0.1 Region Competition for Image Segmentation

We consider the problem of image segmentation, that is, the problem of dividing the image into homogeneous regions. The hope is that these regions correspond to objects or object parts in the image. Let \( \Omega \) denote the domain of the image, and \( I : \Omega \to \mathbb{R} \) be the image, then we seek to partition \( \Omega = \bigcup_{i=1}^{N} R_i \) into \( N \) regions such that \( R_i \) are mutually disjoint, i.e., \( R_i \cap R_j = \emptyset \), and the image in each \( R_i \) is homogeneous with respect to some statistic of the image. Note that like the denoising problem, the segmentation problem is ill-posed, and cannot be done without prior assumptions on the regions. For example, one could segment a discrete image by choosing \( R_i = \{ x_i \} \) where \( x_i \in \Omega \), and in this case, each pixel in \( R_i \) (just one pixel) has homogeneous image statistics. However, this is segmentation is not very useful.

We are going to segment the image for now using just the intensity statistics of the image. We make the following assumptions

1. \( I(x) = a_i + \eta_i(x) \) for \( x \in R_i \) where \( a_i \in \mathbb{R} \) and \( \eta_i(x) \sim N(0, \sigma_n) \) is iid \( x \in \Omega \) and independent in \( i \). We assume that \( p(a_i) \propto 1 \) (uniformly distributed), and further \( a_i \) are mutually independent. That is, within each region, the image is roughly constant up to some additive noise. This assumption comes from the fact that any square integrable function (i.e., \( L^2 \) function) can always be approximated up to arbitrary precision with a step function \(^1\).

2. We assume a prior distribution on both \( R_i \) and the number of regions \( N \). For simplicity, we assume a fixed number of regions \(^2\). We assume that

\[
p(R_i) \propto \exp(-\alpha L(\partial R_i)),
\]

where \( L(\partial R_i) \) denotes the length of the boundary of \( R_i \). This assumption is typically made in many works in the literature. It arises from the observation that \( \partial R_i \) could capture places of many pixels of noise (with homogeneous values) and thus fractalize around such noise, which would not represent the boundary of a typical object seen in natural images. Such fractalized boundaries would have large length, and thus the prior assumes that such large length curves are not probable. This assumption also comes from a minimum description length (MDL) formulation, where the objective is to code \( \partial R_i \) with minimal coding length, and the assumption is that the coding length is proportional to \( L(\partial R_i) \). Note however, that the prior has the (undesirable) property of penalizing large objects \(^3\).

\(^1\)A step function is one of the form \( f(x) = \sum_{i=1}^{N} a_i \chi_{R_i}(x) \), where \( R_i \subset \Omega \), \( \chi_{R_i}(x) = 1 \) when \( x \in R_i \) and \( \chi_{R_i}(x) = 0 \) when \( x \notin R_i \), and \( a_i \) are constants.

\(^2\)This is a critical assumption that is unrealistic since in a typical natural image, the number of objects / object parts are unknown; nevertheless, we will assume it since currently, there is no good way of determining the number of regions automatically.

\(^3\)I personally believe this prior is not the correct one to use, however, it is used in the commonly in literature. We will see other priors in later lectures.
We assume that \( R_i \) are mutually independent from each other, and independent from \( a_i \).

With these assumptions, we now are going to estimate \( R_i, a_i \) from the image \( I \) using the Bayesian paradigm and MAP estimation. Thus, we determine \( p(\{a_i, R_i\}_{i=1}^N | I) \):

\[
p(\{a_i, R_i\}_{i=1}^N | I) = \prod_{i=1}^N p(a_i, R_i | I) \quad \text{(independence of \( a_i \) and \( R_i \) in \( i \) and themselves)}
\]

\[
\propto \prod_{i=1}^N p(I | a_i, R_i) p(a_i, R_i) \quad \text{(Bayes Rule)}
\]

\[
\propto \prod_{i=1}^N p(\eta_i(x) = I(x) - a_i, x \in R_i) p(R_i)
\]

\[
= \prod_{i=1}^N \exp \left\{ -\frac{1}{2\sigma_n^2} \int_{R_i} (I(x) - a_i)^2 \, dx \right\} \exp (-\alpha L(\partial R_i))
\]

\[
= \exp \left\{ \sum_{i=1}^N -\frac{1}{2\sigma_n^2} \int_{R_i} (I(x) - a_i)^2 \, dx - \alpha L(\partial R_i) \right\}.
\]

