

# Chapter 3

## Curvature as regularization

a regularizer

The curvature is a geometric property of curves and it measures the rate of change in curve orientation. A straight line has zero curvature while the curvature at a corner of a square is infinity. The curvature is measured with respect to the arc-length parametrization, and it is invariant to translations, rotations and reflections (rigid transformations).

In imaging, the most widely use of curvature is, perhaps, as a smooth regularization term. That was the case in the Geometric Active Contour and Chan-Vese image segmentation models of Chapter 1. The curvature action in these models is closely related with the *curve-shortening flow* applied to the image level curves. Additionally, the curvature also appears in the derivation of the TV denoising model.

Beyond the smoothing role, curvature can be used to favor connectivity, suggesting that its use could be valuable in the segmentation of thin and elongated objects, a difficult class to standard segmentation models. In image inpainting, the *elastica curve* revealed to be a suitable model to mimic the amodal completion phenomenon, believed to be the process behind the human vision in the perception of occluded objects. However, elastica minimization is a challenging task, mainly because of its non-convexity and the 4<sup>th</sup> order expression emerging from its Euler-Lagrange equation.

In Sections 3.1 and 3.2, we recall the definition of curvature and point out its role in some earlier imaging models. In Section 3.3 we describe the elastica curve, and the challenges involved in the minimization of the elastica energy. We dedicate Section 3.4 to describe discrete models aiming at the minimization of the elastica energy, which is an important topic of this thesis.

### 3.1 Curvature and the curve-shortening flow

is

The curvature is a fundamental geometric property of curves and it is widely present in geometric imaging models. In this section we start by recalling the curvature definition and some of its mathematical representations. Next we describe the *curve-shortening flow* for planar curves, the one dimensional case of the mean curvature flow.

#### Definitions

remainder

In the remain of this section, we assume that every curve is regular, planar and counter-clockwise oriented. Let  $C : [0, L(C)] \rightarrow \mathbb{R}^2$  a curve parameterized by arc-length, i.e.,

$$C(s) = (x(s), y(s)).$$

such that they are respectively

Let  $T(s)$  and  $N(s)$  two orthonormal vectors to  $C(s)$ , i.e., the unitary tangent and normal to  $C(s)$ . Writing  $C(s)$  using these orthonormal vectors we obtain

$$C(s) = (C(s) \cdot T(s)) T(s) + (C(s) \cdot N(s)) N(s)$$

Notice that  $\frac{\partial C}{\partial s} = T(s)$ , therefore

$$\frac{\partial C}{\partial s} \cdot \frac{\partial C}{\partial s} = \left\| \frac{\partial C}{\partial s} \right\|^2 = 1.$$

Derivating the last expression in both sides we obtain

$$\begin{aligned} 0 &= 2 \frac{\partial^2 C}{\partial s^2} \cdot \frac{\partial C}{\partial s} \\ &= 2 \frac{\partial^2 C}{\partial s^2} \cdot T(s). \end{aligned}$$

We conclude that  $\frac{\partial^2 C}{\partial s^2}$  is orthogonal to  $T(s)$ , and that the curvature is given by the normal component of  $\frac{\partial^2 C}{\partial s^2}$ , i.e.,

$$\frac{\partial^2 C}{\partial s^2} = \kappa(s) N(s).$$

We have several formulations for curvature, some of them listed below (for a full derivation see the appendix).

$$\begin{aligned}
 \textbf{Arc-length parameterization:} \quad \kappa(s) &= \left\| \frac{\partial^2 C}{\partial s^2} \right\|. \\
 \textbf{Arbitrary parameterization:} \quad \kappa(u) &= \frac{y'x'' - x'y''}{(x'^2 + y'^2)^{3/2}}. \\
 \textbf{Implicit function:} \quad \kappa(x, y) &= -\frac{f_{xx}^2 - 2f_x f_y f_{xy} + f_{yy}^2}{\|\nabla f\|^3} \\
 &= \nabla \cdot \left( \frac{\nabla f}{\|\nabla f\|} \right).
 \end{aligned}$$

### Curve-shortening flow

Let  $t \geq 0$  and  $u \in [0, U]$ , for some  $U > 0$ . Next, let ~~the~~  $C^{(0)} : [0, U] \rightarrow \mathbb{R}^2$  and  $\mathcal{C}(t, u)$  a family of curves such that

$$\mathcal{C}(0, u) = C^{(0)}(u) \tag{3.1}$$

$$\frac{\partial \mathcal{C}}{\partial t}(t, u) = v(t, u)N(t, u), \tag{3.2}$$

where  $v$  is an arbitrary smooth function. The curve  $C^{(t)}$  is deformed at each point **at speed**  $p$  by an amount  $v$  in the normal direction at  $p$ . Note that any tangent component is irrelevant to curve deformation. From ?? we compute the first variation of curve length of family  $\mathcal{C}$

$$\begin{aligned}
 \frac{\partial}{\partial t} L(t) &= \frac{\partial}{\partial t} \int_0^U \|\mathcal{C}_u\| du \\
 &= \int_0^U \frac{\mathcal{C}_u \cdot \mathcal{C}_{ut}}{\|\mathcal{C}_u\|} du \\
 &= \int_0^U T \cdot (\mathcal{C}_t)_u du.
 \end{aligned}$$

Changing from an arbitrary parameterization  $u$  to an arc-length one and recalling that  $ds = \|\mathcal{C}_u\| du$

hence  $dC_t/ds = \|C_u\| dC_t/du$

$$\frac{\partial}{\partial t} L(t) = \int_0^U T \cdot (\mathcal{C}_t)_s \|\mathcal{C}_u\| du = \int_0^{L(\mathcal{C})} T \cdot (\mathcal{C}_t)_s ds.$$

and  $N_s = -kT$

Using Equation (3.2) we obtain

$$\frac{\partial}{\partial t} L(t) = \int_0^{L(\mathcal{C})} T \cdot (-\kappa v T + v_s N) ds = - \int_0^{L(\mathcal{C})} \kappa v ds = - < \kappa, v > .$$

From which we conclude that length decreases the fastest by choosing  $v = \kappa$ . We define the *curve-shortening flow* (CS flow) of curve  $C^{(0)}(u)$  as the family  $\mathcal{C}(t, u)$  such that

$$\mathcal{C}(0, u) = C^{(0)}(u) \quad (3.3)$$

$$\frac{\partial \mathcal{C}}{\partial t}(t, u) = \kappa(t, u) N(t, u). \quad (3.4)$$

**Example:** Let  $C^{(0)}$  be a circle of radius  $R_0$ , i.e.,  $C^{(0)} = R_0(\cos t, \sin t)$ . From Equation (3.4) we conclude that for every  $t \geq 0$ ,  $C^{(t)}$  is a circle of radius  $R(t)$  where

$$\frac{\partial R}{\partial t} = -\frac{1}{R} \rightarrow R(t) = \sqrt{R_0^2 - 2t}. \quad (3.5)$$

Moreover, the curve collapses to a single point in time  $t = \frac{R_0^2}{2}$ . The curvature flow has many interesting properties [Hui+84; G+86; Eck08]. Among them,

**Comparison principle:** Let  $C_1, C_2$  two closed curves such that  $C_1^{(0)}$  is in the interior of  $C_2^{(0)}$ . Then  $C_1^{(t)}$  is in the interior of  $C_2^{(t)}$  for every  $t$ .

