Segmentation and Classification of Edges Using Minimum Description Length Approximation and Complementary Junction Cues

Tony Lindeberg and Meng-Xiang Li

Computational Vision and Active Perception Laboratory (CVAP)*
Department of Numerical Analysis and Computing Science,
KTH (Royal Institute of Technology), S-100 44 Stockholm, Sweden.
Email: tony@bion.kth.se, mxli@bion.kth.se

Technical report ISRN KTH/NA/P–96/01–SE. January 1996.

To appear in Computer Vision and Image Understanding

Abstract

This article presents a method for segmenting and classifying edges using minimum description length (MDL) approximation with automatically generated break points. A scheme is proposed where junction candidates are first detected in a multi-scale preprocessing step, which generates junction candidates with associated regions of interest. These junction features are matched to edges based on spatial coincidence. For each matched pair, a tentative break point is introduced at the edge point closest to the junction. Finally, these feature combinations serve as input for an MDL approximation method which tests the validity of the break point hypotheses and classifies the resulting edge segments as either "straight" or "curved". Experiments on real world image data demonstrate the viability of the approach.

Keywords: curve segmentation, minimum description length, corner detection, edge detection, curvature, classification, object recognition, computer vision

*We would like to thank Göran Olofsson for the enjoyable collaboration on the application to object recognition as well as for interesting discussions. This work was partially performed under the ESPRIT-BRA project VAP and the ESPRIT-NSF collaboration Diffusion. The support from the Swedish Research Council for Engineering Sciences, TFR, is gratefully acknowledged.

Contents

1 Introduction

Several object recognition systems for man-made objects are based on detection of image features, such as edges, in an initial processing step. To reduce the complexity when matching image features to a database of object models, an important step in the low-level processing is to derive higher-order descriptors from the image primitives. Concerning edges, one example is classification into "straight" or "curved". This step is particularly important within the recognition-by-components paradigm (Binford, 1971; Biederman, 1985; Dickinson et al., 1992; Bergevin and Levine, 1993), where the primitives for object modelling are distinguished by such qualitative properties. More generally, the need for classifying edges into "straight" and "curved" arises whenever performing edge based object recognition of curved objects (Requicha, 1980; Fisher, 1989; Grimson, 1990), or when analysing curved objects by line-drawing-like techniques (Malik, 1987).

In its general form, the problem of classifying whether a given edge segment should be regarded as "straight" or "curved" is not well-defined. For example, for a given edge segment, the classification may be strongly context dependent. This imposes strong limitations concerning the extent to which the results from a low-level classification can be relied upon. Nevertheless, it is in many cases possible to generate reasonable hypotheses. One type of methodology that can be used for obtaining such cues is by approximating a given curve using different types of straight and curved models and then selecting the model that according to some type of judgement is the one that best fits the data. Since a more complex model, in general, gives a smaller residual, it is natural to also take the complexity of the model into account and aim at a trade-off between these two factors. This is the idea behind the minimum description length (MDL) principle, which focuses on the information contents, and selects the description that can be represented by the smallest number of bits (including the model as well as the deviations between the model and data).

For elongated edges, which are often obtained from common edge detection methods, it is usually necessary to segment a given curve into smaller patches before attempting a classification into compact descriptors such as "straight" or "curved". Trying all possible ways of inserting break points into a given (digital) curve obviously leads to combinatorial explosion. Therefore, it is of interest to develop systematic methods for generating candidate points at which the curve is likely to be segmented.

The subject of this article is to show how such candidate break points can be automatically generated. The methodology that will be proposed is based on a bottom-up processing step, where junctions and edges are detected in a complementary manner using multi-scale techniques. The edges are matched to junctions, and each junction–edge pair constitutes a hypothesis about a point along the edge that is likely to correspond to a corner in the scene. The (elongated) edge segments obtained from the edge detector are then segmented into patches and each patch is individually classified using a minimum description length curve approximation method. Finally, adjacent patches are merged if the information contents in a composed representation is smaller than in the sum of the description lengths of the individual patches.

The presentation is organized as follows: Section 2 reviews the method for junction detection with automatic scale selection on which the overall approach is based. An important property of this method is that each junction candidate is associated with an adaptively determined region of interest. This simplifies the subsequent matching to edges as described in section 3. Section 4 reviews the specific MDL method we use for curve classification and demonstrates the advantages of using break points as obtained from the proposed approach. Finally, section 5 presents experimental results and section 6 gives a brief summary and discussion about the approach.

Concerning overall assumptions, we assume that the underlying image data have been acquired from scenes of man-made objects that can be well described by object models of geon-type and for which a classification into the given set of qualitative descriptors (straight and curved) is sufficient.

2 Junction Detection with Automatic Scale Selection: Review

The fact that image structures exist as meaningful entities only over certain ranges of scale shows that a multi-scale approach is essential when extracting information, such as features, from image data. Whereas scale-space theory (Witkin, 1983; Koenderink, 1984) provides a canonical framework for modelling visual operations at multiple scales, it is in many cases necessary to complement this framework by explicit mechanisms for generating hypotheses about appropriate scales (Lindeberg, 1994c).

A junction detector with automatic scale selection has been developed by Lindeberg (1993c, 1994a). It is beyond the scope of this article to extensively describe this method or to motivative the need for a scale selection mechanism with respect to the problem of junction detection. Since, however, a number of the specific properties of this method are important for the proposed methodology, we shall briefly review its most important steps:

The scale-space representation $L\colon \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ of a two-dimensional signal $f\colon \mathbb{R}^2 \rightarrow \mathbb{R}$ is defined as the one-parameter family of functions obtained by convolving f with Gaussian kernels $g\colon \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}$ of different widths,

$$L(\cdot;\ t) = g(\cdot;\ t) * f \tag{1}$$

where t is the scale parameter of the Gaussian $g(x_1, x_2;\ t) = 1/(2\pi t)\, e^{-(x_1^2+x_2^2)/2t}$. In this representation, normalized derivatives are defined by

$$\partial_\xi = \sqrt{t}\, \partial_x. \tag{2}$$

Over the years, a large number of different approaches to corner detection have been developed, see for example (Kitchen and Rosenfeld, 1982; Dreschler and Nagel, 1982; Koenderink and Richards, 1988; Noble, 1988; Deriche and Giraudon, 1990; Blom, 1992; Florack et al., 1992; Rohr, 1992). To detect junction candidates from grey-level images, we shall in this article follow the commonly used differential approach of defining a junction detector in terms of the curvature of level curves multiplied by the gradient magnitude raised to some power. Selecting the power of three gives

$$\tilde{\kappa} = L_{x_2}^2 L_{x_1x_1} - 2L_{x_1} L_{x_2} L_{x_1x_2} + L_{x_1}^2 L_{x_2x_2}. \tag{3}$$

known as the rescaled level curve curvature. Then, replacing each derivative by its corresponding normalized derivative gives *normalized rescaled level curve curvature*

$$\tilde{\kappa}_{norm} = t^2 \tilde{\kappa}. \tag{4}$$

The method in (Lindeberg, 1993c, 1994a) is based on the detection of *scale-space extrema* (simultaneous maxima over scale and space) of $\tilde{\kappa}_{norm}^2$. In addition to allowing for feature detection without external choice of scale levels, an attractive property of this approach is that the selected scale levels at which the maxima over scales are assumed, will be larger for corners that have a large spatial extent in the image domain. In other words, the scale-space maxima serve as indicators reflecting of the spatial extent of the corresponding image structures.

Figure 1(b) shows the 100 most significant junction candidates extracted from an image of an office scene. Each scale-space maximum is illustrated by a circle with the area proportional to the scale at which the maximum is assumed. Note that coarser scales are selected for the junctions having larger spatial extent and vice versa.

In the abovementioned references, it has been argued that such qualitative scale and region descriptors are useful for guiding later processes stages. Here, we shall use this type of attribute information for defining an adaptively determined region of interest around each junction candidate, which will be used when matching junctions to edges. More generally, such regional descriptors can also be used for purposes such as junction classification (Brunnström et al., 1992) and junction tracking (Bretzner and Lindeberg, 1995).

Whereas the junction detection step is conceptually clean, it can certainly lead to poor localization. One way to improve the localization is by applying a modified Förstner operator

(Förstner and Gülch, 1987) which iteratively computes new localization estimates using scale information from the detection step. Besides improving the localization, such an iterative computation of new localization estimates provides a way to suppress a large number of the false positives, by suppressing junction candidates for which the iterative computation of new localization estimates diverges.

