To appear in: Theory and Applications of Image Processing II – Selected papers from the 9th Scandinavian Conference on Image Analysis, (Uppsala, Sweden), June 1995. World Scientitic, Singapore.

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

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 pre-processing 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.

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 (Biederman, 1985; Dickinson et al., 1992), where the primitives for object modelling are distinguished by such qualitative properties.

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 which is 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, in many cases it is necessary to complement this framework with explicit mechanisms for generating hypotheses about appropriate scales (Lindeberg, 1994c).

A junction detector with automatic scale selection has been developed by Lindeberg (1993c, 1994a). Since the specific properties of this method are important for the proposed methodology, we shall briefly review its most important steps:

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

Gaussian kernels $g\colon \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ 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

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

A commonly used entity for junction detection is the curvature of level curves multiplied by the gradient magnitude raised to some power (Kitchen and Rosenfeld, 1982; Koenderink and Richards, 1988). 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 detecting *scale-space extrema* (simultaneous maxima over scale and space) in $\tilde{\kappa}_{norm}^2$. 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 where maxima are assumed serve as indicators 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.

Whereas this processing 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. Figure 1(c) shows the result of applying this scheme to the data in figure 1(b). (The details of the latter step are described in the abovementioned references.)

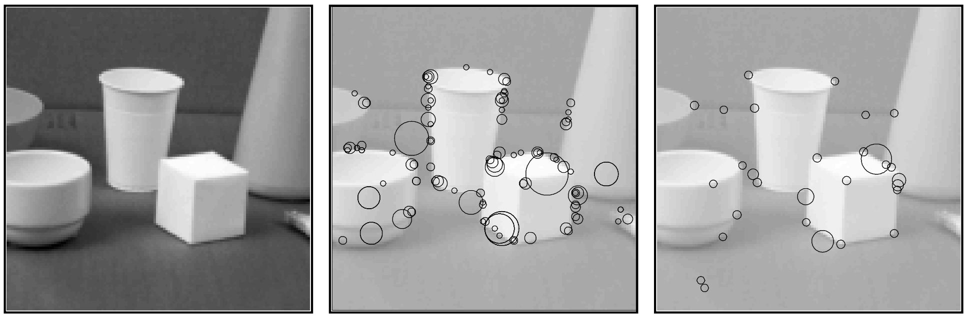


Figure 1: (left) An office scene with geon-type objects. (middle) Junction candidates detected by selecting the 100 scale-space maxima having the strongest (maximal) normalized response. (right) Improved localization estimates.

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 state any explicit threshold on, *e.g.*, the distance.

The left image in figure 2 shows the result of detecting edges at a fine scale ($t = 2$) 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 junction candidate 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.

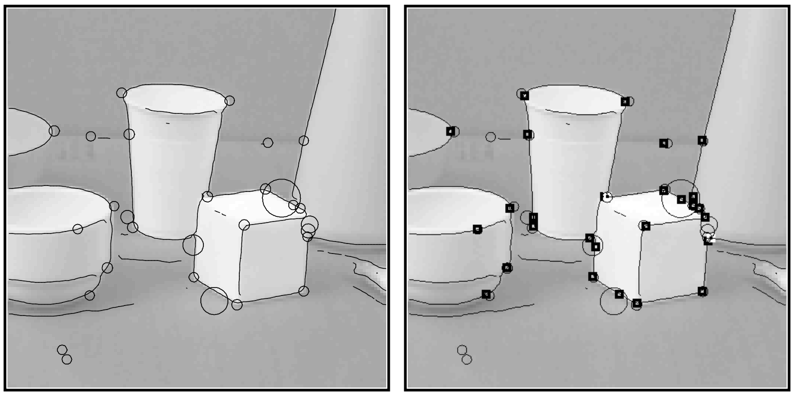


Figure 2: (left) Edges detected by non-maximum suppression with junction candidates overlayed. (right) Matched edge-junction pairs illustrated by squares centered at the edge point closest to the corresponding junction.

