

# Model Driven Edge Detection

Pascal Fua and Yvan G. Leclerc

Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, California, USA

**Abstract:** Standard edge detectors fail to find most relevant edges, finding either too many or too few, because they lack a geometric model to guide their search. We present a technique that integrates both photometric and geometric models with an initial estimate of the boundary. The strength of this approach lies in the ability of the geometric model to overcome various photometric anomalies, thereby finding boundaries that could not otherwise be found. Furthermore, edges can be scored based on their goodness of fit to the model, thus allowing one to use semantic model information to accept or reject the edges.

---

**Key Words:** computer vision, model-based vision, edge detection, energy minimization

---

## 1 Introduction

In real-world images object boundaries cannot be detected solely on the basis of their photometry because of the presence of noise and various photometric anomalies. Thus, all methods for finding boundaries based on purely local statistical criteria are bound to err, finding either too many or too few edges based on arbitrary thresholds.

To supplement the weak and noisy local information, we consider the geometric constraints that object models can provide. We do this by describing a boundary as an elastic curve with a deformation energy derived from the geometric constraints, as suggested previously (Terzopoulos 1987; Kass et al. 1988). Local minima in this energy correspond to boundaries that match the object model exactly. We incorporate the photometric constraints by defining photometric energy as the average along the curve of a scalar energy field derived from the photomet-

ric model of an edge. Local minima in this energy correspond to boundaries that match the photometric model exactly. We find a candidate boundary by deforming the curve in such a way as to minimize its total energy, which is the sum of the deformation and photometric energies. We call this deformation process “optimizing” the curve. Once a curve has been optimized, that is, once it has settled in a local minimum of the total energy, a compromise between the two constraints has been effected. We can then utilize the object models to determine if the curve actually corresponds to an object boundary.

Such “energy-minimizing curves” have two key advantages:

- The geometric constraints are directly used to guide the search for a boundary.
- The edge information is integrated along the entire length of the curve, providing a large support *without including the irrelevant information of points not on the curve.*

Taken together, they allow energy-minimizing curves to find photometrically weak boundaries that local edge detectors simply could not find without also finding many irrelevant boundaries.

Energy-minimizing curves, as we call them, were originated by Terzopoulos, Kass, and Witkin (Terzopoulos 1987; Kass et al. 1988) as “snakes”. This paper was motivated by the need to solve two major problems with snakes. First, the precise relationship between the local minima found by these snakes and the standard definitions of an edge had not been elucidated. Second, snakes were basically designed for use in interactive environments. Consequently, they require fairly extensive parameter adjustments from one image to the next, and even from one part of an image to the next.

In this paper we offer three major contributions. (1) We show the precise relationship between the

---

*Address reprint requests to:* Pascal Fua, Artificial Intelligence Center, SRI International, 333 Ravenswood Avenue, Menlo Park, California 94025 USA.

local minima of the energy-minimizing curve and the standard definitions of an edge. (2) We present a procedure for automatically adjusting the parameters of these curves. (3) We briefly show how the relationship and the procedure can be combined in an automated system that extracts roads and buildings from aerial images, described extensively in Fua and Hanson (1988).

## 2 A Photometric Model: Step Edges

Several authors (Haralick 1984; Canny 1986; Torre and Poggio 1986) define step-edge points as those for which the first directional derivative of image intensity, in the direction of the gradient, is extremal. That is, an edge point  $(x_0, y_0)$  satisfies

$$\left. \frac{\partial^2 \mathcal{F}(x, y)}{\partial \mathbf{g}_0^2} \right|_{(x, y) = (x_0, y_0)} = 0 \quad (1)$$

where

$$\begin{aligned} \frac{\partial}{\partial \mathbf{g}_0} &= \mathbf{g}_0 \cdot \nabla \\ \mathbf{g}_0 &= \mathbf{g}(x_0, y_0) = \frac{\nabla \mathcal{F}(x_0, y_0)}{|\nabla \mathcal{F}(x_0, y_0)|} \end{aligned}$$

and  $\mathcal{F}(x, y)$  represents the image intensities after convolution with a twice differentiable kernel (typically a Gaussian).<sup>1</sup> It can be shown that equation (1) is equivalent to

$$\left. \frac{\partial |\nabla \mathcal{F}(x, y)|}{\partial \mathbf{g}_0} \right|_{(x, y) = (x_0, y_0)} = 0$$

In other words, the Haralick et al. criterion for an ideal step-edge point is equivalent to the criterion that the magnitude of the image intensity gradient be maximal in the gradient direction.

However, for noisy nonideal step edges the gradient direction may become very unreliable, especially at some distance from the edge itself. We therefore use the following somewhat more general definition of an edge:

**Definition** *An edge is a curve  $\mathcal{C}$  whose points have a gradient magnitude that is maximal in the direc-*

*tion normal to the curve. That is, all points along the curve (called edge points) satisfy*

$$\frac{\partial |\nabla \mathcal{F}(\mathbf{f}(s))|}{\partial \mathbf{n}(\mathbf{f}(s))} = 0 \quad (2)$$

where  $s$  is the arc-length of  $\mathcal{C}$ ,  $\mathbf{f}(s)$  is a vector function mapping the arc-length  $s$  to points  $(x, y)$  in the image, and  $\mathbf{n}(\mathbf{f}(s))$  is the normal to  $\mathcal{C}$ .

Note that this definition is equivalent to the Haralick et al. definition for ideal noise-free step edges.

To incorporate this photometric model into the energy-minimizing curve approach, we define the photometric energy of the curve  $\mathcal{C}$  as

$$\mathcal{E}_P(\mathcal{C}) = - \frac{1}{|\mathcal{C}|} \int_0^{|\mathcal{C}|} |\nabla \mathcal{F}(\mathbf{f}(s))| ds$$

where  $|\mathcal{C}|$  is the length of  $\mathcal{C}$ . For simplicity of notation, the limits on this integral will not be repeated in the remainder of the text.

To understand the relationship between optimized curves and edges, consider the following. When  $\mathcal{C}$  is either an open or closed curve that has been optimized, that is, when it is a local minimum of  $\mathcal{E}_P$  with respect to infinitesimal deformations of the curve, we prove in the Appendix that

$$\frac{\partial |\nabla \mathcal{F}(\mathbf{f}(s))|}{\partial \mathbf{n}(\mathbf{f}(s))} = \gamma(s) \left( |\nabla \mathcal{F}(\mathbf{f}(s))| - \frac{1}{|\mathcal{C}|} \int |\nabla \mathcal{F}(\mathbf{f}(s))| ds \right) \quad (3)$$

where  $\gamma(s)$  is the curvature of  $\mathcal{C}$ . Furthermore, we prove that when  $\mathcal{C}$  is an open curve that has been optimized,

$$|\nabla \mathcal{F}(\mathbf{f}(0))| = |\nabla \mathcal{F}(\mathbf{f}(|\mathcal{C}|))| = \frac{1}{|\mathcal{C}|} \int |\nabla \mathcal{F}(\mathbf{f}(s))| ds \quad (4)$$

also holds. These two equations have the following consequences.

Equation (3) implies that the points along an optimized curve are all edge points when either its curvature is zero (i.e., it is a straight-line segment) or the magnitude of the gradient along the curve is constant. *Consequently, optimized curves whose curvature is small or for which the magnitude of the gradient is approximately constant are good approximations to edges.* In particular, we have found that replacing  $|\nabla \mathcal{F}(x, y)|$  by  $\log |\nabla \mathcal{F}(x, y)|$  (which has no effect for ideal step edges) yields more stable solutions because the second term in equation (3) is smaller for nonconstant edge strengths, and hence the approximation is generally better.

