case of an isotropic elasticity independent of a vector field, we recognize significant chances in the behavior of our method: The cluster distribution tends to be more uniform. Furthermore, independent of the initial volume fraction, it is always the harder phase which forms the clusters,

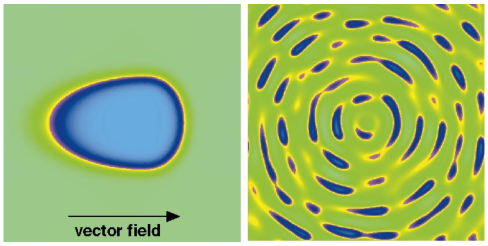


Fig. 8. Constant vector field, single particle (left). Circular vector field (right).

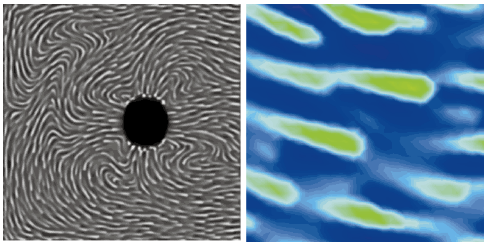


Fig. 9. Control of particle shape. Overview image (left) and close-up image (right).

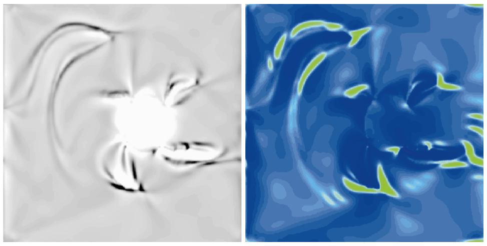


Fig. 10. Effect of anisotropic elasticity. Early timestep (left); late timestep (right).

whereas the softer phase builds the surrounding matrix. In conclusion, elasticity can be used for two goals. First, it can produce cluster-based visualizations of shear layer type data. Second, it can be used as a global control for the cluster size distribution.

7 Discretization of the Diffusion Problem

In what follows, we briefly discuss the discretization and implementation of the evolution problem for the cluster mapping u and the set of clusters $\mathcal{C}(t)$. For this purpose, a finite element discretization in space and some discrete scheme in time are considered. Here, uppercase letters denote discrete quantities which correspond to continuous quantities in lowercase letters. Hence, we consider an appropriate continuous variational formulation for (4.1), (4.2), given by

$$(\partial_t u, \theta) + (\nabla w, \nabla \theta) = 0,$$
$$(w, \chi) = (\Psi'(u), \xi) + (A\nabla u, \nabla \xi),$$

which shall hold for all $\theta, \xi \in C^\infty(\bar{\Omega})$, where $(\cdot, \cdot)$ denotes the L^2 product on the domain Ω. For a finite element implementation, we now replace the continuous solution and test functions in this formulation by discrete approximations in some finite element space. Here, we have restricted ourselves to finite elements on regular adaptive grids $\mathcal{M}_h$ in 2D and 3D generated by recursive subdivision of elements E. On these grids, we consider the bilinear, respectively, trilinear, finite element spaces V^h for the approximation of u and w on Ω. Numerical integration of the L^2 products is based on the lumped masses product $(\cdot, \cdot)_h$ [24]. Furthermore, we consider a center of mass quadrature rule for the bilinear forms $(\nabla \cdot, \nabla \cdot)$ and $(A\nabla \cdot, \nabla \cdot)$. Especially, we replace A by the piecewise constant diffusion tensor A^h, with $A^h|_E = A(c_E)$, where c_E is the element's center of mass.

For the discretization in time, we have taken into account two possibilities: a first order implicit Euler scheme and a second order θ-splitting scheme (see Bristeau et al. [1] and Müller-Urbaniak [14]). Both are known to be strongly A-stable. While we can prove the energy decay property (4.3) for the implicit Euler scheme, we use the θ-splitting for practical computations as it allows larger time steps.

In the case of the implicit Euler scheme, the time derivative is discretized by $\partial_t u((n+1)\tau) \approx \frac{U^{n+1} - U^n}{\tau}$, where τ is the selected time step and U^n an approximation of $u(n\tau)$. A brief introduction to the more complicated θ-splitting can be found in the Appendix.

