An Optimization Framework for Feature Extraction *

P. Fua* and A.J. Hanson†

* SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025, USA
fua@ai.sri.com

† Department of Computer Science
Indiana University
Bloomington, IN 47405, USA
hanson@cs.indiana.edu

Machine Vision and Applications, 4(2):59–87, Spring 1991

Abstract

In this paper, we propose a unified optimization framework for feature extraction that lets us simultaneously take into account image data and semantic knowledge: We model objects using a language that specifies both photometric and geometric constraints and define an information-theoretic objective function that measures the fit of the models to the data. We then treat the problem of finding objects as one of generating the optimal description of the image in terms of this language.

We have validated our framework by performing extensive experiments on detecting objects in aerial imagery described by simple geometric constraints, and have developed two algorithms for generating optimal descriptions. The first starts with a rough sketch of a polygonal object and deforms the initial contour to maximize the objective function, thus finding object outlines. The second automatically extracts complex rectilinear buildings from complex aerial images.

*This research was supported in part by the Defense Advanced Research Projects Agency under Contracts Nos. MDA903-86-C-0084, DACA76-85-C-0004 and and 89-F-737300.

1 Introduction

The problem of labeling objects appearing in an image is difficult because objects are recognized using not only the information present in the image signal but also knowledge about the semantics of the world. Therefore, most practical approaches to model-based vision use models that may be either specific [1, 3, 4, 5, 35] or generic [10, 24, 25, 26, 28]. They usually rely on heuristic rules and measures to generate object hypotheses and select among competing ones. These methods, although they may be effective in the context for which they were designed, are extremely difficult to extend and require the use of many parameters whose significance is not clearly understood.

In this paper, we propose a unified optimization framework for generating scene parses. We model objects using a language that specifies both photometric and geometric constraints and define an information-theoretic objective function that measures the fit of the models to the data. We then treat the problem of finding objects as one of generating the optimal description of the image in terms of this language.

We define our objective functions based upon theoretical arguments similar to those of Feldman et al. [8], Georgeff et al. [16], Rissanen [31], Leclerc [23] and Pednault [27]; we show that the required probability estimates can be computed in the context of two reasonable assumptions.

In principle, given the objective function and unlimited computing power, one could automatically generate optimal parses by simply considering all possible partitions of an image and retaining the best. Such a method would be highly impractical, both because of the size of the search space,[1] and because the models used would have to be extremely carefully defined to provide adequate discrimination. The key to a working system is thus an efficient hypothesis generator that limits the size of the search space to a reasonable subset of the a priori likely candidates. In this paper, we describe two approaches to hypothesis generation:

- **Refinemenent of Crude Hypotheses**: Following the general paradigm proposed by Kass et al. [20], rough shapes are moved to the nearest local optimum of the objective function using a simple gradient ascent procedure. This technique can be used to generate locally optimal hypotheses from rough cues, whatever their source.
- **Hierarchical hypothesis generation**: The hypothesis generator builds model primitives that are optimal with respect to components of the objective function and groups them into higher-level primitives. At the top of the hierarchy of primitives, it produces candidate model instances with high scoring characteristics. To further improve their scores, these instances can themselves be locally optimized.

We begin by introducing our objective function and our photometric and geometric models. Next, we discuss local optimization of the objective function that can be used either for operator-guided refinement of features or as a utility in an automated system. We then describe a heuristic optimization procedure that automatically discovers buildings in aerial imagery, and we evaluate its results in challenging aerial images.

[1] There are $2^{512 \times 512}$ possible sets of pixels in a 512×512 image!

2 The Objective Function

Our goal is to parse a scene in terms of objects conforming to particular models. In this section, we derive an objective function that distinguishes and ranks individual scene features that constitute a *partial description* of the scene, in contrast to techniques designed to find descriptions of the *entire scene* that achieve either the maximum a posteriori probability [15] or the shortest encoding length [23, 31].

2.1 Derivation

To discriminate among competing parses, an objective function must be able to measure the goodness of fit to feature models that include such characteristics as area photometry, edge photometry, shape, and semantic relationships. In this section, we define a basic class of models, discuss the parameters that control our objective functions, derive the theoretical forms of the objective functions themselves, and provide an interpretation of the resulting functions in terms of information theory.

2.1.1 Object Modeling

For the purposes of this work, we define a *model* to be a geometric description of an object in the world characterized by its *geometric constraints* and its *photometric properties*. We will take a *model instance* to be a specific example of a model class, e.g., an image may contain many instances of the same house model. In practice, model instances are represented as three-dimensional objects whose projections are contours in the image.

We define the *evidence* relative to a model instance in a digital image to be the collection of pixel values within the contour defined by the instance, including the border (i.e., the pixels of the contour itself).

We phrase our photometric model in terms of an ideal model plus a noise component [30, 31, 23] and use it to encode the evidence of each instance. We then use the length of this encoding as a measure of the quality of the fit between the data and the model. For an overview of the information-theoretic concepts exploited here, see Appendix B.

This division of the model language into object model plus noise is potentially task-dependent and semantic in nature. For example, if we are interested in *roofs*, we may consider the precise distribution of shingles on the roof to be irrelevant statistical noise; if we are interested in *shingles*, the position of each shingle on the roof becomes critical information. Textured object surfaces may similarly be either important in every detail or irrelevant except for their statistical character.

2.1.2 Essential Parameters of the Objective Function

When a model's geometry is completely determined beforehand, as it is for template-matching approaches [2], there is *no need* for a shape quality measure. However, because we utilize models defined by a general set of geometric constraints and arbitrarily large numbers of parameters, such a measure becomes necessary to select elegant descriptions and reject those conflicting with the chosen geometric language. To control the balance of influences, we introduce two fundamental parameters, the *scale* and the *shape coefficient.*

Scale. The scale is interpretable as the unavoidable dimensional factor that converts dimensional quantities such as area or length into dimensionless probabilities. Area units are thus scaled down by two powers of the dimensional unit, while length terms such as edges are scaled down by a single power. The scale parameter thus controls whether area signature dominates edge signature.

The scale parameter may also be understood by observing that when an image is resampled or zoomed, the area A of a patch will change, but the complexity of the patch, as reflected in its minimal encoding, should remain invariant in some range. Thus there should be some intrinsic zoom factor s that relates the area A to the area $A_0 = A/s^2$ in the zoomed image that has exactly the resolution needed to completely encode the data without redundancy.[2]

In Appendix B, we suggest yet another way of understanding the scale in terms of the minimal sampling rate needed to describe the image and the Nyquist frequency.

Shape Coefficient. An objective function with a shape quality term alone will retain all candidate model instances with the appropriate geometry even if they do not fit the image data. In contrast, an objective function with only a photometric model will make the same errors as a segmentation algorithm. The shape coefficient balances the possibly conflicting requirements of the geometry and photometry; the point where this balance lies must be determined by the context of the application.

Because these parameters are semantic in nature, we have made no effort so far to automate their selection. However, our approach to feature-hypothesis evaluation provides a clear way to justify and understand the essential role of these two parameters, regardless of the other details of a particular system.

2.1.3 The Probability of a Scene Parse

We choose to describe the problem of determining the best image interpretation as the need to maximize the probability $P = p(m_0, m_1, \ldots, m_n | e_1, \ldots, e_n)$ that, given the evidence $E = \{e_i;\ i = 1 \ldots n\}$, describing the scene in terms of a particular set of model instances $M = \{m_i;\ i = 1 \ldots n\}$ and a background m_0 is correct.[3] Each m_i is taken to be a model instance, while e_i is the measurable evidence specific to the i-th model instance, typically a set of associated pixel intensities. We emphasize that the $\{m_i\}$ are not the model definitions themselves, but rather are particular examples of a chosen generic model in the image; they include conjectured labeling information, spatial positions in the image plane, and the parameters for the ideal photometric models. Since we are interested in feature extraction, we do not explicitly represent the background and collect no evidence for it.

[2] The formulas presented later in the paper may thus be alternatively interpreted as expressing the patch encoding cost in terms of the sampling-invariant quantity A_0 instead of A itself.

[3] For example, in terms of a human analyst's perception, or in terms of ground truth.

Since it is essentially impossible to evaluate the conditional probability P in its most general form, we make two assumptions:

- **Assumption 1: Photometry is Specific.** The probability that a model instance corresponds to an actual object depends on its own photometry and on the presence of surrounding objects, but not on the particular evidence of these objects. For example, in an aerial image, whether or not a patch of pixels can be identified as a road may depend on its own photometry and on the presence or absence of neighboring houses, but not on the particular photometric quality of those houses.

- **Assumption 2: Photometry is Local.** The probability that a body of evidence is observed depends on its associated model instance but not on other model instances. This assumption may break down when one object's expected photometry is strongly modified by another object, such as when a superstructure or a separate building occludes or casts a shadow on a roof.

These assumptions are valid for isolated objects. Various situations such as occlusions, cast shadows, and objects sharing edges may render them invalid. In practice, we can often compensate for such phenomena by discounting small anomalies during the computation of the probability values. However, in more extreme cases, it may become necessary to use more sophisticated models for which the assumptions maintain their validity. For example, in the case of a tall building casting a shadow on a neighboring roof, explicitly representing the shadow allows one to consider the photometry in the remaining part of the roof as independent of the tall building, thereby restoring the validity of our assumptions.

Combining our assumptions with Bayes' rule, as shown in Appendix A, it is straightforward to express the probability of the parse as

$$P = p(m_0, m_1, \ldots, m_n | e_1, \ldots, e_n) = p(m_0, m_1, \ldots, m_n) \prod_{i=1}^{n} \frac{p(e_i|m_i)}{p(e_i)}. \tag{1}$$

This expression clearly separates the contribution of the photometry, in the evidence-dependent terms, from the abstract contribution of the geometric and semantic component in $p(m_0, m_1, \ldots, m_n)$ under the stated assumptions. We further expand this term as

$$\begin{aligned} p(m_0, m_1, \ldots, m_n) &= p(m_0|m_1, \ldots, m_n)p(m_1, \ldots, m_n) \\ &= P_0 p(m_1, \ldots, m_n), \end{aligned} \tag{2}$$

where $p(m_1, \ldots, m_n)$ is the probability that these n instances appear in the scene, and P_0 is the probability that no other does. Since we do not account explicitly for the background in this work, we take P_0 to be constant.

