Automatic segmentation of metaphase cells based on global context and variant analysis*

Gunter Ritter† and Le Gao
Universität Passau,
Fakultät für Mathematik und Informatik,
D-94 030 Passau

May 19, 2007

Abstract: We treat the problem of chromosome segmentation with the aid of shape analysis and classification. Our approach consists of a combination of two phases, a purely rule–based phase and a phase driven by constrained discriminant analysis. In the first phase, obvious prototypical shape elements related to touchings and overlaps are recursively identified, in the second, remaining complex and ambiguous cases are treated. The latter phase exploits global context by using variant analysis, a statistical theory of ambiguity recently established. The method turns out to be quite accurate. The system works on whole clinical cells and to a certain degree when band patterns are not or not well visible.

Key words: Automatic chromosome segmentation; karyotyping; shape analysis; context; variant analysis; diagnostic classification.

*Work supported by Deutsche Forschungsgemeinschaft, Ri477/4
†Corresponding author, email: ritter@fmi.uni-passau.de

1 Introduction

1.1 Automation of karyotyping

The period of reproduction during the cell cycle of a eukaryotic organism is called mitosis, that between mitoses the interphase. While chromosomes are in an extended state at interphase, they form oblong, detached objects at two phases of mitosis, the late prophase and the metaphase. It is at these stages that chromosomal structures can be made visible under a light microscope after suitable staining, see Fig. 1. This enables the cytogeneticist to detect gross aberrations in the chromosomal structure caused by pathological processes, genetic degeneration or environmental factors such as radiation. The number of chromosomes in the cell is specific for the species, in the case of humans 46 subdivided in 24 types. Each chromosome carries a constriction called the *centromere*. A chromosome is called *acrocentric* if the centromere is off center and *metacentric*, otherwise.

As a first procedure during the analysis of a pro– or metaphase cell, the cytogeneticist usually produces a karyogram. This is an arrangement of all its chromosomes displaying their biological classes [10] together with their polarities, see Fig. 2.

Figure 1: A human metaphase cell

Karyotyping is a routine cytogenetic task, nowadays usually performed with the support of an interactive system. It lends itself to automation. In fact, its automation has a long history beginning in the nineteen sixties, [17, 18]. Fully automatic systems usually follow a number of consecutive steps.

(i) *Cleaning* of the image from stains and interphase nuclei;

(ii) *segmentation* of the cleaned metaphase cell in its different chromosomes;

(iii) extraction of *features* from all chromosomes;

Figure 2: The karyogram associated with the cell of Fig. 1

(iv) *classification* of the feature sets into the biological classes.

The first two steps employ methods from image processing, the last two from pattern recognition and statistics. Since stains are in general smaller than the smallest chromosome and nuclei are big, round, and dark objects, both can be easily recognized and step (i) is not much of a problem. Opposed to (i), step (ii) remains a challenge, at least if it is to be performed with an accurateness close to that of the expert cytogeneticist and for all kinds of cells and preparations. The reasons are clusters of touching and overlapping chromosomes that regularly appear in the 2D images of modern preparations. Steps (iii) and (iv) are not easy either but have been satisfactorily solved in the past. In step (iii), it is favorable to extract about 30 features from each chromosome. They are mainly the size, the mean density, and a large number of features computed from the density and shape profiles, see [27].

The pattern recognition stages (iii) and (iv) receive from a successful segmentation process all chromosomes of the cell as isolated objects. However, feature extraction needs in addition the shapes of the chromosomes in the form of their longitudinal axes and their polarities. Both are *ambiguous* at this stage, the former due to possibly bent shapes. As a remedy, we proposed the application of a general concept for resolving ambiguities in pattern recognition based on statistical decision theory: variants [30, 34, 9]. A variant of an object is a feature set extracted from an object under a certain interpretation. Since the correct interpretation may not be known at a certain stage in a feature extraction process, several variants corresponding to different interpretations are extracted. The variant corresponding to the correct interpretation is the *regular* variant, the others are *irregular*. The problem is to find the regular variant. Variant analysis has found applications to the recognition of polarities [35] and shape [37] and to motif discovery in regulatory genomics [9].

Of each chromosome, at least two feature sets (variants) are extracted, one for each polarity [35], in the case of bent chromosomes or otherwise unclear shapes even up to twelve [37]. Each one accounts for possible interpretations of shape and polarity. Constrained classifier–selectors were designed that use all

variants at a time [33]. For each chromosome, they estimate the regular variant associated with the true polarity and shape interpretation classifying all feature sets simultaneously. The most accurate *classifier* published to date for the present purpose is a robust maximum–a–posteriori estimator derived from a statistical model of the (random) karyotype. It postulates independent chromosomes after normalization of the features across the cell. The model of (the features of) a chromosome consists of a mixture of a normal distribution and a quadratically asymmetric distribution [29, 31] based on elliptical symmetry. The latter accounts for outliers and is responsible for the robustness of the classifier, see also [8, 32].

At first sight, estimating the biological classes of chromosomes and, at the same time, selecting the correct variants may appear to be computationally infeasible. However, it turns out that the classifier–selector reduces to a transportation problem well known in operations research; there are efficient algorithms for its solution. To our knowledge, the discovery of the connection between a constrained ML–classifier as above and the transportation problem is due to [48]. It was applied to karyotyping in [49] and to classification in the presence of variants in [33].

In this paper, we take a look at the *segmentation* process (ii) showing among other things that variant analysis may be successfully applied to this problem, too. We use it to take into account *global pictorial context*, a notion recognized as important also in other fields of image analysis, see, e.g., Torralba [47].

1.2 State of the art in chromosome segmentation

Automation of segmentation of metaphase images has a long history, two of the earliest sources being Ledley et. al. [18] and Hilditch and Rutovitz [13]. However, in the early years, cells at late metaphase were used for analysis. At this stage, chromosomes are in a contracted state so that touchings and overlaps do not occur frequently and segmentation reduces mainly to finding the connected components in the image. The situation changed when cytogeneticists began to exploit the advantages of the early metaphase and late prophase for their analyses. Such preparations display many more bands and more detail. Modern preparations of amnion and blood cells consist of chromosomes whose shapes resemble short pieces of rope. Due to their greater lengths, chromosomes tend to touch and overlap to a significant degree. Clusters of ten or more chromosomes are not rare. Therefore, a fully automated analysis today cannot dispense with a sophisticated component for disentangling clusters.

The problem consists of automatically *detecting* the clusters caused by touchings and/or overlaps and of *decomposing* them in their constituent chromosomes. Touchings are separated by single cut paths and overlaps by two more or less perpendicular pairs of cuts. The system has to find the right pairs of cut points and paths in all cases. Much progress has been made in the last fifteen years and five main ideas have appeared: shape concavities, pale paths (between touching chromosomes), the skeleton, shape validation, and segmentation driven by classification.

Finding candidates for cut points. In general, two interpenetrating objects such as chromosomes at pro– or metaphase create a number of concave boundary points; cf. also the *principle of transversality*

for convex bodies, [11, 22]. The same is true when the objects touch. Therefore, almost all authors stress the importance of geometric descriptors such as the boundary curvature and concavities along the object boundaries for detecting candidates for cut points, see [14, 15, 1, 21, 20, 44]. Concavities are detected by local methods, points of large negative curvature or a large negative deflection angle. In order to reduce the influence of noise, the latter is measured as the so–called k–deflection angle or by means of the proportion of background and foreground pixels in the neighborhood of the point. Another important descriptor is the skeleton, a 1D representation of the component by strokes of thickness one in its interior. Some workers propose pale paths between concave points as potential cut paths, [14, 15, 44]. The nodes of the skeleton indicate touchings and overlaps and one searches the boundary in their vicinity for potential cut points, [15, 44, 50].

Charters and Graham [5] and Urdiales García et al. [50] take a completely different approach to segmentation. They propose to use partial band profiles for recognizing cuts and disentangling overlapping chromosomes.

Selection of cuts to be executed. Many more, also incorrect, cut points are found than actually needed. It remains to select the cuts that have to be accepted and executed. Some authors recur to validating the potential cuts, again by using geometric information about the components that would result from them, see [14, 15, 1]. A most promising method was first proposed, but not pursued, in [28, 14]: postponing the decision on which of the proposed cuts should actually be executed to the later stage of classification. This method is well known in optical character recognition, see Casey and Lecolinet [4]. To the best of our knowledge, it was for the first time applied to chromosome segmentation by Lerner, Guterman, and Dinstein [21] under the name of *"classification driven"* segmentation. However, these authors use the method for separating clusters of *two touching* chromosomes, only.

2 The proposed approach to segmentation

Our approach consists of two phases. In the first phase, which we call the *clear phase*, we apply a number of stringent, mainly geometric, rules that detect clear, easily identifiable touchings and overlaps with high confidence. All rules use traditional methods from image processing and some of them are based on *local context*. We relate these methods to the present problem of chromosome segmentation in Sect. 2.1. In Sect. 2.2 we explain the various components of the clear phase.

Since the cuts proposed by these rules are definitely carried out, a major concern in the clear phase is keeping the number of false positives low. As a consequence, there is likely to be a number of touchings and overlaps that are not detected here (false negatives). These are mainly elusive and ambiguous cases. This means that, in general, the clear phase does not produce the target number of components (46, 45, or 47) and some clusters remain. Therefore, we let it be followed by a second phase which we call the *ambiguous* phase. Here, we relate the components to the *global context* of a karyogram in order to resolve the difficult cases. The technique that we employ to this end is variant analysis, a general statistical

paradigm designed for resolving ambiguities. In Sect. 2.3 we state a theorem on variant analysis which we apply in Sect. 2.4 to the present problem. This phase is driven by classification and uses implicitly the characteristic internal banding structures.

2.1 Tools from image processing

In a preprocessing step, the 4–connected components of the metaphase cell are extracted by means of a standard algorithm that determines all points reachable from a seed by 4–paths. Big, round, dark components represent interphase nuclei and are removed. Chromosomes of class 21 are smallest, each occupying about 0.9% of the total area, see [10], Chapter 8. Since this number is subject to variation, all connected components of less than 1/220 of the total area are removed as small stains. With the exception of a few artifacts, all remaining connected components are composed of chromosomes. Their boundaries are next smoothed with morphological algorithms [43]. Moreover, we fill small holes of less than 15 pixels since they are usually artifacts within chromosomes, Fig. 3, unlike genuine holes between chromosomes, see Fig. 11.

