Elastica Energy Regularization via Graph Cuts

Daniel Martins Antunes, Jacques-Olivier Lachaud, Hugues Talbot

ELASTICA ENERGY REGULARIZATION VIA GRAPH CUTS*

DANIEL ANTUNES†, JACQUES-OLIVIER LACHAUD‡, AND HUGUES TALBOT§

Abstract. We propose a graph cut model to optimize bidimensional shapes with respect to the elastica energy. At each iteration our model selects the shape of minimum elastica value among a set of candidates generated by a discrete process that we call the balance coefficient flow. In this work we show how the balance coefficient flow relates with the curve-shortening flow and how our model can be included in an image segmentation pipeline. Finally, we provide a study to evaluate the effects of our model in the image segmentation task.

Key words. Shape Optimization, Elastica, Image Segmentation, Mean Curvature Flow, Digital Estimators

1. Introduction. A digital set D is defined as any collection of points that can be positioned in a regular grid. In the bidimensional case, it is a subset of the integer plane, i.e., $D \subset \Omega \subset \mathbb{Z}^2$, where Ω is a compact set. Digital images are one of the most prevalent examples of digital sets and also an important source of applications.

Common tasks in digital images are recognizing shapes or semantically coherent objects (segmentation), removing noise and blur (restoration), interpolate data (inpainting) and compression (image coding). An important class of models optimize a crafted functional energy adapted to the problem to be solved. In this class, the use of geometric priors, such as perimeter, area and curvature are commonly employed.

These models are built on classical mathematical theory, in which semi-continuity is often assumed. A common issue with most models using geometric priors lies in their discretization step, where the digital nature of the images are often considered as a necessary evil. This results in poor estimations of geometric quantities, and that is particularly important for high-order measures as curvature.

An important and challenging energy to optimize is the Elastica. Previous works reported its benefits in inpainting [29, 5, 11] and segmentation [21, 44, 33, 2]. In particular, the squared curvature penalization favors the so called *completion property*, which favors the segmentation of connected components. The completion property is particularly useful in the segmentation of thin and elongated objects, such as blood vessels. In continuous terms, the elastica is defined for a contour $\mathcal{C}$ as

$$E(\mathcal{C}) = \int_{\mathcal{C}} \alpha + \beta\kappa^2 ds.$$

In this paper, we propose a purely discrete model to minimize the Elastica energy using *multigrid convergent estimators*. These estimators are conceived for digital sets and provide guarantees of convergence with respect to finer and finer image grid resolutions. We show that our model evolves digital shapes to the shape of optimum elastica energy, escaping local minimum. Moreover, we show how to use our model in image processing tasks and give several illustrations. Finally, the model is highly parallelizable and present competitive running times with respect to state of the art methods. Our present running times could be further improved in a GPU setting.

2. Related work. The elastica has been introduced in image processing by Mumford [31] where the completion property (i.e., its preference for connected curves)

*A PRELIMINARY VERSION OF THIS PAPER APPEARED IN [3]

†Université Savoie Mont-Blanc, France (martinsantunes.daniel@gmail.com)
‡Université Savoie Mont-Blanc, France (jacques-olivier.lachaud@univ-smb.fr)
§CentraleSupelec, Université Paris-Saclay, Inria (hugues.talbot@centralesupelec.fr)

is particularly emphasized. The completion property is interesting for solving classical problems in imaging such as inpainting and segmentation, either to complete contours of wrongly disconnected segmented regions or to better extrapolate image level sets in regions without data.

In [11], the authors derive the 4th order Euler-Lagrange equation of their elastica regularized model to solve the inpainting problem. A gradient descent alike method is employed to find its root, but the model suffers from numerical instability and high running times. That is one of the main difficulties in minimizing the elastica under a continuous formulation. Some optimization properties of the elastica were studied in [1] and some techniques can be applied to mitigate the numerical instability as was done in [5]. In this work, the curvature is implicitly represented by a vector field acting under some constraints that are incorporated into the optimization energy. This alternative representation induces a reduction of one order in the equations solved during optimization. This strategy has been refined and explored more recently by [41] (inpainting) and [44, 15] (segmentation), where augmented lagrangian methods and variations such as ADMM are utilized. Nonetheless, these models are vulnerable to bad local optima and subject to high running times.

Due to the challenges posed by the necessary global optimization of the elastica, some authors propose to use alternative energies that preserve, in some sense, the elastica properties and are more tractable from the optimization point of view. In [7] a convex relaxation of the elastica is proposed and in [21, 43] the total curvature is used as an alternative to the elastica. The advantage of these models is that they induce convex energies, with methods that can optimize them globally. Nonetheless, the running time issue persists.

Finally, combinatorial methods were also proposed for elastica minimization. The premise behind such methods is that reducing the precision in the energy computation leads to a global optimization with smaller running times and without damping too much the results. In [17], a quadratic binary energy is proposed to solve segmentation with elastica regularization, but only squared angles are considered. A more general binary representation is given in [33] using concepts of integral geometry to obtain nice results for segmentation and inpainting. We also found elastica regularization in [38, 40] models for segmentation. In these works, the epiconvergent curvature estimator of [8] is used to approach the curvature term in a linear programming formulation for segmentation. Finally, graph cuts [4], negative cycle detection [39], and partial enumeration [34, 2] strategies were also explored for the tasks of denoising and segmentation.

In practice, the compromise between precision and running times does not play as well as expected. To obtain results of good quality, these combinatorial methods must run for a long time. As an alternative, one may consider optimization over a subdomain of the image such as in contour-evolution models. The classical active contour model (snakes) [23] and its variants [10, 12] have an embedded curvature component that is reflected on the Curve-Shortening Flow (CSF) of level sets in the case of geodesic active contours, or the CSF of the explicit curve in the case of snakes. The CSF is the unidimensional case of the mean curvature flow and can be reproduced in a discrete setting [30]. This has been explored in processes such as threshold dynamics [18, 19], or morphological operations sequence [32]. However, these methods cannot easily integrate a data term. In [42], a two-step optimization process is proposed to solve a Chan-Vese alike model and threshold dynamics is used to optimize the length penalization term of the model while a separated optimization process is used for the data term.

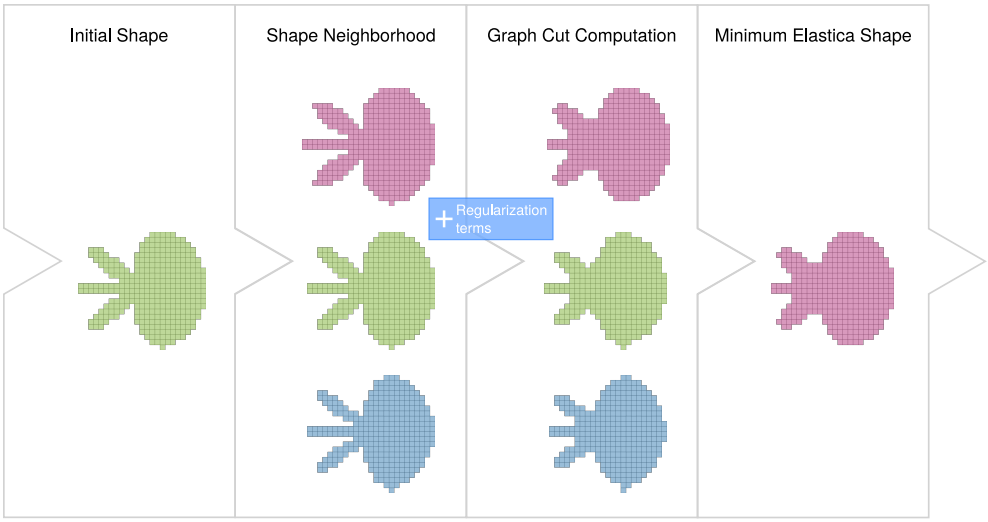


Fig. 3.1: Elastica minimization via graph cuts. Our model builds one graph for each member of a pre-defined neighborhood of the initial shape. A cost function is defined such that the minimum cut of the graphs reflects the energy we aim to optimize with its desired regularization terms. Finally, we select the shape with minimum elastica value, estimated via a multigrid convergent estimator. The process is repeated for the selected shape and it stops after a fixed number of iterations. In our experiments on image segmentation, we use regularization terms for data and curvature.

In the last years, novel estimators of geometric properties have appeared, for instance to estimate accurately the curvature along the boundary of digitized shapes [36, 37, 13, 14]. In common, they have the desirable property of multigrid convergence which defines a notion of convergence across digitizations of the target shape with higher and higher resolutions. This is important because it guarantees an upper bound on the curvature estimation error which decreases to zero as the resolution increases. This is not true for the curvature estimator of Bruckstein *et al.* [8] for example, which is acccurate only for finer and finer *sampling* of continuous curves. These novel estimators motivate us to propose models for imaging tasks that could take advantage of their multigrid convergence property.