Finally, we can derive a fully discrete scheme. For the "hat-shaped" multilinear basis functions Φ_i and the discrete piecewise constant anisotropic diffusion matrix A^h, we define, by

$$M_h := \big((\Phi_i, \Phi_j)_h\big)_{ij},$$
$$L_h(A) := \big((A^h \nabla \Phi_i, \nabla \Phi_j)\big)_{ij},$$

the diagonal lumped mass and the anisotropic stiffness matrix, respectively, and, by $L_h := L_h(\mathrm{Id})$, the standard stiffness matrix. These global matrices M_h, L_h, and $L_h(A)$ are assembled in a grid traversal collecting the contributions on all local grid elements as is standard in finite element programming [4].

If we indicate by a bar coefficient vectors corresponding to finite element functions in the basis $\{\Phi_i\}_i$, we obtain the backward Euler discretization

$$M_h \bar{U}^{n+1} + \tau L_h(\bar{\Psi}'(\bar{U}^{n+1}) +$$
$$\gamma M_h^{-1} L_h(A) \bar{U}^{n+1}) = M_h \bar{U}^n$$

with $U^0 = \mathcal{I}_h u_0$, where $\mathcal{I}_h$ is the interpolation on grid $\mathcal{M}_h$. By obvious notation, $\bar{\Psi}'(\cdot) := (\Psi'(\cdot))_i$ is the vector of nodal wise derivatives of Ψ.

In each step of the discrete evolution, we have to solve this system of nonlinear equations. In order to do this, we apply some Newton scheme which typically converges in a few steps if we consider moderate time steps and pick up the old solution at the old time step as the initial guess for the Newton iteration.

The efficiency of our approach is further increased by an adaptive grid refinement and coarsening strategy. Here, we used a heuristic strategy which refines in interfacial regions and coarsens in the pure phases.

In the case of the implicit Euler scheme, it is possible to prove discrete counterparts of the mass conservation and energy decay properties, i.e.,

$$\int_\Omega U^n \mathrm{d}x = \int_\Omega \mathcal{I}_h u_0 \mathrm{d}x$$

and

$$E_h(U^n) := \int_\Omega \{\mathcal{I}_h \Psi(U^n) + A^h \nabla U^n \cdot \nabla U^n\} \mathrm{d}x$$

is nonincreasing (discrete Lyapunov property) and, thus, gives reason for the discrete multiscale property of our method

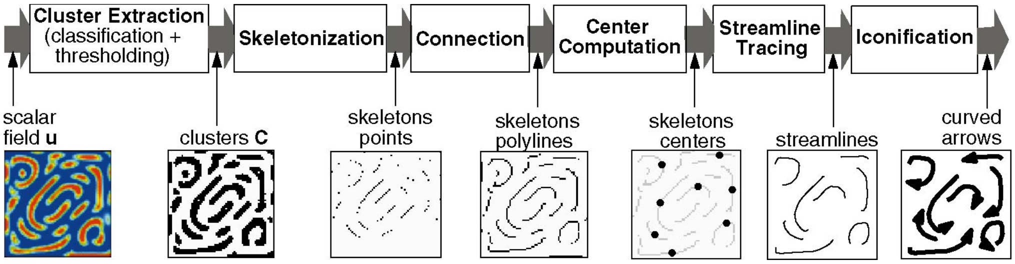


Fig. 11. Pipeline for iconic cluster visualization.

$$E_h(U^{n+1}) \leq E_h(U^n) \leq \cdots \leq E_h(U^1) \leq E_h(\mathcal{I}_h u_0).$$

Considering, in addition, either the directional term which allows control of the particle shape or the elastic stress term in the discrete potential, we proceed analogously to the basic model, except in the elasticity case, where we have to couple the diffusion equation with the balance law for the elasticity. For details, we refer to [8].

8 Iconic Representation of the Clusters

The clustering method described in Section 4 produces clusters which emphasize the spatial coherence in the data. In what follows concerning the iconic representation, we focus on the case of flow data. Nevertheless, this exposition might inspire the reader to think of different applications along the same guidelines. For flow data, cluster interfaces tend to be tangent to the streamlines of the underlying vector field, so the clusters' shapes convey local insight in the vector field direction. On the other hand, the physical phase separation model presented in Section 2 produces clusters which tend to be evenly distributed over the domain of interest, Ω.