Figure 3: A hole as an artifact in the image.

Many components represent isolated chromosomes but, often, also clusters of chromosomes. It is the principal task to detect and to resolve these clusters. Our segmentation process is guided by the following principles.

- a branching of the skeleton with at least two long arms indicates a cluster of chromosomes;
- strongly concave boundary points indicate potential cut points;
- a narrow constriction in a component connects two chromosomes unless it is a narrow centromere;
- An object of size less than 1/220 of the total image area is no chromosome; thus, pieces of this size are not cut off. In particular, an object of size less than 1/110 of the image area is not cut.

In order to recognize the first three situations we employ the following tools.

- The 4–boundary of each component, an 8–connected digital curve that consists of the boundary points in their natural order, cf. Rosenfeld's [38, 53] algorithm;
- the boundary curvature, the sequence of the k–deflection angles along the boundary;
- T–essential minima of noisy functions, a concept introduced in Ritter and Schreib [37];

- the skeleton (a global shape descriptor), see Appendix A.1;
- branching and crossing points of the skeleton;
- the Euclidean distances of the skeleton pixels, in particular of its branching and crossing points, to the complement of the component;
- polygonal approximation of the boundary (another global shape descriptor), see Appendix A.2;
- the average width of the chromosomes in a cell as the most important dynamic scale parameter.

The *average width* may be computed with the nonbranching skeletons since these are likely to be isolated chromosomes or at least no complicated clusters. We compute the *Euclidean* distance of all pixels in these skeletons to the complement of their component. Two times the mean of the distances is an estimate of the average width.

2.2 The clear phase

In this phase, we establish a number of rules for plain cuts to be carried out by means of Bresenham's [3] algorithm. As argued at the beginning of Sect. 2, we have to *avoid dissecting chromosomes* here. In order to achieve this goal, the rules have to be stringent and conservative. They are governed by the shape of the cluster, by the geometry of the cut, and by the complexity of the objects created by the proposed cut. Some of the rules are based on local measurements, others on global descriptors. They do not exploit the banding patterns and, except for one rule, they only need a black and white image. In this phase, the only errors that matter are dissected chromosomes, missed cuts are less important. In the experimental section, we show that the rules generate only few errors and, yet, this phase leads to a segmentation into in most cases 40 or more components.

In order to establish useful rules it is necessary to realize the various cases of possible interactions of two (or more) chromosomes. They may be classified into a few categories of shape elements which reflect the different relative positions that two touching or overlapping chromosomes can assume. The most frequent case is the touching of the tip of a chromosome at an edge of another. The touching may be light or tight. It also happens quite often that the tips of two chromosomes touch in which case the angle between the chromosomes may be acute or obtuse or the chromosomes may be almost in a line. It also happens that the edges of two chromosomes create a perfect occlusion. Finally, two chromosomes may overlap, sometimes near the tip of one of them. Therefore, our rules in the clear phase correspond to four frequent shape elements: *Light touchings* or *bridges*, X–shaped *overlaps*, and *tight touchings* of two chromosomes which we subdivide in M– and T–shaped touchings. Overlaps can create four long branches or three branches and a bulge, see Fig. 5. Of course, these causes can combine in one cluster. (More complicated overlaps such as a threefold overlap at the same point are rare and not considered here.) Our system treats the four cases in the following order.

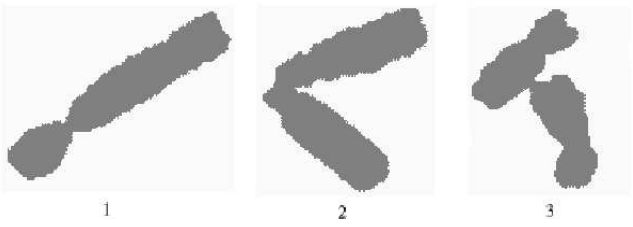

Figure 4: The first chromosome possesses a narrow centromere; it passes the centromere test since its width remains sufficiently small in both directions from the constriction and its boundary is nowhere acutely convex in the neighborhood. The boundary of cluster 2 is acutely convex close to the constriction and cluster 3 widens too much in one direction; these constrictions are recognized as touchings.

(a) **Bridges**. Bridges cover all cases of light touchings of two chromosomes. A light touching creates a short, narrow constriction in a cluster. The two boundary points defining the cut are characterized by two features:

– the boundary curvature has sufficiently negative T–essential minima there,

– they are at a Euclidean distance $\leq 0.37\times$(average width).

The notion of T–essential minimum is introduced in Ritter and Schreib [37]. It serves here to detect significant shape concavities. The factor 0.37 looks quite restrictive. There is, however, a complication that was noted by many authors. The centromeres of some chromosomes are quite narrow so that a larger factor would lead to chromosomes dissected at their centromeres, a bad mistake.

Unfortunately, despite the smallness of the factor, there are still some centromeres that satisfy the two conditions. In order to avoid cutting such chromosomes, we apply a centromere test: in the vicinity of a centromere, a chromosome is nowhere acutely convex and it has a regular width, say $\leq 1.3\times$(average width), there. If a narrow constriction fails to pass this test then it is recognized as a bridge, cf. Fig. 4.

Bridges are identified quite safely and cutting them greatly reduces the complexity of the clusters. For these reasons we put this rule at the beginning. Tighter touchings have several occasions to get cut later.

(b) **Overlaps** and **X–configurations**. We call a shape element that resembles the letter X an X–configuration. X–configurations are the most complex of all. They have to be detected early since

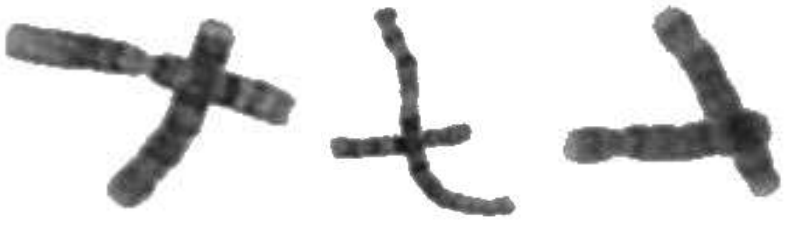

Figure 5: Two plain overlaps and an overlap with a bulge

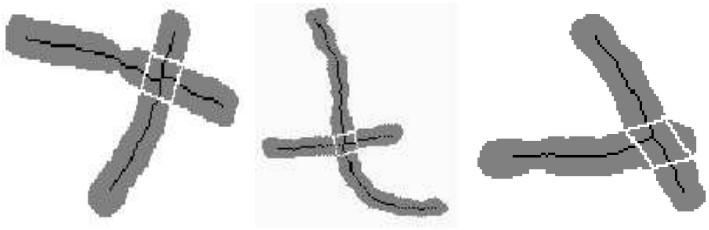

Figure 6: The overlaps of Fig. 5 with their skeletons and cuts

.

Figure 7: X–configurations of types (ii) (the first two) and (iii)

failure to cut them properly will result in dissected chromosomes later. As depicted in Figs. 6 and 7, X–configurations may be caused by (i) overlaps, (ii) two chromosomes touching a third one at the same site from opposite sides ("three–component X"), or (iii) two bent chromosomes touching each other with their knees. The treatments of these three cases are very different. As pointed out by Ji [15], an overlap is characterized by a branching or crossing point of the cluster skeleton, four cut points at the intersection of the boundaries, and a high density within the quadrilateral spanned by them. The last feature distinguishes overlaps from the other types (ii) and (iii). Adapting his criteria, we first describe an X–configuration in a cluster by

– a branching or crossing of the cluster skeleton with at least two long arms,
– four low T–essential minima of the contour curvature close to the branching, and
– convexity of the quadrilateral specified by the four points and interiority of the skeleton node.

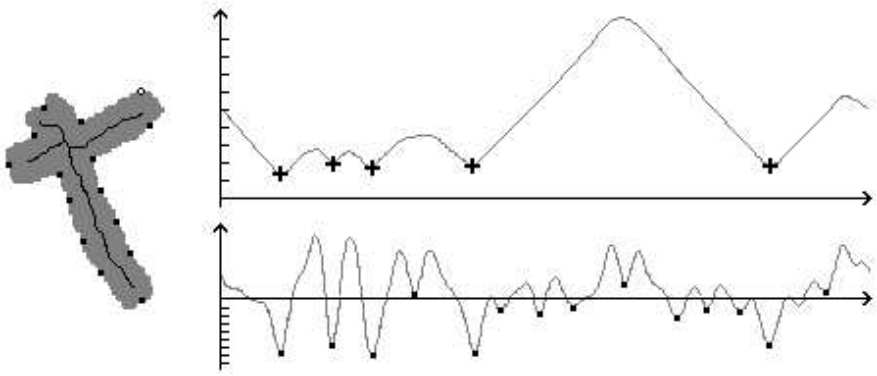

Figure 8: Distance curve w.r.t. the crossing point (top) and boundary curvature of an overlap

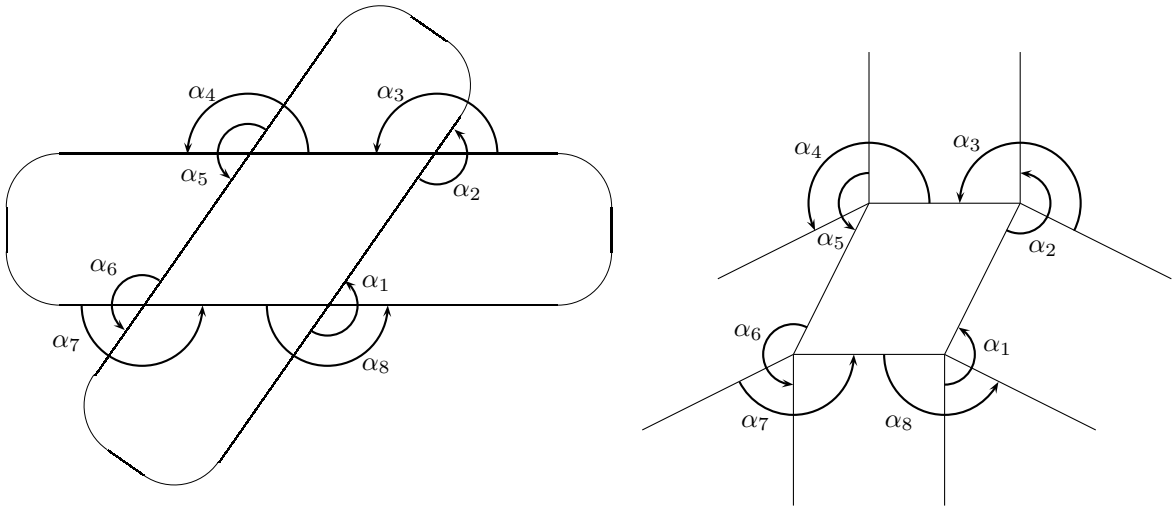

Figure 9: Graphical representation of the cut complexity. Left: an ideal overlap, $\alpha_1 = \alpha_2$, $\alpha_3 = \alpha_4$, $\alpha_5 = \alpha_6$, and $\alpha_7 = \alpha_8$, cut complexity = 0. Right: a 3-component X, the horizontal chromosome is bent and touched from below and above by two other, not perfectly aligned chromosomes. While $\alpha_3 = \alpha_4$ and $\alpha_7 = \alpha_8$, we have $\alpha_1 \neq \alpha_2$ and $\alpha_5 \neq \alpha_6$ and the cut complexity is high.