Therefore, the energy is

\[
E(\{a_i, R_i\}_{i=1}^N) = -\log(p(\{a_i, R_i\}_{i=1}^N | I)) = \sum_{i=1}^N \frac{1}{2\sigma_n^2} \int_{R_i} (I(x) - a_i)^2 \, dx + \alpha L(\partial R_i)
\]

For simplicity, we choose \( 1/(2\sigma_n) = 1 \). This energy is the one considered by [6], and the algorithm to minimize it is known as region competition. We shall see the reason for this terminology as we derive the algorithm to minimize \( E \). The case of two regions with better numeric optimization algorithm is considered by [1] (also see [5]).

### 0.2 Minimizing the energy

We are going to derive an iterative algorithm where we start with guesses for \( R_i, a_i \) and then we optimize in \( a_i \) holding \( R_i \) fixed, and then update \( R_i \) holding \( a_i \) fixed. Note that the energy is convex in \( a_i \) since it is just a quadratic function of \( a_i \). Thus, the global minimum \( a_i \) of \( E \) while holding \( R_i \) fixed is computed by solving :

\[
0 = \frac{\partial}{\partial a_i} E(\{a_i, R_i\}_{i=1}^N) = \int_{R_i} 2(a_i - I(x)) \, dx
\]

we thus see that the optimal choice for \( a_i \) is

\[
a_i = \frac{1}{|R_i|} \int_{R_i} I(x) \, dx,
\]

where \( |R_i| \) denotes the area of \( R_i \), that is, \( a_i \) is just the average value of the image inside \( R_i \).

Now we turn to optimizing in \( R_i \) considering all other regions and all \( a_i \) fixed. We are then let to optimizing \( E \) in \( R_i \). The first thing we ask is if this energy is convex in \( R_i \). Note that to define a convex functional, we must have that the underlying space be a convex space. However, the space of regions do not
form a convex space (there is no easy way to make the space of regions even a vector space, that is, define an addition operation on the space of regions). Thus, the energy above is not convex in \( R \). We are thus led to using a steepest descent or gradient descent procedure to optimize the energy, which would mean that we need to compute the Euler-Lagrange equations of \( E \) with respect to \( R_i \). To do this, we need to use some facts of differential geometry of curves.

0.3 Euler-Lagrange equations in \( R_i \)

Note that we seek to minimize and energy of the form

\[
E(R) = \int_R f(x) \, dx + \alpha L(\partial R)
\]

(9)

where \( f : \Omega \to \mathbb{R} \) (in our case \( f(x) = (I(x) - a_i)^2 \)). To do this we suppose that the boundary of the region is simple closed curve, i.e., that \( \partial R \) forms a smooth non-self-intersecting curve, \( c \), that is, \( c = \partial R_i \). Thus the energy above can be written as an energy depending on \( c \):

\[
E(c) = \int_{int(c)} f(x) \, dx + \alpha L(c),
\]

(10)

where \( int(c) \) denotes the region that \( c \) encloses. Thus, in some sense the optimization problem has become simpler since instead of solving for the region, we solve for a curve (which is a smaller set than the region itself).

