

RESEARCH

Open Access



# Cage Active Contours for image warping and morphing

Jeroni Carandell, Luis Garrido and Laura Igual\* 

## Abstract

Cage Active Contours (CACs) have shown to be a framework for segmenting connected objects using a new class of parametric region-based active contours. The CAC approach deforms the contour locally by moving cage's points through affine transformations. The method has shown good performance for image segmentation, but other applications have not been studied. In this paper, we extend the method with new energy functions based on Gaussian mixture models to capture multiple color components per region and extend their applicability to RGB color space. In addition, we provide an extended mathematical formalization of the CAC framework with the purpose of showing its good properties for segmentation, warping, and morphing. Thus, we propose a multiple-step combined method for segmenting images, warping the correspondences of the object cage points, and morphing the objects to create new images. For validation, both quantitative and qualitative tests are used on different datasets. The results show that the new energies produce improvements over the previously developed energies for the CAC. Moreover, we provide examples of the application of the CAC in image segmentation, warping, and morphing supported by our theoretical conclusions.

**Keywords:** Parametrized active contours, Cage Active Contour, Deformable models, Mean value coordinates, Image morphing, Shape similarity

## 1 Introduction

Cage Active Contours (CACs), proposed in [18], are a framework for segmenting connected objects using a new class of parametric region-based active contours. The evolving contour is parametrized by an ordered set of control points, using mean value coordinates (a distinct generalization of barycentric coordinates), called a *cage*. The CAC approach deforms the contour locally by moving the cage's points through affine transformations. The cage allows to easily introduce other restrictive criteria (e.g., avoid self-intersections), apart from the already intrinsic properties of the mean value coordinates such as smoothness [17]. The properties of the CAC method allow to easily deal with region-based models which proves to be hugely advantageous with respect to most previous parametrized approaches, which are only able to deal with edge-based energies. As far as we know, except for [7], which treats 3D images, there is almost no work in the

field of parametric-based approaches that is able to deal, in a unified manner, with several region-based models. The CAC approach has proven to be quite versatile, for instance in the domain of medical image segmentation, where the structure to be segmented has often only one regular connected component. However, the *status quo* of the CAC is simple and limited. In previous papers, the considered models are based on quite simple assumptions: a region mean, a Gaussian fitting, and a discrete histogram fitting. Moreover, the approach is restricted to gray-scale images, and only the application of the CAC to object segmentation is evaluated. The uniqueness of the method lies in the physical interpretation of the parameters, i.e., the cage vertices, that control the contour deformation. The method has been previously applied to image segmentation, but we believe that the method can be applied to other applications such shape similarity or image morphing, a topic that has not previously been studied in the context of Cage Active Contours.

In this paper, we present several contributions. First, we enhance the CAC segmentation approach in order to be

\*Correspondence: [ligual@ub.edu](mailto:ligual@ub.edu)

Department of Mathematics and Computer Science, Universitat de Barcelona, Gran Via de les Corts Catalanes, 585, 08007 Barcelona, Spain

able to capture more complex properties of the region to segment. For this issue, we present a Gaussian mixture model-based energy function inspired by the *Gaussian energy function* in [18]. We also generalize this energy function to higher dimensionality, by an extension to RGB, and to multicomponent Gaussian called *multivariate Gaussian mixture energy function*. Finally, we introduce into the model prior information by allowing the user to define hard constraints for segmentation by indicating certain pixels (seeds) that absolutely have to be part of the object and certain pixels that have to be part of the background (in the inner and outer regions), in a similar way to Graph Cuts [10]. As it will be seen in the paper, the advantage of the CAC approach is that introducing these enhancements is straightforward in comparison to other classical approaches.

Second, we propose a method for shape similarity computation. The shape similarity approach derives from the mathematical formalization of the CAC properties. We present the concept of a *family of shapes*, defined by the CAC, and prove that a categorization of these can be made if some initial conditions are met (see definition 10, page 11). As a consequence, the CAC avoids the definition of landmark points for shape description purposes. We highlight the properties of the approach in two different applications: automatic image warping and kmorphing.

Finally, we validate the ability of the CAC framework as a multiple-step method for segmentation, warping, and morphing. Images are first segmented using the CAC, then correspondences among the cage control points of the shapes are estimated, and finally, a morphing between the images is constructed. This process is practically automatic since it only needs to define a seed of the object of interest.

From an experimental point of view, we show the improvement achieved with the new multivariate Gaussian mixture energy function in the CAC and we apply the new CAC for a robust warping and morphing.

Besides, we provide a public Python implementation (with some wrapped functions in C) of the CAC with a variety of energies as well as tools for automatic morphing, warping, and shape description<sup>1</sup>.

The rest of the document goes as follows: In Section 2, we review the related work and set the preliminary concepts of Cage Active Contours. In Section 3.2, we present the proposed improvements to enhance the CAC and extend their definition to RGB space. In Section 3.3, we formalize the shape descriptor based on the CAC. In Section 4, we evaluate the proposed the CAC segmentation improvements. In Section 4.5, we show the applications of the CAC in image morphing and warping. Finally, in Section 5, we discuss our conclusions and future work.

## 2 Related work

### 2.1 Active contours

Active contours [23] are a general method for delineating an object outline that can be fit to tackle the problem of single-connected object and have indeed proven to be a very powerful tool in doing so. Also known as snakes, they are deformable models that consist on evolving an interface which is propagated in order to recover the shape of the object of interest.

The description of the interface sub-categorizes these method into *parametric* and *geometric* approaches. The first approach requires, as the name implies, a set of discrete parameters such as points as seen in [23] or basis functions (a basis for a function space) such as B-splines [20, 34]. The advantage of basis functions is that linear combinations have inherent regularity.

Conversely, geometric active contours, defined as the zero level set of a higher dimensional function, have more topological flexibility because contours can break apart or join without the need of re-parametrization. However, this property can prove to be a double edge-sword when the desired shape has to have a specific topology. Level sets are the most representative technique in this category [32].

The evolution of these interfaces is driven by the minimization of an energy function defined so as to express the properties of the object to be segmented in mathematical terms. In this context, we have to differentiate two types of image features in which these properties are expressed: edge-based, such as the image gradient on the contour as in [14], or region-based terms, as introduced by Chan and Vese in [15]. Region-based terms are known to be more robust to noise than edge-based contours and therefore do not require the initial boundary to be so close to the solution [40]. The work of Chan and Vese is based on evolving the interface according to the variance of the gray-level values of both interior and exterior regions allowing for segmentation of objects with boundaries not defined by gradient to be detected. This approach has been extended, since then, to other features such as the Bayesian model [36] and histogram model [29]. These approaches define the whole inner region of the evolving contour as the interior region and its complement as the exterior. Thus, they may fail if these features are not spatially invariant. In [30], a solution is proposed by considering the features in a band around the evolving contour. Another solution proposed by [24] is to consider the inner and outer regions as those points that are in the intersection of their respective regions and in the ball centered on the contour. In [25], a more context-aware solution is introduced where a kernel function is applied to each point to define a region-scalable fitting term. Finally, two fast algorithms are presented in [9] and [38], where a B-Spline parametrization and a discrete approximation-based representation are presented, respectively.

The Cage Active Contours (CACs) are a type of parametric active contours which are fit to work with region energies similar to the ones defined with the geometric (i.e., level set) methods [18]. Because of the theoretical framework upon which level sets are built, complex steps are required in order to evolve the curve, including the application of Euler-Lagrange to solve for a stationary point [13]. As it is seen in Section 3.2, the CAC allows for discretization of the energy function and the calculation of the gradient through partial derivatives as opposed to using Euler-Lagrange.

### 2.2 Shape similarity

Shape comparison is a rich and vast field of research [1, 8]. For this issue, shape descriptors are usually used. Among the best methods for shape description there is discrete Fourier transforms (DFT), which provides a description of the curvature of a shape [8], that is invariant to translation and uniform change in scale. However, the shape descriptor based on DFT is not invariant to rotation. Another interesting method is the curvature scale space (CSS) shape descriptor. This descriptor provides a representation of a contour which represents the time of inflection or union of pairs of points of the shape as it is progressively smoothed [1]. This descriptor is neither invariant to rotation. Usually, the distance computation algorithm is designed so as to make it robust with respect to this issue.

In order to a shape descriptor be useful for shape similarity computation, some properties are usually required: invariance in translation, rotation, and scale, and that each element in this dataset could be indexed so that fast and effective retrieval and comparison may be applied. The latter properties allow its application to retrieval in a large database of images. Both of latter commented methods, very used in this field [4, 44], provide good solutions to indexing and description [44].