The second criterion is illustrated in Fig. 8. We now have to distinguish between overlaps and types (ii) and (iii). Roughly speaking, if after execution of the cut the two parts created look sufficiently smooth near the cuts and if, in addition, the quadrilateral is unusually dark then we consider the X–configuration an overlap. More precisely, our rules are based on the *cut complexity*

$$\text{cut complexity} = \max\{|\alpha_2 - \alpha_1|, |\alpha_4 - \alpha_3|, |\alpha_6 - \alpha_5|, |\alpha_8 - \alpha_7|\}$$

and on the *relative brightness* of the quadrilateral spanned by the four concave boundary points

$$\text{relative brightness} = \frac{\text{black - mean density in quadrilateral}}{\text{black - mean density in cell}}.$$

If a configuration is an ideal overlap then the cut complexity vanishes, see Fig. 9, and the relative brightness is low.

If the pair of values lies in region 1 of Fig. 10 then we decide that the X–configuration is an overlap carrying out two pairs of cuts along the opposite sides of the quadrilateral. If the pair lies in region 3 then we decide that it is no overlap opening it for other cases. Whereas regions 1 and 3 contain mainly (but not exclusively) overlaps and X–configurations of types (ii) and (iii), respectively, region 2 is critical since it contains a mixture of all types. It is not yet clear from the cut complexity and the relative brightness alone which case applies. These configurations are barred from cutting in the clear phase and decided later in the ambiguous phase. The three regions 1, 2 and 3 have been defined by graphical methods with a small part of our data set.

If a three–component X is treated as an overlap, one of the two components created contains the overlap as an additional part but no wrong cut is carried out. On the other hand, if an overlap remains undetected at this stage then some of its branches will most probably be cut off later leading to at least one dissected chromosome.

It happens quite often that more than four concave points are found near a skeleton node. In this case, we form a quadrilateral of the three concave points with the smallest distances from the node together with the concave point with the smallest value of some score based on geometry, cut complexity, and relative brightness. The simple solution to just take the four points closest to the node may be wrong since a spurious concave point (e.g. a centromere) may happen to be very close. More precisely, we follow the algorithm described in Table 1 in order to detect overlaps.

An overlap may cause an additional problem. If a part of its boundary belongs to a hole within the cluster then the cut cannot be executed since the two components would remain connected, see Fig. 11. In this case, the cut is blocked for the treatment of other configurations and executed as soon as the hole is opened. In some rare cases, as in the triple overlap shown in Fig. 11, this will never happen. These are shape element that we did not consider; they contribute to our error rate.

(c) The **M–configuration** Two chromosomes that touch at their tips forming an acute angle as depicted in Fig. 12 create a configuration that reminds of the upper case letter M and which we therefore call an M–configuration. The touching is often too tight to pass as a bridge and needs a different method. We first approximate the cluster contour by a polygon, see Appendix A.2.[1] The M–configuration is then characterized by the following features.

– A trapezoidal suite of three line segments within the polygon, the *hull* of the M; the length of the middle segment is about 2×(average width) and the adjacent segments are at least that long;

– a concave point of the contour section between the two vertices of the hull;

– a V–shaped pair of line segments in the polygon inside of the hull so that the space between the hull and the V is filled with part of the cluster;

[1] An early application of polygon approximation to chromosome analysis appears in [51].

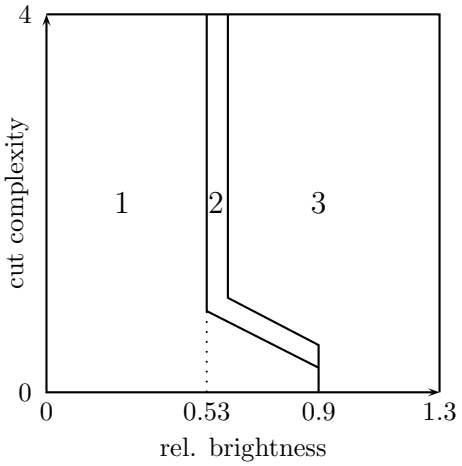

Figure 10: Cut complexity vs. relative brightness (w.r.t. the brightness of the cell) within the quadrilaterals associated with X–configurations. All configurations in region 1 are regarded as overlaps, those in region 3 as no overlaps, and those in region 2 are ambiguous.

Table 1: Recognition of overlaps in the clear phase

recognizeOverlap(image)

INPUT: The image of a component

OUTPUT: Four points that are likely to define an overlap *or* the response "overlap unlikely."

1. compute the skeleton of the image // see Appendix A.1
2. *if* it contains a branching or a crossing with at least two sufficiently long arms
3. *then* compute the smoothed curvature function of the cluster boundary and all its T–essential minima
4. *if* at least four such minima are < -0.4 and at a distance $< 2\times$(average width) from the branching point, and at least three of them are even closer than the average width
5. *then* choose the three points with the smallest distances and create quadrilaterals with each of the remaining points
6. *for* each of the these quadrilaterals
7. *if* it is convex and if the branching point is contained in its interior
8. *then* compute its cut complexity and its relative brightness // see 2.2(b)
9. *if* one of the quadrilaterals belongs to region 1 of Fig. 10
10. *then* return the one with the least score // see 2.2(b)
11. *else if* one of the quadrilaterals belongs to region 2
12. *then* refer the one with the least score to the ambiguous phase and bar the configuration from further treatment in the clear phase
13. *return* "overlap unlikely"

Figure 11: A blocked overlap and a deadlock from a triple overlap

– the arms of the V are about parallel to the exterior segments of the hull at a distance of about the average width.

Thus, detection of an M–configuration is based on five line segments of the approximating polygon. The cut line is defined by the tip of the V and the most concave point of the boundary between the two vertices of the hull.

(d) The **T–configuration and similar tight touchings** A T–shaped configuration is created by the

overlapping of the tip of one chromosome with the body of another or by a tight touching, see Fig. 13. Just as in the case of the M–configuration, we use again the approximating polygon. The T–configuration is characterized by the following features.

– Two angles $\leq 140^o$ in the approximating polygon and an additional "base segment" of length $\geq$ 2×(average width);

– the orthogonal projection of one vertex onto the straight line extending the base segment hits the segment and that of the other hits the segment or the extension not far from an end point of the segment;

– one side of one angle is almost parallel to the base segment and at a distance of about the average width from the base segment;

– the two vertices of the angles are on the same side of the (linear extension of the) base segment at a distance of about the average width from it. (One angle may be inverted as in the middle example in Fig. 13.)

If this is the case then a cut between the two vertices is carried out.

This concludes our description of the clear phase. A typical sequence of cuts, starting from one component, is presented in Fig. 14 and a schematic representation in Fig. 15.

2.3 Variants

As noted in the introduction, segmentation should not be separated from steps (iii) and (iv) since ambiguities arise also, and in particular, during the segmentation process. We show next that a natural way of translating this idea into action is variant analysis. It is carried out in the *ambiguous phase* which uses as input the output of the clear phase, see Fig. 22.

It is useful to first recall the problem of variant selection in an abstract context. Assume that an ambiguous object allows b interpretations so that we extract b competing feature sets (variants) from it. Let E be their common sample space, and let $Z = (Z_1, \cdots, Z_b)$ be the joint vector of all (random) variants in some fixed, predefined order, the regular variant in front. What we observe is the vector Z in disorder. Let $\mathbf{x} = (x_1, \cdots, x_b)$ be such an observed array of variants. We are mainly interested in

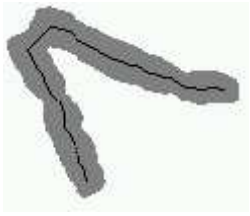

Figure 12: An M–configuration

Figure 13: T–configurations. Left with a branched skeleton, center without branching, right a cluster consisting of two T– and one X–configuration.

estimating the position $h \in 1..b$ of the regular variant in $\mathbf{x}$. Its MAP–estimate, the MAP–*selector* [34], is

$$\mathrm{MAP}_h(\mathbf{x}) = \operatorname*{argmax}_{h \in 1..b} P[T(h) = 1 \mid Z_T = \mathbf{x}],$$

where T stands for the random permutation of all variants. Thus, the vector Z_T represents the variants in their observed order. This selector needs the joint law of all variants which is not available in general. Moreover, the MAP–selector is quite complex, at least if b is large. The question arises whether it can be computed given the information on the regular variant alone. More precisely, we ask the question whether the regular variant Z_1 possesses a reference measure ϱ such that MAP_h equals the intuitively appealing *simple selector*

$$\mathrm{SS}_h^{\varrho}(\mathbf{x}) = \operatorname*{argmax}_{h \in 1..b} q_h f^{\varrho}(x_h).$$

Here, f^{ϱ} is the density function of the regular variant Z_1 w.r.t. ϱ and q_h is the prior probability of h to be the position of the regular variant. The simple selector chooses the variant whose product of density and prior probability is maximum. Ritter and Gallegos [34] stated and proved a number of positive answers to this question, but also counterexamples. In view of a positive answer, let K be the Markov kernel defined by

$$K(x, \mathrm{d}y) = P[Z_{\widehat{1}} \in \mathrm{d}y \mid Z_1 = x], \quad x \in E,\ y \in E^{b-1}.$$

The hat $\widehat{\ }$ indicates the missing first index, that is, $Z_{\widehat{1}}$ represents all irregular variants in their fixed order. Let L be the symmetrization of K,

$$L(x, \mathrm{d}y) = \frac{1}{(b-1)!} \sum_{\sigma \in \mathfrak{S}_{b-1}} K(x, \mathrm{d}y_\sigma).$$

The measure $\varrho \otimes L$ on E^b is defined by $(\varrho \otimes L)(B) = \int_E \varrho(\mathrm{d}x) \int_{E^{b-1}} L(x, \mathrm{d}y) \mathbf{1}_B(x, y)$. A measure on some product space is exchangeable if it is invariant w.r.t. arbitrary permutations of the coordinates. Corollary 3.15 of the paper just mentioned relates variant selection to MAP estimation. It reads

Theorem. Assume that the probability $P[T = \pi]$ depends on the site $\pi^{-1}(1)$ of the regular variant, only. If $\varrho \otimes L$ is exchangeable then the MAP–selector equals the simple selector, that is

$$\mathrm{MAP}_h(\mathbf{x}) = \operatorname*{argmax}_h q_h f^{\varrho}(x_h).$$

Figure 14: Decomposition of a cluster of unambiguous touchings and overlaps in the clear phase. Previews preceding the actual cuts are indicated.

