Chapter 4 Curve, Region, and Surface Inference

Having defined the salient feature inference engine, we proceed to describe how to apply it to infer salient curves, regions, surfaces, and junctions. To make use of the salient feature inference engine, we need to define a voting function for estimating orientation at every site.

In this chapter, we first describe the derivation of the voting function for undirected curve and surface inference in section 4.1. Results of applying the salient structure inference engine with the derived voting functions to binary 2-D and 3-D input are also presented. In section 4.2, the problem of region inference is addressed.

4.1 Undirected Curve and Surface Inference

4.1.1 The derivation of the voting functions

In this section, we derive the voting function for curve and surface inference from local features such as points, curves elements and surface patch elements. The purpose of the voting function is to specify how curve, surface or region boundary orientation is estimated by a voting feature regarding each location in the domain space. Similar problems have been studied by Guy and Medioni [28], Geiger et al. [22, 23], Thornber and Williams [81], Williams and Jacobs [88], and Heitger and von der Heydt [33].

When designing the voting function, we always need to answer these two questions:

(1) What is the pattern of influence?
(2) How to determine the amount of influence?
Pattern of influence

For curve and surface inference, the goal is to find the best curve or surface that complies with local observations. As stated in section 3.3, we need to consider each of the voter’s orientation estimations individually. Therefore, to determine the pattern of influence, we only need to decide: What is the best curve/surface that connects a location to the input site such that its tangent/normal at the input site is equal to a particular orientation?

For smooth surfaces or surface borders interpolation, the orientations of the votes should change gradually in a local neighborhood. It means that any curve element with an orientation, say $\hat{u}$, must be connected to points in a local neighborhood through non-crossing, $C_1$-continuous paths with initial orientations equal to $\hat{u}$. From an energy consumption point of view, a circular path, which is the only $C_1$-continuous path that has constant curvature and is defined for all pairs of points and orientations, should be preferred over other smooth paths. Similarly, surface patch elements should be connected to other points through spherical surfaces.

Amount of influence

To determine the amount of influence, we need to observe that the interpolation of smooth structure requires the influence of a voter to decrease smoothly with distance. Moreover, low curvature is preferred over high curvature. The amount of voter’s influence should therefore attenuate with path length, and with curvature. There are many ways to combine these two factors but no intuitive constraint can be derived by considering the votes from a particular orientation alone. However, since the two attenuation factors interact when the votes of multiple orientation estimates of the voter are combined, we can determine the proper influence of each factor and their relationship by
considering the combined votes. In particular, when connecting points, that is, features with multiple orientation estimations that consist of all directions, some paths are more ‘natural’ than others. We therefore derive the voting strength decay function by applying the intuitive constraints for connecting points.

To illustrate how the voting strength decay function is derived, we use an analogy for curve interpolation in 2-D. Let’s consider the situation in which a particle $p_s$ radiates energy in all directions in 2-D, and energy travels in all circular paths tangent to each direction, as shown in Figure 4.1.

![Figure 4.1 Energy field analogy](image)

It is obvious that the energy field so created is isotropic. Consider the energy received by a particle $p_r$ at a location $l$ units from the sender $p_s$. If the energy is the same for all paths, and does not attenuate along each path, $p_r$ should receive the same amount of energy from all directions, regardless of the distance between $p_r$ and $p_s$. Moreover, the total amount of energy received by $p_r$ is independent of $l$. To enforce smoothness, energy must decay with path length $s$. A natural choice for the energy decay function is the Gaussian function. Our first version of the energy decay function is:

$$DF(s) = e^{\frac{s^2}{\sigma^2}} \text{ where } s = \frac{\theta l}{\sin \theta}$$

(4.1)

with $\theta$ being the initial direction of the path, and $\sigma$ the scale factor that determines the rate of attenuation.
To visualize the effect of this decay on the energy received by $p_r$ as it moves away from $p_s$, we use an ellipse to capture the energy distribution at the receiving site. Figure 4.2 shows that the norm of the ellipse decays smoothly with the parting distance $l$ while the eccentricity of the ellipse increases smoothly with $l$. The norm of the energy distribution ellipse, which measures the total amount of energy received, thus decays as desired. However, the eccentricity, which approximates the energy direction distribution, changes against our intuition. Such change in eccentricity means that as $p_r$ gets closer to $p_s$, energy arrives from all directions with more balanced strengths.