Consequently, such clusters could be used as a starting point for producing a simplified visualization of the structure of the underlying vector field. For this, we propose reducing each cluster to one curved arrow icon. For every cluster, the size and spatial position of the icon should reflect the size of the cluster, whereas the curvature and arrow direction should be related to the vector field inside the respective cluster. We have chosen to use the curved arrow icons as they convey several information levels in a simple, easy to understand manner [23] as compared to other, more abstract icons.

The iconic visualization pipeline based on the multiscale clustering proceeds as follows (see also Figs. 11 and 13). First, the clusters $\mathcal{C}$ are extracted from the Cahn-Hilliard equation solution u. Next, the skeletons of the clusters are computed as sets of discrete points, as shown further in Section 8.2. Next, the center points of the skeletons are detected and used to construct the curved arrows by streamline tracing, as discussed in Section 8.4. The reason why we use such an apparently complicated method is that, typically, the clusters are large, thin, often curved, structures and streamline tracing is rather sensitive with respect to the choice of the starting point. The rest of this section explains the several steps in detail.

8.1 Cluster Extraction

First, we extract the clusters $\mathcal{C}$ from the scalar field u. For this, we classify all the cells of the discretization of the field u as cluster outside, border, or inside cells, based on the sign of $u(x)$. Moreover, all border and inside cells belonging to a given cluster are labeled by the cluster's ID as presented by, e.g., Post et al. in [18].

8.2 Skeletonization

In the second step, clusters are reduced to their *skeletons*. By skeleton, we understand here a set of points which, if connected, produce a "spine" which conveys the shape information of the original cluster in a compact manner.

There are numerous skeletonization algorithms [11], [13]. However, many such algorithms produce skeletons with complex, tree-like topologies. As we intend here to use the skeletons only to produce the arrow icons, we prefer simple, polyline-like topologies.

To produce such skeletons, we use a discrete method based on the eikonal equation [22]. Given a boundary curve Γ in two dimensions (or a boundary surface in 3D) and a function T such that $T = 0$ on Γ, the eikonal equation is $|\nabla T| = 1$. If we regard Γ as being the level set (e.g., isoline or isosurface) of the function T, the above equation describes the evolution in time of Γ in normal direction to Γ, with constant speed equal to 1. In our case, Γ coincides with the previously detected cluster boundaries.

As presented by Sethian [22], we discretize the above equation on the same grid used to solve the Cahn-Hilliard equation, as follows (for the 2D case):

$$\begin{aligned} &max(D_{ij}^{-x}T, 0)^2 + min(D_{ij}^{+x}T, 0)^2 \\ &+ max(D_{ij}^{-y}T, 0)^2 + min(D_{ij}^{+y}T, 0)^2 = F_{ij}^{-2}, \end{aligned} \tag{8.1}$$

where the ij denotes the current grid point and the operators D^+ and D^- denote the forward and backward differences at that grid point. On a 2D regular grid of cell size h, we have $D_{ij}^{+x}T = \frac{T_{i+1,j} - T_{ij}}{h}$ and $D_{ij}^{-x}T = \frac{T_{ij} - T_{i-1,j}}{h}$ and similarly for the y axis.

Equation (8.1) can be iteratively solved for every grid point until the solution T converges. However, we use the more efficient fast marching method, as described in [22]. The fast marching method proceeds by first tagging all grid points as either *Known* (for the points on Γ with known value $T = 0$), *Trial* (all points that are one grid point away from Γ), and *Far*, for all other points.

The algorithm (Fig. 12) constructs the solution T iteratively from the initially *Known* points with $T = 0$ on the boundary Γ. At each step, the solution T is constructed

```
flag all points as Known, Trial or Far ;

for all Trial points P
  insert P in narrowband;

while narrowband not empty
{
   A := point with smallest T in narrowband;

   remove A from narrowband;

   flag A as Known;

   for all neighbors N of A
     if N is Far
     {
        add N to narrowband;
        flag N as Trial;
     }

  if A has no Trial neighbors
     flag A as Extremum;
  else
     for all Trial neighbors N of A
        recompute T at N by solving the eikonal equation;
}
```

Fig. 12. Fast marching method pseudocode.

from the point with the smallest computed T value by stepping away from the boundary in a downwind direction. When a local extremum point of T is encountered, we freeze that point and add it to the skeleton. The boundary is thus marched inward until it collapses into a single line, namely the skeleton points of the cluster Γ.