In this paper, we formally demonstrate the usefulness of the CAC representation for shape similarity computation. Our shape representation has interesting properties that makes it a good candidate for shape descriptor. However, we would like to point out that our purpose in this work is not to focus on the CAC representation as a shape descriptor. This issue is left as future work.

### 2.3 Mean value coordinates

As was introduced in [18], Cage Active Contours use mean value coordinates for deformation. Let  $\mathcal{C}$  be the contour or interface that separates the interior region,  $\Omega_1$ , and the exterior region,  $\Omega_2$ . In order to be able to deform the interface  $\mathcal{C}$ , a point  $p$  belonging to  $\Omega_1$  or  $\Omega_2$  is expressed as an *affine combination* of vertices  $v_1, v_2, \dots, v_N$  of a cage. That is,

$$p = \sum_{i=1}^N \varphi_i(p)v_i \tag{1}$$

where  $\varphi_i(p)$  is the corresponding *affine coordinate* of the point  $p$  with respect to the vertex  $v_i$  and  $N$  is the number of vertices.

A variety of approaches have been presented for the computation of  $\varphi_i(p)$ . In deformation applications, we have harmonic coordinates [21], green coordinates [26], or mean value coordinates [17]. The advantage of the latter over the rest include a simple computation and the convenience of being able to parametrize any point of the space, be inside or outside the polygon demonstrated in [19].

Given a set of ordered of a polygon of  $N$  points disposed in an anticlockwise order, the *mean value coordinates* of a point  $p$  with respect to  $V$  are  $\varphi^V(p) = (\varphi_i^V(p)|i \in (1, \dots, N))^2$ .

$$\varphi_i(p) = \begin{cases} \delta_{ij} & \text{if } p = v_j \\ (1-t)\delta_{ij} + t\delta_{i,j+1} & \text{if } p = v_j(1-t) + v_{j+1}t \\ \frac{w_i}{\sum_{j=1}^N w_j} & \text{otherwise} \end{cases} \tag{2}$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{3}$$

$t \in [0, 1]$  and  $p = v_j(1-t) + v_{j+1}t$  represents a point on the edge between  $v_j$  and  $v_{j+1}$ . The weight  $w_i$  is calculated as

$$w_i = \frac{\tan\left(\frac{\alpha_{i-1}}{2}\right) + \tan\left(\frac{\alpha_i}{2}\right)}{\|v_i - p\|} \tag{4}$$

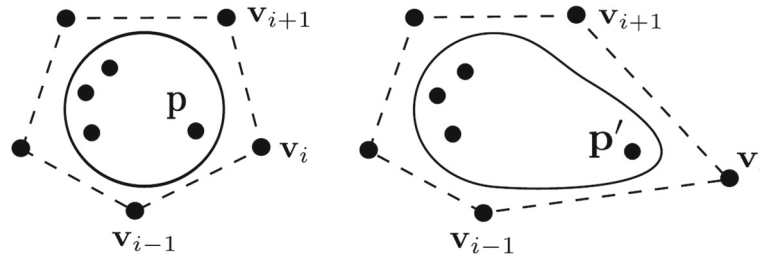
where  $\|v_i - p\|$  is the distance between the vertex  $v_i$  and the considered point  $p$  and  $\alpha_i$  is the *signed angle* of  $[v_i, p, v_{i+1}]$ .

Given the affine coordinates  $\varphi(p)$  of a point  $p$ , the point  $p$  can be recovered with (1). If the vertex  $v_i$  of the cage moves to position  $v'_i$ , the “deformed” point  $p'$  can be recovered as

$$p' = \sum_{i=1}^N \varphi_i(p)v'_i, \tag{5}$$

where note that the point  $p'$  is recovered from the affine coordinates  $\varphi_i(p)$ , see Fig. 1.

Given a set of points, the affine coordinates for each point are computed in an independent way using (2). If a point  $v_i$  of the polygon is stretched in a particular direction, all the points follow the same direction with an associated weight given by  $\varphi_i(p)$  which is inversely proportional to the distance from  $p$  to  $v_i$  since it is the denominator of (4). In Fig. 1, this effect is depicted when point  $v_i$  in the left image is translated to  $v'_i$ . The point  $p$ , near to vertex  $v_i$ , suffers a greater deformation than the



**Fig. 1** Motion of a vertex. Influence of a vertex over the points on the plain (image from [18])

points which are farther where the weight are smaller, and hence, they are barely affected by this deformation.

The following properties are characteristic of the affine coordinates [19]. We enumerate them here since they are necessary for the development of the shape descriptor in Section 3.3:

C.1 *Affine precision:* For any affine function

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}^D, f = \sum_{i=1}^N f(v_i) \varphi_i^V \text{ for } v_i \in V \text{ and where}$$

$\mathbb{R}^D$  is the dimension of the color space.

C.2 *Similarity invariance:* If  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  is a similarity

and for a cage  $V' = f(V)$ , we have that

$$\varphi^V(p) = \varphi^{V'}(f(p))$$

C.3 *Smoothness:*  $\varphi_i$  is  $C^\infty$  everywhere except at the vertices  $v_j$  where it is only  $C^0$ .

C.4 *Edge linearity:*  $\varphi_i^V$  is linear along the edges of the cage  $V$ .

C.5 *Refinability:* If we redefine  $V$  to  $V'$  by splitting an edge between vertices  $v_j$  and  $v_{j+1}$  at

$$v = (1 - t) v_j + t v_{j+1}, \text{ then } \varphi_j^{V'} = \varphi_j^V t + (1 - t) \varphi_{j+1}^V.$$

### 3 Methods

#### 3.1 Cage Active Contour framework

Let us formally define the three major components of a CAC model: an *initial contour*, an *initial cage*, and an *energy function*. We restrict ourselves to the context of  $\mathbb{R}^2$ . Extension to higher dimensions is left as future work.

**Definition 1** A curve on a plane is a continuous mapping  $C : [a, b] \rightarrow \mathbb{R}^2$  such that  $[a, b] \in \mathbb{R}$ .

**Definition 2** A Jordan curve is a non-intersecting, continuous closed curve.

**Definition 3** A contour is used to define the image of a closed curve  $C : [a, b] \rightarrow \mathbb{R}^2$ , such that  $C(a) = C(b)$ .

From now on, however, we use the term curve to mean contour unless it is explicitly distinguished.

The CAC's *initial contour* is a Jordan curve so that by the Jordan Curve Theorem, we can assure that it divides

the plane into two regions  $\Omega_1$  and  $\Omega_2$  which correspond to the interior and the exterior of the curve, respectively.

We define cage as

**Definition 4** A cage is an ordered group of points  $V = (v_1, v_2, \dots, v_N)$  on the plane  $\mathbb{R}^2$ .

By convention, the *initial cage*  $V$  of  $N$  points must define a simple  $N$ -sided polygon since it is a requisite to be able to parametrize points on the plane using *mean value coordinates*<sup>3</sup>. These barycentric coordinates have very good properties which also open the possibility to different applications such as shape descriptors, morphing, warping, and image interpolation in Section 3.3.

The *energy function*  $E$  is a function with respect to a contour; however, since the contour  $C$  is parametrized by a cage  $V$ , and the contours that are able to define depends exclusively on  $V$ , we can define the energy function as

$$E : (\mathbb{R}^2)^N \rightarrow \mathbb{R} \quad (6)$$

$$V \mapsto E(V)$$

The function must be defined in a way so that it is minimum when the object to segment is in the interior region and the background in the exterior. Of course, this idea stems from the assumption that the object differs from that of the background in appearance. The goal is then to minimize the energy function with respect to a cage:

$$\min_{v_1, v_2, \dots, v_N} E(v_1, v_2, \dots, v_N) \quad (7)$$

Since the energy function is in terms of the cage, we can minimize the function by applying gradient descent [31] on the energy function with respect to the control points.

From the very simple models on gray-scale image defined in [18], we can develop more sophisticated energies as more complex properties are taken into consideration.

#### 3.1.1 An example: Gaussian energy function

We next briefly describe only the Gaussian energy function presented in [18] since it will be extended and



improved in the following sections. The input of the system is the image  $I$  to segment and the components of Cage Active Contour: the energy function  $E$ , the vertices  $V$ , and the initial contour  $C$ .

