

# A SOM-based Chan-Vese model for unsupervised image segmentation

Mohammed M. Abdelsamea · Giorgio Gnecco · Mohamed  
Medhat Gaber

Received: date

**Abstract** Active Contour Models (*ACMs*) constitute an efficient energy-based image segmentation framework. They usually deal with the segmentation problem as an optimization problem, formulated in terms of a suitable functional, constructed in such a way that its minimum is achieved in correspondence with a contour that is a close approximation of the actual object boundary. However, for existing *ACMs*, handling images that contain objects characterized by many different intensities still represents a challenge. In this paper, we propose a novel *ACM* that combines - in a global and unsupervised way - the advantages of the Self-Organizing Map (*SOM*) within the level set framework of a state-of-the-art unsupervised global *ACM*, the Chan-Vese (*C-V*) model. We term our proposed model *SOM-based Chan-Vese (SOMCV)* active contour model. It works by explicitly integrating the global information coming from the weights (prototypes) of the neurons in a trained *SOM* to help choosing whether to shrink or expand the current contour during the optimization process, which is performed in an iterative way. The proposed model can handle images that contain objects characterized by complex intensity distributions, and is at the same time robust to the additive noise. Experimental results show the high accuracy of the segmentation results obtained by the *SOMCV* model on several synthetic and real images, when compared to the Chan-Vese model [and other image segmentation models](#).

**Keywords** global region-based segmentation · variational level set method · active contours · Chan-Vese model · self organizing map · neural networks

## 1 Introduction

Active Contour Models (*ACMs*) usually deal with the image segmentation problem as a functional (also called infinite-dimensional) optimization problem, as they try to divide an image into several regions on the basis of

---

Mohammed M. Abdelsamea  
Department of Mathematics, Faculty of Science, University of Assiut, Assiut 71516, Egypt  
E-mail: m.abdelsamea@aun.edu.eg

Giorgio Gnecco  
IMT Institute for Advanced Studies, Piazza S. Ponziano 6, 55100 Lucca, Italy  
E-mail: giorgio.gnecco@imtlucca.it

Mohamed Medhat Gaber  
Robert Gordon University, Aberdeen, UK  
E-mail: m.gaber1@rgu.ac.uk

the maximization/minimization of a suitable energy functional. Starting from an initial contour, the optimization is performed in an iterative way, evolving the current contour with the aim of approximating better and better the actual object boundary (hence the denomination “active contour” models, which is used also for models that evolve the contour but are not based on the explicit minimization of a functional [1]). *ACMs* also allow to integrate learned information within the energy functional so as to guide efficiently the evolution of the current contour.

*Global Active Contour Models (global ACMs)* are called in such a way because they include in the energy functional global information about the intensity distributions of different regions of the image. Among global *ACMs*, we can distinguish between parametrized models (such as the *Snakes* model [2]), in which the contour is represented by a parametric curve, and variational level set methods (e.g., the *Chan-Vese* model [3]), for which the contour is the zero level set of a suitable function. Variational level set methods have the advantage on parametrized models of being able to model arbitrarily complex shapes, and to handle topological changes (e.g., in terms of the presence/absence of internal connectedness) of the regions to be segmented. Among global *ACMs*, region-based models [3–5] use statistical information about the regions to be segmented (e.g., intensity, texture, color distribution, etc.) to construct a stopping functional that is able to stop the contour evolution on the boundary between two different regions.

The main notion of most existing global *ACMs* is to make statistical assumptions on the intensity distributions of different regions of the image by a parametric density estimation approach, and to solve the segmentation problem by the Maximum Likelihood (*ML*) or Maximum A-Posteriori probability (*MAP*) approach [6]. As a result, if such intensity distributions do not match well the assumed parametric density models, such approaches will most likely fail. A possible solution to overcome the needs of a correct statistical information consists in letting a learning machine discover and model the intensity distributions using training images, helping in this way the contour to evolve when test images are presented. In this context, for the case of unsupervised learning, neural networks having the form of Self-Organizing Maps (*SOMs*) [7] have been used extensively for image segmentation, but in general, not in combination with *ACMs* [8, 9]. Moreover, in a few works, *SOMs* have been also used in combination with *ACMs*, with the explicit aim of modelling the contour and controlling its evolution, adopting a learning scheme similar to Kohonen’s learning algorithm, resulting in *SOM*-based *ACMs* (see [10] for a survey of such models). The evolution of the contour in *SOM*-based *ACMs* is guided by the feature space constructed by the *SOM* when learning the weights associated with the neurons of the map. As surveyed in [10], most existing *SOM*-based *ACMs* are sensitive to the contour initialization and to the additive noise, and a leaking problem (i.e., the presence of a final blurry contour) often occurs when an image has ill-defined edges. To overcome the limitations of such models, two supervised *SOM*-based *ACMs*, named respectively Concurrent *SOM*-based Chan-Vese (*CSOMCV*) model [11] and Self-Organizing Active Contour (*SOAC*) model [12], have been recently proposed. Such *SOM*-based *ACMs* integrate two *SOMs* (one for the foreground of the image to be segmented, the other one for the background) into a variational level set framework, in order to improve the robustness to the contour initialization and to the additive noise. The two models rely, respectively, on global and local regional information extracted from the image and learned by the two *SOMs*. Moreover, likewise other variational level set methods, and differently from other *SOM*-based *ACMs*, they are able to handle topological changes implicitly. However, *CSOMCV* and *SOAC* have a supervised component,

and supervision can be very costly in many computer vision applications, and also makes the segmentation process less automatic. This motivates the investigation of unsupervised *SOM*-based *ACMs* based on a variational level set framework.

In this paper, we propose a novel unsupervised global *SOM*-based *ACM*, which we term *SOM-based Chan-Vese (SOMCV)* active contour model, and relies on a set of trained self-organizing neurons. *SOMCV* does not only take advantage of a *SOM* as a tool to discover the intensity distribution of an image in a preprocessing phase, but also integrates the prototypes (weights) of the learned *SOM* into the level set framework of the Chan-Vese model, to better control the evolution of the contour. The main motivation for the *SOMCV* model is to make a set of neurons model globally the intensity distribution of the image by a self-organization learning procedure such that the topological structures of the intensity distributions of different regions of the image are preserved. As a result, the learned prototypes of the neurons, which control the topological preservation procedure, are used to approximate such intensity distributions and to integrate them implicitly - as global Region of Interest (*ROI*) descriptors - into the energy functional of the proposed model, to guide the contour evolution. Finally, we also introduce a simplified version of the model (termed *SOMCV<sub>s</sub>* model), which is obtained by taking the smallest possible cardinality for such sets of neurons, thus reducing the computational cost per iteration.

The contributions of this paper can be summarized as follows. We provide

- novel formulations for unsupervised active contour models, based on self-organizing neurons;
- new global regional descriptors to globally represent the regional intensity distributions during the evolution of the contour, without relying on particular statistical parametric models;
- a procedural guidance about how to implement the proposed models;
- a thorough experimental study pushing the boundaries of state-of-the-art techniques in terms of accuracy, efficiency, and robustness of the proposed models to the additive noise.

The paper is organized as follows. In Section 2 we review and discuss a representative state-of-the-art unsupervised global region-based *ACM* (the Chan-Vese model), and some of its variations. Section 4 presents the formulation of the proposed *SOMCV* and *SOMCV<sub>s</sub>* models in both a scalar-valued image segmentation framework and its extension to vector-valued images, and illustrates their numerical implementations. Section 5 presents experimental results comparing the segmentation accuracy of the proposed models and the one of the Chan-Vese model, on the basis of a number of real and synthetic images. Section 6 provides a discussion.

## 2 Unsupervised global *ACMs*

In this section, we briefly summarize the formulations of some well-known unsupervised global *ACMs*.

The Chan-Vese (*C-V*) model [3] is a well-known representative state-of-the-art global region-based *ACM*, which is based on Mumford-Shah segmentation techniques [13]. After its initialization, the contour in the *C-V* model is evolved iteratively in an unsupervised way with the aim of minimizing a suitable energy functional, constructed in such a way that its minimum is achieved in correspondence with a close approximation of the actual boundary between two different regions. The energy functional  $E_{CV}$  of the *C-V* model for a scalar-valued image

has the expression

$$\begin{aligned}
 & E_{CV}(C) \\
 & := \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}(\text{in}(C)) + \lambda^+ \int_{\text{in}(C)} (I(x) - c^+(C))^2 dx + \lambda^- \int_{\text{out}(C)} (I(x) - c^-(C))^2 dx,
 \end{aligned} \tag{1}$$

where  $C$  is a contour,  $I(x) \in \mathbb{R}$  denotes the intensity of the image indexed by the pixel location  $x$  in the image domain  $\Omega$ ,  $\mu \geq 0$  is a regularization parameter which controls the smoothness of the contour,  $\text{in}(C)$  (foreground) and  $\text{out}(C)$  (background) represent the regions inside and outside the contour, respectively, and  $\nu \geq 0$  is another regularization parameter, which penalizes a large area of the foreground. Finally,  $c^+(C) := \text{mean}(I(x)|x \in \text{in}(C))$  and  $c^-(C) := \text{mean}(I(x)|x \in \text{out}(C))$  are the mean intensities of the foreground and the background, respectively, and  $\lambda^+, \lambda^- \geq 0$  are parameters which control the influence of the two image energy terms  $\int_{\text{in}(C)} (I(x) - c^+(C))^2 dx$  and  $\int_{\text{out}(C)} (I(x) - c^-(C))^2 dx$ , respectively, inside and outside the contour. The functional is constructed in such a way that, when the regions  $\text{in}(C)$  and  $\text{out}(C)$  are smooth and “match” the true foreground and the true background, respectively,  $E_{CV}(C)$  reaches its minimum.

Following [14], in the variational level set formulation of (1), the contour  $C$  is expressed as the zero level set of an auxiliary function  $\phi : \Omega \rightarrow \mathbb{R}$ :

$$C := \{x \in \Omega : \phi(x) = 0\}. \tag{2}$$

In this way, the foreground and the background associated with the contour  $C$  can be also expressed as

$$\text{in}(C) := \{x \in \Omega : \phi(x) > 0\},$$

$$\text{out}(C) := \{x \in \Omega : \phi(x) < 0\}.$$

Note that different functions  $\phi(x)$  can be chosen to express the same contour  $C$ . For instance, denoting by  $d(x, C)$  the infimum of the Euclidean distances of the pixel  $x$  to the points on the curve  $C$ ,  $\phi(x)$  can be chosen as a signed distance function, defined as follows:

$$\phi(x) := \begin{cases} d(x, C), & x \in \text{in}(C), \\ 0, & x \in C, \\ -d(x, C), & x \in \text{out}(C), \end{cases} \tag{3}$$

This variational level set formulation has the advantage of being able to deal directly with the case of a foreground and a background that are not necessarily connected internally. When dealing with contours evolving in time, the function  $\phi(x)$  is replaced by a function  $\phi(x, t)$ .

After replacing  $C$  with  $\phi$  and highlighting the dependence of  $c^+(C)$  and  $c^-(C)$  on  $\phi$ , in the variational level set formulation of the  $C$ - $V$  model the (local) minimization of the cost (1) is performed by applying the gradient-descent technique in an infinite-dimensional setting [3], leading to the following Partial Differential Equation

(PDE), which describes the evolution of the contour<sup>1</sup>:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) [\mu \nabla \cdot (\nabla \phi / \|\nabla \phi\|) - \nu - \lambda^+ (I - c^+(\phi))^2 + \lambda^- (I - c^-(\phi))^2], \quad (4)$$

where  $\|\cdot\|$  is the Euclidean norm  $\|\cdot\|_2$ , and  $\delta(\cdot)$  is the Dirac generalized function. The first term in  $\mu$  of (4) keeps the level set function smooth, the second one in  $\nu$  controls the propagation speed of the evolving contour, while the third and fourth terms in  $\lambda^+$  and  $\lambda^-$  can be interpreted, respectively, as internal and external forces that drive the contour toward the actual object boundary. Then, Eq. (4) is solved iteratively in [3] by replacing the Dirac delta by a smooth approximation, and using a finite difference scheme. Sometimes, also a re-initialization step is performed, in which the current level set function  $\phi$  is replaced by its binarization (ie., a level set function of the form (3) with  $d(x, C)$  replaced by a constant  $\rho > 0$ , and representing the same current contour).

