Digital Curvature Evolution Model for Image Segmentation

Daniel Antunes, Jacques-Olivier Lachaud, Hugues Talbot

Digital Curvature Evolution Model for Image Segmentation*

Daniel Antunes[1,3] Jacques-Olivier Lachaud[1] Hugues Talbot[2,3]

[1] Université Savoie Mont Blanc, LAMA, UMR CNRS 5127, F-73376, France
daniel.martins-antunes, jacques-olivier.lachaud@univ-savoie.fr
[2] Université Paris-Saclay, CentraleSupélec, Centre de Vision Numérique, INRIA OPIS, F-91190 Gif-sur-Yvette, France
hugues.talbot@centralesupelec.fr
[3] Université Paris-Est, ESIEE Paris, LIGM, UMR CNRS 8049, F-91190 Noisy-le-Grand, France

Abstract. Recent works have indicated the potential of using curvature as a regularizer in image segmentation, in particular for the class of thin and elongated objects. These are ubiquitous in biomedical imaging (e.g. vascular networks), in which length regularization can sometime perform badly, as well as in texture identification. However, curvature is a second-order differential measure, and so its estimators are sensitive to noise. The straightforward extensions to Total Variation are not convex, making them a challenge to optimize. State-of-art techniques make use of a coarse approximation of curvature that limits practical applications. We argue that curvature must instead be computed using a multigrid convergent estimator, and we propose in this paper a new digital curvature flow which mimics continuous curvature flow. We illustrate its potential as a post-processing step to a variational segmentation framework.

Keywords: Multigrid convergence · Digital estimator · Curvature · Shape Optimization · Image Segmentation.

1 Introduction

Geometric quantities are particularly useful as regularizers, especially when some prior information about the object geometry is known. Length penalization is a general purpose regularizer and the literature is vast on models that make use of it [3, 1]. However, this regularizer shows its limitations when segmenting thin and elongated objects, as it tends to return disconnected solutions. Such drawback can sometime be overcome by injecting curvature regularization [7].

One of the first successful uses of curvature in image processing is the inpainting algorithm described in [12]. The authors evaluate the elastica energy along the level lines of a simply-connected image to reconstruct its occluded parts. The non-intersection property of level lines allows the construction of an

* This work has been partly funded by CoMeDiC ANR-15-CE40-0006 research grant.

efficient dynamic programming algorithm. Nonetheless, it is still a challenging task to inject curvature in the context of image segmentation.

The state-of-art methods are difficult to optimize and not scalable [7, 18, 13]. In order to achieve reasonable running times, such approaches make use of coarse curvature estimations for which the approximation error is unknown. Improving the quality of the curvature estimator has an important impact on the accuracy of the results, but is computationally too costly in these methods. Recently, new multigrid convergent estimators for curvature have been proposed [17, 5, 16], motivating us to search for models in which they can be applied.

In this work, we investigate the use of a more suitable curvature estimator with multigrid convergent property and its application as a boundary regularizer in a digital flow minimizing its squared curvature. Our method decreases the elastica energy of the contour and its evolution is evaluated on several digital flows. Finally, we present an application of the model as a post-processing step in a segmentation framework. The code is freely available on github[1].

Outline. Section two reviews the concept of multigrid convergence and highlights its importance for the definition of digital estimators. Next, we describe two convergent estimators used in this paper, one for tangent and the other for curvature. They are used in the optimization model and in the definition of the digital elastica. Section three describes the proposed curvature evolution model along with several illustrations of digital flows. Section four explains how to use the evolution model as a post-processing step in an image segmentation framework. Finally, sections five and six discuss the results and point directions for future work.

2 Multigrid Convergent Estimators

A digital image is the result of some quantization process over an object X lying in some continuous space of dimension 2 (here). For example, the Gauss digitization of X with grid step $h > 0$ is defined as

$$D_h(X) = X \cap (h\mathbb{Z})^2.$$

Given an object X and its digitization $D_h(X)$, a digital estimator $\hat{u}$ for some geometric quantity u is intended to compute $u(X)$ by using only the digitization. This problem is not well-posed, as the same digital object could be the digitization of infinitely many objects very different from X. Therefore, a characterization of what constitutes a good estimator is necessary.

