August 1988

Grouping Straight Line Segments in Real Images

Hervé Poilvé
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-88-67.

Grouping Straight Line Segments in Real Images

Abstract

In this paper, we discuss straight line extraction as a part of the image interpretation process. Favoring the use of line drawings as intermediate data for the extraction, we survey the current methods, which all achieve a polygonal approximation of lines, and show that they are not appropriate for the identification of straight elements in a scene. We propose a new approach which uses a scale invariant criterion and is based on the characterization of prime segments in a line, and develop an original method for obtaining these prime segments. Results show that we significantly improve the performance of straight line extraction. The methodology we have used here is applicable to a large class of segmentation problems.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-88-67.

GROUPING STRAIGHT LINE SEGMENTS IN REAL IMAGES

Herve Poilve

MS-CIS-88-67
GRASP LAB 152

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

August 1988

Acknowledgements: This research was supported in part by DARPA grant NOOO14-85-K-0807, NSF grants MCS-8219196-CER, IRI84-10413-AO2 and U.S. Army grants DAA29-84-K-0061, DAA29-84-9-0027.

Grouping straight line segments in real images*

Hervé Poilvé
Computer and Information Science
University of Pennsylvania

Abstract

In this paper, we discuss straight line extraction as a part of the image interpretation process. Favoring the use of line drawings as intermediate data for the extraction, we survey the current methods, which all achieve a polygonal approximation of lines, and show that they are not appropriate for the identification of straight elements in a scene. We propose a new approach which uses a scale invariant criterion and is based on the characterization of prime segments in a line, and develop an original method for obtaining these prime segments. Results show that we significantly improve the performance of straight line extraction. The methodology we have used here is applicable to a large class of segmentation problems.

1 Introduction

Extracting straight lines has become a classic processing step in an image understanding system, and the reasons for that are quite obvious. First, straight lines are mostly related to the human-made environment. When extracting straight lines, we directly select image elements that are likely to be the most useful for an immediate image interpretation : roads, buildings, etc. Considering how arduous the further steps of the image understanding problem are, this is somewhat remarkable.

A second reason is that straight lines are the simplest model of geometric structure. The extraction cost of straight lines is moderate and, from these lines, the restoration of complex structures is manageable. Rectilinear structures are recovered by grouping straight segments that show simple geometric relationships; rectilinear structures form direct candidates for a class of usual scene elements. More generally, the recovery of closed straight contours defines polygonal elements that serve as an intermediate level for the interpretation process. Further tools are available that propose a classification of junctions

*This work is partially supported by DARPA/ONR research contract No. N00014-85-K-0807.

and which produce hypotheses for a three dimensional arrangement of these elements. Let us also mention the perspectivity analysis, that recovers partial organization of the scene from the finding of vanishing points; these points identify the directions of the most noticeable parallel structures. Indeed, powerful techniques for image interpretation may be derived from a straight line model.

Still, the present implementations of these techniques in image understanding systems are somewhat disappointing. They give very partial results: only a small fraction of the structures we know to be described by a straight line model in the image are effectively recovered.

An immediate explanation is the inherent imperfection of the straight line extraction process which produces broken lines and rarely restores any complete polygonal figure. So to improve the detection performance, recent work has promoted the direct extraction of straight edges in the original image. But we somewhat question the relevance of this work to the present issue and we argue that we are facing more of a *modeling* problem than a *detection* problem. Scene objects are rarely simple polyhedral objects. Their description is so complex that we would like to handle only the main frame of the objects in the first steps of the interpretation process. For the considered objects we suppose that this main frame is a straight one. The search for rectilinear structures is indeed the search for these main frames. But there are many details of these objects, such as rounded corners, substructures, and additional parts, that pertain to successive levels of complexity of the description. So, the main frame cannot at once be recovered in full simply because there is no way that a straight line approximation "absorbs" these details.

Until now there has been no study of a systematic reconstruction of the successive description levels.[1] Many authors have tried to use geometric evidence as a way to recover linear constructions from fragmented data. Reynolds and Beveridge [7] proposed, for the identification of rectilinear structures, to test geometric properties such as *spatially proximate* collinearity, parallelism and orthogonality. Mohan and Nevatia [4] proposed the finding of intermediate U structures, restoring three sides of rectangles. Herman and Kanade [2] chose to extend the "legs" of the junctions so as to meet related junctions. These attempts have been only partially successful because these strategies produce numerous ambiguous line relationships that we are unable to resolve. We make the following preliminary observations:

1. It appears that the geometric clues alone are not selective enough. Using as soon as possible a *composite* description, i.e. a description that is not only founded on contour geometry, is likely to greatly improve the pruning of multiple hypotheses.

2. Most of the attempts for recovering linear structures work on a set of iso-

[1] The proposed hierarchical description shouldn't be confused with another classic decomposition: volumes, plane surfaces, segments, vertices, etc.

lated straight segments, and perform the search for geometric relationship in an exhaustive way. This is inefficient. We need an adjacency graph on the straight segments map, and also need to develop heuristic searches on this graph.

3. A large amount of segments are processed in parallel while we look for a few meaningful structures. We need an evaluation of the meaningfulness of the straight line segments, so that we can focus the search around the major straight ones.

Considering observations (1) and (2), we concluded that it is profitable to perform the straight line extraction in two steps: first, apply a classic processing that produces a line drawing, then extract straight segments from this line drawing. The underlying line drawing provides a natural adjacency graph for the extracted segments. The line drawing structure describes some perceptual organization in the image. The following analysis of this image organization may be performed in parallel with the search for rectilinear elements because they share the same support. We thus take advantage of a composite description.

We have reviewed the methods that are currently used for extracting straight lines from line drawings. These methods were designed for producing a piecewise linear approximation of lines. But, we found that they don't provide a relevant analysis of the straight lines' meaningfulness and as a result they are not appropriate to the present problem. The objective of this paper is to develop this point — to show that we address here a specific problem that we call *straight line identification* — and to propose a complete new method that is based on a proper evaluation of the straight segments.

We begin by briefly describing the piecewise linear approximation problem and the related algorithms that were proposed by Ramer [6] and Pavlidis [5] (section 2). Then we will present the purposes of straight line identification and show how they differ from the previous ones (section 3). This analysis will lead us to define a new criterion for evaluating the straightness of the candidate segments (section 4). In the next section we develop a method in which we fit maximum straight segments on a line, and the resulting construction will be called a *tessellation* (section 5). We further describe an efficient algorithm for the implementation of the proposed method (section 6). Finally we present the results and discussion (section 7).

