**USING SHAPE DISTRIBUTIONS AS PRIORS
IN A CURVE EVOLUTION FRAMEWORK**

Andrew Litvin and William C. Karl

May 2004

Boston University

Department of Electrical and Computer Engineering

Technical Report No. ECE-2004-03

**BOSTON
UNIVERSITY**

USING SHAPE DISTRIBUTIONS AS PRIORS IN A CURVE EVOLUTION FRAMEWORK

Andrew Litvin and William C. Karl

fg

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
www.bu.edu/ece

May 2004

Technical Report No. ECE-2004-03

This work was partially supported by Grant F49620-03-1-0257, National Institutes of Health under Grant NINDS 1 R01 NS34189, Engineering research centers program of the NSF under award EEC-9986821.

Summary

In this report we describe our framework of constructing and using a shape prior in estimation problems. The key novelty of our technique is a new way to use high level, global shape knowledge to derive a local driving force in a curve evolution context. We capture information about shape in the form of a family of shape distributions (cumulative distribution functions) of features related to the shape. We design a prior objective function that penalizes the differences between model shape distributions and those of an estimate. We incorporate this prior in a curve evolution formulation for function minimization. Shape distribution-based representations are shown to satisfy several desired properties, such as robustness and invariance. They also have good discriminative and generalizing properties. To our knowledge, shape distribution-based representations have only been used for shape classification. Our work represents the development of a tractable framework for their incorporation in estimation problems. We apply our framework to three applications: shape morphing, average shape calculation, and image segmentation.

Contents

1	Introduction	1
2	Prior work	2
3	General formulation	5
3.1	Energy minimization based on curve evolution.	5
3.2	Shape distributions concept	6
3.3	A prior energy based on shape distributions	8
3.4	Feature choice	9
3.5	Gradient flow computation	10
4	Experiments	14
4.1	Shape morphing	15
4.2	Computing average shape	16
4.3	Application to image segmentation	18
5	Conclusions	20
A	Appendix. Analytical computation of the minimizing flows for distribution difference priors	20
A.1	Feature class #1. Collection of inter-point distances	22
A.2	Feature class #2. Multiscale “curvatures”	24

List of Figures

1	An example of constructing a shape distribution for a curve (left panel) based on curvature measured along the boundary. The middle panel shows curvature function $\kappa(s)$ measured along the curve and the right panel shows the sketch of the cumulative distribution function of curvature $H(\kappa)$	6
2	An example of constructing a curvature based distribution representation for a group of 2 shapes. Each shape (left) yields a collection of feature values (middle). Equivalent representation is a cumulative distribution function (right). Combining sets of feature values into a single set is equivalent to averaging a cumulative distribution function. Resulting representations for a group (bottom) are in turn equivalent.	7
3	Examples of features extracted from the shapes on the left panel. Histograms of feature #1 and feature #3 (defined below) are shown in the center and right panels respectively. The shape depicted using certain color and style of the boundary (left panel) generates feature histograms drawn using the same color and line style.	8
4	Feature #1 (left). Set of distances $\{d_{11}..d_{1n}\}$, black lines; Feature #2 (right). Set of support angles $\{\alpha_{11}..\alpha_{1n}\}$. Green dashed curve depicts the shape.	10
5	Illustration of the gradient decent solution to eq. (19)	15
6	Evolution of the contour under the action of our prior flow: initial (blue, dot-dashed), target (green, dashed), and resulting (red, solid) contours. (A) - prior constructed on the inter-point distances (#1); (B) - prior constructed on multiscale curvatures (#2); (C) - Both feature classes are used.	16
7	Average shape calculation using three shape difference measures. Blue solid contours correspond to prior shapes; red dashed line represent the mean shape; red filled areas correspond to the family of solutions. (A) - asymmetric distance based measure; (B) - area based measure. One of the possible solution is shown by green dashed line; (C) - our distribution difference measure (solid red line - evolution result; dashed red line - scaled result).	18
8	Segmentation results. A: Our method; B: Method in [10]; C: Curve length penalty prior; White - final result; Black - true shape boundary; Dashed line - initial curve. Symmetric area distance (in pixels) between true boundary and final result is shown on the top of each panel. . . .	19
9	Illustration of feature value computation for feature class #2	25
10	Sequential computation of the angles for a particular “base” point s_1 , starting from $r = 0$ (assuming inside of the curve is up-wards).	25
11	Local perturbation of the curve at point $\vec{\Gamma}(s_1 + s_2)$. Perturbation $\epsilon\beta(s_1 + s_2)$ is infinitely small comparing to $ \vec{\Gamma}(s_1, s_1 + s_2) $	27

12	Illustration of 2 cases when the sign of the angle increment $d\alpha^{(1)}$ is different for the same curve perturbation $\epsilon\beta(s)$	27
----	--	----

1 Introduction

The use of information about shape is indispensable in numerous applications of image processing, computer vision and other areas. Shape information can appear in a variety of contexts. One example is the use of information about shape as a prior in an inverse problem. Such a prior can allow robust solutions under difficult imaging conditions. Another broad group of applications involves shape analysis, eg: shape classification, and clustering. In these problems, shape information appears in the form of explicit shape descriptions, probability distribution models, etc.

Numerous approaches have been proposed for low level object (shape) description. In this work we concentrate on one approach to describe a shape that is based on the parametric or geometric boundary description as a non-self-intersecting closed contour. Curve evolution approaches [2] to evolve such parameterized shapes are the basic tool used in this work. Curve evolution methods allow convenient handling of object topology and efficient implementation. In a typical curve evolution implementation, the curve is evolved under the combined action of two classes of forces: those dependent on the observed data (data-dependent forces) and those reflecting prior knowledge about the segmented shape or boundary (regularizing forces). We specify a novel such prior force for curve evolution.

Even though a variety of techniques have been developed to capture and use shape information, one of the open questions in the domain of shape modeling is that of capturing perceptual or visual shape similarity and effectively using obtained models as priors. Visually similar shapes share certain features or properties. For example, polygons have straight portions of the boundary and corners. In this work, we attempt to construct a shape modeling framework that would be able to capture such elements of a class of shapes. Such a model must be easily generalizable to unseen but similar shapes. We want to be able to use our model in continuous-valued inference problems as a prior, in particular, in a curve evolution context. In the inverse problem setting, the model having such properties would favor the shapes visually similar to those in the training data. In addition, the desired property for our model is the invariance to geometric transformations. None of the current shape modeling approaches combines these properties in a tractable framework, inspiring our new technique that employs shape distribution representation of shapes and uses it as prior for curve evolution.

The key element of our model, a shape distribution, is a collection of cumulative distribution functions (*cdfs*) of feature values (one 1D distribution per type of feature) measured along the shape boundary or across the shape area [15]. Although it is possible to construct the prior based on joint *cdfs* of different features, in this work we only consider one-dimensional distributions due to the relative simplicity of resulting prior and implementation. The prior is constructed by designing a shape similarity measure penalizing the difference between shape distributions extracted from curves under comparison. Shape distributions have long been used in the computer graphics community to characterize shapes and more recently have been successfully applied to shape classification problems. They were shown to possess desired properties of

robustness, invariance, flexibility. However, to the best of our knowledge, shape distributions have not been used as a prior in estimation problems, and, in particular, have not been used in curve evolution framework.

Using a variational framework, we derive curve flows minimizing our prior energy. We also propose a numerical solution to find an approximate minimizing flow. The overall result is a new flexible and tractable approach to the inclusion of prior shape information into curve-evolution-based framework. We present preliminary results of applying our method to three typical problems, namely, shape morphing, interpolation, and image segmentation.

In the second section we give a detailed overview of existing shape modeling approaches and motivations behind our techniques. In the third section we give a detailed description of our method. In the forth section we present our results and the fifth section concludes this report.

2 Prior work

In the introduction we state our goal of creating such shape modeling approach that would encode or favor visual similarity of shapes. In addition, we seek a model that would be implementable as a prior in a curve evolution context and would easily generalize to unseen shapes. To date, no existing shape modeling technique combines these properties in a single tractable framework. Here we briefly review existing approaches by dividing them into five categories. The approaches in each category only possess some of the properties desired for our model.

1. Methods using a generic prior

In generic regularization methods, certain properties of the shape such as the perimeter or the area are penalized in order to regularize the estimated boundary curve. This group of methods amounts to generic regularization or geometric “low pass” filtering to limit the effects of noise in the image. Such methods do not construct a shape model in an explicit way [13, 2, 26, 7, 21]. The resulting solution can diminish the effect of the noise but also distorts the result, by, for example, smoothing out salient shape features. The important advantage of generic prior methods is that they usually are easily implementable in a curve evolution context.

2. Extensions of methods using generic prior

More recently, geometric flows which drive an evolving curve toward a polygon were developed in [27]. These flows potentially could be used as a prior force in the curve evolution framework. Unfortunately, the geometric flows in [27] favor polygonal shapes with *predefined* (chosen by the operator) edge orientation directions. Such a prior is not data driven, is highly dependent on extrinsic properties (such as object orientation), and does not appear adaptable to other, non-polygonal, shape classes.

It is possible to improve the generic prior so that it includes more information about a class of shapes but is still expressed as a local penalty, stationary with respect to the shape boundary. One such an alternative data-driven prior shape model was proposed in [10] as a part of a level set based segmentation algorithm. A model of the distribution of curvature and intensity with respect to a segmenting curve was found from training data. This spatially stationary model was then used in a maximum a posteriori (MAP) formulation to segment an image. Although giving better results than generic curve length penalty priors, this approach still tends to suppress salient structures. The reason is that the stationary prior coupled with the MAP criterion attempts to drive the curvature at every point on the curve to the same, constant value corresponding to the mode of the distribution. Our method can also include curvature distributions as a particular case of feature choice, but the usage of these distributions is conceptually different in our method.

3. Deformable templates

Numerous approaches have been proposed to construct prior models based on allowable deformations of a template shape. One group of approaches are based on representing and modeling shape as a set of landmarks (see [5] and references therein). In the Point Distribution Model (PDM), proposed in [4], n labeled points on a boundary are selected to describe each shape in the training set. The space of allowable shapes is then defined as a box in $2n$ dimensional space defined by the spread of points in this space, where each point corresponds to one training shape.