Let u be some geometric quantity of X (e.g. tangent, curvature). We wish to devise a digital estimator $\hat{u}$ for u. It is reasonable to state that $\hat{u}$ is a good estimator if $\hat{u}(D_h(X))$ converges to $u(X)$ as we refine our grid. For example, counting pixels is a convergent estimator for area (with a rescale of h^2); but counting boundary pixels (with a rescale of h) is not a convergent estimator for perimeter. Multigrid convergence is the mathematical tool that makes this

[1] https://www.github.com/danoan/BTools

definition precise. Given any subset Z of $(h\mathbb{Z})^2$, we can represent it as a union of axis-aligned squares with edge length h centered on the point of Z. The topological boundary of this union of cubes is called *h-frontier* of Z. When $Z = D_h(X)$, we call it *h-boundary of* X and denote it by $\partial_h X$.

Definition 1 (Multigrid convergence for local geometric quantites) *A local discrete geometric estimator $\hat{u}$ of some geometric quantity u is (uniformly) multigrid convergent for the family $\mathbb{X}$ if and only if, for any $X \in \mathbb{X}$, there exists a grid step $h_X > 0$ such that the estimate $\hat{u}(D_h(X), \hat{x}, h)$ is defined for all $\hat{x} \in \partial_h X$ with $0 < h < h_X$, and for any $x \in \partial X$,*

$$\forall \hat{x} \in \partial_h X \text{ with } \|\hat{x} - x\|_\infty \leq h, \|\hat{u}(D_h(X), \hat{x}, h) - u(X, x)\| \leq \tau_X(h),$$

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

Multigrid convergent estimators provide a quality guaranty and should be preferred over non-multigrid convergent ones. In the next section, we describe two estimators that are important for our purpose.

2.1 Tangent and Perimeter Estimators

The literature presents several perimeter estimators that are multigrid convergent (see [4, 6] for a review), but in order to define the digital elastica we need a local estimation of length and we wish that integration over these local length elements gives a multigrid convergent estimator for the perimeter.

Definition 2 (Elementary Length) *Let a digital curve C be represented as a sequence of grid vertices in a grid cell representation of digital objects (in grid with step h). Further, let $\hat{\theta}$ be a multigrid convergent estimator for tangent. The elementary length $\hat{s}(e)$ at some grid edge $e \in C$ is defined as*

$$\hat{s}(e) = h \cdot \hat{\theta}(l) \cdot or(e),$$

where $or(e)$ denotes the grid edge orientation.

The integration of the elementary length along the digital curve is a multigrid convergent estimator for perimeter if one uses the λ-MST [10] tangent estimator (see [9]).

2.2 Integral Invariant Curvature Estimator

Generally, an invariant σ is a real-valued function from some space Ω which value is unaffected by the action of some group $\mathfrak{G}$ on the elements of the domain

$$x \in \Omega, g \in \mathfrak{G}, \sigma(x) = v \longleftrightarrow \sigma(g \cdot x) = v.$$

Perimeter and curvature are examples of invariants for shapes on $\mathbb{R}^2$ with respect to the euclidean group (rigid transformations). Definition of integral area invariant and its one-to-one correspondence with curvature is proven in [11].

Definition 3 (Integral area invariant) *Let $X \subset \mathbb{R}^2$ and $B_r(p)$ the ball of radius r centered at point p. Further, let $\mathbb{1}_X(\cdot)$ be the characteristic function of X. The integral area invariant $\sigma_{X,r}(\cdot)$ is defined as*

$$\forall p \in \partial X, \quad \sigma_{X,r}(p) = \int_{B_r(p)} \mathbb{1}_X(x)dx.$$

The value $\sigma_{X,r}(p)$ is the intersection area of ball $B_r(p)$ with shape X. By locally approximating the shape at point $p \in X$, one can rewrite the intersection area $\sigma_{X,r}(p)$ in the form of the Taylor expansion [14]

$$\sigma_{X,r}(p) = \frac{\pi}{2}r^2 - \frac{\kappa(X,p)}{3}r^3 + O(r^4),$$

where $\kappa(X,p)$ is the curvature of X at point p. By isolating κ we can define a curvature estimator

$$\tilde{\kappa}(p) \coloneqq \frac{3}{r^3}\left(\frac{\pi r^2}{2} - \sigma_{X,r}(p)\right), \tag{1}$$

Such an approximation is convenient as one can simply devise a multigrid convergent estimator for the area.