In the case of a vector-valued image  $\mathbf{I}(x) \in \mathbb{R}^D$  made up of  $D$  components (channels)  $I_i(x)$  for  $i = 1, \dots, D$  [15], one can proceed in a similar way, by replacing the functional (1) by

$$\begin{aligned} E_{CV}^{(D)}(C) \\ := \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}(\text{in}(C)) \\ + \frac{1}{D} \sum_{i=1}^D \lambda_i^+ \int_{\text{in}(C)} (I_i(x) - c_i^+(C))^2 dx + \frac{1}{D} \sum_{i=1}^D \lambda_i^- \int_{\text{out}(C)} (I_i(x) - c_i^-(C))^2 dx, \end{aligned} \quad (5)$$

where, for  $i = 1, \dots, D$ ,  $c_i^+(C), c_i^-(C)$  are the mean values of the channels  $I_i(x)$  on the foreground and the background, respectively, and  $\lambda_i^+, \lambda_i^- \geq 0$  are suitable parameters.

The  $C$ - $V$  model can also be derived, in a Maximum Likelihood setting, by making the assumption that the foreground and background follow Gaussian intensity distributions with the same variance [6]. Then, the model approximates globally the foreground and background intensity distributions by the two scalars  $c^+(\phi)$  and  $c^-(\phi)$ , respectively, which are their mean intensities. Similarly, Leventon et al. [16] proposed to use Gaussian intensity distributions with different variances inside a parametric density estimation method. Also, Tsai et al. in [17] proposed to use instead uniform intensity distributions to model the two intensity distributions. However, such models are known to perform poorly in the case of objects with inhomogeneous intensities [6].

As compared to such unsupervised global  $ACMs$ , our proposed solution consists instead in modeling globally the intensity distributions of the image and those of the foreground/background without using parametric models, but relying on a set of prototypes resulting from the training of a  $SOM$ .

### 3 Unsupervised $SOM$ -based $ACMs$

In this section, we briefly review some unsupervised  $ACMs$  based on  $SOMs$ , then we highlight their differences with respect to the proposed solution methods. We refer the reader to [10] for a more complete survey on such unsupervised  $SOM$ -based  $ACMs$ , and to [18] for more details on the relationship between variational level set-based  $ACMs$  and  $SOM$ -based  $ACMs$ .

<sup>1</sup> For both simplicity and uniformity of notation, in writing (4) and other  $PDEs$ , we do not show explicitly the arguments of the functions, which are already specified in other parts of the paper.

The basic idea of unsupervised *SOM*-based *ACMs* [19,20] is to model and implement the active contour using a *SOM* neural map, relying in the training phase on the edge map of the image (i.e., the set of points obtained by an edge-detection algorithm) to update the weights of the neurons of the network, and consequently to control the evolution of the active contour. The architecture of the *SOM* is composed of two layers: an input layer and an output layer. The points of the edge map act as inputs to the network, which is trained in an unsupervised way (in the sense that no supervised samples belonging to the foreground/background, respectively, are provided). As a result, during training the weights associated with the neurons in the output map move toward points belonging to the nearest salient contour.

The unsupervised *SOM*-based *ACM* proposed in [19] is a classical example of an *ACM* based on Self-Organizing Maps. This model requires a rough approximation of the true object boundary as an initial contour. The network is constructed and trained in an unsupervised way, based on the initial contour and the edge map information. The contour evolution is controlled by the edge information extracted from the image. The main steps of the algorithm developed in [19] can be described as follows.

1. Construct the edge map of the image to be segmented.
2. Initialize the contour to enclose the object of interest in the image.
3. Obtain the horizontal and vertical coordinates of the edge points to be presented as inputs to the network.
4. Construct a *SOM* with a number of neurons equal to the number of the edge points of the initial contour and two weights associated with each neuron. The points on the initial contour are used to initialize the weights of the *SOM*.
5. Repeat the following steps for a fixed number of iterations:
  - (a) Select randomly an edge point and feed its coordinates to the network.
  - (b) Determine the best-matching neuron.
  - (c) Update the weights of the neurons in the network by an unsupervised learning scheme composed of a competitive phase and a cooperative one.
  - (d) Compute a neighborhood parameter for the contour according to the updated weights and a threshold.

However, this model is very sensitive to contour initialization, and typically fails if the initial contour is far from the actual object boundary. Moreover, the contours inside an object cannot be extracted if the initial contour is outside the object [1]. In order to improve the model, the *Time Adaptive Self-Organizing Map (TASOM)* model was proposed in [20] such that an individual learning rate and individual neighborhood parameters are used for each neuron. In [21], the *Batch-SOM (BSOM)* was proposed, with the aim of integrating the advantages of unsupervised *SOM*-based *ACMs* with the Snake model. A modified version of *BSOM* was proposed in [22] to increase the smoothness of the contour, control the size of the weights of the network, and prevent the contour from being extended over the object boundary. Similarly, *BSOM* was used in [23] to adjust the initial contour to the exact shape of the pupil. However, likewise the *SOM* model of [19] described above, the edge information is used to train the networks in such works. As a result, also these kinds of *ACMs* are sensitive to the choice of the initial contour. For example, the contours inside an object cannot be extracted if the initial contour is outside the object [1] and both [19,20] require the initialized contour to be close and similar to the actual boundary. Likewise other edge-based *ACMs*, most existing unsupervised *SOM*-based *ACMs* are also very sensitive to the noise and a leaking problem (i.e., the presence of a final blurry contour) occurs when an image has ill-defined edges.

In contrast to such *SOM*-based *ACMs*, our proposed models are *ACMs* implemented by a variational level set method, and integrate a *SOM* into the segmentation framework in order to be robust to the contour initialization and to the additive noise, as they rely on global regional information extracted from the image, besides having the advantage of handling topological changes implicitly, likewise other variational level set methods. Recently, another unsupervised *SOM*-based *ACM* has also been proposed in [24] with the aim of developing an *ACM* that is at the same time effective in handling complex images containing intensity inhomogeneity, and robust with respect to the location of the initial contour and to the additive noise. Such a model relies on the global information coming from selected prototypes associated with a *SOM*, which is trained off-line in an unsupervised way to model the intensity distribution of an image, and used on-line to segment an identical or similar image. However, the model in [24] is also computationally very expensive, because both global and local information are combined in the on-line phase to improve its robustness to the contour initialization.

As observed and discussed above, most existing unsupervised *ACMs* either rely on the use of parametric models or on the updates of the weights of the *SOM* neurons on the basis of the edge information of the image. Such models may lead to unsatisfactory segmentation results due to the lack of extracted global information considered by these models. Our proposed solution, instead, is to globally discover the topological structures of the foreground and background intensity distributions by using unsupervised neural networks. As a result, the prototypes of selected trained neurons are proposed to be used as efficient global regional descriptors to approximate globally the two intensity distributions. These descriptors are adapted and updated to track the dissimilarity in the intensity of the two different regions in the image domain.

#### 4 The *SOMCV* and *SOMCV<sub>s</sub>* models

In this section, we describe our *SOM*-based Chan-Vese (*SOMCV*) active contour model and its modification *SOMCV<sub>s</sub>*. We first consider the case of scalar-valued images in Subsection 4.1. Then, in Subsection 4.2, we briefly discuss the changes needed to deal with the case of vector-valued images. Finally, in Subsection 4.3, algorithmic details are provided.

##### 4.1 The *SOMCV* and *SOMCV<sub>s</sub>* models for scalar-valued images

Both the *SOMCV* and *SOMCV<sub>s</sub>* segmentation frameworks for scalar-valued images are composed of two sessions: an unsupervised training session and a testing session, which are performed, respectively, off-line and on-line.

In the training session, after choosing a suitable number of neurons and topology for the *SOM* (a 1-*D* grid is preferable, since in this case the input to the *SOM* is scalar-valued), the intensity  $I^{(tr)}(x_t)$  of a randomly-extracted pixel  $x_t$  of a training image<sup>2</sup> is applied as input to the *SOM* at time  $t = 0, 1, \dots, t_{\max}^{(tr)} - 1$ , where  $t_{\max}^{(tr)}$  is the number of iterations in the training of the *SOM*<sup>3</sup>. Then, the neurons are trained in a self-organized way

<sup>2</sup> In this paper, training pixels from one image are considered in the training session of the *SOMCV* and *SOMCV<sub>s</sub>* models. Such an image is either identical or similar to the image presented in the testing session. In the first case, using even identical images for the training and testing sessions is not a limitation of the models: one of the reasons is that the training is unsupervised.

<sup>3</sup> Instead of sequential training, batch training [25] of the *SOM* may also be used for a faster convergence.

in order to be able to preserve the topological structure of the image intensity distribution at the end of training. Each neuron  $n$  is connected to the input by a weight vector  $w_n$  of the same dimension as the input (which - in this scalar case - is of dimension 1). After their random initialization, the weights  $w_n$  are updated by the following self-organization learning rule:

$$w_n(t+1) := w_n(t) + \eta(t)h_{bn}(t)[I^{(tr)}(x_t) - w_n(t)], \quad (6)$$

where  $\eta(t)$  is a learning rate. Also,  $h_{bn}(t)$  is a neighborhood kernel around the Best-Matching Unit (*BMU*) neuron  $b$  (i.e., the neuron whose weight vector is the closest to the input of the *SOM*, in this case  $I^{(tr)}(x_t)$  at time  $t$ ). Both functions  $\eta(t)$  and  $h_{bn}(t)$  are designed to be time-decreasing in order to stabilize the weights  $w_n(t)$  for  $t$  sufficiently large. In this way - due to the well-known properties [7] of the self-organization learning rule (6) - when the training session is completed, one can model and often approximate accurately the input intensity distribution by associating each input pixel intensity (or a weighted average intensity of several input pixels) to the weight of the corresponding *BMU* neuron. In particular, in the following we consider the choice

$$\eta(t) := \eta_0 \exp\left(-\frac{t}{\tau_\eta}\right), \quad (7)$$

where  $\eta_0 > 0$  is the initial learning rate and  $\tau_\eta > 0$  is a time constant, whereas  $h_{bn}(t)$  is selected as a Gaussian function centered on the winning neuron, i.e., it has the form

$$h_{bn}(t) := \exp\left(-\frac{\|r_b - r_n\|^2}{2r^2(t)}\right), \quad (8)$$

where  $r_b, r_n \in \mathbb{R}^2$  are the location vectors in the output neural map of neurons  $b$  and  $n$ , respectively, and  $r(t) > 0$  is a time-decreasing neighborhood radius<sup>4</sup>. In particular, in the following we make the choice

$$r(t) := r_0 \exp\left(-\frac{t}{\tau_r}\right), \quad (9)$$

where  $r_0 > 0$  is the initial neighborhood radius of the map, and  $\tau_r > 0$  is another time constant. These choices are in accordance with the recent literature about *SOM*s [26].

Since, after training, the inputs to the network are topologically arranged in the output map on the basis of the prototypes of the neurons that have the smallest distances from the inputs, we say that the learned prototypes have a global *Self-Organizing Topology Preservation (SOTP)* property, which allows one to represent the intensity distributions inside and outside the contour globally during the contour evolution.

Once the training of the *SOM* has been accomplished, the trained network is applied on-line in the testing session, during the evolution of the contour  $C$ , to approximate and describe globally the foreground and background intensity distributions of a similar test image  $I(x)$ . Indeed, during the contour evolution, the two scalar intensities  $\text{mean}(I(x)|x \in \text{in}(C))$  and  $\text{mean}(I(x)|x \in \text{out}(C))$  are presented as inputs to the trained network.

