

一种确定染色体图象着丝点位置的方法^①

张凯丽, 刘辉

(昆明理工大学信息工程与自动化学院, 云南昆明 650051)

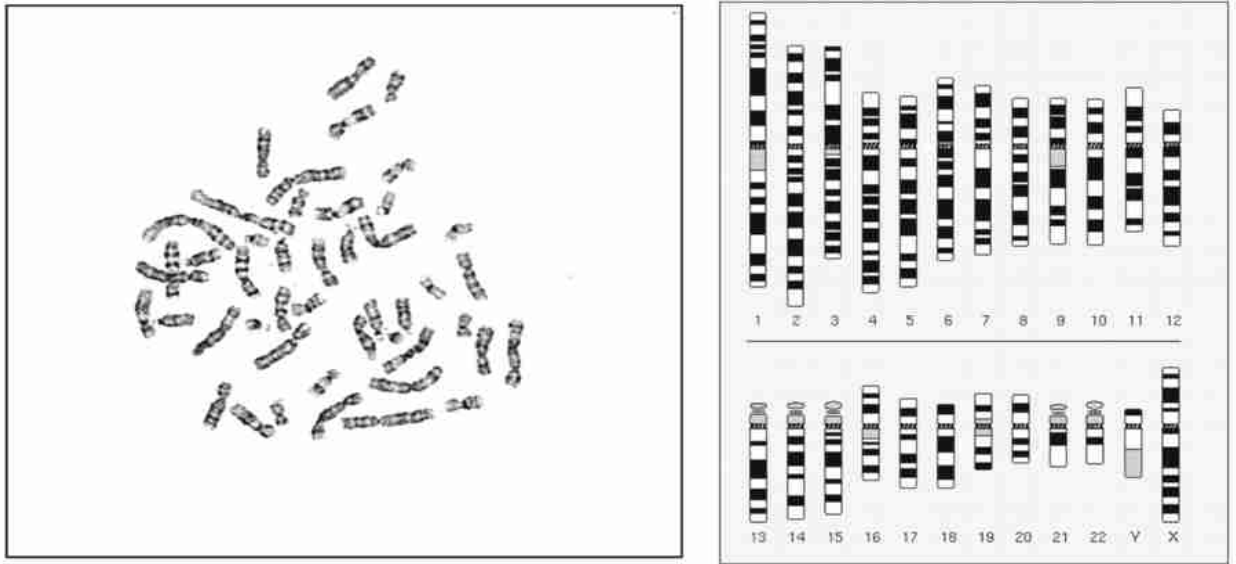
摘要 在染色体自动识别(或自动核型分析)中,染色体着丝粒指数为其分类的重要特征.文中给出了求解着丝粒位置的一种算法,并对算法的结果进行了分析,证明了算法的正确性与有效性.从而为染色体自动识别与分析提供了重要的特征依据.

关键词: 医学图像分析;染色体;着丝粒指数;骨架;最近邻插值;数学形态学

中图分类号: TP391.4 文献标识码: A 文章编号: 1007-855X(2001)03-110-04

0 前言

在用计算机自动识别染色体(或进行核型分析)时,染色体着丝粒指数是一个重要的特征^[1,2].而着丝指数可以通过求染色体的短臂长和总长来实现,这就涉及到对染色体着丝点位置的确定问题.由于染色体着丝粒所在处的特征表现为凹陷(如图 1 a 所示),在该处,染色体的宽度取得局部极小值,由于染色体着丝粒的这个特征,所以可以通过求取染色体宽度极小值位置来确定染色体着丝点的位置.



(a) 染色体图片(560×764×8bit)

(b) 染色体组型模式图

图 1 染色体着丝粒所在处的特征表现

图 1 (b) 中标明了染色体组型中染色体着丝点位置(中间斜线处)、随体的有无(如 13、14、15 和 21、22 有,其它没有)以及染色体的长度这些特征.这些特征是分析和识别染色体的关键特征.其中的黑白相间的带是染色体的显带模式特征(图中显示的为 G-带技术时的带模式),是识别染色体的另一个特征.而染色体骨架和边界基本上含有染色体的所有形状和形态信息^[3],如图 2 (c)、图 2 (d) 所示.所以很自然地想到用骨架和染色体区域边界来确定着丝点位置.