A number of approaches use principal component analysis based on parameterized boundary coordinates or level set functions to obtain a set of shape expansion functions [17, 25, 9, 28] that describe the subspace of allowable shapes. In another approach the subspace of allowable shapes is composed of the set of deformations of a predefined shape template [19]. The restricted shape space is then used to constrain the solution, or to compute the likelihood of a particular boundary configuration.

Unfortunately, these methods are extrinsic and can be overly sensitive to the global appearance of particular shapes in the training data. Models relying on explicit shape parameterization do not generalize well to shapes unseen in the training set. These methods are effective when the space of possible curves is well covered by modeled template deformations.

4. Distance measure construction

Methods in this group define a metric on the space of shapes [8]. This metric can be used to construct a probability distribution, to compare shapes, or in another framework involving calculation with an explicit distance definition, such as the Karcher mean. Typically, the distance is constructed on a parametric shape representation, such as the angle function representation [8]. In [8], significant shape features are associated with specific locations along the curve. The model

tends to average representations which results in difficulty capturing existence of prominent features of shape. Typically, methods in this group suffer from the same drawbacks as the deformable templates based methods, namely, difficulty to capture visual shape similarity.

5. Articulated models

Another approach to the inclusion of prior shape information is based on explicit modeling and extraction of component parts [16, 20]. Such models are also known as articulated models. These models can represent well visual similarity within certain classes of shapes, such as human silhouette shapes or human palm shapes. Unfortunately, articulated models only give an ad hoc solution to certain type of problems.

6. PDF construction

Some methods attempt to construct “true” probability distributions on the space of shapes. In one such technique, motivated by the theories of human perception, Zhu [29] developed maximum entropy models of shape. These models are probabilistic and flexible and thus seemingly good candidates for the shape prior. Zhu’s model was used for segmentation in [11] with moderate success despite fundamental problems in using it in a curve evolution framework. In addition, construction of the model is too computationally demanding for practical use.

Indeed, no current technique is able to combine the properties that we seek for the prior shape model. Our goal is to combine advantages of existing methods in a novel formulation. In particular, we want to have a model that is implementable in a curve evolution framework. Our model must be able to include high level information about prior data as in deformable template models and must be able to encode visual shape similarity as articulated models or Zhu’s model [29]. We aim for a compromise between the focusing ability of the prior model, its generalizability, and its cost to find and implement. Motivated by the ideas in [29, 15], we construct a shape boundary prior as an energy which penalizes the difference between set of feature *distributions* of a given curve and those of the prior.

Now we consider the motivations to choosing shape distributions as a shape representation in our framework. Intuitively, feature distributions capture the existence of certain visual features of the shape regardless the location of these features. This invariance to the location of particular visual cues intuitively leads to capturing visual shape similarity. As an example, [29] shows that a shape with narrow protrusions, similar to those encountered in the contours of animals, will look like the shape of an animal. An important motivation for choosing shape distributions, besides the intuitive consideration, is the successful application of these representations and related shape difference measures in shape classification tasks [6, 15, 14]. A related notion of shape context was developed and applied to shape classification tasks in [1]. These results indicate that feature distributions are robust, invariant, flexible shape representations with good discriminative properties.

Unlike using a conventional curve length penalty, our prior energy term depends on the segmenting curve in a non-local way, making computation of a gradient curve evolution descent flow with respect to such a prior term challenging. We show how the energy minimizing curve flow can be computed analytically for some feature classes considered in this work. We then propose an approximate discrete numerical solution for the flow. Such a solution can be used for feature classes for which an analytical gradient is not available. The first step of this computation is based on solving a histogram modification differential equation, which is motivated by, and similar to that proposed in [18]. Our framework gives the energy interpretation of the histogram modification PDE proposed in [18] (which has not been done previously). At the second step we perform projection of the flow in the space of features onto the manifold corresponding to allowable shape deformations.

3 General formulation

In this part, we present our framework of building a shape prior based on the design of an energy (or dissimilarity measure) that penalizes differences between shape distributions.

3.1 Energy minimization based on curve evolution.

We adopt the widely used energy based curve evolution framework that we apply to different applications. In this framework the 2D object (shape) sought is defined by a closed curve. A curve is evolved under the action of forces trying to minimize an energy. The steps involved in solving a particular problem are the following:

1. The solution (curve) is sought as the minimizer of the energy $E(C)$

$$C^* = \underset{C}{\operatorname{argmin}} E(C) \quad (1)$$

2. The energy typically consists of data term(s) and shape prior term(s).

$$E(C) = E_{data}(C) + \alpha E_{prior}(C) \quad (2)$$

The data term E_{data} favors the fidelity of the solution to the data. This term is application specific. For example, for image segmentation problem, this term can be the minus log-likelihood of image intensities. For tomography problem, it will depend on projections, etc. In this work we concentrate on the prior term E_{prior} which reflects the information about shape. α is a regularization parameter that weighs the strength of the prior.

Energy minimization can be interpreted as a MAP problem; however, in this work we do not exploit such interpretation.

Figure 1: An example of constructing a shape distribution for a curve (left panel) based on curvature measured along the boundary. The middle panel shows curvature function $\kappa(s)$ measured along the curve and the right panel shows the sketch of the cumulative distribution function of curvature $H(\kappa)$.

3. The gradient curve flow minimizing eq. 2 is found as

$$C_t = -\nabla E(C) \quad (3)$$

where t is the artificial time parameter. The curve flow can be interpreted as a force acting on a curve. The curve is then evolved as follows:

$$C^{n+1}(s) = C^n(s) + C_t(s)\vec{N}(s) \quad (4)$$

where $\vec{N}(s)$ is the normal direction to the curve C at the location s (arc-length parameterization).

4. We implement the curve evolution via level set framework. It allows easy incorporation of curve forces defined directly on level set function, implicit handling of topology, and convenient curve re-sampling (needed to compute the prior force).

3.2 Shape distributions concept

Distributions of features measured over shapes in a stationary, uniform way, are called shape distributions [15]. As shown by recent shape classification experiments, such shape distributions can capture the intuitive similarity of shapes while being robust to a small sample size, invariant and flexible.

For example, a method using shape distributions in handwritten digits recognition experiments yielded best performance [15] among other published techniques. In addition, some theories of human perception suggest that matching distribution of features is an important part of the perception of shapes ([23], [29] and references therein).

In a continuous formulation, shape distributions for a shape are defined as a set of cumulative distribution functions of feature values (one distribution per collection of feature values of the same kind) measured along the shape boundary or across the shape area. Joint *CDFs* of multiple features can also be considered, although

Figure 2: An example of constructing a curvature based distribution representation for a group of 2 shapes. Each shape (left) yields a collection of feature values (middle). Equivalent representation is a cumulative distribution function (right). Combining sets of feature values into a single set is equivalent to averaging a cumulative distribution function. Resulting representations for a group (bottom) are in turn equivalent.

in this work we only consider one-dimensional distributions. An illustrative example of shape distribution is shown in figure 1, using boundary curvature as a feature. Curvature values are measured uniformly along the boundary (left panel). The middle panel shows the obtained curvature values as a function of arc-length and the right panel shows the sketch of the cumulative distribution function (shape distribution) of obtained curvature values.

We define the shape distribution for a group of shapes as an average of the cumulative distribution functions corresponding to individual shapes in the group. Averaging cumulative distribution functions is equivalent to combining feature value representations (continuous functions or discrete sets of values) measured on individual shapes into a single set. This process is illustrated in figure 2. It is important to note that the resulting shape distribution for a class of shapes does not necessarily correspond to any particular instance of shape. However, we can search for the closest shape, in terms of cumulative distribution function, to the average distribution for a given class. The shape defined in this way, for a given shape distribution is not necessarily unique, although any shape or a class of shapes uniquely defines a shape distribution. Such non-uniqueness of the shape corresponding to the given shape distribution is actually a feature of shape distribution based representation.

Feature classes for a particular application are chosen according to the visual

Figure 3: Examples of features extracted from the shapes on the left panel. Histograms of feature #1 and feature #3 (defined below) are shown in the center and right panels respectively. The shape depicted using certain color and style of the boundary (left panel) generates feature histograms drawn using the same color and line style.

cues that need to be characterized in this application. For instance in characterizing elongated objects it is reasonable to capture distances between opposite points on the boundary, while in characterizing polygons, boundary curvature is a more logical choice of feature class. We believe more work can be done to automate the choice of optimal feature classes.

The ability of shape distributions to capture intuitive similarity of shapes is qualitatively illustrated by the following example. In Figure 3 we show histograms computed over sets of feature values for 2 feature classes extracted from four different shapes associated with two classes. Two shapes are triangles and two are similar looking elongated objects. The two feature classes used here are the set of distances between all nodes of uniformly discretized curve and the set of curvatures at different scales (more rigorously defined below). It can be seen that the histograms calculated from different classes of shapes are significantly different, while the histograms for the objects of the same kind are closer to each other. We show histograms and not cumulative distribution functions for convenience of comparison.

3.3 A prior energy based on shape distributions

We now introduce our formulation of the shape prior in the continuous domain. In the case of a continuously defined feature $\Phi \in \Omega$ (example: Φ - curvature, Ω - arc length along the curve), let λ be a variable spanning the range of values (Λ) of the feature. Let $H(\lambda)$ be the CDF of Φ :

$$H(\lambda) = \frac{\int_{\Omega} h\{\Phi(\Omega) < \lambda\} d\omega}{\int_{\Omega} d\omega} \quad (5)$$

where $h(x)$ is the indicator function.

We define the *pdf* on the space of shapes as Gibbsian distribution with energy penalizing the difference between shape distributions of a given curve and target

shape distributions. Probability of the curve C is

$$p(C) = e^{-E(C)} \quad (6)$$

where

$$E(C) = \sum_{i=1}^M w_i \int_{\Lambda} [H_i^*(\lambda) - H_i(C, \lambda)]^2 d\lambda \quad (7)$$

where M is the number of feature classes taken into account, $H_i(\lambda)$ is the distribution function of the i^{th} feature class and the set of weights w_i weighs the importance of individual feature classes (set to be equal in the experiments carried out in this work but can be used to give more weight to particular feature classes if needed). Target distributions H_i^* can correspond to 1 shape, an average over group of shapes or given by a priori knowledge.