2 Piecewise linear approximation methods

Early work in picture processing brought up the question of having an efficient line drawing representation which compacts data and provides a handy description. With various formulations to minimize the number of vertices or the approximation error along the line, they approximated a line contour by a polygon. In fact, suboptimal methods that led to reasonable computations were preferred.

For evaluating the fit between a line segment and its straight line approximation various error norms were tested. We will detail some of them and introduce the following notation:

Let $\mathcal{L} = \{\vec{p_i}\}_{0\leq i\leq N}$ be a line of connected points. $\mathcal{S}_{ab} = \{\vec{p_i}\}_{a\leq i\leq b}$ is a segment of the line, and $\mathcal{T} = \{\mathcal{S}_{ab}\}_{0\leq a<b\leq N}$ is the set of all segments of the line.

- Ramer proposed a simple and efficient algorithm that achieves a recursive splitting of the line, for a given bound Δ_0 of the approximation error. It uses a maximum error norm:

  $$e_m(\mathcal{S}_{ab}) = \max_{a\leq i\leq b} \|(\vec{p}_i - \vec{p}_a) \wedge \vec{u}_{ab}\| \quad with \quad \vec{u}_{ab} = \frac{\vec{p}_b - \vec{p}_a}{\|\vec{p}_b - \vec{p}_a\|}$$

  $\vec{u}_{ab}$ is the unit vector of the straight line defined by the endpoints of the segment.

  If the approximation error is higher that the bound Δ_0, then the segment is split at the point of maximum deviation, and the same test is performed on the resulting subsegments. The positioning of the partition points proves to be satisfactory.

- Pavlidis proposed a split-and-merge algorithm that makes use of an initial partition of the line, inserts other breakpoints, then processes the merges as long as they satisfy the fit condition Δ_0. Eventually, it may adjust some breakpoints. It applies a mean square error norm:

  $$e_2(\mathcal{S}_{ab}) = \min_{\vec{u},\|\vec{u}\|=1} \sqrt{\frac{\sum_{i=a}^{b} \|(\vec{p}_i - \overline{\vec{p}_{ab}}) \wedge \vec{u}\|^2}{(b-a+1)}} = \sqrt{\frac{\sum_{i=a}^{b} \|(\vec{p}_i - \overline{\vec{p}_{ab}}) \wedge \vec{u}^{\star}_{ab}\|^2}{(b-a+1)}}$$

  where $\overline{\vec{p}_{ab}}$ is the center of gravity of the line:

  $$\overline{\vec{p}_{ab}} = \frac{\sum_{i=a}^{b} \vec{p}_i}{(b-a+1)}$$

  The choice of this error norm offers two advantages. First, it considers the finding of an optimal straight line, defined by $(\overline{\vec{p}_{ab}}, \vec{u}^{\star}_{ab})$, and gives explicit formulae for computing this line. Second, these formulae are expressed in terms of the first moment and the moment of inertia; thus the merging of line segments calculation is efficient because the moments are additive data.

3 The straight line identification problem

3.1 Description

A straight line world is a convenient abstraction; numerous geometric tools for image interpretation have been developed for it. We would like to know which parts of the line drawing may be *identified* as straight lines, and thus may benefit from these tools. In this problem of straight line identification, we want to recover effective straight features of the scene, so that we can then apply object reasonning such as the finding of vertices, junctions, plane surfaces, volumes, etc.

Let's point out the difference between the straight line identification problem and a piecewise linear approximation. We don't expect the full line to be partitioned into straight line segments, and there is no point in having a global optimization over the line. On the other hand, the identified straight segments carry more semantic information. As an example, if we found two straight segments that adjoin but are not congruent, they identify a corner in the scene.

There is much uncertainty in the decison of marking off a part of the line as being related to the presence of a straight element in the scene. Our only evidence is the geometry of the line drawing. Actually, even originally perfect straight lines have to be somewhat recovered because the intrusion of noise throughout the successive processings of the picture deteriorated these lines. The proposed approach to this problem is to merely state that the straighter the segments in the line drawing, the more likely they stand for straight elements in the scene. This is the minimal assumption in the absence of any information.

Thus, we need to define a criterion for evaluating the straightness of a segment, and will show now a distinctive property of this criterion.

3.2 A property of scale invariance

Up to now, we have refered to straightness as a somewhat intuitive notion. The intuition suggests that we consider the relative proportions of the line, i.e. the straightness evaluation is scale invariant. This is supported by a strong argument that if we want straightness to provide a relevant characterization and we want to use it as a clue for extracting some feature elements in the scene, then the same element should be given *as much as possible* the same straightness value whatever the scene viewpoint. The scale invariance ensures some viewpoint independence because the straightness value will not vary with the distance. But this still remains an approximation, for there is no way to retrieve the effects of foreshortening: foreshortening in the segment's main direction will decrease the apparent straightness whereas foreshortening in its normal plane will increase it. Anyway, we are forced to seek a criterion that is scale invariant and to reject the fit criteria that were defined for the piecewise linear approximation problem. This definitely shows that we here address a distinct problem.

The scale invariance is likely to increase the difficulty of the search for straight segments since we may find segments that qualify for straightness at any scale. Weiss and Boldt [8], in a quite related problem, invoked the same scale invariance constraint and proposed a method that discretizes the scale range in a hierarchy of scale levels as a way to perform the search. We will present here a method that makes it possible to avoid such a discretization.

Obviously, the ability to recover straight segments at various scales is somewhat bounded by the effects of the pixel quantization. We will study further these limits.

4 Definition of a straightness criterion

4.1 Description of the criterion

We chose to define our criterion as the ratio of two homothetic functions of the segment geometry, thus respecting the constraint of scale invariance. These two functions roughly describe the approximation error and the elongation of the segment line.

The error is computed according to a least square error calculus. As described in section 2, this calculation is set up as a minimization problem that silmutaneously gives the mean error and the infinite straight line on which to locate the identified straight segment. For bounding this segment, we chose to project onto this line the end points of the original segment and compute the elongation as the distance between these projected points.

$$c_s(\mathcal{S}_{ab}) = \frac{e_2(\mathcal{S}_{ab})}{l(\mathcal{S}_{ab})} \quad where \quad l(\mathcal{S}_{ab}) = |(\vec{p}_b - \vec{p}_a) \cdot \vec{u}^{\star}_{ab}|$$

The definition of the criterion is quite natural, and it has proved to behave well. We will now present two important points that slightly modify the criterion formulation.

