



## [12] 发明专利说明书

专利号 ZL 01806580.5

[45] 授权公告日 2009 年 7 月 15 日

[11] 授权公告号 CN 100514364C

[22] 申请日 2001.2.23 [21] 申请号 01806580.5

[30] 优先权

[32] 2000.3.14 [33] US [31] 09/524,797

[86] 国际申请 PCT/US2001/005757 2001.2.23

[87] 国际公布 WO2001/069529 英 2001.9.20

[85] 进入国家阶段日期 2002.9.13

[73] 专利权人 英特尔公司

地址 美国加利福尼亚州

[72] 发明人 赖纳·W·林哈特

阿克塞尔·韦尼克

审查员 石 清

[74] 专利代理机构 永新专利商标代理有限公司

代理人 王 英

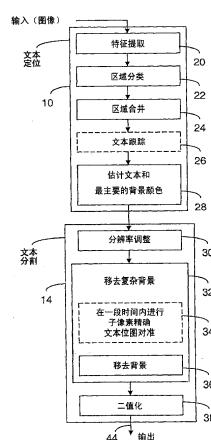
权利要求书 4 页 说明书 30 页 附图 6 页

## [54] 发明名称

在数字图像中定位文本的方法和装置

## [57] 摘要

在一些实施例中，本发明包括一种在数字图像中定位文本的方法。该方法包括：将一个数字图像按比例变换为多种分辨率的若干个图像，和按照像素是否是一个文本区域的一部分来对所述多种分辨率下的像素进行分类。该方法还包括：将各比例进行整合以生成一个比例整合突出图形，和使用该突出图形来生成初始文本界定方框，这是通过将方框从包括至少一个像素的像素矩形扩展成包括由至少一个与所述矩形邻接的像素构成的组来完成的，其中这些组与第一阈值之间具有特定关系。将初始文本界定方框合并。在其他实施例中，一种方法包括：按照像素是否是文本区的一部分来对像素进行分类，生成初始文本界定方框，和合并初始文本界定方框，其中所述合并包括生成具有自适应阈值的水平投影轮廓和具有自适应阈值的垂直投影轮廓。



1. 一种在数字图像中定位文本的方法，包括：

将一个数字图像按比例变换为多种分辨率的若干个图像；

对所述多种分辨率下的像素进行分类以确定所述像素是否是文本区域的一部分；

将所有比例下被分类为文本区域一部分的像素进行整合以生成一个比例整合突出图形；

使用该突出图形来生成初始文本界定方框，这是通过将方框从包括至少一个像素的像素矩形扩展成包括由至少一个与所述矩形邻接的像素构成的组来完成的，其中当所述组中任何给定组的平均亮度超过第一阈值时，将初始文本界定方框扩展成包括该给定组在内；和

合并初始文本界定方框。

2. 根据权利要求 1 的方法，其中所述组包括与所述矩形邻接的一行或一列，并且该矩形从 1 像素乘 1 像素的矩形开始。

3. 根据权利要求 1 的方法，其中所述突出图形的分辨率与所述数字图像被按比例变换为多种分辨率之前的分辨率相同。

4. 根据权利要求 1 的方法，其中所述数字图像是数字视频图像的一部分，并且初始文本界定方框的合并包括生成具有自适应阈值的水平投影轮廓和具有自适应阈值的垂直投影轮廓。

5. 根据权利要求 4 的方法，其中水平投影轮廓的自适应阈值是水平投影轮廓的最小和最大值的函数，垂直投影轮廓的自适应阈值是垂直投影轮廓的最小和最大值的函数。

6. 根据权利要求 1 的方法，其中初始文本界定方框的合并包括重复地执行水平分割算法和垂直分割算法。

7. 根据权利要求 6 的方法，其中水平分割算法包括将文本界定方框在顶部和底部进行扩展，扩展的量为：原始文本方框高度的一半和最大可能文本高度的一半之中的较小值。

8. 根据权利要求 1 的方法，还包括计算边缘方位以识别多种分辨率

下的图像特征。

9. 根据权利要求 1 的方法，还包括使用基于特征谱的跟踪来向前和向后识别包括文本对象中的文本的帧，该操作是从一个已经通过基于图像的方法从中识别出所述文本的帧开始的。

10. 根据权利要求 1 的方法，还包括通过生成文本和包围文本的非文本部分的颜色直方图来估计图像中的文本的颜色。

11. 一种在数字图像中定位文本的装置，包括：

用于将一个数字图像按比例变换为多种分辨率的若干个图像的模块；

用于对所述多种分辨率下的像素进行分类以确定所述像素是否是文本区域的一部分的模块；

用于将所有比例下被分类为文本区域一部分的像素进行整合以生成一个比例整合突出图形的模块；

用于使用该突出图形，通过将方框从包括至少一个像素的像素矩形扩展成包括由至少一个与所述矩形邻接的像素构成的组来生成初始文本界定方框的模块，其中当所述组中任何给定组的平均亮度超过第一阈值时，将初始文本界定方框扩展成包括该给定组在内；和

用于合并初始文本界定方框的模块。

12. 根据权利要求 11 的装置，其中所述组包括与所述矩形邻接的一行或一列，并且该矩形从 1 像素乘 1 像素的矩形开始。

13. 根据权利要求 11 的装置，其中所述突出图形的分辨率与所述数字图像被按比例变换为多种分辨率之前的分辨率相同。

14. 根据权利要求 11 的装置，其中所述数字图像是数字视频图像的一部分，并且所述用于合并初始文本界定方框的模块生成具有自适应阈值的水平投影轮廓和具有自适应阈值的垂直投影轮廓。

15. 根据权利要求 14 的装置，其中水平投影轮廓的自适应阈值是水平投影轮廓的最小和最大值的函数，垂直投影轮廓的自适应阈值是垂直投影轮廓的最小和最大值的函数。

16. 根据权利要求 11 的装置，其中所述用于合并初始文本界定方框的模块重复地执行水平分割算法和垂直分割算法。

17. 根据权利要求 16 的装置，其中水平分割算法包括将文本界定方框在顶部和底部进行扩展，扩展的量为：原始文本方框高度的一半和最大可能文本高度的一半之中的较小值。

18. 根据权利要求 11 的装置，还包括：用于计算边缘方位以识别多种分辨率下的图像特征的模块。

19. 根据权利要求 11 的装置，还包括：用于使用基于特征谱的跟踪来向前和向后识别包括文本对象中的文本的帧的模块，其中所述识别是从一个已经通过基于图像的方法从中识别出所述文本的帧开始的。

20. 根据权利要求 11 的装置，还包括：用于通过生成文本和包围文本的非文本部分的颜色直方图来估计图像中的文本的颜色的模块。

21. 一种在数字图像中定位文本的方法，包括：

对像素进行分类以确定所述像素是否是文本区域的一部分；

生成初始文本界定方框，这是通过将方框从包括至少一个像素的像素矩形扩展成包括由至少一个与所述矩形邻接的像素构成的组来完成的，其中当所述组中任何给定组的平均亮度超过第一阈值时，将初始文本界定方框扩展成包括该给定组在内；和

合并初始文本界定方框，其中所述合并包括生成具有自适应阈值的水平投影轮廓和具有自适应阈值的垂直投影轮廓。

22. 根据权利要求 21 的方法，其中水平投影轮廓的自适应阈值是水平投影轮廓的最小和最大值的函数，垂直投影轮廓的自适应阈值是垂直投影轮廓的最小和最大值的函数。

23. 根据权利要求 21 的方法，其中初始文本界定方框的合并包括重复地执行水平分割算法和垂直分割算法。

24. 根据权利要求 23 的方法，其中水平分割算法包括将文本界定方框在右边和左边进行扩展，扩展的量为：原始文本方框高度的一半和最大可能文本高度的一半之中的较小值。