3. Main contributions. We describe a discrete process to evolve digital shapes and we show that under some conditions this process behaves similarly to the curve-shortening flow. Next we demonstrate experimentally that the former process can be used to optimize shapes with respect to their elastica value and that the optimum shape is attained in some cases. Finally, we propose a graph cut model that integrates our first two results to execute the task of image segmentation. (the model is diagramized in Figure 3.1). We present several illustrations of our segmentation model and we compare it against the grabcut model [35]. Source code, experiments and figures are made publicly available [1]

4. Estimation of geometric quantities on digital data. Let $\mathcal{C} : [0, t] \to \mathbb{R}^2$ a parameterized plane curve with continuous first and second derivatives. In this case,

[1] https://danoan.github.io/graph-flow

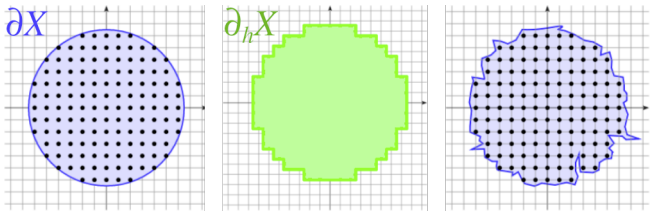


Fig. 4.1: **Digitization ambiguity.** The middle image (in green) is a valid digitization for both the left and right continuous shapes (in blue).

we can easily compute the curvature at some point $P(t) = \big(x(t), y(t)\big) \in \mathcal{C}$ by using the formula

$$\kappa(t) = \frac{y'(t)x''(t) - x'(t)y''(t)}{(x'(t)^2 + y'(t)^2)^{3/2}}.$$

What about if we do not know the curve equation, but instead, we have a finite set of points that is an *exact sampling* $\mathcal{C}$? That is, a finite sequence P is an *exact sampling* of $\mathcal{C}$ whenever each element of P is an element of $\mathcal{C}$ and the order of the points in the sequence corresponds to the ordering of points P when moving along $\mathcal{C}$. In this case, we can approach the curve $\mathcal{C}$ by a sequence of straight lines joining consecutive points of P and estimate the curvature by computing the angle defect between consecutive lines. This estimation is convergent (in the epi-convergent sense) as long as the sampling points are sufficiently numerous [9, 8].

The result above is not valid for curves lying in digital domains like images. We do not have an exact sampling. Instead, curve samples are constrained to lie in the digital grid. That condition creates ambiguities, which are clearly illustrated in Figure 4.1 where the same digitization represents two quite distinct shapes. Of course, one may refine the grid to reach a precision close to an exact sampling, but this is highly undesirable due to memory and running time complexity, in particular for image processing tasks. Furthermore, the polygonal contour is still locally very jagged, with only 4 possible directions. The above mentionned curvature estimator would not be convergent whatever the refinement.

Consequently we need a criterion to evaluate the quality and speed of convergence of geometric estimators according to the resolution of the digital grid. This criterion is the *multigrid convergence property* (e.g., see [24]). Let X be a Euclidean shape and ∂X its topological boundary. We further denote $D(X, h) := X \cap h\mathbb{Z}^2$ its Gauss digitization as a collection of points, $Q(x, h)$ its pixel representation (a collection of squares of side h), $\Delta_h X$ its interpixel contour as a collection of axis aligned edge segments of length h, and finally $\partial_h X$ the digitized contour, which is the union of all edge segments of $\Delta_h X$. See Figure 4.2 for an illustration of notations.

Definition 4.1 (Multigrid convergence for local geometric quantites). *A local discrete geometric estimator $\hat{z}$ of some geometric quantity z is uniformly multigrid convergent for some family $\mathbb{X}$ of Euclidean shapes if and only if, for any $X \in \mathbb{X}$, there exists a grid step $h_X > 0$ such that the estimate $\hat{z}(D(X, h), P, h)$ is defined for all $P \in \partial_h X$ with $0 < h < h_X$, and for any $Q \in \partial X$,*

$$\forall P \in \partial_h X \text{ with } \|P - Q\|_\infty \leq h, \|\hat{z}(D(X, h), P, h) - z(X, Q)\| \leq \tau_X(h),$$

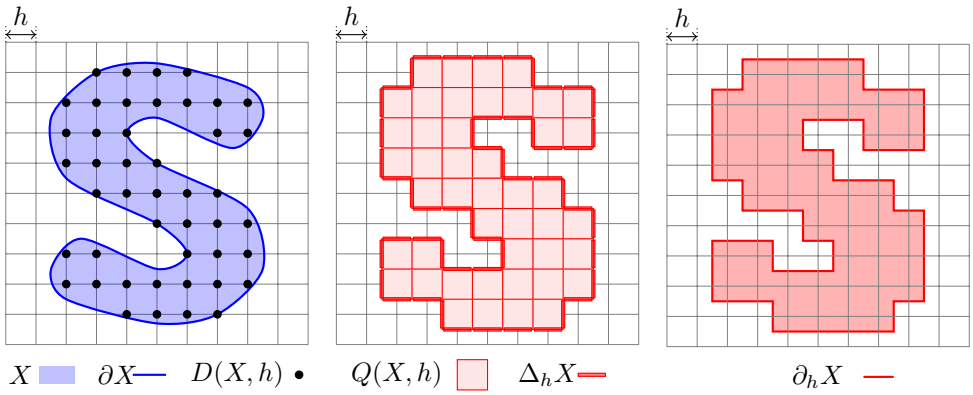


Fig. 4.2: Illustration of notations: the shape X, its topological boundary ∂X, its (Gauss) digitization $D(X,h) = X \cap h\mathbb{Z}^2$ seen as a collection of points, its pixel representation $Q(X,h)$ which is a set of squares of side h, the interpixel contour $\Delta_h X$ as a set of axis-aligned edge segments, and the digitized boundary $\partial_h X$, a subset of $\mathbb{R}^2$ that is the union of all edges of the interpixel contour, or equivalently the topological boundary of the union of all the squares of $Q(X,h)$.

where $\tau_X : \mathbb{R}^+ \setminus \{0\} \to \mathbb{R}^+$ *has null limit at* 0. *This function defines the speed of convergence of* $\hat{z}$ *towards* z *for* X.

For a global geometric quantity (e.g. perimeter, area, volume), the definition remains the same, except that the mapping between ∂X and $\partial_h X$ is no longer necessary.

Recently, estimator for curvature and perimeter estimators have been proved multigrid convergent. We propose to use such estimators to estimate the elastica energy of digital shapes. Let us define a digital analogue to the Elastica energy, which uses two such estimators:

$$\hat{E}_r\big(D(X,h),h,\alpha,\beta)\big) = \sum_{e \in \Delta_h X} \hat{s}(e)\,\big(\,\alpha + \beta \hat{\kappa}_r^2(D(X,h),\dot{e},h)\,\big). \tag{4.1}$$

The symbol $\dot{e}$ denotes the center of the edge e. The function $\hat{s}$ denotes the elementary length estimator, i.e., a measure of length is assigned to each edge e of the digital curve $\Delta_h X$. The elementary length is computed using the λ-*MST* estimator of tangent [27, 25], proven multigrid convergent for the family of convex shapes that are twice differentiable and have continuous curvature. The speed of convergence is $O(h^{1/3})$. It simply defines the elementary length of an edge as the scalar product between the edge vector and the convergent tangent vector estimate.

The function $\hat{\kappa}_r$ denotes an estimator of curvature. We use the integral *invariant estimator* [13], proven multigrid convergent for the family of compact shapes in the plane with 3-times differentiable contour. The radius of the integration ball is a parameter of this curvature estimator and its convergence speed is of the order of $O(h^{\frac{1}{3}})$ for radii chosen as $r = \Theta(h^{\frac{1}{3}})$ [26]. We present its definition since it is at the core of the model proposed in this paper.

$$\tilde{\kappa}_r(D(X,h),P,h) \coloneqq \frac{3}{r^3}\left(\frac{\pi r^2}{2} - \widehat{\mathrm{Area}}\big(D(B_r(P),h) \cap D(X,h),h\big)\right), \tag{4.2}$$

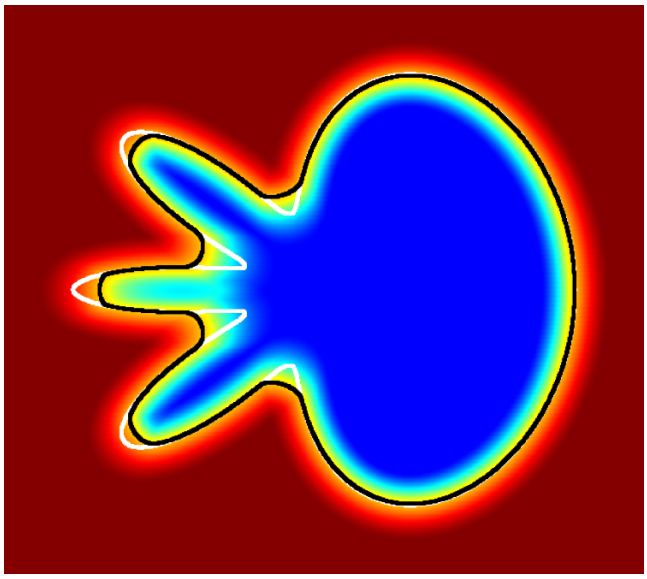


Fig. 5.1: **Balance coefficient zero-level set**. Evolving the initial contour (colored in white) to the zero-level set of the balance coefficient (colored in black) is closeley related with the curve-shortening flow.

where the estimation of area for some digital set D is defined as $\widehat{\text{Area}}(D,h) := h^2\text{Card}\,(D)$.