**Convexity preserving:** A convex curve  $C^{(0)}$  stays convex for all  $t$ .

**Point collapsing:** Let  $C^{(0)}$  a closed curve. There exists a time  $t$  in which  $C^{(t)}$  describes a circle and it follows Equation (3.5) until collapsing into a single point.  
(Note: this is true only in 2D)

**Perimeter minimization:** The curvature flow is the continuous deformation that decreases the perimeter of a single closed curve the fastest. at the fastest speed

The CS flow appeared in the Chan-Vese and Geometric Active Contour models for image segmentation in Chapter 1 in its level set formulation [OS88]

$$\frac{\partial f}{\partial t} = \|\nabla f\| \nabla \cdot \left( \frac{\nabla f}{\|\nabla f\|} \right). \quad (3.6)$$

In this case, each of the image level curves describes a CS flow.

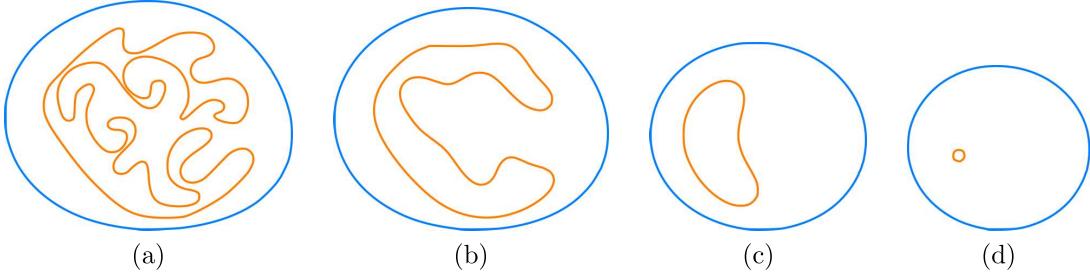


Figure 3.1: **CS flow in action.** CS flow at different times in evolution. In this example we observe the CS flow properties listed in the text.

## 3.2 Diffusion and level curves motion

Several models in [Chapter 1](#) are stated as diffusion processes, but quite often a level curve evolution interpretation is more convenient. In this section we analyze the role of curvature, its properties in earlier models of image processing and how the dual interpretation of these models help us to gain extra insight on them.

### Curvature in denoising and image segmentation

The necessary optimality condition of the total variation denoising energy is given by  
[\(see some ref or see some previous section\)](#)

$$\nabla \cdot \left( \frac{\nabla f}{\|\nabla f\|} \right) + \lambda(f_{\tilde{I}} - f) = 0. \quad (3.7)$$

The solution of [Equation \(3.7\)](#) is the steady state of the anisotropic diffusion

$$u(0, x) = f(x) \quad (3.8)$$

$$\frac{\partial u}{\partial t} = \nabla \cdot \left( \frac{\nabla f}{\|\nabla f\|} \right) + \lambda(f_{\tilde{I}} - f). \quad (3.9)$$

*Note the similarity between [Equation \(3.9\)](#) and the level set formulation of the CS flow in [Equation \(3.6\)](#). If  $\lambda \equiv 0$ , we denote [Equations \(3.8\)](#) and [\(3.9\)](#) the TV flow[BCN02]. Interestingly, both CS flow and TV flow decrease total variation of  $f$ , but in different ways. The CS flow tends to deform the boundary of objects and decreasing total variation by perimeter minimization. The TV flow, on the other hand, preserves the boundary for [longer](#) and it decreases the TV by decreasing the surface height, i.e., the color intensity of the pixels(see [Figure 3.2](#)).*

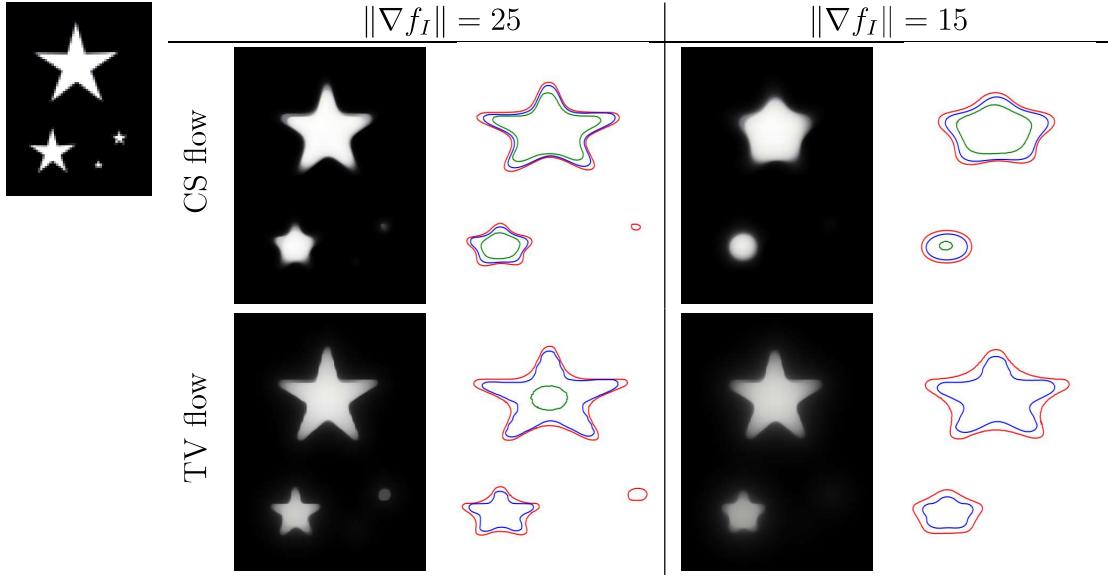


Figure 3.2: **Minimizing TV energy with TV flow and CS flow.** Both flows decrease the image total variation, but CS flow tends to deform object boundaries while doing so. On the other hand, TV flow reduces the image contrast.