For an open-ended curve, equation (4) must also be satisfied. This equation implies that the curve is stable only when the gradient magnitude at the end

<sup>1</sup> This definition is a good approximation to the position of discontinuities in the underlying intensity function when the diameter of support of the operator is much smaller than the radius of curvature of the edge. In particular, it is not applicable at corners and junctions of edges.

points equals the average. In other words, an open-ended energy-minimizing curve placed exactly along an edge will shrink or expand, following the edge, until this condition is met. When the gradient magnitude is constant, of course, this condition is always met and curves therefore remain unchanged. Thus, again, using  $\log |\nabla \mathcal{F}(x, y)|$  yields more stable solutions because this equation is more nearly satisfied for nonconstant edge strengths. An important note is that if we had chosen to use the integral of gradient magnitude, as opposed to the average, this equation would not hold: Curves would simply expand without bound.

After optimization candidate edges can be scored on how well they match the photometric model by computing the proportion of points along the curve that satisfy equation (2) to within some tolerance. We use this score in our applications to accept or reject any given edge.

### 3 A Geometric Model: Smooth Curves

#### 3.1 Theory

We have seen in the previous section that energy-minimizing curves match edges well wherever the curves have a low curvature. To ensure stable results, we define the deformation energy of such curves so that their curvature remains small and the minimum energy state is close to the photometric edge. This can be achieved simply by defining the deformation energy of a curve  $\mathcal{C}$  as

$$\mathcal{E}_D(\mathcal{C}) = \int \gamma(s)^2 ds$$

where  $\gamma(s)$  is the curvature of  $\mathcal{C}$ . The total energy of such a curve is then  $\mathcal{E}(\mathcal{C}) = \mathcal{E}_P(\mathcal{C}) + \lambda \mathcal{E}_D(\mathcal{C})$ , where  $\mathcal{E}_P(\mathcal{C})$  is the photometric energy defined in the previous section.

To delineate a photometric edge using one of these curves, one must provide an initial estimate of the location of the curve and then optimize it using the procedure described in the next subsection. Previously (Kass et al. 1988),  $\lambda$  was chosen interactively. We have found that a single value of  $\lambda$ , however carefully chosen, is inadequate for use across a wide variety of images. Instead, we compute  $\lambda$  in the following adaptive manner.

Let  $\mathcal{C}_i$  be the initial curve and  $\mathcal{C}_f$  be the curve after optimization. Let  $\delta_i$  denote the operator

$$\delta_i = \left. \frac{\delta}{\delta \mathcal{C}} \right|_{\mathcal{C}=\mathcal{C}_i}$$

that is,  $\delta_i$  is the variational operator evaluated at the initial curve. Define  $\delta_f$  similarly for the final curve.

Since, by definition,  $\delta_f \mathcal{E}(\mathcal{C}) = 0$ , then

$$\lambda = \frac{|\delta_f \mathcal{E}_P(\mathcal{C})|}{|\delta_f \mathcal{E}_D(\mathcal{C})|}$$

When the initial estimate is known to be close to the final answer, we choose

$$\lambda = \frac{|\delta_i \mathcal{E}_P(\mathcal{C})|}{|\delta_i \mathcal{E}_D(\mathcal{C})|}$$

so that we remain as close to the initial estimate as possible. If instead we want to smooth the initial estimate, we can use higher values of  $\lambda$ . This is in fact the simplest way of imposing a geometric model, in this case smoothness, upon the data.

#### 3.2 Implementation

In the actual implementation the curves are described as polygons with  $n$  equidistant vertices  $X = \{(x_i, y_i), i = 1, \dots, n\}$  and the deformation energy can be made discrete as

$$\mathcal{E}_D = \sum_i (2x_i - x_{i-1} - x_{i+1})^2 + (2y_i - y_{i-1} - y_{i+1})^2 \quad (5)$$

when  $\gamma(s)$  is small. To perform the optimization, we could use a simple gradient descent technique (Leclerc and Fua 1987), but it would be extremely slow for curves with a large number of vertices. Instead, we embed the curve in a viscous medium and solve the equation of dynamics:

$$\frac{\partial \mathcal{E}}{\partial X} + \alpha \frac{dX}{dt} = 0$$

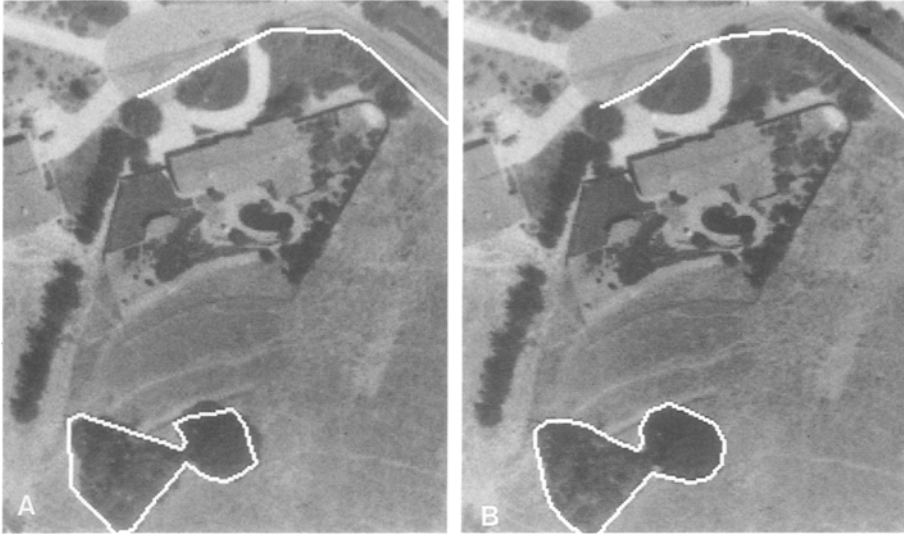
where  $\mathcal{E} = \mathcal{E}_P + \lambda \mathcal{E}_D$  and  $\alpha$  is the viscosity of the medium (Terzopoulos 1987). Since the deformation energy  $\mathcal{E}_D$  in equation 5 is quadratic, its derivative with respect to  $X$  is linear, and therefore

$$\frac{\partial \mathcal{E}_D}{\partial X} = KX$$

where  $K$  is a pentadiagonal matrix. Thus, each iteration of the optimization amounts to solving the linear equation

$$KX_t + \alpha(X_t - X_{t-1}) = \left. \frac{\partial \mathcal{E}_P}{\partial X} \right|_{X_{t-1}} \quad (6)$$

Because  $K$  is pentadiagonal, the solution to this set of equations can be computed efficiently in  $O(n)$  time using LU decomposition and backsubstitution. Note that the LU decomposition needs to be recomputed only when  $\alpha$  changes. When  $\alpha$  is constant,



**Figure 1.** (A) Initial estimates for both an open and a closed boundary; (B) final outlines after optimization.

only the backsubstitution step is required. For a curve with 100 vertices, the LU decomposition step takes 0.05 seconds on a 3675 Symbolics Lisp Machine with a Floating Point Accelerator, and the backsubstitution step takes only 0.012 seconds.

Previously (Terzopoulos 1987),  $\alpha$  was chosen interactively and remained fixed throughout the optimization process. Since this has proven to be inadequate for use across a variety of images, we use the following adaptive procedure for computing  $\alpha$ .