25. 根据权利要求 23 的方法，其中垂直分割算法包括将文本界定方框在顶部和底部进行扩展，扩展的量为：原始文本方框高度的一半和最大可能文本高度的一半之中的较小值。

26. 一种在数字图像中定位文本的装置，包括：

用于对像素进行分类以确定所述像素是否是文本区域的一部分的模块；

用于通过将方框从包括至少一个像素的像素矩形扩展成包括由至少一个与所述矩形邻接的像素构成的组来生成初始文本界定方框的模块，其中当所述组中任何给定组的平均亮度超过第一阈值时，将初始文本界定方框扩展成包括该给定组在内；和

用于合并初始文本界定方框的模块，其中该模块生成具有自适应阈值的水平投影轮廓和具有自适应阈值的垂直投影轮廓。

27. 根据权利要求 26 的装置，其中水平投影轮廓的自适应阈值是水平投影轮廓的最小和最大值的函数，垂直投影轮廓的自适应阈值是垂直投影轮廓的最小和最大值的函数。

28. 根据权利要求 26 的装置，其中所述用于合并初始文本界定方框的模块重复地执行水平分割算法和垂直分割算法。

29. 根据权利要求 28 的方法，其中水平分割算法包括将文本界定方框在右边和左边进行扩展，扩展的量为：原始文本方框高度的一半和最大可能文本高度的一半之中的较小值。

30. 根据权利要求 28 的装置，其中垂直分割算法包括将文本界定方框在顶部和底部进行扩展，扩展的量为：原始文本方框高度的一半和最大可能文本高度的一半之中的较小值。

## 在数字图像中定位文本的方法和装置

### 本发明背景

#### 技术领域

本发明一般地涉及图像中的文本定位和/或分割。

#### 背景技术

目前在文本识别方面所做的工作主要集中于印刷或手写文件中的字符的光学识别（称为光学字符识别（OCR）），以满足对于办公自动化系统的文件阅读器的巨大市场需求。这些系统已经达到了很高的成熟度。在工业应用中可以看到进一步的文本识别工作，其中的大部分集中在非常窄的应用领域。一个例子是汽车牌照的自动识别。

人们已经提出一些关于检测复杂图像和视频中的文本以及从复杂图像和视频中提取文本的方案。但是，从这些方案的描述中可以看出，每种方案均在某些方面不具有通用性。另外，其中一些方案不包括将已定位的文本从其背景移去。例如，Etemad K 等，“Page Segmentation Using Decision Integration and Wavelet Packets”，Proceedings of the IAPR International Conference on Pattern Recognition，Jerusalem，Oct. 9-13, 1994, vol. 2, conf. 12, pp. 345-49，描述了一种识别方法，其中在文件图像中的文本、图像和图形区域被视为三种不同的“纹理（texture）”类别。通过在邻近块上传播和整合软局部判决来执行分割。但是，Etemad 等人没有公开如在本公开中所描述的突出图形或者直方图。作为另一个例子，Jain A.K. “Fundamentals of Digital Image Processing”，1989, Prentice Hall, chapter 9, page 412, 描述了基于区域的分割技术，涉及识别图像中具有类似特征的各种区域。图像被划分为灰度级恒定的基本区域。类似的相邻的区域被合并。但是，Jain 并没有公开如在本公开中所描述的突出图形或者直方图。

作为另一个例子，Sato, T.等，“Video OCR for Digital News Archive”，Proceedings 1998 IEEE Int'l Conf. on Content-Based Access of Image and Video Database, Bombay, India, 3 January 1998，描述了一种在大的数字新闻视频文档中定位感兴趣的题目的技术。这篇文章描述了应用内插滤波器、多帧整合以及四滤波器组合。通过基于识别的分割方法进行对字符的分割，中间的字符识别结果被用于改善分割。但是，Sato 等没有公开在本公开中所描述的使用突出图形或者直方图。

因此，人们需要一种通用的文本定位和分割方法。

## 发明内容

在一些实施例中，本发明包括一种在数字图像中定位文本的方法。该方法包括将一个数字图像按比例变换为多种分辨率的若干个图像，对所述多种分辨率下的像素进行分类以确定所述像素是否是文本区域的一部分，以及将各比例进行整合以生成一个比例整合突出图形。所述方法还包括使用该突出图形来生成初始文本界定方框，这是通过将方框从包括至少一个像素的像素矩形扩展成包括由至少一个与所述矩形邻接的像素构成的组来完成的，其中这些组与第一阈值之间具有特定关系，以及合并初始文本界定方框。还描述和要求保护了其它实施方式。

根据本发明的一个方面，提供了一种在数字图像中定位文本的方法，该方法包括：将一个数字图像按比例变换为多种分辨率的若干个图像；对所述多种分辨率下的像素进行分类以确定所述像素是否是文本区域的一部分；将所有比例下被分类为文本区域一部分的像素进行整合以生成一个比例整合突出图形；使用该突出图形来生成初始文本界定方框，这是通过将方框从包括至少一个像素的像素矩形扩展成包括由至少一个与所述矩形邻接的像素构成的组来完成的，其中基于所述组中任何给定组的平均亮度与第一阈值之间的比较结果来确定是否将初始文本界定方框扩展成包括该给定组在内；和合并初始文本界定方框。

根据本发明的一个方面，提供了一种在数字图像中定位文本的装置，该装置包括：用于将一个数字图像按比例变换为多种分辨率的若干个图像

的模块；用于对所述多种分辨率下的像素进行分类以确定所述像素是否是文本区域的一部分的模块；用于将所有比例下被分类为文本区域一部分的像素进行整合以生成一个比例整合突出图形的模块；用于使用该突出图形，通过将方框从包括至少一个像素的像素矩形扩展成包括由至少一个与所述矩形邻接的像素构成的组来生成初始文本界定方框的模块，其中基于所述组中任何给定组的平均亮度与第一阈值之间的比较结果来确定是否将初始文本界定方框扩展成包括该给定组在内；和用于合并初始文本界定方框的模块。

根据本发明的一个方面，提供了一种在数字图像中定位文本的方法，该方法包括：对像素进行分类以确定所述像素是否是文本区域的一部分；生成初始文本界定方框，这是通过将方框从包括至少一个像素的像素矩形扩展成包括由至少一个与所述矩形邻接的像素构成的组来完成的，其中基于所述组中任何给定组的平均亮度与第一阈值之间的比较结果来确定是否将初始文本界定方框扩展成包括该给定组在内；和合并初始文本界定方框，其中所述合并包括生成具有自适应阈值的水平投影轮廓和具有自适应阈值的垂直投影轮廓。

根据本发明的一个方面，提供了一种在数字图像中定位文本的装置，该装置包括：用于对像素进行分类以确定所述像素是否是文本区域的一部分的模块；用于通过将方框从包括至少一个像素的像素矩形扩展成包括由至少一个与所述矩形邻接的像素构成的组来生成初始文本界定方框的模块，其中基于所述组中任何给定组的平均亮度与第一阈值之间的比较结果来确定是否将初始文本界定方框扩展成包括该给定组在内；和用于合并初始文本界定方框的模块，其中该模块生成具有自适应阈值的水平投影轮廓和具有自适应阈值的垂直投影轮廓。

### 附图说明

从下面给出的对于本发明实施例的详细描述和附图可以更充分地理解本发明，但是，这些描述以及附图不应该被用来将本发明限制为所描述的特定实施例，而仅仅是为了说明和理解本发明。