<sup>4</sup> This choice of the function  $h_{bn}(t)$  implies that, for fixed  $t$ , when  $\|r_b - r_n\|$  increases,  $h_{bn}(t)$  decreases to zero gradually to smooth out the effect of the *BMU* neuron on the weights of the neurons far from the *BMU* neuron itself, and when  $t$  increases, the influence of the *BMU* neuron becomes more and more localized.

We now define, for each neuron  $n$ , the quantities

$$A_n^+(C) := |w_n - \text{mean}(I(x)|x \in \text{in}(C))|, \quad (10)$$

$$A_n^-(C) := |w_n - \text{mean}(I(x)|x \in \text{out}(C))|, \quad (11)$$

which are, respectively, the distances of the associated prototype  $w_n$  from the mean intensities of the current approximations of the foreground and the background, and are also repeatedly calculated during the testing session. Then, we define the two sets

$$\{w_j^+(C)\} := \{w_n : A_n^+(C) \leq A_n^-(C)\}, \quad (12)$$

$$\{w_j^-(C)\} := \{w_n : A_n^+(C) > A_n^-(C)\}, \quad (13)$$

of cardinalities  $N^+(C) := |\{w_j^+(C)\}|$  and  $N^-(C) := |\{w_j^-(C)\}|$ , which are the sets of neurons whose prototypes are associated, respectively, with the current approximations of the foreground and the background. Such prototypes are chosen as representatives of the foreground and background intensity distributions according to their closeness to the two mean intensities. So, they are extracted as global regional intensity descriptors and included in the energy functional to be minimized in our proposed *SOMCV* model, which has the following expression:

$$E_{SOMCV}(C) := \lambda^+ \int_{\text{in}(C)} e^+(x, C) dx + \lambda^- \int_{\text{out}(C)} e^-(x, C) dx, \quad (14)$$

$$e^+(x, C) := \sum_{j=1, \dots, N^+(C)} \left( I(x) - w_j^+(C) \right)^2, \quad (15)$$

$$e^-(x, C) := \sum_{j=1, \dots, N^-(C)} \left( I(x) - w_j^-(C) \right)^2, \quad (16)$$

where the parameters  $\lambda^+, \lambda^- \geq 0$  are, respectively, the weights of the two image energy terms  $\int_{\text{in}(C)} e^+(x, C) dx$  and  $\int_{\text{out}(C)} e^-(x, C) dx$ , inside and outside the contour.

Now, as in Section 2, we replace the contour curve  $C$  with the level set function  $\phi$ , obtaining

$$E_{SOMCV}(\phi) = \lambda^+ \int_{\phi > 0} e^+(x, \phi) dx + \lambda^- \int_{\phi < 0} e^-(x, \phi) dx, \quad (17)$$

where we have also made explicit the dependence of  $e^+$  and  $e^-$  on  $\phi$ . In terms of the Heaviside step function  $H : \mathbb{R} \rightarrow \mathbb{R}$ , defined as

$$H(z) := \begin{cases} 1 & \text{if } z \geq 0, \\ 0 & \text{if } z < 0, \end{cases} \quad (18)$$

the *SOMCV* energy functional can be also written as follows:

$$E_{SOMCV}(\phi) = \lambda^+ \int_{\Omega} e^+(x, \phi) H(\phi(x)) dx + \lambda^- \int_{\Omega} e^-(x, \phi) (1 - H(\phi(x))) dx. \quad (19)$$

Finally, proceeding likewise in Section 2, the evolution of the contour is described by

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[ -\lambda^+ e^+ + \lambda^- e^- \right], \quad (20)$$

which shows how the learned neurons are used to determine the internal and external forces acting on the contour. Apart from this difference, Eq. (20) has a similar form as Eq. (4), and can be solved iteratively using the same smoothing and discretization techniques.

Another difference with the  $C$ - $V$  model is the absence in the functional (14) of the regularization terms in  $\mu$  and  $\nu$ , which appear, instead, in the functional (1). This can be justified as follows. As pointed out in [27, 28], the convolution of the current level set function with a Gaussian filter can be used as an efficient and robust approach to regularize it. In such an approach, the width of the Gaussian filter is used to control the regularization strength, as the parameters  $\mu$  and  $\nu$  do in the  $C$ - $V$  model. So, likewise in [27], we have not included in our formulation the regularization parameters  $\mu$  and  $\nu$  which appear in the functional (1), but we perform - at each iteration of a finite-difference approximation of (20) - the regularization of the current level set function by replacing it with its convolution with a Gaussian filter of suitable width. Such a convolution is preceded by a binarization of the function  $\phi$ , without loss of information about the current contour. So, in this sense,  $SOMCV$  is a Gaussian Regularizing Level Set Model ( $GRLSM$ ). In [27, 28], such kind of regularization was shown to be more efficient and effective than the one employed in the  $C$ - $V$  model through its regularization terms in  $\mu$  and  $\nu$ .

In the following, we describe also a simplification of the  $SOMCV$  model (which we term  $SOMCV_s$  model), which is based on an energy functional whose evaluation is easier from a computational point of view than the one of (14). This is obtained by replacing the sets  $\{w_j^+(C)\}$  and  $\{w_j^-(C)\}$  above by single prototypes  $w_b^+$  and  $w_b^-$ , defined as follows:

$$w_b^+(C) := \operatorname{argmin}_n |w_n - \operatorname{mean}(I(x)|x \in \operatorname{in}(C))|, \quad (21)$$

$$w_b^-(C) := \operatorname{argmin}_n |w_n - \operatorname{mean}(I(x)|x \in \operatorname{out}(C))|, \quad (22)$$

where  $w_b^+(C)$  is the prototype of the  $BMU$  neuron to the mean intensity inside the current contour, while  $w_b^-(C)$  is the prototype of the  $BMU$  neuron to the mean intensity outside it. Then, we define the functional of the  $SOMCV_s$  model as

$$E_{SOMCV_s}(C) := \lambda^+ \int_{\operatorname{in}(C)} e_s^+(x, C) dx + \lambda^- \int_{\operatorname{out}(C)} e_s^-(x, C) dx, \quad (23)$$

$$e_s^+(x, C) := \left( I(x) - w_b^+(C) \right)^2, \quad (24)$$

$$e_s^-(x, C) := \left( I(x) - w_b^-(C) \right)^2. \quad (25)$$

Then, proceeding as above, after replacing  $C$  with the level set function  $\phi$ , the evolution of the contour is described by

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[ -\lambda^+ e_s^+ + \lambda^- e_s^- \right]. \quad (26)$$

The two terms  $e_s^+(x, C)$  and  $e_s^-(x, C)$  in (24) and (25), respectively, have expressions similar to the corresponding ones used in the *CSOMCV* model. The difference is that the terms  $e_s^+(x, C)$  and  $e_s^-(x, C)$  in the *CSOMCV* model are constructed in a supervised way (e.g., they are originated from two different *SOMs*, one trained on a subset of pixels of the foreground, and the other one on a subset of pixels of the background), whereas the terms  $e_s^+(x, C)$  and  $e_s^-(x, C)$  in the *SOMCV<sub>s</sub>* model are constructed in an unsupervised way, employing only one *SOM* trained on a subset of pixels from both regions. Although the expressions  $e_s^+(x, C)$  and  $e_s^-(x, C)$  in (24) and (25) are similar to those of the terms  $(I(x) - c^+(C))^2$  and  $(I(x) - c^-(C))^2$  used in the formulation (1) of the *C-V* model, the prototypes  $w_b^+(C)$  and  $w_b^-(C)$  may represent globally the two regional intensity distributions better than the mean intensities in the two regions. This can be shown in the following way: suppose that the current contour  $C$  coincides with the actual object boundary, but that the image contains additive noise: then, the values of the mean regional intensities  $c^+(C) := \text{mean}(I(x)|x \in \text{in}(C))$  and  $c^-(C) := \text{mean}(I(x)|x \in \text{out}(C))$  depend on  $C$  in a continuous way, likely making the contour evolve toward a worse approximation of the object boundary. Instead, the values of  $w_b^+(C)$  and  $w_b^-(C)$  may not change at all for small changes of  $C$ , providing more robustness of the model with respect to the additive noise. In order to obtain such a behavior, one should keep the size of the network small. Otherwise, when using a network with a large number of neurons (then of prototypes), one may more likely obtain  $w_b^+(C) \cong \text{mean}(I(x)|x \in \text{in}(C))$  and  $w_b^-(C) \cong \text{mean}(I(x)|x \in \text{out}(C))$ , losing the just-mentioned robustness.

Moreover, when the foreground/background intensity distributions are characterized by many different intensities, minimizing the functional (1) - in which the dependence on the foreground/background intensity distributions is expressed only in terms of the mean regional intensities  $c^+(C)$  and  $c^-(C)$  - may result in under(over)-segmentation problems. Of course, such problems are still not solved by replacing  $c^+(C)$  and  $c^-(C)$  with the prototypes  $w_b^+(C)$  and  $w_b^-(C)$ , since also  $w_b^+(C)$  and  $w_b^-(C)$  are only scalar quantities. So, in the case of skewness/multimodality of the two distributions, one expects better segmentation results when using the functional (14) of the *SOMCV* model, which represents the foreground/background intensity distributions by larger sets of weights for each of the two regions, as compared to the functional (23) of the *SOMCV<sub>s</sub>* model.

In Section 5, the robustness of the proposed model to the additive noise and to intensity distributions characterized by many intensity values is investigated experimentally.

#### 4.2 The *SOMCV* and *SOMCV<sub>s</sub>* models for vector-valued images

The *SOMCV* and *SOMCV<sub>s</sub>* models can be extended to the case of vector-valued images. Such an extension is particularly useful for the segmentation of multi-spectral images (see Section 5 for some related experiments). In the vectorial case, the image  $\mathbf{I}(x)$  is made up of  $D$  channels  $I_i(x)$  ( $i = 1, \dots, D$ ), and also the *SOM* weights are vectors of dimension  $D$ . The only significant changes with respect to the scalar case described in Subsection 4.1 are that, in the determination of the *BMU* neuron, the absolute values in formulas (10) and (11) are replaced by Euclidean norms in  $\mathbb{R}^D$ , and that in this case also a higher-dimensional grid (e.g., a  $2-D$  grid) is appropriate for the *SOM*, since its input has now dimension greater than 1.

### 4.3 Algorithmic description

Having discussed the formulations of the  $SOMCV$  and  $SOMCV_s$  models, in the following, the procedural steps of their training and testing sessions are summarized in Algorithm 1 (to avoid redundancy, only the case of scalar-valued images is detailed here).

---

**Algorithm 1**  $SOMCV$  and  $SOMCV_s$  segmentation frameworks for scalar-valued images
 

---

**1: procedure**

- Input:
  - Training and test scalar-valued images.
  - Number of neurons and topology of the network (with 1-dimensional prototypes).
  - Number of iterations  $t_{\max}^{(tr)}$  for training the neural map.
  - Maximum number of iterations  $t_{\max}^{(evol)}$  for the contour evolution.
  - $\eta_0 > 0$ : starting learning rate.
  - $r_0 > 0$ : starting radius of the map.
  - $\tau_\eta, \tau_r > 0$ : time constants in the learning rate and contour smoothing parameter.
  - $\lambda^+, \lambda^- \geq 0$ : weights of the energy terms, respectively, inside and outside the contour.
  - $\sigma > 0$ : Gaussian contour smoothing parameter.
  - $\rho > 0$ : constant in the binary approximation of the level set function.
- Output:
  - Segmentation result.

*TRAINING SESSION:*

- 2: Initialize randomly the prototypes of the neurons.
- 3: **repeat**
- 4:     Choose randomly a pixel  $x_t$  in the image domain  $\Omega$  and determine the  $BMU$  neuron to the input intensity  $I^{(tr)}(x_t)$ .
- 5:     Update the prototypes  $w_n$  using (6), (7), (8), and (9).
- 6: **until** learning of the prototypes is accomplished (i.e., the number of iterations  $t_{\max}^{(tr)}$  is reached).