Equation (4.2) estimates the curvature as a scaling factor of the difference between the intersection area of the digitized shape with a disk of radius r centered at point P of the contour and half the disk area. Therefore, we can say that the curvature is lower at points in which the balance between intersected and non intersected points is closer to zero.

In the following, we simply write D to specify a digital shape and we omit the grid step h to simplify expressions (or, putting it differently, we assume that the shape of interest is rescaled by $1/h$ and we set $h = 1$).

5. Balance coefficient flow. Let $X \subset \mathbb{R}^2$ be a Euclidean shape, r a positive real number and P an arbitrary point of $\mathbb{R}^2$. We define the *balance coefficient* u_r of X at P as

$$u_r(X,P) = \left(\frac{\pi r^2}{2} - \text{Area}(B_r(P) \cap X)\right).$$

We observe that the balance coefficient definition is similar to the integral invariant estimator of curvature in equation (4.2) (we can model the continuous domain as a digital domain with an infinitely small grid step). However, we do not use it to estimate the curvature, but rather as an indicator of the local degree of convexity of the shape, even for points distant to the shape boundary.

In the following, we use the balance coefficient to define the balance coefficient flow

Definition 5.1 (Balance coefficient flow). *Let $X \subset \Omega$ be a Euclidean shape and $r > 0$ the radius of the disk used to compute the balance coefficient. We define the*

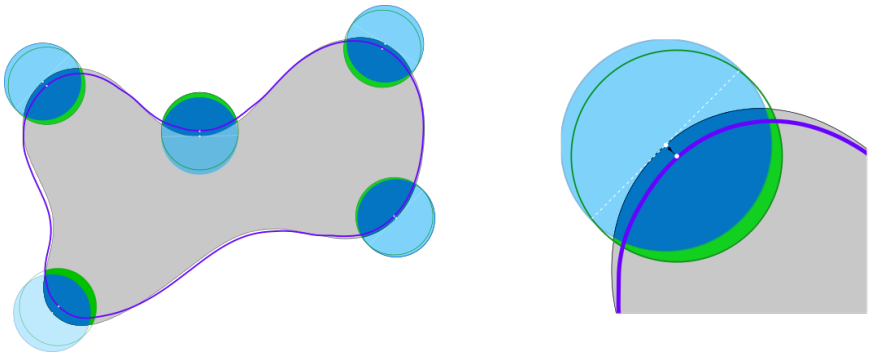


Fig. 6.1: **Geometric interpretation of the balance coefficient flow.** A single iteration of the balance coefficient flow evolves the shape X to the shape X' such that any disk of radius r centered at any point of the contour of X' intersects an area of $\pi r^2/2$ of the initial shape X.

balance coefficient flow *for discrete time steps as the sequence of Euclidean shapes*

$$X^{(0)} := X,$$
$$\forall k \in \mathbb{Z}, k > 0, \quad X^{(k+1)} := \left\{ x \in \Omega \mid u_r(X^{(k)}, x) \leq 0 \right\}. \tag{5.1}$$

6. Balance coefficient and curve-shortening flow. One iteration of the balance coefficient flow applied on shape X is equivalent to find the shape X' such that for any disk of radius r centered at any point of the contour of X', the intersection of the disk with the shape X equals $\pi r^2/2$. This interpretation is illustrated in Figure 6.1.

Let us show the link between the balance coefficient flow and the curve-shortening flow. First we rewrite the balance coefficient flow as a contour evolution. Let us denote $\mathcal{C}^{(0)} := \partial X^{(0)}$ the boundary of $X^{(0)}$. Then, for $x \in \mathcal{C}^{(0)}$, let $\epsilon^{(0)}(x)$ be a solution to the equation

$$u_r(X^{(0)}, x + \epsilon^{(0)}(x)\mathbf{n}^{(0)}(x)) = 0, \tag{6.1}$$
$$|\epsilon^{(0)}(x)| < r,$$

with $\mathbf{n}^{(0)}(x)$ the outward normal vector at x to the boundary of the shape $X^{(0)}$. Specifying $|\epsilon^{(0)}(x)| < r$ clearly imposes a unique solution to (6.1), provided $r < 1/\kappa$ (κ being the curvature value at x). Then the contour evolution of $\mathcal{C}^{(0)}$ is defined as:

$$\mathcal{C}^{(1)} := \left\{ \sigma^{(0)}(x), x \in \mathcal{C}^{(0)} \right\}, \tag{6.2}$$

where $\sigma^{(0)}$ is the mapping from $\mathcal{C}^{(0)}$ to $\mathcal{C}^{(1)}$, such that x maps to $x + \epsilon^{(0)}(x)\mathbf{n}^{(0)}(x)$.

The following proposition indicates that $\mathcal{C}^{(1)}$ coincides with the boundary of $X^{(1)}$ under some hypotheses.

PROPOSITION 6.1. *If r is small enough, the boundaries of shapes $X^{(0)}$ and $X^{(1)}$ are Jordan curves, and the boundary of shape $X^{(0)}$ is twice differentiable, then $\mathcal{C}^{(1)}$ is the boundary of $X^{(1)}$, and the mapping $\sigma^{(0)}$ between $\mathcal{C}^{(0)}$ and $\mathcal{C}^{(1)}$ is bijective.*

Proof. By definition, the curve $\mathcal{C}^{(0)}$ is the boundary of shape $X^{(0)}$. Being twice differentiable the curve $\mathcal{C}^{(0)}$ has a reach that is greater than some $\rho > 0$.[2] Hence, any point at distance lower than ρ from $\mathcal{C}^{(0)}$ has a unique closest point on this curve, in the direction normal to the curve.

Let x be a point on $\mathcal{C}^{(0)}$ and let $y = \sigma^{(0)}(x) = x + \epsilon^{(0)}(x)\mathbf{n}^{(0)}(x)$. Obviously, the smaller is r, the closer is $\mathcal{C}^{(1)}$ from $\mathcal{C}^{(0)}$. From the definition of balance coefficient, one can see that the displacement that is solution to (6.1) cannot exceed $\frac{1}{2}\kappa r^2$. So if $r < \sqrt{2\rho/\kappa_{\max}}$, where $\kappa_{\max}$ is the maximum absolute curvature of the curve, then the point y is within the reach of the curve $\mathcal{C}^{(0)}$. It follows that the mapping $\sigma^{(0)}$ is bijective. It is also continuous since all functions are continuous (the area of the shape intersected by a ball is continuous with respect to a displacement of the ball center). Hence $\mathcal{C}^{(1)}$ is a Jordan curve within the reach of $\mathcal{C}^{(0)}$. It has therefore an interior component I.

It is clear also that $y \in X^{(1)}$ since $u_r(X^{(0)}, y) = 0$. Let us denote $z(\lambda) = (1-\lambda)x + \lambda y$.

If $u_r(X^{(0)}, x) < 0$ then $u_r(X^{(0)}, z(\lambda))$ is by construction strictly increasing for increasing values of $0 \leq \lambda \leq 1+\nu$, $\nu \ll 1$, and so becomes positive for $1 < \lambda < 1+\nu$. In this case $\mathcal{C}^{(0)}$ was concave at x, so the straight segment $[x, y]$ is included in $X^{(1)}$ and y lies on its boundary. We have also $X^{(1)} \cap [x, y] \subset I$, and x lies in the interior of $X^{(1)}$.

If $u_r(X^{(0)}, x) > 0$ then $u_r(X^{(0)}, z(\lambda))$ is by construction strictly decreasing for increasing values of $0 \leq \lambda \leq 1+\nu$, $\nu \ll 1$, and so becomes negative for $1 < \lambda < 1+\nu$. In this case $\mathcal{C}^{(0)}$ was convex at x, so the straight segment $[x, y[$ is excluded from $X^{(1)}$ and y lies on its boundary. We have also $X^{(1)} \cap [x(1), x(1+\nu)] \subset I$ and $x(1+\nu)$ lies in the interior of $X^{(1)}$.