Definition 4 *Given a digital shape $D \subset (h\mathbb{Z})^2$, a multigrid convergent estimator for the area $\widehat{Area}(D,h)$ is defined as*

$$\widehat{Area}(D,h) \coloneqq h^2\, Card\,(D)\,. \tag{2}$$

In [5], the authors combine the approximation(1) and digital estimator (2) to define a multigrid convergent estimator for the curvature.

Definition 5 (Integral Invariant Curvature Estimator) *Let $D \subset (h\mathbb{Z})^2$ a digital shape. The integral invariant curvature estimator is defined as*

$$\hat{\kappa}_r(D,x,h) \coloneqq \frac{3}{r^3}\left(\frac{\pi r^2}{2} - \widehat{Area}\,(B_r(x) \cap D, h)\right).$$

This estimator is robust to noise and can be extended to estimate the mean curvature of three dimensional shapes.

3 Digital Curvature Evolution Model

Our goal is to deform a digital object in order to minimize the elastica energy along its contour. Our strategy is to define the digital elastica by using the elementary length and the integral invariant curvature estimators and minimize its underlying binary energy. However, the derived energy is of order four and difficult to optimize. Therefore we propose an indirect method to minimize it.

3.1 Ideal Global Optimization Model

We evaluate the quality of a boundary by evaluating the elastica energy along it. Let $\kappa(\cdot)$ denote the curvature function evaluated on the contour of some shape X. In continuous terms, the elastica energy is defined as

$$E(X) = \int_{\partial X} (\alpha + \beta\kappa^2) ds, \quad \text{for } \alpha \geq 0, \beta \geq 0.$$

We are going to use the digital version of the energy, using multigrid convergent estimators. The energy, in this case, is also multigrid convergent.

$$\hat{E}(D_h(X)) = \sum_{x \in \partial D_h(X)} \hat{s}(x) \left(\alpha + \beta \hat{\kappa}_r^2(D_h(X), x, h) \right), \tag{3}$$

where $\partial D_h(X)$ denotes the 4-connected boundary of $D_h(X)$. In the following we omit the grid step h to simplify expressions (or, putting it differently, we assume that X is rescaled by $1/h$ and we set $h = 1$).

A segmentation energy can be devised by including some data fidelity term g in (3), but we need to restrict the optimization domain to consistent regions. We cannot properly estimate length and curvature along anything different from a boundary. Let Ω be the digital domain and $\mathcal{T}$ the family of subsets of Ω satisfying the property

$$D \in \mathcal{T} \implies D \subset \Omega \text{ and } 4B(\partial D),$$

where $4B(\cdot)$ is the 4-connected closed boundary predicate.

For some $\gamma > 0$, the segmented region $D^\star$ is defined as

$$D^\star = \operatorname*{arg\,min}_{D \in \mathcal{T}} \sum_{x \in \partial D} \hat{s}(x) \left(\alpha + \beta \hat{\kappa_r}^2(D, x) \right) + \gamma \cdot g(D). \tag{4}$$

In its integer linear programming model [18], Schoenemann restricts the optimization domain by imposing a set of constraints that enforces compact sets as solutions. However, the main difficulty here is the minimization of a third order binary energy. We are going to explore an alternative strategy.

3.2 Nonzero Curvature Identification

We can use the curvature estimator to detect regions of positive curvature. Given a digital object D embedded in a domain Ω, we define its pixel boundary set $P(D)$ as

$$P(D) = \{\, x \mid x \in D, |\mathcal{N}_4(x) \cap D| < 4 \,\},$$

where $\mathcal{N}_4(x)$ denotes the 4-adjacent neighbor set of x (without x). The following optimization regions are important in our process.

$$\begin{array}{lll} O & = P(D) & \text{Optimization region.} \\ F & = D - P(D) & \text{Trust foreground.} \\ B & = \Omega - D & \text{Trust background.} \\ A & = P(F) \cup P(B) & \text{Computation region.} \end{array}$$

Note that our definition of the optimization region guarantees that only connected solutions are produced. The computation region is defined around O for symmetric issues. We proceed by minimizing the squared curvature energy along A with respect to the optimization region O.

$$Y^\star = \underset{Y \in \{0,1\}^{|O|}}{\arg\min} \sum_{p \in A} \hat{\kappa}_r^2(p). \tag{5}$$

We expand the squared curvature esimator for a single point $p \in A$ using (1). Define constants $c_1 = (3/r^3)^2$, $c_2 = \pi r^2/2$. Hence,

$$\begin{aligned} \hat{\kappa}_r^2(p) &= c_1 \cdot \big(\, c_2 - \sigma_{D,r}(p) \,\big)^2 \\ &= c_1 \cdot \big(\, c_2^2 - 2c_2\sigma_{D,r}(p) + \sigma_{D,r}(p)^2 \,\big). \end{aligned}$$