图 1 中的流程图表示在本发明的一些实施例中完成的各种功能。

图 2 中的流程图表示在本发明的一些实施例中的定位的各个阶段的图像。

图 3 示出了在具有文本和背景的一个帧中的一个图像的初始界定方框

的例子。

图 4 示出了垂直和水平投影轮廓的例子。

图 5 示出了应用于图 3 的文本的一部分的垂直分割。

图 6 示出了应用于图 3 的文本的一部分的水平分割。

图 7 示出了网站上的一个图像，该图像具有文本和背景。

图 8 以部分方框图、部分流程图的方式表示出根据本发明一些实施例的通过量化来完成的颜色估计。

图 9 中的流程图表示根据本发明的一些实施例的视频监视和文本跟踪之间的关系。

图 10 中的方框图表示能实现根据本发明的一些实施例的功能的计算机。

## 具体实施方式

### 1. 简介

本发明的各个实施例包括图像中文本的定位和/或分割，其中所述图像可以是静止的或运动的图像，例如视频或网页中的图像。网页可以包括视频或非视频图像。不要求文本处于图像中的特定位置或具有特定的颜色。另外，背景（也称为非文本）可以是简单的（例如单色的）或复杂的背景。

数字视频的高效索引和检索是多媒体数据库的一个重要的方面。视频中的文本对于检索来说是一种强大的高级索引。检测、提取和识别文本可以建立这样一种索引。它使得用户可以提交很复杂的查询条件，例如由约翰·韦恩主演或斯蒂芬·斯皮尔伯格导演的所有电影的列表。或者，该索引可以用来跳转到关于一个特定主题的新闻报道，因为新闻广播的标题经常提供其下面的新闻报道的高度概括。例如，可以通过搜索“财经新闻”一词来得到当天的财经新闻。该索引也可用来记录广告的播出时间和日期，从而为那些替他们的客户检查其广告是否已经按预定时间在预定的电视频道播出的人们提供帮助。如果可以自动地、可靠地识别数字视频中的文本，很多其他有用的高级用途是可以想象的。分割和识别网页的非文本部

分中的文本也是一个重要的问题。越来越多的网页在图像中显示文本。现有的文本分割和文本识别算法不能提取文本。因而，所有现有的搜索引擎都不能正确地对具有丰富图像的网页的内容进行索引。文本分割和文本识别也有助于为大显示器设计的网页自动转换成适合在小 LCD 上显示，因为图像中的文本内容可以被检索出来。

## 2. 综述

图 1 是一个流程图，示出了关于本发明的一些实施例的概况。图 1 包括一个文本定位块 10 和一个文本分割块 14。说明书中提到“实施例”、“一个实施例”、“一些实施例”或“其他实施例”时，指的是所描述的与实施例有关的特定特征、结构或特性包括在本发明的至少一些实施例中，但不一定包括在所有实施例中。提到“实施例”、“一个实施例”或“一些实施例”时不一定总是指相同的实施例。

2.1 文本定位：数字输入信号（该信号通常包括一个图像）由文本定位块 10 的特征提取块 20 接收。在一些实施例中，在特征提取块 20 接收输入信号之前或同时，将任何 ASCII 或相关文本（例如 HTML 文本）移去。应注意，一个网页可能具有多个图像，这些图像被当作是独立的图像。文本定位块找到图像中文本的位置并用紧凑的文本界定方框来对其进行标记。在一些实施例中，这些界定方框应该仅包围一个文本列的一个文本行。但是，如下面所述，文本列不限于单个字符。在一些实施例中，定位包括以下步骤：

(1) 特征提取（块 20）：从输入图像中提取特征，获取那些文本特有的特征。

(2) 区域分类（块 22）：特征图像中的每个像素按照是否属于文本被分类。基于该信息生成初始文本界定方框。

(3) 区域合并（块 24）：细化（refine）文本界定方框，使得每个方框仅包含一行及一列文本。

(4) 文本跟踪（块 26）：如果输入为视频，则将本块加到处理过程中。这里我们利用视频的时间冗余性来提高文本界定方框的精度并消除很

多错误警报。

(5) 估计文本和最主要的背景颜色 (块 28)。

2.2 文本分割: 文本分割阶段 (块 14) 将背景 (非文本像素) 移去并生成一个输出信号。输出端 44 上的输出信号是一个图像文本表示。图像文本表示的一个例子是文本位图。所述文本位图例如可以包括白色背景上的黑色文本，而不管原来的文本和背景的颜色是什么。该位图可以被文本识别软件用来识别已经被块 10 和 14 定位和分割的特定文本。作为一个例子，文本识别软件可以是标准 OCR 软件，该软件预期白色背景上的黑色文本，不过本发明不限于产生这样一种输出信号。

为了改进分割，每个文本方框被按比例变换为高度为例如 100 像素 (块 30)。接着，移去背景 (块 32 和 36)。对于背景像素的搜索在文本界定方框的边界上开始。对于视频，可以在该步骤之前对同一文本的位图进行子像素精确对准 (块 34)。其余的像素可以被二值化 (块 38)。如上面提到的，所得到的二元位图可以被送入标准 OCR 软件以将其内容转换为例如 ASCII。

本发明不限于图 1 的特定块 (10 和 14)。在不同的实施例中，这些块 (20—38) 的细节可以是不同的，另外，一些块可以被省去、被合并或具有不同的顺序。

### 3. 其他的综述信息和概要

文本定位和分割系统的一些实施例属于一种自顶向下的方法。在视频的情况下，通过利用其时间冗余性来细化可能的文本行 (小节 5)。象在文本定位中一样，文本分割也可以利用视频的时间冗余性来改进分割结果。几个基本的决定包含在一些实施例中。它们包括：

(1) 仅考虑水平文本，因为 99% 以上的人工文本的出现属于这种情况。将任何书写方向都考虑在内的一些较早系统的经验表明，剩下的 1% 的文本出现率会带来高得多的错误警报率。只要是实现视频和图像中正确分割的文本大于 90% 这样一个性能仍然比较困难，则可以忽略非水平文本。

(2) 非文本区域比文本区域的可能性大得多。因此，我们决定将粗略文本检测器训练成尽可能紧密（对于特定位置的特定大小的文本进行训练）。通过将我们的文本检测器应用在所有比例下的所有位置可以实现与比例和位置无关。

另一个决定是，只有当文本由至少两个字母或数字组成时才考虑该文本的出现。

但是，本发明不限于上面提到的特定细节。可以理解，对于某些特定的应用，将会使用垂直文本，在这种情况下，可以对本发明做出适应性修改。另外，如果关于该图像的其他信息是已知的，可以修改本发明的特定实施例以利用所述已知信息。

#### 4. 文本定位

参见图 2，图像 50 被按比例变换为不同大小的多个图像 52、54、56、58 和 60。这些图像可以是静止图像或视频中的图像帧。虽然示出了五个图像，图像的数目可以多于或少于五。确定图像中的像素的边缘方位以生成特征图像 62、64、66、68 和 70（见小节 4.1）。使用一个固定比例文本适配器来对边缘方位图像中的像素进行分类，以生成图像 72、74、76、78 和 80（见小节 4.2）。图像 72—80 被整合为与一个图像相关的一个突出图形 84（见小节 4.3）。从该突出图形 84 生成初始文本界定方框（见小节 4.4.1）。文本界定方框和一个与图像 50 相同或相似的相关图像由块 86 表示。修改块 86 的文本界定方框（例如合并）（见小节 4.4.2）以生成修改后的文本界定方框，由块 88 表示，该块 88 也表示与块 86 相关的图像。应注意，文本界定方框不是所述图像的一部分，而是与所述图像相关。