2.1.4 Minimal Encoding Length and Model Effectiveness

We choose to express the quality of a parse as the (base 2) logarithm[4] of Eq. (1). As discussed in Appendix B, classical information theory [17, 33] leads us to interpret the resulting score S in

[4] All logarithms in this paper are base 2 logarithms.

terms of encoding length:

$$S = \log \frac{P}{P_0} = F - G, \tag{3}$$

where we define

$$\begin{aligned} F &= \sum_{i=1}^{n} F_i = \sum_{i=1}^{n} \{-\log p(e_i) + \log p(e_i|m_i)\} & (4) \\ G &= -\log p(m_1, \ldots, m_n). & (5) \end{aligned}$$

Note that while $(-\log P)$ is negative definite, S is shifted to include a positive range when $P_0 < 1$; the sign of S itself thus has no real significance. However, we can easily see that pulling additional model hypotheses out of the background is only worthwhile if their *incremental* contribution to S is positive.

In Eq. (4), F is what we call the *encoding effectiveness* of the set of models. The $-\log p(e_i)$ terms give the number of bits needed to describe the evidence in the *absence* of the model, while the $-\log p(e_i|m_i)$ terms give the number of bits needed to describe the evidence *using the modeling language*. The use of the term *effectiveness* is thus motivated by the fact that F represents the *number of bits saved* by representing the evidence using the model; F increases as the fit improves.

G is the number of bits needed to encode the evidence-free model representation information, and quantifies the elegance of the chosen set of model instances with respect to the model language as well as their dependencies.

2.2 Photometry: Computing F

Two of the main characteristics of an object in an image are its interior photometry and its contrast with the background, which produces edges. Here we explore simple models for the area and for the edges of an object that have proven useful in analyzing aerial imagery. When working with stereo pairs of images, we also incorporate a stereoscopic model, and compute the depth parameters of an object in the scene by optimizing the corresponding stereo effectiveness.

We have seen that the effectiveness F is computed as $-\log p(e) + \log p(e|m)$ where e represents the grey-level values of the pixels that are enclosed by the contour m. For the sake of exposition, let us distinguish the evidence e_A relative to the interior of the patch and the evidence e_E relative to the boundary. Formally, we can write

$$\begin{aligned} p(e|m) &= p(e_A|m)p(e_E|m, e_A) \\ p(e) &= p(e_A)p(e_E|e_A) \quad . \end{aligned}$$

We assume that contrast with the background can be measured by using local image derivatives, while ignoring the grey levels of the boundary pixels. This contrast depends on the grey-level of background pixels that do not appear in the object descriptions, and can therefore be considered as independent of the interior object photometry. Thus we write F_i in Eq. (4) as the sum of area and edge components:

$$F_i = F_{i,A} + F_{i,E}$$

$$\begin{aligned} F_{i,A} &= -\log p(e_A) + \log p(e_A|m) \\ F_{i,E} &= -\log p(e_E) + \log p(e_E|m) \quad . \end{aligned}$$

This prescription must be modified when dealing with objects that share edges, since the contrast of the shared edges is completely determined by the photometry of the regions on both sides of the edge. In this case, the shared boundaries do not contribute to the edge effectiveness term.

When additional images are available and m is a three-dimensional model, additional evidence e_S can be gathered using the projection of m onto each image. We write

$$\begin{aligned} p(e, e_S|m) &= p(e|m)p(e_S|m, e) \\ p(e, e_S) &= p(e)p(e_S|e) \quad . \end{aligned}$$

In the case of a pair of stereo images, e is the evidence measured in the first image and e_S the evidence in the second image relative to the instance's projection into that image. For a stereo pair, we therefore add to the effectiveness a *stereo effectiveness* term,

$$F_S = -\log p(e_S|e) + \log p(e_S|m, e) \quad . \tag{6}$$

In the following subsections, we present the modeling requirements of our applications and show how effectiveness measures that satisfy these requirements can be designed. While a wide range of statistical models could be used to compute the encoding costs, those presented here have proven both simple and effective in practice.

2.2.1 Area Model for Homogeneous Regions

We model the interior intensities of an image region by a smooth intensity surface with a Gaussian distribution of deviations from the surface. Since objects in real images typically have anomalies that do not lie on the smooth surface, we encode such anomalous pixels as outliers. As we shall see later, this can critically enhance the discriminatory power of the area-encoding effectiveness.

An ideal area measure should find the best compromise among the following goals:

- Goodness of fit to the surface intensity model.
- Small number of anomalies.
- Large area.

The goodness-of-fit criterion guarantees that a region larger than the actual object and with a poorer fit to the surface intensity model also has a lower effectiveness. Conversely, the large area requirement ensures that a subregion of an object has a lower effectiveness than the object itself.

In our application of the approach to extracting buildings from aerial imagery, we take the intensity surface to be a plane. The choice of the planar area model is based on simplicity and an experimental observation. Theoretically, one would expect building roofs to be planes that are approximately Lambertian reflectors, yielding constant intensity patches in the image. Experimentally, the combination of photometry, film processing characteristics, and digitization artifacts that characterize the vast majority of the digital aerial images at our disposal do not produce constant intensity patches, but patches that are much better described by a plane in intensity space. For more complex objects, the plane could be replaced by any other parametric surface without changing the encoding cost computations described below.

(a) (b) (c)

Figure 1: (a) Complex building. (b) Roof outline with anomalous pixels overlaid in black. (c) Histogram of deviations from planar fit.

In Figure 1, we show (a) an image of a complex building, (b) the outline of the roof, and (c) the histogram of deviations from the planar fit to the intensity surface. The pixels whose grey levels are not between the two vertical bars are considered as anomalous and overlaid in black in figure 1(b).

In an 8-bit image, it would take $8A$ bits to encode the pixel values if we did not take advantage of dependencies among pixels. An ideal description would take $k_A A$ bits to encode the same information using our region model, where

$$k_A A = n(\log \sigma + c) + 8\overline{n} + E(n, \overline{n}) \quad . \tag{7}$$

As discussed in Appendix B, $n(\log \sigma + c)$ is the cost of Huffmann-encoding [17] the pixels in a Gaussian peak, $8\overline{n}$ is the cost of encoding the outliers, and

$$E(n, \overline{n}) = -\left[n \log \frac{n}{A} + \overline{n} \log \frac{\overline{n}}{A} \right] \tag{8}$$

is the entropy, i.e., the cost of specifying whether a pixel is or is not anomalous. σ is the measured variance of the n pixels belonging to the Gaussian peak, $\overline{n} = A - n$, and $c = (1/2)\log(2\pi e)$. Note that in the computation of the encoding cost, we have not included the cost of encoding the six internal parameters of the model: three for the plane, two for the Gaussian, and one for the probability n/A that a pixel lies in the main peak. It can be shown [30, 34] that these costs are approximately equal to $(1/2)\log A$ bits per internal parameter of the statistical distribution, and are therefore negligibly small compared to $k_A A$.

We weight all areas and lengths using the scale parameter s (see section 2.1.2) so that the area-encoding effectiveness becomes

$$F_{i,A} \quad = \quad \text{bits(without model)} - \text{bits(with model)}$$

$$
\begin{aligned}
&= (8 - k_A)\frac{A}{s^2} \\
&= \frac{1}{s^2}\left((8 - c - \log\sigma)n - E(n, \overline{n})\right) \quad . \qquad (9)
\end{aligned}
$$

From this expression, it is easy to see that F_A satisfies our requirements since it increases when A grows, when the ratio of n to $\overline{n}$ increases, and also when σ decreases.

Figure 2: Area and edge effectiveness of a square patch as a function of candidate radius, with (solid) and without (dotted) anomaly discounting. The white overlays in the top left image represent the square patches of radius 20 and 45.

Effect of Anomaly Discounting. In the central column of Figure 2, we plot the area-encoding effectiveness F_A as a function of the radius of a square patch centered at the center of the images shown in the left column: a good but noisy synthetic image of a square, the same image with gross area anomalies, and an image of a similar but distorted square. When we compare the results obtained *after discounting anomalies* (solid lines) with those results found without anomaly discounting (dotted lines), we see that anomaly discounting *must* be included to make the objective function reliably select the same shape a human observer perceives. This is potentially a critical factor in the practical application of this approach because, as we see in Figure 1, real images nearly always have significant anomalous components.

2.2.2 Edge Model

We require that at least portions of an object's boundary have a measurable contrast with the background and use the image gradient as an indicator of contrast. However, we have observed that the absolute magnitudes of the gradients are not as relevant to our analysis as are their relative magnitudes: boundaries can be adequately characterized as local maxima of the gradient [7, 19, 32] independent of the value of the gradient. An ideal edge measure should concern itself with whether an edge exists, and give equal weight to equally good edge candidates regardless of their gradient strengths.

We propose an edge model that quantifies the quality of a contour by the proportion of pixels that are maxima of the gradient in the direction normal to the contour. In the absence of a statistical model for the distribution of edge pixels, it would take 1 bit per pixel to encode whether or not a contour pixel passes the maximality test. Given a contour of length L in which l pixels pass the test and $\bar{l} = L - l$ do not, we can use the probability $p = l/L$ that a pixel passes the test to Huffmann-code [17] this information using

$$k_E = -\left[\frac{l}{L}\log\frac{l}{L} + \frac{\bar{l}}{L}\log\frac{\bar{l}}{L}\right] \tag{10}$$

bits per boundary pixel. We then weight all lengths by the scale factor s and estimate the edge-encoding effectiveness to be

$$\begin{aligned} F_{i,E} &= \text{bits(without model)} - \text{bits(with model)} \\ &= (1 - k_E)\frac{L}{s} \quad . \end{aligned} \tag{11}$$

As in the case of the area term, we have neglected the $(1/2)\log(L/s)$ bits required to encode p, the only parameter of the statistical model [30, 34].

This criterion satisfies our requirement because it increases with the ratio of pixels that pass the maximality test to those who do not without depending on the absolute gradient magnitudes.

However, this criterion is essentially Boolean and does not support computations of local derivatives of F_E with respect to small deformations of the contour. When such computations are required, as for the gradient ascent optimization of section 3, we replace F_E by a measure F_{grad} that is a differentiable function of the gradient magnitudes and is such that F_E is maximized when F_{grad} is maximized (see section 3.1).

(a) (b) (c)

Figure 3: (a) Roof candidate in left image of a stereo pair. (b) F_S as a function of the assumed disparity between left and right image. (c) The projection of the contour in the right image using the best disparity value.

2.2.3 Stereography

The simplest stereo model assumes that corresponding pixels have the same grey levels in both images. In practice, to compute the stereo effectiveness of Eq. (6), we determine the number of bits required to encode the projected patch in the second image, while knowing its photometry in the first. We compute the deviations of the intensities from their predicted values and encode them using the same Gaussian model with anomalies that we used for the area term. The anomaly discounting is required because of the possibility of occlusions. We also take into account the edge quality of the contour in the second image and its edge-encoding effectiveness.