0.3.1 Basic differential geometry of curves

Let \( c : S^1 \to \Omega \subset \mathbb{R}^2 \) denote a closed, simple curve in the plane. We note that \( S^1 = [0, 1]/\{0, 1\} \) which means that \( S^1 \) is the interval \([0, 1]\) and the endpoints 0 and 1 are considered the same point; this effectively means that \( c(0) = c(1) \) (which makes \( c \) closed). For example,

\[
c(p) = (\cos (2\pi p), \sin (2\pi p)), \quad p \in S^1
\]

(11)

traces out the unit circle in \( \mathbb{R}^2 \). We say that \( c \) parameterizes the unit circle. Note that are many ways (indeed infinitely many parameterizations of a curve) to parametrize the unit circle (or any other curve), for example,

\[
c(p) = (\cos (2\pi p^2), \sin (2\pi p^2)), \quad p \in S^1.
\]

(12)

Note that our energy above only depends on the geometry of the curve and not a particular parameterization of the curve \(^4\).

Note that \( c'(p) = c_p(p) \) is the velocity vector of the curve, which is tangent to the curve. We say that a parametrization of a curve is immersed if \( c_p(p) \neq 0 \) for all \( p \in S^1 \). For such a curve, we can define the unit tangent vector:

\[
T(p) = \frac{c_p(p)}{|c_p(p)|},
\]

(13)

\(^4\)This is desirable since we do not want our algorithm for energy minimization to vary depending on the parameterization we choose; we would like our algorithm to be independent of parameterization. We are interested in the points of the curve not some particular parameterization.
which obviously has norm 1. We define the arclength $s$ of $c$ as
\begin{equation}
    s(p) = \int_0^p |c_p(p)| \, dp, \quad ds = |c_p(p)| \, dp.
\end{equation}
Note that $s(p)$ denotes the length of the curve $c$ traced out from $c(0)$ to $c(p)$, and $ds$ is the infinitesimal arclength parameter. We also note that
\begin{equation}
    L(c) = \int_{S^1} |c_p(p)| \, dp = \int_c ds.
\end{equation}
We define differentiation with respect to the arclength parameter as
\begin{equation}
    \frac{d}{ds} = \frac{1}{|c_p|} \frac{d}{dp}.
\end{equation}
Note then by this definition, we have that
\begin{equation}
    T(p) = \frac{d}{ds}c(p) = c_s(p).
\end{equation}
We note that $|c_s(p)| = 1$. The unit inward normal vector to the curve is
\begin{equation}
    N(p) = JT(p), \quad J = \pm \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}
\end{equation}
the plus/negative denotes the fact that depending on the orientation that the curve is traversed (clockwise or counterclockwise), the inward normal will be either a 90 degree counterclockwise or counterclockwise rotation.

Recall from your physics class that the acceleration vector of the curve $c_{pp}(p)$ contains both a tangential acceleration term and a normal acceleration term, i.e.,
\begin{equation}
    c_{pp}(p) = (c_{pp}(p) \cdot T(p))T(p) + (c_{pp}(p) \cdot N(p))N(p).
\end{equation}
The former measures acceleration along the curve (e.g. the acceleration that can be seen with changes in your speedometer) and the later measures acceleration due to curvature in the path. The tangential acceleration is dependent on the parameterization of the curve. From your physics class, we know that the speed of the curve squared divided by the radius of curvature is the normal component of acceleration:
\begin{equation}
    c_{pp}(p) \cdot N(p) = \frac{|c_p(p)|^2}{R(p)} = |c_p(p)|^2\kappa(p).
\end{equation}
where $\kappa(p)$ is one over the radius of curvature; that is,
\begin{equation}
    \kappa(p) = \frac{c_{pp}(p) \cdot N(p)}{|c_p(p)|^2}.
\end{equation}
We define the curvature vector, $K(p)$, to be the unit normal times the curvature, that is,
\begin{equation}
    K(p) = \kappa(p)N(p),
\end{equation}
and indeed, by direct computation, one can show that
\begin{equation}
    K(p) = c_{ss}(p) = \kappa(p)N(p);
\end{equation}
that is the curvature vector is the second derivative of the curve with respect to arclength.
0.3.2 Euler-Lagrange Equations of Functionals Defined on Curves