*TESTING SESSION:*

- 7: Choose a subset  $\Omega_0$  (e.g., a rectangle) in the image domain  $\Omega$  with boundary  $\Omega'_0$ , and initialize the level set function as:

$$\phi(x) := \begin{cases} \rho, & x \in \Omega_0 \setminus \Omega'_0, \\ 0, & x \in \Omega'_0, \\ -\rho, & x \in \Omega \setminus (\Omega_0 \cup \Omega'_0). \end{cases} \quad (27)$$

- 8: Choose the functional to be minimized (the  $E_{SOMCV}$  functional (14) or the  $E_{SOMCV_s}$  functional (23)).
- 9: **repeat**
- 10:   **if**  $E_{SOMCV}$  functional (14) has been chosen **then**
- 11:     Determine, for each neuron, the quantities  $A_n^+$  and  $A_n^-$  from (10) and (11), then the sets  $\{w_j^+\}$  and  $\{w_j^-\}$  from (12) and (13).
- 12:     Evolve the level set function  $\phi$  according to a finite difference approximation of (20).
- 13:   **else**
- 14:     Calculate  $w_b^+$  and  $w_b^-$  from (21) and (22).
- 15:     Evolve the level set function  $\phi$  according to a finite difference approximation of (26).
- 16:   **end if**
- 17:   At each iteration of the finite-difference scheme, re-initialize the current level set function to be binary by performing the update

$$\phi \leftarrow \rho (H(\phi) - H(-\phi)), \quad (28)$$

then regularize by convolution the obtained level set function:

$$\phi \leftarrow g_\sigma * \phi, \quad (29)$$

where  $g_\sigma$  is a Gaussian kernel with  $\int_{\mathbb{R}^2} g_\sigma(x) dx = 1$  and width  $\sigma$ , and  $*$  is the convolution operator.

- 18:   **until** the curve evolution converges (i.e., the curve does not change anymore) or the maximum number of iterations  $t_{\max}^{(evol)}$  is reached.
  - 19: **end procedure**
-

Algorithm 1 can be explained as follows. Once the topology of the neural map is defined, the neurons of the map start to be trained using a learning algorithm composed of a competitive phase and a cooperative one (see formula (6)). As a result, through the prototypes of the neurons, the set of the trained neurons carries significant information about the intensity distribution of the given image, which reflects the topological structure of the intensity distribution. Once the training is accomplished, the prototypes of selected neurons in the case of the functional (14) of the *SOMCV* model - or of the best-matching neurons to the mean intensities in the two regions, in the case of the functional (23) of the *SOMCV<sub>s</sub>* model - are used as global regional descriptors for the foreground and background intensity distributions. Then, in the testing phase, they are used as core components of the level set energy functional to guide the evolution of the contour. Moreover, in order to keep the contour and the level set function smooth at each iteration without losing information on the displacement of the current contour, the current level set function  $\phi$  is first re-initialized to be binary, then convolved with a Gaussian kernel function. The smoothness degree of the updated level set function is controlled by the width parameter  $\sigma$  of the Gaussian as described in Subsection 4.1.

Fig. 1 illustrates the off-line and on-line components of the *SOMCV* and *SOMCV<sub>s</sub>* models in a vector-valued (more specifically, *RGB*) image segmentation framework (the scalar case is similar, but uses scalar prototypes and preferably a 1-*D* grid). Fig. 1(a) shows the input layer of the *SOM*, whose dimension is equal to the one of the voxel intensities of the image to be segmented. For example, in the case of *RGB* images, the input layer of the map has dimension 3, since it receives the *R*, *G*, and *B* channels of the vector-valued image. The red cube in Fig. 1(a) represents a voxel intensity presented as input to the *SOM*, in this case made up of  $3 \times 3$  neurons (Fig. 1(b)). The small circles in Fig. 1(b) represent the neurons of the map, where each neuron is associated with a three-dimensional prototype, of the same dimension as the input. The prototypes of the neurons are modified during the training phase. This is accomplished by finding the best-matching neuron (the blue circle in Fig. 1(b)) to each input voxel intensity, and updating its prototype and the ones of all its neighbors as described in formulas (6), (7), (8), and (9), extended to the three-dimensional case as described in Subsection 4.2. Once the learning is accomplished, the prototypes associated with selected neurons of the learned map (Fig. 1(b)) are ready to be integrated into the energy functional (14) during the on-line session (i.e., during the curve evolution process) as global regional intensity descriptors. Fig. 1(d) represents a test image to be segmented (the gray circle represents the foreground). Starting from an initial contour (the black curve in Fig. 1(d)), the mean intensities of inside and outside the contour are presented as inputs to the learned map in Fig. 1(c) to classify (see Fig. 1(e), top) the prototypes associated with the neurons into foreground (in red) and background (in black) global intensity descriptors. Then, the contour evolution is guided by the extracted prototypes associated with the two sets of foreground and background neurons. In the case of *SOMCV<sub>s</sub>* (see Fig. 1(e), down), only one prototype is used as a global intensity descriptor for each region.

## 5 Experimental study

In this section, we demonstrate the effectiveness and robustness of the *SOMCV* and *SOMCV<sub>s</sub>* models, compared to the *C-V* model described in Section 2, in handling real and synthetic images. For a fair comparison, the *SOMCV*, *SOMCV<sub>s</sub>* and the *C-V* model used in this experiment are all implemented in Matlab R2012a on a

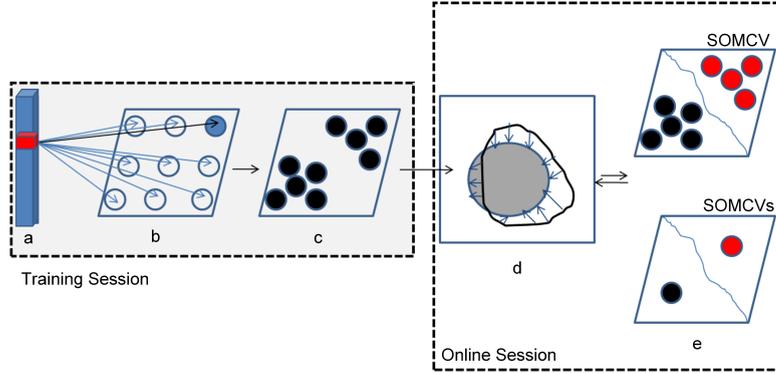


Fig. 1: The architecture of *SOMCV* for *RGB* images: (a) the input intensities of a training voxel; (b) a  $3 \times 3$  *SOM* neural map (with a three-dimensional prototype associated with each neuron); (c) the trained *SOM*; (d) the contour evolution process; and (e) the foreground (in red) and background (in black) representative neurons for the *SOMCV* (top) and the *SOMCV<sub>s</sub>* (down) models. For a scalar-valued image, a similar model is used, but the prototypes have dimension 1, and a 1-*D* grid is used.

PC with the following configuration: 2.5 GHz Intel(R) Core(TM) 2 Duo, and 2.00 GB RAM<sup>5</sup>. In each experiment, the  $r_0$  and  $\sigma$  parameters are expressed in pixels. Moreover, the *SOMCV* and *SOMCV<sub>s</sub>* parameters are fixed<sup>6</sup> as follows:  $\eta_0 = .9$ ,  $\sigma = 1.5$ , and the weight parameters (i.e.,  $\lambda^+$ ,  $\lambda^-$  for the scalar-valued case, and  $\lambda_i^+$ ,  $\lambda_i^-$  in the vector-valued case) are fixed to 1. Also,  $r_0 := \max(M, N)/2$ , where  $M$  and  $N$  are the numbers of rows and columns of the installed neural map,  $t_{\max}^{(tr)} = 10000$ ,  $t_{\max}^{(evol)} = 1000$ ,  $\tau_\eta := t_{\max}^{(tr)}$ ,  $\tau_r := t_{\max}^{(tr)} / \ln(r_0)$ ,  $\rho = 1$ . For the experiments performed on the scalar-valued images considered in the paper, the *SOM* network has been chosen as a 1-*D* neural map composed of 5 neurons (i.e.,  $M = 5$  and  $N = 1$ ), whereas for the case of vector-valued images, it was a  $3 \times 3$  grid of neurons in most experiments ( $M = N = 3$ ) and a  $2 \times 2$  grid ( $M = N = 2$ ) for the other experiments (see Tables 3 and 4). In the *C-V* model,  $\lambda^+$ ,  $\lambda^-$  for the scalar-valued case and  $\lambda_i^+$ ,  $\lambda_i^-$  in the vector-valued case are also fixed to 1,  $\mu$  is chosen such that the final contour is smooth enough and  $\nu = 0$  (as made in [15, p. 268]). Moreover, the *SOMCV<sub>s</sub>* model is considered in the comparison with the same parameters of the *SOMCV* model. Unless stated otherwise, the training image used in the unsupervised training session coincides with the test image. Otherwise, it is an image similar to the test image (obtained, e.g., by adding Gaussian noise). In all the testing sessions, the initial contour has been chosen as rectangular. For the case of gray-level images, the range of the values assumed by the intensity is 0-255 as all the considered gray-level images are 8-bit images.

Fig. 2 shows the fast convergence of *SOMCV* (and its variation *SOMCV<sub>s</sub>*) for scalar-valued images and the associated contour evolution process when compared to the *C-V* model. As Fig. 2 shows, the contours obtained by the *SOMCV* and *SOMCV<sub>s</sub>* models have converged to the respective final contours with similar numbers of performed iterations and similar performances because of the large intensity homogeneity of the image considered in the experiment.

Fig. 3 illustrates the effectiveness and robustness with respect to noise of *SOMCV* in handling images containing objects characterized by many different intensities and skewness/multimodality of the foreground intensity distribution. The figure also compares the proposed *SOMCV* model with other unsupervised *ACMs*.

<sup>5</sup> The developed code is available at <http://mohammedabdelsamea.weebly.com>.

<sup>6</sup> In the experiments presented in Fig. 3 and 8, also the choice  $\sigma = .5$  was considered, together with  $\sigma = 1.5$ .

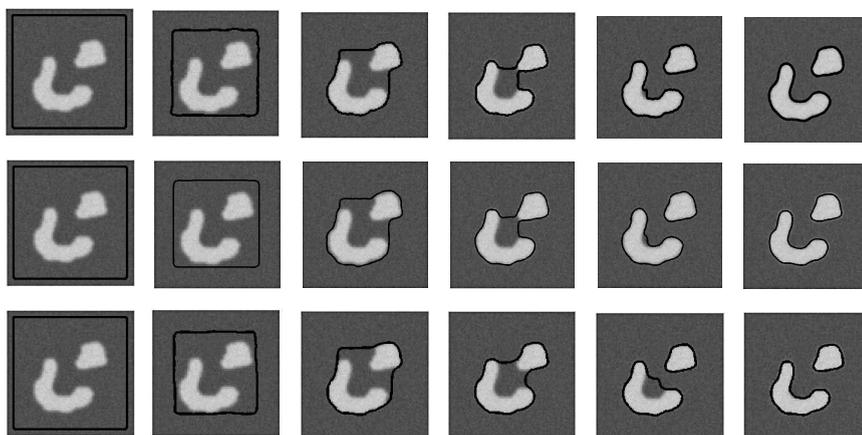


Fig. 2: The rapid contour evolution of the  $SOMCV$  and  $SOMCV_s$  models when compared to the contour evolution of the  $C-V$  model, in the scalar case. The first and second rows show, respectively, the contour evolution of  $SOMCV$  and  $SOMCV_s$ . From left to right: initial contour (in black), contour after 3, 6, 9, 12 iterations, and final contour (15 iterations). The third row shows the contour evolution of the  $C-V$  model. From left to right: initial contour (in black), contour after 50, 100, 150, 200 iterations, and final contour (260 iterations).