In both case we can see that y lies on the boundary of $X^{(1)}$. So $\mathcal{C}^{(1)} \subset \partial X^{(1)}$. Since both are Jordan curves they must coincide and $\mathcal{C}^{(1)} = \partial X^{(1)}$. Finally, the interior of $X^{(1)}$ must coincide with I since both are finite, they share the same boundary, and they share points (whether the boundary is convexe or concave). ∎

The proof tells us that the radius r must be smaller that $\min(\sqrt{2\rho/\kappa_{\max}}, 2/\kappa_{\max})$, for $\rho > 0$ smaller than the reach of the shape and $\kappa_{\max}$ the maximal absolute curvature of the shape.

Let us recall that the curve-shortening flow for a small time step $t > 0$, may be defined as:

$$\begin{aligned} \Gamma^{(0)} &:= \partial X \\ \Gamma^{(t)} &:= \left\{ x + t\kappa(x)\mathbf{n}(x), x \in \Gamma^{(0)} \right\}, \end{aligned} \tag{6.3}$$

denoting by $\kappa(x)$ the curvature of the curve at point x and by $\mathbf{n}(x)$ the outward normal vector at x.

Of course, there are implicitly defined variants of this formulation that allows topology changes, but we will restrict ourselves here to proving the similarity between the balance coefficient flow and the curve-shortening flow when there is no topology change.

PROPOSITION 6.2. *For small enough r, setting $t = \frac{1}{6}r^2$, assuming $\mathcal{C}^{(0)} = \Gamma^{(0)}$ is a twice differentiable Jordan curve, then the Hausdorff distance between $\mathcal{C}^{(1)}$ and $\Gamma^{(t)}$*

[2] Let us recall recall that the *reach* is the infimum distance between a shape and its medial axis. Any point within the reach has a unique closest point to the shape by definition.

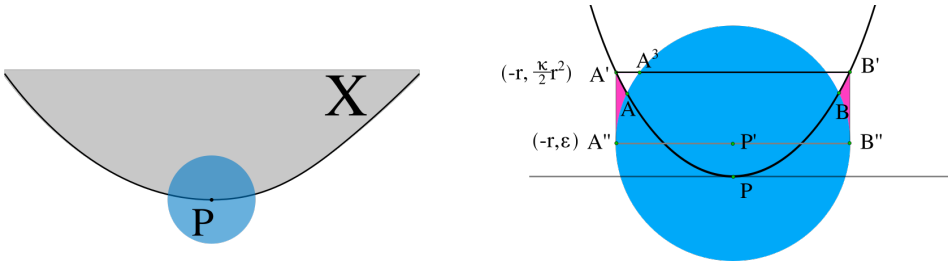


Fig. 6.2: **Balance coefficient and curve-shortening flow (CSF).** We approximate the contour ∂X at P by a parabola. Next, we compute the point P' such that the balance coefficient equals to zero. Our approximation produces an error Δ, highlighted in magenta, that is of order $O(r^4)$.

is some $O(t^2)$, more precisely we get:

$$d_H(\mathcal{C}^{(1)}, \Gamma^{(t)}) \leq \frac{3}{4}\kappa_{\max}^3 t^2.$$

Proof. Let $y \in \mathcal{C}^{(1)}$. According to Proposition 6.1, there is a point $P \in \mathcal{C}^{(0)}$ such that $y = \sigma^{(0)}(P)$. Let $y' = P + t\kappa(P)\mathbf{n}(P)$. We show that $\|y - y'\| \leq \frac{3}{4}\kappa_{\max}^3 t^2$, which induces the result (we could have chosen y' first, and obtain the same y).

We center the reference frame on P with y-axis aligned with the normal vector $\mathbf{n}(P)$. Since r is small enough, the contour is well approximated by the parabola $f(x) := \frac{\kappa(P)}{2}x^2$ in this frame.

We know that $y = P + \epsilon^{(0)}(P)\mathbf{n}(P)$ according to (6.1), where $\epsilon^{(0)}(P)$ is the local displacement of P which gives a zero balance coefficient. Let us determine this displacement $\varepsilon := \epsilon^{(0)}(P)$ in the normal direction, such that the intersection of the disk with X equals half of the disk area. Let us say that the point of zero balance coefficient is P' and that A_ε is the intersection area of the displaced disk (see Figure 6.2, right, for notations). Then,

$$A_\varepsilon = \frac{\pi r^2}{2} \qquad \Rightarrow \qquad 2r\varepsilon = \int_{-r}^{r} \frac{1}{2}\kappa x^2 dx \pm \Delta \qquad \Leftrightarrow \qquad \varepsilon = \frac{1}{6}\kappa r^2 \pm \frac{\Delta}{2r}.$$

It remains to compute an upper bound for the error Δ. Let A^3 be the leftmost intersection point between the line $A'B'$ and the estimation disk. It is clear that $\Delta/2$ is smaller than the region between $A''A'A^3$ and the arc going from A^3 to A''. Let d be the length of the segment $A'A''$.

$$\begin{aligned} Area(A''A'A^3) &= \int_0^d r - \sqrt{r^2 - x^2}dx \\ &= -\frac{r^2}{2}sin^{-1}\left(\frac{d}{r}\right) + dr - \frac{d}{2}\sqrt{r^2 - d^2} \\ &\approx \frac{d^3}{6r}. \end{aligned}$$

In general, d is bounded by $\frac{1}{2}\kappa r^2$, hence it follows:

$$\frac{\Delta}{2} \leq \frac{d^3}{6r} \leq \frac{1}{48}\kappa^3 r^5.$$

Therefore,

$$\varepsilon = \frac{1}{6}\kappa r^2 \pm \frac{1}{48}\kappa^3 r^4.$$

To conclude the argument, we have:

$$\begin{aligned}
\|y - y'\| &= \|P + \epsilon^{(0)}(P)\mathbf{n}(P) - (P + t\kappa(P)\mathbf{n}(P))\| \\
&= |\varepsilon - t\kappa(P)| \\
&\leq \left|\frac{1}{6}\kappa(P)r^2 - t\kappa(P)\right| + \frac{1}{48}|\kappa(P)|^3 r^4 \\
&\leq \frac{1}{48}|\kappa(P)|^3 r^4 && (\text{since } t = \frac{1}{6}r^2) \\
&\leq \frac{3}{4}\kappa_{\max}^3 t^2 && (\text{since } |\kappa(P)| \leq \kappa_{\max})
\end{aligned}$$

which shows that the two contours are close in the Hausdorff sense. ∎

Applying Proposition 6.1 at each time step k (writing (k) instead of (0) and $(k+1)$ instead of (1)) allows us to redefine the balance coefficient flow as the sequence $\mathcal{C}^{(k)}$, provided that the boundaries of $X^{(k)}$ are twice differentiable Jordan curves. But Proposition 6.2 applied on these consecutive curves tells us that the curves $\mathcal{C}^{(k)} = \partial X^{(k)}$ are very close to the curve shortening flow, which induces twice differentiable curves except at critical points. Proving that the flow $X^{(k)}$ also mimicks the curve-shortening flow across critical points would require much more mathematical work and is outside the scope of this paper.

The CSF has many interesting properties [22, 20, 16]. Among those, the CSF is the continuous deformation that decreases the perimeter of a single closed curve at the fastest speed; and it also preserves convexity. In particular, the CSF eventually collapses the initial curve to a single point.

There are interesting links between the CSF and a variant of the heat equation defined for the indicatrix function of a set [30]. In this same work, the authors informally give a geometric interpretation for the CSF that is equivalent to our zero-level set of the balance coefficient. A technique that emerged from the interpretation of CSF as a heat equation is the so called threshold dynamics [18, 19]. However, the use of threshold dynamics for image processing tasks is not immediate due to the difficulty to inject a data fidelity term. That is not the case in our approach.

We are going to apply the balance coefficient flow in a discrete setting. In theory, that means that we have some limitations regarding the choice of the estimation radius r, the grid resolution h and the real curvature value at the estimation point. In general, we need to respect

$$h \ll r \ll \frac{1}{\kappa}.$$

In the next section we show how to encode the balance coefficient flow in a graph cut framework to optimize digital shapes with respect to (4.1) and how it can be embedded in a image segmentation model as an elastica prior.

7. Elastica minimization of digital shapes via graph cuts. A cut $C = (S, T)$ of a graph $(\mathcal{V}, \mathcal{E})$ is a partition of its vertices on subsets S and T. The set of edges connecting vertices of S to vertices of T is called the cut-set of C and it defines the value of the cut, which is given by the sum of the edge capacities in the cut-set.

We are going to model our optimization problem by defining a graph on the top of our digital object (image). On this setting, a point (pixel) in the digital object is mapped to a vertex in the graph and the edges and their capacities reflect the problem we are trying to solve. In image segmentation, for example, the goal is to find a partition of the pixels that separates the image in background and foreground pixels. One can find this partition by finding the minimum cut of the graph [6].