We start with an initial step size  $\Delta$  and compute the viscosity so that

$$\alpha = \frac{\sqrt{2n}}{\Delta} \left| \frac{\partial \mathcal{E}}{\partial X} \right|$$

where  $n$  is the number of vertices. This ensures that the initial displacement of each vertex is on the average of magnitude  $\Delta$ . Because of the nonlinear term, we must verify that the energy has decreased from one iteration to the next. If, instead, the energy has increased, the curve is reset to its previous position, the step size is decreased, and the viscosity is recomputed accordingly. This is repeated until the step size becomes less than some threshold value. In most cases, because of the presence of the linear term that propagates constraints along the whole curve in one iteration, it only takes a small number of iterations to optimize the initial curve. For example, going from the initial estimate of the closed curve with about 20 vertices shown in Figure 1a to the optimized result shown in Figure 1b took 20 iterations and 3.9 seconds on a 3675 Symbolics Lisp Machine with a Floating Point Accelerator. Only 18 iterations and 2.8 seconds were required for the open curve. Note that the time actually spent

solving the equations of the dynamics represents only a relatively small fraction of the total time required to perform the complete optimization; most of the time is actually spent updating the snakes and fetching the image values.

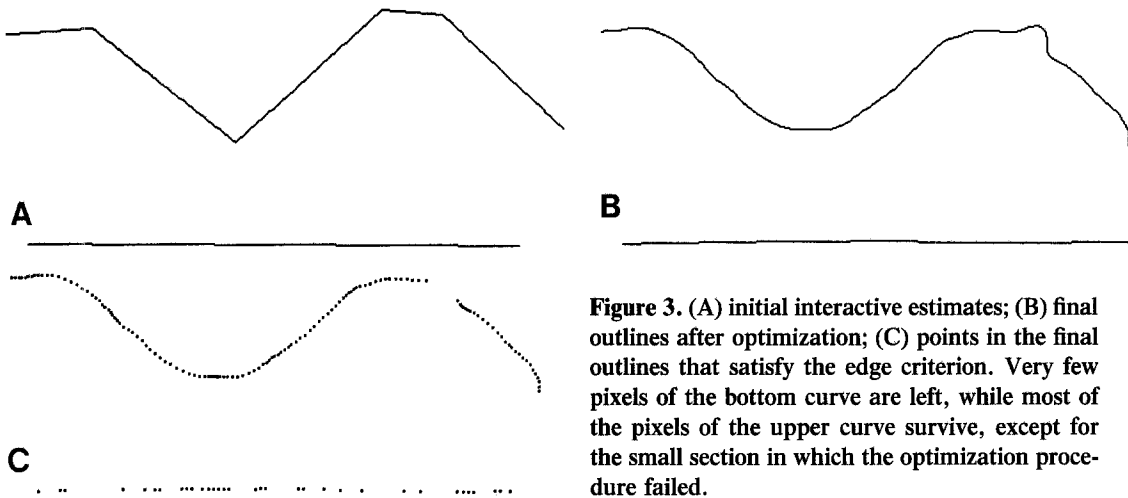
### 3.3 Evaluation Criterion

To illustrate the utility of our edge-point criterion [equation (2)], consider the synthetic image shown in Figure 2, wherein one boundary of the central region is smooth while the other is jagged. Two energy-minimizing curves were entered interactively (Figure 3a) and optimized using the smoothness constraint above (Figure 3b). Removing all points along the optimized curves that do not satisfy equation (2) yields Figure 3c.

Although most of the pixels belonging to the smooth boundary satisfy the edge criterion, very few of those belonging to the jagged boundary do. Also note that a small part of the topmost curve



**Figure 2.** A synthetic image generated by adding white noise of variance 20 to a foreground with mean value 150 and a background with mean value 100. The lower and upper boundaries of the foreground are sinusoids of different frequencies.



**Figure 3.** (A) initial interactive estimates; (B) final outlines after optimization; (C) points in the final outlines that satisfy the edge criterion. Very few pixels of the bottom curve are left, while most of the pixels of the upper curve survive, except for the small section in which the optimization procedure failed.

failed to settle correctly along the edge because of noise. However, this failure is easily identified because it is the only part of the curve for which a significant fraction of the points fail to satisfy the criterion.

Thus, computing the proportion of edge points in the optimized edges is an efficient and reliable way to measure how well the photometric boundaries match the smoothness constraint.

## 4 Applications

In this section we present two applications of our energy-minimizing curves. First, we describe an application of our model to road delineation in an interactive context, where the user provides a rough initial estimate of the location of the contours. This interactive procedure has been integrated into the SRI Cartographic Modeling Environment (Hanson and Quam 1988). It allows a cartographer to sketch roughly features of interest and then interactively adjust the resultant optimized contours. We then outline an application of our model to road and building delineation in the context of the fully automated system described in detail in Fua and Hanson (1988).

### 4.1 Interactive Road Delineation

We can apply the smooth curve model of the previous section directly to the problem of finding road boundaries in aerial images. We model roads as ribbons whose smoothly curved edges are approximately parallel. A ribbon is implemented as a polygonal curve forming the center of the road. Associated with each vertex  $i$  of this curve is a width  $w_i$ , which defines the two curves that are the candidate road boundaries. The photometric energy is defined

as the sum of the photometric energies of the two boundary curves, and the deformation energy is the sum of the smoothness term described earlier and an additional term that enforces the parallel constraint:

$$\sum_i (w_i - w_{i-1})^2$$

We have applied this model to two road segments in the aerial image of Figure 4a. Figure 4b shows the initial estimates for two road segment boundaries and Figure 4c shows the final optimized boundaries.

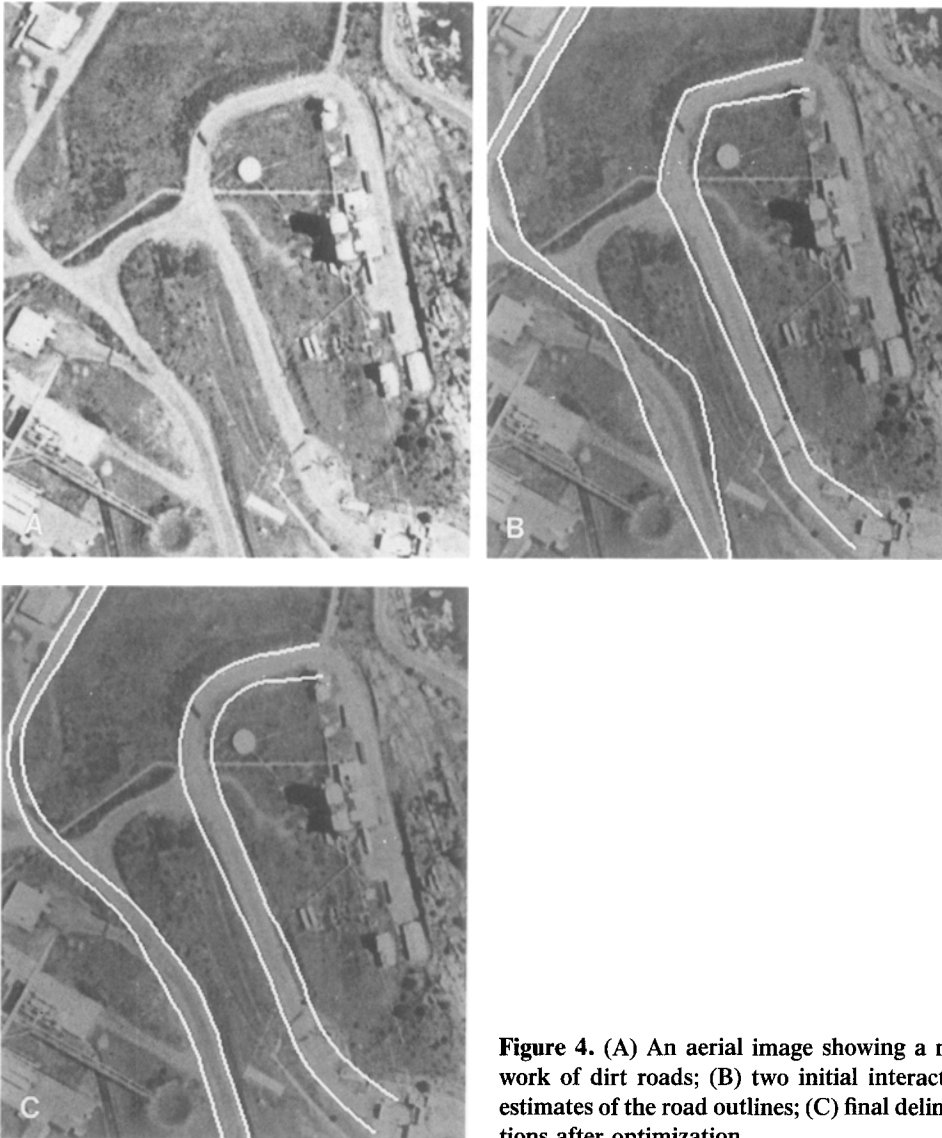
This image is an excellent example of a situation where a local edge detector such as the Canny edge detector is insufficient, by itself, to find the relevant boundaries. Because the roads are dirt roads with ill-defined edges, no single edge-strength threshold yields a satisfactory set of boundaries: When the threshold is too high, most of the road edges are lost, whereas when the threshold is too low, there is a plethora of irrelevant edges (see Figure 5).