##### 4.1 图像特征

研究人员普遍地将人工文本的出现表征为具有高对比度和高频率的区域。有很多不同的方法来放大这些特征。一种方法是使用 RGB（红、绿，蓝）输入图像  $I(x, y) = (I_r(x, y), I_g(x, y), I_b(x, y))$  的梯度图像来计算复值边缘方位图像  $E$ 。 $E$  按如下方式定义：令  $A_c(r, \varphi)$  为颜色平面  $c$  的直角坐

标系的微分图像  $\nabla I_c(x, y)$  的角坐标表达式。则  $E$  被定义为  $A(r, \varphi \bmod 180^\circ) = \sum_{c \in (r, g, b)} A_c(r, \varphi \bmod 180)$  的直角坐标系的表达式。模 180 度用来将方向转换为方位。 $E$  是我们用于文本定位的特征。

另一种方法是使用图像带  $b$  的方向微分  $D_x^c$  和  $D_y^c$  来计算方向边缘强度

$$E_x = \sum_{c \in (r, g, b)} |D_x^c| \text{ 和 } E_y = \sum_{c \in (r, g, b)} |D_y^c|.$$

及其总边缘强度

$$E = 1/3 \sum_{c \in (r, g, b)} ((D_x^c)^2 + (D_y^c)^2)^{1/2}.$$

#### 4.2 固定比例文本检测器

在一些实施例中，使用一个固定比例文本检测器来根据边缘方位图像  $E$  中每个像素附近的局部区域来对所述每个像素进行分类，以确定其是否是一特定大小的文本区域的一部分。例如，给定边缘方位图像  $E$  中的一个  $20 \times 10$  像素的区域，固定比例的文本检测器对于该区域是否包含特定大小的文本进行分类。有许多不同的技术来建立一个分类器。这些例子包括贝叶斯分类器（Bayes classifier），混合高斯分类器（Mixed-gaussian classifier）和前馈型神经网络（其具有好的归纳能力）。对于我们的工作，我们将采用内曼-皮尔逊（Neyman-Pearson）标准的贝叶斯分类器的性能和实值与复值前馈型神经网络的性能进行了对比。具有双曲正切激活函数的复值神经网络可以提供更出色的性能。在一些实验中，在可比的命中率（90%）的情况下，其在验证集合上的错误命中（0.07%）比可比的实值网络低两倍多。

网络结构： 可以采用各种网络结构。在一些实施例中，由  $E$  中的  $20 \times 10$  边缘方位区域馈送的 200 个复值神经元作为网络输入。该感受范围的大小在性能和计算的复杂性之间达到很好的折衷。 $30 \times 15$  神经元的输入层不能获得更好的分类结果，但是计算上代价更高。另一方面，使用少于 10 行的输入层导致结果差得多。应注意，感受范围的行数决定了被检测的字体的大小，因为所有训练文本模型被按比例变换，使得字体的大小等于行数。输入层再与一个 2 复值神经元的隐藏层连接。同样，使用更多的隐藏

---

神经元并不能导致任何性能的改进，而仅使用一个隐藏神经元将错误警报率增加到三倍。隐藏层被整合为一个实值输出神经元。

网络训练：有多种方法完成网络训练。下面描述了一些方法，但是本发明并不限于此。训练和验证测试集合应该尽可能小，但仍然具有代表性。它应该包含所有典型的文本模型和非文本模型。理论研究表明，如果训练集合中的文本和非文本样本数目之间的关系对应于应用中二者之间的关系，神经网络将是最有效的。满足该条件的大量训练样本被获得。虽然怎样获得不同类型的文本的例子是易懂的，但是得到有代表性的非文本集合要更困难一些。

这个问题的一个解决方案是所谓的“自引导（bootstrap）”方法。训练集合的组成可能严重影响网络性能。在一些实施例中，收集到具有 30180 个文本模型和 140436 个非文本模型的有代表性的集合。最初的 6000 个文本模型和 5000 个非文本模型是为训练而随机选择的。仅允许非文本模型集合增加另外 3000 个通过“自引导”方法收集的模型。该方法由一个初始的非文本模型集合开始，以训练神经网络。然后，使用一个与训练集合不同的验证集合（这里：所有模型减去训练集合）来估计经训练的网络。验证集合的一些错误分类的模型被随机地加到训练集合中，并用这种扩展的而且是改进的训练集合训练出一个新的、有望增强的神经网络。再次用验证集合对所得到的网络进行估计，并将仍错误分类的非文本模型加到训练集合中。重复该训练和定向加入新模型的操作，直到验证集合中的错误分类的模型的数目不再减少，或者，象我们的例子中那样，直到已经加入了 3000 个非文本模型（并且仅仅是非文本模型）。该迭代的训练过程保证了一个多样化的训练模型集合。

给出一个经正确训练的神经网络，一个  $20 \times 10$  像素窗口滑过边缘方位图像 E，并在每个位置被估计。当且仅当网络输出值超过  $th_{network} = 0$  时（在 -1 与 1 之间），用该网络输出值来填充一个所谓的响应图像中的一个相关的  $20 \times 10$  区域，由此将网络响应存储在所述响应图像中。由于步长为 1 可能使计算量大到不适合于大图像或高清晰度电视（HDTV）视频序列的程度，我们在 x 和 y 方向分别采用步长因子 3 和 2。该子采样可以不

降低精确度，但将速度提高至 6 倍。

在其他实施例中，使用一个实值网络，逻辑激活函数(logistic activation function)，在每个窗口位置，测试神经网络的输出是否超过了  $th_{network}=0.85$ (在 0 和 1.0 之间)。如果是，则可以将一个由神经网络输出值填充的  $20\times10$  的方框加到响应图像中的相关位置。

#### 4.3 比例整合

在一些实施例中，所有比例下的粗略固定比例文本检测结果(图像 72—80)被整合为一个文本突出图形(saliency map)，以便恢复初始文本界定方框。(见图 2，方框 82)。在很多情况下，文本位置的特点在于在多个比例下的正确命中，而错误警报在多个比例下的一致性较低。可以通过将文本置信度(confidence of being text)投影为该图像的原始比例来生成突出图形。(文本置信度的一个例子是神经网络输出的激活程度)。突出图形可以初始化为零。然后，对于在每个比例下检测到的界定方框，将其文本置信度值按照在原始图像比例下该界定方框的大小加入到突出图形中。在一个特定区内，某一给定比例的界定方框可能多于一个。在一些实施例中，突出图形可以反映一个特定区内的所有图像比例的界定方框的总数。

#### 4.4 文本界定方框的提取

##### 4.4.1 初始文本界定方框

有各种方法来生成文本界定方框。下面描述了一些实施例中采用的技术，但本发明不限于这些细节。为了生成一个初始的文本界定方框的集合，其中所述方框包围明显突出的区域，该算法开始在突出图形中搜索下一个尚未处理的、其值大于预定阈值  $th_{core}$  的像素。阈值的选择是由避免为非文本区域生成文本方框的目标而确定的。非文本区域应该不那么突出。对于我们的分类器， $th_{core}=5.0$  工作得很好，但是，有时可能必须调整该阈值(例如，如果训练一个新的神经网络)。该阈值可以不是 5.0。一旦在突出图形中找到一个其值  $P(x, y) > th_{core}$  的像素(称为核心像素)，则将其作为一个高度和宽度分别为 1 的新文本方框的种子。然后以迭代的方式扩展该新文本方框。下面的伪码(称为伪码例 1)给出初始文本方框生成算

法的一个例子。

初始文本方框生成算法（伪码例 1）：

- (1) search for next core pixel and create a new text box of width and height 1
- (2) do
- (3)       extendNorth(box)
- (4)       extendEast(box)
- (5)       extendSouth(box)
- (6)       extendWest(box)
- (7) while(box changed)