To implement the algorithm efficiently, we use a heap structure *narrowband* to maintain the set of *Trial* points. The heap is maintained sorted in ascending order on the value of T. Finding the *Trial* point with the smallest T value in *narrowband* is thus $O(1)$. Inserting a new point in the heap is $O(logM)$ in the worst case for a heap of M points. Overall, the fast marching method is $O(NlogN)$ in the worst case for a grid of N points [22]. Practically, our implementation of the above algorithm completes in a few seconds on grids of around 100,000 cells on an SGI O2 R5500 machine.

8.3 Reconnection and Center Detection

The skeletonization produces a set of usually disjoint skeleton points (Fig. 13). The reason for this is that the inward marching of the boundary described in the previous section is accurate only up to the size of a grid cell. However, the desired skeleton should be exactly one grid cell thick. By looking at the extracted skeletons, we estimated empirically that about 10 percent of the skeleton points are not extracted by the fast marching method. To remediate this problem, we reconnect the extracted disjoint points in a postprocessing step based on a closest point strategy in order to produce a polyline. For every cluster, we then compute the center of its polyline and use it in the next step of the pipeline.

8.4 Icon Construction

From the skeleton centers detected in the previous step, streamlines are traced in the vector field until they reach the borders of the clusters within which they evolve. Next, curved arrow geometries are constructed around the extracted streamlines. Finally, we discuss the application of the continuous clustering method and the associated curved arrow visualization to various vector fields.

The leftmost image in Fig. 13 shows a solution $u(t, \cdot)$ of the modified Cahn-Hilliard process driven by a 3-vortex vector field on a 64×64 2D grid.

The thresholding of the continuous signal u into clusters is shown in the second image of Fig. 13. The clusters overlaid with the extracted skeleton points are shown in the third image of Fig. 13. The rightmost image in Fig. 13 visualizes the vector field with streamline-based icons. The curved arrows, initiated from the skeleton centers depicted as small balls, are clipped by the borders of the clusters into which they evolve.

A similar visualization is shown, for two different clustering time instants, in Fig. 14. The multiscale feature of the clustering is visible in the reduction of the arrow count. An enhancement of the proposed curved arrow visualization is shown in Fig. 15 by the addition of a spot noise textured background. Finally, Fig. 16 shows the proposed method applied on a circular 2D vortex.

9 Discussion

In this section, we compare the presented continuous clustering method with the discrete clustering method presented in [23]. Similarly to the method presented here, discrete clustering builds a vector field multiscale representation by merging neighboring cells with similar vector values. The time parameter of the Cahn-Hilliard equation is equivalent to the iteration count in the bottom-up discrete cluster merging. The continuous clustering method delivers a continuous scale of successively coarser cluster sets. In contrast, discrete clustering proceeds in distinct steps, where two clusters are merged at each step.

Fig. 18 shows the discrete clustering of the two vector fields discussed in the previous section. Regarding the cluster shapes, the continuous clustering explicitly constrains the shape via the minimization of the interfacial

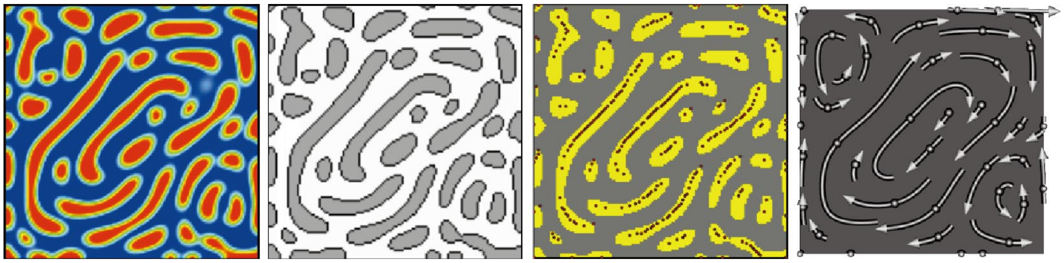


Fig. 13. Clustering pipeline, from left to right: diffusion solution, clusters, skeleton points, curved arrow visualization. The underlying grid is of size 257^2.