The presented Gaussian energy function assumes a Gaussian distribution of pixel gray-level values in inner and outer regions,  $\Omega_h$  where  $h \in \{1, 2\}$  respectively, and is

$$E_{\text{Gauss}} = \sum_{h=1}^2 \sum_{p \in \Omega_h} -\log(P_h(I(p))) \quad (8)$$

where

$$\log(P_h(I(p))) = -\log\left(\sqrt{2\pi}\sigma_h\right) - \frac{(I(p) - \mu_h)^2}{2\sigma_h^2} \quad (9)$$

and  $P_h$  is the probability an intensity of  $p$ ,  $I(p)$  belongs to the normal distribution defined by region  $h$ 's seed, a sub-sample of points that are representative of the region. The parameters of the Gaussian distribution,  $\sigma_h$  and  $\mu_h$ , are automatically updated at each iteration of the minimization algorithm as is done in [36]. The Gaussian energy function minimization algorithm presented in [18] stops when the parameters of the inner and outer regions,  $\Omega_h$  with  $h \in \{1, 2\}$ , have stable statistics  $\mu_h$  and  $\sigma_h$ . In other words, the curve stops evolving when each region has points whose values have a higher probability of being in that region than otherwise. A more thorough description of the segmentation process can be found in [18].

So far, Cage Active Contours have only been applied to gray-scale images in both 2D [18] and 3D [43] scenarios. That is, the image is a function defined as  $I : \mathbb{R}^D \rightarrow \mathbb{R}$ . The advantage of this type of image lies in the simplicity of having the information in a single value which is also highly interpretable by humans. However, this has two negative consequences: first is that color information is lost, and secondly, since image intensity is directly affected by illumination, methods that rely only on this model are prone to fail under different settings.

On the other hand, observe that the approach also assumes that the Gaussian function only has one component. Extension to multicomponent Gaussian models, for both the interior and exterior regions, may enhance the model.

We thus propose to enhance the Gaussian energy model of [18], see Eq. (8), to a multicomponent model within a RGB color space defined as  $I : \mathbb{R}^D \rightarrow \mathbb{R}^3$  where  $I(p) = (r, g, b)$  for  $p \in \mathbb{R}^D$ . Indeed, the approach presented in the next section is valid for any color space, such as the RGB depth, but due to lack of space, we will focus only on the more simple RGB color space.

### 3.2 Cage Active Contour energy extensions

To define a new energy function, we have to consider which features characterize a good energy function, namely **E.1** Differentiable, **E.2** Few local minima, and **E.3** Little dependence on the starting contour. The energies implemented in [18] can only capture a region's model with a single component, being either the mean value of a region (mean energy function) or a normal distribution of the values (Gaussian energy function), or maximize the difference between distribution of values of each region (histogram energy function), with no regard on prior information on the resulting object to detect. What these energies have in common is that their strategy is to polarize the values in each region. Although this proves to be useful in some cases, it is very limiting when trying to segment objects and background that have multiple Gaussian components. Furthermore, by sampling the model of each region at every iteration, not only it is computationally expensive but also the contour has to rely on a good initialization to capture the description of each region.

#### 3.2.1 Multivariate Gaussian mixture energy function

The proposed energy function attempts to solve these problems by introducing initial information about the object and background through *seeds*. This enhances **E.3** and allows for each region to capture various dominant values inside an image so that in each region, different colors or shades can have a representation proportional to their presence. In order to best capture a model, we need to define a density function which is differentiable in the color space so that we are able to minimize it using gradient descent (**E.1**) and that allows us capture best the distribution of values. With these properties, the Gaussian mixture probability density is a candidate that satisfies both of these criteria since any other continuous (and therefore, all differentiable functions) distributions can be expressed as a mixture of Gaussians given enough components [12, 39]. Moreover, the Gaussian mixture inherits good properties from its normal components, as well as a number of good methods to estimate their parameters, such as the expectation-maximization [42]. However, instead of using directly the Gaussian mixture probability density function, we use its logarithm to smoothen the exponential effect and thus avoid numerical problems during minimization. This approach, commonly used in the literature [2, 22], is also adopted in the Gaussian model defined in [18].

With these criteria, we present the multivariate Gaussian mixture energy function (MGM), which is expressed in the following way:

$$E_{\text{MixtGauss}} = \sum_{h=1}^2 \sum_{p \in \Omega_h} -\log(P_h(I(p))) \quad (10)$$

where  $P_h$  as the Gaussian mixture probability density function of the value of pixel  $p$  to belong to region  $h$ :

$$P_h(I(p)) = \sum_{i=1}^{r_h} \frac{w_i}{2|\Sigma_i|\sqrt{2\pi}} e^{-\frac{(I-p)^T \Sigma_i^{-1} (I-p)}{2}} \quad (11)$$

This probability density function has  $r_h$  normal components, each of which has a mean  $\mu_i$ , a covariance matrix  $\Sigma_i$ , and a weight  $w_i$  such that  $\sum_{i=1}^{r_h} w_i = 1$ , where  $w_i \geq 0$  for  $i \in \{1, 2, \dots, r_h\}$ .

The minimum is reached when a slight movement of the contour implies a loss of pixels in each region whose values have a higher log-likelihood of belonging to the regions' model than the other. The minimum can be obtained by using a gradient descent method. Observe that the gradient has to be computed with respect the control points. The gradient of the energy function is:

$$\nabla_{v_j} E_{\text{MixtGauss}} = \sum_{h=1}^2 \sum_{p \in \Omega_h} -\frac{1}{P_h(I(p))} \nabla_{v_j} P_h(I(p)) \quad (12)$$

where  $P_h(I(p))$  is the Gaussian mixture defined by the seed in region  $h$  which has  $r$  Gaussian components. The gradient is expressed in the following way:

$$\nabla_{v_j} P_h(I(p)) = \sum_{i=1}^{r_h} \left( \frac{w_i e^{-\frac{(I-p)^T \Sigma_i^{-1} (I-p)}{2}}}{\sqrt{|\Sigma_i|} \sqrt{2\pi}} (\mu_i - I(p)) \Sigma_i^{-1} \cdot \nabla I(p) \varphi_j(p) \right) \quad (13)$$

Multicomponent Gaussian has been applied in the context of level sets [6]. However, as commented previously, level-sets require the application of the Euler-Lagrange equations to solve for a stationary point. Once the equations for the stationary point have been obtained, equations are discretized to be able to apply them to an image. As has been seen here, the CAC begins with the discretization of the energy function to be minimized. The stationary point can then be obtained by using a gradient descent method.

### 3.3 Cage Active Contour shape similarity

One of the challenges in shape similarity is that it is often hard to find relevant points in a region that might help to determine structure or orientation of an object that apparently has none. These points are commonly called *landmarks* and are used to build the shape models of an object [16]. In medical imaging, it is often the case that these points are unseen, latent, or that they are difficultly characterized by their shape. Using cage properties to define a shape descriptor can be extremely powerful since they allow to define a similarity measure between different shapes.

To formalize the properties of cage parametrization and describe the advantages in the applications of image morphing and warping and shape descriptors, we first need a way to compare similar contour shapes. Assume we fix an initial regular (or standard) contour and cage configuration. For every new cage obtained by deforming the initial cage, the corresponding initial contour defines a deformed contour shape according to (5). Intuitively, similar cages provide similar contour images under certain initial conditions. Formally, we want to find a criteria which allows us to link an ordered configuration of points (i.e., a cage) with contour shapes so that we may use the existing tools to determine shape similarity between different contours, for cages. The existing tools can be borrowed from polygon similarity, such as the turning function [11], or from point configuration similarity, like Procrustes analysis. The turning function is a distance measure which reflects the difference between two shapes and fulfills the distance properties (identity, symmetry, and triangle inequality), whereas the Procrustes function is not a distance. Furthermore, the turning function is invariant to translation, rotation, or scaling, and this distance has a strong correlation with human intuition [5]. Figure 2 illustrates the turning function performance.

Next, we present the following definitions which lead up to Proposition 1 and its proof.