4.2 The effects of pixel quantization

It appears that if we apply our criterion to the smallest of all segments — those defined between two connected pixels — they are found to be perfectly straight. This is obviously an artefact of the pixel quantization. The quantization suppresses the subpixel agitation of the line while it introduces some location error for the points sampled along the line. It is most difficult indeed to propose a model of the quantization process, which is the transformation of virtual continuous lines in pixels, as a unique model for the various algorithms that produce line drawings. As a general rule, we are not able to retrieve the quantization effects that are highly discontinuous and anisotropic. A simple and quite effective solution is to assign a minimal "thickness" to any line. For example, in a 4-connected representation, the horizontal and vertical lines are not favored against diagonal lines that have a staircase profile. That is:

$$c_s(\mathcal{S}_{ab}) = \frac{e'_2(\mathcal{S}_{ab})}{l(\mathcal{S}_{ab})} \quad \textit{with} \quad e'_2(\mathcal{S}_{ab}) = \max(e_2(\mathcal{S}_{ab}), \varepsilon_0)$$

Considering the worst case, we chose $\varepsilon_0 = 0.5$.

If now we test, using the criterion, the ability to detect the same straight feature at various scales, we can verify the limits imposed by the pixel quantization. For example, suppose we are looking for segments that attain a 0.025 straightness ratio; these segments should be at least 20. long. No smaller segment can be recognized as significantly straight. Yet that allows for quite a large range of elongation (> 30) in a 512 by 512 image.

4.3 Taking into account the imprecision of line drawings

Actually, there are other artefacts that may affect the straightness evaluation. The various algorithms that produce line drawings show very different behavior: while a Canny edge detector will render noisy weak edges by giving wandering or disrupted lines, a region growing process may draw perfect straight lines while bounding poorly discernible regions. We still haven't looked at how the performance of the line drawing process affects the validity of the results of the straightness calculation, but we obviously first rely on this process for giving meaningful lines (i.e. lines that may be supported by some physical interpretation). Next, we must consider that the positioning of these lines is not perfect.

We propose to estimate the location uncertainty of the lines while producing the line drawing and keeping track of this information. One difficulty is that in the algorithms that are currently used, the location uncertainty is not apparent and is not easily derived[2], and bounding the positioning error may not be appropriate as this error is likely to vary a great deal along the lines.

Having an estimation of the location uncertainty, we then consider that this uncertainty decreases the ability to recover straight segments in the image. We will integrate this information in the straightness calculation by expressing some local thickness of the line. Let $\vec{\sigma}_i$ be the variance estimation along the local normal to the line we suppose we have identified, we define then the following error:

$$e_2(\mathcal{S}_{ab}) = \min_{\vec{u}} \sqrt{\frac{\sum_{i=a}^{b} \|(\vec{p}_i - \overline{\vec{p}_{ab}}) \wedge \vec{u}\|^2 + \sum_{i=a}^{b} \|\vec{\sigma}_i \wedge \vec{u}|^2}{(b-a+1)}}$$

[2]There is some relation with the local significance or strength of the line, that is a quite simple amplitude measure (gradient magnitude, etc.), but not any strict dependency.

5 Building a straight segments tessellation

We are now given the following problem. We have defined a criterion for evaluating the straightness of a segment and have assumed some monotonic relation with a likelihood function. We want to extract segments that have a good chance to reveal straight features in the scene. So, we choose to apply a confidence threshold on straightness values for our segment selection; the choice of the threshold is the result of a tradeoff between straight features of the scene to be missed and wrong segments to be selected. The issue then is to produce a *partial* description of the line built on segments that qualify for straightness.

The first experience is that wherever we find any valid straight segment, we are likely to find numerous other valid ones that all intersect with it. We must use some principle of exclusion. We propose the following definition: $\mathcal{S}$ is a *prime segment* if

$$\forall \mathcal{S}' \in \mathcal{T},\ \mathcal{S} \cap \mathcal{S}' \neq \emptyset \implies c_s(\mathcal{S}') > c_s(\mathcal{S})$$

These segments achieve some local maximum of straightness. But some of them may be not significantly straight. Also, notice that this definition doesn't ensure that the line is fully covered by the set of prime segments[3], but that won't affect our approach.

First, we propose to extract the prime segments of a line, producing what we call a tessellation by analogy with the 2D process of interlaying little squares like a mosaic work; then to retain only the segments that satisfy the threshold condition. We thus, in some way, achieve a partition where parts are either straight segments or are recognized not to be described by a straight line model. The parts not described by the straight line model are left aside.

In an overview of the line partitioning problem, Fischler and Bolles [1] proposed the two principles of stability, and complete, concise and complexity-limited explanation as the foundation of any competent method. Regarding these principles we argue that:

- The produced partition is stable while tuning the confidence threshold. This results in the deletion of some straight segments or the creation of new ones. This partition is also stable under small perturbations of the line that produce continous moving of the segments.[4]
- For the explanation of the line, we don't intend to give a complete explanation with this partition, since it uses the sole straight line model. But it defines which parts of the line are related to this model and which are not.

[3] Actually, such cases occur very rarely. Points $\vec{p}_k$ that do not pertain to prime segments are those such that $\forall \mathcal{S} \in \mathcal{T} \,/\, \vec{p}_k \in \mathcal{S},\ \exists \mathcal{S}' \in \mathcal{T} \,/\, (\vec{p}_k \notin \mathcal{S}',\ \mathcal{S} \cap \mathcal{S}' \neq \emptyset,\ \textit{and}\ c_s(\mathcal{S}') < c_s(\mathcal{S}))$.

[4] In some cases, when there are two resolution levels that have quite similar straightness significance, our tessellation, as it inclines towards the best straightness value, may switch to the other level.

We explored other choices for building a tessellation for a straight segments description and found that a main alternative is to choose the longest segments that qualify for a given straightness C_0, that is to define a prime segment $\mathcal{S}$:

$$c_s(\mathcal{S}) < C_0 \ \ and \ \ \forall \mathcal{S}' \in \mathcal{T},\ (\mathcal{S} \cap \mathcal{S}' \neq \emptyset \ and \ c_s(\mathcal{S}') < C_0) \implies l(\mathcal{S}') < l(\mathcal{S})$$

and build the resulting tessellation. Still, whereas the previous tessellation depended only on the geometry of the line, this one is dependent on the confidence threshold (for the setting of which we have few clues for now) and acts as a reduction of the straight line description complexity. For this reason, we think the first proposed tessellation to be a more intrinsic description and more appropriate for the task of identifying straight line features.