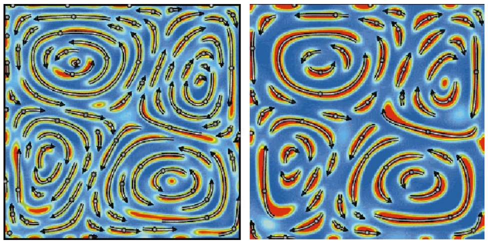


Fig. 14. From left to right: the results of two successive time steps of the clustering evolution with corresponding icons.

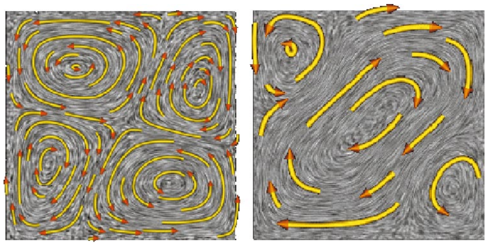


Fig. 15. Two different convective vector fields are depicted by the clustering method. On the left, a fine representation is shown, whereas, on the right, a resulting coarse representation is depicted.

energy in order to obtain vector-aligned, smooth-shaped clusters. In contrast, the discrete clustering does not constrain the cluster shapes in any manner, assuming that their growth to a "natural" partition of the vector field can be governed only by the intercluster similarity function. This can lead however to "badly" shaped (e.g., thin and long) clusters, which are hard to represent by curved arrow icons. In this respect, we see the controlling of the cluster shape in the continuous clustering method as an advantage. However, discrete clustering always merges the two most resembling clusters, so the intrinsic symmetry of the underlying vector field remains visible in the clustering (see [23] for details). This may be seen as an advantage of the discrete clustering method, see Fig. 18. Finally, the shapes produced by the continuous clustering are not constrained to simple convex ones, as in the method presented by Heckel et al. [9]. We have applied the continuous clustering method also to the visualization of 3D fields. Fig. 17 shows the visualization of a 3D circular vortex field from two different viewpoints. The produced arrow icons illustrate the clustering of the data in the center of the domain, where the flow is dominated by a vertical swirling motion, and along the domain's boundary, where the flow mainly rotates in horizontal planes.

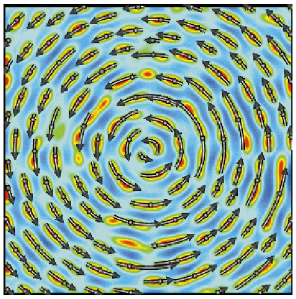


Fig. 16. Visualization of a circular flow field by the clustering method.

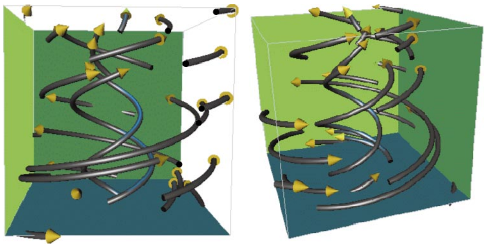


Fig. 17. A 3D vectorfield is visualized by the clustering method.

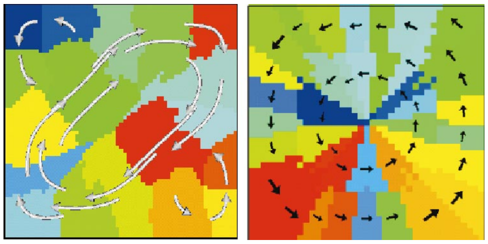


Fig. 18. Discrete clustering of a 3-vortex field (left) and a circular vortex field (right).

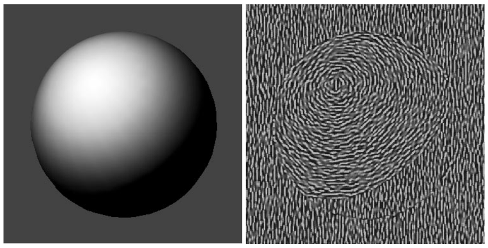


Fig. 19. Simple ball image (left) and brush-stroke representation (right).