在整个边缘强度图像中该方框的总宽度上方相邻行的像素的平均亮度被当作在该方向增长的判断准则。如果平均亮度大于  $th_{region}=4.5$ ，则将该行加到方框中。这个值被选择成比  $th_{core}$  略小一些，以便不是仅仅得到一个包括文本区域的核心的文本方框，而是要得到一个包括该文本的所有部分的文本方框。接着，使用同样的判断准则将方框向左方、下方和右方扩展。只要界定方框持续增长，则重复该迭代的方框扩展（见伪码例 1）。

图 3 示出了视频帧 110 中的一个图像中的日期和时间和初始界定方框的例子，但本发明不限于这些特定的例子。帧 110 的背景可以是一种单一颜色（例如图中示出的白色）的背景，或是具有各种形状的不同颜色的更复杂的背景。文本界定方框用虚线表示。图像 110 中可以有其他的文本。

#### 4.4.2 修改的文本界定方框

初始界定方框经常不能最恰当地给图像中的文本加上边框：在实践中，一些方框不包含任何文本（错误警报）；而另外一些方框则跨越多于一行和/或列文本，并且在很多情况下，背景占去很大一部分像素。好在通过一种利用包含在所谓的投影轮廓中的信息的迭代的后处理过程（iterative post-processing procedure）可以克服这些缺点。

一个图像区域的投影轮廓是空间像素内容分布的简洁表示，并且已经成功地应用在文件文本分割中。直方图仅获取例如像素亮度的某种图像特征的频率分布（丢失所有空间信息），而亮度投影轮廓能保留大致的空间

分布，其代价是像素内容更加密集。水平/垂直投影轮廓可以定义为每一列/行上像素亮度和的矢量。

图 4 示出了一个例子，其中垂直和水平投影轮廓被绘制为沿特征图像的 x 和 y 轴的条线图。文本行上边界的标志是垂直投影轮廓中的陡然上升，而下边界的标志是陡然下降。类似地，文本对象的右和左边界上的标志是水平投影轮廓中的陡然上升和下降。这些陡然的上升和下降可以被识别为轮廓图穿过一条自适应地设置的阈值线的位置。从下到上的转换由一条长线表示，从上到下的转换由一条短线表示（如图 4 中标出的那样）。

术语“文本对象”按下面的方式使用。在单个图像的情况下，一个文本对象是一个文本边界方框（包括已经经过修改处理的情形）。在视频的情况下，一个文本对象包括来自不同时间的帧的多个文本界定方框（包括已经经过修改处理的那些情形）。换言之，在视频的情况下，文本对象包括同一文本的不同例子，它们来自不同的帧（图像）。

垂直分割算法的一个例子在伪码例 2 中以伪码形式给出。水平分割算法的一个例子在伪码例 3 中以伪码形式给出。但是，本发明不限于伪码例 2 和 3 中所示的特定细节。其他方法也可实现本发明的实施例。应注意，在本小节中使用“分割”一词时，是与修改初始界定方框有关，而在小节 6 中，通常指的是从背景移去文本。

#### 垂直分割算法（伪码例 2）：

- (1) expand box at the top and bottom by the minimum of half the height of the original text box and half the possible maximal text height
- (2) calculate vertical projection profile of the | E |
- (3) get minimum and maximum profile values
- (4) calculate the segmentation threshold
- (5) set change=false
- (6) for all rows of the profile
- (7) if(profile [current row]>threshold)
- (8) if(no upper boundary yet)

- 
- (9) set upper boundary=current row
  - (10) else
  - (11) if(no lower boundary yet)
  - (12) set lower boundary=current row
  - (13) if(upper boundary)
  - (14) create new box using the values of upper and lower boundaries
  - (15) unset current upper and lower boundaries
  - (16) set change=true
  - (17) delete processed box

水平分割算法（伪码例3）：

- (1) expand box at the left and right by the minimum of half the height of the original text box and half the possible maximal text height
- (2) calculate horizontal projection profile of the | E |
- (3) get minimum and maximum profile values
- (4) calculate the segmentation threshold
- (5) for all columns of the profile
- (6) if(profile [current column]>threshold)
- (7) if(no left boundary yet)
- (8) set left boundary=current column
- (9) else if(right boundary)
- (10) if(gap between current column and right boundary is large enough)
- (11) create new box from left and right boundaries
- (12) unset left and right boundaries
- (13) else
- (14) unset right boundary
- (15) else if(no right boundary)
- (16) set right boundary=current column
- (17) if(left && no right boundary)

- 
- (18) right boundary =last column
  - (19) if(left and right boundaries)
  - (20) update processed box to current right/left boundaries
  - (21) else
  - (22) delete processed box

参见伪码例 2，在一些实施例中，应用到每个文本方框的垂直分割算法按下面的方式工作，但本发明不限于这些细节。在顶部和底部扩大该方框（伪码例 2 中的第（1）和（2）句）。该扩大是需要的，因为正确的边界可能位于当前方框的外部，因而初始边界偶尔可能切掉文本的一部分。为了正确地恢复这些边界，应该考虑原始方框外部的一些行。我们将顶部和底部的扩大量设定为原始文本方框高度的一半和最大可能的文本高度的一半中的较小者。原始文本方框高度的一半看起来是一个比较好的对于初始垂直边界中的缺陷的最差情况估计，而采用最大可能的文本高度的一半的限制条件是因为原始文本方框可能包含多于一行文本，因而造成文本方框高度的一半可能大于最大可能的文本高度的一半。

接着，计算特征图像 | E | 的扩大后的方框上的垂直投影轮廓和该轮廓中的最大和最小值  $\max_{\text{profile}}$  和  $\min_{\text{profile}}$ 。为了确定投影轮廓中的单个值是否属于一个文本行，可以将阈值  $\text{thresh}_{\text{text}}$  计算为  $\text{thresh}_{\text{text}} = \min_{\text{profile}} + (\max_{\text{profile}} - \min_{\text{profile}}) \times 0.175$ 。（注意伪码例 2 中的第（4）句）。因子 0.175 是用实验方法选择的，在其他实施例中可以不同。垂直轮廓值超过  $\text{thresh}_{\text{text}}$  的每一行被分类为包含文本。

在伪码例 2 的第（6）—（8）句，算法开始从顶部搜索第一个从下到上的转换。这一行被标记为文本方框的可能的上边界（第 9 句）。然后，在投影轮廓中搜索下一个从上到下的转换（第 13 句）。如果找到的话，则生成一个具有当前的上及下边界的新文本方框。继续搜索新的一对从下到上和从上到下的转换，直到投影轮廓中的所有元素都被处理了。最后，可以删除原始文本方框。文本方框现在被分解为其文本行。见图 5，示出了应用到图 3 的帧的一部分的垂直分割。应注意，可以对图 5 所示的界定

方框进行额外的修改。

类似地，使用水平分割算法（伪码例 3）来保证处于一行中但不属于一个整体的文本被分开。但是，在一些实施例中，伪码例 2 与例 3 相比可能有两个不同之处：

(1) 在计算  $\text{thresh}_{\text{text}}$  时使用因子 0.25 而不是 0.175。实验表明这个值对于水平分割来说是非常好的。

(2) 加入了一个间隔参数。与垂直分割不同，“相同”列中的单词不应该因为各个单词之间有小的间隔而被分开。因此，需要间隔参数来弥补这些较低的水平轮廓值（如果必要的话）。如果该算法已经找到了一对从下到上和从上到下的转换，也就是找到了一对可能的左及右边界，而且如果该找到的从上到下的转换与当前列之间的间隔足够大，则在当前列中找到的从下到上的转换被判断为一个新的文本对象的左边界，并且从先前找到的那对转换生成一个新的方框。当前列被标记为新的可能的左边界。如果间隔不够大，则该算法将轮廓中的凹部判断为太小，并因此将其忽略（删除到目前为止找到的可能的左边界）。该算法继续处理轮廓中的下一个值。本发明不限于这些细节。