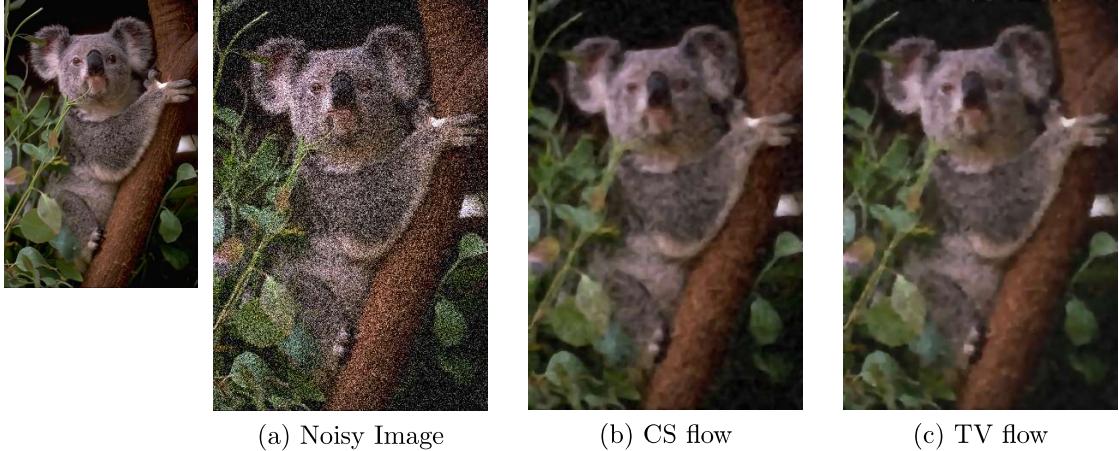
Naturally, the CS flow can also be used to do image denoising. If we think of noise as small artifacts with high gradient of color, the level curves corresponding to noise will have a very small radius, and will collapse faster than other level curves (see Figure 3.3). is it characteristic function or same lambda as above ?

In the geometric active contours model for image segmentation, the level sets of a predefined function  $u$  (e.g., a smoothed version of function  $1 - \lambda_C$  where  $C$  is a set that contains the object to be segmented) describes a CS flow motion modulated by an edge detector function  $g$  (e.g.,  $g = (1 + \|\nabla f\|)^{-1}$ ).

$$\begin{aligned} u(0, x, y) &= u(x, y) \\ \frac{du}{dt} &= g(\|\nabla f\|) \|\nabla u\| \nabla \cdot \left( \frac{\nabla u}{\|\nabla u\|} + v \right), \end{aligned}$$

### Curvature and the connectivity principle applied to inpainting

We recall that the inpainting problem consists into reconstructing a collection of missing patches  $\mathcal{P}$  of the image. The earlier inpainting model [Ber+00] diffuses image information through the patches boundaries. The particularity of this model is that the Laplacian (heat equation) is diffused along the image level lines. Besides despite



**Figure 3.3: TV denoising using CS and TV flow.** Results are quite similar, but TV flow image is sharper, although with lower contrast than CS flow. Images (b) and (c) have the same TV value.

its impressive results, this models does not possess an important property which is expected in inpainting models: the *connectivity principle*.

Studies of the Gestalt school of psychology suggest that the human vision, by a process called amodal completion [Mum94], completes the boundary of partially occluded object with a short and low curvature curve, as illustrated in Figure 3.4a. The model proposed in [CS01] is a TV flow modulated by the curvature of the image level curves.

$$\begin{aligned} u(0, x, y) &= f(x, y) \\ \frac{\partial u}{\partial t} &= \nabla \cdot \left( |\kappa(t)| \frac{\nabla f}{\|f\|} \right). \end{aligned} \quad (3.10)$$

The diffusion in Equation (3.10) is stronger at points of high curvature with respect to its level curves. This model partially achieves the connectivity principle (see Figure 3.5), but the completion is done only for very small regions.

The connectivity principle suggests that the curve connecting the endpoints of an occluded object should minimize some energy with respect to the curvature. Moreover, the segmentation of thin and elongated objects may also benefit from a curvature term, since classical methods have difficulties to give connected solutions. In the next section we describe the Elastica, the curve that minimizes the squared curvature along its length.

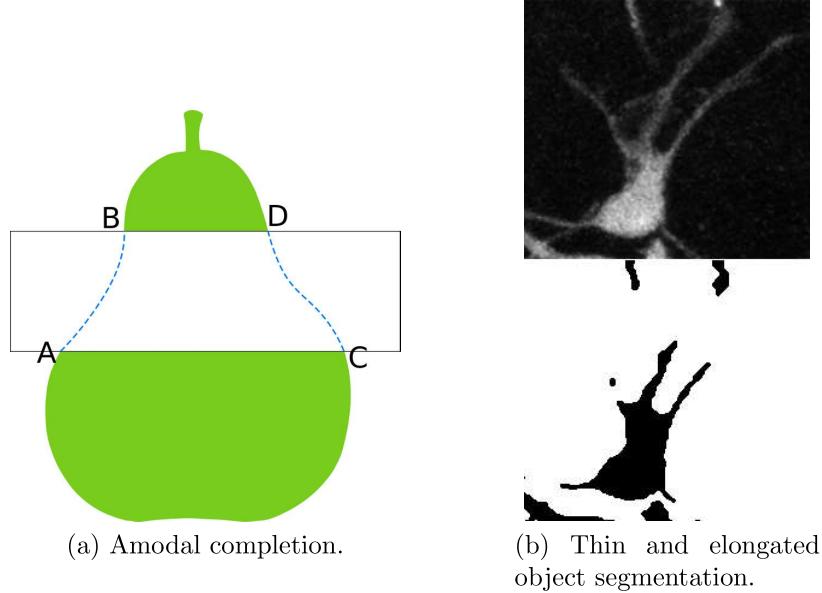


Figure 3.4: **Connectivity principle.** In Figure 3.4a, an illustration of amodal completion, the process followed by the human vision to complete the boundary of occluded object with curves of low curvature [ant](#) that conform the most with the direction of the endpoints. In Figure 3.4b a vessel segmentation with length penalization term only. Curvature favors connected objects.

and



Figure 3.5: **Inpainting by curvature driven diffusions.** [CS01] The diffusion process modeled by [Equation \(3.10\)](#) completes the features disconnected by the inpainting mask.

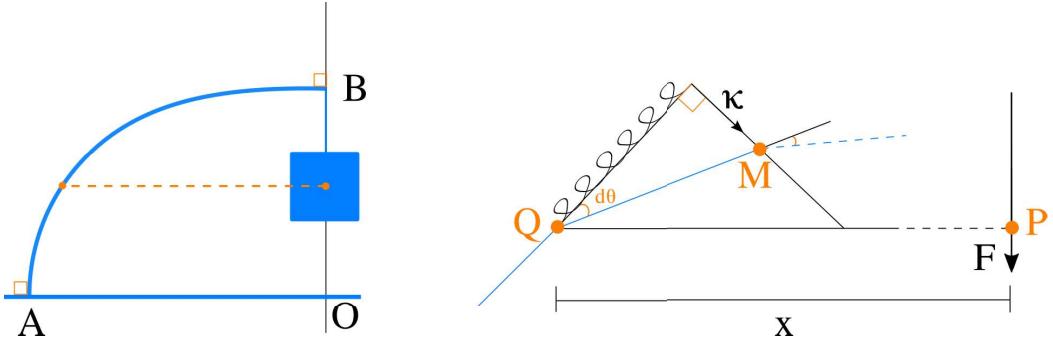


Figure 3.6: **Elastic beam and James Bernoulli's solution.** In equilibrium, the moments at the two ends of the virtual lever  $PQM$  should cancel, implying that the curvature is a linear function of  $x$  if we assume the spring behaves in accordance with Hooke's law.