Finally we present an application of our approach for 2D image processing, where we generate a scale of brush stroke type representations of a greyscale image. We consider the intensity of an image as a scalar function $s : \Omega \rightarrow \mathbb{R}$. If we intend to release brush strokes along regions of homogeneous values of the scalar quantity s, we need to energetically favor interfaces which have a tangent space locally perpendicular to ∇s. Hence, we choose a corresponding quadratic form with

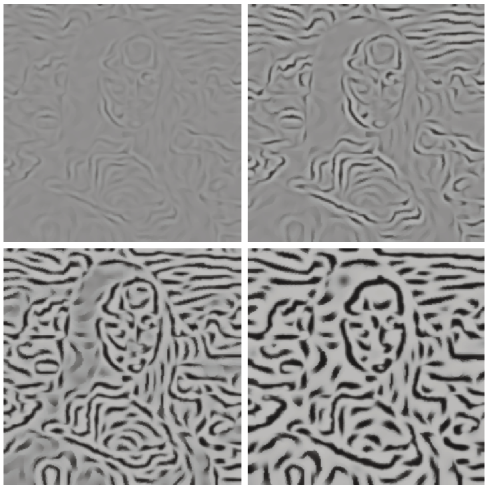


Fig. 20. Multiscale brush-stroke representation of a grayscale Mona Lisa image.

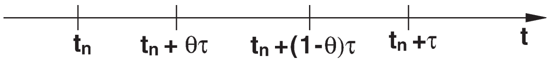


Fig. 21. θ-splitting, subdivision of time steps.

$$A := B(\nabla s)^T \begin{pmatrix} \alpha & 0 \\ 0 & \mathrm{Id}_{n-1} \end{pmatrix} B(\nabla s),$$

where, for given $r \in \mathbb{R}^n$, the mapping $B(r) \in SO(n)$ is again a coordinate rotation with $B(r)r = \|r\| e_1$ and $1 > \alpha 0$. Fig. 19 shows the application of out method on a simple ball image in which the image gradient varies smoothly. However, when the image is more complex, it is harder to recognize the original image features in the processed images (cf. Fig. 20).

10 Conclusions

We have presented a new multiscale clustering approach which is based on a continuous model for clustering on scientific data. The approach is motivated by well-known physical clustering models describing the phase separation and coarsening process in metal alloys. As a case study, we focused on the clustering of flow fields. Future research could be directed to:

- the improvement of the performance of the method with respect to computing time, where parallelization or implementation of the underlying diffusion in texture hardware may help to overcome the computational bottleneck (the clustering process takes—depending on the size and resolution of the data—from several minutes up to a couple of hours on a fast workstation),
- the construction of further appropriate interfacial energies for different applications,
- a detailed classification of the skeleton shapes and the selection of appropriate icons, e.g., for saddle points or vortices in flow fields.

Appendix

Here, we briefly outline the implementation of the θ-splitting scheme. Due to its strong stability properties, it allows a much larger timesteps scheme and we have used it in the current implementation of our clustering model. The scheme divides any time step in three substeps (see Fig. 21). In each substep, the linear operator is split up into two parts, with coefficients α and $1-\alpha$, respectively, one of which is taken implicitly, the other explicitly. The nonlinear term is taken implicitly in the middle substep only.

For the parameter $\alpha \in (0.5; 1]$, $\beta = 1 - \alpha$, and $\theta \in (0; 0.5)$, the scheme reads as follows:

$$\begin{aligned}
&\Big(M_h + \alpha\theta\tau L_h M_h^{-1} L_h(A)\Big)\bar{U}^{n+\theta} \\
&= \Big(M_h - \beta\theta\tau L_h M_h^{-1} L_h(A)\Big)\bar{U}^{n} - \theta\tau L_h \psi'\big(\bar{U}^{n}\big), \\
&\Big(M_h + \beta(1-2\theta)\tau L_h M_h^{-1} L_h(A)\Big)\bar{U}^{n+1-\theta} \\
&\quad + (1-2\theta)\tau L_h \psi'\big(\bar{U}^{n+1-\theta}\big) \\
&= \Big(M_h - \alpha(1-2\theta)\tau L_h M_h^{-1} L_h(A)\Big)\bar{U}^{n+\theta}, \\
&\Big(M_h + \alpha\theta\tau L_h M_h^{-1} L_h(A)\Big)\bar{U}^{n+1} \\
&= \Big(M_h - \beta\theta\tau L_h M_h^{-1} L_h(A)\Big)\bar{U}^{n+1-\theta} \\
&\quad - \theta\tau L_h \psi'\big(\bar{U}^{n+1-\theta}\big).
\end{aligned}$$

References

[1] M.O. Bristeau, R. Glowinski, and J. Periaux, "Numerical Methods for the Navier-Stokes Equations: Applications to the Simulation of Compressible and Incompressible Viscous Flows," *Computer Physics Report,* Research Report UH/MD-4, Univ. of Houston, 1987.