在染色体骨架确定以后,结合骨架和染色体边界得到染色体在各处的跨度(宽度),宽度最小处即为着丝点位置.着丝指数(短臂长与总长之比)是染色体的重要的形态特征,是识别染色体的重要特征之一.

① 收稿日期: 2000-10-31;

第一作者简介: 张凯丽(1975.10~),女,硕士研究生,主要研究数字图像处理/分析.

所以染色体着丝点位置的确定对染色体识别相当重要.

如图 2 所示, 图 2 (a) 为从分裂中期染色体显微图片 (如图 1 (a) 所示) 中分割出来的单个染色体的灰度图像, 图 2 (b) 为经过二值化, 形态学处理后的染色体区域, 图 2 (c) 为二值区域的边界, 图 2 (d) 为区域细化的结果, 即染色体的骨架. 图 2 (e) 给出了染色体骨架和边界间的关系, 而图 2 (f) 则是我们本文所采用的算法的示意图, 线段表示该处的宽度(垂直于骨架)^[4, 5].

由于着丝点位置的重要性, 所以要找出一合适的算法来确定. 本文采用求宽度极小值的方法来求取染色体着丝粒的位置. 文中对染色体各处宽度的求解给出了详细的算法以及算法的结果, 并讨论了算法的正确性, 有效性等.

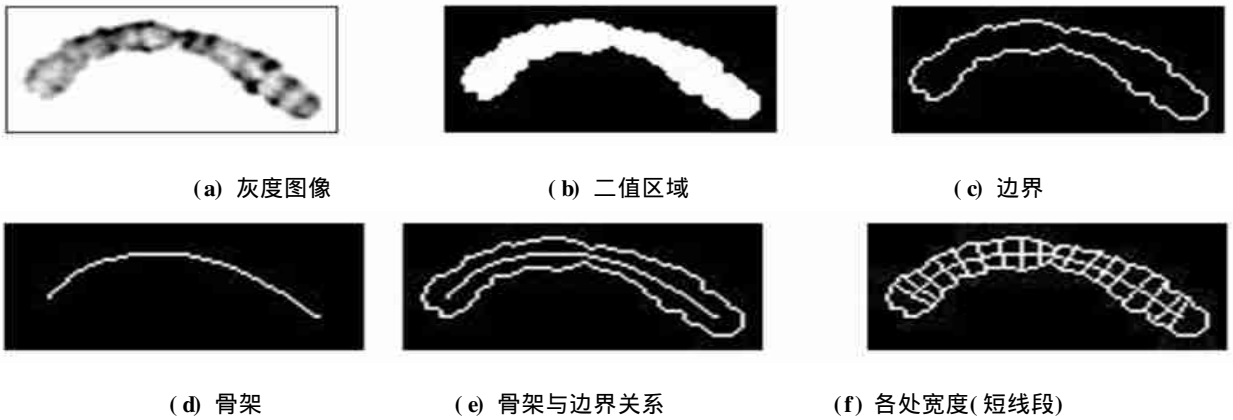


图 2 染色体显微图片

1 理论分析及算法

根据引言中所述, 我们可以用两步的方法来解决染色体各处的宽度.

(1) 由骨架得到骨架上各点处的斜率. 首先要找出染色体骨架的端点, 从该点开始搜索(沿骨架上各点的 8-邻域)骨架上的近邻点, 然后用这些点的 x, y 坐标来计算骨架在该点处的斜率. 离散情况下计算曲线的斜率有多种方法.