Let $F_r(p) \subset F$ denote the intersection set between the estimating ball applied at p with the foreground region. The subset $Y_r(p) \subset Y$ is defined analogously. Substituting $\sigma_{D,r}(p) = |F_r(p)| + \sum_{y_i \in Y_r(p)} y_i$, we obtain

$$\hat{\kappa}_r^2(p) = c_1 \cdot \left(C + 2\,(|F_r(p)| - c_2) \cdot \sum_{y_i \in Y_r(p)} y_i + \sum_{y_i \in Y_r(p)} y_i^2 + 2 \cdot \sum_{\substack{y_i, y_j \in Y_r(p) \\ i<j}} y_i y_j \right),$$

where $C = c_2^2 - 2c_2 \cdot |F_r(p)| + |F_r(p)|^2$. By ignoring constants and multiplication factors and using the binary character of the variables, problem (5) is equivalent to

$$Y^\star = \underset{Y \in \{0,1\}^{|O|}}{\arg\min} \sum_{p \in A} \left((1/2 + |F_r(p)| - c_2) \cdot \sum_{y_i \in Y_r(p)} y_i + \sum_{\substack{y_i, y_j \in Y_r(p) \\ i<j}} y_i y_j \right). \tag{6}$$

Energy (6) is non-submodular, and minimizing it is a NP-Hard problem in the general case. The QPBOP [15] method provides a partial labeling for the optimization variables with the property that all labeled variables belong to some optimal solution. However, some pixels may be left unlabeled. The optimization method is further discussed in section 3.4. For $r = 3$, evaluation of the model on a digital square produces figure 1a.

(a) (b) (c)

Fig. 1: Figure (a): White pixels denote variables labeled as one by QPBOP, while light gray pixels variables are labeled zero; Figure (b): Inverted labeling; Figure (c): Regions of interest: Background (black); Foreground (dark gray); Computation (light gray); and Optimization (white) regions.

We interpret positive curvature at some point p as a shortage of intersection points between the digital object and the estimating ball. The curvature can be reduced if the estimating ball is pulled towards the interior of the digital object, which is done by removing the highlighted pixels in figure 1a. In other words, the partial labeling is inverted, and unlabeled pixels remain unchanged. Points with negative curvature are detected similarly if we evaluate the model on the digital object complement.

3.3 Digital Curvature Flow

We derive the digital curvature flow by iteratively evaluating model (6) with a slight modification. We extend the computation region to take into account more level sets (ℓ) of the original object. As a practical consequence, zones of high curvature are more likely to be detected, leading to a smaller number of unlabeled pixels by QPBOP.

$$A = \bigcup_{i \leq \ell} \partial F^{-i} \cup \partial B^{-i},$$

where the $-i$ exponent means an erosion by a square of side i. Figure 1c illustrates the different regions of the optimization model.

At each flow step, the model is evaluated twice. In the second evaluation, we take care of concavities. The model is executed on $\overline{D^{+1}}$, the complement of the dilation by a square of side one, and we swap foreground and background

regions. Figure 2 presents several digital curvature flows and table 1 lists the initial and final digital elastica energy for the tested shapes.

We observe that the choice of ball radius (r) and level sets (ℓ) should take into account the image scale. For example, using a radius that is too large might lead to a disconnected intersection zone and the accuracy of the estimator is compromised. This explains the difference in flows in figure 2. In practice, we observe that using a ball of radius 3 is sufficient to produce good results while achieving a reasonable running time.

	Digital Elastica			
	Ball	Triangle	Square	Flower
Initial value	0.156	2.55	1.81	4.196
$\mathbf{r = 3, \ell = 3}$	0.192	0.335	0.286	0.298
$r = 5, \ell = 2$	0.156	0.556	0.423	1.477
$r = 5, \ell = 3$	0.166	0.375	0.321	0.364
$r = 5, \ell = 4$	0.207	0.508	0.311	0.174
$\mathbf{r = 5, \ell = 5}$	0.193	0.52	0.278	0.163
$\mathbf{r = 10, \ell = 10}$	0.216	1.33	0.333	0.159

Table 1: Evaluation of digital elastica ($\alpha = 0$) for start and end curves of the flow. Except for the ball, all the elastica energies were decreased significantly.