In case when target distributions correspond to 1 shape, eq. 7 can be expressed as a distance between two curves. For two curves C_1 and C_2 , measure of dissimilarity is

$$d(C_1, C_2) = \sum_{i=1}^M w_i \int_{\Lambda} [H_i(C_1, \lambda) - H_i(C_2, \lambda)]^2 d\lambda \quad (8)$$

The main requirements for the distance are to be a metric, and to be differentiable. Both requirements are satisfied under this definition of the distance. This measure is exactly the one used in [15] in the shape classification framework.

In eq. (7) and (8), differences between distributions are weighted uniformly across the range Λ of the value of a feature. However, non-uniform weighting can be introduced in order to emphasize different parts of CDF. For example, such weighting can be naturally given by variance between feature distributions of prior shapes.

In practice, we compute the feature values, corresponding shape distributions, and evolve curves by discretizing curves (sampling curves uniformly). For the discretized curve, instead of continuous feature function $\Phi_i \in \Omega$, we compute a set of values F_i . Corresponding shape distributions are computed on discrete sets of values as follows

$$H(\lambda_j) = \frac{1}{M} \sum_{n=1}^M h\{F_i[n] < \lambda_j\} \quad (9)$$

where M is the number of elements in the set F^i . In order to obtain the value of $H(\lambda)$ for any λ , we use linear interpolation.

3.4 Feature choice

We term a “feature class” a rule of extracting measures (feature values) from a shape. We choose rules that represent measures uniformly sampled along the shape boundary or volume. The factors governing the choice of feature classes for a particular problem include the ability of a chosen feature class to capture the important visual characteristics of the shapes in a particular problem. Another factor in choosing the

Figure 4: Feature #1 (left). Set of distances $\{d_{11}..d_{1n}\}$, black lines; Feature #2 (right). Set of support angles $\{\alpha_{11}..\alpha_{1n}\}$. Green dashed curve depicts the shape.

feature class is computational complexity. Separate feature classes capturing different visual characteristics of shapes can be combined in a single framework, creating a more versatile prior. Here we define two feature classes (#1 and #2) that we use in our experiments:

- **Feature # 1.** Inter-node distances.

The feature vector is composed of distances between all combinations of nodes of the discrete curve. The distances can be further normalized by the mean distance, making final vector scale invariant.

$$F^1 = \frac{\{d_{ij} \ \forall(i, j)\}}{\text{mean}(\{d_{ij} \ \forall(i, j)\})} \quad (10)$$

This feature captures global shape configuration by capturing the relative position of all shape boundary points.

- **Feature # 2.** Multiscale curvature.

The feature vector is composed of support angles measured at different distances along the curve for each node on the curve. The full feature values vector for a given discrete curve consists of the set of angles

$$F^2 = \{\alpha_{i-j, i, i+j} \ \forall(i, j)\} \quad (11)$$

where angle $\alpha_{i,j,k} = \angle(ijk)$. This feature class can be interpreted as multiscale curvature descriptor.

The technique of constructing features #1 and #2 is illustrated in Figure 4. For feature #1, only linelets d_{ij} lying in the interior of the shape are taken into account.

3.5 Gradient flow computation

In order to use the measure in eq. (7) as a prior in a curve evolution context we must be able to compute the curve flow that minimizes this measure. For simplicity, we

consider a measure defined on a single feature class. Since eq. (7) is additive in the different features classes, minimizing flows for individual feature classes can be added with corresponding weights to obtain the overall minimizing flow.

The derivation of the gradient curve flow for our energy in eq. (7) presents a challenge because the energy depends on the whole curve in non-trivial (non-additive) way. Therefore, the minimizing flow at any location on the curve is also a function of the whole curve and not just of the local curve properties (as it is the case of curve length minimization prior).

We propose 2 methods to derive the flow. Exact analytical derivation based on variational approach and approximate numerical computation.

3.5.1 Analytical derivation of the gradient flow

We first rewrite our generic shape distribution difference prior energy term based on one feature class:

$$E = \int [H(\Gamma, \lambda) - H^*(\lambda)]^2 d\lambda \quad (12)$$

Computation of the gradient curve flow minimizing the energy using variational approach consists of the following steps (see [3] for more details)

1. Finding Gâteaux semi derivative of the energy in eq. 12 with respect to perturbation β . Using the definition of the Gâteaux semi derivative, linearity of the integration and the chain rule we obtain

$$\begin{aligned} \mathcal{G}(E, \beta) &= \int \mathcal{G} [(H(\Gamma, \lambda) - H^*(\lambda))^2] d\lambda = \\ &= 2 \int [H(\Gamma, \lambda) - H^*(\lambda)] \mathcal{G} [H(\Gamma, \lambda), \beta] d\lambda \end{aligned} \quad (13)$$

2. If Gâteaux semi derivative of a linear functional f exists, using to Rietz representation theorem, it can be represented as

$$\mathcal{G}(f, \beta) = \langle \nabla f, \beta \rangle \quad (14)$$

where ∇f is the gradient (flow) minimizing the functional f .

3. If Gâteaux semi derivative $\mathcal{G}([H(\Gamma, \lambda), \beta])$ is found, the flow minimizing the energy in eq. 12 is given by

$$\nabla E = 2 \int [H(\Gamma, \lambda) - H^*(\lambda)] \nabla(H(\Gamma, \lambda)) d\lambda \quad (15)$$

Therefore, the problem of finding the gradient curve flow minimizing our prior energy resides in finding $\mathcal{G}(H(\Gamma, \lambda), \beta)$ and the gradient $\nabla(H(\Gamma, \lambda))$. In appendix A we give the detailed derivation of the curve flows minimizing eq. (12) for feature classes #1 and #2 defined previously. These latest results were not used in experiments reported here. Instead, we use the numerical approximate computation scheme to obtain the minimizing flows (presented below).

3.5.2 Approximate numerical computation of the energy minimizing curve flow

Here we propose a numerical computation scheme for the curve flows minimizing energy in eq. (8). We assume the discretized curve $C[k] = \{x[k], y[k]\}$ and discretized sets of feature values $F_i[n]$ computed from the curve. Correspondingly, we seek a discrete curve flow denoted by $dx[k]$ (displacement of the curve node k in the direction of the normal to the curve at node k). Numerical approach is not necessary for the feature classes used in this work (since analytical expressions are available) but can be useful for the initial testing of new feature classes when analytical flow derivation is deemed difficult or too time demanding.

Our approach consists of 2 steps:

1. Compute the derivative of the cost with respect to individual feature values in the space Ω , thus defining the steepest descent flow on feature values $dF[k]$. The flow on the space of feature values is obtained without considerations of consistency, i.e. the computed change in feature values are not necessarily realizable by any curve deformation.
2. In order to find the curve deformation corresponding to flow computed at step 1, we find a projection of the flow onto the subset spanned by feature value changes corresponding to valid underlying curve deformations. By finding such a projection, we automatically obtain the curve flow that constitutes the gradient direction reducing the cost in eq.(7).

The first step of our approach yields an elegant and efficient solution for the flow $dF[k]$ that can be computed directly on feature values $F[k]$ without explicitly computing feature distribution functions $H(\lambda)$. The complexity of the second step depends on the feature class used. Now we present our 2 step process of computing the gradient curve flow corresponding to eq. (7) in detail.

At the first step, we compute the gradient descent flow $dF[k]$ minimizing eq. (7).

$$dF[p] = -\frac{\partial E(F)}{\partial F[p]} = \left[1 - \frac{1}{P^*} \sum_{i=1}^{P^*} u(F^*[i] - F_C[p]) - \frac{1}{P_C} \sum_{i=1}^{P_C} u(F_C[i] - F_C[p]) \right] G[p] \quad (16)$$

where F^* is the feature values vector corresponding to the prior; F_C is the feature values vector extracted from the curve C ; P^* and P_C are lengths of vectors F^* and F_C respectively; $G[p] = \frac{\partial H(C, \lambda)}{\partial \lambda} \Big|_{\lambda=F[p]}$; and $u(x)$ is a unit step function. given by

$$u(x) = \begin{cases} 1 & | \ x \geq 0 \\ 0 & | \ x < 0 \end{cases} \quad (17)$$

The stationary point of this flow corresponds to the case when $H^*(\lambda) = H(C, \lambda)$, i.e. the distribution function for given curve matches the target distribution function. Our flow in eq. (16) is similar to the histogram equalization flow introduced in [18] without

a variational interpretation. Eq. (16) can also be given in continuous formulation which we omit here for clarity.

In the previous discussion, it is assumed that the elements of the feature space can vary independently (such as image pixel intensities). However, the evolution of the feature values $F[k]$ is only possible on the manifold defined by allowable displacements of the contour points. Thus, one can minimize eq. (7) by finding allowable feature values augmentations $dF_{\perp}[k]$, yet close to target feature flow given by PDE in eq. (16). The process of finding such $dF_{\perp}[k]$ and corresponding curve deformation constitutes the second step of our numerical flow calculation scheme.

In order to maintain uniform contour discretization, we parameterize the deformations of the discrete contour by defining a displacement in the normal direction for node k as $dx[k]$. One needs to find such projection $dF_{\perp}$ (a point on the manifold) that corresponds to allowable shape deformation $dx_{\perp}[k]$, yet is as close to the target dF as possible. This is done by minimizing the following functional (Euclidean distance between target flow and its projection)

$$g(dx_{\perp}) = \sum_{k=1}^{P_C} (dF_{\perp}[k]_{(dx_{\perp})} - dF[k])^2 \quad (18)$$

where P_C is the length of the feature value vector. The optimal curve deformation is found as

$$dx_{\perp}^* = \underset{dx_{\perp}}{\operatorname{argmin}} g(dx_{\perp}) \quad (19)$$

Currently, we have no theoretical proof that such a technique will yield a gradient related direction minimizing eq. (7) for all cases, although experiments show that the flow obtained using this procedure is close to the flow obtained using an analytical solution (appendix A) for the cases where one is available.

For feature class #1, a closed form solution to Eq. (19) is found, while for feature class #2 and other possible feature classes, a numerical solution is proposed. We present here both cases in detail.