We consider now computing the Euler-Lagrange equations of an energy that is defined on curves, that is, of the form \( E : M \to \mathbb{R} \) where

\[
M = \{ c : S^1 \to \Omega : |c'(p)| \neq 0, \ c \text{ is smooth} \}. \tag{24}
\]

Note that we want to calculate the directional derivative of \( E \) at \( c \). In order to do this, we need to know the space of permissible perturbations or the directions of a curve \( c \); we denote the space \( \mathcal{V}_c \) (the subscript denotes that the space is dependent on the curve \( c \)). The space of permissible perturbations are simply vector fields defined on \( c \), that is,

\[
\mathcal{V}_c = \{ h : S^1 \to \mathbb{R}^2 : h \text{ is smooth} \}. \tag{25}
\]

A perturbation deforms the curve \( c \) as follows:

\[
c(p) + th(p), \text{ for } t \text{ small} \tag{26}
\]

(see picture: in class). Note that if \( t \) is small, then \( c + th \in M \), that is,

\[
|(c + th(p))'(p)| \neq 0. \tag{27}
\]

We now define the directional derivative as

**Definition 1.** Let \( E : M \to \mathbb{R} \) be an energy, then the directional derivative of \( E \) at \( c \in M \) in the direction \( h \in \mathcal{V}_c \) is

\[
dE(c) \cdot h = \left. \frac{d}{dt} E(c + th) \right|_{t=0}. \tag{28}
\]

Necessary conditions for a local minimum are obtained by solving for the \( c \) that satisfy

\[
dE(c) \cdot h = 0, \text{ for all } h \in \mathcal{V}_c. \tag{29}
\]

Since the energy \( E \) is non-convex, there is no guarantee that the solution of these equations will lead to a global optimum of the energy. Thus, in general, one uses a gradient descent technique. We now define the gradient (similar to our earlier definition in Lecture 2):

**Definition 2.** The gradient of \( E : M \to \mathbb{R} \) at \( c \) is a permissible perturbation \( g = \nabla E(c) \in \mathcal{V}_c \) such that

\[
dE(c) \cdot h = \int_c h(p) \cdot g(p) \, ds(p) = \int_c h(p) \cdot g(p) \|c_p(p)\| \, dp, \text{ for all } h \in \mathcal{V}_c \tag{30}
\]

**Remark 1.** Note that if we choose \( h = -\nabla E(c) = -g(p) \), then

\[
dE(c) \cdot h = - \int_c |g(p)|^2 \, ds(p) \leq 0, \tag{31}
\]

and so the energy is reduced by moving in the negative gradient direction. Also, note that

\[
\langle h, k \rangle_{L^2} = \int_c h(s) \cdot k(s) \, ds \tag{32}
\]

is an inner product (called the geometric \( L^2 \) inner product). Therefore, by the Cauchy-Schwartz inequality, we have that

\[
|dE(c) \cdot h| \leq \|h\|_{L^2} \|g\|_{L^2}, \text{ or } \frac{|dE(c) \cdot h|}{\|h\|_{L^2}} \leq \|g\|_{L^2}, \tag{33}
\]

which means that \( h = g \) is also the steepest direction with respect to the \( L^2 \) inner product.

The Euler-Lagrange equation for \( E \) is simply \( \nabla E(c) = 0 \).
0.3.3 Gradient Descent of Length

Let us compute the gradient of the length functional $L(c)$ first:

\[
\frac{d}{dt} L(c + th) \bigg|_{t=0} = \frac{d}{dt} \int_0^1 |c_p(p) + th_p(p)| \, dp \bigg|_{t=0} \tag{34}
\]

\[
\int_0^1 \frac{d}{dt} |c_p(p) + th_p(p)| \bigg|_{t=0} \, dp \tag{35}
\]

\[
\int_0^1 \frac{d}{dt} \frac{c_p(p) + th_p(p)}{|c_p(p) + th_p(p)|} \cdot h_p(p) \bigg|_{t=0} \, dp \tag{36}
\]