In particular, as compared to the  $C-V$  model, the  $SOMCV$  model has shown better results, due to its automated ability to preserve the topological structure of the foreground intensity distribution (this is not needed, instead, for the background distribution, which is simpler). Moreover, Fig. 3 also compares the proposed  $SOMCV$  model to the unsupervised  $SOM$ -based  $ACM^7$  recently proposed in [24], which relies on both local and global intensity information with the aim of improving the robustness to the contour initialization. As demonstrated in Fig. 3,  $SOMCV$  has been able to find better the objects and their boundaries. Finally, the segmentation performance of the  $SOMCV_s$  model is quite similar to the one of the  $SOMCV$  model, but is also more sensitive to the noise.

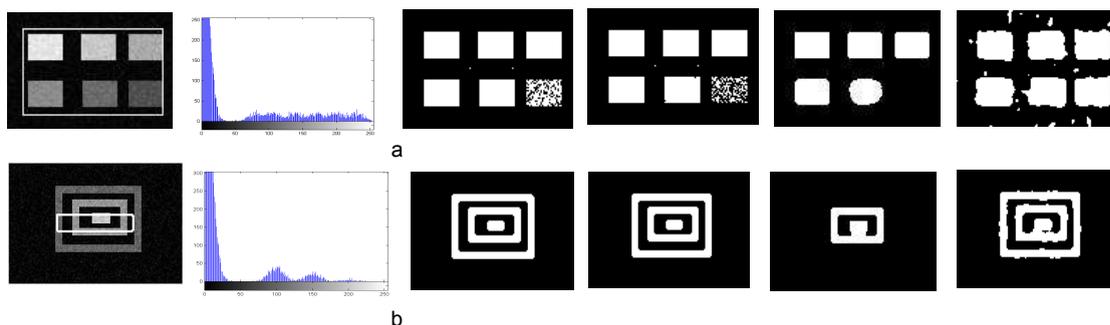


Fig. 3: The effectiveness of the  $SOMCV$  model in dealing with objects characterized by many different intensities and skewness/multimodality of the foreground intensity distribution. Arranged in rows there are: (a) a noisy  $140 \times 100$  image (with Gaussian noise added, standard deviation  $SD = 10$ ) with six different intensities 80, 100, 140, 170, 200, and 230 in its foreground; (b) a noisy  $90 \times 122$  image (with Gaussian noise added, standard deviation  $SD = 10$ ) with three different intensities 100, 150, and 200 in its foreground. The columns from left to right show: the images with the additions of the initial contours, the histograms of the intensities of the images, and, respectively, the segmentation results of the  $SOMCV$  model and the  $SOMCV_s$  model ( $\sigma = .5, 1.5$  have been used, respectively, for (a) and (b)), of the  $C-V$  model, and of the model proposed in [24].

<sup>7</sup> Since in the paper we are interested in unsupervised  $ACMs$ , we have not compared the proposed  $SOMCV$  model to the  $SOM$ -based  $ACMs$  proposed in [11] and [12], which are supervised models.

As a motivating example for the development of the proposed *SOMCV* model, Fig. 4 shows the behavior of several other unsupervised *ACMs* from different classes, which have been tested on the image of Fig. 3 (second row), providing unsatisfactory segmentation results. As illustrated in Fig. 4, the model proposed in [29] - which is an unsupervised global *ACM* based on the means of sign pressure forces, and relies on strong statistical assumptions - has failed in finding properly the foreground. Similarly, the boundary-based active contour model from [30] - whose energy functional includes a Laplacian image term - has failed in segmenting the same image. Also the models of [31] and [32] - which are local *ACMs* (i.e., they do not take into account spatial dependencies among the pixels) - have shown unsatisfactory results, possibly due to the high sensitivity to the contour initialization and noise, which is typical of local *ACMs*.



Fig. 4: The segmentation results obtained by some well-known unsupervised *ACMs* on the image of Fig. 3 (second row). From left to right: the original image with the initial contour, and the segmentation results obtained by the unsupervised *ACMs* proposed in [29], [30], [31], and [32], respectively.

Then, in order to demonstrate the robustness of *SOMCV* and *SOMCV<sub>s</sub>* to the additive noise, in the experiment described in Fig. 5 we have used the top left image of Fig. 3 in the training session of *SOMCV* and *SOMCV<sub>s</sub>*, then the trained *SOM* (whose values of the weights are common to the two models) has been applied on-line to various test images obtained adding to such an image different levels of Gaussian noise. As shown in Fig. 5, for this case *SOMCV* is more robust and less sensitive to the additive noise than *SOMCV<sub>s</sub>*, since the regions of the foreground are detected more accurately by *SOMCV*.

Similarly, the image of Fig. 3(b) has been used in the training session of *SOMCV* and *SOMCV<sub>s</sub>*, then the trained *SOM* has been applied on-line to various test images obtained by adding to such an image different levels of Gaussian noise, as shown in Fig. 6. The results of these two experiments show the ability of *SOMCV* to find all the different regions of the object (which is characterized by many different intensities), and also its robustness to the additive noise and to the skewness/multimodality of the foreground intensity distribution. They also demonstrate that, in the case of images containing objects characterized by many different intensities or by skewed/multimodal intensity distributions, *SOMCV* is expected to produce better results than *SOMCV<sub>s</sub>*.

Fig. 7 illustrates the effectiveness of *SOMCV* in handling real and synthetic scalar-valued images. The segmentation results of the *SOMCV* model on the real images shown in the first and second columns show the ability of *SOMCV* to segment objects with blurred edges and background, while the *C-V* model provides a worse segmentation for the image in the first column, and incurs in an under-segmentation problem for the image in the second column. Similarly, *SOMCV* outperforms *C-V* also in handling synthetic images as shown in the third and fourth columns. Moreover, *SOMCV* and *SOMCV<sub>s</sub>* behave exactly the same as *C-V* in handling binary gray images as in the case of the image shown in the right-most column. This is because in this case the mean intensities inside and outside the contour are accurate enough to approximate the foreground/background intensity distributions. For the images presented in Fig. 7, *SOMCV* outperforms also *SOMCV<sub>s</sub>*.

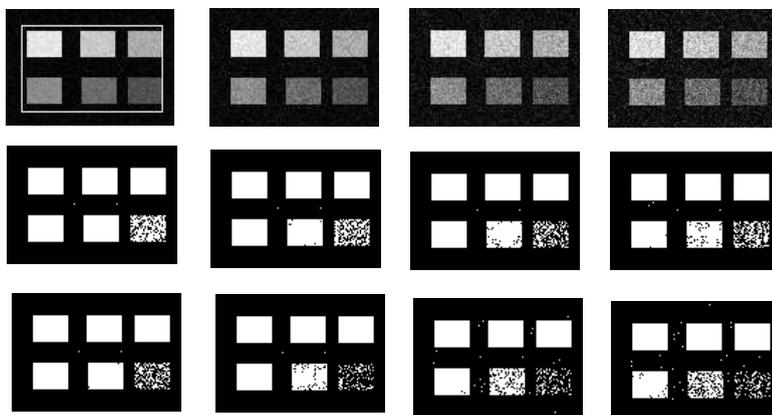


Fig. 5: The robustness of the  $SOMCV$  and  $SOMCV_s$  models to the additive noise: the first row shows, from left to right, the image of Fig. 3(a) with the addition of different Gaussian noise levels (standard deviation  $SD = 10, 15, 20,$  and  $25$ , respectively); the second and third rows show, respectively, the corresponding segmentation results of  $SOMCV$  and  $SOMCV_s$ .

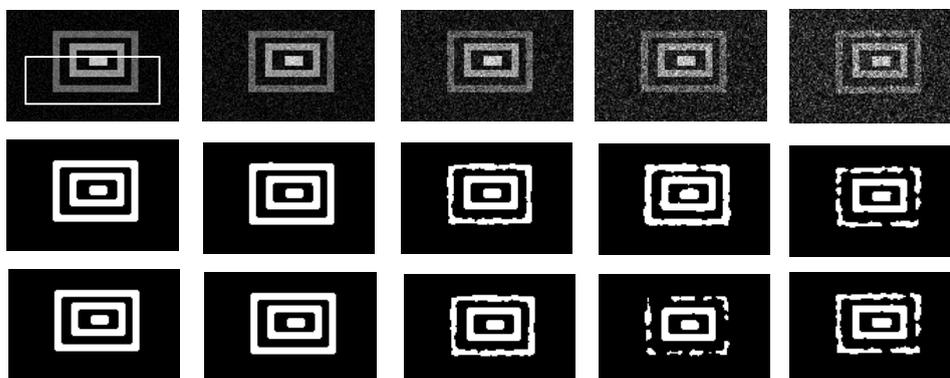


Fig. 6: The robustness of the  $SOMCV$  and  $SOMCV_s$  models to the additive noise: the first row shows, from left to right, the image of Fig. 3(b) with the addition of different Gaussian noise levels (standard deviation  $SD = 10, 20, 30, 40,$  and  $50$ , respectively); the second and third rows show, respectively, the corresponding segmentation results of  $SOMCV$  and  $SOMCV_s$ .

To illustrate the effectiveness of  $SOMCV$  and its variation  $SOMCV_s$  in handling real and synthetic vector-valued images, we have tested the extension of  $SOMCV$  and  $SOMCV_s$  to the vectorial framework on  $RGB$  real and synthetic images, which is shown in Fig. 8 in comparison with the vectorial  $C-V$  model from [15]. The segmentation results of  $SOMCV$  are similar to the ones of  $C-V$  in handling the image shown in the fourth column, while  $SOMCV$  outperforms  $C-V$  in all the other shown images. For these images,  $SOMCV$  outperforms also  $SOMCV_s$ , which, however, provides better results than  $C-V$ , apart from the cases of the images considered in the first two columns, for which the results are similar.

In the following, we provide also a quantitative study to confirm the effectiveness of  $SOMCV$  and  $SOMCV_s$ , when compared to  $C-V$ . To demonstrate quantitatively the accuracy of the  $SOMCV$  and  $SOMCV_s$  models in segmenting the images shown in Fig. 7 and 8, we have also compared the obtained segmentation results with their corresponding ground-truth data by adopting the Precision ( $P$ ), Recall ( $R$ ), and  $F$ -measure metrics. They are defined as follows:

$$\text{Precision}(P) := \frac{TP}{TP + FP}, \quad (30)$$

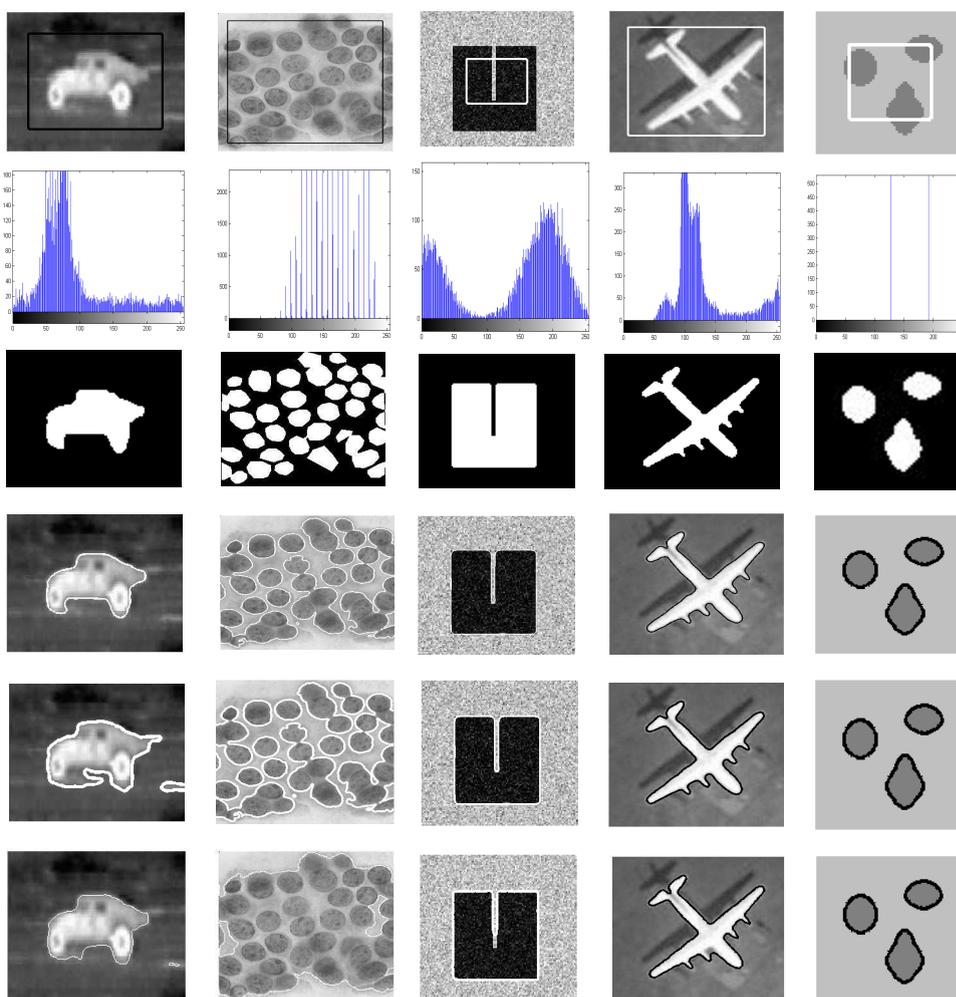


Fig. 7: The segmentation results obtained on real and synthetic scalar-valued images. The first, second and third row show the original images with the initial contours, the histograms of the image intensities and their ground truth, respectively, while the fourth, fifth, and sixth rows show, respectively, the corresponding segmentation results of the *SOMCV*, *SOMCV<sub>s</sub>* and *C-V* models.