- **Closed form solution for feature class #1.**

Let us designate the change of the distance between nodes n and k as dF_{nk} and the angle between the outward normal at node n and the link between nodes k and n as α_{n-kn} . The normal direction at node k is defined as the bisector of the angle between two linelets connecting node k to its neighbors, i.e. k to $k-1$ and k to $k+1$. Recalling that $dx[k]$ is the displacement of the node k in the normal direction to the curve, eq. (19) can be rewritten as

$$dx_{\perp}^* = \underset{dx_{\perp}}{\operatorname{argmin}} g(dx_{\perp}) = \underset{dx_{\perp}}{\operatorname{argmin}} \sum_{n=1}^{P_C} \sum_{k=1}^{P_C} (dx[n] \cos \alpha_{n-kn} + dx[k] \cos \alpha_{k-kn} - dF_{nk})^2 \quad (20)$$

Since eq. (20) is quadratic with respect to $dx[k]$, it possesses a unique minimum. Differentiating with respect to $dx[p]$ yields the following equation

$$\frac{\partial E}{\partial dx[p]} = \sum_{\{J(p)\}} 2 (dF_{pk} - dx[p] \cos \alpha_{p-kp} - dx[k] \cos \alpha_{k-pk}) \cos \alpha_{p-kp} = 0 \quad (21)$$

Resulting system of equations can be rewritten as

$$dx[p] \sum_{\{J(p)\}} \cos^2 \alpha_{p-kp} + \sum_{\{J(p)\}} dx[k] \cos \alpha_{p-kp} \cos \alpha_{k-pk} = \sum_{\{J(p)\}} dF_{pk} \cos \alpha_{p-kp} \quad \forall p \quad (22)$$

System (22) is easily solved by left matrix division, yielding $dx_{\perp}^*$ up to a constant multiplier, stemming from the neglect of the normalization factor in eq. (10).

- **Method II. Numerical gradient descent.**

At each iteration, the gradient $\frac{\partial g(dx_{\perp})}{\partial dx_{\perp}[k]}$ is evaluated. The proposed change of configuration is updated according to

$$dx_{\perp}^+[k] = dx_{\perp}[k] - s \frac{\partial g(dx_{\perp})}{\partial dx_{\perp}[k]} \quad (23)$$

where s is a dynamically updated step size. The change of configuration is accepted only if the cost in eq. (18) improves ($g(dx_{\perp}^+) < g(dx_{\perp})$). This condition, together with dynamic step size management guarantees convergence toward $dx_{\perp}^*$ (local minimum of $g(dx_{\perp})$).

In Figure 5 we illustrate the solution to the eq. (19) by the gradient decent on the manifold. In the shown 3D space, each dimension corresponds to one feature value.

Although this approach can to find the minimizing curve flow for any feature class, this method is computationally demanding and requires finding a trade-off between precision and computational time by carefully choosing the stopping criteria.

4 Experiments

This section presents preliminary results for the following applications:

- Shape morphing
- Mean shape calculation
- Image segmentation

Figure 5: Illustration of the gradient decent solution to eq. (19)

4.1 Shape morphing

In order to test the ability of a constructed shape representation to efficiently capture shapes and the ability of the computed curve flow to drive the curve toward the prior curve we perform curve morphing experiment. In curve morphing setup, the curve is evolved solely under the action of the prior, that is, the energy in eq. (2) does not include any terms other than E_{prior} . Moreover, the prior distributions $H^*(\lambda)$ are computed on the single prior (target) shape. That means that we want to minimize the distance (eq. (8)) between evolving and target shapes, thus matching shape distributions of the evolving curve and the target curve.

Prior shape distribution based on features #1 and #2 (set of all inter-point distances and multiscale curvatures respectively) is constructed from a prior shape. In 3 separate experiments the curve is evolved under the force induced by minimizing eq. (8) constructed solely on feature #1, solely on feature #2 and on both features #1 and #2 combined. The evolution starts from the curve shown by the blue dash-dotted line in Figure 6. The target shape (on which the target distributions are computed) is shown by the green dashed line. For each of the 3 experiments, the resulting curve is shown by the red solid line.

All 3 experiments yield shape very similar to the target shape, but small differences are worth noting. The flow based on feature #1 (panel A in Figure 6) yields the elongated shape which is slightly bent. In fact, boundary curvature is not captured directly by feature #1; therefore, it is expected that differences in global bending deformation are not effectively corrected by the flow based on this feature. However, the relative position of the boundaries captured by feature #1 is well preserved in the resulting shape. The flow based on feature #2 (panel B in Figure 6) yields shape highly similar to the target shape but slightly non-symmetric, cone-shaped

Figure 6: Evolution of the contour under the action of our prior flow: initial (blue, dot-dashed), target (green, dashed), and resulting (red, solid) contours. (A) - prior constructed on the inter-point distances (#1); (B) - prior constructed on multiscale curvatures (#2); (C) - Both feature classes are used.

and of inflated size. This is explained by the fact that feature #2 is designed to capture boundary curvature rather than relative boundary position. In fact, we observe correct curvatures (straight lines and circular regions) in the result for this flow but boundary relative positions do not match those of the prior. Finally, both flows combined (panel C in Figure 6) yield nearly perfect shape. Flows for both features combined work to correct for deficiencies of one another.

The experiment presented in this section illustrates the curve morphing capabilities of our prior. Curve morphing is an interesting application in computer vision and we believe more interesting results can be obtained in this domain using our methods.

4.2 Computing average shape

Our proposed framework provides a natural and interesting approach to the determination of the average shape $\overline{C}$ of a collection of N shapes C_i . Our goal is to find such an average shape that would share important features with prior shapes (be visually similar) and would be equally close to all prior shapes in some sense.

One way to define a mean shape of the collection of shapes C_i is to find a shape C that has minimum sum of squared distances to all shapes C_i . Such shape can be found using the *Karcher* mean formula [12, 3]:

$$\overline{C} = \operatorname{argmin}_C \sum_{j=1}^N d^2(C, C_j) \quad (24)$$

To this end, let us modify the induced distance (eq. (8)) between curves as follows:

$$d(C, C_j) = \sqrt{\sum_{i=1}^M w_i \int_0^{\lambda_{max}} [H^i(C, \lambda) - H^i(C_j, \lambda)]^2 d\lambda} \quad (25)$$

By substituting eq. (25) into eq. (24) it can be shown that the resulting mean curve $\bar{C}$ is the curve which minimizes the distance to the average of feature distributions corresponding to prior curves C_i . Therefore, one effectively needs to minimize eq. (7) using average distributions as the target $H^*(\lambda)$; our framework can be used to find the solution of eq. (24) using curve evolution.

Although eq. (24) yields a mean shape for any chosen curve distance measure, using traditional curve distance measures often yields unsatisfactory mean shapes. In order to illustrate this, we present an example problem. On figure 7, we show two shape instances (blue solid lines), whose mean shape we would like to find. From the visual similarity point of view, the mean of these two triangles should be a triangle with the right corner located somewhere between the corresponding corners of the two prior shapes.

An example of a generic curve distance measure is the Chamfer distance [24] that can be defined as

$$d(C_1, C_2) = \int_{C_1} \min ||x - C_2|| ds \quad (26)$$

where the integration is carried out along C_1 accumulating the Euclidean distance between current point on C_1 and curve C_2 . In figure 7 (A) we show the resulting mean shape corresponding to the two given shapes. Clearly, such a mean shape is flawed from visual similarity point of view.

Another often used shape difference measure based on the total area between shapes is the *Hamming* distance [22]

$$d(C_1, C_2) = \int_{A: \text{sign}(DT(C_1)) \neq \text{sign}(DT(C_2))} dS \quad (27)$$

where $DT(C_1)$ and $DT(C_2)$ are signed distance transforms for shapes C_1 and C_2 respectively. When used in eq. (24), this shape difference measure yields an infinite number of solutions for the mean shape. These solutions are located in areas shaded in red in figure 7 (B). None of those shapes is a triangle with the right proportions (a perceptual mean shape). A similar result is obtained using Hausdorff distance [3]. For this measure, the solution to eq. (24) is not unique, although the set of solutions includes a perceptually correct mean shape.

Finally, we use our measure of shape difference in eq. (25) based on the difference between shape feature distributions. We construct the difference measure using 2 feature classes: set of all inter-point distances and multiscale curvatures. In figure 7 (C) we show the result of applying our method to find the mean shape. The contour produced by our iterative process is shown by the solid red line. The size of this shape is smaller than that of the original shapes due to initialization which is consistent with the scale, translational and rotational invariance of our measure. We manually scale and shift the resulting contour to match the position of original shapes for visualization purposes. One can see that the scaled result produces the expected “mean shape” from visual similarity point of view.

Figure 7: Average shape calculation using three shape difference measures. Blue solid contours correspond to prior shapes; red dashed line represent the mean shape; red filled areas correspond to the family of solutions. (A) - asymmetric distance based measure; (B) - area based measure. One of the possible solution is shown by green dashed line; (C) - our distribution difference measure (solid red line - evolution result; dashed red line - scaled result).

It is important to note that using shape models (measures) that explicitly capture correspondence of shape features, such as point distribution model (PDM) in [4], would automatically find an average shape from visual similarity point of view. Our achievement is to obtain a similar good result *without* correspondence but only using the aggregate shape dissimilarity measure based on uniformly sampled information about prior shapes. No explicit topology information is assumed in our framework.

4.3 Application to image segmentation

4.3.1 Formulation

We pose the segmentation problem as a contour-based energy minimization problem using the framework of [10]. More detailed description of this framework can be found in [10] and references therein. The curve C segmenting the object from the background is represented as the zero level set of the signed distance function U and is found by minimizing the functional:

$$C^* = \underset{U(C)}{\operatorname{argmin}} [E_d + \alpha E_C + \beta E_U] \quad (28)$$

where E_d is a data fidelity term, E_U approximately enforces $|\nabla U| = 1$, so that U remains a distance function as the evolution progresses, and E_C is a prior term on the curve boundary. Using otherwise similar algorithm and regularization parameters we compare the results for different forms of E_C , namely, generic curve length prior $E_C = \int_C ds$, Leventon's prior in [10], and our prior in eq. (7). The evolution of the curve is carried out using the level set technique.