6 Algorithm description

After having exposed the formal construction of the tessellation, we present an algorithm for its implementation.

6.1 The algorithm frame

We use the following strategy : we start with an initial partition and then build a pyramidal structure made of successive merges. We will soon show that the choice of the initial partition and the choice of the criterion for ordering the merges are such that the subset of segments formed is likely to contain the prime segments we are looking for. The pyramid built is an unbalanced binary tree. While merging, we compute the straightness value of each new segment, and keep record of the best straightness value that has been found amid all the underlying segments. Having pursued the merging process till it gives a unique segment for the whole line, we are finally able to produce the suggested tessellation in a single pass implementing a depth-first search in the tree down to the point where there is no included segment of better straightness.

6.2 A principle of local evidence

Let us now first explain the criterion defined for ordering the merges. Suppose that we are in the current state of the merging process and we consider the decision of merging a segment with either of its adjacent neighbours. What is the risk? If this segment is to be a prime segment (in the tessellation we suppose we know from elsewhere) or if the three segments pertain to the same prime segment, there is no risk and the merges may be carried out in any order. But if this segment and its left neighbour are parts of the same prime segment while its right neighbour belongs to another prime segment, then the risk is that if we merge it *first* with its right neighbour we won't be able to recover the proper tessellation.

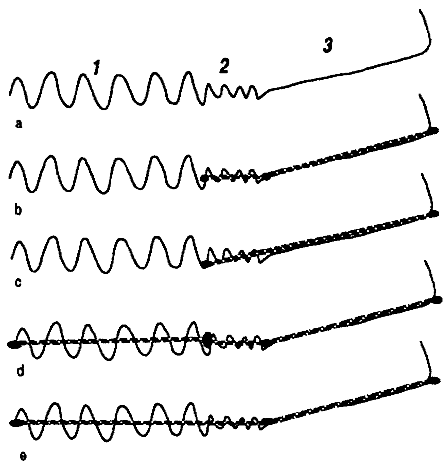

Figure 1: Case contradicting the principle of local evidence
In this case successive noise discontinuities divide the line in three parts. (**a**). When increasing the allowable error for merging, parts 2 and 3 are merged first (**c**). As a result the final tessellation distinguish three segments (**d**), whereas for the optimal partition parts 1 and 2 should be merged (**e**).

We need something for guiding our decision and at this point we must invoke some principle of local evidence. We propose the following: assume that at various resolutions, defined as successive levels of approximation error, the "fracture" between two prime segments provides sufficient evidence for keeping them distinct in the merging process. Cases that contradict this assumption are cases where we locally find two peaks of straightness significance at different resolutions that result in two overlapping partitions; we have built such a counter example (fig. 1), but this is somewhat artificial. So, we choose to rely on this assumption, and we process the merges by continuously increasing the allowable error along the line. The prime segments appear then disappear when included in longer segments. They will be recovered by the depth-first search stage.

Interestingly, the principle of local evidence was also proposed by Lowe [3] for detecting meaningfull groupings in a random field of points as a way to reduce the computationnal complexity, and the authors argued that human vision in similar conditions, makes use of the same principle.

Regarding the choice of the initial partition, the intent is to do a little bit better than merely starting with the elementary partition made of unit segments between connected pixels. For that, we build a partition in which we limit the maximum error norm of the linear approximation to the error level which we know results from pixel quantization. In that way the produced set of straight segments is indeed a valid hypothesis of the line (i.e. it is rendered by the present line in a pixel representation).

6.3 Complexity analysis

Let N be the total number of pixels of the line. The setting of the initial partition gives a number of segments $N' = \alpha N$ (typically, $\alpha = 0.1$). The algorithm then proceeds in successive merge operations. The number of these merges is in the worst case $\sim N'$ (when the resulting tree is fully unbalanced). At each merge step, we have to select the merge with the lower error value and update the the set of remaining merges for the next step. We propose so to sort the candidate merges in a height balanced binary tree, in wich the insertion or the deletion of a merge will cost $O(\log n)$, where n is the number of nodes in the tree. In our case, $0 \leq n \leq N'$. As a conclusion, the complete cost of the algorithm is $O(n \log n)$.

7 Results and discussion

This algorithm has been implemented and tested on various images. We present here a comprehensive analysis of the results obtained from one of these images: an aerial view of the campus (fig. 2).

The segmentation is produced by a region growing algorithm. The region boundaries define a lattice structure, and we apply our algorithm to the lattice branches. For successive values of the straightness threshold, we display the selected segments. Parts of the main frames appear first; indeed, we could expect that the main frames provide the most significant straight segments. Then appear other segments that complete these frames or introduce new levels of complexity of the description. And, the lower the straightness value, the more we find segments that are falsely identified as far as we can tell from the original. (fig. 3).

We then compare these results with those given by applying the Ramer algorithm. In order to make things as equal as possible for the comparison, but according to our standards, we have reevaluated the segments of the Ramer partition with our criterion and displayed the segments that statisfy the same straightness condition (fig. 4 and fig. 5). A detailed inspection of these results shows that our method ensures a better positioning and identifies segments that are approximately recovered when applying the Ramer algorithm, but not for a common value of the error bound choice.

We would like to have some quantitative measure of the performance of our method, relative to the previous ones. We could count the number of extracted segments as a function of the straightness threshold c_t

$$n(c_t) = \sum_{c_s(S_i) < c_t} 1$$

and we display such a graph for the set of prime segments $\{\mathcal{S}p_i\}$ we have extracted (fig. 6); but this doesn't provide an effective evaluation of the other methods: whereas the principle of exclusion we have used ensures that the prime segments identify distinct straight parts of the line, other methods may produce some segments that pertain to the same straight part (i.e. the resulting merge of these segments has a better criterion value than any of its subsegments). This is obviously a faulty partition, as it has broken a straight entity and thus will mislead the interpretation of the line, but there is no "natural" way of penalizing such errors.