In this section, we are going to use the graph cut approach to optimize pure digital shapes (that is with no color information) with respect to energy (4.1). After validation of this process, we are going to incorporate it into a graph cut model for image segmentation model [6].

We recall our estimator for the elastica energy

$$\hat{E}_{\Theta}(D) = \sum_{\vec{e}\in\partial D} \hat{s}(\vec{e}) \left(\alpha + \beta \hat{\kappa}_r^2(D, \vec{e}) \right). \tag{7.1}$$

Notice that we group all parameters in vector $\vec{\Theta} = (h, r, \alpha, \beta)$. We remark that, since it is composed of multigrid convergent estimators, the elastica estimator is also multigrid convergent [25]. Our process can be divided in three parts. First we define a neighborhood of shapes with respect with the given digitized object. Next, for each shape in the neighborhood we create a candidate graph. A graph-cut optimization is done in each candidate graph and the result of each optimization gives us a candidate shape. Finally, we compute the elastica on the candidate shapes and select the one with lowest elastica value.

7.1. Neighborhood of shapes. We start by defining a very simple neighborhood of shapes. Of course, richer neighborhoods could be used, but we stick here to this elementary one, which will prove to be sufficient in our experiments.

Definition 7.1 (Neighborhood of shapes). *Let $D \subset \mathbb{Z}^2$ a digital shape. We define its* neighborhood $\mathcal{N}(D)$ *as the set*

$$\mathcal{N}(D) = \left\{ D, D^{+1}, D^{-1} \right\},$$

where D^{+1} (D^{-1}) denotes a morphologic dilation (erosion) by a square of side 1.

7.2. Computation of candidates. We are going to define a directed weighted graph and set its edge weights such that we can evolve the shape towards the zero-level set of its balance coefficient. Since the balance coefficient is a local quantity, it is sufficient to define the graph in a band around the initial contour. This somewhat complex approach to computing a zero-level set of a function has the great advantage of allowing the integration of other terms in its formulation, like fitting to data. Experiments will show that it has a tremendous effect on the quality of image segmentation results.

Let $d_D : \Omega \to \mathbb{R}$ be the signed Euclidean distance transformation with respect to shape D. The value $d_D(P)$ gives the Euclidean distance between $P \notin D$ and the closest point in D. For points $P \in D$, $d_D(P)$ gives the negative of the distance between P and the closest point not in D. Let n be a positive integer number.

Definition 7.2 (Optimization band). *Let $D \subset \Omega \subset \mathbb{Z}^2$ be a digital set . The optimization band $O_n(D)$ is defined as*

$$O_n(D) := \{ P \in \Omega \mid -n \leq d_D(P) \leq n \}.$$

Table 7.1: **Edge capacity function.** M is defined as $\max_{e\in\mathcal{E}} c(e)$.

edge e	**c(e)**	**for**
$\{v_P, v_Q\}$	$\frac{1}{2}\left(u_r(D,P) + u_r(D,Q)\right)$	$\{v_P, v_Q\} \in \mathcal{E}_u$
$\{v_P, s\}$	M	$v_P \in \mathcal{V}_s$
$\{v_P, t\}$	M	$v_P \in \mathcal{V}_t$

Definition 7.3 (Candidate graph). *Let $D \subset \Omega \subset \mathbb{Z}^2$ a digital set. The* candidate graph $\mathcal{G}(n, D, \mathcal{V}, \mathcal{E})$ *of D with optimization band n is defined such that:*

$$\begin{aligned}
\mathcal{V} &= \left\{\, v_P \mid P \in O_n(D) \,\right\} \cup \{s, t\},\\
\mathcal{E} &= \mathcal{E}_u \cup \mathcal{E}_{st},\\
\mathcal{E}_u &= \left\{\, \{v_P, v_Q\} \mid P \in O_n(D) \text{ and } Q \in \mathcal{N}_4(P) \,\right\},\\
\mathcal{E}_{st} &= \left\{\, \{s, v_P\} \mid d_D(P) = -n \,\right\} \cup \left\{\, \{v_P, t\} \mid d_D(P) = n \,\right\}.
\end{aligned}$$

The vertices s, t are virtual vertices representing the source and target vertices as it is common in a minimum cut framework. We denote $\mathcal{N}_4(P)$ the set of 4-adjacent neighbors of P. The innermost (outermost) pixels of the optimization band are connected to the source (target), and we identify such vertices as

$$\begin{aligned}
\mathcal{V}_s &:= \{v_P \in \Omega \mid d_D(P) = -n\},\\
\mathcal{V}_t &:= \{v_P \in \Omega \mid d_D(P) = n\}.
\end{aligned}$$

The set $\mathcal{E}_{st}$ comprises all the edges having the source as their starting point or the target as their endpoint. In Table 7.1 we describe the edge capacity function.

A cut in a graph separates the vertices in source and target components. Our model is defined such that the source component of the minimum cut gives the next shape in balance-coefficient flow.

7.3. Candidate Selection. The graph-cut optimization applied to each candidate graph results in a candidate shape. At this step, we simply compute the elastica energy on each candidate shape using (7.1) and then we select the candidate with lowest elastica value.

The *Graph Flow Algorithm* (GFA), Algorithm 7.1, summarizes the process. Some experiments are shown on Figure 7.1 and Figure 7.2.

Algorithm 7.1 Graph Flow Algorithm (GFA).

Input: A digital set D; the optimization band n; parameter vector $\vec{\theta} = (h, r, \alpha, \beta)$; the maximum number of iterations $maxIt$;
$D^{(0)} \longleftarrow D$
$t \longleftarrow 0$
while $t < maxIt$ **do**
  $\Gamma^{(t)} \longleftarrow \bigcup_{D' \in \mathcal{N}(D^{(t)})} \left\{ mincut(\mathcal{G}(D') \right\}$ {Computation of candidates}
  $D^{(t+1)} \longleftarrow \underset{D' \in \Gamma^{(t)}}{\arg\min}\, \hat{E}_{\vec{\theta}}(D')$ {Candidate selection}
  $t \longleftarrow t + 1$
end while

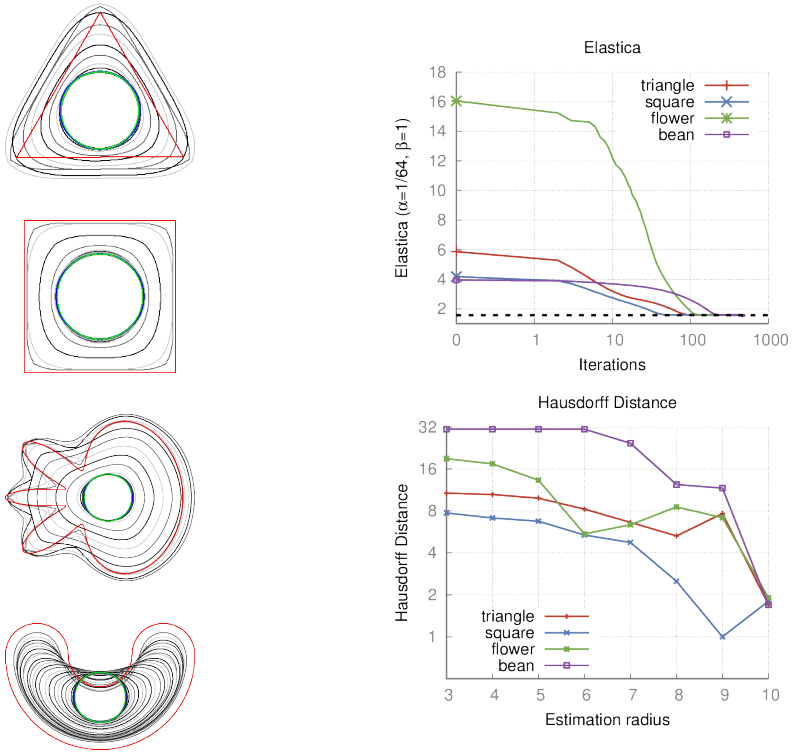
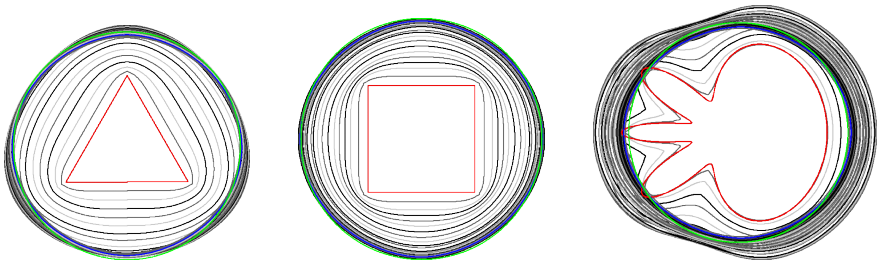


Fig. 7.1: **GFA experiments**. In the left, the evolution of various shapes given by the GFA ($h = 1/8, r = 2$), that is, the estimation disk has a radius of 16 pixels. The red and blue contours highlight initial and final contours. The green contour is the optimum solution. The top right graph describes the reduction in elastica energy. The dotted line marks the optimum energy value (an Euclidean disk of radius 8, or a disk digitization with 64 pixels of radius with $h = 1/8$). The bottom right graph displays the evolution of the Hausdorff distance between the optimal shape and the final shape given by GFA with respect to the radius of the estimation disk. It points out the importance of a right choice of the balance coefficient radius.