We use a simplified version of the data fidelity term E_d matched to the synthetic bimodal images used to test our algorithm. Given the average image intensities u

Figure 8: Segmentation results. A: Our method; B: Method in [10]; C: Curve length penalty prior; White - final result; Black - true shape boundary; Dashed line - initial curve. Symmetric area distance (in pixels) between true boundary and final result is shown on the top of each panel.

and v , inside and outside the shape boundary respectively, in case of Gaussian noise, the data fidelity term E_d can be formulated as minus log likelihood of the image and is given by

$$E_d = \int_{R^u} (I - u)^2 dA + \int_{R^v} (I - v)^2 dA \quad (29)$$

where integration is carried out over inside and outside areas R^u and R^v respectively. The curve flow corresponding to the gradient descent with respect to eq. (29) is given by

$$\frac{d\vec{C}}{dt} = -\nabla E_d = \frac{(u - v)}{2} \left(I + \frac{u + v}{2} \right) \vec{N} \quad (30)$$

4.3.2 Experiment

In figure 8 we compare the segmentation results given by our prior, the prior proposed in [10], and the standard curve length penalty prior. Independent Gaussian noise (SNR=-17.5 db) was added to a bimodal image of the triangle to create the data image. The boundary of the triangle (ground truth contour) is shown by black solid line. We show the segmentation obtained using our proposed prior model in frame (A); the result using the curvature density prior of [10] in frame (B); the result using the standard curve length penalty in frame (C). The regularization parameters were chosen to obtain the best result for each prior.

Our prior is constructed using 2 features classes: set of all inter-point distances and multiscale curvatures. The target distributions for each feature class are computed as averages of distributions of the 4 prior triangular shapes.

While the difference between later two methods (B, C) is small (and can be due to the regularization parameter choice), our method (A) gives a significantly different

result, which is, most importantly, visually similar to the ground truth shape. On the top of each panel, we show a measure of segmentation error, calculated symmetric distance (in pixels) between true boundary and final result. It is important that while giving some improvement of this error measure, our result is visually significantly superior. The resulting error (symmetric area distance in this case) for our prior is mostly attributed to the bias in location and angular position of the resulting shape, while the error for other two methods is due to shape "distortion".

5 Conclusions

In this work, we introduce the novel method of constructing and using prior shape. Our method relies on modeling distributions of certain significant features of shapes and creating a measure of similarity between these distributions. A key to the useful implementation of obtained measure lies in our ability to incorporate it into curve evolution framework as a prior energy term. We create a framework allowing such incorporation and present initial results for 3 different applications.

To our best knowledge, shape prior based on shape distributions is for the first time formulated and applied to practical problems. Our shape morphing experiment shows the ability of the prior curve flow to transform one curve into another curve resembling the prior example. Our experiments illustrate the dependence of captured properties of shape on chosen feature class. In another experiment we show the ability of our prior to find visual mean of given shapes, unattainable by equivalent existing approaches. Preliminary segmentation experiments show the ability of our prior to effectively drive the result to the shape similar to the training data shapes. Comparison of our result with existing regularization approaches shows that while achieving similar segmentation error based on symmetric distance between the result and the true shape, our method yields the shape visually significantly better.

In order to illustrate the generalizing abilities of our prior, we plan to design experiments comparing our model and other techniques such as PCA based methods [28]. Better understanding of the ability of a particular feature class to capture the shape knowledge is desirable to facilitate the feature choice. The issue of feature choice for more complex shapes will be addressed in future research.

A Appendix. Analytical computation of the minimizing flows for distribution difference priors

Definitions:

$\{s, s_1, s_2, p\} \in [0, 1]$ - curve arc length parameterization variables

λ - variable spanning the range of the feature values for a given feature class

Γ - curve

$\vec{\Gamma}(s) = \{x(s), y(s)\}$ - curve coordinates

$\vec{\Gamma}(s_1, s_2) = \vec{\Gamma}(s_1) - \vec{\Gamma}(s_2)$ - point-to-point vector with coordinates $\{x(s_1) - x(s_2), y(s_1) -$

$y(s_2)\}$

$\beta(s) \in R^1$ - continuous differentiable function used to perturb the curve. Perturbations are assumed to be orthogonal to local tangent vectors

$\vec{n}(s)$ - normal vector as point s

$\vec{\beta}(s) = \beta(s)\vec{n}(s)$ - deformation vector at point s

$H(\Gamma, \lambda)$ - cumulative distribution function defined on curve Γ

$H^*(\lambda)$ - prior cumulative distribution function

$h(x) = h(x > 0)$ - indicator function

$\mathcal{G}(f)$ - Gâteaux derivative

The following derivations are similar to those made in [3]. We refer the reader to [3] and references therein for more detailed arguments and functional analysis background. We first rewrite our generic shape distribution difference prior term:

$$E = \int [H(\Gamma, \lambda) - H^*(\lambda)]^2 d\lambda \quad (31)$$

The following steps are needed to compute the curve flow minimizing the energy:

1. Finding Gâteaux semi derivative of the energy in eq. 31 with respect to perturbation β . Using the definition of the Gâteaux semi derivative, linearity of the integration and the chain rule we obtain

$$\begin{aligned} \mathcal{G}(E, \beta) &= \int \mathcal{G} [(H(\Gamma, \lambda) - H^*(\lambda))^2] d\lambda = \\ &= 2 \int [H(\Gamma, \lambda) - H^*(\lambda)] \mathcal{G} [H(\Gamma, \lambda), \beta] d\lambda \end{aligned} \quad (32)$$

2. If Gâteaux semi derivative of a linear functional f exists, using to Rietz representation theorem, it can be represented as

$$\mathcal{G}(f, \beta) = \langle \nabla f, \beta \rangle \quad (33)$$

where ∇f is the gradient (flow) minimizing the functional f .

3. If Gâteaux semi derivative $\mathcal{G}([H(\Gamma, \lambda), \beta])$ is found, the flow minimizing the energy in eq. 31 is given by

$$\nabla E = 2 \int [H(\Gamma, \lambda) - H^*(\lambda)] \nabla(H(\Gamma, \lambda)) d\lambda \quad (34)$$

We concentrate on step 2, that is finding Gâteaux derivative $\mathcal{G} [H(\Gamma, \lambda), \beta]$ and corresponding representation yielding the flow $\nabla(H(\Gamma, \lambda))$

A.1 Feature class #1. Collection of inter-point distances

More definitions:

$d(\vec{\Gamma}(s_1), \vec{\Gamma}(s_2))$ - Euclidean distance between points s_1 and s_2 on the curve

For this feature class, the cumulative distribution function for a curve Γ is defined as:

$$H(\Gamma, \lambda) = \int_0^1 \int_0^1 h(d(\vec{\Gamma}(s_1), \vec{\Gamma}(s_2)) > \lambda) ds_1 ds_2 \quad (35)$$

Now we apply the perturbation $\epsilon\beta(s)$ to the curve Γ , yielding deformed curve

$$\vec{\Gamma}'(s) = \vec{\Gamma}(s) + \epsilon\beta(s)\vec{n}(s) \quad (36)$$

It is easy to see that for small ϵ the distance between 2 points on the perturbed curve is

$$d(\vec{\Gamma}'(s_1), \vec{\Gamma}'(s_2)) = d(\vec{\Gamma}(s_1), \vec{\Gamma}(s_2)) + \epsilon [\beta(s_1)\vec{n}(s_1) - \beta(s_2)\vec{n}(s_2)] \cdot \frac{\vec{\Gamma}(s_1, s_2)}{|\vec{\Gamma}(s_1, s_2)|} \quad (37)$$

It means that distance between 2 points on the curve is augmented by a projection of difference between deformation vectors on the vector $\vec{\Gamma}(s_1, s_2)$.

We can now write the Gâteaux derivative $\mathcal{G}[H(\Gamma, \lambda), \beta]$ according to its definition

$$\begin{aligned} \mathcal{G}[H(\Gamma, \lambda), \beta] &= \lim_{\epsilon \rightarrow 0} \frac{H(\Gamma', \lambda) - H(\Gamma, \lambda)}{\epsilon} = \\ &= \lim_{\epsilon \rightarrow 0} \frac{\int_0^1 \int_0^1 [h(d(\vec{\Gamma}'(s_1), \vec{\Gamma}'(s_2)) > \lambda) - h(d(\vec{\Gamma}(s_1), \vec{\Gamma}(s_2)) > \lambda)] ds_1 ds_2}{\epsilon} = \\ &= \lim_{\epsilon \rightarrow 0} \frac{\int_0^1 \int_0^1 \left[h\left(d(\vec{\Gamma}(s_1), \vec{\Gamma}(s_2)) + \epsilon [\beta(s_1)\vec{n}(s_1) - \beta(s_2)\vec{n}(s_2)] \cdot \frac{\vec{\Gamma}(s_1, s_2)}{|\vec{\Gamma}(s_1, s_2)|} > \lambda\right) - \right. \\ &\quad \left. - h(d(\vec{\Gamma}(s_1), \vec{\Gamma}(s_2)) > \lambda) \right] ds_1 ds_2}{1} \end{aligned} \quad (38)$$

In the following we introduce simplified notation:

$$d = d(\vec{\Gamma}(s_1), \vec{\Gamma}(s_2))$$

$$d\beta = [\beta(s_1)\vec{n}(s_1) - \beta(s_2)\vec{n}(s_2)]$$

$$d\Gamma = \frac{\vec{\Gamma}(s_1, s_2)}{|\vec{\Gamma}(s_1, s_2)|}$$

$$\{n_x(s), n_y(s)\} = \vec{n}(s) \text{ - components of the normal vector at } s$$

Now we can rewrite

$$\mathcal{G}[H(\Gamma, \lambda), \beta] = \lim_{\epsilon \rightarrow 0} \frac{\int_0^1 \int_0^1 [h(d - \lambda + \epsilon d\beta \cdot d\Gamma) - h(d - \lambda)] ds_1 ds_2}{\epsilon} \quad (39)$$

In order to find the limit, we use smooth approximation of the indicator function:

$$h(x) \longrightarrow \phi^\alpha(x) = \frac{\operatorname{atan}\left(\frac{x}{\alpha}\right) + 1}{2} \quad (40)$$

where the parameter α defines the degree of approximation. In the limiting case of $\alpha = 0$ the approximation becomes an equality. We perform the derivation using the approximation and at the last step consider the limiting case of $\alpha = 0$.