图 6 给出了水平分割算法的结果的一个例子。应注意，对于更复杂的文本布局，可以对界定方框进行额外的修改。

图 7 示出了包括背景 124 的图像 120 中的文本“DOW JONES Commodities trading is risking and is not for everyone”。图像 120 在一个网页 126 中。背景 124 可以是单色的背景，或复杂的背景（例如，具有不同形状的很多颜色）。垂直分割算法可能不会一开始就将“Commodities trading is risking and is not for everyone.”的不同文本行分开。只要设想一下各文本方框的垂直投影轮廓是什么样子就可以理解这是为什么。左列中的文本方框可能会挡住右侧的较小文本的垂直轮廓，因而后者不能被分为两个文本行。另一方面，两个文本列之间的间隔足够大，从而在采用水平分割算法后能够被分开。实验中的结果是，对文本方框进行了几个周期的（或几次）垂直和水平分割后，几乎每种布局都可以被分为其文本行和列。

由于图像及视频帧中的文本高度是有限的，在一些实施例中，高度为  
 $height < min_{textheight} = 8pt$

或

$height > max_{textheight} = image_{height}/2$

的方框被分类为非文本区域，并因此被丢弃。另外，由于水平分割确保文本方框包含诸如单词或文本行的文本对象，正确分割的文本方框的高度应该比其宽度小。结果，其高度大于宽度的方框也可以被丢弃。最后，那些具有相同上及下边界并且接近到互相接触或重叠程度的文本方框可以被加入到一个文本方框中。这样减小了复杂度，而且以后可以使得在整个处理过程中文本跟踪更稳定。

#### 4.4.3 估计文本颜色和背景颜色

在一些实施例中，对于每个文本界定方框进行文本颜色和背景颜色估计。该估计可以用来确定一文本界定方框是包含常规文本（明亮背景上的深色文本）还是反向文本（深色背景上的明亮文本）。图像通常是多色的。即便是一个肉眼看上去是单色的区域，例如视频帧中的一个字符，也是由具有很多不同但相近颜色的多个像素构成的。因此，可以通过将颜色量化为例如四个最主要的颜色来降低每个文本界定方框中颜色分布的复杂度。可以使用多种矢量量化器。在我们的工作中，使用了快速矢量量化器，市场上很容易买到。

文本颜色直方图提供了一种量度，该量度表示出界定方框中文本所包括的量化颜色的量。该量度可以是文本的样本，例如，文本界定方框的四个中心行。由文本颜色直方图计量的颜色通常也可以包括混合在字母之间或某些字母（例如“o”）内部的某些背景。当然，除了所述四个中心行以外，文本的其他部分也可用于文本颜色直方图。

背景颜色直方图可以提供一种量度，该量度表示出背景的某些部分中包括的量化颜色的量。例如，这些部分可以是文本方框上面紧挨着的两行和下面紧挨着的两行（总共四行）。应注意，该背景颜色直方图可以包括来自两个背景颜色直方图（例如一个来自文本上方，另一个来自文本下方）的分量。或者，也可以只有一个来自文本上方的背景颜色直方图，或

者一个来自文本下方的颜色直方图。

在一些实施例中，我们计算文本与背景直方图之间的差异直方图。差异直方图的最大颜色很可能对应于文本颜色，而差异直方图的最小颜色很可能对应于最主要的背景颜色。实验表明，该方法对于单色文本是很可靠的。当然，对于多色的文本，该方法可能失灵，但多色文本是很少见的。

根据估计的文本颜色和最主要的背景颜色，我们估计一个文本界定方框是包含常规文本还是反向文本，如上面所述。如果文本颜色的灰度值比最主要的背景低，我们则假定该文本为常规文本，否则为反向文本。

图 8 的方框图表示根据本发明的一些实施例，使用矢量量化器和使用颜色直方图来估计颜色。其他的实施例具有不同的细节。参见图 8，块 130 表示一个矢量量化（VQ）前的被界定的文本方框及周围的背景。块 134 表示被界定的经矢量量化的文本信号和背景。在 VQ 后，包括背景的文本信号只有四种颜色。从例如通过文本中心的一个带状区（例如四个中心行）生成颜色文本直方图  $CH_T$ 。分别从文本上方的一个带状区（例如两行）和文本下方的一个带状区（例如两行）生成上部和下部颜色直方图  $CH_U$  和  $CH_L$ 。在该例子中，允许有 4 种颜色。因此，颜色直方图表示出，在 VQ 之后包括在这些带状区中的每种颜色 C1、C2、C3 和 C4 的量。生成一个差异颜色直方图  $CH_D$ ，其中  $CH_D=CH_T-CH_U-CH_L$ 。如上所述，在从  $CH_T$  减去颜色直方图  $CH_U$  和  $CH_L$  之前可以将这两者相加。

应注意，可以按照下面小节 6.2.2 和 6.3 中所述来使用估计的颜色。但是，在小节 5 至小节 6.2.1 中和小节 6.2.2 的第一部分中，可以使用具有灰度颜色的图像（例如图 2 中的图像 88）。

## 5.利用视频中的信息冗余性

视频与静止图像和非视频网页的区别在于时间冗余性。通常，每个文本行在几个连续的帧中出现。该时间冗余性可以用来：

- (1) 提高定位文本的几率，因为相同的文本可能在不同的情况下逐帧出现，
- (2) 消除单独的帧中的错误文本警报，因为在整个处理过程中它们

通常是不稳定的，

- (3) 将“偶然”丢失的文本行的位置插入单独的帧中，
- (4) 通过一段时间内的位图整合来改进文本分割的效果。

但是，利用该冗余性在计算上可能是代价很大的，并且采用小节 4 中描述的我们的文本定位方案可能代价过高。为明白这一点，假设基于图像的文本定位器对于每个 MPEG-I 视频帧需要大约 5 秒钟。处理一分钟的视频一共需要 2.5 小时！MPEG 指运动图像专家组。当前的和建议中的 MPEG 格式包括 MPEG-1（“高达 1.5Mbps 的用于数字存储介质的运动图像和相关音频的编码”，ISO/IEC JTC 1 CD IS-11172 (1992)），MPEG-2（“运动图像和相关音频的通用编码”，ISO/IEC JTC 1 CD 13818 (1994)），和 MPEG-4（“甚低比特率音频-视频编码”状态：94 年 11 月征求意见，96 年 11 月作出草案）。有不同版本的 MPEG-1 和 MPEG-2。也可以使用除 MPEG 以外的各种格式。

### 5.1 文本对象

在静止图像的情况下，所有已定位的文本界定方框通常是独立的，彼此无关。为了利用视频中固有的冗余性，连续帧中的相同内容的文本界定方框可以根据这些文本界定方框的视觉内容概括成一个文本对象。在视频的情况下，一个文本对象描述一段时间内的一个文本行，这是用该文本行在各帧中的图像文本表示（例如位图）、大小和位置以及它出现的时间范围来描述的。在两阶段处理过程中提取视频中的整个文本对象，以便降低计算的复杂度。下面描述了一些实施例中的操作，但本发明不限于此。在第一阶段，以一较粗的时间分辨率监视视频信号（见图 9）。例如，仅对每个第 20 帧（例如，图 9 中的帧 F80, F100, F120 等）应用小节 4 中描述的基于图像的文本定位器。如果检测到文本（例如在帧 120 中），则将进入第二阶段即文本跟踪。在这个阶段，在监视阶段找到的文本行被（按时间）向后跟踪（例如帧 F119）和向前跟踪（例如帧 F121），直到其出现的第一帧（例如帧 F115）和最后一帧（例如帧 F134）。这个阶段结合使用基于特征谱(signature)的文本行搜索和基于图像的文本定位。基于特征谱的搜索的计算强度没有基于图像的文本定位的计算强度大（小节 4）。