$$\text{Recall}(R) := \frac{TP}{TP + FN}, \quad (31)$$

$$F\text{-measure}(F\text{-m}) := \frac{2PR}{P + R}, \quad (32)$$

where  $TP$ ,  $FP$ , and  $FN$  represent, respectively, the numbers of true positive, false positive, and false negative foreground pixels. Precision and Recall are sensitive to the amount of over-segmentation and under-segmentation, respectively, in the sense that over-segmentation is associated with a small Precision score, whereas under-segmentation leads to a small Recall score. Finally, the  $F$ -measure quantifies the overall accuracy.

Tables 1 and 2 illustrate the high segmentation accuracy of the *SOMCV* model and its variation *SOMCV<sub>s</sub>* when compared to the *C-V* model, in terms of the three metrics defined above. As the two tables illustrate, the *SOMCV* model has shown a better performance than the *C-V* model in both the scalar and vectorial cases and for all the tested images used, respectively, in Fig. 7 and 8. Moreover, the *SOMCV<sub>s</sub>* model has usually shown a similar performance as the *SOMCV* model.

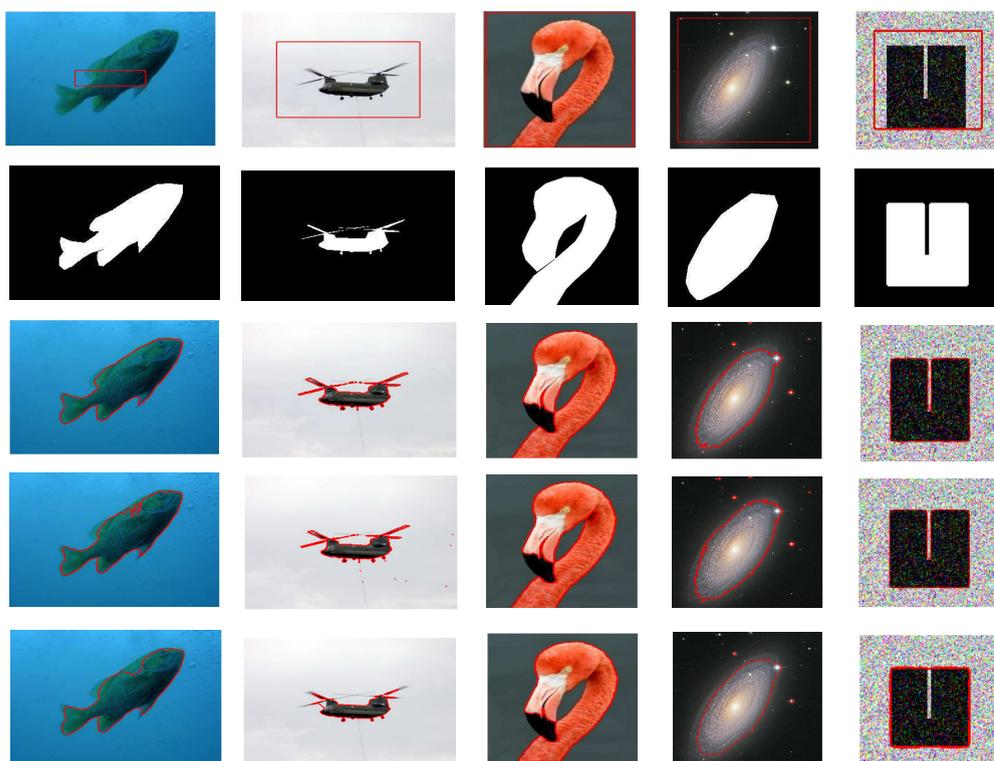


Fig. 8: The segmentation results on real images from [33, 34], and synthetic vector-valued images. The first and second rows show the original images with the initial contours, respectively, while the third, fourth, and fifth rows show, respectively, the corresponding segmentation results of the vectorial versions of the *SOMCV*, *SOMCV<sub>s</sub>* and *C-V* models. Note that  $\sigma = .5$  has been used by *SOMCV* and *SOMCV<sub>s</sub>* for the image in the second column.

Table 1: The Precision, Recall, and *F*-measure metrics for the scalar *SOMCV*, *SOMCV<sub>s</sub>* and *C-V* models in the segmentation of the scalar images shown in Fig. 7.

Image in	<i>SOMCV</i>			<i>SOMCV<sub>s</sub></i>			<i>C-V</i>		
	<i>P</i> (%)	<i>R</i> (%)	<i>F</i> -m (%)	<i>P</i> (%)	<i>R</i> (%)	<i>F</i> -m (%)	<i>P</i> (%)	<i>R</i> (%)	<i>F</i> -m (%)
column 1	98.8	99.9	99.3	75.5	100	86	91.8	83.3	87.4
column 2	60.6	98.5	75	60.6	98.5	75	42.7	98.5	59.6
column 3	100	100	100	100	100	100	99.2	88	93.3
column 4	96.3	99.3	97.8	98.8	98.4	98.6	96.5	96.4	96.4
column 5	100	100	100	100	100	100	99	100	99.5

Table 2: The Precision, Recall, and *F*-measure metrics for the vectorial *SOMCV*, *SOMCV<sub>s</sub>* and *C-V* models in the segmentation of the *RGB* images shown in Fig. 8.

Image in	<i>SOMCV</i>			<i>SOMCV<sub>s</sub></i>			<i>C-V</i>		
	<i>P</i> (%)	<i>R</i> (%)	<i>F</i> -m. (%)	<i>P</i> (%)	<i>R</i> (%)	<i>F</i> -m. (%)	<i>P</i> (%)	<i>R</i> (%)	<i>F</i> -m. (%)
column 1	89.6	96.8	93	91.3	91.8	91.5	94.7	83.1	88.5
column 2	71.7	97.6	82.7	72.3	97.3	82.9	84.5	81.9	83.2
column 3	94.4	90.1	92.2	95	89	91.9	89.5	88.9	89.2
column 4	96.1	85.5	90.5	93.5	91.7	92.6	96.1	86.9	91.3
column 5	99.6	100	99.8	100	100	100	96.8	89.6	93.1

To demonstrate the computational efficiency of the *SOMCV* and *SOMCV<sub>s</sub>* models when compared to the *C-V* model, Table 3 shows, for each of the three methods, the *CPU* time (in seconds) that was required for the contour evolution (i.e., the time required in the testing session) and the number of iterations performed before convergence for the real and synthetic images used in Fig. 7.

Moreover, the computational effectiveness of the vectorial versions of  $SOMCV$  and  $SOMCV_s$  with respect to the vectorial  $C-V$  model is illustrated in Table 4 for the  $RGB$  images in Fig. 8 by showing, for all methods, the  $CPU$  times and the number of iterations required in the testing session (note that, in the common training session of  $SOMCV$  and  $SOMCV_s$ , the  $CPU$  time is fixed by the number of iterations  $t_{\max}^{(tr)}$ ). The sizes of the training and test scalar-valued and vector-valued images are also listed in the two tables. From these tables, we can observe that the  $SOMCV$  and  $SOMCV_s$  models were much faster than the  $C-V$  model in all the listed cases, as the contour evolution for  $SOMCV$  and  $SOMCV_s$  required less iterations to converge than for the  $C-V$  model, and also the computational time per iteration for the  $SOMCV$  and  $SOMCV_s$  models was smaller than the one for the  $C-V$  model. This is due to the fact that  $SOMCV$  and  $SOMCV_s$  models are Gaussian Regularizing Level Set Models, whereas the original  $C-V$  model has not this feature.

Concluding, the results shown in Tables 1-4 highlight several advantages of the  $SOMCV$  and  $SOMCV_s$  models with respect to the  $C-V$  model.

Table 3: The contour evolution time and number of iterations required by the  $SOMCV$ ,  $SOMCV_s$ , and  $C-V$  models to segment the foreground for the scalar-valued images shown in Fig. 7.

Image in	Image size	$SOM$ topology	$SOMCV$		$SOMCV_s$		$C-V$	
			$CPU$ Time (s)	# Iter.	$CPU$ Time (s)	# Iter.	$CPU$ Time (s)	# Iter.
column 1	$118 \times 93$	$5 \times 1$	0.03	10	0.01	9	6.22	137
column 2	$256 \times 256$	$5 \times 1$	1.0	30	0.73	30	104.2	406
column 3	$114 \times 101$	$5 \times 1$	0.14	16	0.1	16	5.6	100
column 4	$135 \times 125$	$5 \times 1$	0.15	16	0.15	16	13.1	266
column 5	$64 \times 61$	$5 \times 1$	0.03	7	.01	7	4.38	97

Table 4: The contour evolution time and number of iterations required by the  $SOMCV$ ,  $SOMCV_s$ , and  $C-V$  models to segment the foreground for the vector-valued images shown in Fig. 8.

Image in	Image size	$SOM$ topology	$SOMCV$		$SOMCV_s$		$C-V$	
			$CPU$ Time (s)	# Iter.	$CPU$ Time (s)	# Iter.	$CPU$ Time (s)	# Iter.
column 1	$300 \times 225$	$2 \times 2$	1.4	20	1.2	20	43	356
column 2	$300 \times 225$	$3 \times 3$	3.08	37	2.57	37	46	400
column 3	$300 \times 451$	$3 \times 3$	14.35	80	12.19	80	588.4	551
column 4	$272 \times 297$	$3 \times 3$	2.9	25	2.79	27	212.3	1612
column 5	$114 \times 101$	$2 \times 2$	.9	16	.4	16	4.6	78

In the following experiments, for a fair comparison, we compare the behavior of our  $SOMCV$  model, as a variational level set-based  $SOM$ -based  $ACM$ , with the  $CSOMCV$  model [11] and the Local-Global  $SOM$ -based  $ACM$  from [24], as related state-of-the-art variational level set-based  $SOM$ -based  $ACMs$ . Moreover, we also include in the comparison the  $SOM$ -based Hierarchical Agglomerative Clustering ( $SOM-HAC$ ) model from [35], as a  $SOM$ -based - but not variational level set-based - image segmentation model.  $SOM-HAC$  relies on local features (including, for each pixel, its local mean intensity and standard deviation), which are used in a first stage as inputs to train a  $SOM$ . Then, in a second phase, it makes use of hierarchical agglomerative clustering to perform an additional clustering process of the output prototypes of the  $SOM$ , hence producing the final segmentation.