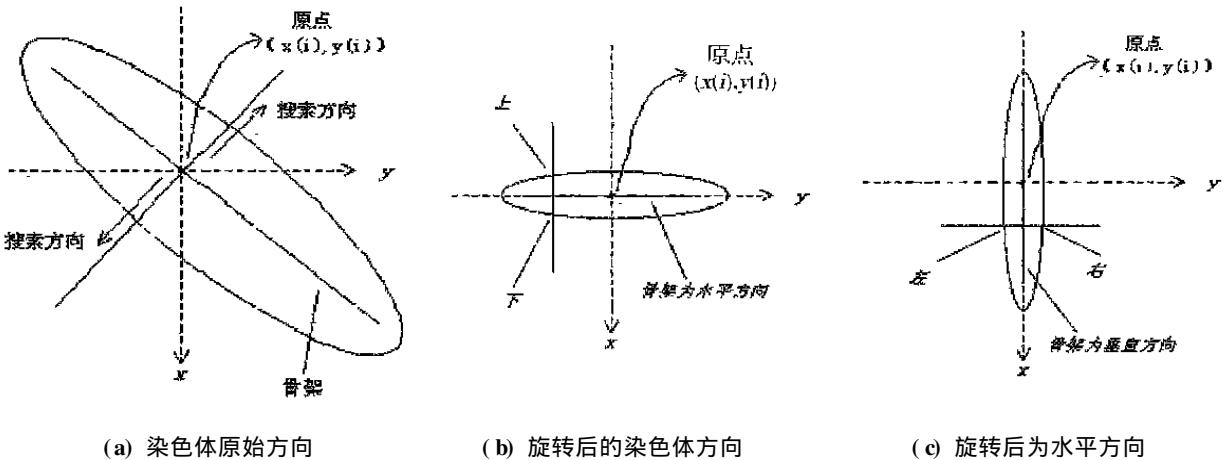


图 3 染色体方向

(2) 结合骨架点的斜率和染色体区域来逐一求染色体各处的宽度. 在骨架点处做垂直于骨架的直线, 跨过染色体区域的线段的长度即为染色体在该处的宽度. 该宽度的求解也可以有多种方法. 如, 第一, 根据染色体该处斜率, 将染色体旋转, 使该点处的倾斜为 90 度(即竖直如图 3 (b) 所示), 这时直接用水平的或竖直的最大值减最小值即得该点处的宽度. 或者, 第二, 如图 3 (a) 中所示, 以骨架上某一点 $(x(i), y(i))$ 为起始点 (x_0, y_0) , 设该处的斜率为 k , 从该点开始搜索, 方向如图 3 (a) 所示的搜索方向,

由于染色体的坐标是离散的, 直接在垂直于骨架的方向上搜索, 可能会出现跳过某点的情况, 所以我们首先分析染色体骨架各点处斜率的大小, 然后直接沿 x 或 y 方向(具体方向在以后算法中描述), 最后用最近邻插值的方法来得到搜索点的另一坐标(y 或 x)^[6]. 即 $x = x_0 + i$ 或 $y = y_0 + i$ (均为非零整数), 则 $l = ki, 0 < k < l, l = kj, k > 1; l = i, k = 1$; 坐标为 (x, y) .

对于每一沿搜索方向得到的点 (x, y) 都要进行边界检查, 看是否超出边界(看该点是否还在染色体区域内, 即是否为“1”), 若到就停止, 否则继续每搜索一次宽度加 1, 最后就得到染色体在该点处的宽度. 我们用如图 4 所示的网格来解释第二种方法(每个交叉点为一网格点即像素点, 其它处的点为插值点, 1, 2, 3, 4, 5, 6 表示染色体骨架上点处的方向(斜率), 描述了六种情况, 即斜率绝对值大于、等于和小于 1, 等于 0 或 ∞ 的情况图中没有标出). 具体算法如下算法 1.

算法 1:

- 第一步: 初始化. $i = 2$, 初始点 $x = x(i), y = y(i)$;
- 第二步: 求该点处的斜率, 设为 k , 并判断 k 的取值范围;
- 第三步: 求各处的宽度 $width(i)$, 宽度初始化 $width(i) = 1$, 及循环次数 $j = 1$.