The stereographic effectiveness term F_S is therefore the sum of an edge and an area term,

$$\begin{aligned} F_S &= F_{AS} + F_{ES} \\ F_{AS} &= (8 - k_{A_2})\frac{A_2}{s^2} \\ F_{ES} &= (1 - k_{E_2})\frac{L_2}{s} \ , \end{aligned} \tag{12}$$

where A_2 is the area of the projected patch in the second image, L_2 is its boundary length, and k_{A_2} and k_{E_2} are the corresponding model encoding costs.

We can use the effectiveness measure of Eq. (12) to optimize the elevation parameters of a two-dimensional delineation found in the first image. The search space is extremely constrained since the projected shape is known and the only degree of freedom is epipolar motion in the second image.

Let us consider the stereo pair of images in Figure 3(a,c). Assuming that the roof is horizontal, we plot in Figure 3(b) the value of F_S as a function of the assumed disparity between the candidate outline in the left image (a) and the projected outline in the right image. We note that F_S has a sharp peak for the correct match outlined in (c).

2.2.4 Stability of the Effectiveness Measures

Note that that F_A and F_S increase with the area of the model instance, while F_E grows with its length. Large image patches can potentially have large effectivenesses even though their fit to the photometric model is relatively poor. For example, in Figure 2, if we allow the radius of the square hypothesis to grow indefinitely, its area effectiveness eventually becomes larger than that of the actual object. This problem stems from the fact that we are dealing with a partial description of the scene, as opposed to a complete one.[5]

To resolve this issue, we require that the effectiveness be not only high but also *stable* with respect to small deformations of the contour. In practice, our hypothesis-generation algorithms use a local optimization procedure to enforce maximality of the objective function and reject instances that do not pass a stability test.

[5] In the example of Figure 2, a better parse of the scene would be in terms of *two* model hypotheses, one square and one square-shaped ring covering the rest of the image, rather than one square plus random background.

2.3 Geometry: Computing G

The geometric cost G defined by Eq. (5) is a *measure of quality* of a set of object hypotheses. G should be considered as a measure of appropriateness of the shape language: our rectilinear polygons can be used to efficiently describe buildings in modern cities (and therefore yield low values of G), but would be completely inadequate to describe medieval architecture. If our goal was to find medieval cathedrals, we would have to design a more complex but better adapted language that might very well be inappropriate for modern buildings.

One way to handle the potentially difficult problem of dependencies among objects is to require that there be no conflicts within a particular set of hypotheses; formally we write

$$\begin{aligned} p(m_i|m_j) &= \begin{cases} p(m_i) & \text{if } m_i \cap m_j = \emptyset \text{ or } m_i \subseteq m_j, \\ 0 & \text{otherwise,} \end{cases} \\ \Rightarrow p(m_1 \ldots m_n) &= \begin{cases} \prod_i p(m_i) & \text{if no conflict} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

It follows that, in the absence of conflict, G can be expressed as

$$G = -\log p(m_1 \ldots m_n) = \gamma \sum_{i=1}^{n} G_i, \tag{13}$$

where $G_i \propto -\log p(m_i)$ is a model quality measure that increases as the shape degrades, and γ is the arbitrary *shape coefficient.*

As noted in Section 2.1.4, if we write the overall score in the form

$$S = \sum_{i=1}^{n} (F_i - \gamma G_i),$$

we deduce that we should accept additional model instances only if $(F_i - \gamma G_i) > 0$, since these are the only ones that improve the likelihood of the full scene parse.

The simplest effective model for G_i is the sum of the cost of chain-encoding the boundary of the object's area plus a constant cost for introducing a new object; this gives a geometric cost

$$G_i = \frac{L_i}{s} + c. \tag{14}$$

In Figure 4(a), we show how the length term of Eq. (14), which gives preference to compact objects, influences the parse when a split square is interpreted alternately as a single compact square or two adjacent rectangles. The bottom graph takes three images, with noise variance 40, 20, and 10, and plots the ratios (two-rectangle score)/(square score) as a function of scale for fixed $\gamma = 1$. Note that increasing the scale in this example amounts to looking at a subsampled image in which fine details are no longer visible. The interesting value of the scale is that for which the scores are *equal*, i.e., the ratio is one. Thus we plot in the upper graphs the locus of points where the ratio is

(a) (b)

Figure 4: (a) Ratio of single-square to double-rectangle score as a function of noise variance (40, 20, 10). (b) Similar plot comparing the score of the square interpretation to the "U" interpretation.

unity as a function of γ as well as scale. In Figure 4(b), we carry out a similar plot for an image of a square with a missing portion that makes it "U"-shaped. We see that the ratio ("U" score)/(square score) behaves so that the square interpretation is preferred at a large scale in the best image, and at a much lower scale in the noisier images.

We observe a similar phenomenon in the real image of Figure 5. The automated system of section 4 finds two conflicting interpretations for the building: one in terms of a single polygon enclosing both wings, as in Figure 5(b), the other in terms of two polygons, one for each wing, as in Figure 5(c). At low scale, the latter will be preferred because of its better fit to the photometric data, while at high scale, the former will dominate because of its lower geometric cost.

In the application to the analysis of aerial imagery presented in section 4, we take advantage of this property of the objective function to control its behavior by fixing the shape coefficient and using the scale as a control parameter.

3 Local Optimization

The simplest way to generate stable local optima of the objective function is direct optimization, which we implement by using a gradient procedure to deform initially supplied contours. Among the potential applications of this paradigm are

- Testing the nature and effectiveness of particular models incorporated into the objective function.

(a) (b) (c)

Figure 5: (a) A complex building. (b) Interpretation in terms of a single polygon. (c) Interpretation in terms of two polygons.

- Improving the characteristics of automatically generated feature hypotheses.
- Relieving human operators of the burden of metrically accurate feature delineation by automatically optimizing a rough sketch.

3.1 The Approach

We address this problem by describing object contours as geometrically constrained curves moving in an effective potential whose iterative solution converges to local maxima of the objective function; the resultant outline will then conform to the nearest object in the image that corresponds to the model represented by the objective function. Such curves were originated by Terzopoulos, Kass, and Witkin as "snakes" [20, 36]. In their approach, boundaries are described as polygonal curves with a score that includes geometrical constraints and a measure of edge strength. "Snakes" do not take into account any photometric evidence outside the edge; they yield good results only if the initial position of the curve is close enough to the boundary of the object to be influenced by its edges. Since we also use *interior area information*, our curves can easily grow or shrink if the initial position is very inaccurate. By integrating more information and incorporating anomaly discounting, we also make our algorithm more stable and less sensitive to photometric anomalies.

The Potential. In theory, the potential used by the optimization procedure should be the objective function itself. In practice, however, the objective function used for scoring is inappropriate for snake-like optimization procedures because neither the edge measure nor the geometry measure are smooth enough to form a potential that acts over a reasonable distance.

As mentioned in section 2.2.2, we replace the edge effectiveness by a differentiable measure F_{grad} for the purpose of optimization. We define F_{grad} to be the sum along the boundary of the logarithm of the gradient. The resulting edge term is smoother and therefore better suited for optimization. Furthermore, it can be shown [14] that points on curves that maximize F_{grad} are local maxima of the gradient in the direction normal to the curve. They therefore satisfy the edge criterion of section 2.2.2, and the corresponding curve has a high edge effectiveness. We define the sum of the area effectiveness and this gradient term to be the *effective potential* to be used in the optimization procedure.

The geometric constraints may be highly nonconvex, e.g., when we are dealing with rectilinear polygons having an arbitrary number of vertices. Instead of adding a geometric term to the effective potential, we therefore enforce the geometric constraint on the optimization procedure by first deforming the curve in the direction of the gradient of the potential, then fitting the best shape matching the geometric model to the deformed curve.

The Optimization Procedure. Our optimization procedure includes the following steps:

1. Compute derivatives of the effective potential.

2. Increment the curve in the direction of the derivatives.

3. Smooth the curve and fit the geometric model.

4. Update the curve data structure.

These steps, and their Connection Machine[6] implementation, are described in detail in Appendix D.

Examples.

Smooth Contours in Two Dimensions. In the simplest case, we only smooth the curve during each iteration without imposing a geometric model. The curve then tends to shrink (or expand) to match the contours of an object and yields a smooth outline. Because of the smoothing, deformations are propagated along the curve at every iteration, making this procedure considerably faster and more stable than ordinary gradient ascent. For example, going from the initial estimates of the closed curve shown in Figure 6(a) to the final result shown in Figure 6(d) took only 10 iterations. Figures 6(b) and 6(c) show the position of the curve after 3 and 7 iterations respectively. In the aerial image of Figure 7(a), the four initial contours shown in Figure 7(b) yield, after optimization, the final outlines of Figure 7(c). Note that the corners of the house are slightly rounded because of the presence of the smoothing term.

Rectilinear Contours in Two and Three Dimensions. In the application domain of buildings, we fit a rectilinear polygon to the deformed, smoothed curve at every iteration. Given the two initial contours shown in Figure 8(a), the algorithm generates the outlines shown in Figure 8(b). Using a second image, the elevation of the contours can be automatically determined by maximizing the stereo effectiveness F_S defined by Eq. (12). For all hypothesized elevations within a given range, the projection of the outlines in the second image and the corresponding value of F_S are computed. The elevation for which F_S is maximal is the height with the strongest supporting

[6] Trademark, Thinking Machines Corporation

(a) (b)

(c) (d)

Figure 6: (a) A synthetic image of a circle and the initial position of the curve. (b) The position of the curve after 3 iterations and (c) after 7 iterations. (d) The final outline.

evidence. In Figure 8(c), we show the computed projections of the contours in a second image of the same scene.

3.2 Strengths and Weaknesses

The strengths and weaknesses of our direct optimization approach can be summarized as follows.

- **Strengths:**
  - We combine both edge and area information with geometric constraints. The initial contours can therefore be some distance away from the object outline and shrink or expand to match them.
  - We have implemented the algorithm on the Connection Machine,[7] which allows the optimization to be done rapidly enough to permit interactive applications of the method.
- **Weaknesses:** Our optimization method utilizes gradient descent, making it difficult to avoid local maxima of the objective function that may fail to correspond to real objects. One possible

[7] Trademark, Thinking Machines Corporation.

(a) (b) (c)

Figure 7: (a) An aerial image of a suburban scene. (b) Interactively entered initial contours. (c) Final outlines after optimization.

remedy for this problem is to supplement the optimization procedure with a randomization procedure that helps the system escape undesirable local maxima. In the automated system described in the next section, we have incorporated such a step (see Appendix E.4).

We now turn from direct optimization of the objective function to an heuristic procedure that is adapted to automated optimization of the objective function.