We propose to assign a unit weight to the distinct straight parts found from the extracted segments, and a weight x_i to a segment $\mathcal{S}_i$ as the fraction value of the straight part it belongs to. We then express:

- the correctness of the partition as the ratio

$$\frac{\sum_{c_s(\mathcal{S}_i)<c_t} x_i}{\sum_{c_s(\mathcal{S}_i)<c_t} 1}$$

  function of the straightness threshold c_t (by construction, the prime segments set is fully correct).

- the compared performance with the prime segments set as the ratio

$$\frac{\sum_{c_s(\mathcal{S}_i)<c_t} x_i}{\sum_{c_s(\mathcal{S}p_i)<c_t} 1}$$

  function of the straightness threshold c_t.

We applied these two measures for a comparative evaluation of the Ramer method, using successive bounds of the approximation error (fig. 7). Whereas a too large error bound gives poor performance results, the correctness of the partition decreases rapidly when choosing a too small error bound. The best partition that we can obtain with the Ramer method achieves some trade-off between correctness and performance. Evaluating the same way the Pavlidis method, we found very similar results (fig. 8). In both cases, the method we propose significantly increases the extraction performance.

Experiments show that the performance gain varies from one image to another. We explain these variations in the following way. The piecewise linear methods produce a partition of a line for a given error bound of the linear approximation. When we apply these methods for recovering straight elements in the image, we somewhat assume that a constant level of "noise" has been added to perfectly straight lines. For images that are homogeneous, i.e. that effectively show such a constant level of noise, these methods may give good results, provided that we have well estimated the noise level for the setting of the error bound parameter. As soon as we deviate from this case, performances of the piecewise linear methods deteriorate.

In our method, we propose to fit on a line the main significant straight segments, according to the scale invariant criterion, and this makes the extraction fairly insensitive to the noise level, as long as these main elements remain perceptible. Then, using the estimation of the segments straightness, we can first initiate the interpretation on the most reliable ones. And the principle of construction of the partition ensures that it doesn't produce multiple segments for a common straight part. So the principle of construction gives a correct basis for the search for rectilinear structures.

We have also implemented an extension of our algorithm so as to extract not only straight segments that lay on a single branch but also those which spread throughout the nodes of the lattice. In a second pass, starting from the partitions produced on the separate branches, we perform a similar merging process over the whole lattice. We use the same principle of local evidence for selecting the best candidate among the possible merges between adjoining segments at each node of the lattice. We show here some first results of this extension (fig. 9). Though in some cases the local evidence may be misleading, the algorithm extension is fairly efficient and recovers most of the meaningful straight segments that bound multiple regions. This provides new information regarding the structure of the segmented image.

We think the methodology we have developped here for the identification of straight line segments has a wider range of applications and may be used successfully for other problems of one dimensional segmentation that meet the following conditions:

- We are interested in fitting on the line the best segments according to a certain criterion.
- This criterion behaves such that a priori we have to search among segments of all sizes. Typically, the criterion is scale invariant.
- We can assert some principle of local evidence for guiding the process of successive merging.

We are currently exploring this methodology for segmenting a line in second order curves, and also its two dimensional extension for problems of region segmentation.

Acknowledgements

I would like to thank particularly Ruzena Bajcsy for her advice and guidance throughout this research.

References

[1] M.A. Fischler and R.C. Bolles, Perceptual organization and curve partitioning, *IEEE PAMI* **8**, 1986, pp 679-698.

[2] M. Herman and T. Kanade, The 3D MOSAIC scene understanding system: incremental reconstruction of 3D scenes for complex images, in *Proceedings, DARPA Image Understanding Workshop, New Orleans, 1984*, pp 137-148.

[3] D.G. Lowe and T.O. Binford, Segmentation and aggregation: an approach to figure-ground phenomena, *Proceedings, DARPA Image Understanding Workshop, Palo Alto, 1982*, pp 168-198.

[4] R. Mohan and R. Nevatia, Perceptual grouping with applications to 3D shape extraction, *Proceedings, IEEE Computer Society, Workshop on Computer Vision, Miami, 1987*, pp 158-163.

[5] T. Pavlidis, Segmentation of plane curves, *IEEE Transactions on Computers* **C 23**, 1974, pp 860-870.

[6] U. Ramer, An iterative procedure for the polygonal approximation of plane curves, *Computer Graphics and Image Processing* **1**, 1972, pp 244-256.

[7] G. Reynolds and J.R. Beveridge, Searching for geometric structure in images of natural scenes, *Proceedings, DARPA Image Understanding Workshop, Los Angeles, 1987*, pp257-271.

[8] R. Weiss and M. Boldt, Geometric grouping of straight lines, *Proceedings, IEEE Conf. Computer Vision and Pattern Recognition, Miami, 1986*, pp 489-495.