Figure 1(c) shows the result of applying this type a post-processing scheme to the data in figure 1(b).[1] As can be seen, a compact set of junction candidates is obtained, and the accuracy of the localization estimates has been substantially improved.

3 Matching Junctions to Edges

Given these junction descriptors, it is straightforward to perform matching to edges based on spatial coincidence. Since the regions of interest are adaptively determined from the local image structure, useful result can be obtained from the simple matching criterion that a junction region and an edge should overlap. Thereby, it is not necessary to introduce any explicit threshold on, *e.g.*, the distance between the corner and the edge, and we avoid the inherent limitations of applying such a global thresholding operation to data in which the size variations are large.

The left image in figure 2 shows the result of detecting edges at a fine scale ($t = 1$) using an implementation of non-maximum suppression (Canny, 1986) expressed within the scale-space framework (Lindeberg, 1994c) (A useful property of this specific algorithm expressed in terms of level curves is that it gives comparably low fragmentation, provided that the edges are detected at sufficiently coarse scales.) The right image shows the result of matching these edges to the previously extracted junction candidates. For each edge that crosses a junction region, a match is registered. Then, for graphical illustration, a marker (a black or white square) has been placed at the edge point closest to the corresponding junction. Note how these points serve as natural break points for segmenting the edges into patches with geometrically more meaningful interpretation. These patches will (with slight modification) serve as input for the MDL classifier.

Besides the specific application that these matching relations will be used for in this article, they are more generally useful for problems related to object recognition (Lindeberg and Olofsson, 1995).

4 Minimum Description Length Curve Approximation

To decide whether a given curve segment should be regarded as "straight" or "curved" is a problem closely related to model selection. In this section, we shall use the minimum description length principle for making such judgements. This is an information theoretic approach closely related to maximum-likelihood and maximum-a-posteriori estimation, although with the attractive property that it allows for model selection in addition to parameter estimation (Rissanen, 1987; Leclerc, 1989).

Intuitively, the basic idea behind the MDL principle is to select the simplest model that explains the data. More technically, this approach can be described as follows: Let $\phi = (\phi_1, \dots, \phi_k)$ be a parameter vector with k components and let $p(x \mid \phi)$ represent a parameterized class of probability functions that assigns a probability to any observation $x = (x_1, \dots, x_n)$. In classical maximum-likelihood estimation, the problem is to find the $\phi \in \mathbb{R}^k$ that for a given x (and a given number of parameters k) maximizes $p(x \mid \phi)$. Equivalently, this problem can be formulated as finding the $\phi \in \mathbb{R}^k$ that minimizes the negative logarithm of the likelihood

$$\mathcal{L}(x \mid \phi) = -\log_2 p(x \mid \phi). \tag{5}$$

(In information theory, this entity is called the coding (or description) length.) In minimum description length estimation, the information required to represent the model parameters is

[1] The details of the second localization stage are described in the abovementioned references. An outline of the combined two-staged method for junction detection can also be found in appendix A.1.

taken into account as well, leading to minimization of

$$\mathcal{L}'(x, \phi) = \mathcal{L}(x \mid \phi) + \mathcal{L}(\phi) = -\log_2 p(x \mid \phi) + \mathcal{L}(\phi). \tag{6}$$

where $\mathcal{L}(\phi)$ is a measure of the information contents in the parameters.

4.1 Curve Classification by Minimum Description Length Approximation

In computer vision, the MDL approach has been applied to several problems; see for example (Georgeff and Wallace, 1985; Darell et al., 1990; Deren et al., 1990; Axelsson, 1992; Sheinvald et al., 1992). Here, we shall consider the scheme for MDL-based curve approximation developed by (Li, 1993) (which will be extended in several ways). It concerns the problem of representing a digital curve using the following models:

- a set of randomly distributed points,
- a linear model with or without outliers,
- a segment of an ellipse with or without outliers.

Given these shape classes, any edge segment is classified as "straight" if the linear model gives the shortest description and as "curved" if the ellipse approximation results in the shortest one. For each model, the description length is measured by

$$\mathcal{L}'(x, \phi) = (\mathcal{L}_{\#\ points} + \mathcal{L}_{parameters} + \mathcal{L}_{model\ points} + \mathcal{L}_{outliers})(x, \phi) \tag{7}$$

where

- $\mathcal{L}_{\#\ points}(x, \phi)$ is the number of bits needed to represent the total number of points. (This term is similar for all models and not relevant for comparisons.)
- $\mathcal{L}_{parameters}(x, \phi)$ is the description length of the model parameters.
- $\mathcal{L}_{model\ points}(x, \phi)$ is the description length for the $(N - \lambda)$ points that belong to the model.
- $\mathcal{L}_{outliers}(x, \phi)$ specifies the description length of λ points classified as outliers.

An essential parameter when measuring the description length in a minimum description length approximation method is the spatial resolution ε at which the approximation is performed. Assuming that a variable x is uniformly distributed in some interval $[x_0, x_0 + \Delta x]$ of width Δx, the description length of this variable approximated to resolution ϵ can be measured by

$$\mathcal{L}_{coord}(\Delta_x;\ \varepsilon) = \log_2 \frac{\Delta x}{\varepsilon}. \tag{8}$$

When x represents a coordinate of an image point, Δx does of course correspond to the image size. Concerning the choice of ε, it is natural to set this parameter to a value of the same order as the distance between adjacent pixels. (Here, for edges detected with subpixel resolution at scale $t = 1.0$, we have used $\varepsilon = 0.5$.)

Then, based on this construction, and assuming an image of size $M * M$ pixels, the description lengths of the different terms in (7) can be measured as follows:

- *Model parameters:* To parameterize a straight line segment, four parameters are needed. We can, for example, take the four coordinates determining its end points, which gives

  $$\mathcal{L}_{line\text{-}segment}(x, \phi) = 4\mathcal{L}_{coord}(N;\ \varepsilon) = 4 \log_2 \frac{M}{\varepsilon}. \tag{9}$$

  The same idea can be applied to the ellipse segment model, which can be described by seven parameters. The center of the ellipse (x_c, y_c) and the lengths of the two semi-axes a and b can be modelled by uniformly distributed coordinates in the image resulting in a description length of the same form as in (9). Then, we can add three angular descriptors describing the orientation of the ellipse as well as the two end points of the ellipse segment. Since the ellipse size will, in general, be much smaller than the image size, we can parameterize these descriptors by their projections on the coordinate axes quantized relative to the spatial extent of the ellipse. If we model the size of the ellipse by $2\max(a,b)$, the total description length of the ellipse segment will then be of the form

$$\begin{aligned}\mathcal{L}_{ellipse\text{-}segment}(x,\phi) &= 4\mathcal{L}_{coord}(N;\ \varepsilon) + 3\mathcal{L}_{coord}(2\max(a,b);\ \varepsilon)\\ &= 4\log_2\frac{M}{\varepsilon} + 3\log_2\frac{2\max(a,b)}{\varepsilon} \qquad (10)\end{aligned}$$

- *Model points:* To capture approximation errors, the description length of the $(N-\lambda)$ model points is measured by

$$\mathcal{L}_{model\ points}(x,\phi) = (N-\lambda)\, L_{offset}(\xi;\sigma). \qquad (11)$$

where $L_{offset}(\xi;\sigma)$ is the coding length of the offset error, *i.e.*, the amount of information required for representing the distance between a data point and the closest point belonging to the model.

Here, the latter entity is approximated by the expected coding length of the outcome of a centered Gaussian distribution $\xi \sim N(0,\sigma^2)$ quantized with resolution ε. With $\Phi(\xi;\ \sigma^2) = \int_{\eta=-\infty}^{\xi} g(\eta;\ \sigma^2)\,d\eta$, and $[\xi]_\varepsilon$ denoting the integer multiple of ε closest to ξ, we have

$$\begin{aligned}L_{offset}(\xi;\sigma) &= -\log_2\left(\int_{\eta=[x]_\varepsilon+\varepsilon/2}^{[x]_\varepsilon-\varepsilon/2} g(\eta;\ \sigma^2)\,d\eta\right) \qquad (12)\\ &= -\log_2\left(\Phi([\xi]_\varepsilon+\tfrac{\varepsilon}{2};\ \sigma^2) - \Phi([\xi]_\varepsilon-\tfrac{\varepsilon}{2};\ \sigma^2)\right)\\ &\approx \log_2\left(g(\xi;\ \sigma^2)\,\varepsilon\right) = \frac{\log_2 e}{2}\frac{\xi^2}{\sigma^2} + \log_2\left(\frac{\sigma}{\varepsilon}\right) + \frac{\log_2(2\pi)}{2}, \qquad (13)\end{aligned}$$

where Φ represents the primitive function of the Gaussian kernel and σ is estimated from the actual deviations between the model and the data (see also Förstner (1989)).