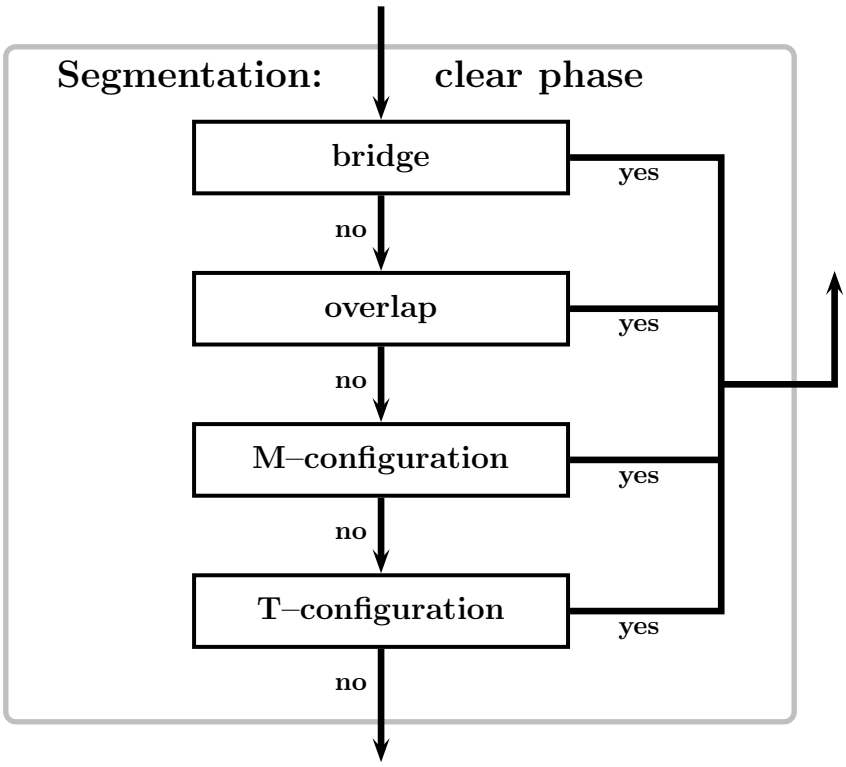

Figure 15: Schematic representation of the segmentation process in the clear phase with its four components

Now, consider a normal human cell and let each variant of a chromosome be represented by a d–dimensional feature vector. In the abstract setting above, the sample space E becomes $\mathbb{R}^{46\cdot d}$ and the regular variant Z_1 is the random karyogram of the cell.[2] In the present case, variants arise at three stages. *First*, during low level processing it is but elementary geometric information about the connected components of the cell that is available. The correct segmentation is not clear from this elementary information and we resort to extracting a number of *segmentation variants* $\{\mathbf{x}_s \mid s \in S\}$ from the cell. We decompose the output of the clear phase in various ways into 46 components by applying a set of rules loose enough so that the correct segmentation (interpretation of the cell) is among them with high confidence.[3] The details about this process are explained in Section 2.4. *Second*, even if we knew the correct segmentation, the interpretation of its 46 chromosomes would not be clear due to the polarity of the oblong chromosomes. Hence, even in the simplest case, each chromosome generates two variants, one for each polarity. Bent chromosomes may give rise to additional complications and even more variants. We thus obtain, from each segmentation variant $s \in S$ and each choice of polarities and shape interpretations, a *shape variant* $\mathbf{x}_{s,\mathbf{k}}$, $\mathbf{k} \in H(s)$. Although estimation of the classification is usually done by discriminant analysis we may simply view the unknown classification as a *third* source of ambiguity considering each

[2]For the sake of easier presentation, we assume that there is only one kind of cells, no males and females. The extension to the general case is straightforward, see again [35, 37].

[3]We also allow solutions with 45 and 47 chromosomes to account for certain pathologies such as Turner's, Pätau's (triple 13), Edward's (triple 18), Down's (triple 21), and Klinefelter's (XXY) syndromes.

permutation π of the 46 chromosomes of a segmentation with fixed shape variants a *permutation variant* $\mathbf{x}_{s,\mathbf{k},\pi}$. We thus end up with a huge, b–element set $\{\mathbf{x}_{s,\mathbf{k},\pi} \mid s \in S,\ \mathbf{k} \in H(s),\ \pi \in \mathcal{S}_{46}\}$ of variants. We assume here that their stochastic model satisfies the exchangeability required in the theorem with Lebesgue measure ϱ dropping the superscript ϱ.

Surprisingly and fortunately, the complexity arising from this large number of variants can be controlled. Let f be the likelihood function of the (random) karyogram. Applied to the present case, the Bayesian criterion for variant selection and classification shown in the theorem above reads

$$\max_{s,\mathbf{k},\pi} f(\mathbf{x}_{s,\mathbf{k},\pi}) = \max_{s} \max_{\mathbf{k},\pi} f(\mathbf{x}_{s,\mathbf{k},\pi}) = \max_{s} \widehat{f}_s, \tag{1}$$

where prior probabilities are assumed to be uniform. This means that, for each segmentation $s \in S$, we have to maximize the function $f(\mathbf{x}_{s,\mathbf{k},\pi})$ w.r.t. all its shape and permutation variants $\mathbf{k} \in H(s)$ and $\pi \in \mathcal{S}_{46}$ and, by the theorem, the MAP– estimate of the segmentation is the one for which the maximum $\widehat{f}_s$ is maximal w.r.t. s. Determination of each maximum $\widehat{f}_s$ means the application of an MAP–classifier–selector to the segmentation variant s. One may note that it is this estimator that does it all, segmenting the cell, finding the polarities, and classifying the chromosomes into their biological classes. Segmentation, feature extraction, and classification join in one unit that solves the whole problem.

Despite its appearance, the burden of computing $\widehat{f}_s$ is light. The general classifier–selector, constrained to the known number of objects, was discussed in detail in Ritter and Gallegos [33] and applied to chromosome classification in Ritter and Schreib [37] and we refer the interested reader to these papers and their forerunners.

It remains maximization w.r.t. all segmentation variants $s \in S$. This set being unstructured, we do not know of a better way of computing the maximum than enumerating it. This is infeasible if S is large. Since the number of different components involved in all segmentation variants of a cell is fairly small and features can be extracted from them beforehand, we only have to worry about the time necessary for their classifications. One classification takes about 0.1 sec on a 2 GHz processor, so that we can afford a few thousand segmentation variants with a processing time of a few minutes. Now, the number of segmentation variants grows combinatorially with the number of cuts necessary, in general 46 minus the number of connected components. This means that the method just described can treat cells that initially contain about 40 or more components, only. If there are fewer then we generally find too many possible individual cuts and their combination to segmentation variants leads to a combinatorial explosion. Since many metaphase cells initially contain less than 40 components, see Fig. 23, we preceded the above method with the *clear phase*. Its output is the input to the process outlined above. In most cases, it consists of a number of components large enough to allow the processing of all segmentation variants created from them; see again Fig. 23.

A simple illustration of the effect of segmentation variants is this: Assume, e.g., that the four longest chromosomes in the image are clearly identifiable. Assume further that the image contains a composite component of the same length, e.g. two shorter chromosomes joined at their tips. Then the classifier

will associate an unfavorable score with any segmentation variant that contains this component as a single object since the classifier will correctly assign the four longest chromosomes to their types and since the assignment of the composite component to another type spoils the score. Our statistical model of a karyogram offers no room for this object. The segmentation tool is thus encouraged to select a segmentation variant with this object cut. In this way, cuts in any component may be influenced by possibly remote objects, the four chromosomes in this example. Therefore, our method exploits global context.

2.4 Variant generation in the ambiguous phase

Here, we treat the ambiguous X–configurations of region 2 in Fig. 10, look for more bridges and T–configurations, but with relaxed rules and/or parameters, and introduce two new configurations, L and B, in order to generate tentative cuts. Their number is just restricted by complexity considerations since each component that allows a cut increases the number of proposed karyograms to be investigated multiplicatively. If a cut leads to a piece that grossly fails to look like a chromosome then the cut is not executed in order to reduce combinatorial explosion. Combining the remaining cuts to solutions with 45, 46 and 47 chromosomes in all possible ways, we generate from the output of the clear phase an, in general moderate, number of potential karyograms. As described in Section 2.3, the classifier–selector selects the final solution and, thereby, the cuts that are actually executed. Contrary to the clear phase, this phase heavily exploits the complete statistical information on the karyotype and, thus, the interior band patterns. In this sense, it is the classifier that resolves the subtle cases.

(a) **Ambiguous X–configurations** can essentially have four different interpretations. They may be left as they are, they may be cut as an overlap, and there are two ways of interpreting them as three–component X's, see Sect. 2.2(b) and Fig. 16. We do not consider X–configurations of type (iii) since they are very rare.

Figure 16: An ambiguous X–configuration of type (ii) and its four variants. The third is correct.

(b) **Bridges** The relaxed bridge is characterized by two boundary points with the following properties.

– one boundary point has a sufficiently negative T–essential minimum,

– its Euclidean distance from the other boundary point is small, $\leq \frac{1}{2}\times$(average width),

Of course, many centromeres satisfy these conditions. However, no centromere test is necessary, here, since we rely on the classifier to reject wrong cuts.