4 Automated Building Extraction

The direct optimization procedure of the previous section required an initial cue that was relatively close to the actual object. In this section, we describe a procedure for automatically generating such cues by using a building model consisting of three-dimensional rectilinear outlines conforming to our area, edge, and stereo models. We first argue the need for heuristic rules and then briefly outline the structure of the hypothesis generator. For full details of the hypothesis generation algorithm, see Appendix E.

4.1 The Approach

Heuristic Rules. It is impossible to consider exhaustively all possible partitions of an image. Therefore, in our optimization framework, a working system must be able to generate a sufficiently small set of hypotheses so that they can be evaluated in reasonable time. Furthermore, even if we could consider all possible partitions, it would be difficult to endow the objective function with enough semantic complexity to discriminate among all the competing hypotheses.

Our hypothesis generator uses geometric constraints and heuristic rules to reduce the search space and generate only a relatively small number of candidate model instances that are local

(a) (b) (c)

Figure 8: (a) Initial contours in the left image of stereo pair. (b) Final polygonal outlines after optimization. (c) Matching outlines in the right image.

optima of the objective function. The system rejects model instances that do not pass a stability test and selects the subset of compatible instances that maximizes the overall objective function. Since, by construction, all hypotheses satisfy the geometric constraints, the simple geometric term that appears in our objective function suffices for discrimination purposes.

The Parsing Procedure. The parsing algorithm can be understood as an optimization procedure in which each of the parsing steps listed below is a filtering process that both enforces some model constraint and limits the size of the search space.

1. **Build Edges.** The system presented in this paper first computes Canny edges [7] and links them. It then extracts edge segments with the appropriate geometry from linked edge pixels and optimizes their location [14]. The resulting edges are scored using the edge-quality term of the objective function, and only those with a high edge effectiveness are retained.

   In predecessors of this work [12], instead of linked Canny edges, we used the boundaries of segmentation regions [21, 22, 23] as sources of edge segments. When good segmentations are available, this tends to be more effective because only edges likely to correspond to object boundaries are considered. Furthermore, edges belonging to the boundary of the same region automatically share the intensity characteristics of the region and can be naturally clustered.

   As shown in Figure 9, for both edge and region operators, no single parameter setting can be expected to handle all target objects in one image, much less in multiple images. To solve this problem, we compute *hierarchies* of edge maps (or region segmentations) by performing the computation with several sets of control parameters. When using the Canny edge operator, we compute a series of maps with monotonically decreasing edge strength thresholds. When using segmentation boundaries, we compute a series of segmentations ranging from underseg-

(a) (b) (c)

(d) (e) (f)

Figure 9: (a) An aerial suburban scene. (b) A Laws segmentation with undersegmented roofs. (c) Oversegmentation resulting from a different parameter choice. (d) (e) (f) Canny edge images computed progressively lower edge-strength thresholds. These hierarchies illustrate typical problems of low-level operators: edge detectors will find too few or confusingly many edges, and region segmenters will either undersegment some semantically meaningful objects or break them into several pieces because they fail to take higher-level geometric and semantic knowledge into account in their analysis.

mentation to oversegmentation. We then merge the extracted edges across the hierarchies, retaining only those with the highest edge effectiveness.

Hierarchies greatly improve the performance and image independence of the system because they allow edges that could not be found with a single parameter setting to be linked in the same structure.

The edge extraction process attempts to generate edges that both have the appropriate geometry and maximize the number of pixels that satisfy our edge criterion, consequently maximizing the edge effectiveness [Eq. (11)] of the model instances that will be generated in step 4. In Figure 10 we show a stereoscopic pair of images with a complex building and in Figure 11 (a) the edges found by the system in the left image.

2. **Construct Arcs.** Pairs of edges that are either parallel, perpendicular, or collinear are associated into what we call an *arc*. For each of the arcs, the system computes a rectilinear linking path between the edges, an area in the image that is enclosed by the edges, and the corresponding area effectiveness.

   Since edges may line up accidentally, edge information alone is not sufficient to form a reli-

Figure 10: A stereo pair of images containing a large complex building.

(a) (b)

Figure 11: Steps in the parsing procedure. (a) Straight edges extracted from the original image data. (b) The cluster of edges corresponding to the building.

able opinion about the validity of the association of two edges into an arc. Therefore, only structures whose enclosed area has a high effectiveness are retained, thereby preventing the association of edges that do not belong to the same object and reducing the search space.

In previous implementations that used segmentation regions [12], the system enforced this interior area constraint by building geometric structures only between edges belonging to the same region.

3. **Construct Cycles.** We use the arcs to first cluster related edges as in Figure 11 (b) and then generate circular lists of edges (cycles) that enclose an area in the image. Only arcs whose enclosed areas have similar photometry are grouped, thereby contraining the search space and improving the likelihood that cycles enclose areas with good photometry, such as those shown in Figure 12.

4. **Build Enclosures.** The system now generates *enclosures*, the analogs of the manually

(a) (b)

(c) (d)

Figure 12: Steps in the parsing procedure. (a) A cycle of edges suggesting the presence of a good building object. (b) The enclosure resulting from completing the missing links in the cycle. (c) A cycle of edges that has no consistent semantic interpretation. (d) The resulting enclosure.

sketched hypotheses of section 3, by combining the associated edges and arc completions of each cycle into a single contour. These enclosures are then optimized using a variant of the "snake" algorithm [20, 14], which relies on edge information alone; for images with difficult photometry, we add the interior area information and apply the more sophisticated technique of section 3.

This optimization serves a triple purpose:

- **Compensate for Poor Photometry.** Optimization moves the contour to a local maximum of the objective function. In Figure 13, we show how the optimization of a deficient hypothesis can produce a much-improved building candidate.
- **Stability test.** After optimization we can perform a simple stability test and reject those instances that do not pass. In practice, we require a minimal edge quality and contrast with the background. This test is important because in the case when neighboring regions have low contrast, the system can hallucinate enclosures with good area scores that span both regions and that must be discriminated against, as shown in Figure 14.
- **Collapse Multiple Hypotheses.** The cycle builder typically generates massively re-

(a) (b) (c)

Figure 13: (a) Bare edges in an image containing a building. (b) A closed cycle before optimization, including only a portion of the building. (c) The cycle after optimization has expanded to fill-in the building areas matching the model.

dundant hypotheses with overlapping common portions. Cycles with sufficient overlap will be optimized to identical enclosures, thus serving to reduce the redundancy of the hypotheses to be considered.

When stereo is being utilized, the system assumes that the optimized contours define a plane in three-dimensional space and computes its elevation parameters by optimizing the stereo effectiveness measure.

Finally, the system uses Eq. (14) to evaluate the geometric cost of each enclosure, computes the area, edge, and stereo effectivenesses, and, given a particular choice of the scale s, a final score for each enclosure.

5. **Select Enclosures.** As suggested by Figure 12, the system usually builds a set of overlapping, and therefore conflicting, enclosures. Among the set of enclosures that have been retained, choose the subset of nonoverlapping ones that maximizes the objective function of Eq. (3). Since we impose the geometric constraints on the hypothesized contours, the system may form dubious hypotheses that conform to the desired geometry but have poor photometric characteristics. As suggested in section 2.3, we use the scale parameter s to control the rejection rate for such hypotheses.

This hypothesis-generation procedure can be understood as a heuristic optimization procedure. In a complex image, the search space is much too large to allow for direct optimization of the objective function. We have therefore defined a procedure that hierarchically makes use of the

(a) (b)

Figure 14: Two enclosures generated by the system. The larger one, (a), was built using spurious edges that accidentally lined up with the true building, (b). At small scale, hypothesis (a) dominates because, although it fits the photometric model less well, it is much larger.

components of the objective function at every step and refers directly to the image data to validate and improve the structures it builds.

Experimental Results. We now show the results of running the automated building finder on a series of complex images with widely varying photometry. The hypothesis generator produces several hundred candidate buildings in each image. The objective function ranks theses hypotheses according to their score. The scale parameter is the only parameter we have varied from image to image. Since the scale has a semantic meaning, it is inevitable that a semantic decision to select its value will be required at this stage to achieve the performance goals set by the human operator. We have made no attempt to encode the knowledge needed to automate this selection.

The Effect of Scale. The scale parameter s tunes the scale not of the *physical size* of the objects, but the scale of their *quality*. Objects with close fits to the strict model are selected first as we ramp the scale down from a high value. We illustrate this phenomenon in Figures 15 and 16, in which we compare the results of the selection procedure in four different images, once with scale 7 and again with scale 8.

At scale 7, all the buildings are picked out, but some candidates with marginal characteristics such as yards and parking lots are retained. At scale 8, the rejection ratio of spurious candidates is improved, but now some legitimate buildings are also lost.

In our experience, the scale factor has proved to be an effective control parameter. However, it may fail to perform its task when two objects with low relative contrast are merged into one single instance that will have a high area effectiveness because of its size. As a consequence, this instance may have a score that is larger than the score of either of the smaller instances corresponding to each of the two objects — only the combined score of the two instances is greater than the score of the erroneous one. If, for any reason, the system fails to build one of the two small instances, the wrong ranking results.

(a) (b)

(c) (d)

Figure 15: Subset of enclosures that maximize the objective function at scale 7.

For both failure modes described here, stereo information can be of great assistance in rejecting spurious candidates.

Stereoscopic Buildings. When stereoscopic or multiple imagery such as Figure 10 is available, the ambiguities inherent in the identification of rectilinear, building-like objects in monoscopic imagery are largely resolved.

In Figure 17 we show the outline of the three highest-scoring building candidates found by the system and their relative scores. Note that the incomplete building shown in Figure 17 (a) has very uniform photometry, while the perceptually correct outline shown in Figure 17(c) includes large shadows and darker pixels that degrade its area effectiveness. As a result, when we use only the edge and area terms of the objective function, the scores of the two outlines are extremely close. Not surprisingly, the Laws segmenter [21, 22] produces the segmentation shown in Figure 18(a), in which a region very similar to the erroneous outline has been extracted.

However, when we project the contours found in the left image into the right image and take

(a) (b)

(c) (d)

Figure 16: Subset of enclosures that maximize the objective function at scale 8.

the corresponding stereo effectiveness into account, the score of the “correct” parse becomes considerably larger than the score of the erroneous one. Thus, in a case like this one, the additional stereoscopic information helps the system overcome ambiguities that arise in complex scenes. Furthermore, upon optimizing the stereo effectiveness, we know the three-dimensional position of each rectilinear contour; we can thus produce three-dimensional objects having the observed two-dimensional upper surface. For example, given the elevation of the top ranking model instance of 17(c), we can predict the location of the walls and project them into the image, as shown in Figure 18(b).