- *Outliers:* The λ points classified as outliers are modelled as random points having a uniform distribution in the image domain. Hence, the total description length of the λ points classified as outliers is measured by

$$L_{random\ point} = 2\lambda\,\mathcal{L}_{coord}(N;\ \varepsilon) = 2\lambda\log_2\frac{M}{\varepsilon}. \qquad (14)$$

4.2 Algorithm

Given edges and junctions detected as outlined in section 2 with candidate break points generated from the junction-edge matching in section 3, this data is used as input for the MDL approximation scheme.

The edge points from the junction-edge matches serve as tentative break points for splitting edge segments into shorter ones. Moreover, co-linear and co-curvilinear edge segments (whose end points are adjacent) are candidates for being merged. Depending on which model gives the shortest description, segments are merged (cases (a)–(b)) and split (cases (c)–(e)) as shown in figure 3. To allow for merging of fragmented edges, small gaps are filled in if a composed model gives a more compact description.

Another useful processing step is to move the break point along the curve and select the position that minimizes the total description length (see figure 4). Whereas the validity of each break point is evaluated in this algorithm and a better position estimate is computed as well, the major advantage of the proposed approach is that a conservative set of junction candidates is obtained. Restricting the processing to these points serves as a heuristic principle for reducing the otherwise combinatorial explosion in guessing where to split elongated edge segments into shorter ones.[2]

[2] An obvious alternative would be to determine such candidate break points based on edge information only,

5 Experimental results

Figure 5 shows the final result of applying the composed procedure[3] to the image in figure 1 using the image features and candidate break points shown in figure 2. For graphical illustration, straight edge segments have been marked by black lines and curved ones by white. (In the graphical illustration of the classification, all points classified as outliers have been suppressed, which results in a loss of connectivity at some junctions. Internally, however, the complete representations can be maintained.) Figures 6–11 show corresponding results for a set of images of other geon-type objects. Notice how very reasonable segmentations and classifications are obtained.

e.g. by detecting points where the curvature of the edge is high. Computing the edge curvature, however, leads to a scale problem, concerning the scale at which to define descriptors such as curvature extrema. Whereas we do not argue that such purely edge based approaches should not be used, we argue that the potential in computing breakpoint descriptors directly from the grey-level information should be higher. One major reason for this is that a corner detector operating directly on the image data has access to much more information (the entire grey-level pattern). Another as important reason, and as will be further emphasized in section 6, is that in applications such as object detection, explicit computation of junction candidates from grey-level patterns will nevertheless be highly useful, as will the matching relations be between edges and junctions. (More generally, one could, of course, consider hybrid methods, in which these two approaches are combined. Such a method would obviously be beneficial in situations when either of the approaches fails as a single cue.)

[3] A detailed description of the algorithmic steps involved is given in appendix A.2.

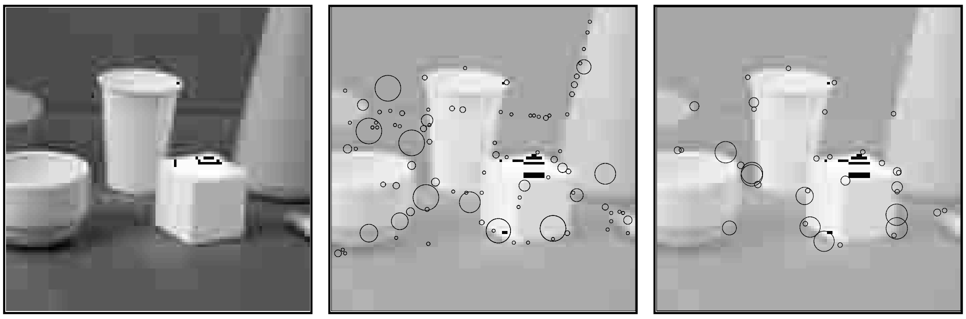


Figure 1: (left) An office scene with geon-type objects. (middle) Junction candidates detected by selecting the 100 scale-space maxima having the strongest (maximal) normalized response. (right) Improved localization estimates obtained by applying a modified Förstner operator to each junction candidate. (From (Lindeberg, 1994a).)

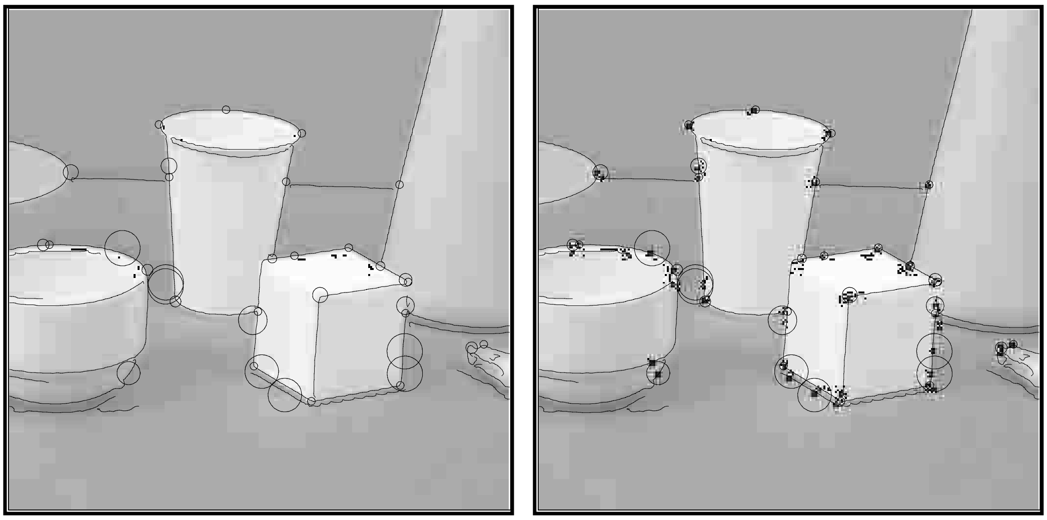


Figure 2: (left) Edges detected by non-maximum suppression with junction candidates overlayed. (right) Matched edge-junction pairs illustrated by squares centered at the edge point closest to the corresponding junction.

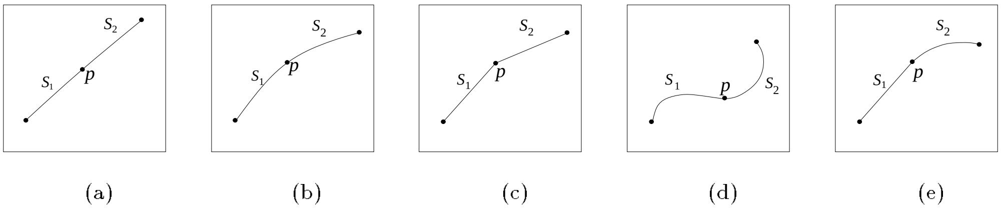


Figure 3: Five examples of splitting or merging of edge segments. (a) S_1 and S_2 are co-linear and belong to the same segment. (b) S_1 and S_2 co-curvilinear and belong to the same curved segment. (c) S_1 and S_2 are two non-collinear straight segments and P is a corner point. (d) S_1 and S_2 are two curved segments that are not co-curvilinear. Most likely, P is an inflexion point. (e) S_1 is straight, S_2 is curved and P is a transition point from straight to curved.

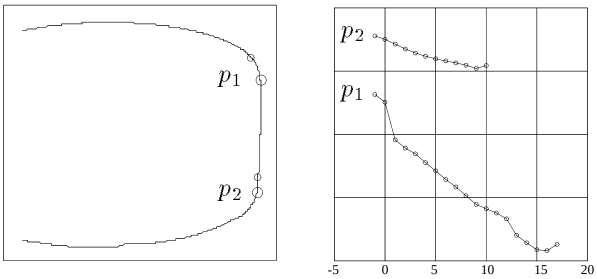


Figure 4: Adjustment of break points. The left figure shows input segments with initial candidate break points illustrated by small circles and new positions marked by larger circles. The new position is obtained by moving the break point along the curve and selecting the break point that minimizes the total description length. The curves in the right figure show how the total description length varies as the break point is moved along the curve.