(c) The **T–configuration** The method is the same as in the clear phase, however with relaxed parameters.

(d) The **L–configuration** Like the M–configuration, the L–configuration is created by two chromosomes that touch at their tips as depicted in Fig. 17. Unlike the M– and T–configurations, there is no second concave point. The L–configuration uses again the polygon approximation; it is characterized by

– a skeleton without branching or crossing whose polygon approximation contains a distinct angle with vertex V_0;

– a convex angle in the polygon approximation of the boundary exterior to the skeleton and whose vertex V_1 is no farther than the average width from V_0;

– a concave angle in the polygon approximation of the boundary interior to the skeleton and whose vertex V_2 is no farther than the average width from V_0.

Figure 17: An L–configuration with its skeleton and its two variants; right is correct

The second condition distinguishes the L–configuration from a bent chromosome. If a component is recognized as an L then two potential cuts are proposed, namely the extensions of the two sides beyond the concave boundary point V_2, see Fig. 17.

(e) The **B–configuration** The B–configuration designates mainly a situation where two chromosomes create a perfect occlusion along two edges as in Fig. 18. The characteristic of a B–configuration is that

– it contains a Euclidean disk of radius about the average width.

Thus, a B–configuration is characterized by a large maximum of the Euclidean distance transform, see [45]. In this case, the negative T–essential minima of the boundary curvature and the polygonal approximation of the boundary give hints to potential cuts.

Now, each component is examined for all configurations indicated, see Fig. 19, the proposed cuts are executed, and the components thus created are recursively treated in the same manner, see Fig. 20. We call the various interpretations of the components "local" variants.

In our implementation, all local variants and the dissected fragments are stored in the data structure exemplified in Fig. 21. Local variants combine to "global" variants, various interpretations of the whole metaphase cell as potential karyograms. Global variants are represented by the paths from the left to the right end in the data structure which contain 45, 46, or 47 components. Of course, if the deterministic

Figure 18: A B–configuration

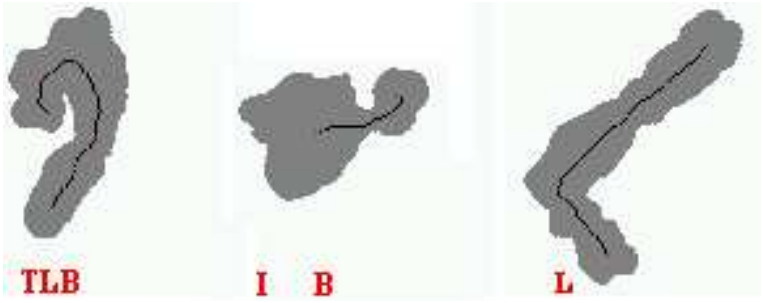

Figure 19: Proposed configurations for variant generation. The letter "I" stands for a bridge.

phase resulted in 46 components then no solution with 45 components can be created; likewise, if the result was 47 components then no further variants are considered.

The combinatorial explosion caused by allowing too many local variants is the main restriction to the number of cuts that can be handled in the ambiguous phase. This is also the reason for the necessity of the clear phase which must not be too restrictive. In the results section, we show that the explosion can be controlled in practice.

2.5 The overall segmentation process

Fig. 22 shows the flow diagram of the complete segmentation process after cleansing from small stains and nuclei. During *preprocessing*, some morphological operations are applied for smoothing. Next, the image is decomposed into its 4–connected components by a standard algorithm that determines all points reachable from a seed by 4–paths. Next, shape descriptors, such as the boundary, the boundary curvature, the skeleton, and the polygon approximation are computed from all components, see Section 2.1. The information related to the skeleton such as the branchings, crossings, and their Euclidean distances from the complement is collected and represented in a *skeleton graph*.

The subsequent clear phase of segmentation detects bridges, overlaps, M's, and T's as described in Section 2.2 and cuts them. The suite consisting of "connected components – shape descriptors – segmentation" is reiterated until no more cuts are executed.

Next, the process enters the ambiguous phase. It starts with the recursive creation of local variants as

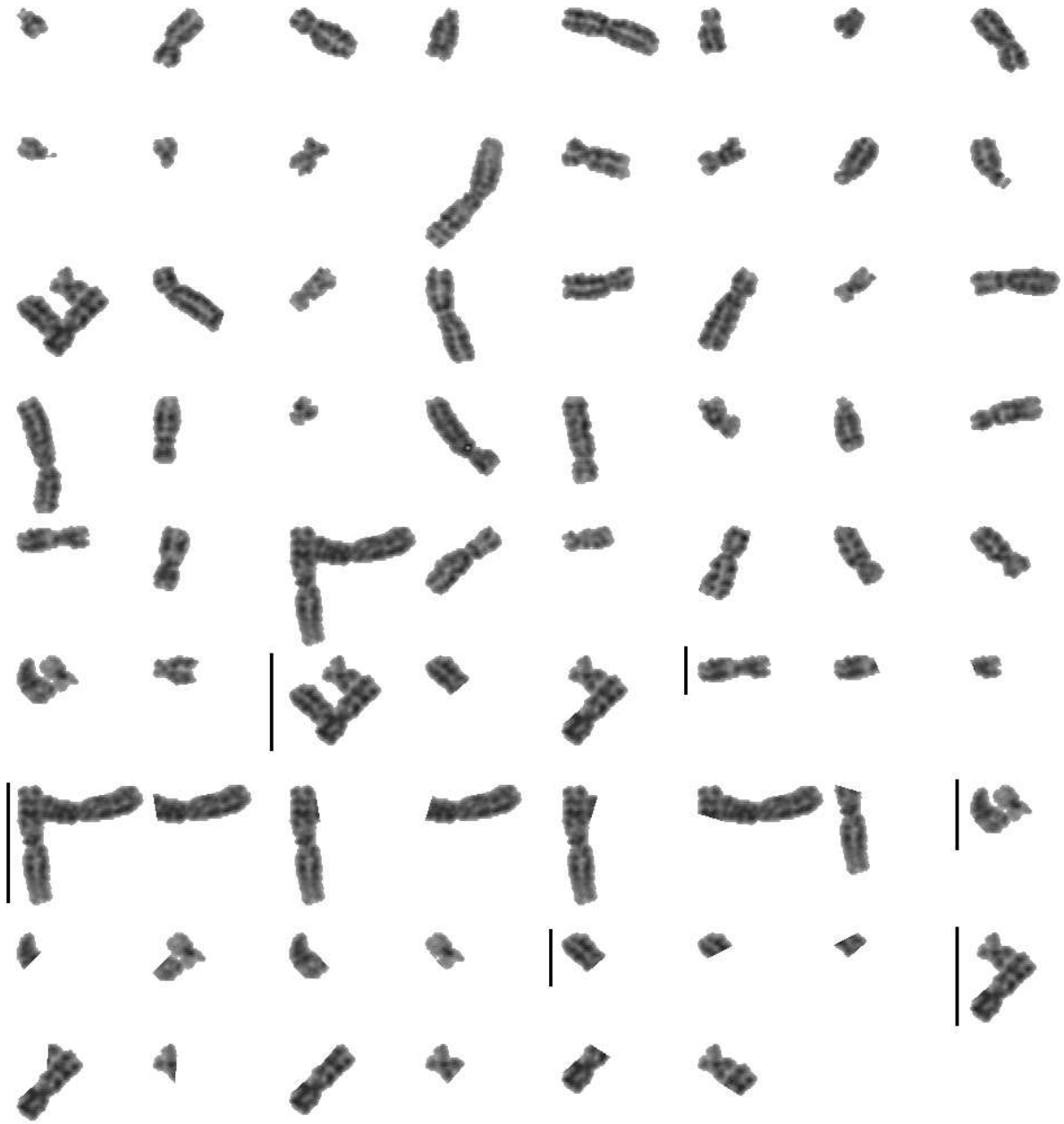

Figure 20: Variant creation, metaphase cell of Fig. 1. The components before the first bar represent the output of the clear phase. Three clusters were not safely recognized as clusters but deemed ambiguous. The components immediately after the bars are proposed for cutting by the relaxed rules in the ambiguous phase. The proposed cuts follow. They contain all correct ones. The classifier–selector correctly selects them later.

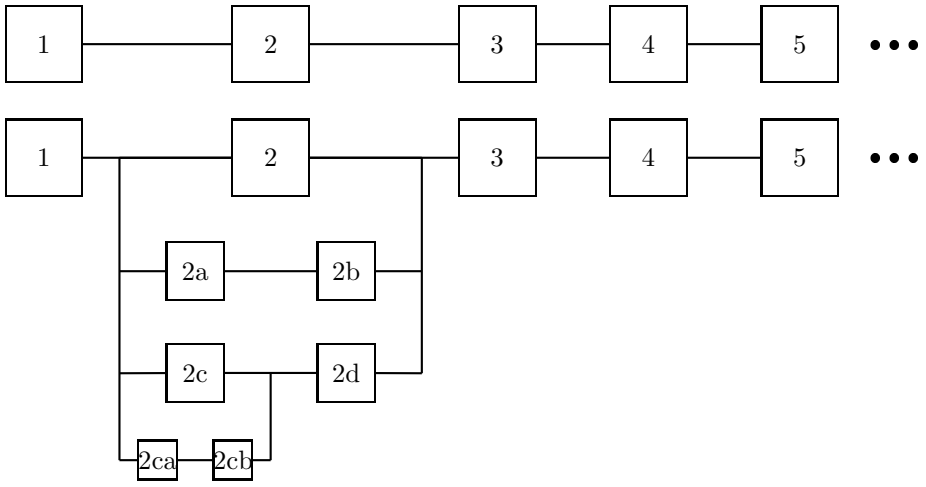

Figure 21: Section of an instance of the data structure for variant generation in the ambiguous phase. Top: components, bottom: various configurations of component 2 lead to local variants.

described in Section 2.4 and their organization in a data structure described above, see Fig. 21. From the data structure, all global variants with an appropriate number of components are extracted. These competing segmentations were classified with the accurate constrained classifier proposed in [36, 37] using the 30 features mentioned in Sect. 1.1 and compared as described in Sect. 2.3. The distribution parameters used for discrimination were computed from the kayograms of the Passau data set Pki, cf. Sect. 3.1. In order to compare segmentations with 45, 46, and 47 chromosomes, the scores (essentially the negative maximum log–likelihoods) of the segmentations with 45 chromosomes were enlarged by a factor of 46/45. The solution with the best score was retained.