[2] B. Cabral and L. Leedom, "Imaging Vector Fields Using Line Integral Convolution," *Computer Graphics (Proc. SIGGRAPH '93),* J.T. Kajiya, ed., vol. 27, pp. 263-272, Aug. 1993.

[3] J. Cahn and J. Hilliard, "Free Energy of a Non-Uniform System I. Interfacial Free Energy," *J. Chemistry and Physics,* vol. 28, pp. 258-267, 1958.

[4] P. Ciarlet and J. Lions, *Handbook of Numerical Analysis. Vol. V: Techniques of Scientific Computing,* Elsevier, 1997.

[5] W. de Leeuw and J.J. van Wijk, "Enhanced Spot Noise for Vector Field Visualization," *Proc. IEEE Visualization '95,* pp. 233-239, 1995.

[6] C.M. Elliott, "The Cahn-Hilliard Model for the Kinetics of Phase Separation," *Numerische Mathemtica,* pp. 35-73, 1988.

[7] L. Forssell, "Visualizing Flow over Curvilinear Grid Surfaces Using Line Integral Convolution," *Proc. IEEE Visualization '94,* pp. 240-246, 1994.

[8] H. Garcke, M. Rumpf, and U. Weikard, "The Cahn-Hilliard Equation with Elasticity, Finite Element Approximation and Qualitative Analysis," *J. Interphases and Free Boundaries,* vol. 3, pp. 101-118, 2001.

[9] B. Heckel, G. Weber, B. Hamann, and K.I. Joy, "Construction of Vector Field Hierarchies," *Proc. IEEE Visualization '99,* pp. 19-25, 1999.

[10] B. Jobard and W. Lefer, "Creating Evenly-Spaced Streamlines of Arbitrary Density," *Proc. Visualization in Scientific Computing '97,* W. Lefer and M. Grave, eds., pp. 43-54, 1997.

[11] F. Leymarie and M.D. Levine, "Simulating the Grassfire Transform Using an Active Contour Model," *IEEE Trans. Pattern Analysis and Machine Intellingence,* vol. 14, no. 1, pp. 56-75, Jan. 1992.

[12] S. Mallat, "A Theory for the Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 676-693, July 1989.

[13] N. Mayya and V.T. Rajan, "Voronoi Diagrams of Polygons: A Framework for Shape Representation," *J. Math. Imaging and Vision,* vol. 4, pp. 355-378, 1996.

[14] S. Müller-Urbaniak, "Eine Analyse des Zweischritt-θ-Verfahrens zur Lösung der instationären Navier-Stokes-Gleichungen," Preprint des SFB 359, 94-01, 1994.

[15] A. Novick-Cohen, "The Cahn-Hilliard Equation: Mathematical and Modelling Perspectives," *Advances in Math. and Science Applications,* vol. 8, pp. 965-985, 1998.

[16] L. Khouas, C. Odet, and D. Friboulet, "2D Vector Field Visualization Using Furlike Texture," *Proc. IEEE Visualization Symp. '99,* pp. 35-44, 1999.

[17] P. Perona and J. Malik, "Scale Space and Edge Detection Using Anisotropic Diffusion," *Proc. IEEE CS Workshop Computer Vision,* 1987.

[18] F.J. Post, T. v. Walsum, F.H. Post, and D. Silver, "Iconic Techniques for Feature Visualization," *Trans. Visualization and Computer Graphics,* vol. 1, pp. 288-295, 1995.

[19] T. Preußer and M. Rumpf, "An Adaptive Finite Rlement Method for Large Scale Image Processing," *J. Visual Comm. and Image Representation,* vol. 11, pp. 183-195, 2000.

[20] T. Preußer and M. Rumpf, "Anisotropic Nonlinear Diffusion in Flow Visualization," *Proc. IEEE Visualization '99,* pp. 325-332, 1999.