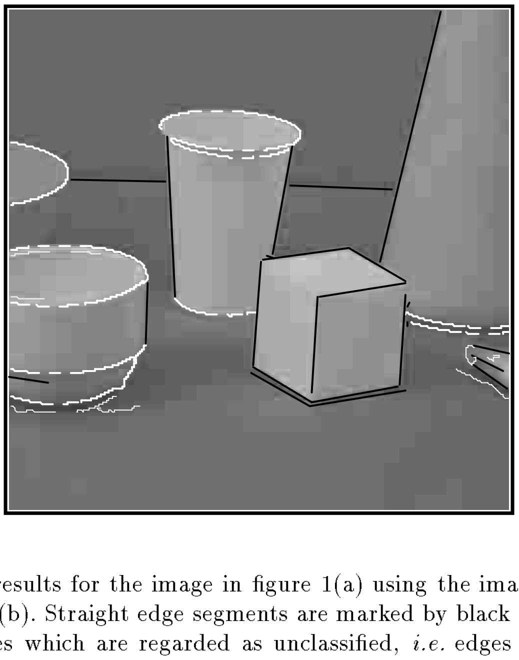


Figure 5: Classification results for the image in figure 1(a) using the image features and candidate break points from figure 2(b). Straight edge segments are marked by black and curved ones by white. Dark curves indicate edges which are regarded as unclassified, *i.e.* edges for which the description lengths for the straight and curved edge models are almost the same (here, differ less than 5 %).

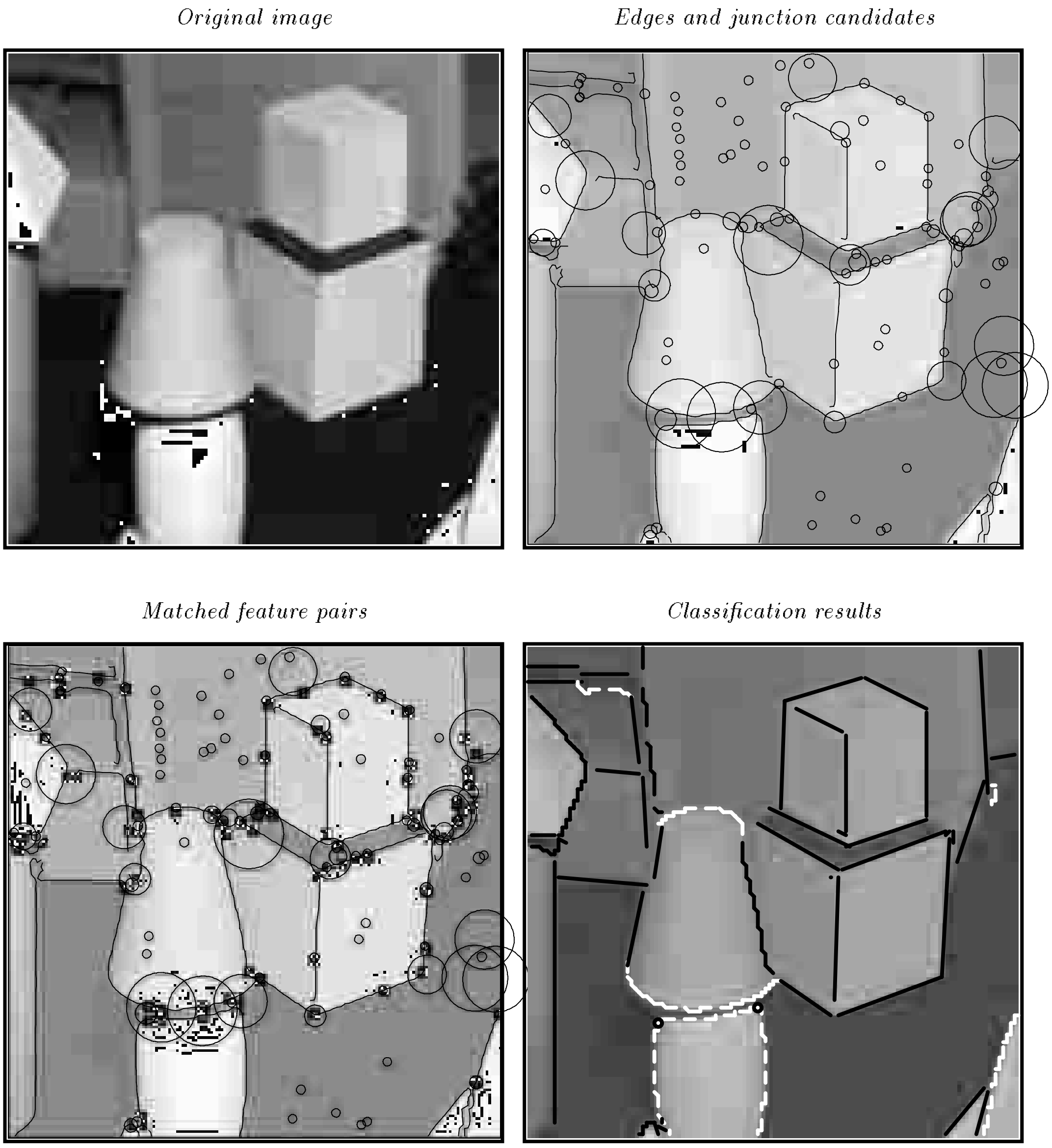
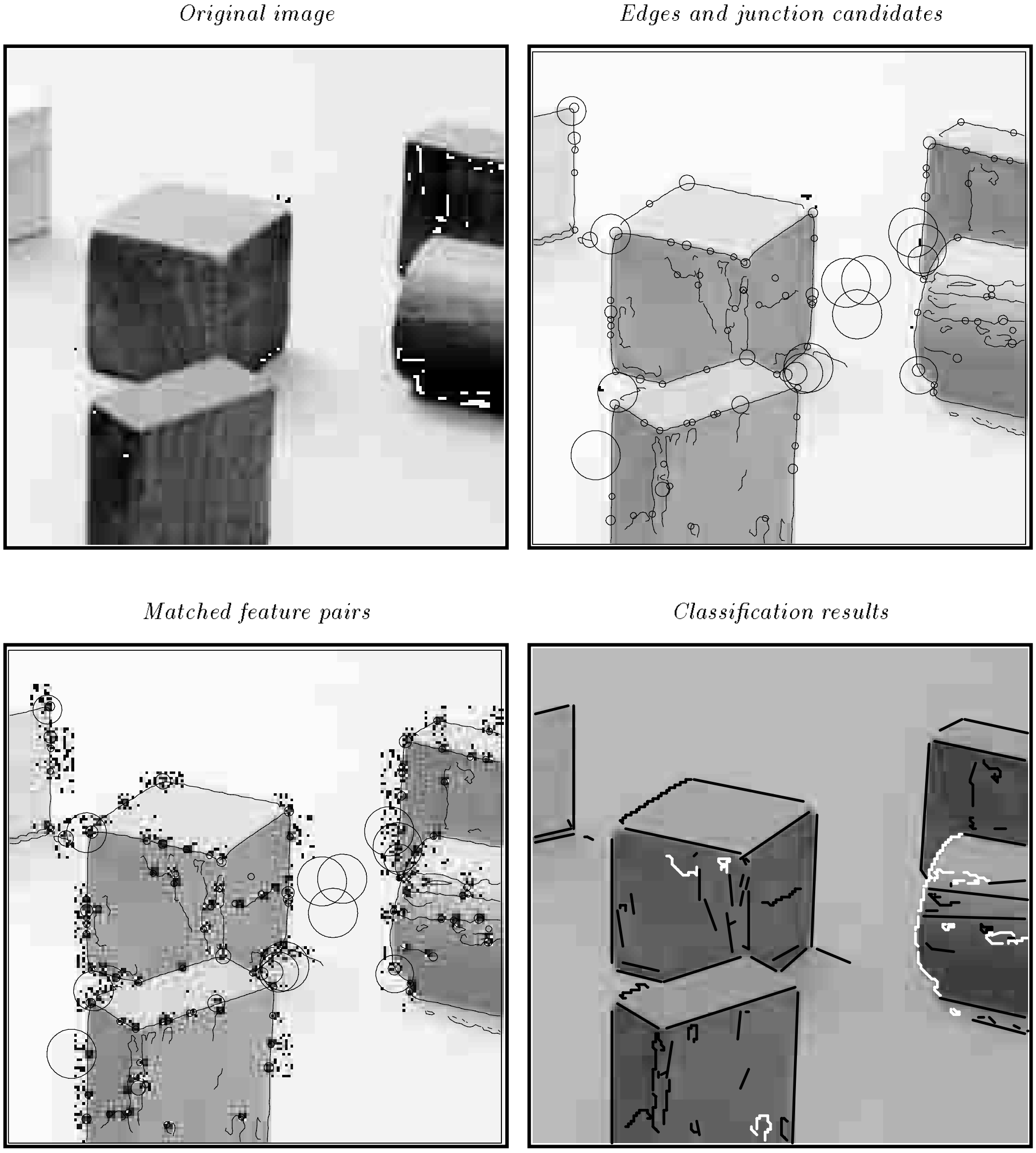


Figure 6: Results of edge classification for a scene with wooden blocks. The images show from top left to bottom right; (a) the original image, (b) edges and junctions detected as described in section 2, (c) break points obtained according to the scheme in section 3, and (d) the final result of the classification. In the last image, the straight edges have been marked by dark lines and the curved edges by bright curves. (Edge segments for which the description lengths for the straight and curved edge models are almost the same (differ less than 5 %) are illustrated as black curves.)