\[
\int_0^1 \frac{c_p(p)}{|c_p(p)|} \cdot h_p(p) \, dp \tag{37}
\]

\[
= - \int_0^1 \frac{d}{dp} \frac{c_p(p)}{|c_p(p)|} \cdot h(p) \, dp \text{ (integration by parts; closed curve - no boundary terms)} \tag{38}
\]

\[
= - \int_0^1 \frac{1}{|c_p(p)|} \frac{d}{dp} |c_p(p)| \cdot h(p) |c_p(p)| \, dp \tag{39}
\]

\[
= - \int_0^1 h(p) \cdot c_{ss}(p) \, ds(p) \tag{40}
\]

Therefore, we see that

\[
\nabla L(c) = -c_{ss} = K = -\kappa N. \tag{42}
\]

The gradient descent then leads to the PDE

\[
\partial_t c = c_{ss}, \tag{43}
\]

that is, we deform the curve infinitesimally in the negative gradient direction. The above equation is known as curvature flow and sometimes also referred to as the geometric heat equation because of its resemblance to the ordinary heat equation $u_t(x, t) = u_{xx}(x, t)$, which we saw in our lecture in denoising. Note however, the geometric equation is non-linear. This is because, $s$, the arclength variable changes with time $t$:

\[
\partial_t c(t, p) = c_{s(s(t))(t, p)} = \kappa(t, p) N(t, p) = \frac{c_{pp}(t, p) \cdot N(p)}{|c_p(t, p)|^2} N(t, p), \tag{44}
\]

and that as we can see is non-linear. The equation has many interesting properties, and it was of significant interest in the mathematical community [3, 4]. We note a few properties:

1. **Maximum Principle**: Any bounding box that tightly bounds the curve is always shrinking as the curve is evolved under the geometric heat equation. That is, more precisely, if

   \[
   \text{int}(c_1(0, \cdot)) \subset \{(x_1, x_2) : |x_1 - y_1| \leq R_1, |x_2 - y_2| \leq R_2\}
   \]

   where

   \[
   \text{int}(c_1(0, \cdot)) \backslash \{(x_1, x_2) : |x_1 - y_1| \leq r_1, |x_2 - y_2| \leq r_2\} \neq \emptyset \text{ for all } r_1 < R_1, r_2 < R_2.
   \]
\[ \text{int}(c_1(t, \cdot)) \subset \{(x_1, x_2) : |x_1 - y_1| \leq R_1(t), |x_2 - y_2| \leq R_2(t) \} \]

where \( R_1(0) = R_1, R_2(0) = R_2 \) and \( R_1(t), R_2(t) \) are decreasing in time \( t \). The bounding box may be with respect to any orthogonal coordinate system \( x \).

2. Comparison Principle: If \( c_1 \) and \( c_2 \) are simple closed curves \(^5\) and \( c_2(0, \cdot) \subset \text{int}(c_1(0, \cdot)) \) then \( c_2(t, \cdot) \subset \text{int}(c_1(t, \cdot)) \) for all time \( t \) where \( c_1 \) and \( c_2 \) are evolved according to the geometric heat equation.

3. Embeddedness: Any simple closed curve will remain simple under the geometric heat equation.

### 0.3.4 Gradient descent of region-based term

We now turn our attention to minimizing the term

\[ E_r(c) = \int_{\text{int}(c)} f(x) \, dx \quad (45) \]

where \( f : \Omega \to \mathbb{R} \) is some function defined on the domain of the image. Note that \( c \) must be simple for the above energy to make sense. We first write \( E_r \) as an integral around the curve rather so that computing the directional derivative becomes easier:

\[ E_c(c) = \int_{\text{int}(c)} f(x) \, dx = \int_c F(c(s)) \cdot N(s) \, ds \quad (46) \]

where we have applied the Divergence Theorem and \( \text{div} \, F(x) = f(x), x \in \Omega \), and we have assumed that \( \Omega \) is simply connected. Note that such an \( F : \Omega \to \mathbb{R}^2 \) exists, i.e., we can choose \( F = \nabla \phi \) where \( \phi : \Omega \to \mathbb{R} \) satisfies \( \Delta \phi = f \), the later is the Poisson equation and has a solution \(^6\) \([2]\). Here \( N \) is the outward normal vector. Note that \( N = JT = Jc_p(p)/|c_p(p)| \) so that

\[ E_r(c) = \int_0^1 F(c(p)) \cdot Jc_p(p) \, dp. \quad (47) \]