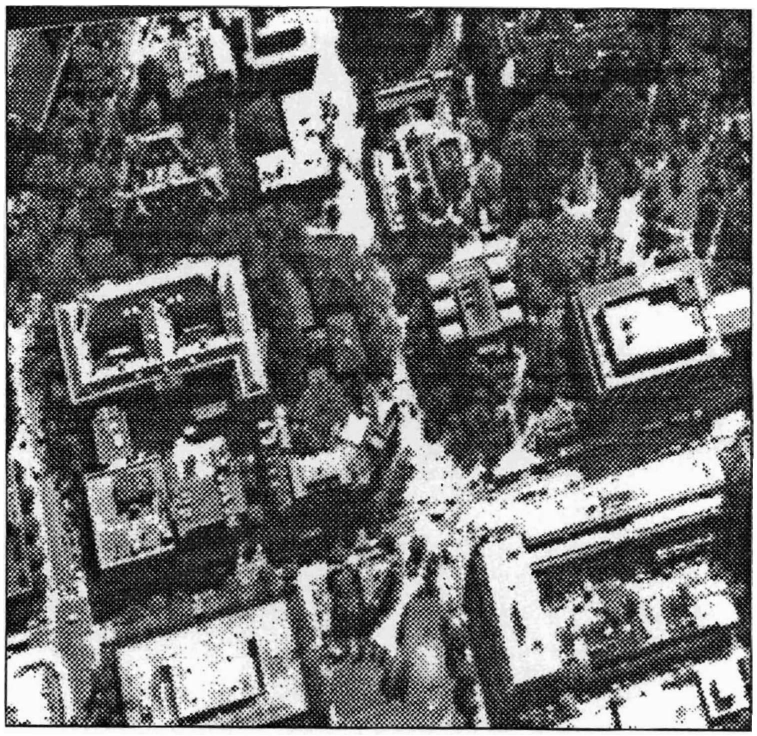

Figure 2: Aerial Photo
University of Pennsylvania campus original

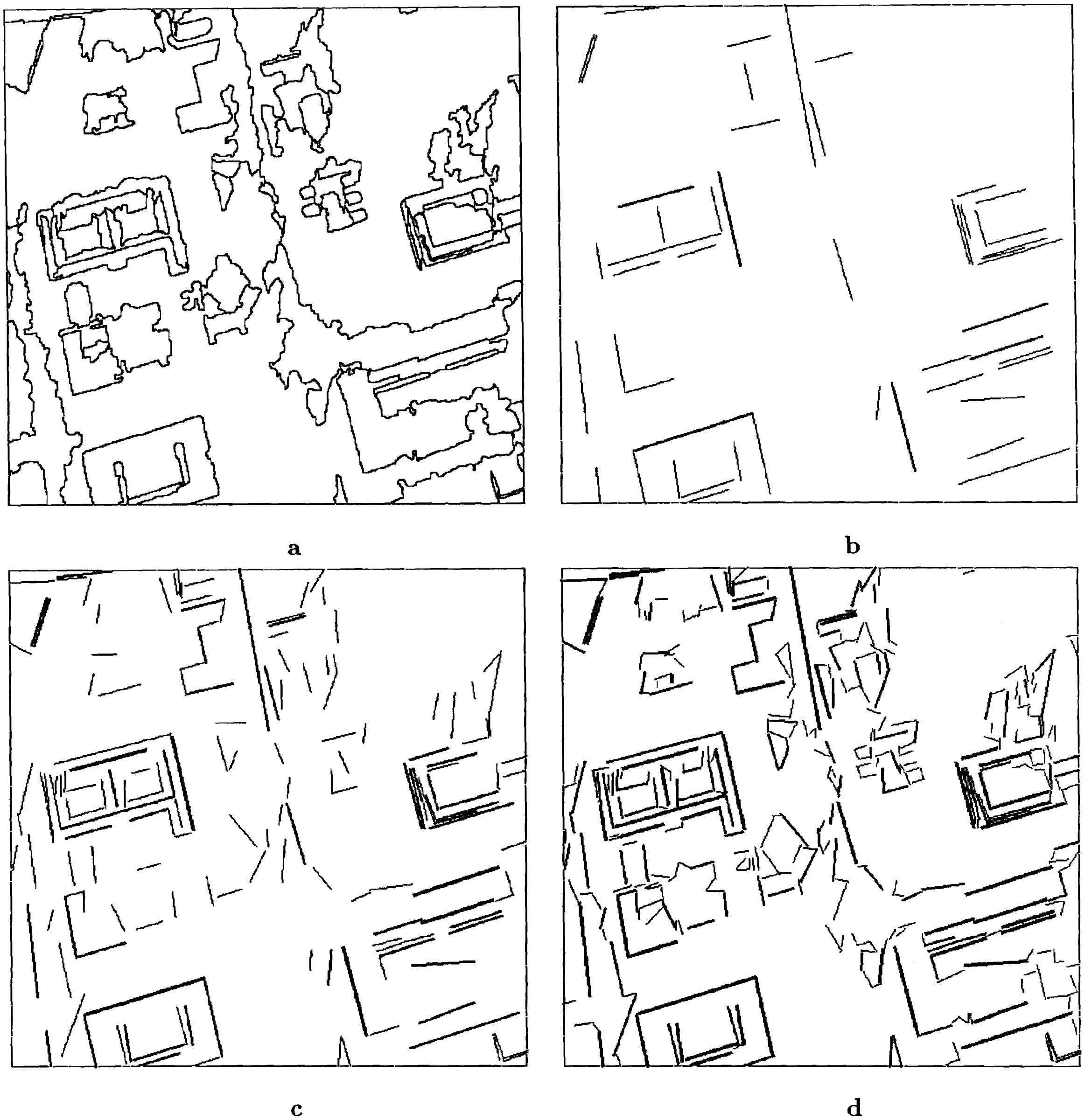

Figure 3: Segmentation and straight lines extraction
The segmentation (**a**) is produced by a region growing algorithm. The proposed straight line extraction algorithm is then applied to the region boundaries. **b**, **c** and **d** show the sets of straight segments that are selected for three successive values of the straightness threshold: 0.02, 0.04 and 0.08. Bold lines enhance the main straight ones.