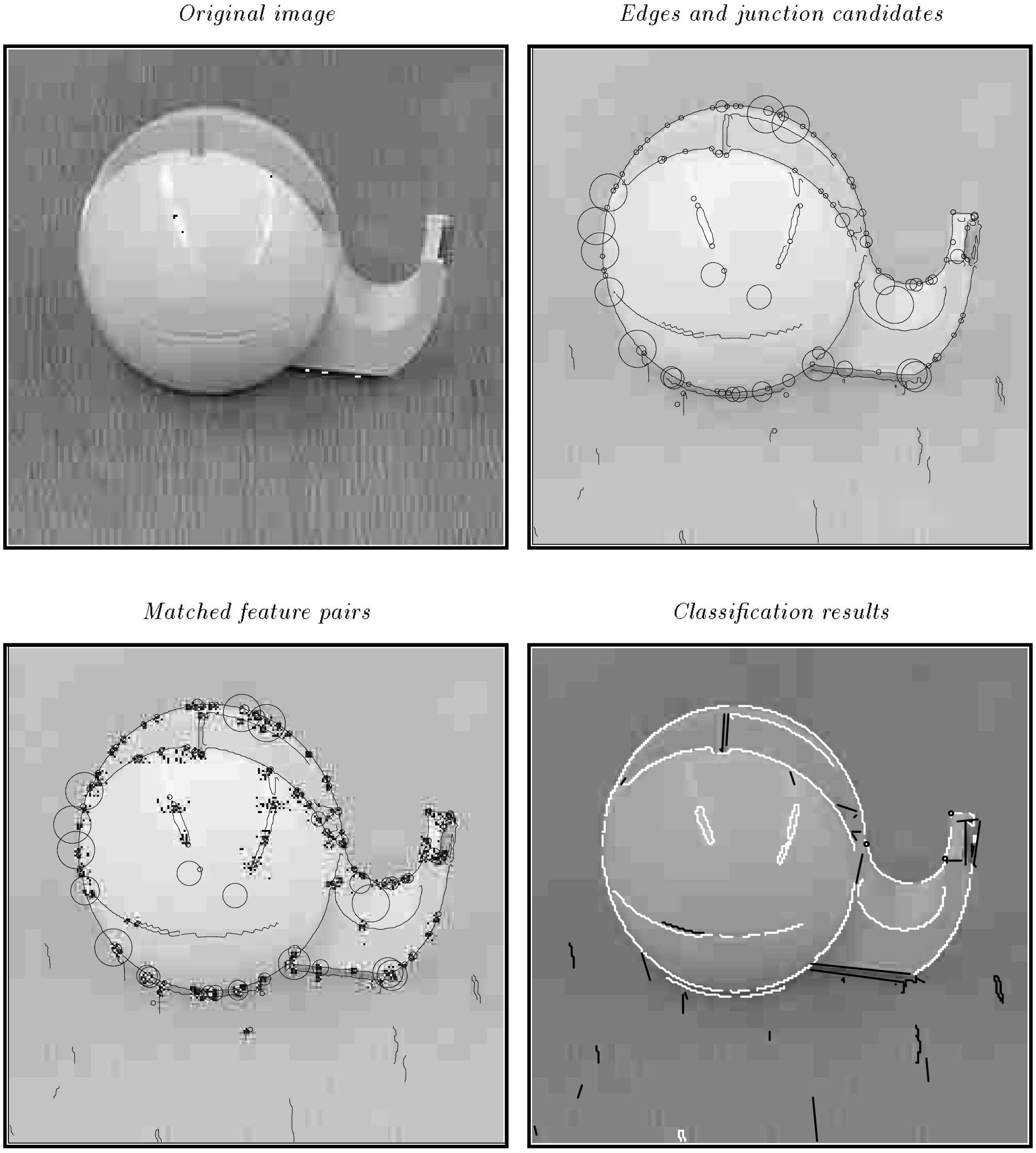
Original image

Edges and junction candidates

Matched feature pairs

Classification results

Figure 7: Corresponding classification results for a scene with textured wooden blocks.



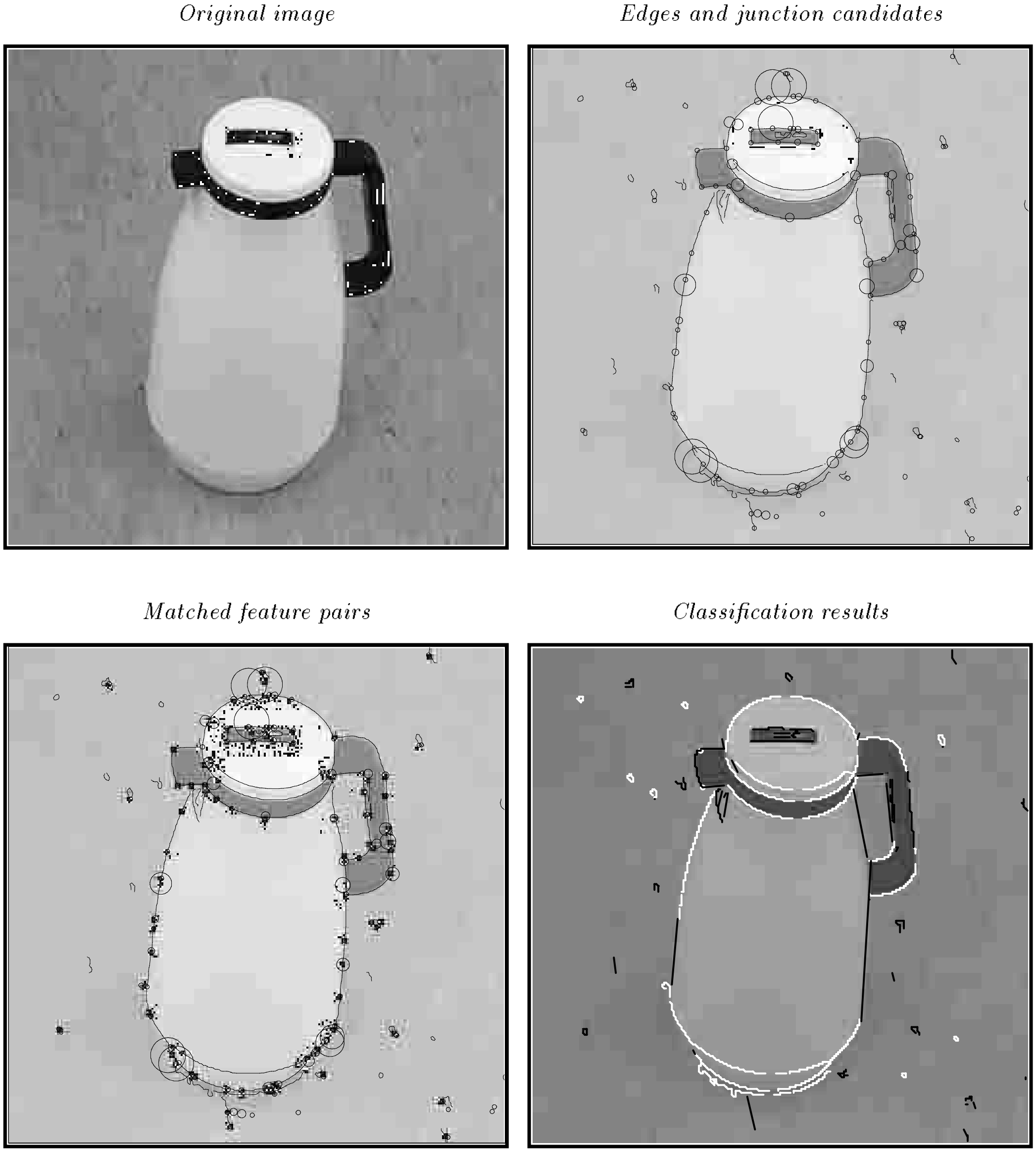
Original image

Edges and junction candidates

Matched feature pairs

Classification results

Figure 8: Corresponding classification results for an image of a tape holder.