In the next section we briefly describe an automated system that exploits the power of energy-minimizing curves to overcome this problem.

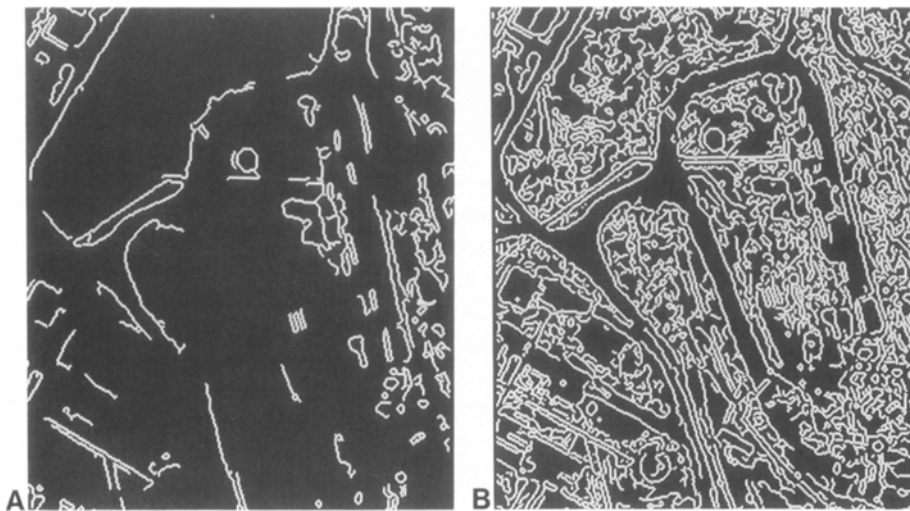
### 4.2 Automated Road and Building Delineation

In our automated system (Fua and Hanson 1988) roads are modeled as described previously, while buildings are modeled as polygons whose edges meet at right angles.

We begin by using the synthetic image of Figure 2 to demonstrate how the system uses energy-minimizing curves to extract smooth and straight edges from a set of Canny edge maps. Since a full description of the system is beyond the scope of this paper,



**Figure 4.** (A) An aerial image showing a network of dirt roads; (B) two initial interactive estimates of the road outlines; (C) final delineations after optimization.



**Figure 5.** Edge images generated by the Canny edge detector for the dirt road image with two different sets of edge-strength thresholds. Note that when the thresholds are too high as in (A) most edges are lost, whereas many irrelevant edges are found when the thresholds are dropped as in (B).

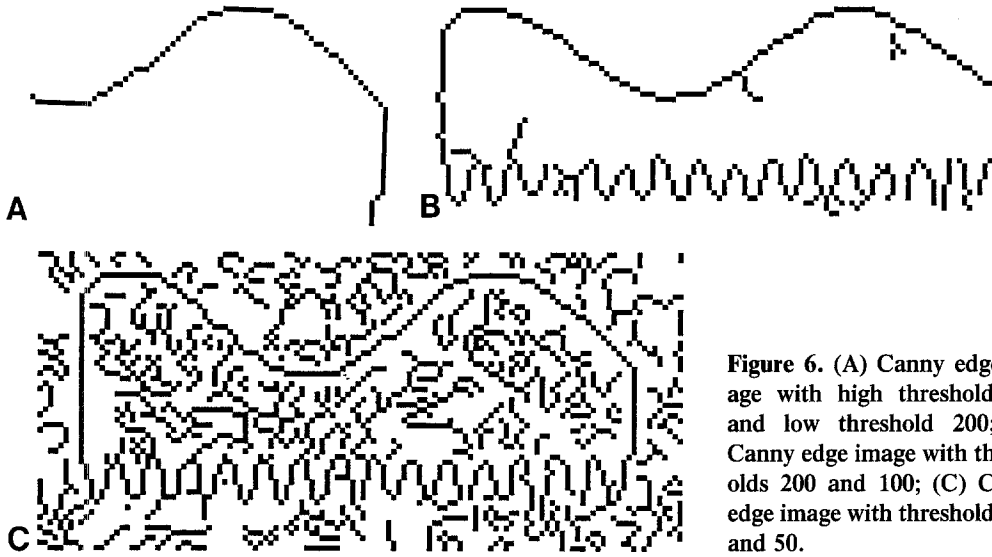


Figure 6. (A) Canny edge image with high threshold 400 and low threshold 200; (B) Canny edge image with thresholds 200 and 100; (C) Canny edge image with thresholds 100 and 50.

we can only briefly outline the subsequent steps that lead to the generation of road and building outlines, and show the results of this system applied to an aerial image.

*4.2.1 Smooth edges.* The system begins by applying the Canny edge detector to the image using several thresholds (Figure 6). The pixels in each of these maps are then linked using a linker (Fischler and Wolf 1983). Each of these linked segments is used as the starting point for an energy-minimizing curve. After optimization the curves are broken into segments wherever sharp bends are found. Those segments for which at least 70 percent of the points satisfy the edge criterion are accepted as candidates for the next step (Figures 7a, 7b, and 7c). This 70 percent figure has been used for all examples in this paper, and indeed for every other application of the automated system. Experimentally, curves with less than 50 percent of the points satisfying the edge criterion are almost always spurious, whereas curves with more than 90 percent are almost always correct. We chose 70 percent as a compromise that rejects most of the spurious edges without losing a significant number of correct ones.