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.

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).

In summary, the basic ideas behind the MDL approach are as follows: Let $\phi = (\phi_1, \ldots, \phi_k)$ 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 $x = (x_1, \ldots, x_n)$. 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 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 Representation

In computer vision, the MDL approach has been applied to several problems; see for example (Georgeff and Wallace, 1985; Darell et al., 1990; Axelsson, 1992; Sheinvald et al., 1992). Here, we shall be concerned with a specific scheme for curve approximation developed by Li (1993). It concerns the problem of representing a digital curve using the following models: (i) a set of randomly distributed points, (ii) a linear model with or without outliers, (iii) a section of an ellipse with or without outliers.

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

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

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.)

- $\mathcal{L}_{outliers}(x, \phi) = \lambda\, L_{random\ point}$ specifies the description length of λ points classified as outliers and $L_{random\ point}$ is the description length required to represent a random point. Given two ranges of parameter values, $[x_0, x_0 + \Delta x]$ and $[y_0, y_0 + \Delta y]$ along the two coordinate directions respectively, and a given resolution ε in the data, the latter entity is measured by

$$L_{random\ point} = \log_2 \frac{\Delta x}{\varepsilon} + \log_2 \frac{\Delta y}{\varepsilon}. \tag{8}$$

- $\mathcal{L}_{model\ points}(x, \phi)$ is the description length for the $(N - \lambda)$ points that belong to the model. To capture spatially non-uniform distributions (*e.g.*, gaps) and approximation errors, this description length is measured by

$$\mathcal{L}_{model\ points}(x, \phi) = (N - \lambda)\,[\tfrac{1}{2} L_{random\ point} + L_{offset}(x; \sigma)]. \tag{9}$$

where $L_{random\ point}/2$ represents the (average) amount of information to represent one coordinate, and $L_{offset}(\xi; \sigma)$ is the coding length of the offset error (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

$$L_{offset}(\xi; \sigma) = -\log_2 \left(\Phi([\xi]_\varepsilon + \tfrac{\varepsilon}{2};\ \sigma^2) - \Phi([\xi]_\varepsilon - \tfrac{\varepsilon}{2};\ \sigma^2)\right)$$
$$\approx \log_2 \left(g(\xi;\ \sigma^2)\, \varepsilon\right) = \frac{\log_2 e}{2} \frac{\xi^2}{\sigma^2} + \log_2 \left(\frac{\sigma}{\varepsilon}\right) + \frac{\log_2(2\pi)}{2}, \tag{10}$$

where σ is estimated from the actual deviations between the model and the data (see also Förstner (1989)).

- $\mathcal{L}_{parameters}(x, \phi)$ is the description length of the model parameters.

  For the linear model, four parameters are needed (*e.g.* the four coordinates determining the end points) giving $\mathcal{L}_{parameters}(x, \phi) = 2L_{random\ point}$. The same idea can be applied to the ellipse model described by seven parameters. (Four coordinates representing the center and the semi-axes and three angular descriptors representing the overall orientation as well as the start and the end points).

Concerning the description length of a random point, $L_{random\ point}$, observe that in $\mathcal{L}_{parameters}(x, \phi)$ we measure Δx and Δy in (8) by the global frame of reference (the image size), whereas in $\mathcal{L}_{model\ points}(x, \phi)$, these size descriptors are measured relative to the local frame of reference given by the model parameters (the spatial extent of the curve). For more details, see the abovementioned references.

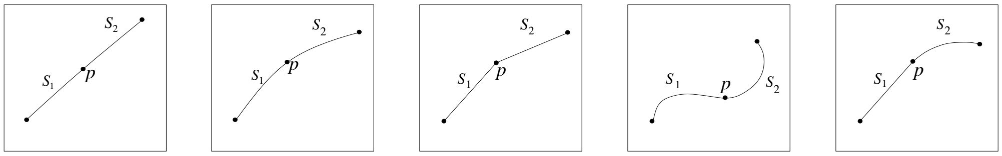
4.2 Algorithm

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.