### 3.3 Elastica curve

In 1691 James Bernoulli pose a challenge to **your** colleagues: What is the form taken by an elastic beam whose ~~its~~-endpoint  $A$  is kept fixed in the ground while a weight is attached to its other endpoint  $B$  such that the tangent at each of the endpoints are perpendicular? A simplified scheme is presented in [Figure 3.6](#).

By using mechanical principles, James concluded that the curvature of such curve at some point  $Q$  can be written as a linear function of the distance from  $Q$  to the line  $BO$ , i.e.,  $\kappa(x) = cx$ . One can show that [[Lev08](#); [Tru60](#)],

$$\frac{\partial y}{\partial x} = \frac{cx^2}{\sqrt{1 - c^2x^4}}. \quad (3.11)$$

The curve modeled by [Equation \(3.11\)](#) is called the *rectangular elastica*. The expression in the right cannot be integrated using elementary functions and is a member of the class of elliptic integrals. Fortunately, Euler has shown that one can express not only the rectangular elastica, but its general version in the form of a minimization principle.

The generalized elastica in 2D is the planar curve  $C$  of fixed length  $L$  whose endpoints have known tangents  $C'(0) = \theta_0, C'(L) = \theta_L$  and minimizes the energy  $\int_C \kappa^2 ds$ .

$$\begin{aligned} & \min_C \int_C \kappa^2 \\ \text{subject to } & \|C\| = L, \\ & C'(0) = \theta_0, \\ & C'(L) = \theta_L. \end{aligned}$$

Given parameters  $\alpha \geq 0, \beta \geq 0$  we incorporate the isoperimetric constraint in the objective function (can be interpreted as a Lagrange multiplier) to define the Elastica energy as

$$E(C) = \int_C \alpha + \beta \kappa^2 ds. \quad (3.12)$$

If we do not impose any constraint in [Equation \(3.12\)](#), its minimization is interesting only if  $\alpha > 0$  and  $\beta > 0$  and we call this problem the *free Elastica*. On the other hand, if one imposes constraints to the minimization of [Equation \(3.12\)](#)(e.g., fixed tangent endpoints), we refer to this problem as the [unconstrained Elastica](#).

One can easily derive the solution of the free Elastica for a closed curve. Setting  $\beta = 0$ , the minimization of [Equation \(3.12\)](#) behaves as the curve-shortening flow of section [Section 3.1](#). On the other hand, if we set  $\alpha = 0$ , the minimum is a disk of infinite radius. For the intermediate case, with both parameters greater than zero, it is easy to see that the solution is a circle  $C_r$  of some radius  $r$ , therefore

$$\begin{aligned} C_r = \arg \min_C E(C) \rightarrow & \frac{\partial}{\partial r} \int_{C_r} \alpha + \beta \kappa^2 ds = 0 \\ & \frac{\partial}{\partial r} (\alpha 2\pi r + \beta \pi / r) = 0 \quad \text{beta 2 pi} \\ & r = \left( \frac{\beta}{\alpha} \right)^{1/2}. \end{aligned}$$

In general, a planar curve  $C$  minimizing the Elastica energy satisfies the following Euler-Lagrange equation [[Cha+02](#); [Sin08](#)]

$$2\kappa_{ss} + \kappa^3 = \frac{\alpha}{\beta} \kappa \Leftrightarrow \frac{\partial^4 C}{\partial s^4} + \frac{\partial^2 C}{\partial s^2}^3 = \frac{\alpha}{\beta} \frac{\partial^2 C}{\partial s^2}. \quad (3.13)$$

This fourth order expression is the main difficult in minimizing the Elastica. In the following we are going to describe some models that attempt to minimize the Elastica and the strategies employed.

### Imaging models using the Elastica

In [Cha+02] the authors propose an inpainting model in which the image level lines interrupted by the inpainting domain are completed by Elastica curves. The key is to define an energy that minimize each of the level curves in terms of the Elastica simultaneously. Given an image  $f_I : \Omega \rightarrow [0, 1]$ , let  $\Gamma_\lambda$  be one of its level curve, i.e.,

$$\Gamma_\lambda = \{x \mid f_I(x) = \lambda\}.$$

Therefore, we minimize each level curve in terms of the Elastica energy by minimizing the functional

$$\int_0^1 \int_{\Gamma_\lambda} (\alpha + \beta \kappa^2) ds d\lambda = \int_0^1 \int_{\Gamma_\lambda} \left( \alpha + \beta \nabla \cdot \left( \frac{\nabla f_I}{\|\nabla f_I\|} \right)^2 \right) ds d\lambda \quad (3.14)$$

Next, let  $dt$  be the length element in the normal direction of the level curves. Therefore,

$$\frac{d\lambda}{dt} = \|\nabla f_I\| \rightarrow d\lambda = \|\nabla f_I\| dt.$$

Replacing the last expression in [Equation \(3.14\)](#) we obtain

$$\int_0^1 \int_{\Gamma_\lambda} \left( \alpha + \beta \nabla \cdot \left( \frac{\nabla f_I}{\|\nabla f_I\|} \right)^2 \right) \|\nabla f_I\| ds dt = \int_{\Omega} \left( \alpha + \beta \nabla \cdot \left( \frac{\nabla f_I}{\|\nabla f_I\|} \right)^2 \right) \|\nabla f_I\| d\Omega. \quad (3.15)$$

The last expression is derived from the fact that  $dt$  and  $ds$  are orthogonal length elements. The authors propose a finite difference scheme for the gradient flow derived from the Euler-Lagrange of [Equation \(3.15\)](#), which happens to be of 4th order. The high order poses some issues with numerical instability, namely the definition of the time step and convergence analysis, which is skipped. The running time is also quite high even for small inpainting domains. Moreover, one can expect a local solution at most, and giving the highly non-convex character of the equation, an acceptable local solution might be very difficult to be found without a conveniently chosen initial solution. Finally, the method tends to produce blurry edges even for small inpainting domains.

The numerical instability arising from [Equation \(3.15\)](#) is lessen by the 3rd order gradient flow proposed in [Bal+01]. The idea is to create a new variable  $\theta$  to

replace the unstable term  $\nabla f_I / \|\nabla f_I\|$  and include a penalization term that forces  $\theta$  to assume the value of its original interpretation. The energy to be minimized is closely related to [Equation \(3.15\)](#) and is written as

$$\int_{\Omega} a |\nabla \cdot \theta|^2 (\alpha + \beta \|k * f_I\|) d\Omega + \int_{\Omega} \|\nabla f_I\| - \theta \cdot \nabla f_I d\Omega, \quad \|\theta\| \leq 1. \quad (3.16)$$

The kernel  $k$  (e.g. Gaussian) smooths image  $f_I$  and is included due to numerical reasons. Nonetheless, [Equation \(3.16\)](#) suffers from similar issues to those of [Equation \(3.15\)](#). Generally, PDE-based methods encounter difficulties in producing solutions with discontinuities, one of the main features of images.