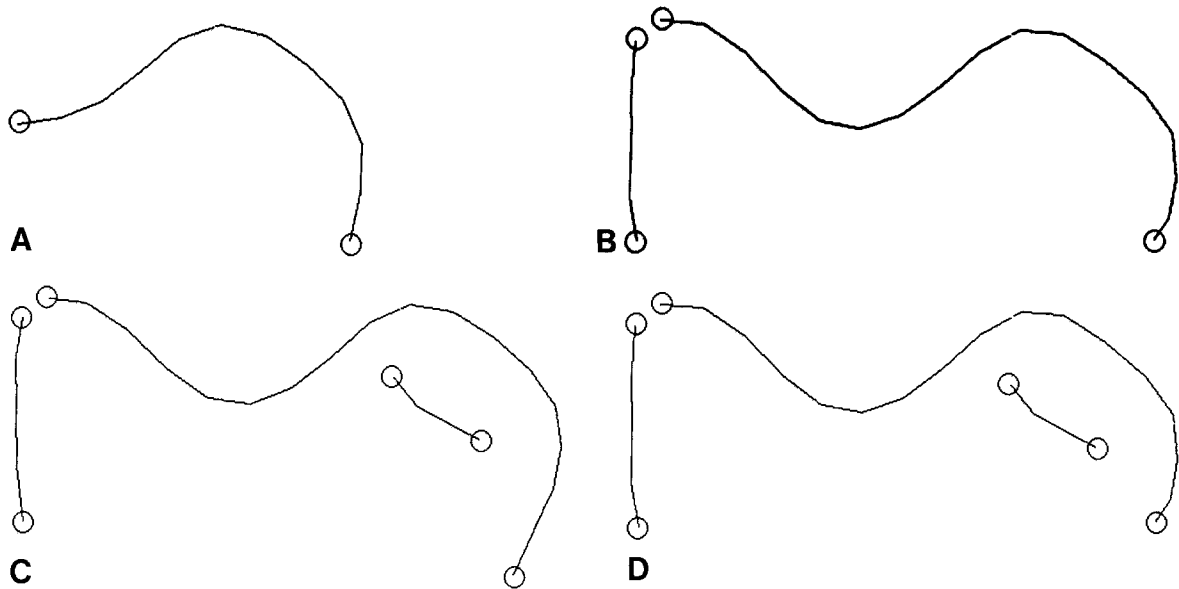
Since the same edge may be discovered, in whole or in part, in several of the Canny edge maps the system must choose an appropriate subset of these curves. The basic goal of this selection process is to choose the longest possible curves that do not overlap and have good edge quality. This goal is achieved by selecting the largest subset of nonoverlapping curves that maximize the number of pixels satisfying the edge criterion minus the number of

pixels that do not. The result of this selection process is shown in Figure 7d.

This technique provides an effective way of selecting the smooth boundaries in the image. Furthermore, by combining the output of the Canny edge detector using several arbitrary thresholds, we have eliminated the necessity of carefully choosing an appropriate threshold. This is a substantial advantage since, as discussed previously, choosing this threshold is in general a very difficult task.

*4.2.2 Straight edges.* The system begins with the same linked Canny edge images as before, finds points of high curvature in the linked segments, and fits straight lines between those points. The straight lines are used as the starting points for energy-minimizing curves with only two vertices. As previously, the curves are optimized and only those with 70 percent of points satisfying the edge criterion are retained (Figures 8a, 8b, and 8c). A subset of these curves is selected using the same procedure as before (Figure 8d). Note that in this particular case all the final curves could have been extracted from a single Canny edge map. This rarely occurs in practice because the noise across a real image is never as uniform as it is in this synthetic image.

*4.2.3 Automatic road delineation.* Applying the smooth edge technique to the aerial image of Figure 9 yields the edges shown in Figure 10a. These edges are then grouped into sets of parallel edges, each of which is used as the starting point for a ribbon, as described in section 4.1. After optimization the system uses a selection criterion to retain



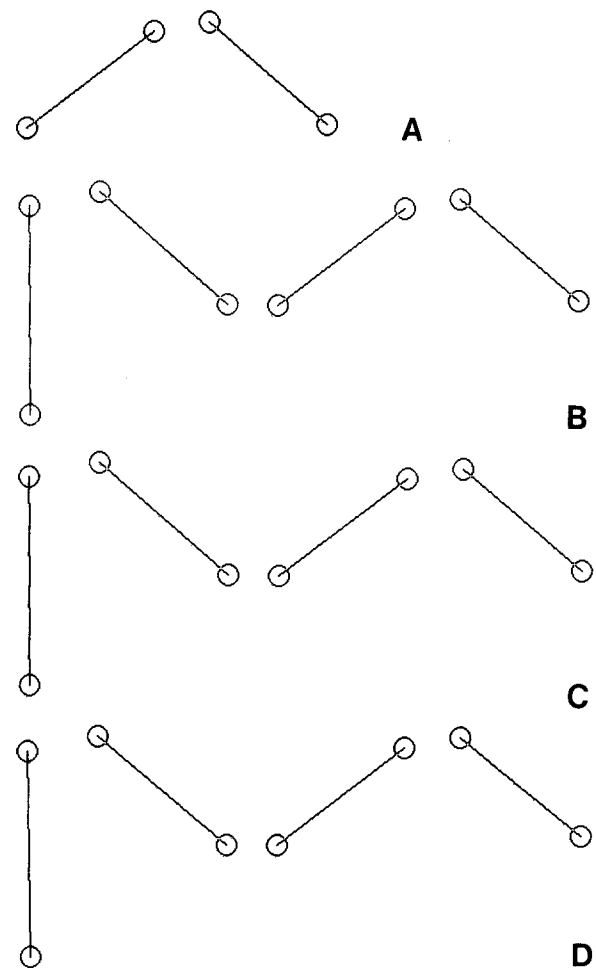
**Figure 7.** (A), (B), (C) Smooth edges extracted from the first, second, and third canny edge images, respectively; (D) final result generated by selecting the best nonoverlapping edges. Note that the longest edge that is retained is the one that was initially found in the second edge image.

only the best road candidates (Figure 10c). The groups from which these candidates arose are shown in Figure 10b.

*4.2.4 Automatic building delineation.* Applying the straight-edge technique to the same aerial image yields the edges shown in Figure 11a. These edges are then grouped into rectilinear edge structures and closed using a simple edge tracker. Finally, the outline of these rectilinear contours is optimized by using an energy-minimizing curve and a rectilinearity constraint. Once again, the system uses a selection criterion to retain only the best building candidates (Figure 11c). The groups from which these candidates arose are shown in Figure 11b.