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

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.

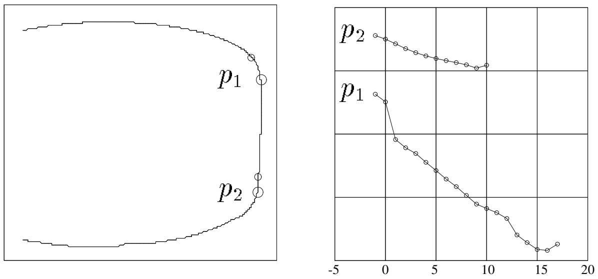


Figure 4: Adjustment of break points. The left figure shows input segments with initial candidate break points illustrated by small circles and new positions marked by larger circles. The new position is obtained by moving the break point along the curve and selecting the break point that minimizes the 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.

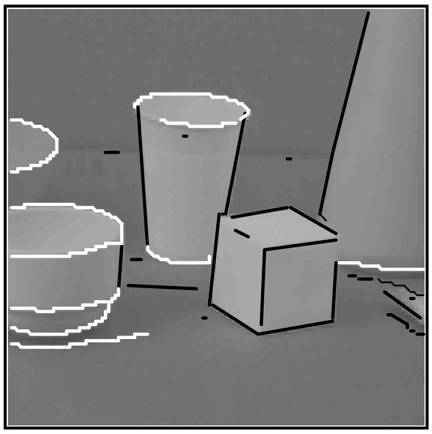


Figure 5: Classification results for the image in figure 1(a) using the image features and candidate break points from figure 2(b). Straight edge segments are marked by black and curved ones by white.

5 Experiments

Figure 5 shows the final result of applying the composed procedure 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 6–7 show corresponding results for two other images of geon-type objects. Notice how very reasonable segmentations and classifications are obtained.

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 more complex natural scenes, extensions are required.

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.

Acknowledgments

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.