A discrete approach to inpainting is proposed in [\[MM98\]](#). The model's first step is to identify pairs of admissible T-junctions, i.e., pixels in the outer boundary of the inpainting domain that have the same color value and orientation (see [Figure 3.7a](#)). Due to a property of level curves, for every value color value  $\lambda$  there is an even number of T-junctions. Next, let  $J$  be the set of T-junctions and  $\Gamma_j^t$  a curve connecting an admissible pair  $(j, t)$ . Additionally, we denote  $\theta_j, \theta_t$  the respective angles the curve  $\Gamma_j^t$  makes with the associated level set at  $j$  and  $t$ . The authors propose to find a matching  $\mathcal{M} = \{(j, t) \mid j, t \text{ admissible}\}$  and its respective curves such that it minimizes the energy

$$\sum_{(j,t) \in \mathcal{M}} \int_{\Gamma_j^t} \alpha + \beta |\kappa| ds + \theta_j + \theta_t. \quad (3.17)$$

Assuming that one knows the optimal matching  $\mathcal{M}$ , the curves connecting the T-junctions pairs are geodesics. In the plane, that means that the T-junctions are connected by polygonal curves. An ingenious dynamic programming algorithm is conceived by exploiting the causality relation imposed by the non-crossing constraint of the level curves and excellent results are obtained. In particular, the algorithm can reconstruct large inpainting domains and does not produce blurry edges (see [Figure 3.7b](#)). However, [Equation \(3.17\)](#) cannot produce curvy level lines.

The work of [\[MM98\]](#) illustrates an advantage of discrete approaches to image models: discontinuities are naturally implemented. To find a discrete model is not a trivial task, and sometimes a compromise needs to be done in order to achieve efficiency, as it was the case in [\[MM98\]](#), in which the absolute value of the curvature is used instead of its square. In the next section we are going to describe some discrete models for Elastica in its most known form, i.e., using the square of the curvature.

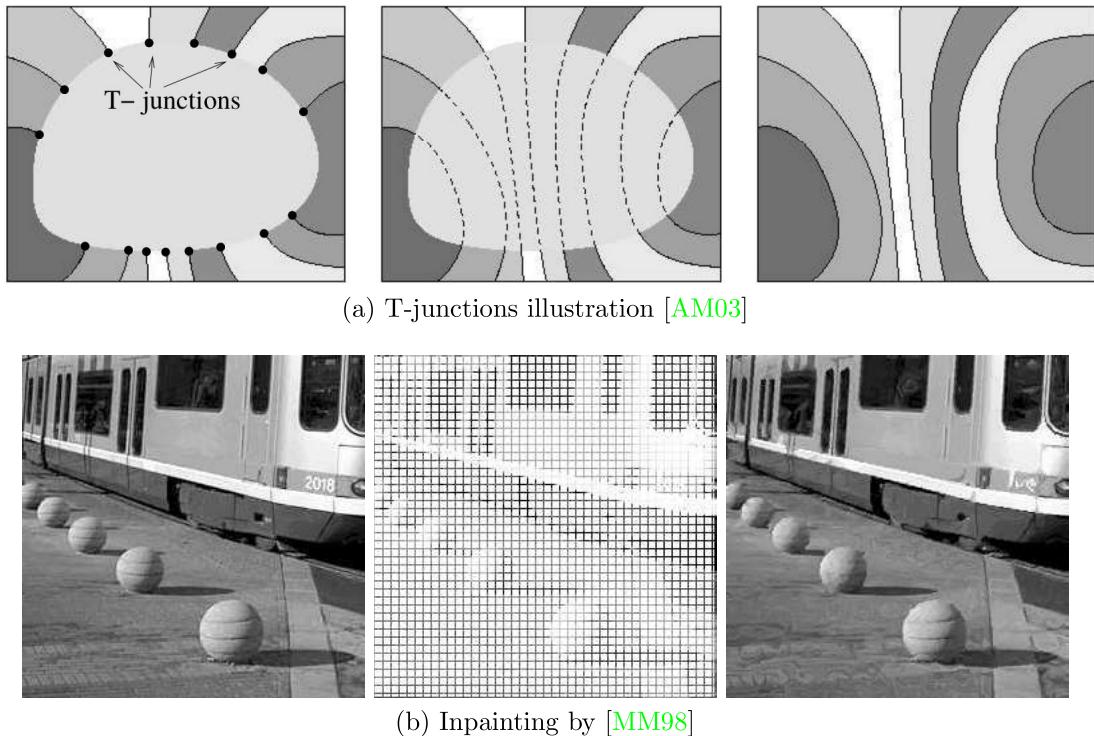


Figure 3.7: **Discrete inpainting.** In Figure 3.7a an illustration of T-junctions and an ideal completion of the level lines; in Figure 3.7b we have the original image in the left, the image to be inpainted in the middle and the result of the discrete inpainting in the right.

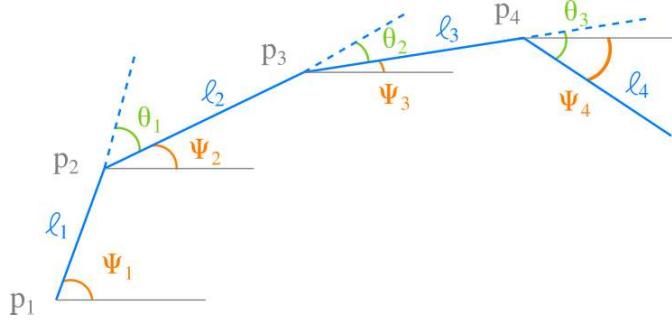


Figure 3.8: **Discrete elastica.** The curvature is approximated by  $\theta_i/l_i$ .

### 3.4 Discrete methods and squared curvature

#### Discrete elastica

Let  $\mathcal{P}_n = \{p_i \mid 1 \leq i \leq n\}$  a sequence of  $n$  points describing a polygonal line. We define the following elements (an illustration is given in Figure 3.8).

$$\begin{aligned} \text{Segment length: } & \ell_i, \quad 1 \leq i \leq n \\ \text{Segment angle: } & \psi_i, \quad 1 \leq i \leq n-1 \\ \text{Angle deviation: } & \theta_i, \quad 1 \leq i \leq n-2. \end{aligned}$$

Given parameters  $\alpha \geq 0, \beta \geq 0$  and a fixed length segment  $\ell$ , the discrete elastica [BHN01] is defined as

$$E(\boldsymbol{\theta}, \ell) = \ell \cdot \sum_{1 \leq i \leq n-2} \alpha + \beta \left( \frac{\theta_i}{\ell} \right)^2 = \sum_{1 \leq i \leq n-2} \alpha \ell + \beta \frac{\theta_i^2}{\ell}. \quad (3.18)$$

Without loss of generality, we assume that the curve starts at the origin with tangent  $\psi_1 = t_1$  and ends at point  $(L, 0)$  with tangent  $\psi_{n-1} = t_{n-1}$ ,  $L$  being the curve length. Then, the general discrete elastica problem is defined as

$$\min_{\psi, \ell} \sum_{1 \leq i \leq n-2} \alpha \ell + \beta \frac{(\psi_{i+1} - \psi_i)^2}{\ell}$$

subject to

$$\begin{aligned} \sum_{1 \leq i \leq n-1} \ell \cos \psi_i &= L, \\ \sum_{1 \leq i \leq n-1} \ell \sin \psi_i &= 0, \\ \psi_1 &= t_1, \\ \psi_{n-1} &= t_{n-1}. \end{aligned}$$

The constraints can be included in the objective function with its respective Lagrange multipliers coefficients and we end up with a nonlinear system. In [BHN01], the authors reported that standard methods as Newton-Raphson return nice approximations of the continuous elastica. In a companion paper, the authors proved that a similar formulation of the discrete elastica, namely

$$E(\boldsymbol{\theta}, l) = \sum_{1 \leq i \leq n-2} \alpha \ell + \beta \left( \frac{\theta_i^2}{\min \{\ell_i, \ell_{i+1}\}} \right) \quad (3.19)$$

is convergent in the sense of  $\Gamma$ -convergence [BNR01], i.e., the minimizer of the discrete elastica approaches the minimizer of the elastica as the number of discretization points goes to infinity.