Remarkably, the GFA escapes premature local minima and even achieves the global optimum of the elastica energy for some cases. In Figure 7.2 we show the results of elastica minimization for $\vec{\Theta} = (r = 2, h = 1/8, \alpha = 1/1024, \beta = 1)$. The GFA correctly expands the shapes to the optimum disk of radius 32. However, we may have a premature interruption of the evolution if a small estimation disk radius is chosen, as illustrated in the bottom right graph of Figure 7.1. The computations can be done in parallel for each member of the neighborhood of shapes. Table 7.2 shows the running times where the candidate graphs are evaluated in parallel. These times can be further improved by parallelizing the computation of the balance coefficient as well as the elastica estimator evaluation. Finally, the GFA is easily modifiable to accomodate image terms, which makes it suitable for image processing tasks. Next, we are going to explore some of these possibilitites in image segmentation.



(a) (b) (c)

Fig. 7.2: **GFA can expand the initial shape**. Shapes evolutions by GFA with $\vec{\Theta}(h = 1/8, r = 2, \alpha = 1/1024, \beta = 1)$ as the elastica estimator parameters. The green contour highlights the optimum shape.

Table 7.2: **Running time of GFA**. The GFA achieves running times lower than one second per iteration (executed in a Intel Corei7 $1.8GHz$ processor with $16gb$ of ram). These running times could be further improved, for instance by parallelization of balance coefficient or elastica energy computations.

	Pixels	It	Time	Time/It
Triangle	33256	120	49.6s	0.4s
Square	51259	60	24.6s	0.4s
Flower	119789	150	132.8s	0.9s
Bean	100504	300	148.4s	0.5s

8. Application in image segmentation. In this section, we explore the potential of the GFA in the image segmentation task by extending it to include a data fidelity term. For images, since the digitization process cannot be done with an arbitrary resolution, we set $h = 1$ in all experiments. The experiments were executed on an Intel Corei7 $1.8GHz$ processor with $16gb$ of RAM. The source code and complete experiment report are available at: https://danoan.github.io/graph-flow.

The goal of this experiment is to illustrate the regularization properties of the GFA and to highlight the role of the data term in our approach. The data term employed in this experiment is the same used by Boykov-Jolly in their classical graph cut model [6].

8.1. Data term. We update the graph construction described in subsection 7.2 to accomodate the data term. In particular, we define two new sets of vertices $\mathcal{V}_{fg}$ and $\mathcal{V}_{bg}$ as the set of foreground and background seeds, respectively. Those are given as input.

Let $\vec{x} \in \{0,1\}^{|D|}$ represent the label of each pixel in the image (0 for background and 1 for foreground). We define the data term as

$$data(D, \gamma_r, \gamma_b) = \gamma_r \sum_{P \in D} \psi(x_P) + \gamma_b \sum_{P \in D} \sum_{Q \in \mathcal{N}_4(P)} \phi_{(P,Q)},$$

where $\gamma_r \geq 0$ and $\gamma_b \geq 0$ are parameters controlling the influence of the regional and boundary terms, respectively. Given the image $I : \Omega \to [0,1]^3$, the unary and pairwise

Table 8.1: **Updated capacity function.** The capacity function of Table 7.1 is updated to accommodate the data term. The constant M is simply the maximum of all capacities, i.e. $M = \max_{e\in\mathcal{E}} c(e)$.

edge e	$\mathbf{c(e)}$	**for**
$\{v_P, v_Q\}$	$\beta \cdot \big(u_r(D,P) + u_r(D,Q)\big) + \gamma_b \cdot \phi_{(P,Q)}$	$\{v_P, v_Q\} \in \mathcal{E}_u$
$\{s, v_P\}$	$\gamma_r \cdot \psi(0)$	$P \in O_n(D), v_P \notin \mathcal{V}_{fg} \cup \mathcal{V}_{bg}$
	M	$v_P \in \mathcal{V}_s \cup \mathcal{V}_{fg}$
$\{v_P, t\}$	$\gamma_r \cdot \psi(1)$	$P \in O_n(D), v_P \notin \mathcal{V}_{fg} \cup \mathcal{V}_{bg}$
	M	$v_P \in \mathcal{V}_t \cup \mathcal{V}_{bg}$

terms are defined as

$$\psi(x_P) = \begin{cases} -\ln H_{bg}\big(I(P)\big), & \text{if } x_P = 0 \\ -\ln H_{fg}\big(I(P)\big), & \text{if } x_P = 1, \end{cases}$$

$$\phi_{(P,Q)} = \begin{cases} \exp\left(-\dfrac{(I(P)-I(Q))^2}{|P-Q|}\right), & Q \in \mathcal{N}_4(P) \\ 0, & \text{otherwise.} \end{cases}$$

The terms H_{bg} and H_{fg} are mixed Gaussian distribution constructed from the foreground and background seeds. The updated capacity function is described in Table 8.1. To handle bias due to the magnitude difference between data and geometry terms, we normalize them in groups. Regional and boundary terms ψ, ϕ are normalized to the interval $[0, 1]$ with respect to their values. The same is done, separately, for the curvature term.

To minimize parameter dependence from the image input, we apply the following normalization factor to the data term.

$$\frac{\hat{E}(I_0, \alpha = 1, \beta = 1)}{data(I_0, \gamma_r = 1, \gamma_b = 1)}.$$

That is, $data(I_0, \gamma_r = 1, \gamma_b = 1)$ will have the same value as $\hat{E}(I_0, \alpha = 1, \beta = 1)$ after normalization.

Table 8.2: **Distribution of selected images.** The quantity of selected images per Coco super-category. A total of 212 images were selected.

Person	**Vehicle**	**Food**	**Animal**	**Outdoor Obj.**	**Sports**
24	22	22	34	16	19

Kitchenware	**Furniture**	**Appliance**	**Electronics**	**Indoor Obj.**
19	17	9	10	19

8.2. Experiments. We need an initial contour to start the GFA. The initial contour is given by the grabcut algorithm [35], a variant of classical graph cut segmentation [6] which is implemented in the OpenCV library.

Table 8.3: **Supervised experiment parameters**. List of the GFA parameters for the supervised segmentation experiment.

Estimation radius (**r**)	**Opt. band width** (**n**)	**Neighborhood size** (**k**)	**Iterations** **(maxit)**
5	4	3	20
Length weight ($\boldsymbol{\alpha}$)	**Curvature weight** ($\boldsymbol{\beta}$)	**Boundary weight** ($\boldsymbol{\gamma_b}$)	**Region weight** ($\boldsymbol{\gamma_r}$)
10	1	1	1

Table 8.4: **Supervised experiment running times**. The average time per iteration is of 0.7s.

Highest	**Lowest**	**Average**	**Average per iteration**
50.3s	3.3	15.1s	0.7s

For this experiment, we used a selection of 212 images of the Validation 2017 subset of the Coco dataset [28]. The Coco dataset comprises over 328k images spreaded over 91 categories and 11 super-categories. Table 8.2 summarizes the quantity of selected images per super-category. The following experiment studies the influence of inserting the data term into the graphcut formulation. It consists in the following steps:

1. manual selection of foreground and background seeds for each selected image;
2. computation of the grabcut segmentation;
3. and then using the grabcut segmentation as input for two versions of the GFA: one with the data term described in the previous section and another without.

In Figure 8.1 we show a sample of the images used in the experiment. All the results are available online at the paper's website. Table 8.3 lists the GFA parameters used for this experiment.

The first observation is that the GFA regularizes the initial grabcut segmentation contour with respect to the elastica energy. In Figure 8.2 we display the tangent profile of both grabcut segmentation and the one corrected by the GFA. We clearly observe the regularization effect of the GFA with respect to the grabcut profile. The value of the elastica energy of the contour and its number of inflection points is also greatly reduced, as summed up in Figure 8.4b.

The second observation is that the data term has an important role in the quality of the segmentation with respect to precision and recall metrics. This is shown in the box plot of Figure 8.4a. That means that a contour evolution model with no data term, such as threshold dynamics, is insufficient to recover segmentations of good quality. In fact, executing the GFA without data term will eventually transform the contour in one or more circles, and we can have an arbitrary bad value for the recall. We can see some of these undesirable effects in the third line of Figure 8.1: the kite area is overly reduced; the motorcycle is separated in two disconnected components; and the giraffe loses parts of its ears.