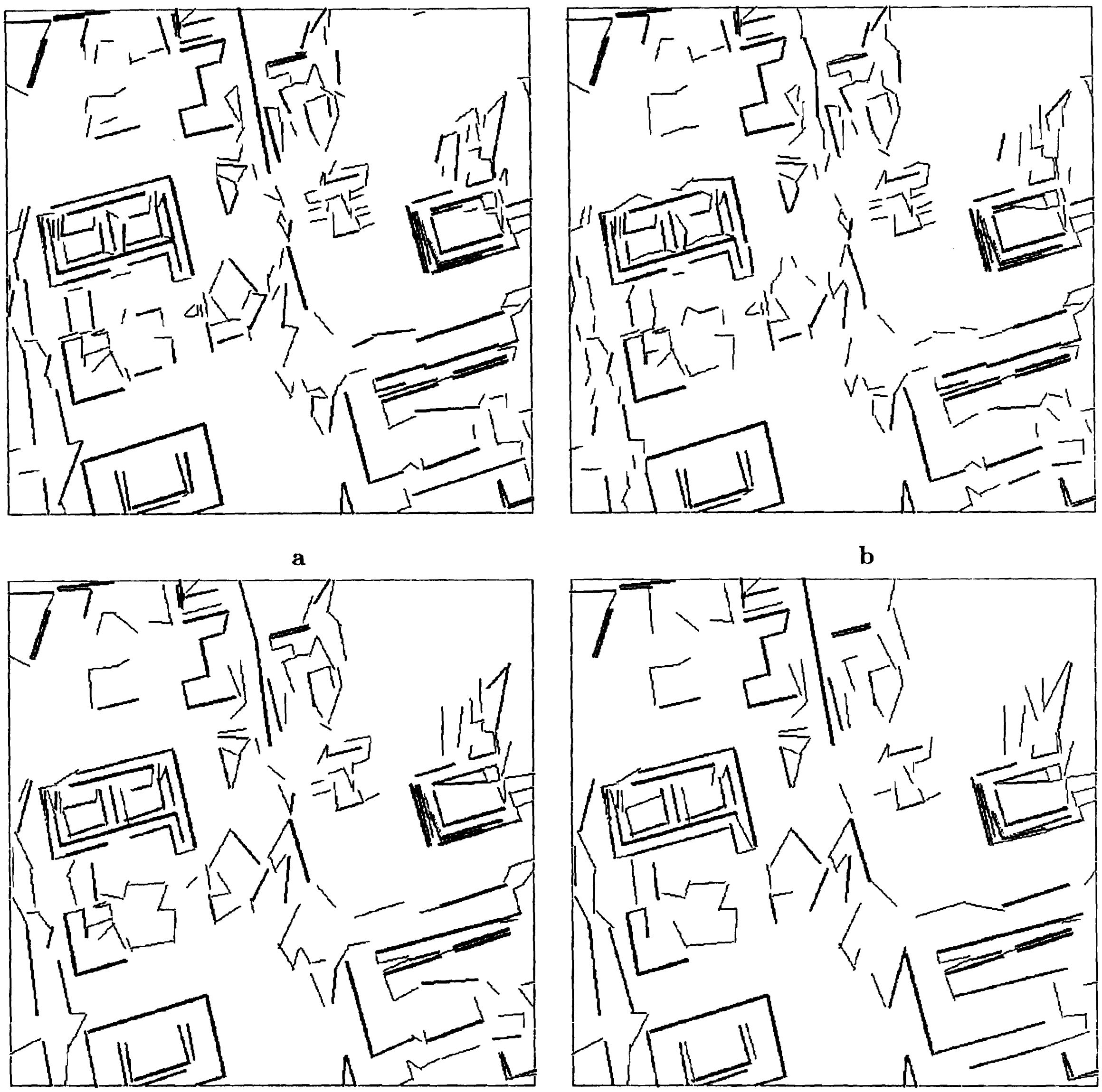

Figure 4: Comparative results

The results of the present method (**a**) are compared with those of the Ramer algorithm (**b**, **c** and **d**) for three successive values of the error bound (3., 6., 9.). Segments produced by the Ramer algorithm are reevaluated with the proposed criterion, they are displayed the same way (optimal least square positioning) and selected with the same straightness threshold of 0.07.

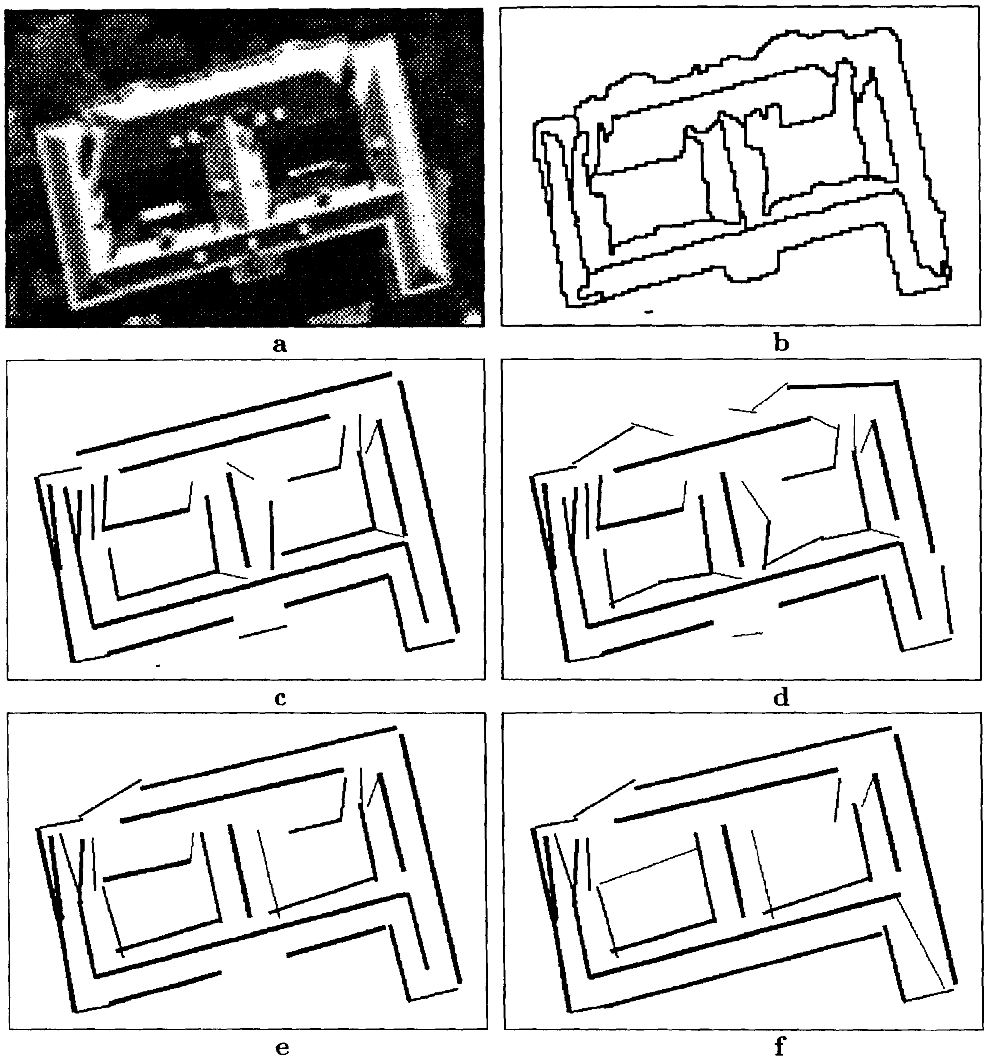

Figure 5: Comparative results (detail)
Enlarged view of a building in the original picture (**a**), of the region segmentation (**b**), of the straight segments extracted by our algorithm (**c**) and of those extracted by the Ramer algorithm, for three successive values of the error bound: 3., 6., 9. (**d**, **e** and **f**). The straightness threshold was set to 0.07.

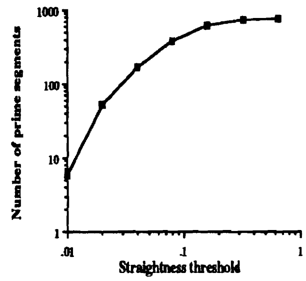

Figure 6: Counting prime segments
Number of prime segments as a function of the straightness threshold in logarithmic scales.