基于特征谱的搜索可能包括将文本的边缘和某些区域与其他帧中的相应内容相比较。它可能包括边缘图（edge map）比较。也可以比较水平轮廓。

### 5.1.1 对于文本出现的视频监视

在一些实施例中，以一个较粗的时间分辨率来监视视频中出现的文本。为此，可以仅将基于图像的文本定位器应用到视频中的一个帧子集，其中这些帧是均匀间隔的。步长是根据不忽略任何文本行这一目标而确定的。但是，文本行是在其最初出现时、最后出现时、还是在其出现的中期被定位可能并不重要。在任何情况下，文本跟踪阶段将恢复每个文本行的实际时间范围。

最大可能的步长可以由文本行出现的最小假定持续时间来给出，我们假定其为 1 秒。视觉研究表明，人们需要 2 到 3 秒钟来处理整幅画面。因而，假定文本至少应该清楚地出现  $2/3$  秒以便能够容易地阅读，这看起来是合理的。对于 30fps 的视频，这换算成步长为 20 帧。

在一些实施例中，如果基于图像的文本定位器没有在帧  $t$  中找到任何文本行，则继续对帧  $t+20$  进行监视处理。但是，如果找到了至少一个文本行，则可以将基于图像的文本定位器应用到帧  $t-1$  和帧  $t+1$ 。接着，对于帧  $t$  中的每个文本行，该算法在帧  $t-1$  和帧  $t+1$  中搜索一个对应的文本行。两个文本行之间的对应可以定义为：在其各自的帧位置，其各自的界定方框的至少 80% 的区域重叠，不过也可以采用其他的值。如果 A 和 B 分别表示描述基准和第二界定方框的点的集合，则重叠的百分比可以被定义为  $\text{overlap} (\text{重叠}) = |A \cap B| / |A|$ 。结果，在这种情况下，如果两个对应的方框出现在连续帧中的相同位置，则它们的大小不能相差 20% 以上，并且/或者如果它们具有相同的大小，则只允许它们之间有微小的移位。对于非静止文本来说，微小的移位是常见的。如果在帧  $t-1$  和帧  $t+1$  中找到了帧  $t$  中的一个文本方框的对应方框，则生成一个新的文本对象（包括这些文本方框），并对其进行标记以便于按时间跟踪。伪码例 4 给出了视频监视处理的概要。

对于文本出现的视频监视算法（伪码例 4）：

- (1)  $\text{video} = \{\text{frame } 0, \dots, \text{frame } T\}$
- (2) for  $t=0$  to  $T$  step  $2/3$  seconds
- (3) localize text in frame  $t$
- (4) if no text line found
- (5) continue with next  $t$
- (6) localize text in frame  $t-1$  and  $t+1$
- (7) for all text lines in frame  $t$  which do not belong to any text object yet
- (8) search for corresponding text line in  $t-1, t+1$
- (9) if search successful
- (10) create new text object
- (11) track textobject backward
- (12) track textobject forward

### 5.1.2 文本跟踪

在一些实施例中，根据在视频监视阶段生成的文本对象中包含的信息，将每个文本对象扩展到包含相应文本行的所有帧。（这减少了将在图 1 中的导线 44 上提供的位图的数目）。可以按时间向后和向前进行文本跟踪。但是，我们仅描述了向前跟踪，因为除了通过视频的方向不同之外，向后跟踪与向前跟踪是相同的。我们的快速文本跟踪器背后的基本思想是，取得当前视频帧中的文本行，计算一个表征性的特征谱，该特征谱使得该文本行区别于具有其他内容的文本行，并且在下一视频帧中搜索与该基准特征谱能最好地匹配的相同尺寸的图像区域。

小节 4.4.2 中定义的垂直和水平投影轮廓用作简洁的并且是表征性的基准特征谱，不过也可以使用其他的特征谱。特征谱的中心可以定义为相关文本行的界定文本方框的中心。两个特征谱之间的相似度可以用特征谱逻辑乘（signature intersection）（例如这两个特征谱中相应元素中的最小值的和）来计量。在特征谱获取一个有关目标并改变背景的情况下，特征谱或直方图逻辑乘要优于 L 范数。为了找到一个文本行在下一帧中的精确位置，可以计算其中心落入围绕基准特征谱中心的一个搜索窗口的所有特征

谱，并将其与基准特征谱进行比较。如果最佳匹配超过所要求的最小相似度，则可以确定找到了文本行并将其加入文本对象。如果最佳匹配没有超过所要求的最小相似度，则决定放弃基于特征谱的搜索(signature-based drop-out)。搜索半径的大小取决于最大的假定文本速度。在我们的实验中，我们假定文本在视频中从左侧移动到右侧至少需要 2 秒钟。在给定视频的帧大小和重放速率的情况下，这可以直接换算为以像素为单位的搜索半径。原则上，我们可以借助到目前为止包含在文本对象中的信息来预测位置，以缩小搜索空间，但是，这可能没有任何计算上的需要。

应注意，基于特征谱的穷尽搜索算法可以类似于用于运动估计的块匹配算法，不同之处在于，相似度的量度是基于一个从实际图像的特征图像导出的特征谱。

有可能的是，基于特征谱的文本行搜索不能检测到一个慢慢变弱的文本行，因为该搜索是基于先前帧中的文本行的特征谱，而不是基于一个固定的和导出的主/原型特征谱。帧间的变化可能小得不能被检测到。另外，基于特征谱的文本行搜索可能不能跟踪一些放大(zooming in)和缩小(zooming out)文本。为克服这些缺点，可以每隔 x 帧用基于图像的文本定位器来取代基于特征谱的搜索，以便重新校准文本行的位置和大小。但是，可以在那里丢弃新检测到的文本方框。

试验中，5 帧的间隔被证明能在速度和可靠性之间取得很好的折衷，但也可以采用更大的间隔。同样，在一些实施例中，对应文本行的界定方框可以至少重叠 80%。

由于视频中的缺陷，例如高噪声、有限的带宽（例如串色）、文本阻塞、压缩伪影（artifact）等，在严格的意义上（例如每帧）进行文本对象的连续识别经常是不可能或不实用的。因此，如果不能在下一帧中找到任何对应的文本行就终止跟踪可能不是一个好主意。代之以，只有当不能在一定数目的连续帧中找到任何对应的文本行时才终止跟踪。为此，可以采用两个阈值  $\max_{DropOut}^{signature-based}$  和  $\max_{DropOut}^{image-based}$ 。每当一个文本对象不能被扩展到下一帧时，则将相应的计数器加 1。每当相关的搜索方法成功时，则将相应的计数器复位为 0。当这两个计数器中的一个超过其阈值  $\max_{DropOut}^{signature-based}$  或

$\max_{DropOut}^{image-based}$  时，立即终止该跟踪处理。在我们的实验中，基于图像的文本定位器的阈值被设置为  $\max_{DropOut}^{image-based} = 3$ ，但也可以采用其他值。这种放弃可能是由噪声很大的视频帧或暂时阻塞的文本造成的。基于特征谱的搜索的阈值被设置为  $\max_{DropOut}^{signature-based} = 4$ ，例如，两个完整的被定位的帧之间的距离，但也可以采用其他值。采用阈值 4 使得可以在基于特征谱的搜索非常困难，例如搜索放大或缩小文本的情况下跟踪文本行。下面的伪码例 5 给出了根据本发明一些实施例的视频监视过程的概要。但是，可以采用具有其他细节的本发明的其他实施例。