**Definition 5** (Contour family) *Given an initial contour  $C$  and an initial cage  $V = (v_1, v_2, \dots, v_N)$ , the family of contours  $\mathcal{F}_C^V$  is the set of all the possible contours that can be produced with all cages of  $N$  points by a deformation through (5) and it is expressed as:*

$$\mathcal{F}_{C^V} = \left\{ C^W \mid W \in (\mathbb{R}^2)^N \right\}$$

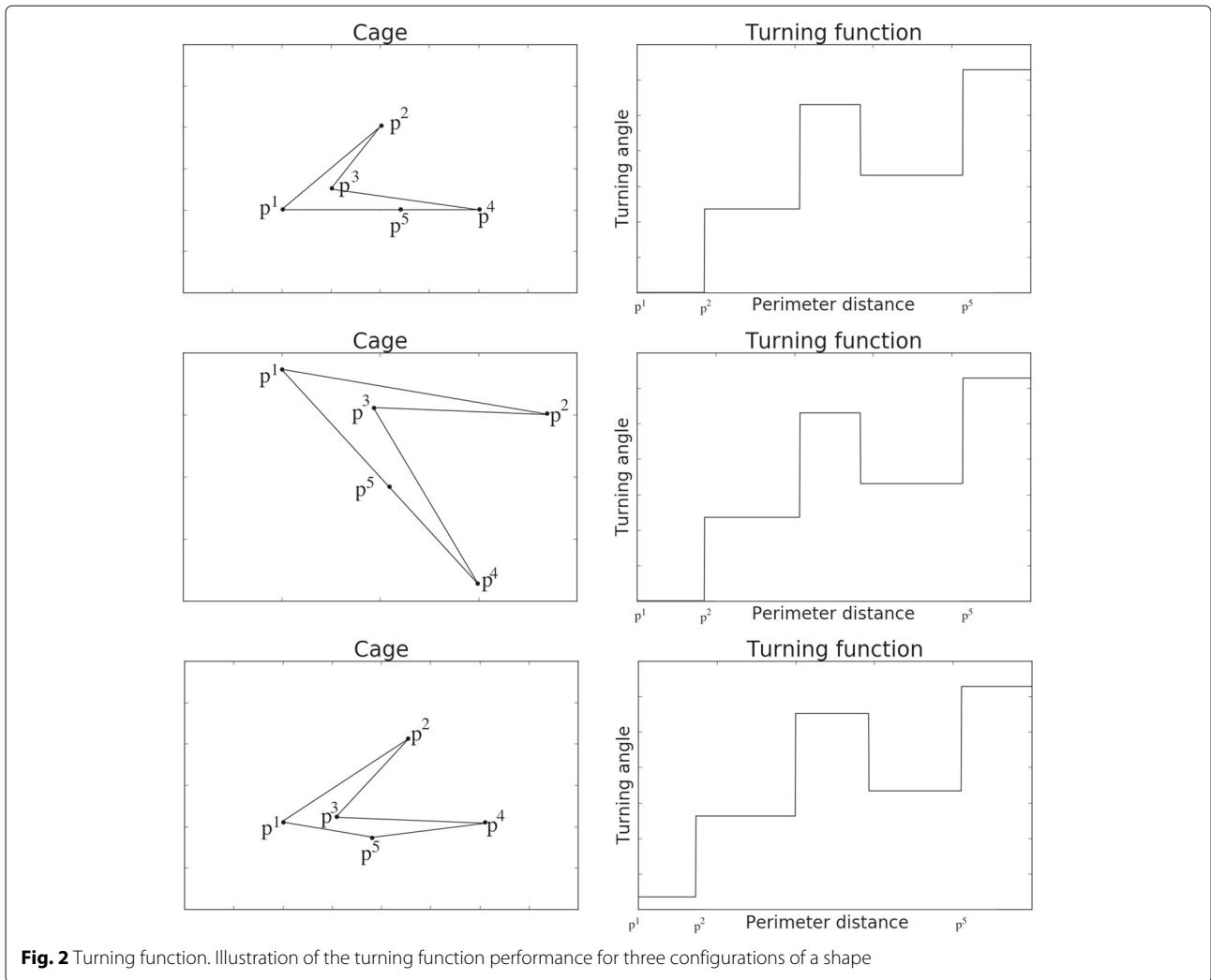
where for any cage  $W \in (\mathbb{R}^2)^N$

$$C^W = \left\{ q \in \mathbb{R}^2 \mid q = \sum_i^N \varphi_i^V(p) w_i, \forall p \in C, W = (w_1, w_2, \dots, w_N) \right\},$$

and  $\varphi_i^V(p)$  are the mean value coordinates of  $p$  with respect to cage  $V$ .  $W$  can be interpreted as a deformation of cage  $V$ .

**Definition 6** (Similarity) *We define a similarity on the plane as an affine transformation  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  composed of rotations, translations, and uniform changes in scale.*

**Definition 7** (Contour similarity) *Two contours are similar if there exists a similarity which maps one to the other.*



**Fig. 2** Turning function. Illustration of the turning function performance for three configurations of a shape

**Definition 8** (Cage similarity) Two cages  $U = (u_1, u_2, \dots, u_N)$  and  $W = (w_1, w_2, \dots, w_N)$  are similar if there exists a similarity function such that  $f(u_i) = w_i$  for each  $i \in \{1, 2, \dots, N\}$ .

**Definition 9** (Shifted cage) A shifted cage of another cage  $W = (w_1, w_2, \dots, w_N)$  is a permutation conserving the order of  $W$ . There are  $N$  shifts (as many as number of points).

$$\begin{aligned}
 W &= W_0 = (w_1, w_2, \dots, w_N) \\
 W_1 &= (w_2, w_3, \dots, w_N, w_1) \\
 &\dots \\
 W_k &= (w_{k+1}, w_{k+2}, \dots, w_k) \\
 &\dots \\
 W_{N-1} &= (w_N, w_1, \dots, w_{N-2}, w_{N-1})
 \end{aligned}$$

In Definition 5, we define the contour family of an initial configuration of a contour  $C$  and a cage  $V$ . However, there are certain properties that we would like to impose on this family. Namely, we are interested in those families where similar cages or similar shifted cages define the same contour. To achieve this property, first, we need a definition.

**Definition 10** A regular initial cage-contour configuration with ratio  $r$  is a set  $(V, C, r)$  consisting of an initial cage  $V = (v_1, v_2, \dots, v_N)$  that defines an  $N$ -sided regular polygon and an initial contour  $C$  that is a circumference concentric to the polygon such that the ratio of the radius of  $C$  and the radius of the polygon is  $r:1$ . For simplicity, we say the ratio is  $r$ .

Having these concepts formally defined, we are able to prove the desired property of the family.

**Proposition 1** Given a regular initial cage-contour configuration  $(V, C, r)$ , then for every contour  $C^W$  and  $C^U$  in the contour family  $F_{V,C}^C$ ,  $C^W$  and  $C^U$  are similar if

- 1  $W$  and  $U$  are similar cages
- or
- 2  $U$  is a shifted cage of a similar cage of  $W$ .

*Proof* The first point is trivial. We want to see if there exists a similarity function  $g$  that sends  $C^W$  to  $C^U$ . So, for every point of  $q^W \in C^W$ , a point  $q^U \in C^U$  has to exist such that  $g(q^U) = q^W$ . By construction of  $C^W$  and  $C^U$ , we know that there exists a point  $p \in C$  such that

$$q^W = \sum_{i=1}^N \varphi_i^V(p)w_i$$

and a point  $p' \in C$  such that

$$q^U = \sum_{i=1}^N \varphi_i^V(p')u_i$$

Since we know that cages  $W$  and  $U$  are similar, we have that, by Definition 8, there exists a similarity  $f$  that maps cage  $U$  to  $W$  (i.e.,  $w_i = f(u_i)$  for all  $i \in \{1, 2, \dots, N\}$ ). It turns out that  $g = f$  and  $p^U = p^W$  define the similarity between contours:

$$q^W = \sum_{i=1}^N \varphi_i^V(p)w_i = \sum_{i=1}^N \varphi_i^V(p)f(u_i) = f\left(\sum_{i=1}^N \varphi_i^V(p)u_i\right) = f(q^U)$$

Therefore, we have that the same similarity that maps  $W$  to  $U$  sends their contours to each other rendering them similar.

To prove the second implication, a more elaborate solution is required. We only need to prove this in the case of  $U$  being the shifted cage of  $W$  since having that, any similar cage would only imply a similarity function. To see that a cage and its shifted cage produces a similar curves, let us take two cages  $W_0 = (w_1, w_2, \dots, w_N)$  and one of its shifted (we take the shift  $k = 1$  for simplicity)  $W_1 = (w_1^1, w_2^1, \dots, w_N^1) = (w_2, w_3, \dots, w_N, w_1)$ .

If we see that their images<sup>4</sup> of  $C$ , respectively  $C^{W_0}$  and  $C^{W_1}$  are congruent, that is  $C^{W_0} = C^{W_1}$ , then they would be similar because the identity function would be the similarity between them.