Fig. 9 shows, for some of the scalar-valued images presented in the paper, a comparison of the segmentation results obtained by the proposed  $SOMCV$  model, the  $CSOMCV$  model (as a supervised global  $SOM$ -based

*ACM*), and the *SOM-HAC* model (as a local *SOM*-based segmentation model). Fig. 11 also shows the choice of the supervised pixels for the *CSOMCV* model. As illustrated in Fig. 9, *SOMCV* has shown a significant performance in segmenting such images, often outperforming such state-of-the-art *SOM*-based models and, in the other cases, showing at least similar results. Moreover, both its effectiveness and efficiency have been confirmed by the quantitative results reported in Tables 5 and 6, respectively. As compared to the *CPU* time of *SOMCV*, the one of *CSOMCV* is sometimes slightly smaller, since *CSOMCV* uses a scalar value to represent an intensity distribution, while the proposed *SOMCV* model uses a set of descriptors for its representation. However, this slightly larger computational effort makes it possible to obtain better final segmentations, despite being *SOMCV* an unsupervised model.

Finally, Table 7 reports the *CPU* time and the associated number of iterations required by the unsupervised *SOM*-based *ACM* presented in [24] (as a Local-Global *SOM*-based *ACM*) to produce, on the last three images of Fig. 7, similar segmentation results as the *SOMCV* and *SOMCV<sub>s</sub>* models (such segmentations are shown in Fig. 10). The table highlights the larger efficiency of the *SOMCV* and *SOMCV<sub>s</sub>* models when compared to the model from [24]. The larger *CPU* times of the latter are due to the fact that it uses both local and global information in each iteration, during the evolution of its active contour.

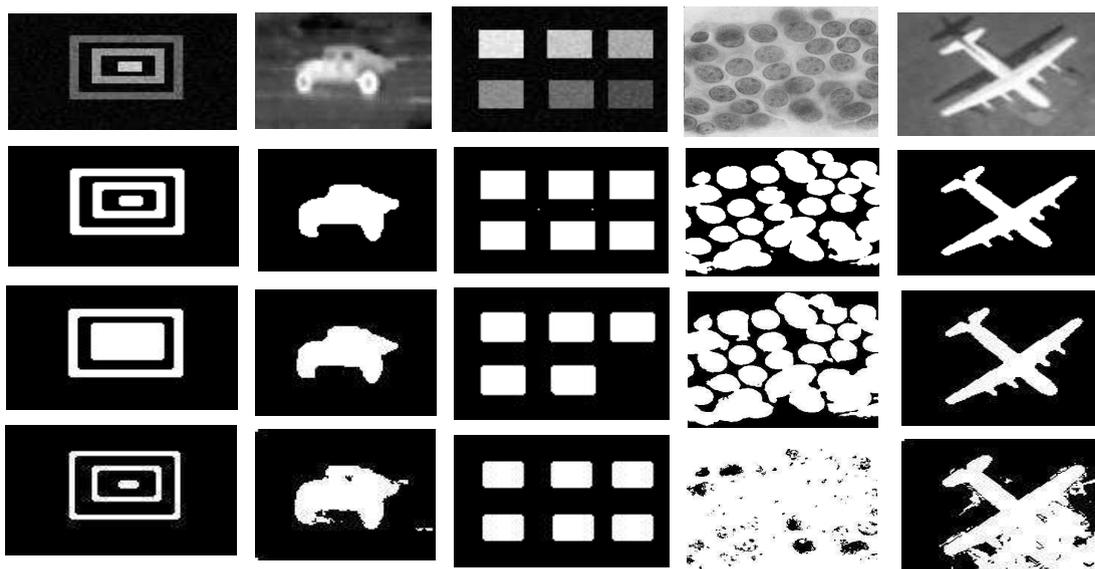


Fig. 9: The binary segmentation results of our proposed *SOMCV* model, as compared to the *CSOMCV* and *SOM-HAC* models. The first row shows the original images. The second row shows the segmentation results, corresponding to the images in the first row, obtained by the *SOMCV* model. The third and fourth rows show the corresponding binary segmentation results obtained by the *CSOMCV* and *SOM-HAC* models, respectively.

In order to compare our *SOMCV* model with some representative global pixel-based segmentation techniques, we have applied the Otsu's method [36] and the multi-threshold Otsu's method [37] to some of the scalar-valued images considered in this paper. Such methods belong to the class of thresholding image segmentation methods, as they segment a scalar-valued image by comparing the pixel intensity with one or multiple thresholds, respectively. The main reason for selecting the Otsu's method is that its threshold is chosen in such a way to optimize a trade-off between the maximization of the inter-class variance (i.e., between pairs of pixels belonging to



Fig. 10: The segmentation results obtained by the unsupervised *SOM*-based *ACM* proposed in [24] on some of the images presented in Fig. 7.

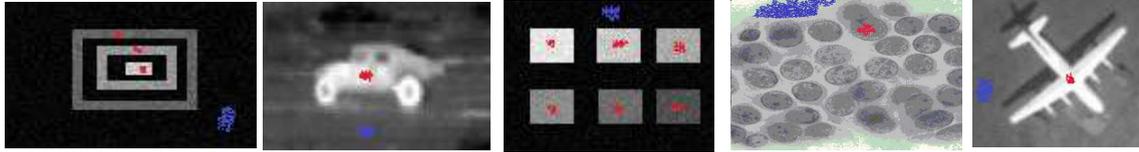


Fig. 11: The training images used by the *CSOMCV* model together with the supervised foreground pixels (red) and the supervised background pixels (blue) used in the training sessions of the model.

Table 5: The Precision, Recall, and *F*-measure metrics for the *CSOMCV* and *SOM-HAC* methods in the segmentation of the images shown in Fig. 9, as compared to the *SOMCV* model.

Image in	<i>SOMCV</i>			<i>CSOMCV</i>			<i>SOM-HAC</i>		
	<i>P</i> (%)	<i>R</i> (%)	<i>F</i> -m (%)	<i>P</i> (%)	<i>R</i> (%)	<i>F</i> -m (%)	<i>P</i> (%)	<i>R</i> (%)	<i>F</i> -m (%)
column 1	100	84.4	91.5	77.4	69	73	100	39.6	56.7
column 2	98.8	99.9	99.3	100	78	87	89.9	80.7	85.1
column 3	100	90.5	95	100	74.6	85.4	100	100	100
column 4	60.6	98.5	75	92	86.6	89.2	45.2	69.4	54.8
column 5	96.3	99.3	97.8	78.3	81.3	79.8	24.3	78.1	37

Table 6: The contour evolution time and number of iterations required by the *SOMCV* model as compared to the *CSOMCV* model to segment the foreground for some of the scalar-valued images shown in Fig. 9, in addition to the convergence time required by the *SOM-HAC* model, for the same images.

Image in	<i>SOMCV</i>		<i>CSOMCV</i>		<i>SOM-HAC</i>
	<i>CPU</i> Time (s)	# Iterations	<i>CPU</i> Time (s)	# Iterations	<i>CPU</i> Time (s)
column 1	0.03	10	.03	8	2.65
column 2	0.05	10	0.04	10	3.9
column 3	0.35	14	0.3	12	3.1
column 4	4.06	32	2.1	32	9.25
column 5	0.92	17	0.32	16	5.5

Table 7: The contour evolution time and number of iterations required by the unsupervised *ACM* proposed in [24] to segment the foreground for some of the scalar-valued images shown in Fig. 7, as compared to the contour evolution time and number of iterations required by the *SOMCV* and *SOMCV<sub>s</sub>* models, for the same images.

Image in	<i>SOMCV</i>		<i>SOMCV<sub>s</sub></i>		Model in [24]	
	<i>CPU</i> Time (s)	# Iterations	<i>CPU</i> Time (s)	# Iterations	<i>CPU</i> Time (s)	# Iterations
column 3	0.14	16	0.1	16	12.15	30
column 4	0.15	16	0.15	16	15.19	16
column 5	0.03	7	.01	7	.45	7

the foreground and the background, respectively) and the minimization of the intra-class variance (i.e., between pairs of pixels belonging to the same region). The multi-threshold the Otsu's method is similar but uses more thresholds, segmenting the image in more than 2 regions. Fig. 12 shows the segmentation results obtained by the Otsu's method (second row) and the multi-threshold Otsu's method (third row) on some of the scalar-valued images considered in this paper. For a fair comparison, in the case of multi-threshold Otsu's method we have also

merged some of the objects found for different numbers of thresholds (as shown in the fourth row), then we have applied the classical Otsu’s method to the resulting image (fifth row). As illustrated by Fig. 12, the Otsu’s and multi-threshold Otsu’s methods demonstrated to be more sensitive to noise than our proposed *SOMCV* model. As an additional drawback, post-processing operations were also required for the multi-threshold Otsu’s method. The quantitative results corresponding to Fig. 12 are reported in Table 8.

Table 8: The Precision, Recall, and  $F$ -measure metrics for the Otsu’s method and the multi-threshold Otsu’s method (with post-processing) in the segmentation of the images shown in Fig. 12 (second and fifth rows, respectively) compared to the *SOMCV* model (sixth row).

Image in	Otsu’s method			Multi-threshold Otsu’s method			<i>SOMCV</i>		
	$P$ (%)	$R$ (%)	$F$ -m (%)	$P$ (%)	$R$ (%)	$F$ -m (%)	$P$ (%)	$R$ (%)	$F$ -m (%)
column 1	97.7	98	97.8	100	64.3	78.3	98.8	99.9	99.3
column 2	100	78.8	88.1	100	55.2	71.1	100	90.5	95
column 3	94.4	84.1	89	98.7	52.3	68.4	100	84.4	91.5

Finally, we have trained the neural map on a single frame of a real aircraft video [38] (the top left image in Fig. 13(a)) and applied the trained network on-line to segment individually - using *SOMCV* - some of its *RGB*-frames, which are shown in Fig. 13(a) (the initial contours for the video frames are similar to the initial contour - shown in red - which has been used for the first image). Fig. 13(b) shows the segmentation results of *SOMCV* in handling the selected frames in Fig. 13(a) and demonstrates its robustness to scene changes and object motions. Concluding, this experiment highlights the robustness of *SOMCV* model to the contour initialization, scene changes and illumination variations when being used in an on-line framework.

## 6 Discussion

Unsupervised global *ACMs* are powerful segmentation techniques which are able to segment images in an unsupervised global way by dealing with the segmentation process as an optimization problem. However, a limitation of existing unsupervised global *ACMs* is in the statistical assumptions made on the image intensity distribution.

Motivated by the above observation, we have proposed a novel unsupervised global *ACM*, termed *SOM-based Chan-Vese (SOMCV)*. The *SOMCV* model is a global and an unsupervised *ACM* that integrates effectively the advantages of *ACMs* and self-organizing networks. *SOMCV* has a *Self-Organizing Topology Preservation (SOTP)* property, which allows to preserve the topological structures of the foreground/background intensity distributions during the active contour evolution. Indeed, *SOMCV* relies on a set of self-organized neurons by automatically extracting the prototypes of selected neurons as global regional descriptors and iteratively, in an unsupervised way, integrates them during the evolution of the contour.

In order to highlight the robustness of *SOMCV*, several synthetic and real images with different kinds of intensity distributions have been handled effectively in the experimental studies presented in Section 5. Also the variation of *SOMCV* - the *SOMCV<sub>s</sub>* model - has provided good results in most cases. The capability of *SOMCV* and *SOMCV<sub>s</sub>* to handle images globally without relying on a particular statistical assumption is the main contribution of this paper. Moreover, the effectiveness and robustness of *SOMCV* and *SOMCV<sub>s</sub>* may find applications in various other problems in computer vision.