Gâteaux derivative becomes:

$$\mathcal{G}[H(\Gamma, \lambda), \beta] = \lim_{\epsilon \rightarrow 0} \frac{\frac{1}{2} \int_0^1 \int_0^1 \operatorname{atan}\left(\frac{\lambda-d}{\alpha}\right) + \operatorname{atan}\left(\frac{\epsilon d\beta \cdot d\Gamma + d - \lambda}{\alpha}\right)}{\epsilon} \quad (41)$$

For small ϵ , using the Taylor expansion we obtain

$$\operatorname{atan}\left(\frac{\epsilon d\beta \cdot d\Gamma + d - \lambda}{\alpha}\right) = \operatorname{atan}\left(\frac{d - \lambda}{\alpha}\right) + \operatorname{atan}'\left(\frac{d - \lambda}{\alpha}\right) \left(\frac{\epsilon d\beta \cdot d\Gamma}{\alpha}\right) + O(\epsilon^2) \quad (42)$$

Now we can find the Gâteaux derivative as follows

$$\begin{aligned} \mathcal{G}[H(\Gamma, \lambda), \beta] &= \lim_{\epsilon \rightarrow 0} \frac{\frac{1}{2} \int_0^1 \int_0^1 \operatorname{atan}'\left(\frac{d-\lambda}{\alpha}\right) \left(\frac{\epsilon d\beta \cdot d\Gamma}{\alpha}\right)}{\epsilon} = \\ &= \frac{1}{2\alpha} \int_0^1 \int_0^1 \operatorname{atan}'\left(\frac{d-\lambda}{\alpha}\right) (d\beta \cdot d\Gamma) ds_1 ds_2 = \frac{1}{2\alpha} \int_0^1 \int_0^1 \frac{ds_1 ds_2}{1 + \left(\frac{d-\lambda}{\alpha}\right)^2} (d\beta \cdot d\Gamma) = \\ &= \frac{1}{2\alpha} \int_0^1 \int_0^1 \frac{ds_1 ds_2}{1 + \left(\frac{d-\lambda}{\alpha}\right)^2} [(n_x(s_1)\beta(s_1) - n_x(s_2)\beta(s_2))(x(s_1) - x(s_2)) + \\ &\quad + (n_y(s_1)\beta(s_1) - n_y(s_2)\beta(s_2))(y(s_1) - y(s_2))] = \\ &= \frac{1}{2\alpha} \int_0^1 \left[\int_0^1 \frac{n_x(s_1)(x(s_1) - x(s_2)) + n_y(s_1)(y(s_1) - y(s_2))}{1 + \left(\frac{d-\lambda}{\alpha}\right)^2} ds_2 \right] \beta(s_1) ds_1 - \\ &= \frac{1}{2\alpha} \int_0^1 \left[\int_0^1 \frac{n_x(s_2)(x(s_1) - x(s_2)) + n_y(s_2)(y(s_1) - y(s_2))}{1 + \left(\frac{d-\lambda}{\alpha}\right)^2} ds_1 \right] \beta(s_2) ds_2 \quad (43) \end{aligned}$$

Changing the order of integration in the second integral we obtain

$$\begin{aligned} \mathcal{G}[H(\Gamma, \lambda), \beta] &= \\ &= \frac{1}{\alpha} \int_0^1 \left[\int_0^1 \frac{n_x(s_1)(x(s_1) - x(s_2)) + n_y(s_1)(y(s_1) - y(s_2))}{1 + \left(\frac{d-\lambda}{\alpha}\right)^2} ds_2 \right] \beta(s_1) ds_1 \quad (44) \end{aligned}$$

According to eq. (33), the expression in square brackets is the gradient

$$\begin{aligned} \nabla H(\Gamma)(s) = \frac{1}{\alpha} \int_0^1 \frac{n_x(s)(x(s) - x(t)) + n_y(s)(y(s) - y(t))}{1 + \left(\frac{d-\lambda}{\alpha}\right)^2} dt = \\ \frac{1}{\alpha} \int_0^1 \frac{n(s) \cdot d\Gamma(s, t)}{1 + \left(\frac{d-\lambda}{\alpha}\right)^2} dt \end{aligned} \quad (45)$$

Gradient flow minimizing energy in eq. (31) is therefore

$$\begin{aligned} \nabla E(\Gamma)(s) = 2 \int_{\lambda} d\lambda [H(\Gamma, \lambda) - H^*(\lambda)] \left[\frac{1}{\alpha} \int_0^1 \frac{n(s) \cdot d\Gamma(s, t)}{1 + \left(\frac{d-\lambda}{\alpha}\right)^2} dt \right] = \\ 2 \int_{\lambda} d\lambda [H(\Gamma, \lambda) - H^*(\lambda)] \left[\int_0^1 \frac{\alpha n(s) \cdot d\Gamma(s, t)}{\alpha^2 + (d - \lambda)^2} dt \right] \end{aligned} \quad (46)$$

For $\alpha \approx 0$, the expression inside the integral is only non zero when $d = \lambda$. Changing the order of integration we obtain:

$$\begin{aligned} \nabla E(\Gamma)(s) = 2 \int_0^1 \vec{n}(s) \cdot \frac{\vec{\Gamma}(s, t)}{|\vec{\Gamma}(s, t)|} dt \int_{\lambda} [H(\Gamma, \lambda) - H^*(\lambda)] \frac{\alpha}{\alpha^2 + (d - \lambda)^2} d\lambda = \\ 2 \int_0^1 \vec{n}(s) \cdot \frac{\vec{\Gamma}(s, t)}{|\vec{\Gamma}(s, t)|} [H(\Gamma, |\vec{\Gamma}(s, t)|) - H^*(|\vec{\Gamma}(s, t)|)] dt \end{aligned} \quad (47)$$

The obtained expression is simple and has an intuitive interpretation. The flow at each point s on the curve is computed as the integral along the curve. For each point p on the curve the expression under the integral is the projection of the normal vector at s onto the vector pointing from s to p , modulated by the difference between current and model distributions evaluated at the distance between s and p . This projection is intuitively the projection of the “force” acting on the feature (link between s and p). The computational complexity of the flow computation is $O(N^2)$ were N is the number of nodes of the discretized curve.

A.2 Feature class #2. Multiscale “curvatures”

A.2.1 Computation of feature values.

We recall that the feature values are the “support” angles α , see figure 9. The cumulative distribution is computed on the set of angles measured for all possible combinations of “base” point s_1 and symmetrical “side” points $s_1 - s_2$ and $s_1 + s_2$ as

$$H(\Gamma, \lambda) = \int_0^1 \int_0^1 h(\alpha(s_1 - s_2, s_1, s_1 + s_2) > \lambda) ds_1 ds_2 \quad (48)$$

We choose to define the “inner” angle α as shown on figure 9 as the angle between vectors $d\vec{\Gamma}(s_1, s_1 - s_2)$ and $d\vec{\Gamma}(s_1, s_1 + s_2)$ measured always in the same half-space, separated by the difference vector between ray directions $d\vec{\Gamma}(s_1 - s_2, s_1 + s_2) = \frac{\vec{\Gamma}(s_1, s_1 - s_2)}{|\vec{\Gamma}(s_1, s_1 - s_2)|} - \frac{\vec{\Gamma}(s_1, s_1 + s_2)}{|\vec{\Gamma}(s_1, s_1 + s_2)|}$. This half-space is fixed to be the inside of the curve for $s_2 = 0$. By other words, the angle α must be a continuous function of s_2 , and $\alpha_{(s_2=0)} = \pi$.

Figure 9: Illustration of feature value computation for feature class #2

For each “base” point s_1 the angle α is computed sequentially for s_2 increasing from 0 to 1. This process is illustrated in Figure 10. For each base point s_1 , we start

Figure 10: Sequential computation of the angles for a particular “base” point s_1 , starting from $r = 0$ (assuming inside of the curve is up-wards).

from $s_2 = 0$; $r(s_1, s_2) = \frac{\text{sign}(\kappa(s_1)) + 1}{2}$; $\alpha(s_1, s_2) = \pi$. Flag function $r(s_1, s_2)$ for $s_2 > 0$ is defined as

$$r(s_1, s_2) = \begin{cases} 0 & \text{if } \alpha(s_1, s_2) \leq \pi \\ 1 & \text{otherwise} \end{cases} \quad (49)$$

We define a mean direction vector as

$$\vec{\Gamma}_m(s_1 - s_2, s_1 + s_2) = \frac{\vec{\Gamma}(s_1, s_1 - s_2)}{|\vec{\Gamma}(s_1, s_1 - s_2)|} + \frac{\vec{\Gamma}(s_1, s_1 + s_2)}{|\vec{\Gamma}(s_1, s_1 + s_2)|} \quad (50)$$

In the process of computing $\alpha(s_1, s_2)$, as s_2 increases by ds , we capture the change of orientation of the mean direction vector by looking at the scalar product

$$C = \vec{\Gamma}_m(s_1 - s_2, s_1 + s_2) \cdot \vec{\Gamma}_m(s_1 - s_2 - ds, s_1 + s_2 + ds) \quad (51)$$

If C becomes less than zero for some $s_2 = l$ we change the sign of $r(s_1, l)$ for consecutive values of $s_2 > l$. Angle between rays $\angle(\vec{\Gamma}(s_1, s_1 - s_2), \vec{\Gamma}(s_1, s_1 + s_2)) \in [0, \pi]$ is measured as inverse cosine. After sequential computation is performed for all s_1 we end up with a 2D functions $\angle(s_1, s_2)$ and $r(s_1, s_2)$. The latest is used to correct the values of $\angle(\vec{\Gamma}(s_1, s_1 - s_2), \vec{\Gamma}(s_1, s_1 + s_2))$ that must be greater than π . After all, angles $\alpha(s_1, s_2)$ are given by

$$\alpha(s_1, s_2) = (2\pi)r(s_1, s_2) + (-1)^{r(s_1, s_2)} \arccos \left(\frac{d\vec{\Gamma}(s_1, s_1 - s_2)}{|d\vec{\Gamma}(s_1, s_1 - s_2)|} \cdot \frac{d\vec{\Gamma}(s_1, s_1 + s_2)}{|d\vec{\Gamma}(s_1, s_1 + s_2)|} \right) \quad (52)$$

A.2.2 Flow computation

In order to find the energy (eq. (31)) minimizing flow we perform essentially the same procedure as for feature class #1. We refer the reader to the previous section for omitted details.