3.4 Optimization Method

Let F be a function of n binary variables, i.e.

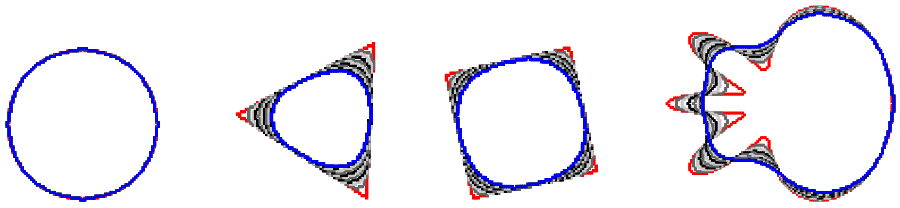
$$F(y_1, \cdots, y_n) = \sum_i F_i(y_i) + \sum_{i<j} F_{i,j}(y_i, y_j)$$

Function F is submodular if and only if the following inequality holds for each pairwise term $F_{i,j}$ [8]

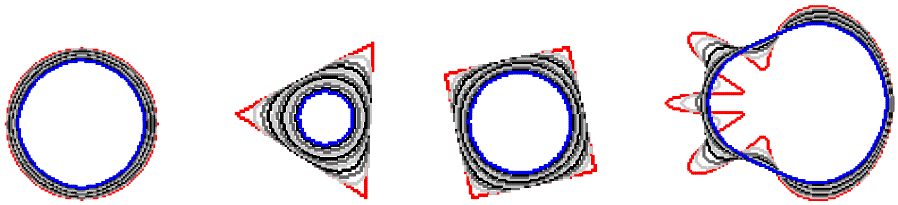
$$F_{i,j}(0,0) + F_{i,j}(1,1) \leq F_{i,j}(0,1) + F_{i,j}(1,0)$$

Energy (6) is non-submodular and optimizing it is a difficult problem, which constrained us to use heuristics and approximation algorithms. The QPBO method [15] transforms the original problem in a max-flow/min-cut formulation and returns a full optimal labeling for submodular energies. For non-submodular energies the method is guaranteed to return a partial labeling with the property that the set of labeled variables is part of an optimal solution. That property is called partial optimality.

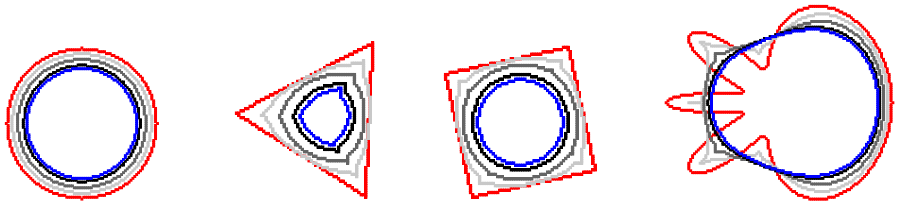
In practice, QPBO can leave many pixels unlabeled. There exist two extensions of QPBO that ameliorate this limitation: QPBOI (improve) and QPBOP



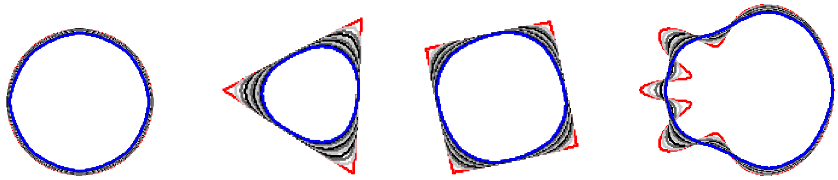
(a) $r = 3, \ell = 3$



(b) $r = 5, \ell = 5$



(c) $r = 10, \ell = 10$



(d) $r = 5, \ell = 5$ (1.5× scaling)

Fig. 2: Digital curvature flow for four different shapes. A total of 20 iterations were executed for each flow, except for (c) (7 iterations). Curves are displayed every 2 iterations. The initial curve (countour of the original shape) is in red and the end curve in blue.

(probe). The first is an approximation method that is guaranteed to not increase the energy, but we lost the property of partial optimality. The second is an exact method which is reported to label more variables than QPBO. We use QPBOP. The extended computation region also regularizes the energy and we have checked that it induces a higher number of labeled variables.