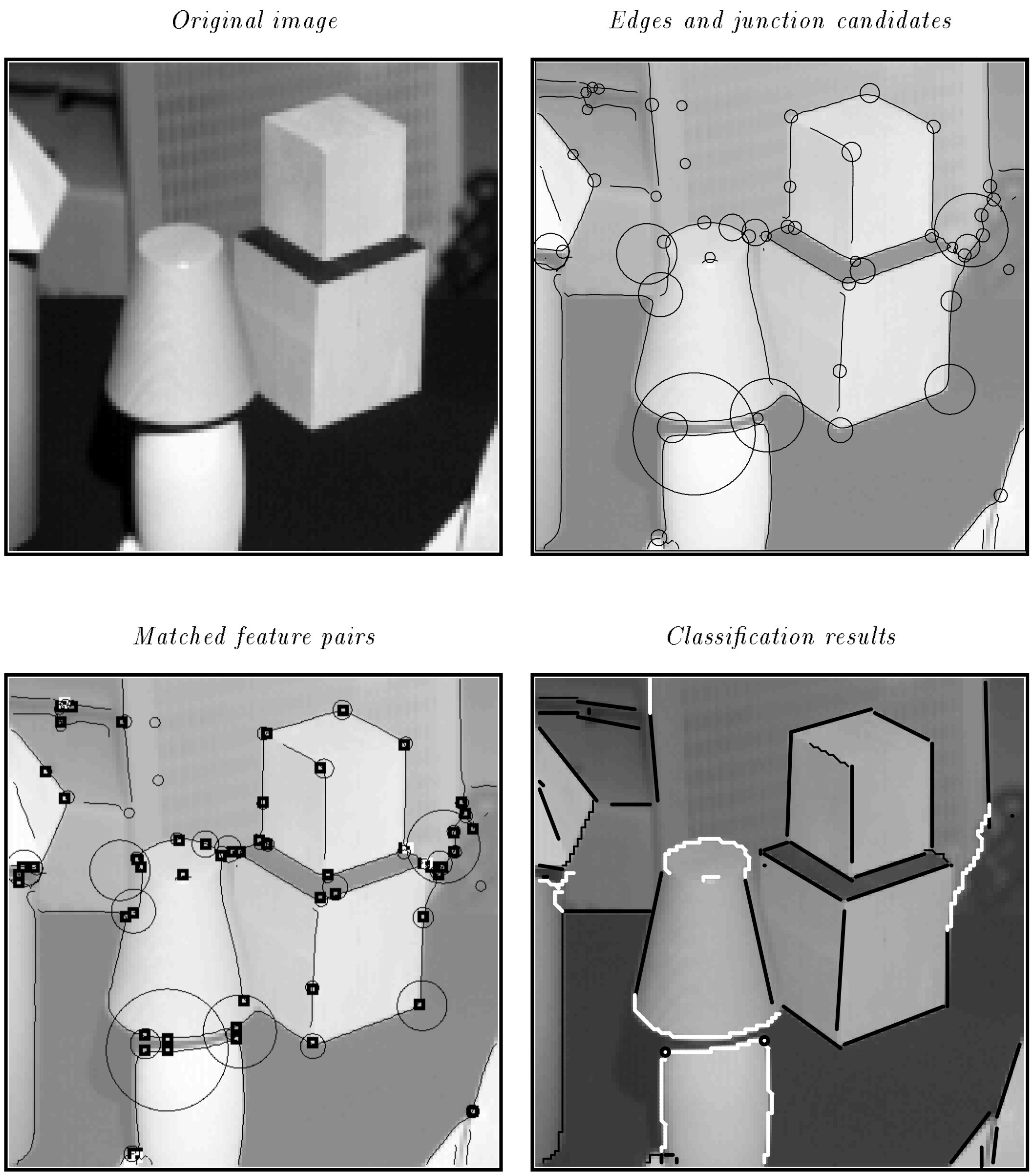


Figure 6: Results of edge classification for a scene with geon-type wooden blocks. The images show from top left to bottom right; the original image, edges and junctions detected as described in section 2, break points obtained by matching edges and junctions according to the scheme in section 3, and 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.