First, we perturb the curve in the normal direction by $\epsilon\beta(s)$. It is important that all 3 points (“base” point and 2 “side” points) on the curve change their positions. Let α' be the angle after the perturbation. Using the continuous approximation of the indicator function, we obtain

$$\begin{aligned} \mathcal{G}(H(\lambda), \beta) &= \lim_{\epsilon \rightarrow 0} \frac{\int \int ds_1 ds_2 [h((\alpha') > \lambda) - h((\alpha) > \lambda)]}{\epsilon} = \\ &= \lim_{\epsilon \rightarrow 0} \frac{\int \int ds_1 ds_2 [\phi^\gamma(\alpha' - \lambda) - \phi^\gamma(\alpha - \lambda)]}{\epsilon} = \\ &= \lim_{\epsilon \rightarrow 0} \frac{\frac{1}{2} \int \int ds_1 ds_2 \left[\text{atan}\left(\frac{\alpha' - \lambda}{\gamma}\right) - \text{atan}\left(\frac{\alpha - \lambda}{\gamma}\right) \right]}{\epsilon} = \\ &= \lim_{\epsilon \rightarrow 0} \frac{\frac{1}{2} \int \int ds_1 ds_2 \text{atan}'\left(\frac{\alpha - \lambda}{\gamma}\right)(\alpha' - \alpha)}{\epsilon} = \lim_{\epsilon \rightarrow 0} \frac{\frac{1}{2} \int \int ds_1 ds_2 \frac{\gamma}{\gamma^2 + (\alpha - \lambda)^2}(\alpha' - \alpha)}{\epsilon} \end{aligned} \quad (53)$$

We must therefore compute the angle increment $\alpha' - \alpha$ resulting from the perturbation. This increment consists of 3 terms (additive due to the assumption of the small perturbation):

$$\alpha' - \alpha = d\alpha^{(1)} + d\alpha^{(1)} + d\alpha^{(1)} \quad (54)$$

where the first 2 terms result from the displacement of “side” points and the third term results from the displacement of the “base” point $\Gamma(s_1)$.

Figure 11: Local perturbation of the curve at point $\vec{\Gamma}(s_1+s_2)$. Perturbation $\epsilon\beta(s_1+s_2)$ is infinitely small comparing to $|\vec{\Gamma}(s_1, s_1+s_2)|$

We first consider $d\alpha^{(1)}$ ($d\alpha^{(2)}$ is determined similarly). The local geometry of the curve perturbation at point s_1+s_2 is shown in detail in figure 11. It is easy to see that the increment $d\alpha^{(1)}$ can be found as follows

$$\begin{aligned} \frac{p}{\epsilon\beta(s)} &= \sin(\Theta) \\ p &= \epsilon\beta(s)\sqrt{1 - \cos^2 \Theta} \\ d\alpha^{(1)} &= \frac{p}{|d\Gamma|} = \frac{\epsilon\beta(s)\sqrt{1 - \left(\vec{n}(s) \cdot \frac{d\Gamma}{|d\Gamma|}\right)^2}}{|d\Gamma|} \end{aligned} \quad (55)$$

Figure 12: Illustration of 2 cases when the sign of the angle increment $d\alpha^{(1)}$ is different for the same curve perturbation $\epsilon\beta(s)$.

It is important to recognize that the sign of the above angle increment depends on the relative direction of the normal $\vec{n}(s_1+s_2)$ and $\vec{\Gamma}_m(s_1, s_1+s_2)$. Two possible cases are illustrated in figure 12. In case 1, for positive $\beta(s_1+s_2)$, the increment is positive and for case 2, it is negative. We define a flag function $f(s_1+s_2)$ as follows

$$f(s_1+s_2) = \begin{cases} -1 & \text{if points L and P are on the same side of } \Gamma(s_1, s_1+s_2) \\ 1 & \text{otherwise} \end{cases} \quad (56)$$

Under this definition,

$$d\alpha^{(1)} = \frac{\epsilon\beta(s_1 + s_2)\sqrt{1 - \left(\vec{n}(s_1) \cdot \frac{\vec{\Gamma}(s_1+s_2)}{|\vec{\Gamma}(s_1+s_2)|}\right)^2}}{|\vec{\Gamma}(s_1 + s_2)|} f(s_1 + s_2) \quad (57)$$

$$d\alpha^{(2)} = \frac{\epsilon\beta(s_1 - s_2)\sqrt{1 - \left(\vec{n}(s_1) \cdot \frac{\vec{\Gamma}(s_1-s_2)}{|\vec{\Gamma}(s_1-s_2)|}\right)^2}}{|\vec{\Gamma}(s_1 - s_2)|} f(s_1 - s_2) \quad (58)$$

We now proceed to calculate $d\alpha^{(3)}$. Let us assume that angles α' and α are computed as inverse cosines. In such case, for $\alpha > \pi$, the sign of the angle increment must be changed as follows

$$d\alpha^{(3)} = (\alpha' - \alpha)(-1)^{r(s_1, s_2)} \quad (59)$$

where α' is the angle resulting after displacing point s_1 . Using cosine theorem and abbreviations from figure 9, we can write the angle increment due to the displacement of point s_1 as

$$\begin{aligned} \alpha' - \alpha &= \arccos\left(\frac{a^2 - b'^2 - c'^2}{-2b'c'}\right) - \arccos\left(\frac{a^2 - b^2 - c^2}{-2bc}\right) = \\ &\arccos\left(\frac{a^2 - \left(b - n(s_1) \cdot \frac{\Gamma_-}{|\Gamma_-|}\beta(s_1)\epsilon\right)^2 - \left(c - n(s_1) \cdot \frac{\Gamma_+}{|\Gamma_+|}\beta(s_1)\epsilon\right)^2}{-2\left(b - n(s_1) \cdot \frac{\Gamma_-}{|\Gamma_-|}\beta(s_1)\epsilon\right)\left(c - n(s_1) \cdot \frac{\Gamma_+}{|\Gamma_+|}\beta(s_1)\epsilon\right)}\right) - \\ &\quad - \arccos\left(\frac{a^2 - b^2 - c^2}{-2bc}\right) = \\ &\arccos\left(\frac{a^2 - b^2 - c^2 + 2\beta(s_1)\epsilon\left(n(s_1) \cdot \frac{\Gamma_-}{|\Gamma_-|}b + n(s_1) \cdot \frac{\Gamma_+}{|\Gamma_+|}c\right)}{-2bc + 2\beta(s_1)\epsilon\left(n(s_1) \cdot \frac{\Gamma_-}{|\Gamma_-|}c + n(s_1) \cdot \frac{\Gamma_+}{|\Gamma_+|}b\right)}\right) - \\ &\quad - \arccos\left(\frac{a^2 - b^2 - c^2}{-2bc}\right) \end{aligned} \quad (60)$$

where $()'$ indicates the quantity after point s_1 displacement; $\Gamma_+ = \vec{\Gamma}(s_1, s_1 + s_2)$ and $\Gamma_- = \vec{\Gamma}(s_1, s_1 - s_2)$. Using Taylor expansion

$$\frac{m + \epsilon_1}{n + \epsilon_2} = \frac{m}{n} + \frac{\epsilon_1}{n} + \frac{\epsilon_2 m}{n^2} \quad (61)$$

we obtain

$$\begin{aligned} \alpha' - \alpha &= \arccos\left(\frac{a^2 - b^2 - c^2}{-2bc} + \frac{\beta(s_1)\epsilon\vec{n}(s_1) \cdot \left(\frac{\Gamma_-}{|\Gamma_-|}b + \frac{\Gamma_+}{|\Gamma_+|}c\right)}{-bc} - \right. \\ &\quad \left. - \frac{\beta(s_1)\epsilon\vec{n}(s_1) \cdot \left(\frac{\Gamma_-}{|\Gamma_-|}c + \frac{\Gamma_+}{|\Gamma_+|}b\right)(a^2 - b^2 - c^2)}{2b^2c^2}\right) - \arccos\left(\frac{a^2 - b^2 - c^2}{-2bc}\right) \end{aligned} \quad (62)$$

Again using Tailor expansion of the cosine function we obtain

$$\begin{aligned}
 & \alpha' - \alpha = \arccos' \left(\frac{a^2 - b^2 - c^2}{-2bc} \right) \epsilon \beta(s_1) \times \\
 & \vec{n}(s_1) \cdot \left(-\frac{\Gamma-}{c|\Gamma-|} - \frac{\Gamma+}{b|\Gamma+|} - \frac{a^2 - b^2 - c^2}{2b^2c^2} \left(\frac{\Gamma-}{|\Gamma-|}c + \frac{\Gamma+}{|\Gamma+|}b \right) \right) = \\
 & -\frac{1}{\sqrt{\cos(\alpha)}} \frac{\epsilon \beta(s_1) \vec{n}(s_1)}{2bc} \cdot \left(\frac{\Gamma+}{|\Gamma+|} \frac{a^2 + c^2 - b^2}{c} + \frac{\Gamma-}{|\Gamma-|} \frac{a^2 + b^2 - c^2}{b} \right) = \\
 & -\frac{1}{\sin(\alpha)} \frac{\epsilon \beta(s_1) a}{bc} \underbrace{[\cos(\beta) \cos(n(s_1), \Gamma+) + \cos(\gamma) \cos(n(s_1), \Gamma-)]}_K \quad (63)
 \end{aligned}$$

We finally obtained the expression for $d\alpha^{(3)}$. We only need to resolve the uncertainty arising when $\alpha \approx \pi$. In such case, we assume $\alpha = \pi - \delta$, $\beta = O(\delta)$, $\gamma = O(\delta)$. Using Tailor expansion we can write the square bracket as

$$\begin{aligned}
 K &= \frac{\left(1 - \frac{1}{2}\beta^2\right) \cos(\alpha - \angle(n(s_1), \Gamma-)) + \left(1 - \frac{1}{2}\gamma^2\right) \cos(\angle(n(s_1), \Gamma-))}{\delta} = \\
 & \frac{-\cos(\delta + \angle(n(s_1), \Gamma-)) + \cos(\angle(n(s_1), \Gamma-))}{\delta} = \\
 & -\frac{\cos(\angle(n(s_1), \Gamma-)) - \sin(\angle(n(s_1), \Gamma-))\delta - \cos(\angle(n(s_1), \Gamma-))}{\delta} = \\
 & \sin(\vec{n}(s_1), \vec{\Gamma}(s_1, s_1 - s_2)) \quad (64)
 \end{aligned}$$