The third observation is that the GFA indeed presents the completion effect, which was expected from regularization by the elastica energy. This property is particularly useful for the segmentation of thin and elongated objects, but not only. It also helps

(a) Coco annotations for kite, moto and giraffe

(b) Grabcut segmentation

(c) GFA without data term

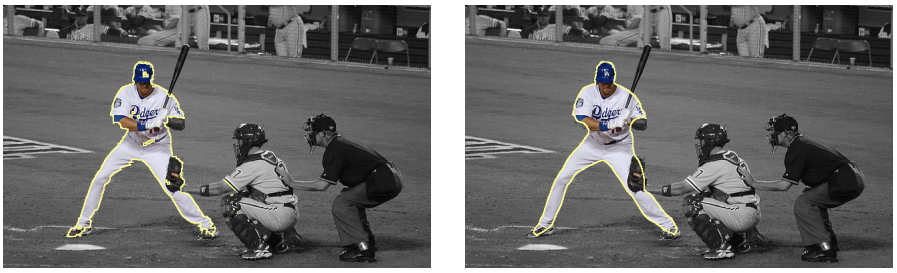
(d) GFA with data term

Fig. 8.1: **Some results of the image segmentation experiment**. Coco annotations are shown in the first line. In the next rows we present the same image segmented by grabcut and corrected by the GFA without and with data term.

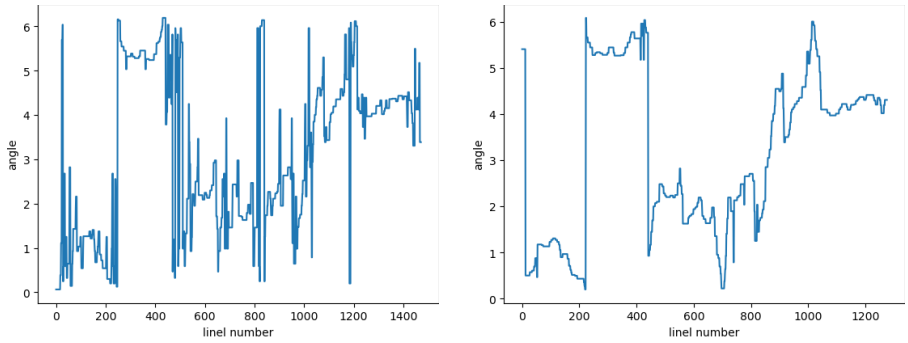
to remove oversegmented components which are particularly common in graph cut based models. In Figure 8.3 we give some examples of contour completion.

Finally, we remark that all image segmentations were executed using the same set of parameters. That is not ideal. Low resolution images should be segmented using a smaller estimation radius, for example. Therefore, all the results presented here could be eventually improved by tuning the parameters accordingly. An example of this is given in Figure 8.5.

A summary of the running time is presented in Table 8.4. The average running time per iteration is of 0.7s. We remark that in several cases we need fewer than 5 iterations to greatly improve the regularization metrics such as elastica and inflection points. However, to recover the completion effect we may need more iterations.

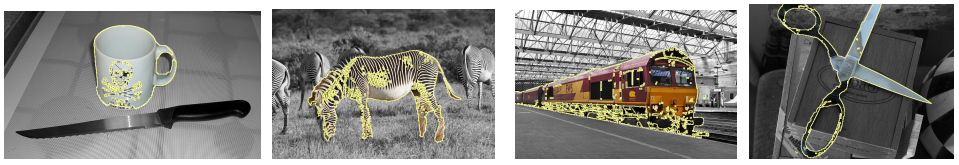


(a) Grabcut segmentation (left) and corrected segmentation by GFA (right).



(b) Tangent profile for grabcut (left) and GFA (right) segmentations.

Fig. 8.2: **Contour regularization.** The GFA normalizes the contour with respect to the elastica energy. This is illustrated by the tangent profile of the grabcut segmentation and the one corrected by GFA.

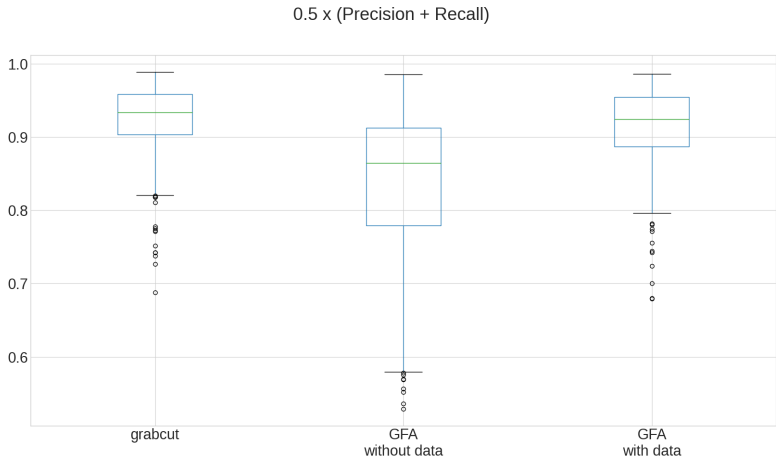


(a) Segmentation by grabcut

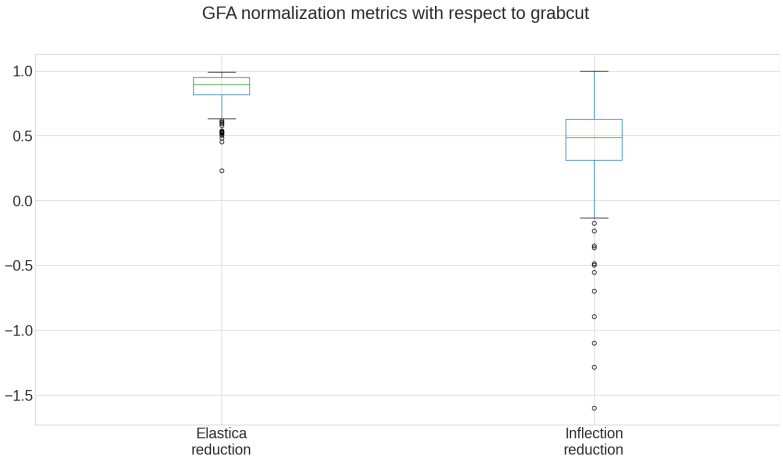
(b) Segmentation corrected by GFA

Fig. 8.3: **Contour completion.** The GFA favors connected components due to the completion effect of the elastica energy. That is particularly useful to avoid oversegmentation artefacts.

9. Conclusion. We presented a discrete shape evolution algorithm driven by the elastica energy. The GFA is built on recent results on the multigrid convergence of curvature and tangent estimators and its main step consists in computing the



(a) Recall and precision results with respect to Coco annotations.



(b) Contour regularization metrics for GFA with respect grabcut.

Fig. 8.4: **Summary statistics.** The GFA give results as good as grabcut with respect to precision and recall, but with a much simpler and easy to describe (and to store) contour.

minimum cut of candidate graphs. A candidate graph is constructed for each element in a neighborhood of shapes of the current digital set and its minimum cut gives a candidate shape. At each iteration, the GFA selects the candidate shape with lowest elastica energy. We have shown that our model can escape local energy minima during the shape evolution by using a very simple neighborhood of shapes. Indeed, our experiments converged to the shape of minimum elastica energy.

Next, we presented some applications in image segmentation. The GFA produces contours with fewer inflection points, smoother tangent profiles and lower elastica energy than those produced by grabcut. That is done while keeping high values of precision and recall with respect to the Coco annotated images used in our experiments.

One of the strengths of our model is that the curvature estimation is based on

(a) Using $r = 5$ ($(P + R)/2 = 0.78$) (b) Using $r = 3$ ($(P + R)/2 = 0.94$)

Fig. 8.5: **Parameter tuning.** The GFA was executed with the same set of parameters for all images, but we can recover better results by tuning the parameter for each image separately. In this example, the low scale of the image asks for a lower estimation radius. The precision plus recall average goes from 0.78 to 0.94 by using a radius of 3 instead of 5.

digital data solely and it is not attached to a curve model, which may restrict the curve evolution and pose technical difficulties regarding its update. Secondly, the GFA is highly parallelizable and we believe that a GPU implementation will greatly reduce the running times of our algorithm . The bottleneck is in the minimum cut computation, as it is difficult to come up with a parallel implementation, but since the latter is computed along a thin band of the shape contour, we believe that this is a minor problem.

There are some possible paths for future work. Firstly, the graph construction part of the algorithm can be optimized. In the current version, the graph is constructed at every iteration, but most of the time, the graph structure changes very slightly and this can be used to optimize its construction. Secondly, we use a very simple neighborhood of shapes in the supervised segmentation problem, i.e., based on dilations and erosions of the initial shape. The contour completion property of the model could be enhanced by employing different neighborhoods. For example, we could elongate the initial shapes in regions of high curvature to obtain a stretched neighborhood of shapes. This could be particularly useful in the segmentation of thin and elongated objects such as blood vessels.