4 Application in Image Segmentation

The digital curvature flow can be applied as a post-processing step in an image segmentation framework. We use graph cut [2] as segmentation method and we execute the flow for n iterations. We include the graph cut data fidelity term g and standard length penalization s to the flow energy.

$$\min_{Y} \sum_{y \in Y} (\alpha \cdot s(y) + \gamma \cdot g(y)) + \beta \cdot \sum_{p \in A} \hat{\kappa}_r^2(p). \tag{7}$$

Let $\mathcal{N}_4(p)$ denote the four neighborhood of pixel p. Length penalization is defined as

$$s(p) = \sum_{p_k \in \mathcal{N}_4(p)} (p - p_k)^2.$$

In Figure 3 we show some results. The flow clearly regularizes the contour of figures produced by the comparison segmentation via graph cut. In both figures, the flow is able to correct zones of high positive curvature and expand regions of low negative curvature, but without invading the background zone. Nonetheless, the flow does not expand zones of convexity. Unfortunately, as we follow a local strategy, we are unable to expand some zones that clearly belongs to the segmented object, like the cow's leg.

5 Conclusion

We have shown that the integral invariant curvature estimator can be integrated into an optimization model and can be applied together with classical penalization terms as length and data fidelity in an image processing task. We demonstrated its potential by designing a digital curvature flow that mimics continuous flow in an accurate way. Finally, we show how it can be used as a post-processing tool in an image segmentation framework.

We have some directions for future work. First, optimize the code and evaluate a runtime analysis to compare with competitor methods. We also think that we can reformulate the model in [18] using the digital estimator $\hat{\kappa_r}$.

References

1. Appleton, B., Talbot, H.: Globally optimal geodesic active contours. Journal of Mathematical Imaging and Vision **23**(1), 67–86 (Jul 2005)
2. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. vol. 1, pp. 105–112 vol.1 (2001)
3. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision **22**(1), 61–79 (Feb 1997)

4. Coeurjolly, D., Klette, R.: A comparative evaluation of length estimators of digital curves. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(2), 252–258 (Feb 2004)
5. Coeurjolly, D., Lachaud, J.O., Levallois, J.: Integral based curvature estimators in digital geometry. In: Gonzalez-Diaz, R., Jimenez, M.J., Medrano, B. (eds.) Discrete Geometry for Computer Imagery. pp. 215–227. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
6. Coeurjolly, D., Lachaud, J.O., Roussillon, T.: Multigrid Convergence of Discrete Geometric Estimators, pp. 395–424. Springer Netherlands, Dordrecht (2012)
7. El-Zehiry, N.Y., Grady, L.: Fast global optimization of curvature. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 3257–3264 (June 2010)
8. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(2), 147–159 (Feb 2004)
9. Lachaud, J.O.: Non-Euclidean spaces and image analysis : Riemannian and discrete deformable models, discrete topology and geometry. Ph.D. thesis, Université Sciences et Technologies - Bordeaux I (Dec 2006), https://tel.archives-ouvertes.fr/tel-00396332
10. Lachaud, J.O., Vialard, A., de Vieilleville, F.: Fast, accurate and convergent tangent estimation on digital contours. Image Vision Comput. **25**(10), 1572–1587 (Oct 2007)
11. Manay, S., Hong, B.W., Yezzi, A.J., Soatto, S.: Integral invariant signatures. In: Pajdla, T., Matas, J. (eds.) Computer Vision - ECCV 2004. pp. 87–99. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
12. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269). pp. 259–263 vol.3 (Oct 1998)
13. Nieuwenhuis, C., Toeppe, E., Gorelick, L., Veksler, O., Boykov, Y.: Efficient squared curvature. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 4098–4105 (June 2014)
14. Pottmann, H., Wallner, J., Huang, Q.X., Yang, Y.L.: Integral invariants for robust geometry processing. Computer Aided Geometric Design **26**(1), 37 – 60 (2009)
15. Rother, C., Kolmogorov, V., Lempitsky, V.S., Szummer, M.: Optimizing binary mrfs via extended roof duality. 2007 IEEE Conference on Computer Vision and Pattern Recognition pp. 1–8 (2007)
16. Roussillon, T., Lachaud, J.O.: Accurate curvature estimation along digital contours with maximal digital circular arcs. In: Aggarwal, J.K., Barneva, R.P., Brimkov, V.E., Koroutchev, K.N., Korutcheva, E.R. (eds.) Combinatorial Image Analysis. pp. 43–55. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
17. Schindele, A., Massopust, P., Forster, B.: Multigrid convergence for the MDCA curvature estimator. J. Math. Imaging Vis. **57**(3), 423–438 (Mar 2017)
18. Schoenemann, T., Kahl, F., Cremers, D.: Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In: 2009 IEEE 12th International Conference on Computer Vision. pp. 17–23 (Sept 2009)