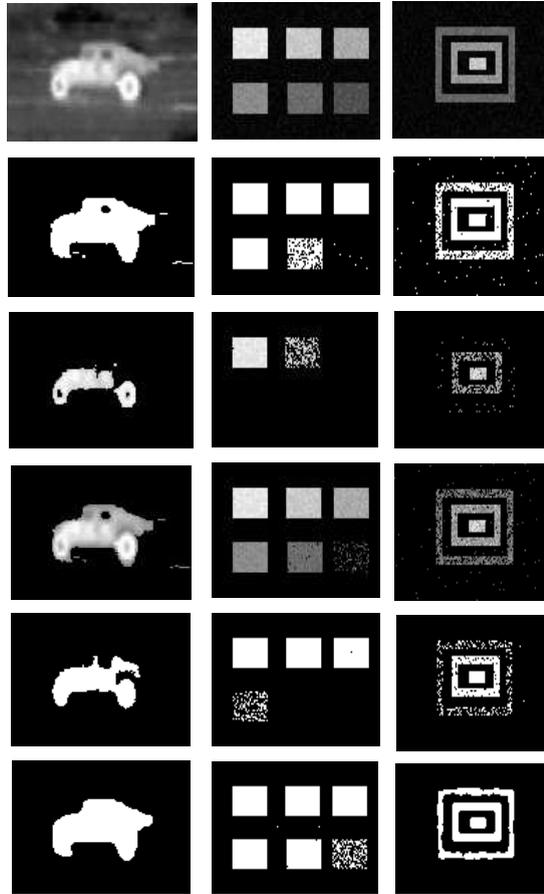


Fig. 12: The segmentation results of the Otsu's and the multi-threshold Otsu's methods on some of the scalar-valued images considered in this paper. The first row shows the original images. The second row shows the segmentation results, corresponding to the images of the first row, obtained by the Otsu's method. The third row shows the object of interests obtained by the multi-threshold Otsu's method when the number of thresholds is five. The fourth row shows the merged objects obtained by first applying the multi-Otsu's method when the number of thresholds is 2, 3, 4, and 5, then merging some of the obtained objects. The fifth row shows the segmentation results of the Otsu's method applied on the images of the fourth row. Finally, the sixth row shows the segmentation results obtained by *SOMCV* on the images of the first row.

The following are some possible directions for future developments. The first one consists in extending the *SOMCV* and *SOMCV<sub>s</sub>* models such that the underlying neurons are incrementally added/removed and trained to overcome the limitation of manually adapting the topology of the network. Moreover, one may apply suitable tools (e.g., genetic algorithms) to identify the best *SOM* topology, thus reducing the number of parameters to be tuned manually. Furthermore, one may use both local and global information to enable *SOMCV* and *SOMCV<sub>s</sub>* to handle, still in a robust and unsupervised way, images with larger intensity inhomogeneity of the foreground/background and more complex intensity distributions.

### Compliance with Ethical Standards

Conflict of Interest: The authors declare that they have no conflict of interest.

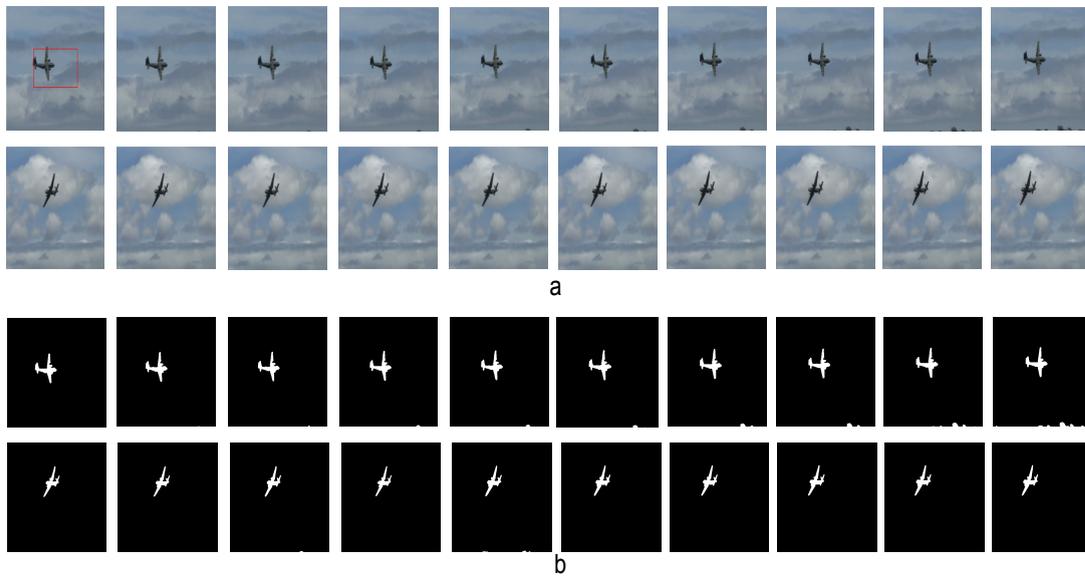


Fig. 13: The robustness of the *SOMCV* model to scene changes and moving objects. (a) The first row shows the original early frames (frames 50-59, from left to right) of a real-aircraft video while later frames (frames 350-359, from left to right) are shown in the second row. (b) shows the segmentation results obtained by *SOMCV*, on the frames shown in part (a).

## References

1. Y. V. Venkatesh, S. Kumar Raja, N. Ramya, A novel SOM-based approach for active contour modeling, in: Proceedings of the 2004 Conference on Intelligent Sensors, Sensor Networks and Information Processing, 2004, pp. 229–234.
2. C. Xu, J. L. Prince, Snakes, shapes, and gradient vector flow, *IEEE Transactions on Image Processing* 7 (3) (1998) 359–369.
3. T. F. Chan, L. A. Vese, Active contours without edges, *IEEE Transactions on Image Processing* 10 (2) (2001) 266–277.
4. M. F. Talu, ORACM: Online region-based active contour model, *Expert Systems with Applications* 40 (16) (2013) 6233 – 6240.
5. M. M. Abdelsamea, S. A. Tsafaris, Active contour model driven by globally signed region pressure force, in: Proceedings of the 18th International Conference On Digital Signal Processing, 2013.
6. S. Chen, R. J. Radke, Level set segmentation with both shape and intensity priors, in: Proceedings of the 12th IEEE International Conference on Computer Vision, 2009, pp. 763–770.
7. T. Kohonen, Essentials of the self-organizing map, *Neural Networks* 37 (2013) 52–65.
8. S. R. Vantaram, E. Saber, Survey of contemporary trends in color image segmentation, *Journal of Electronic Imaging* 21 (4), article ID: 040901, 28 pages, 2012.
9. S. Skakun, A neural network approach to flood mapping using satellite imagery, *Computing and Informatics* 29 (6) (2010) 1013–1024.
10. M. M. Abdelsamea, G. Gnecco, M. M. Gaber, A survey of SOM-based active contours for image segmentation, in: Proceedings of the 10th Workshop on Self-Organizing Maps (WSOM 2014), *Advances in Intelligent Systems and Computing*, Vol. 295, Springer, 2014, pp. 293–302.
11. M. M. Abdelsamea, G. Gnecco, M. M. Gaber, A concurrent SOM-based Chan-Vese model for image segmentation, in: Proceedings of the 10th Workshop on Self-Organizing Maps (WSOM 2014), *Advances in Intelligent Systems and Computing*, Vol. 295, Springer, 2014, pp. 199–208.
12. M. M. Abdelsamea, G. Gnecco, M. M. Gaber, An efficient self-organizing active contour model for image segmentation, *Neurocomputing* 149, Part B (2015) 820–835.
13. D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, *Communications on Pure and Applied Mathematics* 42 (5) (1989) 577–685.
14. H.-K. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, *Journal of Computational Physics* 127 (1) (1996) 179–195.

15. T. F. Chan, Sandberg, L. A. Vese, Active contours without edges for vector-valued images, *Journal of Visual Communication and Image Representation* 11 (2) (2000) 130–141.
16. M. E. Leventon, W. E. L. Grimson, O. Faugeras, Statistical shape influence in geodesic active contours, in: *Proceedings of the 5th IEEE EMBS International Summer School on Biomedical Imaging*, 2002, pp. 316 – 323.
17. A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, A. Willsky, A shape-based approach to the segmentation of medical imagery using level sets, *IEEE Transactions on Medical Imaging* 22 (2) (2003) 137–154.
18. M. M. Abdelsamea, G. Gnecco, M. M. Gaber, E. Elyan, On the relationship between variational level set-based and som-based active contours, *Computational Intelligence and Neuroscience* 2015, article ID, 109029, 19 pages.
19. Y. V. Venkatesh, N. Rishikesh, Self-organizing neural networks based on spatial isomorphism for active contour modeling, *Pattern Recognition* 33 (7) (2000) 1239–1250.
20. H. Shah-Hosseini, R. Safabakhsh, A TASOM-based algorithm for active contour modeling, *Pattern Recognition Letters* 24 (910) (2003) 1361 – 1373.
21. Y. V. Venkatesh, S. Kumar Raja, N. Ramya, Multiple contour extraction from graylevel images using an artificial neural network, *IEEE Transactions on Image Processing* 15 (4) (2006) 892–899.
22. G. S. Vasconcelos, C. A. C. M. Bastos, I. R. Tsang, G. D. C. Cavalcanti, BSOM network for pupil segmentation., in: *Proceedings of the 2011 IEEE International Joint Conference on Neural Networks (IJCNN)*, 2011, pp. 2704–2709.
23. C. A. C. M. Bastos, I. R. Tsang, G. S. Vasconcelos, G. D. C. Cavalcanti, Pupil segmentation using pulling & pushing and BSOM neural network, in: *Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2012, pp. 2359–2364.
24. M. M. Abdelsamea, G. Gnecco, Robust local-global SOM-based ACM, *Electronics Letters* 51 (2) (2015) 142–143.
25. J. Vesanto, J. Himberg, E. Alhoniemi, J. Parhankangas, Self-organizing map in MATLAB: the SOM toolbox, in: *In Proceedings of the Matlab DSP Conference*, 2000, pp. 35–40.
26. S. Haykin, *Neural Networks and Learning Machines*, 3rd Edition, Pearson Prentice Hall, 2008.
27. K. Zhang, H. Song, L. Zhang, Active contours driven by local image fitting energy, *Pattern Recognition* 43 (4) (2010) 1199–1206.
28. K. Zhang, L. Zhang, H. Song, D. Zhang, Reinitialization-free level set evolution via reaction diffusion, *IEEE Transactions on Image Processing* 22 (1) (2013) 258–271.
29. K. Zhang, L. Zhang, H. Song, W. Zhou, Active contours with selective local or global segmentation: a new formulation and level set method, *Image and Vision Computing* 28 (4) (2010) 668–676.
30. K. Zang, S. Xu, W. Zhou, B. Liu, Active contours based on image Laplacian fitting energy, *Chinese Journal of Electronics* 18 (2) (2009) 281–284.
31. C. Li, R. Huang, Z. Ding, J. Gatenby, D. Metaxas, J. Gore, A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI, *IEEE Transactions on Image Processing* 20 (7) (2011) 2007–2016.
32. C. Li, C. Kao, J. Gore, Z. Ding, Minimization of region-scalable fitting energy for image segmentation, *IEEE Transactions on Image Processing* 17 (10) (2008) 1940–1949.
33. <http://www.wisdom.weizmann.ac.il/~vision>.
34. S. Alpert, M. Galun, R. Basri, A. Brandt, Image segmentation by probabilistic bottom-up aggregation and cue integration, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
35. C. Dongxiang, C. Weizhong, A hybrid clustering method for automatic medical image segmentation, *Journal of Computational Information Systems* 6 (6) (2010) 1983–1993.
36. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man and Cybernetics* 9 (1) (1979) 62–66.
37. P.-S. Liao, T.-S. Chen, P.-C. Chung, A fast algorithm for multilevel thresholding, *Journal of Information Science and Engineering* 17 (5) (2001) 713–727.
38. A. S. Mian, Realtime visual tracking of aircrafts, in: *Proceedings of the 2008 IEEE International Conference on Digital Image Computing: Techniques and Applications*, 2008, pp. 351–356.