Relying on the clear phase alone would force too complex rules and descriptors. Even then, to our experience, there would always be many counterexamples where they fail. Relying on the ambiguous phase alone would lead to a combinatorial explosion in the majority of cases. Therefore, we strive for a balance of the two.

2.6 Comparison of methods

The main novelties of our approach are

- segmentation of whole clinical cells by exploiting global context;
- the use of variant analysis for this purpose;
- the definition of prototypical shape elements such as “bridges,” “T,” “M,” ... by means of shape descriptors;
- the application of T–essential minima in order to detect meaningful concavities.

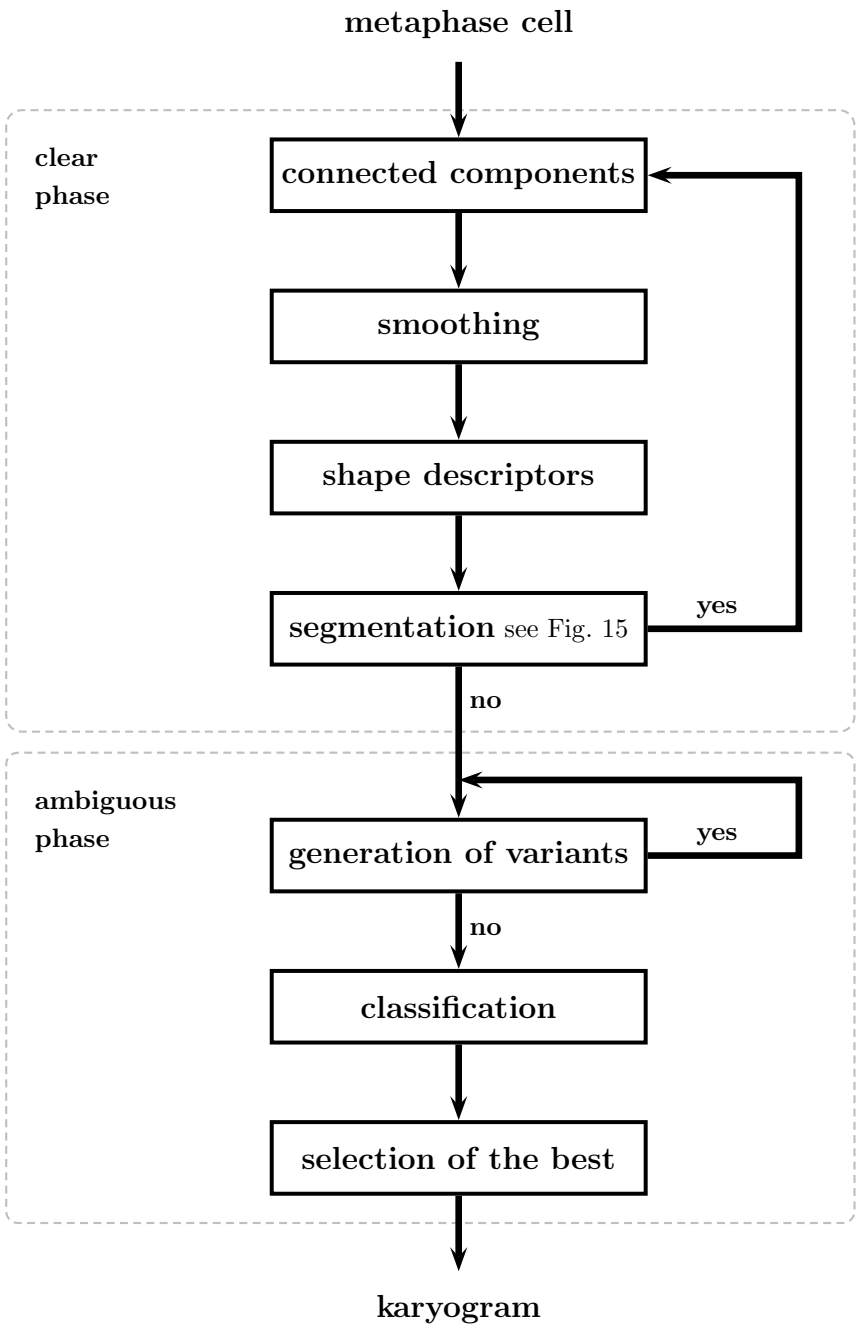

Figure 22: Schematic representation of the overall process. Input is a metaphase cell cleansed from small stains and interphase nuclei. The top two boxes in the so–called clear phase contain preprocessing steps. In the third box, the shape descriptors for the segmentation process are provided: boundary, polygonal approximation, skeleton. In the cycle of the clear phase, new components are iteratively split off until no rule further applies. The process now enters the ambiguous phase. In the top box, relaxed rules based on the same shape descriptors recursively generate tentative splits and local variants that are assembled to global variants. Their density values w.r.t. the statistical model of a karyogram are determined with a constrained type classifier as described in Sect. 2.3. In the bottom box, the global variant with the maximum likelihood (1) is selected as the proposed karyogram.

communication	Vanderheydt et al. [51]	Ji [14, 15]	Agam and Dinstein [1]	Lerner et al. [21]	Charters and Graham [5]	Urdiales et al. [50]	this
concave points	×	×	×	×			×
polyg approx.	×		×				×
skeleton		×				×	×
pale paths		×					
band pattern and classification driven				×	×	×	×
local context	×	×	×	×	×	×	×
global context							×

Table 2: Comparison of various methods for chromosome segmentation found in the literature. The tools in the first three lines relate to shape information, the following two to grey–value information

A schematic comparison of tools proposed in the literature is presented in Table 2. However, the fact that different authors use the same tools does not mean that they use them the same way. Vanderheydt et al. use an *approximating polygon* in order to detect meaningful convex and concave boundary points. Agam and Dinstein [1] use a *bounding polygon* in order to evaluate the fit of components obtained to prototypes of chromosome shapes. By contrast, we use the relative positions of the sides of an approximating polygon in order to describe and detect some of the different configurations.

Our treatment of overlaps in the clear phase is inspired by Ji's [15] who uses the nodes in the skeleton in combination with the boundary curvature as indicators. We use the skeleton also for detecting L–configurations. Urdiales et al. [50] exploit skeletons with branchings for matching templates of banding patterns with component endings, thus identifying valid chromosomes and cuts.

All authors use *local* pictorial context. To this end, Ji [14, 15] analyses the geometry of the vicinity of cut lines. Agam and Dinstein [1] exploit the shape of the whole component. A more efficacious way of using local context is to control the segmentation by information on the chromosome types. A cut is considered reasonable if the band patterns of the resulting objects are judged meaningful by a type classifier. Algorithmically, this leads to a feedback between segmentation and classification already proposed by Piper et al. [28]. For this purpose, Lerner et al. [21] use a chromosome–by–chromosome classifier in the form of a forward neural network that contains the information on the band profiles for selecting the proposed cut from a set of potential cuts. Charters and Graham [5] and Urdiales et al. [50] use a classifier based on templates of band pattern segments.

The present communication is the first to use also *global* pictorial context. We use our classifier designed in [33, 35, 37]. It contains a statistical model of the whole karyogram and is constrained to the correct numbers of chromosomes of each type, in normal cells and in several types of abnormal cells with one

missing or one excess chromosome. It thereby profits from the complete prior information available for classification of a cell as a guide for segmentation; see also Sect. 2.3. Our approach is applicable to clinical cells with large clusters of chromosomes.

3 Results

3.1 Data set

We use again the image data set Pki described in [36]. It consists of 971 Giemsa stained pro– and amnion and blood cells at pro– or metaphase compiled at the local cytogenetical institute in Passau/Germany for routine cytogenetic screening. All cells at late metaphase, characterized by split arms, were removed since our system is not designed to handle such cases. The remaining data set consists of 612 cells; we gave it the name Pki-3. It contains 28148 chromosomes in 21089 connected components. Sometimes a cut does not create a new component as the blocked overlap in Fig. 11 shows. Therefore, segmentation of the data set needs at least $28148 - 21089 = 7059$ cuts. Karyograms of all cells, manually produced by experienced cytogeneticists, are available.

3.2 Calibration and segmentation error rates

Unfortunately, the error rate of segmentation cannot be easily determined automatically. It would require the locations of the cuts performed by the cytogeneticists in all cells of Pki-3. This information is lost and it does not seem simple to make a system that restores it from the associated karyograms. Therefore, it was not possible to automatically calibrate the parameters, e.g. by means of cross validation. We, therefore, developed our system calibrating it manually with the aid of the first 40 cells of Pki-3 as a training set.

In order to assess the error rate of the segmentation process, we think it best to count the number of wrong cuts (in chromosomes) and the number of missing cuts. A cut that simultaneously dissects two chromosomes causes two errors. This occurs, e.g., if an overlap is cut diagonally. A wrong cut in an L counts as two errors since besides the wrong cut there is also a missing cut. An overlap cut as a three–component X counts as two errors because one chromosome is cut into three pieces. If a three–component X is cut as an overlap then no chromosome is dissected and one of the two components is correct. The other erroneously contains a copy of the overlap. This counts as one error unless the latter part, which consists of two chromosomes, is not dissected. In this case we have two errors.

All 612 cells of Pki-3 were tested and the errors of the automatically generated karyograms were manually determined, in questionable cases by comparison with the karyograms. Out of the 612 cells, there were 339 cells completely correctly segmented. In the remaining 273 cells, we manually counted about 200 wrong cuts in the clear and ambiguous phases, each, and about 300 missing cuts in the ambiguous phase, a total of about 700 errors. (There cannot be any missing cuts in the clear phase for logical reasons.)

With the exception of Ji [14, 15], the approaches proposed in the literature were not tested on large image data bases of pro– or metaphase cells. We note that determining the error rate of a large number of such images is a tedious manual task hard to automate, even in the presence of the associated karyograms. Ji reports error rates between 5 and 10% on data bases of several hundreds of cells but does not quantify his error rate in terms of numbers of missing and wrong cuts. The authors of [1, 21, 5, 50], too, offer error rates. However, there is no standard data base and all these authors use different and small data bases so that a fair comparison of error rates is not possible. Moreover, Charters and Graham [5] consider a set of *simulated overlaps*, only, and Lerner et al. [21] a data base of *pairs of touching* chromosomes.