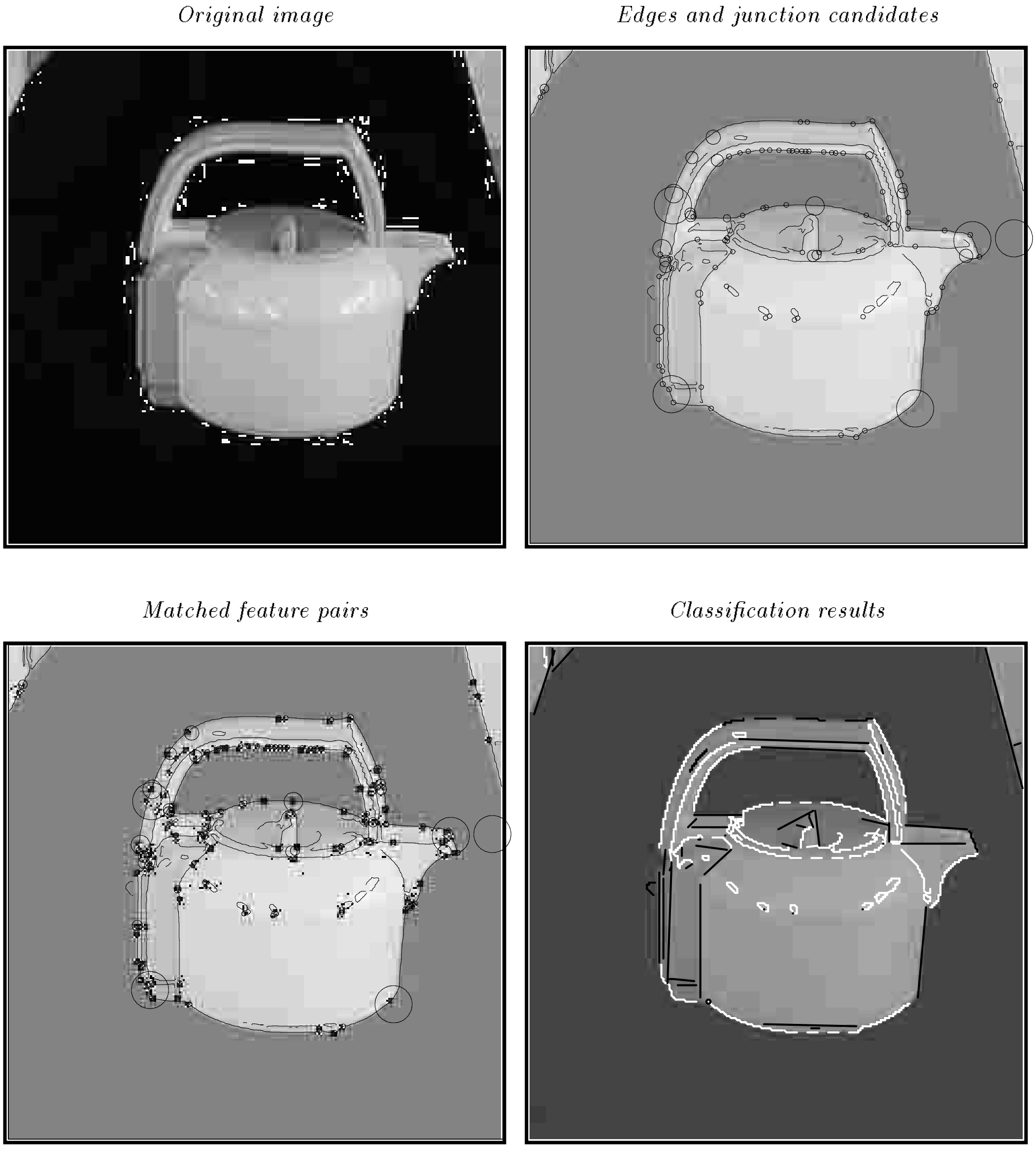
Original image

Edges and junction candidates

Matched feature pairs

Classification results

Figure 9: Corresponding classification results for an image of a coffee thermos.



Original image

Edges and junction candidates

Matched feature pairs

Classification results

Figure 10: Corresponding classification results for an image of an electric kettle.

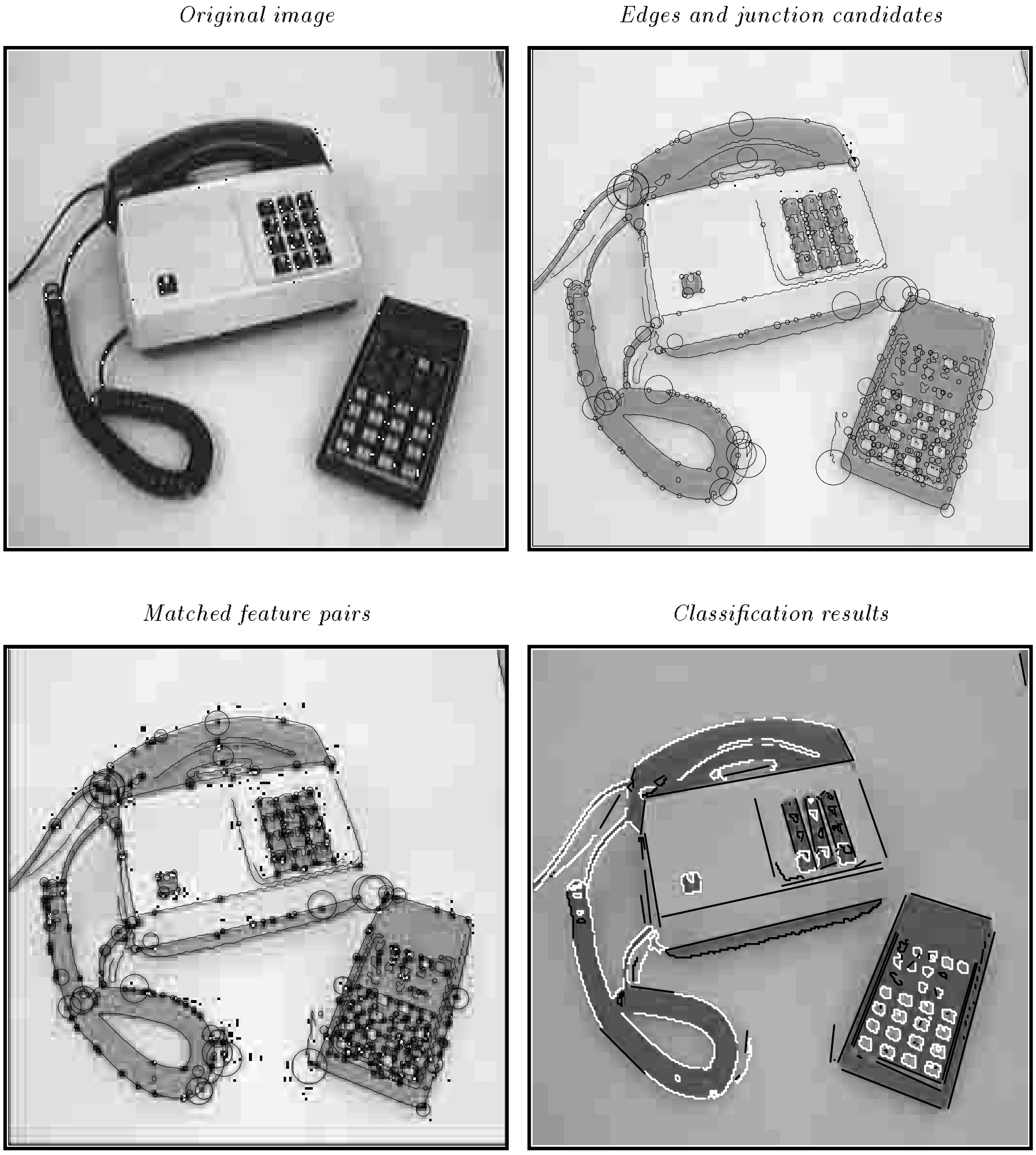


Figure 11: Corresponding classification results for an image of a telephone and a calculator.