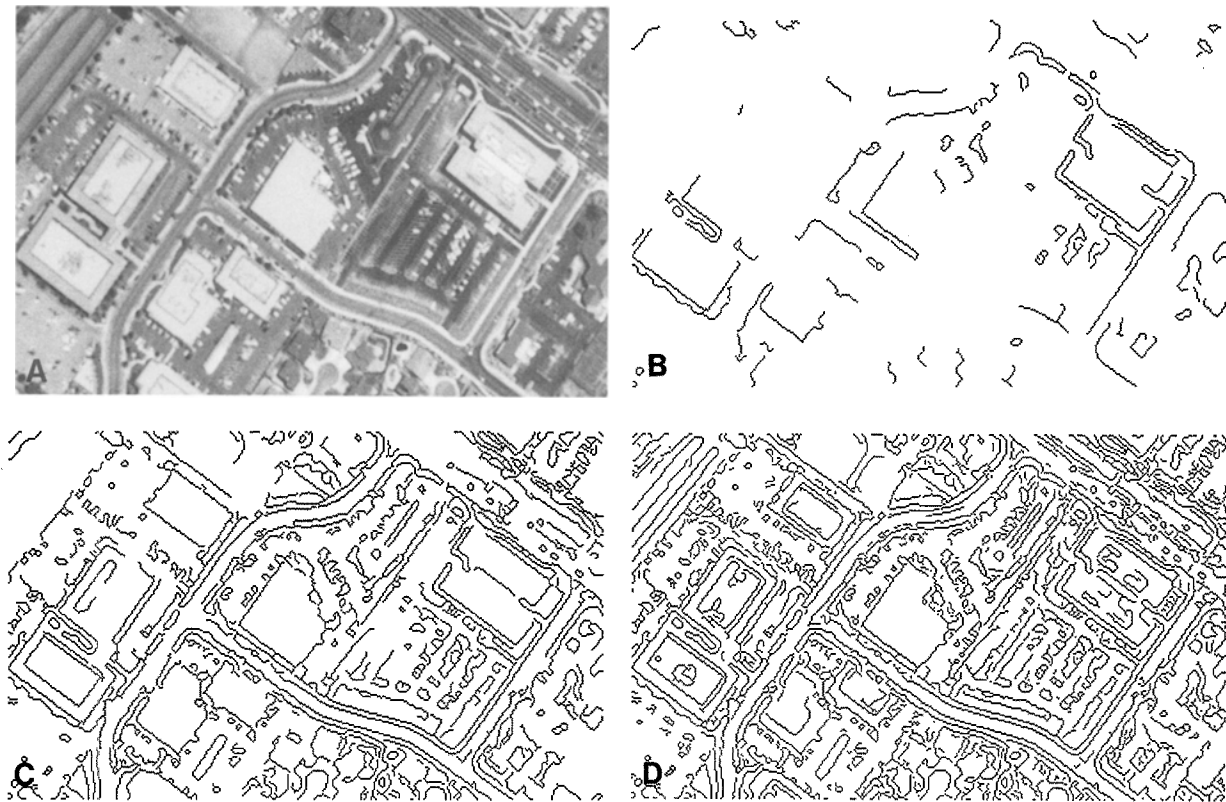
## 5 Summary and Conclusion

We have presented a technique for finding object boundaries that integrates both photometric and geometric models with an initial estimate of the boundary. The models are incorporated by defining an energy function that is minimal when the models are satisfied exactly. The initial estimate is used as the starting point for finding a local minimum of this energy function by embedding the initial curve in a

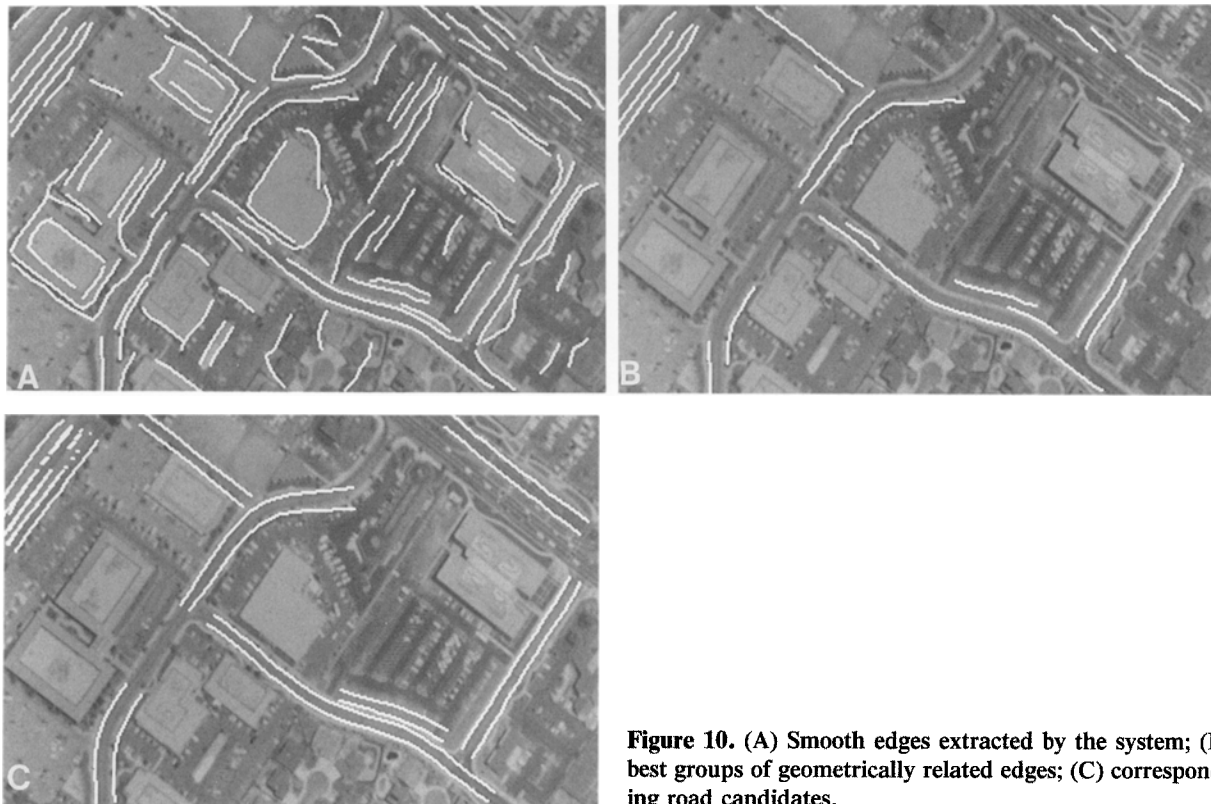


**Figure 8.** (A), (B), (C) Straight edges extracted from the first, second, and third canny edge images, respectively; (D) final result generated by selecting the best nonoverlapping edges. In this case (B), (C), and (D) are identical.