Fig. 23 shows a histogram of the numbers of components before and after the clear phase. In most cases, the clear phase produces 40 or more components. This means that at most a few hundred global variants, only, had to be processed so that there was no combinatorial explosion.

4 Discussion

A method for completely automatic segmentation and classification of images of eukaryotic blood and amnion cells at pro– or early metaphase in their individual chromosomes was proposed. Segmentation consists of two phases. In the first, clear phase, shape elements that clearly indicate touchings or overlaps are detected and cut. In the second, all remaining subtle and ambiguous cases are treated. Here, variant analysis is employed to select the correct cuts from the proposed ones based on global statistical information on the karyogram. In both phases, prototypical shape elements are detected, in the first in a conservative, in the second in an offensive way. Our tests with a data base of more than 600 blood and amnion pro– and metaphase cells routinely used for clinical screening show that the complexity in the second phase can be controlled.

Major problems that had to be overcome are

- bent chromosomes hard to distinguish from two chromosomes touching at their tips;
- narrow centromeres that resemble two lightly touching chromosomes in a line;
- artificial holes in chromosomes that look like genuine holes between touching or overlapping chromosomes.

Main causes of errors are three or more chromosomes interfering at the same point. Our method does not apply to badly shaped chromosomes, for instance from chorion or bone marrow cells or cells at late metaphase where acrocentric and metacentric chromosomes appear Y– and X–shaped, respectively.

The system has two disadvantages. First, in order to describe the various shape elements in the two phases, about 170 real–valued parameters are needed. We were not able to optimally calibrate them. Second, the system needs a runtime between one minute in simple cases and a few hours in very complex cases on a 2 GHz processor. However, these drawbacks of the system are compensated by its accuracy.

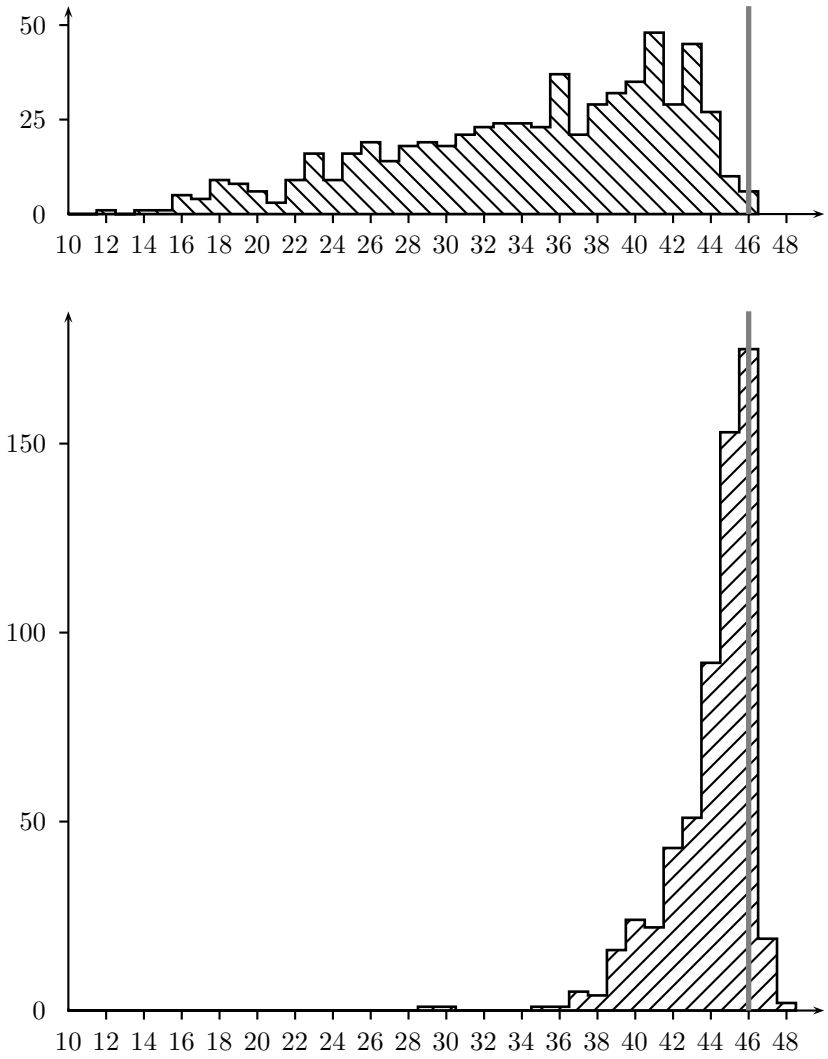

Figure 23: Histograms of the numbers of components in the cells of the data set Pki-3. Top before, bottom after the clear phase.

A possible further improvement may be the use of pale paths within components, [14, 15, 44]. Pale paths may first serve to detect some edge–to–edge touchings in thick components, see Fig. 18, although tight touchings often do not display a pale path. They may secondly be used for defining more precise cut paths which would enhance the supervised classification during the ambiguous phase. Light regions between bands might, however, confuse the algorithm.

Acknowledgements. We thank Herr Holger Harth of Praxis für medizinische Genetik, Neukirchen/Inn, for providing us with the image data set. We also thank Dr. Gernot Schreib for helping us with the program.

References

[1] Gady Agam and Its'hak Dinstein. Geometric separation of partially overlapping nonrigid objects applied to automatic chromosome classification. *IEEE Trans. Patt. Anal. Mach. Int.*, 19:1212–1222, 1997.

[2] Richard Bellman. On the approximation of curves by line segments using dynamic programming. *Comm. ACM*, 4:284, 1961.

[3] Jack E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[4] Richard G. Casey and Eric Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Trans. Patt. Anal. Mach. Int.*, 18:690–706, 1996.

[5] Graham C. Charters and Jim Graham. Trainable grey-level models for disentangling overlapping chromosomes. *Patt. Rec.*, 32:1335–1349, 1999.

[6] Yung-Sheng Chen and Wen-Hsing Hsu. A systematic approach for designing 2–subcycle and pseudo 1–subcycle parallel thinning algorithms. *Patt. Rec.*, 22:267–282, 1989. Comments in Patt. Rec. 25, 1545–1546 (1992).

[7] James George Dunham. Optimum uniform piecewise linear approximation of planar curves. *IEEE Trans. Patt. Anal. Mach. Int.*, 8:67–75, 1986.

[8] María Teresa Gallegos and Gunter Ritter. Outlier treatment: A new statistical method for automatic chromosome classification. In A. Prat and E. Ripoll, editors, *COMPSTAT Proceedings in Computational Statistics, 12th Symposium*, pages 55–56, Barcelona, 1996.

[9] María Teresa Gallegos and Gunter Ritter. Parameter estimation under ambiguity and contamination with the spurious model. *J. Multivariate Analysis*, 97:1221–1250, 2006.

[10] A.J.F. Griffiths, J.H. Miller, D.T. Suzuki, R.C. Lewontin, and W.M. Gelbart. *An Introduction to Genetic Analysis*. Freeman and Company, New York, 1993.

[11] V. Guillemin and A. Pollack. *Differential Geometry*. Prentice–Hall, Englewood Cliffs, New Jersey, 1974.

[12] C. Judith Hilditch. Linear skeletons from square cupboards. *Machine Intelligence*, 4:403–420, 1969.

[13] Judith Hilditch and Denis Rutovitz. Some algorithms for chromosome recognition. *Annals N.Y. Acad. Sci.*, 157(1):339–364, 1969.

[14] Liang Ji. Intelligent splitting in the chromosome domain. *Patt. Rec.*, 22:519–532, 1989.

[15] Liang Ji. Fully automatic chromosome segmentation. *Cytometry*, 17:196–208, 1994.

[16] Liang Ji and Jim Piper. Fast homotopy-preserving skeletons using mathematical morphology. *IEEE Trans. Patt. Anal. Mach. Int.*, 14(1):653–664, 1992.

[17] R.S. Ledley and F.H. Ruddle. Chromosome analysis by computer. *Scientific American*, 214(4):40–46, 1966.

[18] R.S. Ledley, F.H. Ruddle, J.B. Wilson, M. Belson, and J. Albarran. The case of the touching and overlapping chromosomes. In G.C. Cheng, R.S. Ledley, D.K. Pollock, and A. Rosenfeld, editors, *Pictorial Pattern Recognition*, pages 87–97. Thompson Book Company, Washington D.C., 1968.

[19] C.L. Lee and P.S.P. Wang. A simple and robust thinning algorithm. *Int. J. Patt. Rec. Art. Int.*, 13:357–366, 1999.

[20] Boaz Lerner. Toward a completely automatic neural-network-based human chromosome analysis. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 28:544–552, 1998.

[21] Boaz Lerner, Hugo Guterman, and Its'hak Dinstein. A classification-driven partially occluded object segmentation (CPOOS) method with application to chromosome analysis. *IEEE Transactions on Signal Processing*, 46:2841–2847, 1998.

[22] Sven Loncaric. A survey of shape analysis techniques. *Patt. Rec.*, 31:983–1001, 1998.

[23] Majed Marji and Pepe Siy. A new algorithm for dominant points detection and polygonization of digital curves. *Patt. Rec.*, 36:2239–2251, 2003.

[24] Nabil Jean Naccache and Rajjan Shinghal. An investigation into the skeletonization approach of Hilditch. *Patt. Rec.*, 17:279–284, 1984.

[25] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization*. Prentice–Hall, Englewood Cliffs, New Jersey, 1982.

[26] Theo Pavlidis. A thinning algorithm for discrete binary images. *Computer Graphics and Image Processing*, 13:142–157, 1980.

[27] Jim Piper and Erik Granum. On fully automatic feature measurement for banded chromosome classification. *Cytometry*, 10:242–255, 1989.

[28] Jim Piper, Erik Granum, Denis Rutovitz, and H. Ruttledge. Automation of chromosome analysis. *Signal Processing*, 2:203–221, 1980.

[29] Gunter Ritter. Quadratically asymmetric distributions and their application to chromosome classification. In A. Prat and E. Ripoll, editors, *COMPSTAT 1996 Proceedings in Computational Statistics, 12th Symposium, Short Communications*, pages 99–100, Barcelona, 1996.

[30] Gunter Ritter. Classification and clustering of objects with variants. In Wolfgang Gaul, Otto Opitz, and Martin Schader, editors, *Data Analysis, Scientific Modeling and Practical Application*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 41–50. Springer, Berlin, Heidelberg, 2000.