We now compute the directional derivative:

\[ \frac{d}{dt} E_r(c) \cdot h = \left. \frac{d}{dt} E_r(c + th) \right|_{t=0} \]

\[ = \left. \frac{d}{dt} \int_0^1 F(c(p) + th(p)) \cdot J(c_p(p) + th_p(p)) \, dp \right|_{t=0} \]

\[ = \int_0^1 \left. \frac{d}{dt} F(c_p(p) + th(p)) \cdot J(c_p(p) + th_p(p)) \right|_{t=0} \, dp \]

\[ = \int_0^1 (DF(c(p))h(p)) \cdot Jc_p(p) + F(c(p)) \cdot Jh_p(p) \, dp \quad (51) \]

\(^5\)Simple means that there are no self-intersections of the curve.

\(^6\)In the case that \( \Omega \) is rectangular, \( \Omega = [a, b] \times [c, d] \), we have a simple solution, we may choose \( F^1(x) = \frac{1}{2} \int_a^{x_1} f(\xi, x_2) \, d\xi \), and \( F^2(x) = \frac{1}{2} \int_c^{x_2} f(x_1, \xi) \, d\xi \).
where
\[ DF(x) = \begin{pmatrix} \frac{\partial F^1}{\partial x_1}(x) \\ \frac{\partial F^2}{\partial x_2}(x) \end{pmatrix} = \begin{pmatrix} \frac{\partial F^1}{\partial x_1}(x) \\ \frac{\partial F^2}{\partial x_2}(x) \end{pmatrix} \] (52)
is the Jacobian of \( F \). Integrating by parts we find that
\[
dE_r(c) \cdot h = \int_0^1 (DF(c(p))h(p)) \cdot Jc_p(p) - \frac{d}{dp} F(c(p)) \cdot (Jh(p)) \, dp \] (53)
\[
= \int_0^1 (DF(c(p))h(p)) \cdot Jc_p(p) - (DF(c(p))c_p(p)) \cdot (Jh(p)) \, dp \] (54)
\[
= \int_0^1 (Jc_p(p))^T DF(c(p))h(p) - (J^T DF(c(p))c_p(p)) \cdot h(p) \, dp \] (55)
(above we use \( (Ax) \cdot y = x \cdot (A^T y) \) for \( A \in \mathbb{R}^{n \times n}, x, y \in \mathbb{R}^n \))
\[
= \int_0^1 [c_p(p)^T J^T DF(c(p)) - c_p(p)^T DF(c(p))^T J] \, h(p) \, dp \] (56)
\[
= \int_0^1 c_p(p)^T [J^T DF(c(p)) - DF(c(p))^T J] \, h(p) \, dp. \] (57)

Note that \( J^T DF(c(p)) - DF(c(p))^T J \) is in the form \( A - A^T \) where \( A = J^T DF(c(p)) \), and note that
\[
A - A^T = (a_{12} - a_{21})J \] (59)
where
\[
a_{12} = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{\partial F^1}{\partial x_2}(x) \\ \frac{\partial F^2}{\partial x_2}(x) \end{pmatrix} = \frac{dF^2}{dx_2}(x), \quad a_{21} = \begin{pmatrix} -1 & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial F^1}{\partial x_1}(x) \\ \frac{\partial F^2}{\partial x_1}(x) \end{pmatrix} = -\frac{\partial F^1}{\partial x_1}(x), \] (60)
and therefore,
\[
J^T DF(c(p)) - DF(c(p))^T J = \text{div} F(c(p))J = f(c(p))J. \] (61)