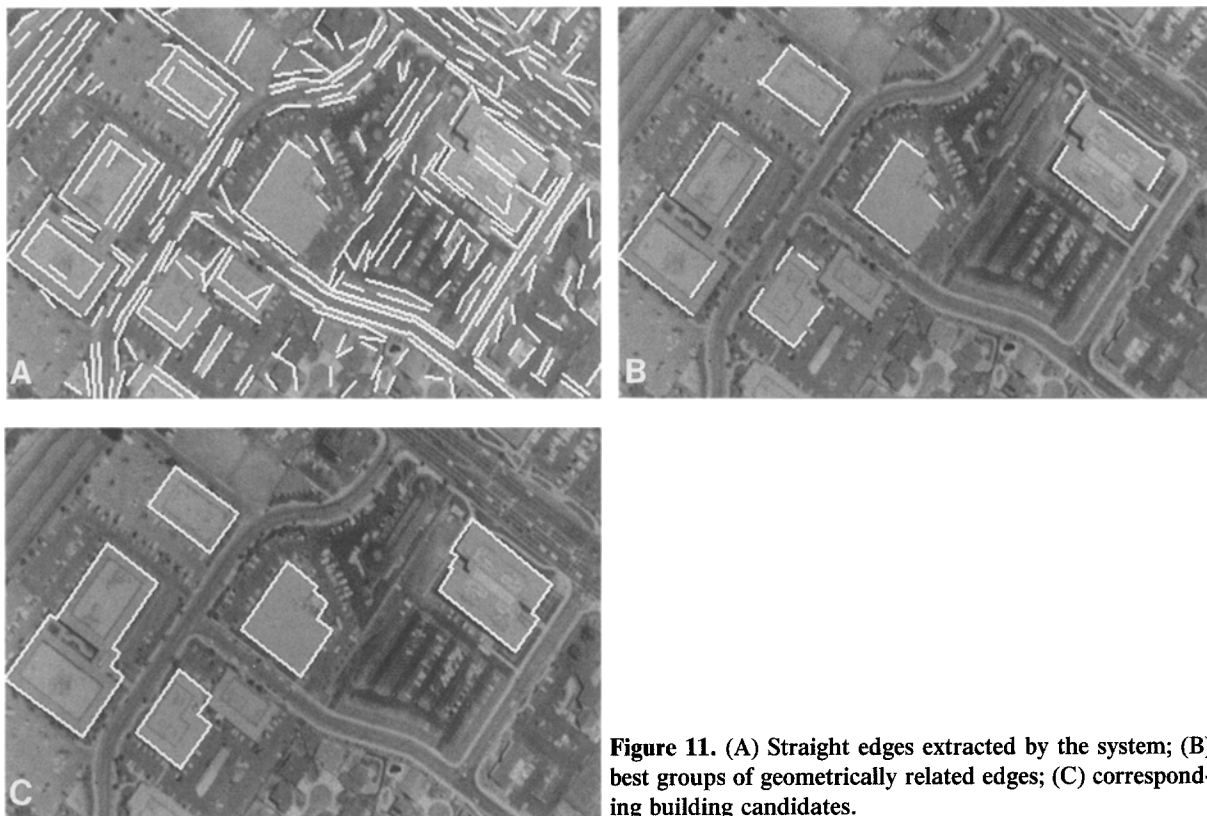




**Figure 9.** (A) An aerial scene with industrial buildings and roads. (B), (C), (D) Canny edge images with thresholds (400, 200), (200, 100), and (100, 50), respectively. No single setting of the edge parameters can be used to extract all relevant edges.



**Figure 10.** (A) Smooth edges extracted by the system; (B) best groups of geometrically related edges; (C) corresponding road candidates.



**Figure 11.** (A) Straight edges extracted by the system; (B) best groups of geometrically related edges; (C) corresponding building candidates.

viscous medium and solving the equations of dynamics.

The strengths of this “energy-minimizing curve” approach are that the geometric constraints are directly used to guide the search for a boundary and that the edge information is integrated along the entire length of the curve, thereby providing a large support.

We have shown the precise relationship between optimized curves and a standard definition of an edge and that this relationship can be used to determine when an optimized curve should be accepted as a candidate edge for further processing. We have also presented methods for automatically choosing the optimization parameters and how they should change over time. We found that these contributions were crucial to our ability to embed energy-minimizing curves in the automated system briefly discussed here and presented in detail elsewhere (Fua and Hanson 1988); these methods have provided us with the ability to find photometrically weak boundaries that local edge detectors simply could not find without also finding many irrelevant boundaries.

*Acknowledgments.* The work reported here was partially supported by the Defense Advanced Research Projects

Agency under contracts DACA76-85-C-0004 and MDA903-86-C-0084. We wish to thank Demetri Terzopoulos for the many discussions we have had concerning this work. An earlier version of this paper appeared in the Proceedings of the 1988 DARPA Image Understanding Workshop.

## References

- Canny J (1986) A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Vision*, 8:679–698
- Fischler M, Wolf H (1983) Linear delineation. In: *Proceedings of the IEEE Conference on Computer Vision and Image Processing*, Washington, D.C., pp. 351–356
- Fua P, Hanson A (1988) Extracting generic shapes using model-driven optimization. In: *Proceedings of the 1988 DARPA Image Understanding Workshop*, Boston, Massachusetts, pp. 994–1004
- Hanson AJ, Quam L (1988) Overview of the SRI cartographic modeling environment. In: *Proceedings of the Image Understanding Workshop*, Boston, Massachusetts, pp. 576–582
- Haralick RM (1984) Digital step edges from zero crossings of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Vision*, 6:58–68
- Kass M, Witkin A, Terzopoulos D (1988) Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331

- Leclerc YG, Fua P (1987) Finding object boundaries using guided gradient ascent. Topical Meeting on Machine Vision, Technical Digest Series, Optical Society of America, Washington, D.C., Vol. 12, pp. 168–171
- Terzopoulos D (1987) On matching deformable models to images. Topical Meeting on Machine Vision, Technical Digest Series, Optical Society of America, Washington, D.C., Vol. 12, pp. 160–167
- Torre V, Poggio T (1986) On edge detection. IEEE Transactions on Pattern Analysis and Machine Vision, 8: 147–163

## Appendix

In this appendix, we show that a necessary and sufficient condition for an open curve  $\mathcal{C}$  to be a local extremum of

$$\mathcal{E}(\mathcal{C}) = \frac{\int G(\mathbf{f}(s)) ds}{\int ds} \quad (1)$$

with respect to all infinitesimal deformations is that

$$\frac{dG(\mathbf{f}(s))}{d\mathbf{n}(s)} = \gamma(s) \left[ G(\mathbf{f}(s)) - \frac{\int G(\mathbf{f}(s)) ds}{\int ds} \right] \quad (2)$$

and

$$G(\mathbf{f}(0)) = G(\mathbf{f}(\mathcal{C})) = \frac{\int G(\mathbf{f}(s)) ds}{\int ds} \quad (3)$$

where the curve  $\mathcal{C}$  is parameterized by its arc-length  $s$ ;  $\mathbf{f}(s)$  is a vector function mapping the arc-length  $s$  to points  $(x, y)$  along the curve;  $\mathbf{n}(s)$  is the normal to the curve;  $\mathbf{t}(s)$  is the tangent to the curve; and  $\gamma(s)$  is its curvature. Furthermore, equation (2) alone is the necessary and sufficient condition for closed curves.

To prove this result, consider deformations of the curve  $\mathcal{C}$ , which we shall call  $\mathcal{C}_\lambda$ , such that the mapping from arc-length  $s$  to points  $(x, y)$  is of the form

$$\mathbf{f}_\lambda(s) = \mathbf{f}(s) + \lambda(\eta(s)\mathbf{n}(s) + \tau(s)\mathbf{t}(s)) \quad (4)$$

where  $\eta(s)$  and  $\tau(s)$  are arbitrary continuous and differentiable functions. When  $\mathcal{C}$  is a closed curve,  $\eta(s)$  and  $\tau(s)$  are constrained to be periodic of period  $|\mathcal{C}|$ . Since  $\mathcal{C}$  is a local extremum of  $\mathcal{E}(\mathcal{C})$  if and only if

$$\left. \frac{d\mathcal{E}(\mathcal{C}_\lambda)}{d\lambda} \right|_{\lambda=0} = 0 \quad (5)$$

for all  $\eta(s)$  and  $\tau(s)$ , we now prove the following theorem.