To see this, we have to see if every point  $q$  in  $C^{W_0}$  is in  $C^{W_1}$ . We have that every point in  $C^{W_0}$  can be expressed as

$$q = \sum_{j=1}^N \varphi_j^V(p)w_j$$

where  $p \in C$  is in the initial contour. If we can find a point  $p_1$  in  $C$  such that

$$q = \sum_{j=1}^N \varphi_j^V(p_1)w_j^1$$

it would do.

The mean value coordinates of a point  $p$  with respect to control point  $v_i$  are calculated using the angles  $\alpha_1$  and  $\alpha_2$  with its neighboring control points  $v_{i-1}$  and  $v_{i+1}$ , respectively. In Fig. 3, we have an example with the circumference contour  $C$  and the cage  $V = (v_1, v_2, \dots, v_N)$  ( $N = 6$  in the image). Point  $p$  has the mean value coordinates  $\varphi^V(p) = (\lambda_1, \lambda_2, \dots, \lambda_N)$ . If we apply a rotation  $R_1$  of  $\alpha_{R_1} = -\frac{2\pi}{N}$  radians and center  $p_c$ . We have that  $R_1(v_i) = v_{i+1}$ , and the rotated point  $p_1 = R_1(p)$  would still be on the contour  $C$ . Furthermore, it would maintain the distance to the rotated control point  $R_1(v_i) = v_{i+1}$ , as well as the angles to their rotated points, because of the property of angle invariance through similarities.

Therefore, we can say that for every point,  $p$ , there exists a point  $p_1 = R_1(p)$  such that, the mean value coordinates are the same but shifted: this can be done for any  $R_k(p) = -\frac{2\pi}{N} k$  for  $k \in \{1, 2, \dots, N\}$ ;

$$\begin{aligned} \varphi^V(R_1(p)) &= (\lambda_2, \lambda_3, \dots, \lambda_N, \lambda_1) \\ \varphi^V(R_2(p)) &= (\lambda_3, \lambda_2, \dots, \lambda_{N-1}, \lambda_2) \\ &\dots \\ \varphi^V(R_k(p)) &= (\lambda_{k+1}, \lambda_{k+2}, \dots, \lambda_{N-k-1}, \lambda_k) \\ &\dots \\ \varphi^V(R_{N-1}(p)) &= (\lambda_N, \lambda_1, \dots, \lambda_{N-2}, \lambda_{N-1}) \end{aligned}$$

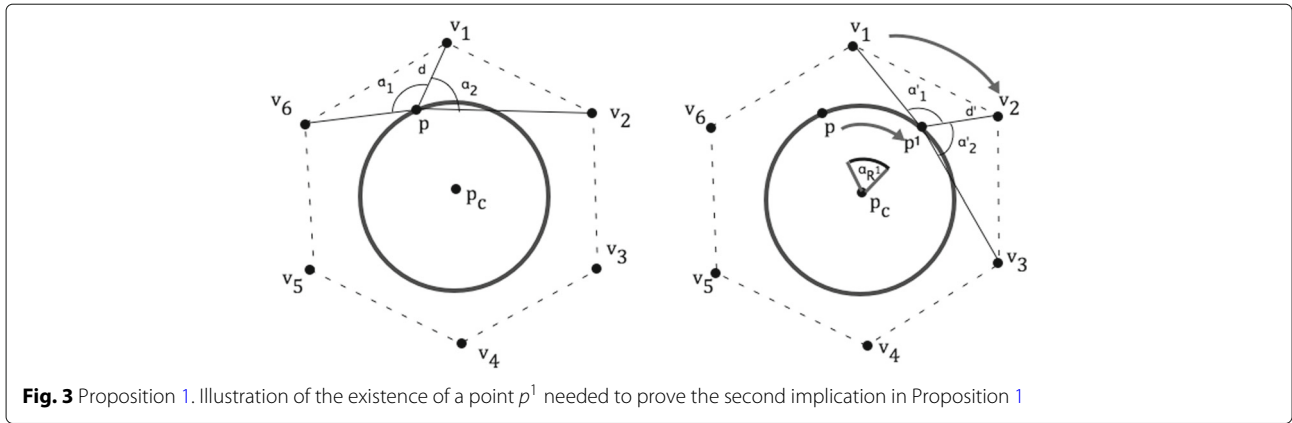
So, once we have these points, we know that given any point  $q \in C^{W_0}$ , there exist a point  $p'_1 \in C$  so that  $q = \sum_j^N \varphi_j^V(p)w_j^1$  and it is, in particular,  $p' = R_1(p)$ , considering we have the following:

$$\begin{aligned} q &= \sum_j^N \varphi_j^V(p)w_j^1 = w_1\lambda_1 + w_2\lambda_2 + \dots + w_N\lambda_N = \\ &= w_2\lambda_2 + w_3\lambda_3 + \dots + w_N\lambda_N + w_1\lambda_1 = \sum_j^N \varphi_j^V(p')w_j^1 \end{aligned}$$

Since we can generalize for any shift  $k \in \{1, 2, \dots, N\}$  with rotation  $R_k$ , the Proposition is proven.  $\square$

In Proposition 1, we show a way to compare a family of shapes defined by the CAC. Thus, we provide a new way to describe the shape. This shape descriptor does not depend on landmarks or keypoints, avoiding the manual, and many times difficult, definition of these landmark points in a set of images. This property can be very useful in certain applications, as medical image. Moreover,





**Fig. 3** Proposition 1. Illustration of the existence of a point  $p^1$  needed to prove the second implication in Proposition 1

the shape descriptor can be used in applications such as automatic image morphing and warping. Image morphing is the result of the interpolation between two *objects*, with new shape and texture, while warping is the deformation of the shape of an image. Thus, morphing requires warping. To perform a morphing from an object into another, we proceed as follows. We assume that we have two objects  $O^1$  and  $O^2$  in images  $I^1$  and  $I^2$ , respectively. We start, for each object, with a regular cage-contour configuration,  $(C, V, r)$ . Let  $V^1$  and  $V^2$  be the resulting cages after minimization. Then, we can state:

- 1 By Proposition 1, if the resulting cages  $V^1$  and  $V^2$  are similar or similar to a shifted cage, the contours are similar.
- 2 By property 2.3, if there exists a similarity  $f$  between cages, then by that similarity, the mean value coordinates of  $O^1$  with respect to  $V^1$  are equal to the mean value coordinates of  $f(O^2)$  with respect to  $V^2$ .
- 3 In the proof of Proposition 1, we show that we can always find a shift of a shifted cage so that we may find the similarity  $f$ .

Given the segmentation of  $O^1$  and  $O^2$  defined by the two cages  $V^1$  and  $V^2$ , respectively, if  $V^1$  is similar to (a shifted version of)  $V^2$ , then the same similarity maps  $O^1$

to  $O^2$ . This property allows to perform a proper image morphing. If we want to morph two objects  $O^1 \in I^1$  and  $O^2 \in I^2$  which, respectively, have segmentation  $V^1$  and  $V^2$ , then we can define an intermediate cage by the following *interpolation*:

$$V^w = V^1 w + V^2 (1 - w), \tag{14}$$

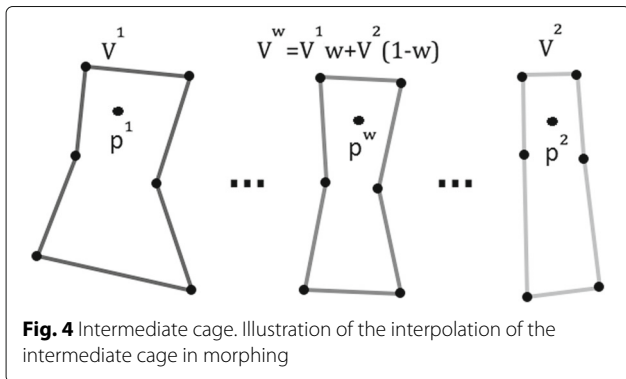
where  $w \in [0, 1]$ , such that if two cages are similar, they are also similar to their intermediate. In Fig. 4, we illustrate the result of the interpolation showing the intermediate cage for two cages ( $V^1$  and  $V^2$ ).