6 Summary and Discussion

We have shown how break points for MDL classification can be generated in a straightforward manner by combining edge data with a specific type of junction descriptor associated with a natural region of interest. Instead of operating on the curves as isolated objects and trying to find break points based on, *e.g.*, differential properties such as curvature extrema, these points are obtained by a complementary technique based on junction cues computed directly from the grey-level information. When integrated with edge cues, these junction cues not only reduce the (combinatorial) computational complexity in the algorithm for computing the minimum description length approximation; they also provide important object features for making the segmentation and classification results more robust to noise and outliers. Experimentally, this approach has been demonstrated to give highly useful results on real-world data, especially considering the fact that this low-level processing operates without any access to higher-level information.

Concerning limitations of the work, it has throughout been assumed that the image data originate from man-made objects and that the simple shape classes "straight line", "ellipse segment", and "random points" are sufficient for modelling the data and providing reasonable measures of the description length. For more complex natural scenes, extensions are required, in particular concerning the choice of primitives for measuring the description length.

Another limitation is due to the inherent difficulty in making a final decision between "straight" and "curved" at a low level. (For example, the intuitive judgement of whether a given segment should be regarded as "straight" or "curved" may vary depending on what additional information is available.) When using these data as input to an object recognition system, it is therefore more natural to associate a confidence measure with each feature classification (*e.g.* the ratio between the shortest and the next shortest description length) to allow for final decisions to be made at higher processing levels where more information is available.

Underlying philosophies of the approach. Let us finally remark that whereas this article has been mainly concerned with the rather specific problem of classifying edges as "straight" or "curved" using minimum description length approximation as the decision rule, the main intention has not been to propose specific algorithms, but to illustrate computational principles. There are a number of general ideas underlying this approach, which we argue should be of much wider applicability:

(i) By making complementary use of edges and corners, we have access to a much richer source of information than if basing the analysis on edge cues only. For example, the relations between edge and corner descriptors, possibly combined with associated feature classifications, will be highly useful for problems such as object recognition.

(ii) By using a bottom-up multi-scale preprocessing step to select interesting scale levels, rank image structures on saliency and delimit regions of interest, we can simplify the tasks for further/refined processing. Such local context information can serve as a heuristic guide for reducing the search space for reasoning algorithms and for reducing the combinatorial complexity in evaluating decision criteria, such as those based on the minimum description length paradigm.

A Appendix: Algorithmic details

This section gives a more in-depth description of the major implementation issues.

A.1 Preprocessing: Junction detection, junction localization and edge matching

The combined method for junction detection, junction localization and edge matching in sections 2-3 is based upon the following algorithmic steps (Lindeberg, 1994a, 1994b):

1. *Detection:* Given a discrete image f (here: of size 128×128 or 256×256 pixels), select a scale range for the analysis (here: $t_{min} = 4$ and $t_{max} = 256$). Within this range, distribute a set of scale levels t_k (here: 20 or 40 levels) such that the ratio between successive scale levels t_{k+1}/t_k is approximately constant. (You can choose these levels such that the difference in effective scale (Lindeberg, 1993b) $\tau_{k+1} - \tau_k$ is constant.)
2. For each t_k, compute the scale-space representation of f by convolution with the discrete analogue of the Gaussian kernel T (Lindeberg, 1994c): $L(\cdot;\, t_k) = T(\cdot;\, t_k) * f$.
3. For each point at each scale, compute discrete derivative approximations of $L(\cdot;\, t_k)$ by central differences (Lindeberg, 1993a) and multiply the first- and second-order differences by $\sqrt{t_k}$ and t_k, respectively. (More accurately, you can determine a discrete normalization factor such that the l_1 norm of the corresponding discrete derivative approximation kernel is constant over scales (Lindeberg, 1994c).) Combine these normalized derivatives into discrete approximations to $\tilde{\kappa}^2_{norm}$ at each point using (3).
4. Finally, in the three-dimensional volume generated, detect local maxima (as points whose values are greater than or equal to the values of their 26 discrete neighbours) and select the N (here: 100 or 400) points having the strongest normalized response.
5. *Localization:* For each junction candidate detected in this way, determine an improved localization estimate by computing the following weighted integrals

$$A = \int (\nabla L)\, (\nabla L)^T \, dx', \quad b = \int (\nabla L)\, (\nabla L)^T \, x' \, dx', \quad c = \int x'^T \, (\nabla L)\, (\nabla L)^T \, x' \, dx'.$$

   using a Gaussian window function with (integration) scale value equal to the detection scale and with the center at the candidate junction. At a number of scales (here: 10 levels), uniformly distributed between a lower scale (here: 0.01) and the detection scale, vary the (local) scale at which derivatives are computed and select the (local) scale that minimizes the normalized residual over scales

$$\tilde{d}_{min} = \min_{t \in [0, t_{det}]} \min_{x \in \mathbb{R}^2} \frac{x^T A\, x - 2\, x^T b + c}{\operatorname{trace} A} = \min_{t \in [0, t_{det}]} \frac{c - b^T A^{-1} b}{\operatorname{trace} A}. \tag{15}$$

6. At this scale, the new localization estimate is $\hat{x} = A^{-1} b$.
7. Iterate the localization steps (5–7) until either the increment is sufficiently small (here: within the same pixel) or an upper bound (here: 3 iterations) has been reached. Suppress all points for which the scheme diverges (here: when the total update is larger than the detection scale measured in dimension [length]).
8. *Edge detection.* Concerning the edge detection step, we have throughout this work assumed that edges are given as input. For the experiments presented in this article, edges have been detected at a fixed scale ($t_{edge} = 1.0$), using hysteresis thresholding on the gradient magnitude. Unless otherwise stated, the low and high thresholds ($|\nabla L| > 4.0$ and $|\nabla L| > 8.0$) have been the same for all images.

   Whereas substantial improvements could be obtained by integrating the evaluation of edges with other processing modules, and by including explicit mechanisms for scale selection (Lindeberg, 1995) with locally adapted thresholding operations, we have no aim of contributing to the problem of edge detection in this article.
9. *Edge matching.* Represent each remaining junction candidate with a circle with area equal to the detection scale t_{det}. For each (connected) edge that intersects such a circle, register an edge match at the edge point closest to the localized junction.

When implementing this method in practice, the following observations improve the computational efficiency: On a serial computer, it is not necessary to pre-compute $\tilde{\kappa}^2_{norm}$ at all scales before detecting scale-space maxima; it is sufficient to keep three images in the memory. To reduce the combinatorics in the junction-edge matching, the junctions can be stored in an image-like representation to avoid exhaustive search in a list of junction candidates.

To reduce the computational work further, it can in many situations be sufficient to use more narrow scale ranges than indicated here (*e.g.* $t_{max} = 20$), a smaller number of scale levels (*e.g.* 7 or 10 scales in the detection stage and 2 or 3 levels in the localization stage), and to perform just one iteration in the iterative refinement. The computational efficiency of this processing step is also improved substantially by reducing the spatial sampling density (subsampling the data) at coarser scales in scale-space.

A.2 Minimum description length curve approximation

The input to this curve approximation algorithm is a set of edge segments (represented as lists of points) and a set of candidate break points at which the edge segments may be split into parts. For each edge segment, the algorithm approximates the curve by a straight and a curved model according to section 4.1, considers possible ways of merging edge segments, and outputs a judgement of whether the segment should be regarded as straight or curved. In summary, for each candidate break point, the following operations are performed:

1. *Segment definition.* Include edge points from the current point in the forward and backward directions along the edge segment until either another break point has been reached or an end point is encountered. Denote these two curves by S_1 and S_2.

2. *Local models.* Compute the description lengths for the following models of S_1 and S_2:

   $M_{\overline{ll}}$: S_1 and S_2 are collinear, so S_1 and S_2 belong to the same segment,

   $M_{\frown{cc}}$: S_1 and S_2 co-curvilinear, e.g., they belong to a piece of elliptical segment,

   $M_{\widehat{ll}}$: S_1 and S_2 are two straight segments but not collinear,

   M_{cc} : S_1 and S_2 are two curved segments but they are not co-curvilinear,

   M_{lc} : S_1 is straight and S_2 is curved,

   M_{cl} : S_1 is curved and S_2 is straight,

   In this step, straight line and ellipse models are first fitted to S_1, S_2 and the concatenation of S_1 and S_2 using standard approximation techniques. Then, outliers are removed iteratively in a local greedy fashion, by identifying the point with the highest residual, re-doing the fit without this point, and repeating this removal until the description length does not decrease.