REFERENCES

[1] L. Ambrosio and S. Masnou, *A direct variational approach to a problem arising in image reconstruction*, Interfaces and Free Boundaries, 5 (2003), pp. 63–81.

[2] D. Antunes, J.-O. Lachaud, and H. Talbot, *An elastica-driven digital curve evolution model for image segmentation*, Journal of Mathematical Imaging and Vision, (2020).

[3] D. M. Antunes, J.-O. Lachaud, and H. Talbot, *A maximum-flow model for digital elastica shape optimization*, in Discrete Geometry and Mathematical Morphology, J. Lindblad, F. Malmberg, and N. Sladoje, eds., Cham, 2021, Springer International Publishing, pp. 429–440.

[4] E. Bae, J. Shi, and X.-C. Tai, *Graph cuts for curvature based image denoising*, IEEE Transactions on Image processing, 20 (2010), pp. 1199–1210.

[5] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, *Filling-in by joint interpolation of vector fields and gray levels*, IEEE Transactions on Image Processing, 10 (2001), pp. 1200–1211, https://doi.org/10.1109/83.935036.

[6] Y. Y. Boykov and M. P. Jolly, *Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images*, in Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 1, 2001, pp. 105–112.
[7] K. Bredies, T. Pock, and B. Wirth, *A convex, lower semicontinuous approximation of Euler's elastica energy*, SIAM Journal on Mathematical Analysis, 47 (2015), pp. 566–613, https://doi.org/10.1137/130939493.
[8] A. Bruckstein, A. Netravali, and T. Richardson, *Epi-convergence of discrete elastica*, Applicable Analysis, 79 (2001), pp. 137–171, https://doi.org/10.1080/00036810108840955.
[9] A. M. Bruckstein, R. J. Holt, and A. N. Netravali, *Discrete elastica*, Applicable Analysis, 78 (2001), pp. 453–485.
[10] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic active contours*, International Journal of Computer Vision, 22 (1997), pp. 61–79.
[11] T. F. Chan, S. H. Kang, Kang, and J. Shen, *Euler's elastica and curvature based inpaintings*, SIAM J. Appl. Math, 63 (2002), pp. 564–592.
[12] T. F. Chan and L. A. Vese, *Active contours without edges*, IEEE Transactions on Image Processing, 10 (2001), pp. 266–277.
[13] D. Coeurjolly, J.-O. Lachaud, and J. Levallois, *Integral based curvature estimators in digital geometry*, in Discrete Geometry for Computer Imagery, R. Gonzalez-Diaz, M.-J. Jimenez, and B. Medrano, eds., Berlin, Heidelberg, 2013, Springer Berlin Heidelberg, pp. 215–227.
[14] D. Coeurjolly, J.-O. Lachaud, and T. Roussillon, *Multigrid Convergence of Discrete Geometric Estimators*, Springer Netherlands, Dordrecht, 2012, pp. 395–424.
[15] Y. Duan, W. Huang, J. Zhou, H. Chang, and T. Zeng, *A two-stage image segmentation method using euler's elastica regularized mumford-shah model*, in 2014 22nd International Conference on Pattern Recognition, IEEE, 2014, pp. 118–123.
[16] K. Ecker, *Heat equations in geometry and topology*, Jahresber. Deutsch. Math.-Verein, 110 (2008), pp. 117–141.
[17] N. Y. El-Zehiry and L. Grady, *Fast global optimization of curvature*, in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 2010, pp. 3257–3264.
[18] S. Esedoglu, S. Ruuth, and R. Tsai, *Threshold dynamics for shape reconstruction and disocclusion*, in IEEE International Conference on Image Processing 2005, vol. 2, IEEE, 2005, pp. II–502.
[19] S. Esedoglu, S. J. Ruuth, and R. Tsai, *Threshold dynamics for high order geometric motions*, Interfaces and Free Boundaries, 10 (2008), pp. 263–282.
[20] M. Gage, R. S. Hamilton, et al., *The heat equation shrinking convex plane curves*, Journal of Differential Geometry, 23 (1986), pp. 69–96.
[21] B. Goldluecke and D. Cremers, *Introducing total curvature for image processing*, in 2011 International Conference on Computer Vision, Nov 2011, pp. 1267–1274, https://doi.org/10.1109/ICCV.2011.6126378.
[22] G. Huisken et al., *Flow by mean curvature of convex surfaces into spheres*, Journal of Differential Geometry, 20 (1984), pp. 237–266.
[23] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, International Journal of Computer Vision, 1 (1988), pp. 321–331.
[24] R. Klette and A. Rosenfeld, *Digital geometry: Geometric methods for digital picture analysis*, Morgan Kaufmann, 2004.
[25] J.-O. Lachaud, *Non-Euclidean spaces and image analysis : Riemannian and discrete deformable models, discrete topology and geometry*, PhD thesis, Université Sciences et Technologies - Bordeaux I, Dec. 2006, https://tel.archives-ouvertes.fr/tel-00396332.
[26] J.-O. Lachaud, D. Coeurjolly, and J. Levallois, *Robust and convergent curvature and normal estimators with digital integral invariants*, in Modern Approaches to Discrete Curvature, Springer, 2017, pp. 293–348.
[27] J.-O. Lachaud, A. Vialard, and F. de Vieilleville, *Fast, accurate and convergent tangent estimation on digital contours*, Image Vision Comput., 25 (2007), pp. 1572–1587.
[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft coco: Common objects in context*, in European conference on computer vision, Springer, 2014, pp. 740–755.
[29] S. Masnou and J. M. Morel, *Level lines based disocclusion*, in Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269), Oct 1998, pp. 259–263 vol.3.
[30] B. Merriman, J. K. Bence, and S. Osher, *Diffusion generated motion by mean curvature*, in AMS Selected Lectures in Mathematics Series: Computational Crystal Growers Workshop,

vol. 8966, American Mathematical Society, 1992, pp. 73–83.

[31] D. Mumford, *Elastica and computer vision*, in Algebraic geometry and its applications, Springer, 1994, pp. 491–506.

[32] P. Márquez-Neila, L. Baumela, and L. Alvarez, *A morphological approach to curvature-based evolution of curves and surfaces*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 36 (2014), pp. 2–17.

[33] C. Nieuwenhuis, E. Toeppe, L. Gorelick, O. Veksler, and Y. Boykov, *Efficient squared curvature*, in 2014 IEEE Conference on Computer Vision and Pattern Recognition, June 2014, pp. 4098–4105.

[34] C. Olsson, J. Ulén, Y. Boykov, and V. Kolmogorov, *Partial enumeration and curvature regularization*, in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2936–2943.

[35] C. Rother, V. Kolmogorov, and A. Blake, *"GrabCut": Interactive foreground extraction using iterated graph cuts*, ACM Trans. Graph., 23 (2004), pp. 309–314.

[36] T. Roussillon and J.-O. Lachaud, *Accurate curvature estimation along digital contours with maximal digital circular arcs*, in Combinatorial Image Analysis, J. K. Aggarwal, R. P. Barneva, V. E. Brimkov, K. N. Koroutchev, and E. R. Korutcheva, eds., Berlin, Heidelberg, 2011, Springer Berlin Heidelberg, pp. 43–55.

[37] A. Schindele, P. Massopust, and B. Forster, *Multigrid convergence for the MDCA curvature estimator*, J. Math. Imaging Vis., 57 (2017), pp. 423–438.

[38] T. Schoenemann, F. Kahl, and D. Cremers, *Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation*, in 2009 IEEE 12th International Conference on Computer Vision, Sept 2009, pp. 17–23.

[39] T. Schoenemann, S. Masnou, and D. Cremers, *The elastic ratio: Introducing curvature into ratio-based image segmentation*, IEEE Transactions on Image Processing, 20 (2011), pp. 2565–2581.

[40] P. Strandmark and F. Kahl, *Curvature regularization for curves and surfaces in a global optimization framework*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, Y. Boykov, F. Kahl, V. Lempitsky, and F. R. Schmidt, eds., Berlin, Heidelberg, 2011, Springer Berlin Heidelberg, pp. 205–218.

[41] X.-C. Tai, J. Hahn, and G. J. Chung, *A fast algorithm for Euler's elastica model using augmented lagrangian method*, SIAM J. Img. Sci., 4 (2011), pp. 313–344.

[42] D. Wang, H. Li, X. Wei, and X.-P. Wang, *An efficient iterative thresholding method for image segmentation*, Journal of Computational Physics, 350 (2017), pp. 657–667.

[43] Q. Zhong, Y. Li, Y. Yang, and Y. Duan, *Minimizing discrete total curvature for image processing*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9474–9482.

[44] W. Zhu, X.-C. Tai, and T. Chan, *Image segmentation using Euler's elastica as the regularization*, Journal of Scientific Computing, 57 (2013), pp. 414–438.