Once we have an interpolated cage  $V^w$ , the associated interpolated image  $I^w$  can be obtained from  $I^1$  and  $I^2$  by applying the following equations:

$$p^1 = \sum_{i=1}^N \varphi_i(p^w) v_i^1, p^2 = \sum_{i=1}^N \varphi_i(p^w) v_i^2, \tag{15}$$

$$I^w(p^w) = w I^1(p^1) + (1 - w) I^2(p^2) \tag{16}$$

In our approach, image morphing using the CAC is performed obtaining  $V^1$  and  $V^2$  by means of an energy function minimization technique such as the multivariate



**Fig. 4** Intermediate cage. Illustration of the interpolation of the intermediate cage in morphing

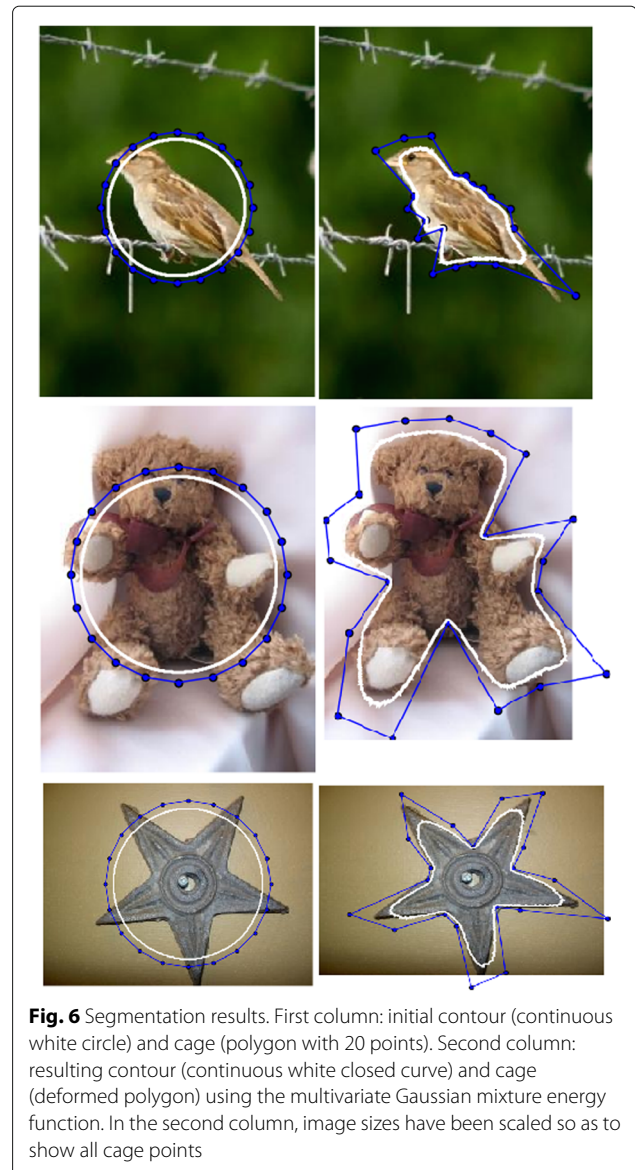
**Table 1** Comparison of the multivariate Gaussian mixture (MGM), Gaussian mixture (GM), and Gaussian segmentation energies in the CAC with other existing related methods

Method	AlpertGBB07		BSDS300	
	Dice (%)	Std.	Dice (%)	Std.
ChanVese	70.71	14.14	64.45	14.59
Shi	61.34	20.23	52.15	21.50
Caselles	58.68	17.02	63.15	13.76
MGM CAC	<i>77.88</i>	1.3	55.58	2.89
GM CAC	65.15	2.3	44.90	5.06
Gaussian CAC	57.13	3.38	44.90	5.06

*Italic means best performance*

**Table 2** Comparison of computational time of the CAC energies with other related methods in  $300 \times 225$  images

Method	Mean time ( sec.)	Std.
ChanVese	3.50	0.24
Shi	114.52	113.84
Caselles	3.47	0.69
MGM CAC	38.72	15.74
GM CAC	24.18	14.20
Gaussian CAC	22.78	11.96



Gaussian mixture model. Thus, the main advantage of the morphing with the CAC is that it is completely automatic. We automatically start from an initial cage configuration (see Definition 10, page 11), and it is not necessary to manually set points in the image, as it is the case of many other applications (of mean value coordinates) [41]. We have also directly available a similarity between cages, and it is not necessary to compute them.

#### 4 Results and discussion

We show in this section the experimental results obtained for the enhanced Gaussian energy function as well as for the shape similarity approach. We begin first with enhanced Gaussian energy function.

**Table 3** Distance matrix of segmented images

	Balloon	Bowl	Pumpkin	Sewer	Bird	Bear	Star
Balloon	0.0	0.87	0.13	0.55	0.3	0.06	1.1
Bowl		0.0	0.84	0.68	0.3	0.96	0.12
Pumpkin			0.0	0.74	0.7	0.26	1.03
Sewer				0.0	1.03	0.64	0.91
Bird					0.0	0.94	1.09
Bear						0.0	1.17
Star							0.0

#### 4.1 Datasets

We used two datasets in order to test our methods. The first dataset is a subset of 40 images from the Single Object Database (AlpertGBB07) [3]. This dataset is characterized by having well-defined backgrounds from the foreground. We discarded those images that we did not consider fitting the criteria for which Cage Active Contours were created, that is, images with single-connected objects with no holes and visually distinct from the background. The second dataset is the Berkeley Segmentation Dataset and Benchmark (BSDS300)[28]. This dataset consists of 300 real images which are much more complex than the Single Object Dataset since they are chosen in order to evaluate image segmentation in general and not object segmentation. Nevertheless, we have chosen a subset of 20 images from this dataset that was used in [35] and whose ground truth they provide for object segmentation.

#### 4.2 Evaluation measures

We have chosen to consider the Sørensen-Dice coefficient because of its simplicity and use in object image segmentation. This overlap ratio measure ranges from 0 to 100%, from least to most congruent. They are sensitive to misplacement of the segmentation label, although, in general, they do not capture shape fidelity.

Let  $X$  be the segmentation region and  $Y$  the ground truth segmentation region. The Sørensen-Dice coefficient is

$$\text{Dice}(X, Y) = 2 \frac{|X \cap Y|}{|X| + |Y|} \quad (17)$$

#### 4.3 Model validation

Cage Active Contours are adaptive methods with no learning. By adapting, we mean that through a few basic

rules, imposed in this case on the energy function and the cage, a certain intelligence emerges. The more elaborated these set of rules are, the more complex objects it will be able to segment. From simple rules, a more abstract and complex behavior emerges.

Usually, in model evaluation, there are two main points that we want to know: The overall score of a method and the best model for that method. In our case, the method corresponds to an energy function on the CAC while a model is a set of parameters. The model is evaluated as the mean score result throughout the whole dataset. The best model would then be that which best scores in a dataset.

To evaluate the *method* without over-fitting, we use threefold cross-validation.

#### 4.4 Results

We have carried out several quantitative experiments for comparing different energies in the CAC to evaluate our improvements and for comparing our methods to other existing ones to see where ours stand. We have considered the energies Gaussian CAC (8), multivariate Gaussian mixture (MGM) CAC (10), and Gaussian mixture (GM) CAC which is the same as the MGM with only intensity color. As comparison methods, we have chosen three active contour methods implemented in Creaseg [33] and reported to have the best results: the Geodesic Active Contours presented by Vicent Caselles [13], the Chan & Vese [15], and the Shi [37]. We have used the default parameters in [33].

In Table 1, we see the mean Sørensen-Dice coefficient and its standard deviation for each method and 60 images (40 from AlpertGBB07 and 20 from BSDS300). Our multivariate Gaussian mixture energy function scored best in the AlpertGBB07 dataset and third best in the BSDS300 dataset. These positive results were expected given that it uses RGB information while the methods from Creaseg use gray-scale images. For this reason, we have also decided to show the Gaussian mixture energy function which is the equivalent energy function in the gray-scale space. In this case, the Shi and the Caselles method were outperformed in the AlpertGBB07. In the case of BSDS300 dataset, the Chan-Vesse obtains the best mean performance; however, the CAC methods prove to be more stable since the standard deviation is much lower.



**Fig. 7** Morphing result. Morphing a family car to a sports car automatically through mean value coordinates from a segmentation with the CAC (initial image from [http://www.wellclean.com/wp-content/themes/artgallery\\_3.0/images/car1.png](http://www.wellclean.com/wp-content/themes/artgallery_3.0/images/car1.png), final image from <http://clipart-library.com/clipart/8i65pygMT.htm>)





**Fig. 8** Morphing result. Morphing from an apple to a pear with a CAC segmentation (initial and final images from [27])

In terms of computational time, in Table 2, we see that Caselles and Chan & Vese methods are extremely fast while the Shi, that is supposed to be fast, took the longest because of the default number of iterations in the Creaseg Implementation. Note that our approach is not able to outperform, from a computational point of view, other approaches. This is due to the fact that at each iteration, the pixels  $p$  of  $\Omega_1$  and  $\Omega_2$  have to be recovered and that for each pixel  $p$ , the affine coordinates have to be computed. This has, according to our experiments, a high computationally load and can be improved using parallelization languages such as OpenCL.