[21] C. Rezk-Salama, P. Hastreiter, C. Teitzel, and T. Ertl, "Interactive Exploration of Volume Line Integral Convolution Based on 3D-Texture Mapping," *Proc. IEEE Visualization '99,* pp. 233-240, 1999.

[22] J. Sethian, *Level Set Methods and Fast Marching Methods.* Cambridge Univ. Press, 1999.

[23] A.C. Telea and J.J. van Wijk, "Simplified Representation of Vector Fields," *Proc. IEEE Visualization '99,* pp. 35-42, 1999.

[24] V. Thomée, *Galerkin—Finite Element Methods for Parabolic Problems.* Springer, 1984.

[25] G. Turk and D. Banks, "Image-Guided Streamline Placement," *Computer Graphics (Proc. SIGGRAPH '93),* 1996.

[26] R. Wegenkittl, E. Gröller, and W. Purgathofer, "Animating Flowfields: Rendering of Oriented Line Integral Convolution," *Proc. IEEE Visualization '97,* pp. 119-125, 1997.

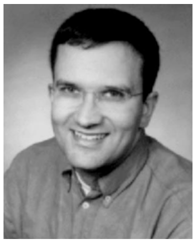
[27] J. Weickert, *Anisotropic Diffusion in Image Processing.* Teubner, 1998.

[28] P. Wong and D. Bergeron, "Hierarchical Representation of Very Large Data Sets for Visualization Using Wavelets," *Scientific Visualization,* G. Nielson, H. Hagen, and H. Mueller, eds., pp. 415-429, 1997.

[29] M. Zöckler, D. Stalling, and H.-C. Hege, "Interactive Visualization of 3D-Vector Fields Using Illuminated Streamlines," *Proc. IEEE Visualization '96,* pp. 107-113, 1996.

Harald Garcke received the Diploma degree (1989) and the PhD degree (1993) in mathematics from the University of Bonn, Germany. After completing his PhD, he was a research fellow at the University of Sussex (Brighton, United Kingdom) with a fellowship from the European Science Foundation (ESF). Since 1994, he has held a position at the University of Bonn, where he finished his habilitation thesis in 2000. His research interests include the mathematical modeling of phase transitions, nonlinear partial differential equations, free boundary problems, numerical analysis, and visualization.

Tobias Preußer studied mathematics at the University of Bonn, Germany and at the Courant Institute of New York University. He received the Diploma degree in 1999. Currently, he is working on his PhD thesis. His research interests are anisotropic diffusion methods in vector field feature visualization and geometric image processing.



Martin Rumpf received the Diplom degree and the PhD degree in mathematics from Bonn University in 1989 and 1992, respectively, and did his postdoctoral research at Freiburg University. He is a professor of applied mathematics at Duisburg University, Germany. Between 1996 and 2001, he has been professor at Bonn University. Since 2001, he has held the chair for Numerical Mathematics and Scientific Computing at Duisburg. His research interests are numerical methods for nonlinear partial differential equations, adaptive finite element methods, image and surface processing. Furthermore, he is concerned with new flow visualization techniques and efficient multiscale methods in scientific visualization.

Alexandru C. Telea received the MS degree in electrical engineering and computer science from the Polytechnical University of Bucharest, Romania, in 1997. He received the PhD degree in computer science from the Eindhoven University of Technology, The Netherlands, in October 2000. Since then, he has been an assistant professor in the Computer Graphics Group at the Eindhoven University of Technology. His research interests include interactive visualization and simulation systems, component-based and object-oriented software architectures, flow visualization, and software visualization.

Ulrich Weikard received the Abitur in 1991 from the Freiherr-vom-Stein Gymnasium in Leverkusen. He received the Diploma in mathematics in 1999 in Bonn. Since then, he has been working on his PhD in Bonn and Duisburg in the field of numerics of partial differential equations, especially on the Cahn-Hilliard equation with elasticity. His further research interests regard image processing and vector field visualization.

Jarke J. van Wijk received the MSc degree in industrial design engineering in 1982 and the PhD degree in computer science in 1986 from Delft University of Technology, both with honors. He is an associate professor of computer graphics at the Eindhoven University of Technology. He has (co)authored 40 publications on visualization. His main research interests are information visualization and flow visualization, both with a focus on the development of new visual representations.

▷ **For further information on this or any computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.