Improving the Models. Our simple models may be insufficient to disambiguate complex situations. In this paragraph, we suggest possible improvements and show their influence on the rankings produced by the objective function.

One simple extension of our model is the introduction of holes in the enclosures whose penalty is given solely by their boundary encoding cost. We have implemented a procedure that extracts

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 17: (a) (b) (c) The three highest-scoring hypotheses generated by the system when parsing the left image of Fig. 10. (d) Ratio of the scores at scale 8 of (b) and (c) to the score of (a) when using stereo information (triangles) or not using it (squares). Without stereo (a) and (c) have similar scores, while (c) dominates with stereo.
(e) (f) (g) (h) The same three hypotheses with holes and the ratio of their scores including stereo. (g) dominates even more clearly and (f) is intermediate between (e) and (g).

such holes and retains them only if the score is improved. When using both this more sophisticated model and the stereographic information, the ranking of the three model instances plotted in Figure 17 (h) is now closer to that of a human.

In Figure 19 we show a stereo pair with a complex multitiered building and in Figure 20 (a – c), the three top-ranking parses automatically produced by the hypothesis generation system when using stereo information. Using our simple model with no holes, at scale 15, (a) dominates, with (b) and (c) having almost identical scores. Introducing the hole model favors (c) over (b) but leaves (a) as the top-ranking parse. In order to illustrate the behavior of the objective function, we introduce a new model that includes shadow prediction and simulate in Figure 20 (d) a parse where the superstructures on the roof and their projected shadows are added. In this more sophisticated model, there is no encoding penalty associated with the presence of the shadows, because their location is completely predicted by the geometry of the superstructures. As a result the parse in which the shadows are explicitly modeled has the highest rank of all, thereby illustrating the reliability of the objective functions ranking when sufficiently sophisticated models are used.

(a) (b)

Figure 18: (a) A Laws segmentation with building region highlighted. (b) The top ranking candidate model instance and and projected walls.

Figure 19: A stereo pair of images containing a multitiered building. Note the two small superstructures on the main roof and their projected shadows.

4.2 Strengths and Weaknesses

We summarize the performance of our automated system as follows.

- **Strengths:**
  - The system simultaneously uses area, edge, stereo, and geometric information. As a result, using relatively simple models, it successfully discovers a high proportion of building-like objects in aerial images with difficult photometry; such images are likely to cause standard image-partition processes to miss many or most object instances.
  - The output of the system is controlled by a small number of parameters with clear theoretical meaning. The heuristics used by the hypothesis generator also require parameters. In theory, these parameters could be learned by the system in order to maximize the score of the parses it produces. In practice, the parameters have evolved during the development of the system to substantially image-independent values.

(a) (b)

(c) (d) (e)

Figure 20: (a) (b) (c) The three top-ranking parses generated by the system when parsing the left image of Fig. 19. In (a) one of the small superstructures is merged with the wrong rooftop while in (c) the main rooftop is broken into two components because of the presence of the large shadow. In (d) we show a simulated parse in which the two superstructures are added and shadows explicitely represented. (e) shows a plot of the scores of the three parses. Using the simple models (squares), the perceptually correct parse (b) and (c) are tied; using the models that include holes (circles), (b) dominates (c) but not (a). However (d) dominates all of them.

- **Weaknesses:**
  - The hypothesis-generator can build candidate instances only for the parts of the buildings that are fully visible with the possible exception of relatively small anomalous areas: the system does not understand the three dimensional implications of occlusions or lighting effects. It also tends, in the absence of stereographic information, to confuse legitimate buildings with other rectilinear-shaped objects such as parking lots or yards.
  - The system performs a global search over the whole image. In scenes with a very high edge density, the combinatorics of the search can overwhelm the current heuristics: the system will not examine as many model candidates as it should and may return a solution that is suboptimal. In such situations, it appears necessary to use more domain knowledge to constrain the search.

To cure the weaknesses of the approach, it appears necessary to introduce more sophisticated models that can explicitly deal with semantic constraints as well as three-dimensional information, e.g., occlusions and shadows.

Three-dimensional information in the current system is limited to the description of single, planar, roof segments and their elevation with respect to the background. This limitation can be removed by utilizing the full, three-dimensional structures available within the context of the SRI Cartographic Modeling Environment [18]. The building models can be generalized to include walls, courtyards, and gabled and peaked roof portions. As shown by the example of Figure 20, such models in conjunction with our objective function approach would help to produce more reliable parses in complex situations. Furthermore, we have no explicit examples of model languages whose geometric term G embodies structural complexity in a satisfying way; our examples of G amount to simplicity requirements alone and should be extended to more complex descriptive languages.

To take the semantics of urban scenes into account, we would have to include such factors as the knowledge that buildings connect via driveways to roads, buildings are adjacent to parking lots and courtyards, and large buildings have ventilation systems and elevator shafts on their roofs. The objective function would be modified to take these dependencies into account by including terms in the geometric component G of Eq. (3) that favor likely configurations such as buildings near roads and penalize unlikely ones such as houses with no connected driveway. One implementation of this technique would be to represent a scene by a number of rigid components, such as a road or a building instance, held together by "springs," as proposed by Fischler and Elschlager [9]. The springs joining the rigid pieces would serve both to constrain their relative movements and to measure the "cost" of the description by how much they are "stretched."

Although these are clearly difficult problems, we feel that they can be straightforwardly addressed within the framework we have established here.

5 Conclusions

In this work, we have formulated the feature-extraction problem as one of finding the optimal description of the scene in terms of a given language and a probabilistic objective function. This framework has allowed us to perform the following tasks successfully:

- **Generic shape extraction**: For many important tasks, the exact shapes of objects of interest are not known. Our models describe common cartographic objects that obey specific geometric constraints but can be arbitrarily complex. The objective function balances the goodness of fit of model instances to the image data against their geometric quality. The system can therefore pick the best object hypotheses without using rigid geometric constraints or templates.

- **Integration of multiple data sources**: In general, objects are not characterized solely by their edge or area signatures. As a result, data-driven edge and region segmentation processes often fail to extract objects as such. We use geometry as well as the photometric characteristics of both the enclosed areas and the edges to generate and evaluate shape hypotheses; we thus make effective use of the available information in a single image, or in several images when using stereoscopic data.

- **Efficient hypothesis generation**: We have implemented algorithms that enable our system to generate candidate model instances that closely match the target objects while avoiding combinatorial explosion by using components of the model constraints to prune the search space. These algorithms make crucial use of adaptive image-based search and optimization techniques that recover expected but missing model components and compensate for photometric anomalies. Thus, using its geometric knowledge, the system can remedy some of the unavoidable failures of low-level operators.

- **Ranking of competing hypotheses**: Our objective function provides an efficient way to deal with competing hypotheses. We do not have to constrain our system to yield unique answers in ambiguous situations. We can allow the system to explore the space of possible hypotheses, thereby increasing the probability that it will generate the perceptually correct ones. As a result, our system finds a large proportion of the objects of interest, even in very difficult images.

We have successfully implemented our approach in the domain of delineating rectilinear cultural structures in aerial images. To make further progress within our framework, more research is needed in the areas of modeling and control strategies. The models we have been using so far must be augmented to support geometrically complex model languages, as well as to include explicit knowledge about shadows and occlusions. Similarly, the hypothesis-generation procedure relies on simple heuristics to reduce its search space; these heuristics may prove insufficient in larger images with more numerous target objects. Increasingly complex control strategies may then be needed in such images if the number of candidates considered by the system is to correspond with the practical limits of computation.

Furthermore, the overall strategies for generating hypotheses proposed in this work are embodied in a sequence of procedures that contain implicit knowledge about the task domain. These procedures invoke the various parsing rules in turn. In order to apply the approach to other domains, it would be necessary to represent the parsing rules and control strategy in a more general form. Ideally, we should develop a domain-independent control structure driven by domain-dependent knowledge bases that could be modified and improved by domain experts rather than by software developers.

Designing sophisticated models and efficient control strategies are both extremely challenging tasks. However, our framework provides conceptual and practical tools that can be used to address these problems in a manner that is both theoretically sound and computationally feasible.

Appendices

A Derivation of the Objective Function

In this appendix, we motivate and prove Eq. (1),

$$P = p(m_0, m_1, \ldots, m_n | e_1, \ldots, e_n)$$

$$= \quad p(m_0, m_1, \ldots, m_n) \prod_{i=1}^{n} \frac{p(e_i|m_i)}{p(e_i)} \ ,$$

where P is the probability that, given the evidence $\{e_i;\ i = 1 \ldots n\}$, parsing the scene in terms of a particular set of model instances $\{m_i;\ i = 1 \ldots n\}$ and a backround m_0 is correct.

Expressing the Assumptions Mathematically. We begin by expressing our two basic assumptions in mathematical terms: Let J, K denote sets of indices referring to model instances and their corresponding bodies of evidence, and i be specific values of the index.

- **Assumption 1.** The probability that a model instance corresponds to an actual object depends on its own evidence and on the presence of surrounding model instances, but not on the particular photometry of those model instances. We express this assumption as follows:

$$\begin{aligned} P(m_i|e_J m_K) &= P(m_i|e_i, m_K) \quad \text{if } i \in J \\ &= P(m_i|m_K) \qquad \text{if } i \notin J \ . \end{aligned} \tag{A.1}$$

$$\tag{A.2}$$

- **Assumption 2.** The probability that a body of evidence is observed depends on its associated model instance but not on other model instances. This assumption is formulated as:

$$\begin{aligned} P(e_i|m_J) &= P(e_i|m_i) \quad \text{if } i \in J \\ &= P(e_i) \qquad \text{if } i \notin J \ . \end{aligned} \tag{A.3}$$

$$\tag{A.4}$$

Proof. The proof then proceeds as follows:

$$\begin{aligned} P &= p(m_0, m_1, \ldots, m_n|e_1, \ldots, e_n) \\ &= \textstyle\prod_i p(m_i|m_1, \ldots, m_{i-1}, e_1, \ldots, e_n) && \text{(Recursive Decomposition)} \\ &= \textstyle\prod_i p(m_i|m_1, \ldots, m_{i-1}, e_i) && \text{(Assumption 1)} \\ &= \textstyle\prod_i p(m_i) p(m_1, \ldots, m_{i-1}, e_i|m_i)/p(m_1, \ldots, m_{i-1}, e_i) && \text{(Bayes' Rule)}\ . \end{aligned} \tag{A.5}$$

Using Assumption 2, we can simplify first the numerator of the intermediate equation

$$\begin{aligned} p(m_0, m_1, \ldots, m_{i-1}, e_i|m_i) &= p(e_i|m_0, m_1, \ldots, m_{i-1}, m_i) p(m_0, m_1, \ldots, m_{i-1})|m_i) \\ &= p(e_i|m_i) p(m_0, m_1, \ldots, m_{i-1})|m_i), \end{aligned} \tag{A.6}$$

and then the denominator,

$$\begin{aligned} p(m_0, m_1, \ldots, m_{i-1}, e_i) &= p(e_i|m_0, m_1, \ldots, m_{i-1}) p(m_0, m_1, \ldots, m_{i-1})) \\ &= p(e_i) p(m_0, m_1, \ldots, m_{i-1}). \end{aligned} \tag{A.7}$$

The final step is to use Bayes' rule again to regroup the terms involving only m_j, yielding

$$\begin{aligned} P &= \prod_i \frac{p(m_0, m_1, \ldots, m_{i-1}|m_i)}{p(m_0, m_1, \ldots, m_{i-1}} p(m_i) \prod_{i>0} \frac{p(e_i|m_i)}{p(e_i)} \\ &= p(m_0, m_1, \ldots, m_n) \prod_{i>0} \frac{p(e_i|m_i)}{p(e_i)} \ , \end{aligned} \tag{A.8}$$

which is our final result.