The discrete elastica was proposed in the context of nonlinear splines with applications in shape design. In imaging, we have to consider additional constraints. For example, digital images are defined in an uniform grid, while the discrete elastica imposes no restriction on the location of the interpolation points in the plane. In fact, this is a fundamental difference between discrete and digital settings. In the list that follows we point out some of the issues one has to handle in order to adapt the discrete elastica to be applied in image segmentation

1. **Data term:** The interpolation points should lie close to the contour of objects and by doing that we lose the implicit description of the curve as a sequence of lengths and angle deviations, being forced to also consider the points coordinates;
2. **Contour complexity:** We are handling closed contours, meaning that we likely have to minimize not one but many elastics along a single contour (not to mention multisegmentation). In practice, that means that we have to chose feature points in the image to force the curve to pass through them;

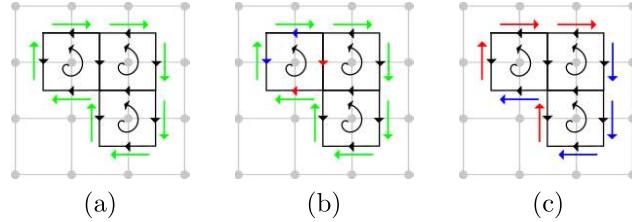


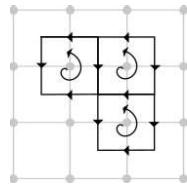
Figure 3.9: **Digital surface and consistency constraints.** In Figure 3.9a, a valid solution made of three cells and a sequence of eight linels. In Figure 3.9b the sum of linel-cell incidence equals to zero (blue are positively incident and red negatively incident). In Figure 3.9c, the sum of linel-edge incidence equals to zero.

**3. Point sampling:** Finally, one has to decide how many points to use in the interpolation and where to position them (a uniform sampling is unlikely a good strategy), bearing in mind the compromise between number of points and computation time.

### Linear programming model for image segmentation using the discrete elastica

In [SKC09], the authors handle the previous list of issues by considering only the discrete elasticas lying on a 2D cellular complex. The 2D cellular complex is composed by non-overlapping surfaces called *cells* and its lower dimension components: *linels* and *pointels*. The cellular **complext** forms a tessellation of the plane and the shape of the cells is arbitrary. The chosen cellular complex has a direct influence in the quality of the discrete elastica, as a finer tessellation covers a larger range of angles (see Figures 3.10a and 3.10b). To simplify exposition, we are going to describe the model for a standard cellular grid complex, in which each cell represents a single pixel in the image.

Let  $n, m$  the number of cells and edges, respectively. We establish a convention on cells and linels orientation. We define the positive orientation of a cell as being counter-clockwise and the positive orientation of linels as being left (horizontal) and down (vertical), as indicated in the figure below.



One binary variable is defined for each cell and two other binary variables are defined for each linel, one for each possible orientation the linel may assume. We

refer to the variables associated to linels as edges. The variables are grouped in vector  $\mathbf{x} \in \{0, 1\}^{n+2m}$  and, hence, a solution is made of active cells and active edges. However, we have to make sure that solutions are consistent, i.e., a solution  $\mathbf{x}$  must encode a digital surface.

A digital surface is composed of a set of cells and a closed path (loop) of edges forming its boundary. To enforce digital surfaces, we define an incidence relation between linels and cells and between linels and edges. This relation is encoded by matrix  $\mathbf{A} \in \{-1, 0, 1\}^{m \times (n+2m)}$

this matrix incorporates cell and linel incidence, no ? this is quite non standard

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \text{if positive incident} \\ -1, & \text{if negative incident} \\ 0, & \text{otherwise.} \end{cases}$$

We say that a linel is positive incident to a cell if their orientations agree and negative incident if they disagree. Similarly, a linel is positive (negative) incident to an edge if they agree (disagree) in orientation. The space of solution is correctly restricted to digital surfaces by imposing

$$\mathbf{Ax} = 0,$$

i.e., the sum of incidences for each linel must equal to zero (see [Figure 3.9](#)). Finally, we set vector  $\mathbf{w} \in \mathbb{R}^{n+2m}$  that is going to hold the data, length and squared curvature penalization terms. For example, data term coefficients are associated to cells variables while length and curvature terms are associated to edges. The complete formulation is written as

$$\begin{aligned} & \min_{\mathbf{x}} \quad \mathbf{w}^t \mathbf{x} \\ & \text{subject to} \quad \mathbf{Ax} = 0 \\ & \quad \mathbf{x} \in [0, 1]^{n+2m}, \end{aligned}$$

where vector  $\mathbf{x}$  was relaxed. The quality of the model naturally depends of the plane tessellation defined by the chosen cellular complex. A simple grid complex is not that interesting, as the discrete elastica can turn only on multiples of  $\pi/2$ . The authors present results for two different cellular complexes. They are equivalent to 8-connectivity or 16-connectivity in a graph-cut framework.

The data term issue is handled by separating the roles of cells and edges. Cells variables holds data terms and edge variables hold curvature and length information. The refinement of the cellular complex is done via pixel subdivision, therefore, every cell lying in the interior of a pixel  $p$  carries data term relative to pixel  $p$ . The consistency constraints guarantees well defined boundaries and the contour

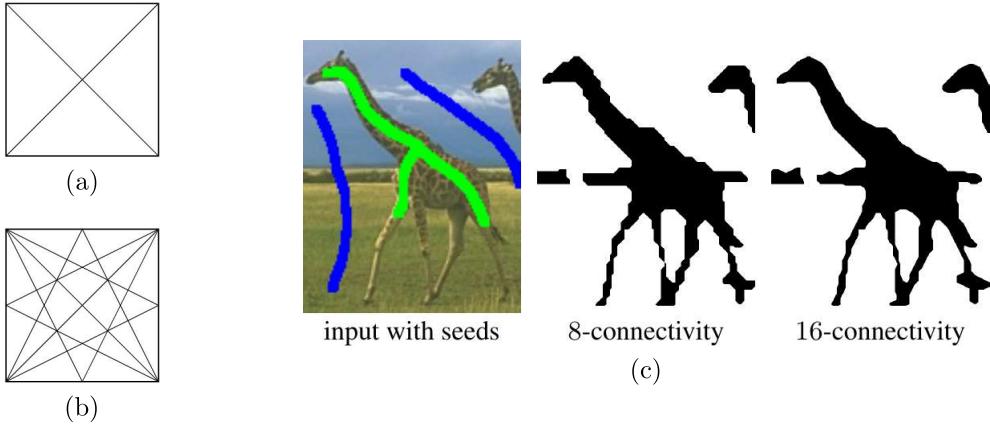


Figure 3.10: **Linear programming model** [SKC09]. In the left, examples of the 8 and 16-connectivity cellular complexes. The authors implementation consists into subdivide a pixel. For example, the 8-connectivity is implemented by rescaling the image twice its size and let the model unit being composed of 4 pixels of the rescaled image. In the right, a result with different connectivities using a data term derived from given foreground (green) and background (blue) seeds.