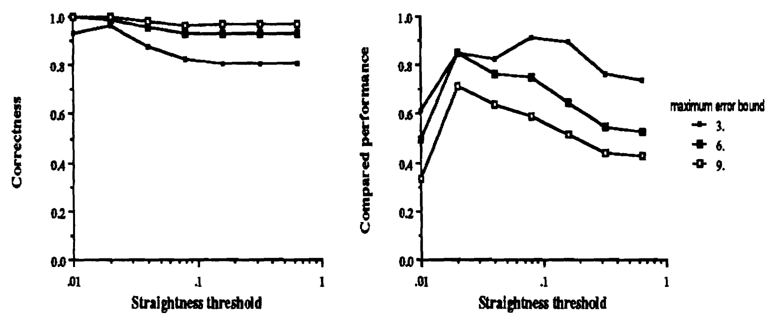

Figure 7: Comparative evaluation of the Ramer method
Results for three successive values of the error bound : 3., 6. and 9..

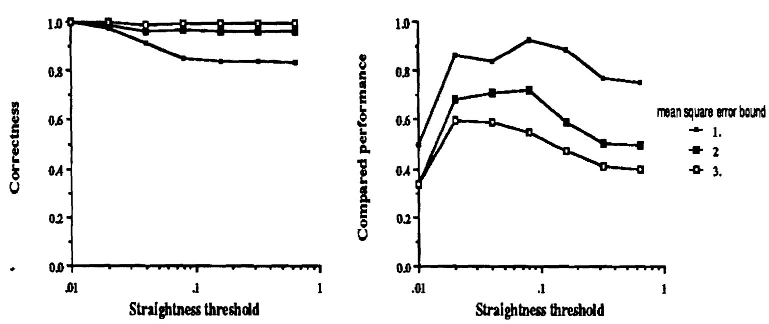

Figure 8: Comparative evaluation of the Pavlidis method
Results for three successive values of the error bound : 1., 2. and 3..

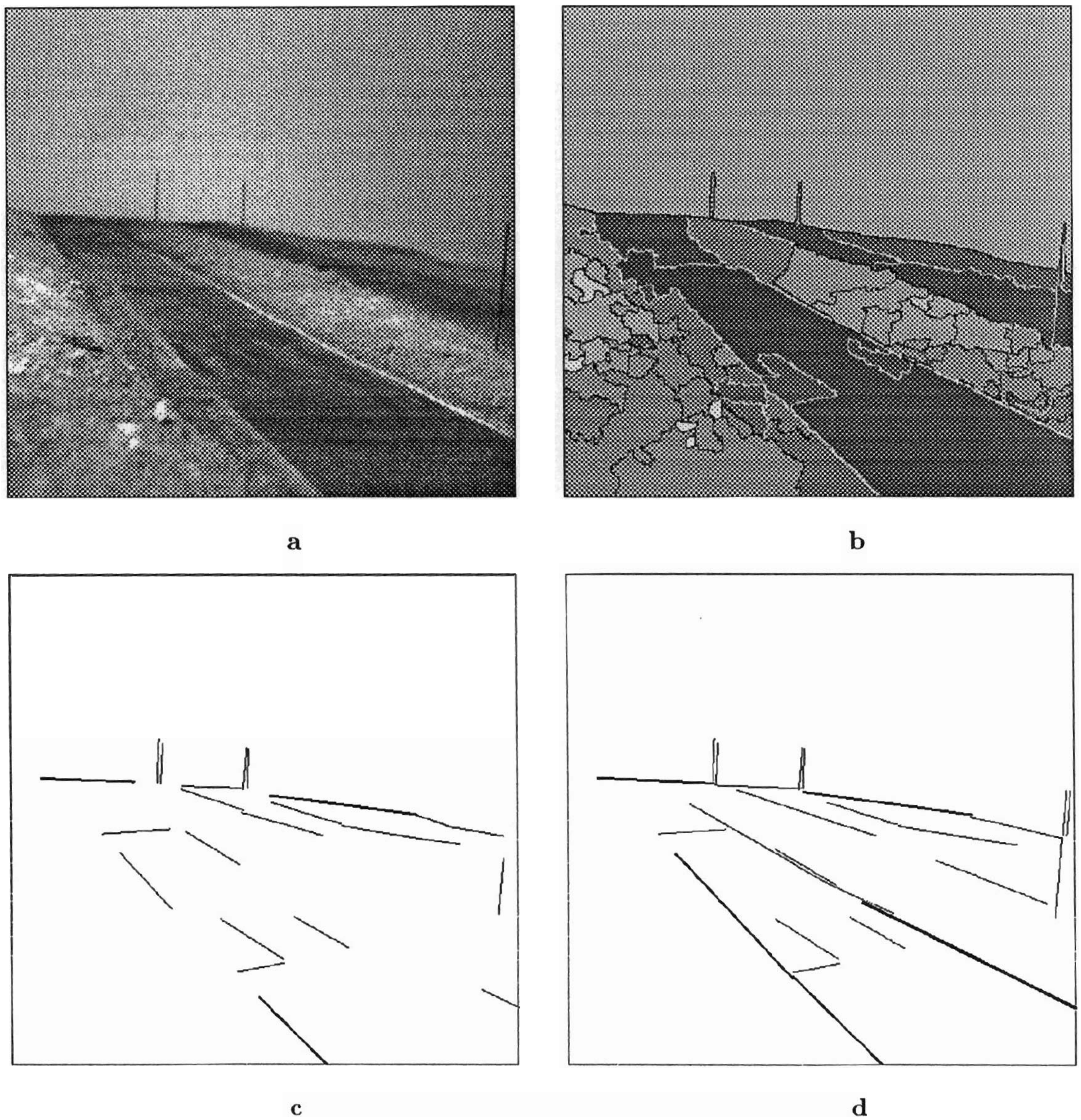

Figure 9: Algorithm extension
Roads in the desert (**a**) and the associated region segmentation (**b**). Due to poor contrast of the roads, a lower segmentation of the picture would result in undesirable merges. Straight line segments extraction (**c**). A second pass of search in the lattice finds immediate extensions of the straight line segments and almost achieves a complete recovery of the roads contours (**d**).