Therefore,
\[
dE_r(c) \cdot h = \int_0^1 c_p(p)^T Jh(p)f(c(p)) \, dp \]
\[
= \int_0^1 h(p) \cdot (J^T c_p(p)) f(c(p)) \, dp \]
\[
= \int_0^1 h(p) \cdot \left( J^T \frac{c_p(p)}{|c_p(p)|} \right) f(c(p))|c_p(p)| \, dp \]
\[
= \int h(s) \cdot (f(c(s)) N(s)) \, ds. \]

Therefore, we see that
\[
\nabla E_r(c) = fN \] (62)
where here \( N \) is the outward normal vector. We see that to increase the energy, \( E_r \), one simply moves the curve in the outward normal direction at points where \( f(c(s)) > 0 \) at a speed of \( f(c(s)) \) and in the inward normal direction when \( f(c(s)) < 0 \). So points on the border of \( R \) (i.e., points on \( c \)) are added to the region if \( f > 0 \) and deleted if \( f < 0 \).
0.4 Putting it all together: Region Competition Algorithm

The region competition energy is

\[
E(\{a_i, R_i\}_{i=1}^N) = \sum_{i=1}^N \int_{R_i} (I - a_i)^2 \, dx + \alpha \int_{\partial R_i} ds
\]

(63)

where \(R_i\) are mutually disjoint and \(\cup_{i=1}^N R_i = \Omega\). We denote by \(c_i = \partial R_i\). When \(R_i\) and \(R_j\) are adjacent, we have that \(c_i \cap c_j \neq \emptyset\). Therefore, we see that for points \(x \in c_i \cap c_j\):

\[
\nabla c_i \cap c_j E(\{a_i, R_i\}_{i=1}^N) = \nabla c_i \cap c_j \left(\int_{R_i} (I - a_i)^2 \, dx + \alpha \int_{\partial R_i} ds\right) + \nabla c_i \cap c_j \left(\int_{R_j} (I - a_j)^2 \, dx + \alpha \int_{\partial R_j} ds\right)
\]

\[
= (I - a_i)^2 N_i + \kappa_i N_i + (I - a_j)^2 N_j - \kappa_j N_j
\]

where \(N_i\) is the unit outward normal of \(c_i\). Note that for points in \(c_i \cap c_j\), we have that \(N_i = -N_j\), and thus,

\[
\nabla c_i \cap c_j E(\{a_i, R_i\}_{i=1}^N) = (I - a_i)^2 N_i - 2\kappa_i N_i
\]

\[
= 2(a_j - a_i) \left(I - \frac{a_i + a_j}{2}\right) N_i - 2\kappa_i N_i, \text{ when } R_i \text{ is adjacent to } R_j.
\]

Note the competition between adjacent regions through the competing terms \((I - a_i)^2 N_i\), \((I - a_j)^2 N_i\), hence the name region competition.

Thus, we deduce the following algorithm:

1. Pick \(N\) the number of regions.
2. Guess \(\{R_i^0\}_{i=1}^N\) where \(R_i^0\) are mutually disjoint and \(\cup_i R_i^0 = \Omega\).
3. Compute

\[
a^k_i = \frac{1}{|R_i^k|} \int_{R_i^k} I(x) \, dx.
\]

(64)

4. Update \(R_i^k\): for \(x \in \cup_i \partial R_i^k\), if \(x \in \partial R_i^k \cap \partial R_j^k (i \neq j)\), then

\[
c_i^{k+1}(s) = c_i^k(s) - \Delta t \left(\frac{a_j^k - a_i^k}{2}\right) N_i - \kappa_i(s) N_i(s), \quad x = c_i^k(s)
\]

(65)

(note above that both curves \(c_i\) and \(c_j\) are updated if they overlap).
5. Repeat 3-5 until convergence.

References