complexity issue is also covered. Finally, the model is extendable to different angular resolutions while keeping a digital setting, and a discrete elastica with an arbitrary sequence of interpolation points can be well approximated given a good choice of the cellular complex. extensible

Some results are displayed in Figure 3.10c. However, the computation time is quite high and the segmentation present sharp angles even when employing a 16-connectivity cellular complex.

### Unconstrained formulations

The high number of variables and constraints explain the high running time of the linear programming approach. In [EG10], the authors encode the squared curvature value in an unconstrained quadratic non-submodular PBF and propose a model for binary image segmentation.

Let  $I$  be a digital image with  $n$  pixels. We associate to each pixel the binary variable  $\mathbf{x}_i$  indicating if the pixel belongs to foreground ( $x_i = 1$ ) or background ( $x_i = 0$ ). For a given segmentation  $\mathbf{x}$ , one can identify 90 degrees turns in the segmentation contour by inspecting sequential pairs of vertices neighbors (given an order on the neighbors), as illustrated in figure Figure 3.11. The turns can expressed as a 3-clique potential, as shown in the table below

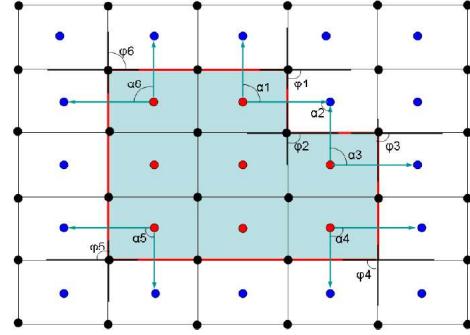


Figure 3.11: **Turn angles detection and triplets** [EG10]. Assuming that red pixels correspond to 1-labeled variables and blue the 0-labeled ones, angle variation is triggered by configurations  $(x_i = 1, x_j = 0, x_k = 0)$  and  $(x_i = 0, x_j = 1, x_k = 1)$ , assuming that  $x_i$  is the centered variable.

$x_i$	$x_j$	$x_k$	$w$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	$w_{ijk}$
1	0	0	$w_{ijk}$
1	0	1	0
1	1	0	0
1	1	1	0

are you sure he is the sole author ?

The coefficients  $w_{ijk}$  are defined accordingly with the Brucksteins discretization formula for the elastica, i.e.,

$$w_{ijk} = \frac{\alpha_i^2}{\min\{|e_{ij}|, |e_{jk}|\}}.$$

Remarkably, the 3-clique potential can be decomposed in three 2-cliques potentials

$$E(x_i, x_j, x_k) = w_{ij}(x_i - x_j)^2 + w_{ik}(x_i - x_k)^2 + w_{jk}(x_j - x_k)^2, \quad (3.20)$$

where

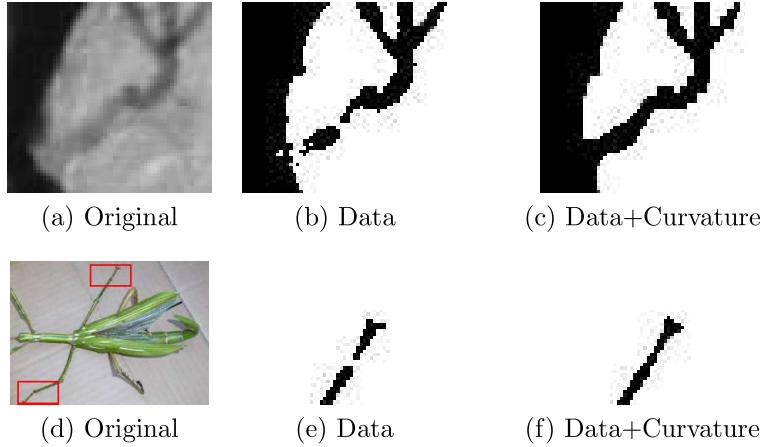


Figure 3.12: **Results for the grid graph model [EG10]**. The original image is displayed in the left; segmentation using only data in the middle; and the segmentation using squared curvature regularization in the right. Connectivity is encouraged, but to a very limited extension.

$$\begin{aligned} w_{ij} &= \frac{1}{2}w_{ijk} \\ w_{ik} &= \frac{1}{2}w_{ijk} \\ w_{jk} &= -\frac{1}{2}w_{ijk}. \end{aligned}$$

Equation (3.20) is a non-submodular PBF (the negative  $w_{jk}$  coefficient creates a non-submodular term) and it is optimized using QPBO or one of its variations. The computation is much faster than the linear programming formulation and the model favors the connectivity principle, although the completion effect seems limited to very small portions (see Figure 3.12). A clear drawback is the limitation in the angle resolution, which tends to produce block artifacts. The authors argue that the model can be extended to any desired connectivity, but it is not clear how this is done.

In [Nie+14] is presented an alternative formulation that is extendable to any desirable angle resolution. The idea is in fact quite similar to the one in [EG10] in the sense that curvature is measured by counting the number of configurations  $(0, 1, 0)$  and  $(1, 0, 1)$  of triplets of vertices. Intuitively, as illustrated in Figure 3.13a, the curvature is higher at regions in which these configurations are more frequent. The authors prove the following theorem

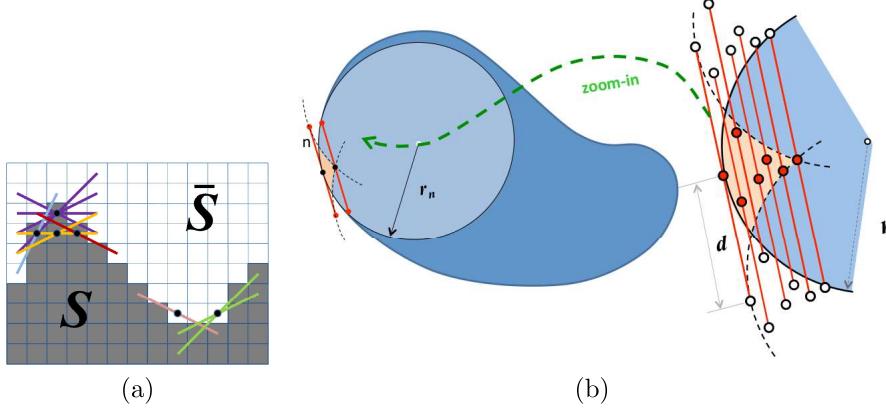


Figure 3.13: **Triplets configurations and integral geometry** [Nie+14]. In Figure 3.13a we intuitively notice that curvature is higher where triplets configurations  $(0, 1, 0)$  and  $(1, 0, 1)$  are perceived. In Figure 3.13b, an illustration of Theorem 1.