B Computing Encoding Costs

The minimal-length description solution to the problem of choosing a single description is based on the observation that it is always possible to design an optimal descriptive language $L_{\mathcal{F}}$ for an ergodic process $\mathcal{F}$ such that the shortest description of the input has the length

$$|\mathcal{D}^*_{\mathcal{F}}(\text{input})| = -\log_2 P(\text{input}), \tag{B.1}$$

in bits[8] [17]. Such a descriptive language is optimal in the sense that no other descriptive language can expect to produce a shorter description than this, on the average.[9] A consequence of this optimality is that there exists a unique shortest description for every input string, because otherwise there would exist "wasted" descriptions, those that map to the same input, that could have been used for other inputs but were not; hence, one could have devised a more efficient descriptive language that made use of these "wasted" descriptions. Note, however, that there are always many different optimal descriptive languages for a given ergodic process, but they are equivalent to each other in the sense that there exist one-to-one mappings between them, as a consequence of the uniqueness of the description.

Independent Symbols : Evaluation of Effectiveness. When the input strings can be represented as $\{x_0, x_1, \ldots, x_n\}$, where the x_i are independently drawn from a known distribution, we see from Eq. (B.1) that the we can design a descriptive language such that the description length is

$$\begin{aligned} |\mathcal{D}_1(\text{input})| &= -\log_2 P(\{x_0, x_1, \ldots, x_n\}) \\ &= -\log_2 \prod_{i=0}^{n} P(x_i) \\ &= \sum_{i=0}^{n} -\log_2 P(x_i) \ . \end{aligned} \tag{B.2}$$

[8] For some distributions, one would need to encode an infinitely long input string in order to achieve exactly this efficiency. A more precise statement is that we can achieve an efficiency as close to this optimum as we like by encoding sufficiently large chunks of the input string at a time.

[9] This is not to say that no other descriptive language can do better on any given finite input string, but only that no other language can do better on the average, or, equivalently, no other language can do better for arbitrarily long input strings.

In the computation of edge and area encoding effectivenesses of Eq. (11) and Eq. (9), we directly use Eq. (B.2) to estimate the cost of encoding whether or not boundary pixels satisfy our edge criterion and whether or not interior pixels are outliers. Similarly, we model the deviations from the planar fit of the interior area photometry by a normal distribution $N(0, \sigma^2)$. These deviations are rounded and represented within a histogram with bin width of 1. Assuming that the deviations are drawn from a normal distribution, the probability of an element with deviation r is

$$\begin{aligned} P(r) &= \int_{r_0}^{r_0+1} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-x^2}{2\sigma^2}\right] dx \\ &\approx \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-r^2}{2\sigma^2}\right] , \end{aligned} \tag{B.3}$$

where r_0 is the integer such that $r_0 \leq r < r_0 + 1$, and the approximation is valid provided $\sigma \gg 1$ (in practice $\sigma > 2$). The total cost of encoding the n pixels within the Gaussian peak is then

$$\begin{aligned} C &= \sum - \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[\frac{-r^2}{2\sigma^2}\right] \\ &= n\frac{\log 2\pi}{2} + n \log e \left(\log_e \sigma + \frac{1}{2\sigma^2} \frac{\sum r^2}{n} \right) \quad . \end{aligned} \tag{B.4}$$

It is easy to see that C is minimized when σ^2 is equal to the variance $\sum r^2/n$ of the deviations, and is given by

$$C = n(\frac{1}{2} \log(2\pi e) + \log \sigma) \quad . \tag{B.5}$$

The first-order statistics of an input string (the probability of occurrence of a symbol) capture some aspects of the structure of the input; however, unless the symbols are independent and identically distributed, one can exploit dependencies to encode the data much more efficiently.

Encoding Images using a Laplacian Pyramid: the Scale Parameter. In the case of image data, Burt et al. [6] have proposed a data compression scheme that achieves this effect. To encode an image, pixel-to-pixel correlations are first removed by subtracting a low-pass-filtered copy of the image from the image itself. The net result is that the data are compressed, since the difference-image has low variance and entropy, and the low-pass-filtered image may be represented at reduced sample density. These steps are then repeated to compress and reduce recursively each low-pass image. Iteration of the process generates a pyramid data structure.

In this pyramid, the upper levels, which are very cheap to encode, describe the low frequencies, while the lower levels, which are more expensive, represent the high frequencies. Using this scheme it takes about 2 bits per pixel to completely encode an 8-bit image such as the one of Figure 19. We can also reconstruct an image using only the upper levels of the pyramid while ignoring the lower ones. The resulting image can be encoded using far less than 2 bits per pixel, but lacks the high frequencies of the original image and appears to be a blurred version of it. However, if we are interested only in large-scale structures, the blurred image may contain all the information we need. Consider once again the example of a shingled roof. The general shape of the roof may be adequately described in the low-frequency image while the shingles are not, since they correspond to higher frequencies. To recognize large objects, we use the global description of the roof but not of the individual shingles: the information encoded in the low-frequency image is therefore perfectly adequate for this purpose.

In this context, we can better understand the role of the *scale parameter* s introduced in the main text. In the absence of a model, the information in an 8-bit image can be encoded using $8/s^2$ bits per pixel. Increasing s amounts to describing the image using fewer and fewer bits of information, which, in the pyramid encoding scheme, can be done by omitting the lower levels of the pyramid and ignoring the high frequencies in the image. The scale can therefore be regarded as a measure of the maximal (Nyquist) frequency of interest in the image, the higher frequencies being regarded as irrelevant noise. The relevant data in the image can be faithfully represented by sampling the signal at twice this frequency.

C Internal Parameter Encoding Cost

Each model can have an arbitrary set of internal parameters $\{\theta\}$, such as the three parameters needed to specify the intensity plane of section 2.2.1, so that

$$p(e_i|m_i) = \int d\theta \;\; p(e_i|m_i, \theta). \tag{C.1}$$

However, as shown by Rissanen and Schwartz [30, 34], $\log p(e_i|m_i)$ can be estimated by finding the optimal θ and using

$$\begin{aligned} \log p(e_i|m_i) &= \log \int d\theta \;\; p(e_i|m_i, \theta) \\ &\approx \max_\theta \log p(e_i|m_i, \theta) - \frac{k}{2} \log N, \end{aligned} \tag{C.2}$$

where k is the number of parameters in $\{\theta\}$ and N is the total number of data samples used to evaluate this model. Thus, in our objective functions, we need not explicitly deal with the internal parameters $\{\theta\}$; in fact, the logarithmic contribution is normally so small relative to the other terms that we can omit this term in practice. For further details, we refer the reader to the original literature.

D Local Optimization Algorithm

In this appendix, we present the details of the procedure used to deform a curve to the nearest local extremum of the objective function by locally optimizing an effective potential. We first describe the effective potential, then present the steps of the optimization procedure and the computation of the derivatives of the potential.

D.1 The Effective Potential

To allow gradient-based optimization, the edge term must have easily computable derivatives. Therefore, we replace the edge effectiveness F_E in the potential by

$$F_{\text{grad}} = +\frac{1}{s} \sum_{\text{curve}} \begin{cases} \log \frac{g(x,y)}{g_0} & \text{if } g > g_0 \\ 0 & \text{otherwise,} \end{cases} \tag{D.1}$$

where g_0 is the median edge strength in the image and

$$g(x,y) = \left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 \quad . \tag{D.2}$$

The logarithm serves the purpose of smoothing out the effect of large values in the gradient itself. The derivatives of F_{grad} with respect to displacements in x and y can be precomputed.

We therefore take the effective potential to be

$$V = F_A + F_{\text{grad}} \quad . \tag{D.3}$$

D.2 Steps in the Optimization Procedure

We describe the curve as an ordered list of contiguous points C represented by the array X of their integer x coordinates and the array Y of their y coordinates. The edge term F_{grad} is computed using the boundary pixels and the area effectiveness F_A of the pixels enclosed by the boundary but not belonging to it. C is then optimized using a gradient ascent procedure that performs the following operations at every step.

1. **Compute the derivatives of the potential.** Compute the derivative of V with respect to deformations of the contour C:

$$\begin{aligned} \frac{\partial V}{\partial X} &= \frac{\partial F_A}{\partial X} + \frac{\partial F_{\text{grad}}}{\partial X} \\ \frac{\partial V}{\partial Y} &= \frac{\partial F_A}{\partial Y} + \frac{\partial F_{\text{grad}}}{\partial Y} \quad . \end{aligned} \tag{D.4}$$

   In the next subsection, we derive expressions for these derivatives.

2. **Increment the curve in the direction of the derivatives.** Since the magnitude of the local derivatives is not related to the current distance of the contour from its optimal location, pick a step size δ and retain only the sign of the derivatives indicating in which direction the contour should move. The result is an array V_x with elements -1,0 and 1 for the local x derivatives and a similar array V_y for the y derivatives. We then normalize the arrays so that $\sum_C ({v_x}^2 + {v_y}^2)/n = \delta^2$ where v_x and v_y are the elements of V_x and V_y, and n is the number of points in C. Finally, increment X by V_x and Y by V_y, thereby ensuring that the displacement of each point is on the average of magnitude δ.

3. **Smooth the curve and fit the geometric model.** Gaussian smooth the curve arrays X and Y. It can be shown [13] that this smoothing procedure is similar in philosophy to the procedure described in the original snake paper [20]; experimentally, the two procedures yield similar results.