[31] Gunter Ritter and Karl Gaggermeier. Automatic classification of chromosomes by means of quadratically asymmetric statistical distributions. *Patt. Rec.*, 32:997–1008, 1999.

[32] Gunter Ritter and María Teresa Gallegos. Outliers in statistical pattern recognition and an application to automatic chromosome classification. *Patt. Rec. Lett.*, 18:525–539, 1997.

[33] Gunter Ritter and María Teresa Gallegos. A Bayesian approach to object identification in pattern recognition. In A. Sanfeliu et al., editor, *Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 418–421, Barcelona, 2000.

[34] Gunter Ritter and María Teresa Gallegos. Bayesian object identification: variants. *Journal of Multivariate Analysis*, 81:301–334, 2002.

[35] Gunter Ritter and Christoph Pesch. Polarity–free automatic classification of chromosomes. *Computational Statistics and Data Analysis*, 35:351–372, 2001.

[36] Gunter Ritter and Gernot Schreib. Profile and feature extraction from chromosomes. In A. Sanfeliu et al., editor, *Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 287–290, Barcelona, 2000.

[37] Gunter Ritter and Gernot Schreib. Using dominant points and variants for profile extraction from chromosomes. *Patt. Rec.*, 34:923–938, 2001.

[38] Azriel Rosenfeld. Connectivity in digital pictures. *J. ACM*, 17:146–160, 1970.

[39] Azriel Rosenfeld and Avinash C. Kak. *Digital Picture Processing*, volume 2. Academic Press, Orlando, San Diego, New York, Austin, Boston, London, Sydney, Tokyo, Toronto, second edition, 1982.

[40] Paul L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Trans. Patt. Anal. Mach. Int.*, 19:659–666, 1997.

[41] Paul L. Rosin. Assessing the behaviour of polygonal approximation algorithms. *Patt. Rec.*, 36:505–518, 2003.

[42] Yukio Sato. Piecewise linear approximation of plane curves by perimeter optimization. *Patt. Rec.*, 25:1535–1543, 1992.

[43] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, San Diego, New York, Berkeley, Boston, Sydney, Tokyo, Toronto, third edition, 1989.

[44] Hongchi Shi, Paul Gader, and Hongzheng Li. Parallel mesh algorithms for grid graph shortest paths with application to separation of touching chromosomes. *The Journal of Supercomputing*, 12:69–83, 1998.

[45] Frank Yeong-Chyang Shih and Owen Robert Mitchell. A mathematical morphology approach to euclidean distance transformation. *IEEE Trans. Image Processing*, 1:197–204, 1992.

[46] Satoshi Suzuki, Naonori Ueda, and Jack Sklansky. Graph–based thinning for binary images. *Internatinal Journal of Pattern Recognition and Artificial Intelligence*, 7:1009–1030, 1993.

[47] Antonio Torralba. Contextual modulation of target saliency. In *Advances in Neural Information Processing Systems*, 2001. http://books.nips.cc/nips14.html.

[48] M.K.S. Tso and J. Graham. The transportation algorithm as an aid to chromosome classification. *Patt. Rec. Lett.*, 1:489–496, 1983.

[49] M.K.S. Tso, P. Kleinschmidt, I. Mitterreiter, and J. Graham. An efficient transportation algorithm for automatic chromosome karyotyping. *Patt. Rec. Lett.*, 12:117–126, 1991.

[50] Cristina Urdiales García, Antonio Bandera Rubio, Fabián Arrebola Pérez, and Francisco Sandoval Hernández. A curvature–based multiresolution automatic karyotyping system. *Mach. Vision. Appl.*, 14:145–156, 2003.

[51] L. Vanderheydt, F. Dom, A. Oosterlinck, and H. Van Den Berghe. Two-dimensional shape decomposition using fuzzy subset theory applied to automated chromosome analysis. *Patt. Rec.*, 13:147–157, 1981.

[52] Peng-Yeng Yin. Ant colony search algorithms for optimal polygonal approximation of plane curves. *Patt. Rec.*, 36:1783–1797, 2003.

[53] S. Yokoi, J.I. Toriwaki, and T. Fukumura. An analysis of topological properties of digital binary pictures using local features. *Comput. Graph. Image Process.*, 4:63–73, 1975.

Appendix

A Algorithmic considerations

In this appendix, we discuss two basic algorithms necessary for our method.

A.1 Thinning and skeleton

The skeleton is a description of a 2D image by means of discrete curve patterns. The importance of the concept is underlined by the fact that many authors have devoted papers to this subject, see [12, 13, 26, 24, 39, 6, 16, 46, 19]. The skeleton contains plenty of information useful for establishing segmentation rules. Its crossings and branchings indicate X–configurations and touchings and it may serve as a basis for estimating the average width of a chromosome.

According to Hilditch and Rutovitz [13], see also Naccache and Shinghal [24], the skeleton of a connected object is the output of the recursive application of a thinning procedure that

– removes only boundary points,

– does not remove end points, that is points with no or only one 8–neighbor, and

– leaves the object 8–connected.

We employ the algorithm proposed in Rosenfeld and Kak [39], Sect. 11.2.3. It alternatingly runs through the northern, southern, eastern, and western 4–boundary points deleting them if they are not 8–end points and if the intersection of their 8–boundary with the image consists of exactly one 8–component.[4] A simple way of determining the number of these components is the 8–connectivity number introduced by Yokoi et al. [53]. Given a pixel (i, j) in an image, it is the number

$$\mathbf{CN8}(\text{image}, i, j) = \sum_{i=0}^{3} (1 - x_{2i}) \wedge (x_{2i+1} \vee x_{2i+2}).$$

Here, $x_0, \cdots, x_7$ are the 8–neighbors of the black pixel (i, j) ordered counterclockwise starting east. The expression is to be interpreted arithmetically and $\wedge$ and $\vee$ stand for the minimum and the maximum, respectively. If (i, j) is a 4–inner point then, plainly, CN8 = 0 and there is one component. Otherwise, CN8 is the number of 8–components. In the latter case, CN8 = 3 means branching and CN8 = 4 means crossing. The algorithm is shown in Table 3. The procedures **bool is4BoundaryPointN**(image,i,j), **bool is4BoundaryPointS**(image,i,j), **bool is4BoundaryPointE**(image,i,j), **bool is4BoundaryPointW**(image,i,j), and **bool isNot8EndPoint**(image,i,j) all receive a pixel (i, j) in a binary image. Their names are self–explanatory.

[4] This means that there are at least two 8–neighbors and that they are 8–connected, see [19].

Table 3: The skeleton algorithm

skeleton(image)

// INPUT: A two-dimensional matrix "image" that represents a binary picture

// OUTPUT: A binary image that represents the skeleton of the image

```
 1. stop ← false
 2. while not stop
 3.    for j ← 0 to width(image) for i ← 0 to height(image)
 4.       if isNot8EndPoint(image,i,j) and is4BoundaryPointN(image,i,j) and CN8(image,i,j)=1
 5.          then mark image[i][j] for deletion
 6.    delete all marked pixels
 7.    repeat 3.–6. with is4BoundaryPointS
 8.    repeat 3.–6. with is4BoundaryPointE
 9.    repeat 3.–6. with is4BoundaryPointW
10.    if no pixels were marked for deletion in 3.–9.
11.       then stop ← true
12. return image
```

A.2 Polygonal approximation

Like skeletonization, polygonal approximation is an effective way of reducing the complexity of the contour of a geometric object while conserving its main shape characteristics. The polygon smoothes the boundary and is insensitive to small perturbations such as the centromere or boundary noise.

The approximating polygon should have the property that each boundary pixel possesses a point on the polygon nearby while having as few vertices as possible. A large number of different methods have been proposed for this purpose, see, e.g., [40, 41, 23, 52]. Besides local heuristics, one of the earliest approaches was the application of dynamic optimization to a global approximation error such as the L_2– or the L_∞–norm or the perimeter difference, see [2, 7, 42]. Sometimes, the number of vertices or the maximum error are used as constraints and the optimization is w.r.t. the other parameter. The problem may also be viewed as a shortest path problem w.r.t a score (or "figure of merit") that combines the two. Given a finite line segment $[a, b]$ and a point x in the Euclidean plane, let $d([a, b], x)$ be their Euclidean distance. Let $\mathbf{b} = b_0, \ldots, b_{n-1}$ be the contour of the object. For any increasing k–tuple of boundary indices $\mathbf{v} = (v_0, \ldots, v_{k-1})$, let

$$d(\mathbf{v}) = \sum_{l<k} \sum_{v_l \le i < v_{l+1}} d^2([b_{v_l}, b_{v_{l+1}}], b_i).$$

The index l is meant mod k. Of course, this distance vanishes and is minimized over all k and all $\mathbf{v}$ by putting $k = n$ and $\mathbf{v} = 0..(n-1)$. In order to use vertices parsimoniously, we add to this value a multiple

λk of k for some $\lambda > 0$. With the weights

$$w(j, j') = \lambda + \sum_{j<i<j'} d^2([b_j, b_{j'}], b_i), \quad 0 \le j < j',$$

we, thus, define the loss of the polygon $\mathbf{v}$ as

$$w(\mathbf{v}) = \sum_{l<k} w(v_l, v_{l+1}) = \sum_{l<k} \Big\{ \lambda + \sum_{v_l \le i < v_{l+1}} d^2([b_{v_l}, b_{v_{l+1}}], b_i) \Big\}.$$

This expression is to be minimized w.r.t. all $\mathbf{v}$'s of length k and all numbers $k \le n$, a shortest path problem in the complete directed graph $0..(n-1)$ with weights $w(j, j')$, $0 \le j < j'$. If the start (= end) point is given then it is solved by Dijkstra's algorithm [25] in a full table of size n^2; it needs $\mathcal{O}(n^2)$ time steps, n being the boundary length. However, this point is unknown and, in principle, the optimization has to be repeated for each point in the boundary in order to find the best solution. This increases the complexity to $\mathcal{O}(n^3)$ and is best done by applying Floyd–Warshall's algorithm which computes the shortest path for all pairs of points. But this approach would mean too heavy a computational burden for our application. We resort to computing the broken line by applying Dijkstra's algorithm twice, first with a starting point of maximum convexity or concavity and then with a vertex on the opposite side of the broken line obtained as starting point. In most cases, it produces the optimal solution. An example is shown in Fig. 24.

Figure 24: Approximation by a polygon. The dots are the locations of the vertices