Figures 5 and 6 show qualitative results of eight images from AlpertGBB07 dataset segmented by MGM CAC method. Images shown are balloon, bowl, pumpkin, and sewer in Fig 5, and bird, bear, and star in Fig. 6. These results were obtained using the best parameters: number of control points 20, ratio 1.1,  $\sigma = 0.25$ ,  $\epsilon = e^{-200}$ . As it can be seen, the CAC method is able to properly segment the objects. The ability to adapt the curve to the object contour in the results depends on the number of control points. This parameter controls the regularization effect. This effect was studied in the previous work [18].

Moreover, it is worth to notice that CAC methods are not designed for high-precision segmentation of arbitrary images, but rather, they provide a smooth general contour of the image which can be used for other purposes and applications, as is illustrated in the next section.

#### 4.5 Applications: image morphing and warping

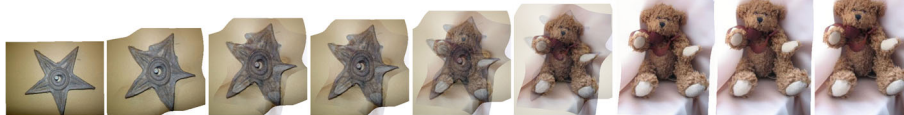
We validate the application of the CAC in shape similarity and image morphing. Table 3 shows the turning function similarity between the seven previously segmented objects in Figs. 5 and 6. As it can be seen, the cage similarity works properly for ordering similar shapes.

Next, we use the approach described in Section 3.3 for the morphing of two objects  $O^1$  and  $O^2$  into each other. As commented before, the morphing is automatic: we start

from two images  $I^1$  and  $I^2$  to which the multiGaussian mixture energy function segmentation method is applied. For both images, an initial regular cage is used. Once segmented cages  $V^1$  and  $V^2$  are obtained, intermediate cages can be obtained, and corresponding intermediate images are computed using interpolation.

In practice, if we segment two different images of the same object, the resulting cages may not necessarily be similar cages according to Definition 8. However, they can be similar up to a deformation of the cage. Thanks to the smooth properties of the warping using mean value coordinates, this allows a good morphing through interpolation of the cages. Figures 7, 8, and 9 show three examples created using cage interpolation (14), warping (15), and morphing (16). In these examples, we show 5 and 8 images. The images on the left and right correspond to the original objects,  $O^1$  and  $O^2$ , respectively, while the others (in the middle) are the interpolated objects. To obtain these images, we repeat the following steps as many times as desired: first, an intermediate cage between the two objects using cage interpolation is created; second, both objects are warped into the intermediate interpolated shape; and finally, a weighted average of the intensities results in the morphed image.

These results illustrate the power of the image morphing and warping method, which directly benefit from the segmentation result and obtain a smooth transition between the original images. In the first example (Fig. 7), the shift of the cages (Definition 9) that best corresponds to a similarity using a turning function is found. Recall that the turning function returns the correspondence of points between the two cages that has the minimum turning distance. The intermediate interpolated image can then be obtained using the correspondence of cages. The second example (Fig. 8) has been obtained by avoiding the step of finding the shift of the cages. The morphing results show smoothness since the segmentation also are similar. In the third example (Fig. 9), we have an example



**Fig. 9** Morphing result. Morphing from a star to a bear with a CAC segmentation

**Table 4** Car distance matrix

	Car1	Car2	Car3	Car4	Car5
Car1	0	0.05	0.09	0.12	0.14
Car2		0	0.05	0.09	0.13
Car3			0	0.05	0.09
Car4				0	0.05
Car5					0

of two images previously segmented with the CAC (see result in Fig. 6). Here, the morphing between the two different objects is smooth and the intermediate images clearly show the transition between the successive pairs. In the Additional file 1 we include an additional file video with an animation of a morphing result. In this animation one can appreciate the smooth transition between images.

Finally, the turning function similarity between the car and fruit shapes can be found in the distance matrices in Tables 4 and 5, respectively.

Note that the computational time associated to the segmentation process is high since, at each iteration of the algorithm, the interior and exterior pixels of the regions have to be computed. This is due to the fact that the latter interior and exterior regions are currently computed using a hole filling algorithm based on the contour drawn on the image. However, once the segmentation has been performed, the morphing process can be computed in an easy and efficient way since it is similar to image interpolation using optical flow. In our case, the point correspondence between the cage points allows to compute, in a fast way, the corresponding points at both original images for the pixels of the image to be interpolated. Interpolation is then fast to compute.

## 5 Conclusions

In this work, we have made various contributions to the framework of the Cage Active Contours (CACs). First, the introduction of energy functions on the RGB color space, Gaussian mixture, and multivariate Gaussian mixture models, which have greatly enhanced the potential of an otherwise limited method. These enhanced versions of the CAC provide the ability to capture multiple

**Table 5** Fruit distance matrix

	Fruit1	Fruit2	Fruit3	Fruit4	Fruit5
Fruit1	0	0.5	0.107	0.11	0.14
Fruit2		0	0.05	0.1	0.15
Fruit3			0	0.04	0.09
Fruit4				0	0.03
Fruit5					0

value components in each region, and the incorporation of an initial seed which provide the energy function with prior information about the foreground and background's distributions. Furthermore, we have mathematically formalized the concepts of *cage*, *contour*, *family of contours*, and others to be able to prove that two contours are similar if their cages are similar given some initial conditions. This theoretical proof, along with the properties of mean value coordinates, have allowed us to define the conditions and strategy for automatic morphing and warping between similar objects. We have also provided a similarity measure which has been used for shape comparison and could be also used in other applications.

Through quantitative and qualitative experiments on different datasets, we have validated the ability of the CAC framework for multiple steps for segmentation, warping, and morphing. The images are first segmented using the CAC, then the correspondences among cage control points of the shapes are estimated, and finally, a morphing between the images is constructed. We have shown that this process is automatic after the objects of interest have been located. This opens the door to different applications that will be considered as future work. A public implementation of Cage Active Contours in Python with some wrappers in C is available in <https://github.com/Jeronics/cac-segmenter/>. The code contains different energy functions presented in the paper and including the ones presented in [18], as well as tools for automatic morphing and warping.

As future work, we are interested in exploring new applications of the CAC framework, as for instance, automatic video interpolation and morphing for articulated object motion. We plan to explore robust functions for proper articulated object segmentation and warping. Moreover, we would like to use multiple dependent cages for local segmentation of object parts in an image, as well as for segmentation of the different objects/parts in a video.

## Endnotes

<sup>1</sup> <https://github.com/Jeronics/cac-segmenter/>

<sup>2</sup> In order to simplify the notation, we use  $\varphi(p)$  instead of  $\varphi^V(p)$  unless there is a possible ambiguity in the context.

<sup>3</sup> A cage defines a polygon by joining its vertices in order, the last with the first and removing the middle point of any consecutive collinear triplet (to fulfill the polygon definition). It is important to note that a cage is *not* a polygon since a cage can have three consecutive collinear points while a polygon cannot by definition.

<sup>4</sup> In this context, image refers to the target set of a function.



## Additional file

**Additional file 1:** Morphing result animation. (GIF 3041 kb)

### Funding

This work was supported by the Spanish Ministry of Science and Innovation (grant TIN2016-74946-P and grant TIN2015-66951-C2-1-R) and by Catalan Government award 2014-SGR-1219. These funding allowed to carry on the research for the design and development of the methods, analysis, and interpretation of the results, as well as writing the manuscript.

### Availability of data and materials

We used publicly available data in order to illustrate and test our methods: The first dataset is the Single Object Database (AlpertGBB07) [3], which can be found in [http://www.wisdom.weizmann.ac.il/~vision/Seg\\_Evaluation\\_DB/scores.html](http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/scores.html).

The second dataset is the Berkeley Segmentation Dataset and Benchmark (BSDS300)[28], which can be found in <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>.