![Figure 4.2 Energy distribution against parting distance (1)](image)

Intuitively, when $p_r$ is close to $p_s$, energy should arrive in a straight path. As $p_r$ moves away from $p_s$, energy may arrive from more directions, with strength decreasing with the curvature of the path. When $p_r$ gets further from $p_s$, the decay along the path should take over and effectively make energy arrive from fewer directions. This means that the objective function of the eccentricity of the energy distribution must have a shape similar to the Laplacian of Gaussian. An intuitive way to approach this objective
function is to attenuate energy with the curvature $\rho$ of the path. Our second expression of the energy decay function is:

$$DF(s, \rho) = e^{-\left(\frac{s^2 + c\rho^2}{\sigma^2}\right)}$$

where $\rho = \frac{2\sin\theta}{l}$ (4.2)

with $s$, $\theta$, and $\sigma$ defined as before, and $c$ as the constant that reflects the relative weight of path length and curvature. Figure 4.3 shows the plot of norm and eccentricity of the energy distribution against the parting distance $l$, with $c = 0.01$. Note that this tuning of free parameters only needs to be done once.

![Figure 4.3 Energy distribution against parting distance (2)](image)

(a) norm vs. I  (b) eccentricity vs. I

*Figure 4.3 Energy distribution against parting distance (2)*

It is obvious that when energy decays as specified in (4.2), the energy field generated by $p_s$, shown in Figure 4.4(b), is the desired voting field for a voter with orientation estimates captured by a plate tensor. For 2-D curve inference, a plate tensor is associated with a point feature. The 2-D point field derived by Guy and Medioni [28] for 2-D curve inference from first principles thus is a first order approximation of the desired voting field.
Notice that when $p_s$ only radiates energy in one direction, energy arrives at each location from one direction only. Figure 4.4(a) depicts the corresponding energy field when energy decays as described in (4.2). This field corresponds to the votes cast by a stick voter and is exactly the same as the extension field for tangent voter as derived by Guy [28] from intuition for 2-D curve inference.

Applying this energy field analogy in 3-D, we obtain the same voting strength decay function for 3-D curve inference. Moreover, by considering the case where energy travels in spherical surfaces, the same voting strength decay function is derived for surface inference, except that the path length and curvature are defined differently.

Figure 4.4 Voting fields for stick and plate voters in 2-D
4.1.2 The voting fields specifying the voting functions

As described in section 3.3, we use 3 tensor fields to represent the voting function. Following the arguments above, the voting field for the stick tensor $T(1,0,0,0,0,0)$ therefore, in polar coordinates as shown below, is:

$$V_1(p(s, \theta, \phi)) = DF(s, p)\hat{\nu}\hat{\nu}^T$$

where

$$\rho = \frac{2\sin \theta}{l} \quad s = \frac{\theta l}{\sin \theta} \quad \hat{\nu} = \begin{bmatrix} \cos 2\theta \\ \sin 2\theta \cos \phi \\ \sin 2\theta \sin \phi \end{bmatrix}$$

in case of curve inference

and

$$\rho = \frac{2\cos \theta}{l} \quad s = \frac{\theta l}{\cos \theta} \quad \hat{\nu} = \begin{bmatrix} -\cos 2\theta \\ \sin 2\theta \cos \phi \\ \sin 2\theta \sin \phi \end{bmatrix}$$

in case of surface inference

The tensor voting fields for plate and ball tensor are obtained by applying equation (3.6) using the stick tensor voting fields defined. We illustrate in Figure 4.4 the stick and plate voting fields for 2-D curve inference. The voting fields for 3-D curve and surface inference are depicted in Figure 4.5.

4.1.3 Illustrations

Having defined the voting function for curve and surface inference, we illustrate the tensor voting process in 2-D. Figure 4.6 depicts two different configurations in which votes of two stick voters are combined. In the first configuration, the two stick voters are parallel and adjacent to each other. The saliency tensor field (figure 4.6(b)) obtained by combining the votes shown in figure 4.6(a) gives a map in which the best path for
Figure 4.5 Voting fields for stick, plate and ball voters in 3-D

(a) vote generation for stick component

(b) a cut of the voting field for stick component

(c) a cut of the voting field for plate component

(d) a cut of the voting field for ball component
Figure 4.6 Illustrations of the Tensor Voting Process
connecting the two voters is determined to be the straight line in the middle of the field. In the second configuration, the two stick voters are about 40° apart from each other. The best path for connecting the two is shown in figure 4.6(d), obtained by combining the votes depicted in figure 4.6(c).

### 4.1.4 Results

We have applied our salient feature inference engine to infer curves in 2-D and surfaces in 3-D from sparse, noisy data. Figure 4.7(a) is a binary image in which an ellipse and an open curve can be seen in a noisy background. After applying the tensor voting process to the image, a dense curve saliency map (figure 4.7(b)) and a dense junction saliency map (figure 4.7(d)) are obtained. The underlying salient curves (figure 4.7(c)) and junctions (figure 4.7(e)) can then be extracted using non-maximal suppression.