Finally,

$$d\alpha^{(3)} = -\frac{\epsilon \beta(s_1) a}{bc} \left\{ \begin{array}{ll} \frac{\cos(\beta) \cos(n(s_1), \Gamma+) + \cos(\gamma) \cos(n(s_1), \Gamma-)}{\sin \alpha} & \text{when } \alpha \neq \pi \\ \sin(\vec{n}(s_1), \vec{\Gamma}(s_1, s_1 - s_2)) & \text{otherwise} \end{array} \right\} (-1)^{r(s_1, s_2)} \quad (65)$$

Combining eq. (53, 54), the Gâteaux derivative can be written as

$$\begin{aligned}
 & \mathcal{G}(H(\lambda), \beta) = \\
 & \int_0^1 \int_0^1 ds_1 ds_2 \frac{\gamma}{\gamma^2 + (\alpha - \lambda)^2} \left[\frac{\beta(s_1 + s_2) \sqrt{1 - \left(\vec{n}(s_1) \cdot \frac{\vec{\Gamma}(s_1 + s_2)}{|\vec{\Gamma}(s_1 + s_2)|} \right)^2}}{|\vec{\Gamma}(s_1 + s_2)|} f(s_1 + s_2) + \right. \\
 & \left. \frac{\beta(s_1 - s_2) \sqrt{1 - \left(\vec{n}(s_1) \cdot \frac{\vec{\Gamma}(s_1 - s_2)}{|\vec{\Gamma}(s_1 - s_2)|} \right)^2}}{|\vec{\Gamma}(s_1 - s_2)|} f(s_1 - s_2) - \right. \\
 & \left. -\frac{\beta(s_1) a}{bc} \left\{ \begin{array}{ll} \frac{\cos(\beta) \cos(n(s_1), \Gamma+) + \cos(\gamma) \cos(n(s_1), \Gamma-)}{\sin \alpha} & \text{when } \alpha \neq \pi \\ \sin(\vec{n}(s_1), \vec{\Gamma}(s_1, s_1 - s_2)) & \text{otherwise} \end{array} \right\} (-1)^{r(s_1, s_2)} \right] \quad (66)
 \end{aligned}$$

Using Ritz representation theorem and changing variables of integration we obtain the gradient flow minimizing $H(\Gamma, \lambda)$

$$\begin{aligned} \nabla H(\Gamma, \lambda)(s) = & \int_0^1 \frac{-\gamma dt}{\gamma^2 + (\alpha(s) - \lambda)^2} \frac{a}{bc} \times \\ & \left\{ \frac{\cos(\beta) \cos(n(s_1), \Gamma+) + \cos(\gamma) \cos(n(s_1), \Gamma-)}{\sin \alpha} \text{ when } \alpha \neq \pi \right. \\ & \left. \sin(\vec{n}(s_1), \vec{\Gamma}(s_1, s_1 - s_2)) \text{ otherwise} \right\} (-1)^{r(s_1, s_2)} + \\ & \int_0^1 \frac{\gamma dt}{\gamma^2 + (\alpha(s-t) - \lambda)^2} \frac{\sqrt{1 - \left(\vec{n}(s) \cdot \frac{\vec{\Gamma}(s-t)}{|\vec{\Gamma}(s-t)|} \right)^2}}{|\vec{\Gamma}(s-t)|} f(s-t) + \\ & \int_0^1 \frac{\gamma dt}{\gamma^2 + (\alpha(s+t) - \lambda)^2} \frac{\sqrt{1 - \left(\vec{n}(s) \cdot \frac{\vec{\Gamma}(s+t)}{|\vec{\Gamma}(s+t)|} \right)^2}}{|\vec{\Gamma}(s+t)|} f(s+t) \end{aligned} \quad (67)$$

Recall that the gradient of the energy is

$$\nabla E(\Gamma)(s) = \int d\lambda [H(\Gamma, \lambda) - H^*(\lambda)] \nabla H(\Gamma, \lambda)(s) \quad (68)$$

Using eq. (67), changing the order of integration and considering the limiting case of small γ , we obtain

$$\begin{aligned} \nabla E(\Gamma)(s) = & - \int_0^1 \left[\left\{ \frac{\cos(\beta) \cos(n(s_1), \Gamma+) + \cos(\gamma) \cos(n(s_1), \Gamma-)}{\sin \alpha} \text{ when } \alpha \neq \pi \right\} \times \right. \\ & \left. \frac{a(-1)^{r(s_1, s_2)}}{bc} [H(\Gamma, \alpha(s)) - H^*(\alpha(s))] + \right. \\ & - f(s-t) \frac{\sqrt{1 - \left(\vec{n}(s) \cdot \frac{\vec{\Gamma}(s-t)}{|\vec{\Gamma}(s-t)|} \right)^2}}{|\vec{\Gamma}(s-t)|} [H(\Gamma, \alpha(s-t)) - H^*(\alpha(s-t))] + \\ & \left. - f(s+t) \frac{\sqrt{1 - \left(\vec{n}(s) \cdot \frac{\vec{\Gamma}(s+t)}{|\vec{\Gamma}(s+t)|} \right)^2}}{|\vec{\Gamma}(s+t)|} [H(\Gamma, \alpha(s+t)) - H^*(\alpha(s+t))] \right] dt \end{aligned} \quad (69)$$

We finally obtain the closed form solution for the flow minimizing the energy corresponding to feature class #3. The computational complexity for the discrete curve with N nodes is $O(N^2)$.

References

- [1] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE trans. on PAMI*, vol. 24, 2002.

- [2] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” *International journal of computer vision*, vol. 22(1), pp. 61–79, 1997.
- [3] G. Charpiat, O. Faugeras, and R. Keriven, “Approximations of shape metrics and application to shape warping and empirical shape statistics,” Tech. Rep. RR-4820, INRIA, Sophia Antipolis, France, Mai 2003.
- [4] T. Cootes, C. Taylor, D. Cooper, and J. Graham, “Active shape models – their training and application,” *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.
- [5] I. Dryden and K. Mardia, *Statistical shape analysis*. John Wiley & Sons, 1998.
- [6] C. Y. Ip, D. Lapadat, L. Sieger, and W. Regli, “Using shape distributions to compare solid models,” 2002.
- [7] J. Kim, A. Tsai, M. Cetin, and A. Willsky, “A curve evolution-based variational approach to simultaneous image restoration and segmentation,” *Proc. of 2002 IEEE International Conference on Image Processing (ICIP)*, Rochester, September 2002.
- [8] E. Klassen, A. Srivastava, and W. Mio, “Analysis of planar shapes using geodesic paths on shape spaces,” *IEEE Trans. on PAMI*, in press.
- [9] M. Leventon, W. Eric L. Grimson, and O. Faugeras, “Statistical shape influence in geodesic active contours,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR’00)*, 2000.
- [10] M. E. Leventon, W. E. L. Grimson, O. Faugeras, and W. M. W. III, “Level set based segmentation with intensity and curvature priors,” *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis Proceedings (MMBIA’00)*, pp. 4–11, 2000.
- [11] A. Litvin and W. Karl, “Image segmentation based on prior probabilistic shape models,” *Proceedings of 2002 IEEE International Conference on Acoustic Speech and Signal Processing (ICASSP)*, 2002.
- [12] P. Maurel and G. Sapiro, “Dynamic shapes average,” *Proc. 2nd IEEE workshop on variational, geometric and level set methods in computer vision*, Nice, October 2003.
- [13] D. Mumford and J. Shah, “Boundary detection by minimizing functionals,” in *Proceedings of CVPR*, (San Francisco), pp. 22–26, June 1985.
- [14] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, “Matching 3d models with shape distributions,” *International Conference on Shape Modeling and Applications. ACM SIGGRAPH, the Computer Graphics Society and EUROGRAPHICS, IEEE Computer Society Press*, pp. 154–166, 2001.

- [15] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM transactions on graphics*, vol. 21(4), pp. 807–832, 2002.
- [16] S. M. Pizer, D. S. Fritsch, P. A. Yushkevich, V. E. Johnson, and E. L. Chaney, "Segmentation, registration, and measurement of shape variation via image object shape," *IEEE Transactions on Medical Imaging*, vol. 18(10), pp. 851–865, 1996.
- [17] M. Rousson and N. Paragios, "Shape priors for level-set representations," *European Conference on Computer Vision (ICCV'02)*, 2002.
- [18] G. Sapiro and V. Caselles, "Histogram modification via partial differential equations," *International Conference on Image Processing Proceedings(ICIP'95)*, 1995.
- [19] S. Sclaroff and L. Liu, "Deformable shape detection and description via model-based region grouping," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 5, pp. 475–489, 2001.
- [20] S.C.Zhu and A.Yuille, "Forms: A flexible object recognition and modeling system," *Int'l Journal of Computer Vision*, vol. 20, no. 3, pp. 187–212, 1996.
- [21] K. Siddiqi, Y. Lauzière, A. Tannenbaum, and S. Zucker, "Area and length minimizing flows for shape segmentation," *CVC TR-97-001/CS TR-1146*, February 1997.
- [22] S. S. Skiena, *The Algorithm Design Manual*. Springer-Verlag, New York, 1997.
- [23] R. Sternberg, *Cognitive Psychology*. Wadsworth Pub Co., 2002.
- [24] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla, "Shape context and chamfer matching in cluttered scenes," in *Proc. Conf. Computer Vision and Pattern Recognition*, vol. I, (Madison, USA), pp. 127–133, June 2003.
- [25] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, W. Grimson, and A. Willsky, "Model-based curve evolution technique for image segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, 2001.
- [26] A. Tsai, A. Yezzi, and A. S. Willsky, "Curve evolution implementation of the the mumford-shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Trans. Image Processing*, vol. 10(8), pp. 1169–1186, 2001.
- [27] G. Unal, H. Krim, and A. Yezzi, "Stochastic differential equations and geometric flows," *IEEE Transactions on Image Processing*, vol. 11(12), pp. 1405–1417, 2002.

-
- [28] Y. Wang and L. Staib, “Boundary finding with prior shape and smoothness models,” *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 7, pp. 738–743, 2000.
 - [29] S.-C. Zhu, “Embedding gestalt laws in markov random fields,” *IEEE trans. on PAMI*, vol. 21, no. 11, 1999.