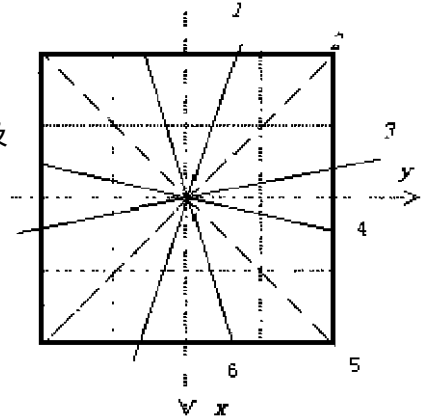


图 4 放大的网格 (骨架点方向的各种情况 1~ 6)

- (1) $k = 0$ (即 $\Delta y = 0$), $width(i) =$ 竖直跨度;
- (2) $k = \pm \infty$ (即 $\Delta x = 0$), $width(i) =$ 水平跨度;
- (3) $k < -1$ 为图中直线 3 的情况, 则沿 x 方向(正、负两个方向). $x = x + 1$ (或 $x = x - 1$);
如果 $|1/k| \times j - INT(|1/k| \times j) \geq 0.5, y = y + 1$ (或 $y = y - 1$), 否则 $y = y$ (四舍五入);
- (4) $-1 < k < 0$, 为图中直线 1 的情况, 沿 y 方向(正、负两个方向). $y = y + 1$ (或 $y = y - 1$);
如果 $|k| \times j - INT(|k| \times j) \geq 0.5, x = x + 1$ (或 $x = x - 1$), 否则 $x = x$ (四舍五入);
- (5) $k = -1$, 图中直线 2 的情况, $y = y + 1$ (或 $y = y - 1$);
- (6) $-1 < k < 0$, 其垂线为图中直线 4 的情况, 则沿 y 方向(正负两个方向). $y = y + 1$ (或 $y = y - 1$);
如果 $|k| \times j - INT(|k| \times j) \geq 0.5, x = x - 1$ (或 $x = x + 1$), 否则 $x = x$ (四舍五入);
- (7) $k > -1$, 其垂线为图中直线 6 的情况, 沿 x 方向(正、负两个方向). $x = x + 1$ (或 $x = x - 1$);
如果 $|1/k| \times j - INT(|1/k| \times j) \geq 0.5, y = y - 1$ (或 $y = y + 1$), 否则 $y = y$ (四舍五入);
- (8) $k = 1$, 为图中直线 5 的情况, $x = x - 1$ (或 $x = x + 1$), $y = y + 1$ (或 $y = y - 1$).

第四步: 判断 (x, y) 是否在染色体区域内, 若在宽度 $width(i) = width(i) + 1, j = j + 1$, 并继续; 否则结束. 得到 $(x(i), y(i))$ 处染色体的宽度;

第五步: 对于骨架上每个点重复以上各步 ($i = 2, 3, \dots, leng - 1, leng$ 为染色体骨架长度), 得到染色体各点处的宽度;


第六步: 结束.

2 算法结果及分析

以图 2 中的染色体为例, 用以上算法得到的结果为 $width(i), (i = 2, \dots, 74)$, 第 1 和最后 75 个均为端点. 宽度为(以像素为单位).

着丝粒位置在 41 或 40 处, 长臂长为 76, 所以短臂长为 $76 - 41 = 35$, 着丝指数为 $35/76 = 0.461$. 这说明了该算法的有效性, 因为实际值与此不相上下. 从表 1 中也可以看出, 着丝粒处的线段长度相对最小. 对于其它的染色体, 我们也用该算法做了实验, 从结果中我们得到, 如果细化后的骨架无分枝, 该算法就能够正确的执行, 并得到有效的结果.