3. *Model selection and merging.* Choose the approximation model with the minimum description length. If the minimum is assumed for $M_{\overline{ll}}$ or $M_{\frown{cc}}$, then merge these edge segments and go to the next break point (step 1). Otherwise, continue with step 4.

4. *Break point refinement.* Move the candidate break point in the forward and backward directions while evaluating the composed approximation models in step 2. Then, split the edge segment at the position at which the minimum description length is assumed.

Finally, allow small edge gaps to be closed and classify the results as follows:

5 *Merging and gap closing.* For each pair of edge segments whose end points are sufficiently close, evaluate the models according to step 2. Merge the segments if the minimum is assumed for $M_{\overline{ll}}$ or $M_{\frown{cc}}$. Otherwise, keep them separate.

6 *Classification.* For each new curve, compute the description lengths for the linear and elliptic approximation models. Classify the segment as straight if the minimum is assumed for the linear model and as curved if it is assumed for the elliptic one.

References

Axelsson, P. (1992). Minimum description length as an estimator with robust properties. In W. Förstner *et al.*, editor, *Proc. of International Workshop on Robust Computer Vision*, pp. 137–150.

Bergevin, R. and Levine, M. D. (1993). Generic object recognition: Building and matching coarse descriptions from line drawings. *IEEE Trans. Pattern Analysis and Machine Intell.*, *15*, no. 1, 19–36.

Biederman, I. (1985). Human image understanding: Recent research and a theory. In *Human and Machine Vision II*, pp. 13–57. Academic Press.

Binford, T. O. (1971). Visual perception by computer. In *IEEE Conference on Systems and Control*, Miami, Florida.

Blom, J. (1992). *Topological and Geometrical Aspects of Image Structure.* PhD thesis, Dept. Med. Phys. Physics, Univ. Utrecht, NL-3508 Utrecht, Netherlands.

Bretzner, L. and Lindeberg, T. (1995). Feature tracking with automatic selection of spatial scales. (Submitted).

Brunnström, K., Lindeberg, T., and Eklundh, J.-O. (1992). Active detection and classification of junctions by foveation with a head-eye system guided by the scale-space primal sketch. In Sandini, G., editor, *Proc. 2nd European Conf. on Computer Vision*, volume 588 of *Lecture Notes in Computer Science*, pp. 701–709, Santa Margherita Ligure, Italy. Springer-Verlag.

Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intell.*, *8*, no. 6, 679–698.

Darell, T., Sclaroff, S., and Pentland, A. (1990). Segmentation by minimum description. In *International Conference on Computer Vision*, pp. 112–116, Osaka, Japan.

Deren, D., Marcus, R., Werman, M., and Peleg, S. (1990). Segmentation by minimum length encoding. In *Proc. 10th Int. Conf. on Pattern Recognition*, pp. 681–683, Atlantic City, N. J.

Deriche, R. and Giraudon, G. (1990). Accurate corner detection: An analytical study. In *Proc. 3rd Int. Conf. on Computer Vision*, pp. 66–70, Osaka, Japan.

Dickinson, S. J., Pentland, A. P., and Rosenfeld, A. (1992). From volumes to views: An approach to 3-D object recognition. *CVGIP: Image Understanding*, *55*, no. 2, 130–154.

Dreschler, L. and Nagel, H.-H. (1982). Volumetric model and 3D-trajectory of a moving car derived from monocular TV-frame sequences of a street scene. *Computer Vision, Graphics, and Image Processing*, *20*, no. 3, 199–228.

Fisher, R. B. (1989). *From Surfaces to Objects.* John Wiley and Sons, Chichester, England.

Florack, L. M. J., ter Haar Romeny, B. M., Koenderink, J. J., and Viergever, M. A. (1992). Scale and the differential structure of images. *Image and Vision Computing*, *10*, no. 6, 376–388.

Förstner, M. A. and Gülch, E. (1987). A fast operator for detection and precise location of distinct points, corners and centers of circular features. In *Proc. Intercommission Workshop of the Int. Soc. for Photogrammetry and Remote Sensing*, Interlaken, Switzerland.

Förstner, W. (1989). Image analysis techniques for digital photogrammetry segmentation by minimum description. In *Proceedings of the 42'nd Photogrammetric Week*, Stuttgart.

Georgeff, M. P. and Wallace, C. S. (1985). A general selection criterion for inductive inference. In O'Shea, T., editor, *Advances in Artificial Intelligence*, pp. 219–229.

Grimson, W. E. L. (1990). *Object Recognition by Computer: The role of geometric constraints.* MIT Press, Cambridge, Massachusetts.

Kitchen, L. and Rosenfeld, A. (1982). Gray-level corner detection. *Pattern Recognition Letters*, *1*, no. 2, 95–102.

Koenderink, J. J. (1984). The structure of images. *Biological Cybernetics*, *50*, 363–370.

Koenderink, J. J. and Richards, W. (1988). Two-dimensional curvature operators. *J. of the Optical Society of America*, *5:7*, 1136–1141.

Leclerc, Y. G. (1989). Constructing simple stable descriptions for image partitioning. *Int. J. of Computer Vision*, *3*, 73–102.

Li, M. (1993). Minimum description length based 2-D shape description. In et. al., H.-H. Nagel, editor, *Proc. 4th Int. Conf. on Computer Vision*, pp. 512–517, Berlin, Germany. IEEE Computer Society Press.

Lindeberg, T. (1993a). Discrete derivative approximations with scale-space properties: A basis for low-level feature extraction. *J. of Mathematical Imaging and Vision*, *3*, no. 4, 349–376.

Lindeberg, T. (1993b). Effective scale: A natural unit for measuring scale-space lifetime. *IEEE Trans. Pattern Analysis and Machine Intell.*, *15*, no. 10, 1068–1074.

Lindeberg, T. (1993c). On scale selection for differential operators. In K. A. Høgdra, B. Braathen, K. Heia, editor, *Proc. 8th Scandinavian Conf. on Image Analysis*, pp. 857–866, Tromsø, Norway.

Norwegian Society for Image Processing and Pattern Recognition.

Lindeberg, T. (1994a). Junction detection with automatic selection of detection scales and localization scales. In *Proc. 1st International Conference on Image Processing*, volume I, pp. 924–928, Austin, Texas. IEEE Computer Society Press.

Lindeberg, T. (1994b). Scale selection for differential operators. Technical Report ISRN KTH/NA/P--94/03--SE, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology. (Submitted).

Lindeberg, T. (1994c). *Scale-Space Theory in Computer Vision*. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Dordrecht, Netherlands.

Lindeberg, T. (1995). Edge detection and ridge detection with automatic scale selection. (Submitted).

Lindeberg, T. and Li, M. (1995a). Segmentation and classification of edges using minimum description length approximation and complementary junction cues. In Borgefors, G., editor, *Proc. 9th Scandinavian Conference on Image Analysis*, pp. 767–776, Uppsala, Sweden. Swedish Society for Automated Image Processing.

Lindeberg, T. and Li, M. (1995b). Segmentation and classification of edges using minimum description length approximation and complementary junction cues. In Borgefors, G., editor, *Theory and Applications of Image Analysis II: Selected Papers from the 9th Scandinavian Conference on Image AnalysisUppsala, Sweden*, Singapore. World Scientific Publishing. (In press).

Lindeberg, T. and Olofsson, G. (1995). In preparation.

Malik, J. (1987). Interpreting line drawings of curved objects. *Int. J. of Computer Vision*, *1*, 73–104.

Noble, J. A. (1988). Finding corners. *Image and Vision Computing*, *6*, no. 2, 121–128.

Requicha, A. G. (1980). Representations for rigid solids: Theory, methods, and systems. *Computing Surveys*, *12*, no. 4, 437–464.

Rissanen, J. (1987). Minimum-description length principle. *Encyclopedia of Statistical Sciences*, *5*, 523–527.

Rohr, K. (1992). Modelling and identification of characteristic intensity variations. *Image and Vision Computing*, , no. 2, 66–76.

Sheinvald, J., Dom, B., Niblack, W., and Banerjee, S. (1992). Detecting parameterized curve segments using MDL and Hough transformation. In *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, pp. 547–552.

Witkin, A. P. (1983). Scale-space filtering. In *Proc. 8th Int. Joint Conf. Art. Intell.*, pp. 1019–1022, Karlsruhe, West Germany.