   In the case of rectilinear polygons, we fit the geometric model to the curve as follows: look for maxima of curvature along the curve, fit straight line segments between them, compute the average direction of the segments modulo 90°, and force every segment to be parallel or perpendicular to the average direction. Parallel and contiguous segments are merged, while perpendicular ones form corners. In this way, corners appear or disappear as needed to optimize V.

4. **Update the curve.** Recompute C, F_A and F_{grad} either by drawing lines between points that are no longer contiguous and merging points that have identical coordinates, or by extrapolating corners to a common vertex.

Since the objective function is highly nonconvex, after each iteration we recompute the score and verify that it has increased. If the score has decreased, the curve is reset to its previous position and the system tries to use a different step size. The optimization proceeds until the curve stabilizes.

D.3 Derivatives of potential

D.3.1 Area term F_A.

To estimate the derivatives of F_A, we first compute the contribution dF_A of every point (x, y) in the image when added to the patch defined by C. We recall that

$$\begin{aligned} F_A &= (8 - k_A)\frac{A}{s^2} \\ &= \frac{1}{s^2}\left((8 - c - \log\sigma)n - E(n, \overline{n})\right) \ , \end{aligned} \tag{D.5}$$

where $c = \frac{1}{2}\log(2\pi e)$ and n and $\overline{n}$ are the numbers of normal and anomalous pixels, respectively, which we can rewrite as:

$$F_A \ = \ n(c_1 - \frac{\log v}{2}) + n\log n + \overline{n}\log\overline{n} - A\log A, \tag{D.6}$$

where $c_1 = 8.0 - c$ and $\mathrm{v} = \sigma^2$. To evaluate the contribution of an individual pixel we must distinguish two different cases.

1. The pixel's deviation d from the planar fit lies in the main Gaussian peak. In that case, n and A must be incremented by 1, while the the overall variance v is modified by $dv \approx (d^2 - v)/n$. Therefore dF_A can be computed as follows:

$$\begin{aligned} dF_A &= (c_1 - \frac{\log v}{2}) - \frac{c_2}{2} n \frac{dv}{v} + \log n - \log A \\ &= (c_1 - \frac{\log v}{2}) - \frac{c_2}{2}(\frac{d^2}{v} - 1) + \log n - \log A \ , \end{aligned} \tag{D.7}$$

where $c_2 = \log_e 2$.

2. The pixel does not belong to the main peak. Its contribution to $\overline{n}$ and dF_A can then be taken as

$$dF_A \quad = \quad \log \overline{n} - \log A.$$

Having computed dF_A, we can now estimate $\partial F_A / \partial X$ using finite differences. Let us consider a boundary point $P = (x, y)$. Our implementation assumes that the boundary points themselves do not belong to the patch. There are four possible patterns for the 3×1 horizontal neighborhood centered around P:

Case a:	1	P	0
Case b:	0	P	1
Case c:	1	P	1
Case d:	0	P	0

where 0 represents a point that does not belong to the patch and 1 represents a point that does.

- Case a: If P moves to the right, the center point is added to the patch and F_A becomes $F_A + dF_A(x, y)$; conversely if P moves to the left, the left point is removed from the patch and the F_A becomes $F_A - dF_A(x - 1, y)$. $\partial F_A / \partial x$ is therefore estimated to be

$$\frac{\partial F_A}{\partial x} = + \frac{dF_A(x, y) + dF_A(x - 1, y)}{2}. \tag{D.8}$$

- Case b: Similarly

$$\frac{\partial F_A}{\partial x} = - \frac{dF_A(x, y) + dF_A(x + 1, y)}{2}. \tag{D.9}$$

- Case c and d: The boundary is locally horizontal,

$$\frac{\partial F_A}{\partial x} = 0. \tag{D.10}$$

$\partial F_A / \partial X$ is the array of the $\partial F_A / \partial x$ for all the points in C. $\partial F_A / \partial Y$ is computed similarly by replacing horizontal neighborhoods by vertical ones. Note that dF_A can be computed on a pixel per pixel basis and therefore in parallel for all pixels in the image.

D.3.2 Edge term $F_{\textbf{grad}}$.

Referring to Eq. (D.2), we write F_{grad} as

$$F_{\text{grad}} = \frac{1}{s} \sum_{C(x,y)} \log \frac{g(x, y)}{g_0}. \tag{D.11}$$

Here g_0 is the minimum gradient threshold required for an edge to be considered. In practice, we precompute, once and for all, the quantity Γ defined by

$$\Gamma(x,y) = \begin{cases} \log(g(x,y)/g_0) & \text{if } s > g_0 \\ 0 & \text{otherwise} \end{cases} . \tag{D.12}$$

We also precompute the derivative of Γ, $\partial\Gamma/\partial x$ and $\partial\Gamma/\partial y$. At each iteration, $\partial F_{\text{grad}}/\partial X$ and $\partial F_{\text{grad}}/\partial Y$ are simply the arrays whose components are the values of $\partial\Gamma/\partial x$ and $\partial\Gamma/\partial y$ at the current boundary points.

The results shown in the text have been computed using the values that appear in Table 1.

Parameters	**Values**
Gaussian smoothing	Gaussian of variance 1.0.
Step sizes	$\delta = 4, 2, 1$ and 0.5
Scale	$s = 2$

Table 1: Parameters for local optimization.

E Hypothesis-Generation Algorithm

In this appendix, we describe in detail the hypothesis generation procedure that we have implemented for automated building extraction. To generate all the results that appear in section 4, we have used the single setting of the control parameters defined by Tables [2,3,4,5,6]; although some of these parameters are arbitrary, they exhibit a high degree of image independence.

E.1 Edges

Procedure.

- **Build Canny edge hierarchy.** Build an edge hierarchy by applying the Canny edge operator [7] to the image with several sets of edge-strength parameters.

- **Link edge points.** In each Canny image separately, link the edge points into segments. We use the linker developed by Fischler et al. [11], which attempts to produce the straightest, longest segments possible.

- **Partition linked edges into straight segments.** Convolve the x and y coordinates of each line segment with derivatives of Gaussians to compute the curvature. Define edges by drawing straight lines between the maxima of curvature and, for each line and optimizing the location of both endpoints to maximize the average gradient along the segment. In reference [14], this optimization procedure is described in detail, and it is shown that along an optimized segment, the number of pixels satisfying the maximal-gradient criterion of section 2.2.2 is maximized. Edges whose percentage of maximal-gradient pixels exceeds the chosen threshold are retained for further processing.

Since the width of the Gaussians is arbitrary, we reduce the parameter dependence of our procedure by repeating the operation for a set of progressively wider Gaussians. Several sets of possibly overlapping edges are thus produced. The system retains the subset of nonoverlapping edges that maximizes the overall number of pixels satisfying the maximal-gradient criterion.

- **Merge segments across hierarchy.** Similarly, the system chooses from the set of possibly overlapping edges found independently in each of the Canny images the subset that maximizes the number of pixels satisfying our edge criterion.

Parameters	**Values**
Minimum edge length	10 pixels.
Quality threshold	70% of pixels must satisfy maximal-gradient criterion.
Curvature computation	Gaussian of width 2, 4, 6, 8.
Canny thresholds	High and low thresholds $\{(400, 200), (200, 100), (100, 50)\}$.
Canny smoothing kernel	Gaussian of width 1.
Overlapping edges	Crossing or parallel with centers less than 3 pixels apart.

Table 2: Parameters for Edge extraction.

The parameters that we have used appear in Table 2. Both the curvature computation and Canny threshold parameters are arbitrary, but, since we use a hierarchy of values, the output of our system is largely insensitive to the values chosen. The edge length parameter can in principle be estimated from the image digitization scale. The quality threshold is heuristically chosen to reject enough of the bad edges to limit the size of the search space effectively without losing meaningful ones.

Remarks. The hierarchy of Canny edge images has proven useful because the output of an edge linker can change drastically as the edge density increases. Since we cannot predict which level is right, we choose to perform the computation several times and allow our edge criterion to pick the best answer automatically.

E.2 Arcs

Procedure.

- **Define directional edges.** For each edge segment, define two directed edges pointing in opposite directions that will be used to build counter-clockwise contours. Associate a chamfer

Figure 21: Elementary geometric relationships: corner, parallel and collinear, with their completion paths (dotted lines) and associated area masks (shaded areas).

mask to the left side, the logical interior, of each directed edge and compute its mean grey-level value.

- **Find elementary geometric relationships.** For every pair of edges within a distance limit, check whether or not they can form one of the elementary geometric *arc* structures (corner, parallel or collinear) shown in Figure 21. The precise definitions we use for these geometric relationships are very close in spirit to those described by Reynolds and Beveridge [29]. Note that only edges whose directions are consistent with the counterclockwise-orientation convention are grouped.

(a) (b) (c)

Figure 22: (a) A search window, the points to be connected are the leftmost and rightmost corners (b) The oriented grid in which the search for a connecting path is constrained. (c) The optimal path.

- **Connect edges.** In general, edges that form arcs are not adjacent. To form a continuous linear structure, bridge the gap between related edges with a rectilinear path of maximum gradient. To achieve this result, define a search rectangle cooriented with the structure in which to look for the path, sum and histogram the edge strengths along the length and width of the search rectangle, and mark the peaks of these histograms. Define a rectilinear grid using these peaks, as shown in Figure 22 and search in this grid for the highest-scoring connecting path.

- **Construct area mask.** Construct a mask that corresponds to the logical interior of each arc as shown in Figure 22. Fit a plane to the intensities of the masked pixels and compute the corresponding area effectiveness. Reject arcs for which the previously computed edges' grey levels do not lie on the mask's intensity plane; also reject those whose area effectiveness is too low. This procedure guarantees that the arcs have both the right geometric and photometric characteristics.

Parameters	Values
Min. parallel width	3 pixels
Max. parallel width	50 pixels
Max. colinear gap	50 pixels
Angular tolerance	15°
Quality threshold	Interior area encoding in less than 7.5 bits per pixel.
Proximity to plane	Grey level belonging to peak of histogram of deviations from planar fit to mask intensities (see section 2.2.1).

Table 3: Parameters for arc contruction.

Like the edge-length threshold of section E.1, the first three parameters in Table 3 are scale-dependent and are in principle computable from the image digitization scale. The quality threshold serves the same purpose as the edge quality threshold of section E.1. The angular tolerance accounts for both digitization errors and oblique imagery.

E.3 Cycles

Procedure.

- **Cluster similar edges.** Define the distance between edges forming an arc to be the number of bits per pixel required to encode the arc's masked pixels; edges that do not belong to a common arc are considered as infinitely distant. Use this distance measure and a nearest-neighbor clustering algorithm to group edges into sets smaller than a specified size.