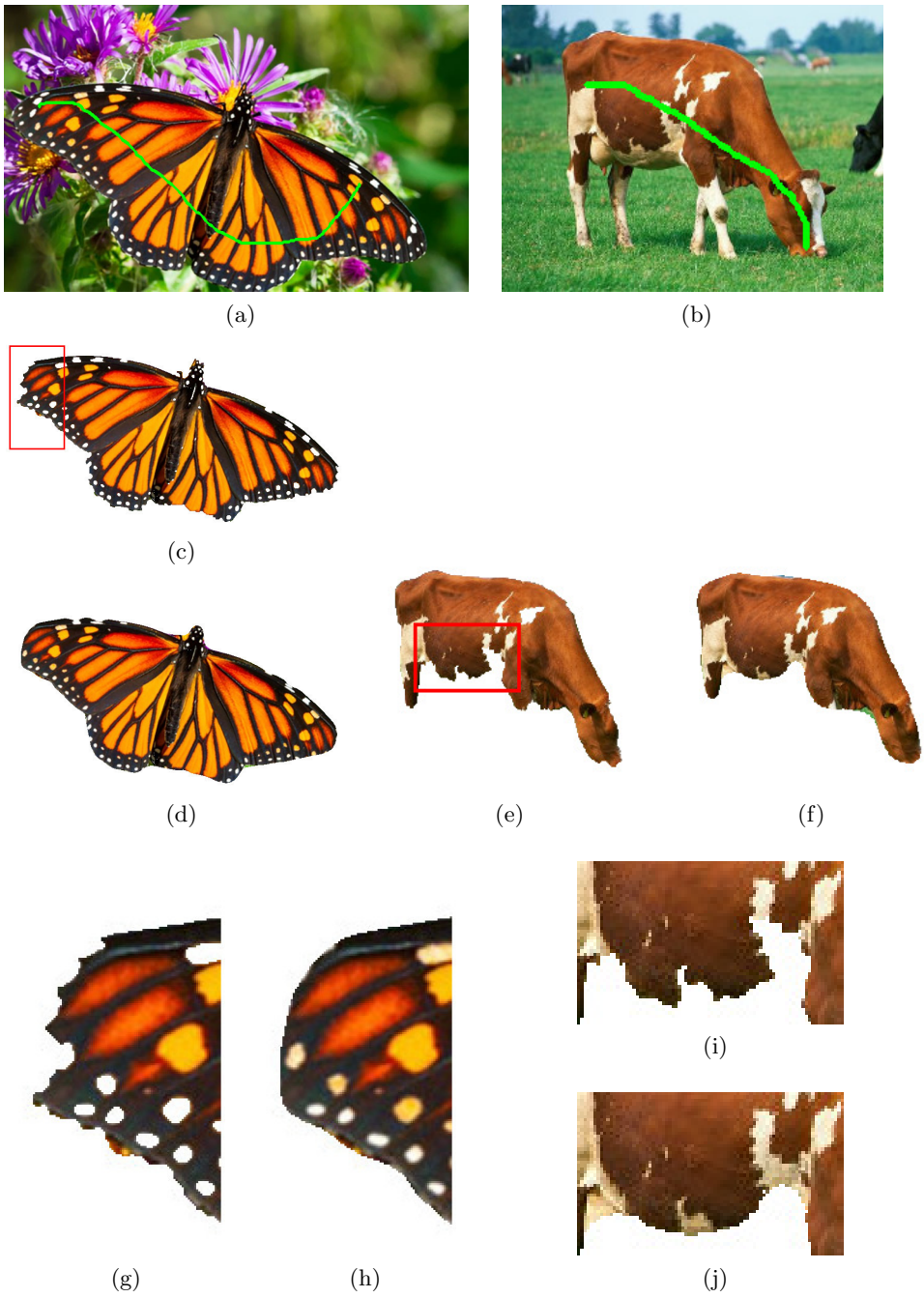


Fig. 3: Digital flow post-processing results for a total of 5 iterations ($\alpha = 0.1, \beta = 1, \gamma = 1$).