**Theorem.** Equation (5) holds for all  $\eta(s)$  and  $\tau(s)$  if and only if equations (2) and (3) are satisfied.

**Proof.** By definition,

$$\mathcal{E}(\mathcal{C}_\lambda) = \frac{\int G(\mathbf{f}_\lambda(s)) ds_\lambda}{\int ds_\lambda} \quad (6)$$

where  $s_\lambda$  is the arc-length of  $\mathcal{C}_\lambda$ . We can rewrite this in terms of  $s$  by using the equality

$$ds_\lambda = |\mathbf{f}'_\lambda(s)| ds \quad (7)$$

Thus,

$$\mathcal{E}(\mathcal{C}_\lambda) = \frac{\int G(\mathbf{f}_\lambda(s)) |\mathbf{f}'_\lambda(s)| ds}{\int |\mathbf{f}'_\lambda(s)| ds} \quad (8)$$

and therefore, to within the nonzero factor of  $1/(\int |\mathbf{f}'_\lambda(s)| ds)^2$

$$\begin{aligned} \frac{d\mathcal{E}(\mathcal{C}_\lambda)}{d\lambda} &= \frac{d}{d\lambda} \left[ \int G(\mathbf{f}_\lambda(s)) |\mathbf{f}'_\lambda(s)| ds \right] \int |\mathbf{f}'_\lambda(s)| ds \\ &\quad - \left[ \int G(\mathbf{f}_\lambda(s)) |\mathbf{f}'_\lambda(s)| ds \right] \frac{d}{d\lambda} \int |\mathbf{f}'_\lambda(s)| ds \\ &= \left[ \int \frac{dG(\mathbf{f}_\lambda(s))}{d\lambda} |\mathbf{f}'_\lambda(s)| + G(\mathbf{f}_\lambda(s)) \frac{d|\mathbf{f}'_\lambda(s)|}{d\lambda} ds \right] \\ &\quad \times \int |\mathbf{f}'_\lambda(s)| ds - \left[ \int G(\mathbf{f}_\lambda(s)) |\mathbf{f}'_\lambda(s)| ds \right] \\ &\quad \times \int \frac{d|\mathbf{f}'_\lambda(s)|}{d\lambda} ds \end{aligned} \quad (9)$$

The first term we need to evaluate in equation (9) is  $d|\mathbf{f}'_\lambda(s)|/d\lambda$ , for which we need the following equalities:

$$\frac{d\mathbf{f}(s)}{ds} = \mathbf{f}'(s) = \mathbf{t}(s)$$

$$\frac{d\mathbf{t}(s)}{ds} = \mathbf{t}'(s) = \gamma(s)\mathbf{n}(s)$$

$$\frac{d\mathbf{n}(s)}{ds} = \mathbf{n}'(s) = -\gamma(s)\mathbf{t}(s)$$

Therefore,

$$\begin{aligned} \mathbf{f}'_\lambda(s) &= \mathbf{f}'(s) + \lambda[\eta(s)\mathbf{n}(s) + \eta(s)\mathbf{n}'(s) + \tau'(s)\mathbf{t}(s) \\ &\quad + \tau(s)\mathbf{t}'(s)] \\ &= \mathbf{t}(s) + \lambda[\eta'(s)\mathbf{n}(s) - \eta(s)\gamma(s)\mathbf{t}(s) \\ &\quad + \tau'(s)\mathbf{t}(s) + \tau(s)\gamma(s)\mathbf{n}(s)] \\ &= \mathbf{t}(s)[1 + \lambda(\tau'(s) - \eta(s)\gamma(s))] \\ &\quad + \mathbf{n}(s)[\lambda(\eta'(s) + \tau(s)\gamma(s))] \end{aligned} \quad (10)$$

Since  $\mathbf{t}(s)$  and  $\mathbf{n}(s)$  are, by definition, orthogonal unit vectors, we have

$$\begin{aligned} |\mathbf{f}'_\lambda(s)| &= \sqrt{[1 + \lambda(\tau'(s) - \eta(s)\gamma(s))]^2 + [\lambda(\eta'(s) + \tau(s)\gamma(s))]^2} \end{aligned} \quad (11)$$

and therefore,

$$\begin{aligned} \frac{d|\mathbf{f}'_\lambda(s)|}{d\lambda} &= \frac{[1 + \lambda(\tau'(s) - \eta(s)\gamma(s))](\tau'(s) - \eta(s)\gamma(s)) + \lambda(\eta'(s) + \tau(s)\gamma(s))^2}{\sqrt{[1 + \lambda(\tau'(s) - \eta(s)\gamma(s))]^2 + [\lambda(\eta'(s) + \tau(s)\gamma(s))]^2}} \end{aligned} \quad (12)$$

The second term we need to evaluate is

$$\begin{aligned} \frac{dG(\mathbf{f}_\lambda(s))}{d\lambda} &= (\eta(s)\mathbf{n}(s) + \tau(s)\mathbf{t}(s)) \cdot \nabla G(\mathbf{f}_\lambda(s)) \\ &= \eta(s) \frac{dG(\mathbf{f}_\lambda(s))}{d\mathbf{n}(s)} + \tau(s) \frac{dG(\mathbf{f}_\lambda(s))}{d\mathbf{t}(s)} \end{aligned} \quad (13)$$

Substituting equations (11), (12), and (13) into equation (9), evaluating at  $\lambda = 0$ , and multiplying by the nonzero factor  $1/(\int |\mathbf{f}'_\lambda(s)| ds)^2$ , we obtain

$$\begin{aligned} \frac{d\mathcal{E}(\mathcal{C}_\lambda)}{d\lambda} \Big|_{\lambda=0} = 0 &= \int \eta(s) \frac{dG(\mathbf{f}(s))}{d\mathbf{n}(s)} + \tau(s) \frac{dG(\mathbf{f}(s))}{d\mathbf{t}(s)} \\ &\quad + G(\mathbf{f}(s))(\tau'(s) - \eta(s)\gamma(s)) ds \int ds \\ &\quad - \int G(\mathbf{f}(s)) ds \int (\tau'(s) - \eta(s)\gamma(s)) ds \end{aligned}$$

Dividing by  $\int ds$  and rearranging terms, we have

$$\begin{aligned} 0 &= \int \eta(s) \left[ \frac{dG(\mathbf{f}(s))}{d\mathbf{n}(s)} - \gamma(s) \left( G(\mathbf{f}(s)) - \frac{\int G(\mathbf{f}(s)) ds}{\int ds} \right) \right] ds \\ &\quad + \int \tau(s) \frac{dG(\mathbf{f}(s))}{d\mathbf{t}(s)} + \tau'(s) \left[ G(\mathbf{f}(s)) - \frac{\int G(\mathbf{f}(s)) ds}{\int ds} \right] ds \end{aligned}$$

A necessary and sufficient condition for the first integral to be zero for all  $\eta(s)$  is that the term multiplying  $\eta(s)$  be zero, which is equation (2).

Integrating the second integral by parts, we have

$$\begin{aligned} \int \tau(s) \frac{dG(\mathbf{f}(s))}{d\mathbf{t}(s)} ds + \tau(s) \left( G(\mathbf{f}(s)) - \frac{\int G(\mathbf{f}(s)) ds}{\int ds} \right) \Big|_0^{|\mathcal{C}|} \\ - \int \tau(s) \frac{dG(\mathbf{f}(s))}{d\mathbf{t}(s)} ds = 0 \end{aligned} \quad (14)$$

therefore,

$$\begin{aligned} \tau(|\mathcal{C}|) \left( G(\mathbf{f}(|\mathcal{C}|)) - \frac{\int G(\mathbf{f}(s)) ds}{\int ds} \right) \\ - \tau(0) \left( G(\mathbf{f}(0)) - \frac{\int G(\mathbf{f}(s)) ds}{\int ds} \right) = 0 \end{aligned} \quad (15)$$

A necessary and sufficient condition for this equality to hold for all  $\tau(s)$  is that both terms be zero, which is equation (3).

Thus, a necessary and sufficient condition for an open curve  $\mathcal{C}$  to be a local extremum of  $\mathcal{E}(\mathcal{C})$  is that both equations (2) and (3) hold.

Furthermore, for closed curves,  $\tau(0) \equiv \tau(|\mathcal{C}|)$  and  $G(\mathbf{f}(0)) \equiv G(\mathbf{f}(|\mathcal{C}|))$ , hence equation (15) is always satisfied. Therefore, a necessary and sufficient condition for a closed curve  $\mathcal{C}$  to be a local extremum of  $\mathcal{E}(\mathcal{C})$  is that equation (2) holds.