- **Find linkable arcs.** Within a cluster, define two arcs as *linkable* if they share a compatibly oriented edge and their planar intensity fits are compatible, i.e., the centroid of each plane must be less than a fixed distance from the other plane.

- **Build cycles within clusters.** Choose a quality threshold and build maximal cycles by chaining linkable arcs whose quality is above the threshold. Increment the threshold and iterate the procedure until the threshold reaches the minimum quality threshold of section E.2.

- **Select cycles.** Some cycles may be proper subsets of others: retain only those proper subsets that are more compact than the cycle in which they are included. This heuristic has proven effective because maximal cycles typically do correspond to objects of interest except when irrelevant edges form an appendage or an intrusion. To generate the object contour, one must then use a more compact subcycle in which the appendage has been removed.

Parameters	**Values**
Max. cluster size	100 edges.
Max. cycle size	30 edges.
Min. quality threshold	Interior area encoding in less than 5.0 bits per pixel.
Max. quality threshold	Interior area encoding in less than 7.5 bits per pixel.
Increment of threshold	0.125 bits per pixel.

Table 4: Parameters for cycle construction.

Remarks. Building all possible cycles, even in a small image, would result in combinatorial explosion. The above procedure attempts to build all relevant cycles while avoiding this explosion.

If, as in a perfect "blocks world," we knew exactly which arcs belonged to objects and which did not, we could generate candidate model instances by combining these arcs into maximal cycles, which would then indeed correspond to actual objects. Having no such knowledge, we use the number of bits per pixel required to encode the arc's masked pixels as a quality measure; because we do not know of a quality threshold that would guarantee that only relevant arcs are taken into account, we use a hierarchy of quality thresholds and then merge across the hierarchy. We use inexpensive heuristic quality measures, namely the previously computed area-effectiveness of the arcs and compactness of the resulting cycles, to select a reasonably small set of cycles. Only these cycles will be used in the next step to generate closed contours upon which more expensive tests will be performed.

E.4 Build Enclosures

Procedure.

- **Optimize contours.** The edges of the cycles, along with the arcs' completions, are used to generate closed contours. These contours are then adjusted to optimize their average edge strength using a technique inspired by the "snake" algorithm [20] and described in detail elsewhere [14].

  Alternatively, for images with difficult photometry, we adjust the contours using the more sophisticated technique described in section 3. To avoid shallow local minima of the objective function we add a randomization step: we randomly displace the vertices of the initial contour several times, perform the optimization starting with each of these displaced contours, and retain only the highest-scoring result.

- **Compute elevation.** We assume that the detected contours lie in planes whose position in space is defined by three parameters: elevation, tilt and roll. The system first performs a global search on these three parameters by coarsely quantizing the search space, computing the value of the stereo effectiveness F_S of Eq. (6) for each set of values, and retaining the ones that maximize F_S.

  To adjust these parameters more precisely, the system then performs gradient ascent in parameter space to maximize the average edge strength of the contour's projection onto the second image.

- **Compute score.** Given the complete contours and their elevations, the system can now compute their score according to Eq. (3).

Parameters	**Values**
Contour optimization	See Table 1.
Elevation quantization	Height between 0 and 60 feet at intervals of 5 feet, tilt = 0.

Table 5: Parameters for enclosure construction.

E.5 Select Enclosures

Procedure.

- **Select contours.** Test the contours for stability by requiring both that a minimal edge quality be met and that the grey levels of the pixels immediately outside the contour do not belong to the intensity plane. Note that, as mentioned section 2.3, for a particular value of the scale s we need consider only those contours that have a positive score.

- **Find nonconflicting contours.** For each contour, compute an interior area mask and define nonconflicting contours as those whose masks are either disjoint or fully included in one another.

- **Find best subset of nonconflicting contours.** Find all subsets of compatible enclosures, compute their total score as the sum of the individual scores, and rank them.

Parameters	Values
Scale	$6 < s < 10$
Geometric cost	$G_i = 20 + L/s$
Border quality	70% of pixels must satisfy maximal-gradient criterion.
Stability criterion	No more than 70% of the pixels in a 2-pixel-wide border outside the region lie on the interior intensity plane.

Table 6: Parameters for enclosure selection.

Remarks. The form of the stability criterion chosen here is motivated by the observation that, in some situations, a good feature may be adjacent to a large area with very similar intensity that has a parallel edge somewhere in the middle; if this edge becomes associated with the maximal cycle, a rectilinear enclosure results that has a long edge pair (across the large area) with little contrast across it. The rectilinear geometry is perfect, but the structure is erroneous and unstable because any displacement of the contrast-free edge results in a similar structure with a similar score. Requiring high contrast makes this type of misidentification less likely.

References

[1] N. Ayache and O. Faugeras, "HYPER: a New Approach for the Recognition and Positioning of Two-Dimensional Objects," IEEE Trans. PAMI **8**(1), pp. 44–54 (1986).

[2] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," Pattern Recognition, **13**, pp. 111–122 (1981).

[3] T.O. Binford, "Survey of Model-Based Image Analysis Systems," The International Journal of Robotics Research **1**, No. 1, pp. 18–64 (Spring, 1982).

[4] R.C. Bolles and R. Horaud, "3DPO, A Three-Dimensional Part Orientation System," International Journal of Robotics Research **5**, pp. 3–26 (1986).

[5] R.A. Brooks, "Symbolic Reasoning Among 3-D Models and 2-D Images," Artificial Intelligence Journal **16**, (1981).

[6] P.J. Burt and E.H. Adelson, "The Laplacian Pyramid as a Compact Image Code," IEEE Trans. on Comm. **4**, pp. 532–540 (1983).

[7] J. Canny, "A Computational Approach to Edge Detection," IEEE Trans. PAMI **8**, pp. 679–698 (1986).

[8] J.A. Feldman and Y. Yakimovsky, "Decision Theory and Artificial Intelligence: I. A Semantics-Based Region Analyzer," Artificial Intelligence **5**, pp., 349–371 (1974).

[9] M.A. Fischler and R.A. Elschlager, "The Representations and Matching of Pictorial Structures," IEEE Trans. on Computers **C-22** pp. 67–92 (1973).

[10] M.A. Fischler, J.M. Tenenbaum, and H.C. Wolf, "Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique," Computer Graphics and Image Processing **15**, pp. 201–223 (1981).

[11] M.A. Fischler and H.C. Wolf, "Linear delineation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, D.C., 351–356 (1983).

[12] P. Fua and A.J. Hanson, "Using Generic Geometric Models for Intelligent Shape Extraction," Proceedings of the AAAI Sixth National Conference on Artificial Intelligence, pp. 706–711 (July 1987).

[13] P. Fua, "Object Delineation as an Optimization Problem, A Connection Machine Implementation," Proceedings of the Fourth International Conference on Supercomputing, Santa Clara, CA (May 1989).

[14] P. Fua and Y.G. Leclerc, "Model Driven Edge Detection," Machine Vision and Applications, in press.

[15] S. Geman and D. S. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," IEEE PAMI, **6**, pp. 721–741 (1984).

[16] M.P. Georgeff and C.S. Wallace, "A General Selection Criterion for Inductive Inference," Proceedings of the Advances in Artificial Intelligence Conference, Pisa, Italy (September 1984).

[17] R.W. Hamming, "Coding and Information Theory," Prentice Hall, New Jersey (1985).

[18] A.J. Hanson and L. Quam, "Overview of the SRI Cartographic Modeling Environment," in Proceedings of the Image Understanding Workshop, Boston MA, pp. 576–582 (April 1988).

[19] R.M. Haralick, "Digital Step Edges from Zero Crossings of Second Directional Derivatives," IEEE Trans. PAMI **6**, pp. 58–68 (1984).

[20] M. Kass, A. Witkin, and D. Terzopoulos "Snakes: Active Contour Models," International Journal of Computer Vision, **1**(4) pp. 321–331 (1988).

[21] K.I. Laws, "Goal-Directed Texture Segmentation," Technical Note 334, Artificial Intelligence Center, SRI International, Menlo Park, CA (September 1984).

[22] K.I. Laws, "Integrated Split/Merge Image Segmentation," Technical Note 441, Artificial Intelligence Center, SRI International, Menlo Park, CA (July 1988).

[23] Y. G. Leclerc, "Constructing Simple Stable Descriptions for Image Partitioning," International Journal of Computer Vision, **3**(1), pp. 73–102 (1989).

[24] D. McKeown, W.A. Harvey, and J. McDermott, “Rule-Based Interpretation of Aerial Imagery,” IEEE Trans. PAMI **7**, pp. 570–585 (1985).

[25] A. Huertas and R. Nevatia, “Detecting Buildings in Aerial Imagery,” Computer Vision, Graphics and Image Processing **41**, pp. 131–152 (1988).

[26] Y. Ohta, T. Kanade, and T. Sakai, “A Production System for Region Analysis,” Proc. 6th IJCAI Conference, pp. 684–686 (1979).

[27] E.P.D Pednault, “Some Experiments in Applying Inductive Inference Principles to Surface Reconstruction,” Proc. 11th IJCAI Conference, Detroit Mi (1989).

[28] L.H. Quam, “Road Tracking and Anomaly Detection in Aerial Imagery,” Proceedings: Image Understanding Workshop, pp. 51–55 (May 1978).

[29] G.Reynolds and J.Ross Beveridge, “Geometric Line Organization Using Spatial Relations and a Connected Components ALgorithm,” COINS Technical Report 87-03, University of Massachusetts at Amherst (January 1987).

[30] J. Rissanen, “A Universal Prior for Integers and Estimation by Minimum Description Length,” The Annals of Statistics **2**, pp. 416–431 (1983).

[31] J. Rissanen, “Minimum-Description-Length Principle,” in *Encyclopedia of Statistical Sciences*, **5**, pp. 523-527, (1987).

[32] A. Rosenfeld, “A Nonlinear Edge Detection Technique,” Proc. IEEE, **58**. pp. 814–816 (1970).

[33] C.E. Shannon, “A Mathematical Theory of Communication,” Bells Systems Tech J., **27**, pp. 623–656 (1948).

[34] G. Schwarz, “Estimating the Dimension of a Model,” The Annals of Statistics **6**, pp. 461–464 (1978).

[35] M.O. Shneier, R. Lumia, and E.W. Kent, “Model-Based Strategies for High-Level Robot Vision,” Computer Vision, Graphics and Image Processing **33**, pp. 293–306 (1986).

[36] D. Terzopoulos, “On Matching Deformable Models to Images,” Topical Meeting on Machine Vision, Technical Digest Series, Optical Society of America, Washington, D.C., **12**, pp. 160–167 (1987).