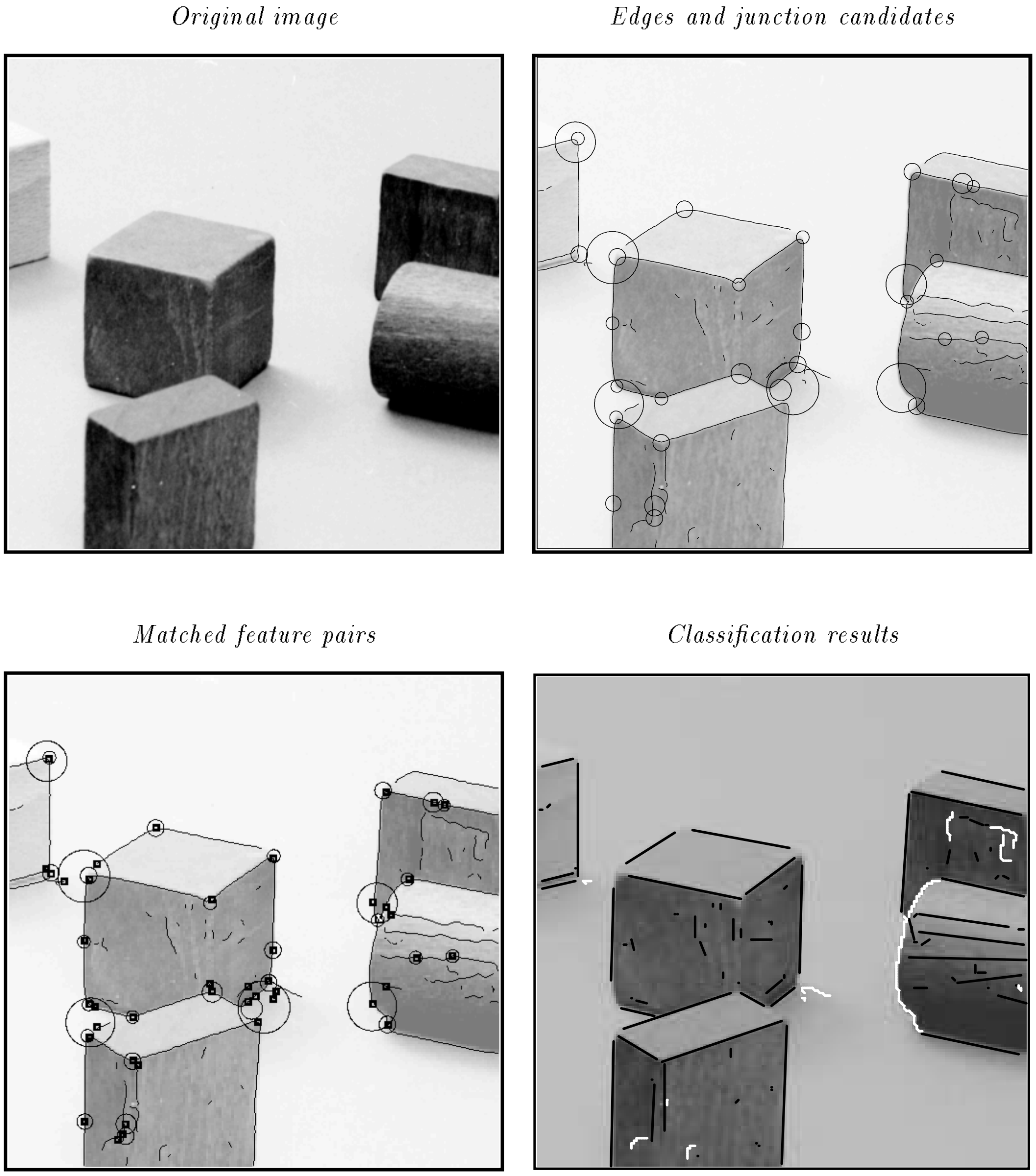


Figure 7: Results of edge classification for a scene with wooden blocks. The images show from top left to bottom right; the original image, edges and junctions detected as described in section 2, break points obtained by matching edges and junctions according to the scheme in section 3, and the final result of the classification. In the last image, the straight edges have been marked by dark lines and the curved edges by bright curves.
(Note that in the graphical illustration of the classification, all points classified as outliers have been suppressed, which results in a loss of connectivity at some rounded corners. Internally, however, the complete and connected representations can be maintained.)

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 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

   $$\tilde{d}_{min} = \min_{t \in [0, t_{det}]} \min_{x \in \mathbb{R}^2} \frac{x^T A\, x - 2\, x^T b + c}{\operatorname{trace} A} = \min_{t \in [0, t_{det}]} \frac{c - b^T A^{-1} b}{\operatorname{trace} A}. \qquad (11)$$

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 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 precompute $\tilde{\kappa}^2_{norm}$ 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 preprocessing 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 and 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 colinear, so S_1 and S_2 belong to the same segment,

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

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

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

   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 fit to 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, redoing 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. (Note that this step includes an implicit penalty on the length of the gap, since $\mathcal{L}_{model\ points}(x, \phi)$ increases with the length of the edge segment (Δx and Δy increase in (8)).)

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 *et al.*, W. Förstner, (ed.), *Proc. of Int. Workshop on Robust Computer Vision*, 137–150.

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

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*, 112–116, 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.

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., (ed.), *Advances in Artificial Intelligence*, 219–229.

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, (ed.), *Proc. 4th Int. Conf. on Computer Vision*, 512–517, Berlin, Germany.

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, (ed.), *Proc. 8th Scandinavian Conf. on Image Analysis*, 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, 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*. Kluwer Academic Publishers, Dordrecht, Netherlands.

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

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*, 547–552.

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