**Theorem 1([Nie+14]):** Let contour point  $n$  have tangent orientation  $t$  and osculating ball  $B_r$  of radius  $r = 1/|k|$ . Then, the set of all points  $p \in B_r$  such that  $\|p - n\| \leq r$  and  $(p \pm d \cdot t) \notin B_r$  for given distance  $d < r$  has area

$$A(\kappa, d) = \frac{|\kappa|d^3}{4} + O(d^4).$$

Notice that the theorem is valid for cliques aligned with the tangent at the point of curvature computation. The set of available cliques orientations is defined according to the chosen neighborhood. The neighborhood system of size  $(2d+1) \times (2d+1)$  for pixel  $p$  is similar to a digital circle of radius  $d$  centered at  $p$  (although not the same), in the sense that the neighbors of  $p$  are chosen in order to have approximately the same distance  $d$  from  $p$  (see Figure 3.14). The larger the value of  $d$ , the greater the chances to have a triplet orientation that matches the tangents along the object contour.

Let  $d$  be the chosen neighborhood system and  $c_i$  the corresponding triplets in this neighborhood. Moreover, assume that the triplets are ordered with respect the angle made with the horizontal axis. Let  $\Delta\theta_i$  be the angle difference between two consecutive triplets and  $i(p)$  the triplet index with the closest orientation to the tangent at  $p$ . The integral squared curvature is approximated by

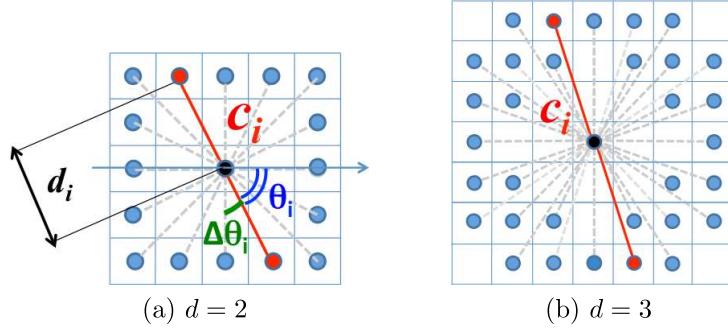


Figure 3.14: **Neighborhood system** [Nie+14]. The higher the value of  $d$ , the higher is the angle resolution.

$$\begin{aligned} \int_C \kappa^2 ds &\approx \sum_p |\kappa_p| \Delta\theta_{i(p)} \\ &= \sum_p \frac{4\Delta\theta_{i(p)}}{d_{i(p)}^3} \cdot A(\kappa_p, d_{i(p)}) \\ &\approx \sum_i \sum_p \frac{4\Delta\theta_i}{d_i^3} \cdot \delta(c_i(p)), \end{aligned}$$

where the function  $\delta$  is one for triplets configurations  $(0, 1, 0)$  or  $(1, 0, 1)$  and zero otherwise. The function `delta` can be expressed as

`\delta`

$$\begin{aligned} \delta(x_a, x_b, x_c) &= x_i(1 - x_j)x_k + (1 - x_i)x_j(1 - x_k) \\ &= x_j + x_ix_k - x_ix_j - x_jx_k. \end{aligned}$$

The resulting energy is non-submodular (the positive term in the  $\delta$  function is non-submodular) and is solved using the LSA-TR method [Gor+14], reported to produce better results than QPBO for some non-submodular instances. Some results for inpainting and segmentation are shown in Figure 3.15. The model is unconstrained, extendable to arbitrary angle resolution and can be easily modified to include data and length terms. Although an argument is provided to justify the curvature approximation, it is difficult to do a multigrid analysis, i.e., to analyse its convergence over different resolutions. The experiments provided by the authors are limited to the computation of the elastica along a disk, and we are not sure about the behaviour of such estimator in shapes of more complex geometry. The fundamental goal of this thesis is to investigate imaging models using geometric

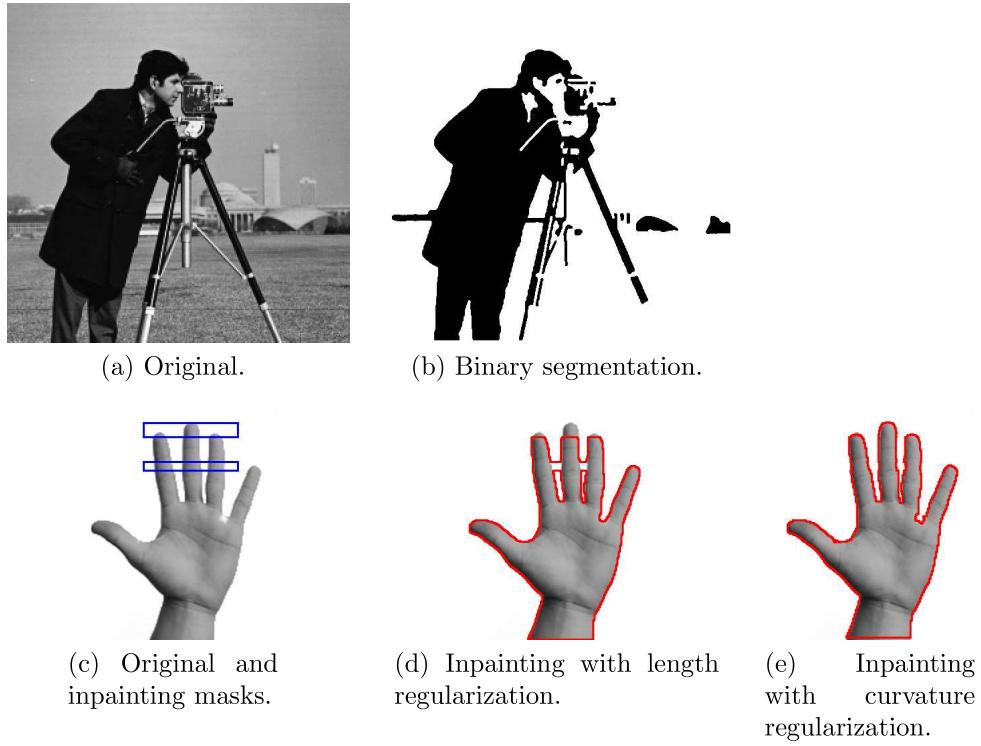


Figure 3.15: **Segmentation and inpainting results** [Nie+14]. The connectivity principle is perfectly illustrated in the inpainting problem but one would expect a completion of the rightmost bar of the camera tripod.

penalization estimated by proven multigrid convergent estimators. In the next chapter we explore the multigrid convergence property and some other concepts of digital geometry.