We have also used images that can be found in: [http://www.wellclean.com/wp-content/themes/artgallery\\_3.0/images/car1.png](http://www.wellclean.com/wp-content/themes/artgallery_3.0/images/car1.png) <http://clipart-library.com/clipart/8i65pygMT.htm> <http://eprints.fri.uni-lj.si/2132/>.

Moreover, we have load all the material (code and test sets) in a Github repository: <https://github.com/Jeronics/cac-segmenter/>.

### Authors' contributions

LG and LI were responsible for the conceptualization, funding acquisition, project administration, resources, and supervision of the study. JC, LG, and LI were responsible for the formal analysis, investigation, methodology, and validation of the study as well as for writing the original draft, editing, and reviewing of the manuscript. JC was responsible for the data curation and visualization. All authors read and approved the final manuscript.

### Ethics approval and consent to participate

Not applicable

### Consent for publication

Not applicable

### Competing interests

The authors declare that they have no competing interests.

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 20 June 2017 Accepted: 10 January 2018

Published online: 09 February 2018

### References

1. S Abbasi, F Mokhtarian, J Kittler, Curvature scale space image in shape similarity retrieval. *Multimedia. Syst.* **7**(6), 467–476 (1999)
2. MS Allili, D Ziou, in *12th IEEE International Conference on Image Processing (ICIP) (1)*. An automatic segmentation of color images by using a combination of mixture modelling and adaptive region information: a level set approach (IEEE Signal Processing Society, Piscataway, 2005), pp. 305–308
3. S Alpert, M Galun, R Basri, A Brandt, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Image segmentation by probabilistic bottom-up aggregation and cue integration (IEEE Computer Society, Los Alamitos, 2007)
4. A Amanatiadis, V Kaburlasos, A Gasteratos, S Papadakis, Evaluation of shape descriptors for shape-based image retrieval. *Image Process. IET.* **5**(5), 493–499 (2011)
5. E Arkin, L Chew, D Huttenlocher, K Kedem, J Mitchell, An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Pattern. Anal. Mach. Intell.* **13**(3), 209–216 (1991)
6. E Arkin, L Chew, D Huttenlocher, K Kedem, J Mitchell, Geodesic active regions and level set methods for supervised texture segmentation. *Int. J. Comput. Vis.* **46**(3), 223–247 (2002)
7. D Barbosa, T Dietenbeck, J Schaerer, J D'hooge, D Friboulet, O Bernard, B-spline explicit active surfaces: an efficient framework for real-time 3-D region-based segmentation. *IEEE Trans. Image Process.* **21**(1), 241–251 (2012)
8. I Bartolini, P Ciaccia, M Patella, Warp: accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(1), 142–147 (2005)
9. O Bernard, D Friboulet, P Thévenaz, M Unser, Variational B-spline level-set: a linear filtering approach for fast deformable model evolution. *IEEE Trans. Image Process.* **18**(6), 1179–1191 (2009)
10. YY Boykov, MP Jolly, in *International Conference on Computer Vision (ICCV)*. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images, vol. 1 (IEEE Computer Society, Los Alamitos, 2001), pp. 105–112
11. A Bykat, On polygon similarity. *Inf. Process. Lett.* **9**(1), 23–25 (1979)
12. M Carreira-Perpinan, Mode-finding for mixtures of gaussian distributions. *Pattern. Anal. Mach. Intell. IEEE Trans.* **22**(11), 1318–1323 (2000)
13. V Caselles, F Catte, T Coll, F Dibos, A geometric model for active contours. *Numer. Math.* 694–6999 (1993)
14. V Caselles, R Kimmel, G Sapiro, Geodesic active contours. *Int. J. Comput. Vis.* **22**, 61–79 (1997)
15. T Chan, L Vese, Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
16. TF Cootes, CJ Taylor, DH Cooper, J Graham, Active shape models—their training and application. *Comput. Vis. Image Underst.* **61**(1), 38–59 (1995)
17. MS Floater, Mean value coordinates. *Comput. Aided Geom. Des.* **20**(1), 19–27 (2003)
18. L Garrido, M Guerrieri, L Igual, Image segmentation with Cage Active Contours. *IEEE Trans. Image Process.* **24**(12), 5557–5566 (2015)
19. K Hormann, M Floater, Mean value coordinates for arbitrary planar polygons. *ACM Trans. Graph.* **25**(4), 1424–1441 (2006)
20. M Jacob, T Blu, M Unser, Efficient energies and algorithms for parametric snakes. *IEEE Trans. Image Process.* **13**(9), 1231–1244 (2004)
21. P Joschi, M Meyer, T DeRose, B Green, T Sanocki, in *SIGGRAPH*. Harmonic coordinates for character articulation (ACM, New York, 2007)
22. X Jun, H Tsui, X Deshen, in *16th International Conference on Pattern Recognition*. Multiple objects segmentation based on maximum-likelihood estimation and optimum entropy-distribution (mle-oed), vol. 1 (IEEE Computer Society, Los Alamitos, 2002), pp. 707–710
23. M Kass, A Witkin, D Terzopoulos, Snakes: active contour models. *Int. J. Comput. Vis.* **1**(4), 321–331 (1988)
24. S Lankton, A Tannenbaum, Localizing region-based active contours. *IEEE Trans. Image Process.* **17**(11), 2029–2039 (2008)
25. C Li, C Kao, JC Gore, Z Ding, Minimization of region-scalable fitting energy for image segmentation. *IEEE Trans. Image Process.* **17**(10), 1940–1949 (2008)
26. Y Lipman, D Levin, D Cohen-Or, in *SIGGRAPH*. Green coordinates (ACM, New York, 2008), pp. 78:1–78:10
27. M Škrjanec, Automatic fruit recognition using computer vision. PhD thesis (2013)
28. D Martin, C Fowlkes, D Tal, J Malik, in *8th International Conference on Computer Vision*. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, vol. 2 (IEEE Computer Society, Los Alamitos, 2001), pp. 416–423
29. O Michailovich, Y Rathi, A Tannenbaum, Image segmentation using active contours driven by the Bhattacharyya gradient flow. *IEEE Trans. Image Process.* **16**(11), 2787–2801 (2007)
30. J Mille, L Cohen, in *Int. Conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMM/CVPR)*. A local normal-based region term for active contours (Springer-Verlag Berlin Heidelberg, 2009), pp. 168–181. Printed in Germany
31. J Nocedal, SJ Wright, *Numerical optimization, 2nd edn.* (Springer, New York, 2006)
32. S Osher, JA Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* **79**(1), 12–49 (1988)
33. N Paragios, R Deriche, in *17th IEEE International Conference on Image Processing (ICIP)*. Creaseg: a free software for the evaluation of image segmentation algorithms based on level-set (IEEE Signal Processing Society, Piscataway, 2010), pp. 665–668

34. F Precioso, M Barlaud, T Blu, M Unser, Robust real-time segmentation of images and videos using a smoothing-spline snake-based algorithm. *IEEE Trans. Image Proc.* **14**(7), 910–924 (2005)
35. C Rother, V Kolmogorov, A Blake, Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* **23**(3), 309–314 (2004)
36. M Rousson, R Deriche, in *IEEE Proceedings of the Workshop on Motion and Video Computing. A variational framework for active and adaptative segmentation of vector valued images* (IEEE Computer Society, Los Alamitos, 2002), pp. 56–61
37. Y Shi, W Karl, A real-time algorithm for the approximation of level-set-based curve evolution. *Image Process. IEEE Trans.* **17**(5), 645–656 (2008)
38. Y Shi, WC Karl, A real-time algorithm for the approximation of level-set-based curve evolution. *IEEE Trans. Image Process.* **17**(5), 645–656 (2008)
39. D Titterton, A Smith, U Makov, *Statistical Analysis of Finite Mixture Distributions*. (Wiley, New York, 1985)
40. J Vergés Llahí, Color constancy and image segmentation techniques for applications to mobile robotics (2005). PhD thesis
41. G Wolberg, *Digital Image Warping*. (IEEE Computer Society Press, Los Alamitos, 1990)
42. L Xu, M Jordan, On convergence properties of the em algorithm for gaussian mixtures. *Neural Comput.* **8**, 129–151 (1995)
43. Q Xue, L Igual, A Berenguel, M Guerrieri, L Garrido, in *Int. Conference on Computer Vision Theory and Applications. Active contour segmentation with affine coordinate-based parametrization* (Science and Technology Publications, Lda (SciTePress), Setúbal, 2014), pp. 5–14
44. D Zhang, G Lu, A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval. *J. Vis. Commun. Image Represent.* **14**(1), 39–57 (2003)

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](http://springeropen.com)

---