Figure 4.8(a) show an image of a 3-D object from which we obtained sparse 3-D measurement of the object surface. The data obtained is shown in figure 4.8(b). Applying tensor voting for surface inference, we have inferred the underlying surfaces (figure 4.8(c)) and surface discontinuities (figure 4.8(d)). Notice that surface orientations around the discontinuities are wrong, and need to be rectified by the structure integration process (figure 4.8(e)).

### 4.2 Region Inference

#### 4.2.1 Directed curve inference

Using the voting function defined in section 4.1, together with the inference of polarity saliency described in section 3.4, directed curves are inferred in a similar manner as the undirected curves. To illustrate the importance of polarity in directed curve inference, we consider the salient feature inference for a pair of filled circles and that for a pair of empty circles. Figure 4.9(b) shows the resulting directed curve saliency map for the pair...
Figure 4.7 Inference of curves in 2-D
Figure 4.8 An example of surface inference in 3-D
of the directed circles that describe the filled circle in Figure 4.9(a). Compare to the
curve saliency map shown in Figure 4.9(d) obtained for a pair of hollow circles in
Figure 4.9(a), the high agreement between the circles, which is acceptable for the emp-
ty circle pair but not the filled circle pair, is eliminated by properly incorporating the
polarity information.

![Filled circles](a) filled circles

![Directed curve saliency](b) directed curve saliency

![Circles](c) circles

![Curve saliency](d) curve saliency

*Figure 4.9 Empty and filled circle pairs*

An interesting effect of handling polarity and feature saliency separately is that they
can be combined to detect structures other than directed curves. Consider a thin ribbon
in 2-D. We can treat it as a region bounded by a contour with two straight portions very
close to each other, as shown in Figure 4.10. In fact, this is the way most edge detectors
represent a ribbon. The two straight portions of such a feature are parallel but of oppo-
site polarities, and thus produces high orientation saliency but low polarity saliency for
the site in between the contour. We therefore can recover the ribbon by locating the local
minima in the directed curve saliency map. Hence polarized and unpolarized features
can be handled uniformly in our framework of salient structure inference.

![Region boundaries](region boundaries)

![Directed curve saliency](directed curve saliency)

![Antiparallel curve](antiparallel curve)

*Figure 4.10 A thin ribbon*
4.2.2 Boundary Inference

While the incorporation of polarity allows us to infer regions properly, this polarity information is not always available. When data are sparse and irregular, local boundary detection becomes hard. A typical example is shown in Figure 4.12(a). Since boundary detection is about locating discontinuities, it is possible to use the salient feature inference engine to detect points on the boundary. In the spirit of our methodology, we again seek to compute at every site in the domain space a measure we call boundary saliency which relates the possibility of having the site being on the boundary of a region. A boundary point has the property that most of its neighbors are on one side. We therefore can identify boundary points by computing the directional distribution of neighbors at every data point, as illustrated in Figure 4.11(a). This local discontinuity estimation is similar to the orientation estimation for salient structure inference, except that accurate orientation estimate is irrelevant to discontinuity estimation and thus does not require the use of the full tensor, but only the vector part of it. Therefore the voting function for boundary inference can be characterized as a radiant pattern with strength decaying with distance from the center. We depict in Figure 4.11(b) the corresponding voting field which uses the Gaussian decay function.

![Figure 4.11 Boundary inference](image-url)
Since a point on the boundary will only receive votes from one side, while a point inside the region will receive votes from all directions, the size of the vector sum of the polarized votes should indicate the “boundariness” of a point, that is, the boundary saliency. On the other hand, the direction of the resulting vector relates to the polarity information at the site. Figure 4.12(b) and (c) presents the result of this boundary inference on the 2-D data set depicted in Figure 4.12(a). Observe that points are not labeled in absolute terms as borders or non-borders, but are given a degree of boundary saliency, which needs to be compared to their neighbors value.

Once the boundary points are identified, the problem is reduced to that of inferring directed curves. Using the boundary saliencies as the initial saliencies, the boundariness of the data points can be further verified by locating the surfaces/curves that describe the region boundaries. Figure 4.12(d) and (e) depict the computed directed curve saliency and junction saliency from the boundary information shown in Figure 4.12(b) and (c). The integrated description of the dot cluster is shown in Figure 4.12(f). Note that we not only find the region, but also accurately find the corners.
Figure 4.12  Example of region inference

(a) input  
(b) boundary saliency  
(c) boundary polarity  
(d) directed curve saliency  
(e) junction saliency  
(f) integrated description in terms of region and junctions