表 1 结 果

染色体的宽度	位置	实际结果图(间隔 4)
(11, 11, 9, 9, 10, 10, 8, 8, 12, 11,	2- 11	
8, 9, 9, 10, 10, 11, 11, 11, 11, 10,	12- 21	
11, 10, 10, 11, 11, 11, 12, 12, 12, 13,	22- 31	
13, 13, 12, 11, 10, 9, 9, 8, 7, 7,	32- 41	
9, 8, 8, 10, 8, 8, 9, 8, 8, 9,	42- 51	
8, 9, 9, 8, 10, 11, 10, 10, 12, 12,	52- 61	
12, 9, 10, 14, 9, 10, 10, 8, 8, 9, 9, 9, 11)	62- 74	

而对于骨架有分枝存在或骨架不光滑的染色体(如图 5 所示), 该算法不能正确的执行, 需要改进. 在算法的一开始先判断染色体骨架是否有分枝存在和是否光滑, 通过求骨架的端点数目来得到是否有分枝存在, 如果端点数目为 2, 则没有分枝, 用该算法进行求解; 若为大于 2, 则有分枝存在, 需要先对分枝进行处理. 针对图 1 中染色体图片, 不光滑有如图所示的情况, 还有不光滑点向下, 向右的情况, 共有六条染色体存在这些情况, 有 3 条染色体的骨架有分枝存在, 有 1 条染色体两种情况都有.

3 结 论

由于染色体的形态各异, 有粘连, 交叉, 折叠或扭曲的存在, 这就给我们的工作带来麻烦. 首先, 如果染色体弯曲度大于 90 度并且弯曲不在着丝点处, 那么染色体就可能被折叠. 第二, 如果染色体看起来具有两个着丝点, 那么它也可能是扭曲的. 如果有折叠或扭

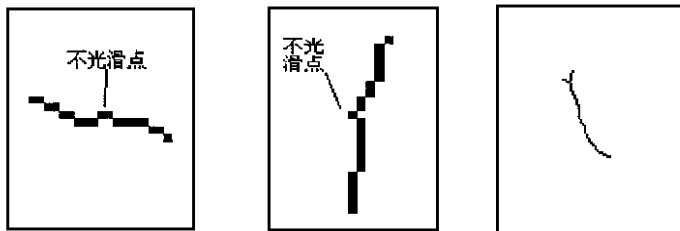


图 5 特殊情况(不光滑或有分枝, 最小的块表示一个像素)

曲存在, 我们在寻找着丝粒位置时, 就会有一定的困难. 我们文中所给出的算法对于无折叠扭曲, 并且其骨架不存在分枝和不光滑现象的染色体来说时可行的, 也时有效的. 对于分枝的存在我们可以去分枝, 对于不光滑的骨架, 我们可以将其平滑, 这样问题就会解决了. 正确的寻找染色体着丝粒位置, 为正确的识别染色体提供了有效的特征. 而对于折叠和扭曲的存在, 我们需要寻找其它的新的方法来解决.

参考文献:

- [1] 左汲, 黎立瑾. 医学细胞生物学[M]. 上海: 上海医科大学出版社, 1999. 56~ 64.
- [2] 寿天德, 徐耀忠. 现代生物学导论[M]. 北京: 中国科学技术大学出版社, 1998, 159~ 165.
- [3] 徐建华. 图像处理与分析[M]. 北京: 科学出版社, 1992, 165~ 254.
- [4] 龚炜, 石青云, 程德民. 数字空间中的数学形态学——理论及其应用[M]. 北京: 科学出版社, 1997. 1~ 30.
- [5] Kenneth R. Castleman. 数字图像处理[M]. Prentic Hall 出版公司, 北京: 清华大学出版社, 1998. 406~ 412.
- [6] 李庆扬, 王超能, 易大义. 数值分析[M]. 武汉: 华中理工大学出版社, 1996. 19~ 58.

A Method of Obtaining the Centromere Index In Chromosome Images

ZHANG Kai-li, LIU Hui

(The Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650051, China)

Abstract In the automatic karyotype analyzing system, centromere index is a very important feature to classify and pair the chromosomes. In this paper we propose a new algorithm to compute the position of the centromere of a chromosome, analyze the result, and justify the validity of the algorithm. So we can have a useful feature in the succeeding analysis.

Key words: medical image analysis; chromosome; centromere index; skeleton; nearest interpolation; mathematics morphology