给定文本对象的向前文本跟踪算法（伪码例 5）：

- (1) `sigBased_DropOuts=0`
- (2) `imageBased_DropOuts=0`
- (3) `while not (beginning or end of video||`
  - `sigBased_DropOuts>maxSigBased_DropOuts||`
  - `imageBased_DropOuts>maximageBased_DropOuts)`
- (4) `get next frame t`
- (5) `if(frame has to be localized)`
- (6) `localize text in frame t`
- (7) `search localized text box that matches to the box in the last frame of the`  
`text object`
- (8) `if(search successful)`
- (9) `add text box to the text object`
- (10) `reset sigBased_DropOuts and reset imageBased_DropOuts`
- (11) `else`
- (12) `increment imageBased_DropOuts`
- (13) `else`
- (14) `calculate feature image for frame t`
- (15) `estimate search area a for the text line`
- (16) `create a window w with the dimension of the text box in frame t-1`
- (17) `get signature s1 of the text box in t-1`

- 
- (18) for (each possible position of w in a)
  - (19) calculate signature s2 for w
  - (20) calculate error between s2 and s1
  - (21) memorize minimal error
  - (22) if(minimal error<threshold)
  - (23) add text box to the text object
  - (24) reset sigBased\_DropOuts
  - (25) else
  - (26) increment sigBased\_DropOuts

### 5.1.3 后处理

为了准备一个用于文本分割的文本对象，可以将其削减到已经以较高置信度被检测出来的部分。因此，在一些实施例中，每个文本对象在时间上被削减为基于图像的文本定位器检测到文本行的第一帧和最后一帧。接着，如果发生以下情况则丢弃该文本对象，例如，

- (1) 它出现的时间少于 1 秒钟，或者
- (2) 它的放弃率大于 25%。

也可以采用其他值。第一种情况来自于我们的观察：文本行通常需要至少 1 秒钟才能被看见，短于此时间的文本行通常是错误警报。第二种情况移去那些后续处理过程不能处理的、来自于不稳定跟踪的文本对象。不稳定跟踪可能是由强压缩伪影或非文本造成的。

最后，在一些实施例中，可以对于每个文本对象确定下面的一个或多个全局特征。在不同的实施例中，具体细节可能不同。

(1) 文本对象的文本颜色：假定同一文本行的文本颜色不随着时间的推移而改变，则将文本对象的文本颜色确定为每一帧的所有确定的文本颜色（例如，通过小节 4.4.3 获得的颜色）的中值。文本颜色并非必须被选择为中值。可以采用另一种平均或非平均的量。

(2) 文本大小：文本界定方框的大小可以是固定的，或者是随时间改变的。如果是固定的，我们通过宽度和高度的集合的中值来确定其宽度

和高度。

(3) 文本位置: 文本行可以在一个坐标轴或两个坐标轴方向上是静止的。如果文本行在每帧中的平均移动小于 0.75 像素，则将文本行看成在 x 和/或 y 方向上是静止的。平均移动是基于该文本行的第一次和最后一次文本出现的位置之间的差别由帧数归一化而计算的。

如果文本行是静止的，我们用中值文本界定方框取代所有文本界定方框。中值文本界定方框的左/右/上/下边界是所有左/右/上/下边界的中值。如果该位置仅在一个方向上固定，例如仅在 x 或 y 轴方向上固定，分别用中值来取代左和右或者上和下边界。

## 6.文本分割

文本分割涉及从文本移去背景。这不应与小节 4.4.2 中的分割混淆。

### 6.1 分辨率调整（见图 1 中的块 30）

可以对于再次改变比例的 (rescaled) 图像（例如，通过三次内插）进行文本分割操作，使得所考虑的文本对象的文本高度为固定的高度，例如为 100 像素，并且保留高宽比。再次改变比例的原因有两个：

#### (1) 增强较小的字体尺寸（其能带来更好的分割结果）的分辨率

当前的视频中的文本提取和文本识别的一个主要问题是其分辨率很低。对于 MPEG-I 编码的视频，各字符的高度经常小于 12 像素。虽然对于人来说在该分辨率下仍能够识别文本，但对于当今的标准 OCR 系统来说则比较困难。这些 OCR 系统被设计用来识别文件中的文本，这些文件是以至少 200dpi 至 300dpi 的分辨率扫描的，造成最小文本高度为至少 40 像素。为了用标准 OCR 系统获得好的结果，人们希望增强文本行的分辨率。

增强文本位图的视觉质量是按比例放大较小的文本位图的另一个并且是更重要的原因。该更高的分辨率使得能够进行小节 6.2.2 中的子像素精确文本对准（相对于原始分辨率）。

#### (2) 对于较大的字体尺寸节省计算量

大于固定高度（例如 100 像素）的文本高度不能改进分割或 OCR 性

能。减小其尺寸可以显著降低计算复杂度。应注意，由于我们的方法事实上是多分辨率方法并且在网页和 HDTV 视频序列上以高达 1920 乘 1280 像素的分辨率工作，较大的字体尺寸是很可能的。100 像素只是帧高度的 1/12。

## 6.2 移去背景（包括复杂背景）

如上面所述，可以移去背景。（见图 1 中的块 32）。复杂背景比简单背景具有更大的变化。但是，本发明不限于特定类型的背景（它可以是复杂和简单的背景）。但是，如上所述，如果关于图像背景的特定信息是已知的，可以修改本发明的实施例以便利用该信息。

### 6.2.1 图像

文本的出现应该与其背景形成反差，以便能容易地阅读。此处利用该特征来移去复杂背景的较大的部分。在一些实施例中是按下面所述的方式工作的，但本发明不限于此。基本的思想是增大文本界定方框，使得没有文本像素落在边界上，然后将文本界定方框边界上的每个像素当作种子，以便用背景颜色填充差别不大于  $\text{threshold}_{\text{seedfill}}$  的所有像素。（应注意，在一些实施例中，在一开始时仅仅是记录所述被填充像素的颜色的改变，即改变为背景颜色，而并不在位图上实际执行。可以在对于方框边界上的所有像素采用了种子填充(seed fill)之后实际执行。）对于反向文本来说背景颜色是黑的，而对于常规文本来说背景颜色是白的。由于边界上的像素不属于文本，而且由于文本与其背景形成反差，种子填充算法不会移去任何字符像素。（种子填充算法在本领域是公知的。）我们将这个新构建的位图称为  $B^r(x, y)$ 。

在我们的实验中，RGB 颜色之间的欧几里德距离被用作距离函数，并且该种子填充算法利用 4 邻域。另外，为确保所有字母完全包含在文本界定方框中，我们将其在水平方向上扩展 20%，在垂直方向上扩展 40%。可以采用其他值。

不是所有背景像素都需要被删除，因为由种子算法填充的区域的大小可以受一个像素与其邻接像素之间的最大允许色差限制。可以利用其余颜色区域的大小，以便用背景颜色来填充背景的其余区域。在一些实施例

中，每个像素可以是一个用于种子填充算法的种子。然后可以假想地将 8 邻域种子填充算法应用到  $B^r(x, y)$ ，以便确定可以被填充的区域的尺寸。背景区域应该比文本字符区域小。因此，高度小于  $\min_{\text{height}}$  像素且宽度小于  $\min_{\text{width}}$  或大于  $\max_{\text{width}}$  的所有区域被删除，（设定为背景颜色）。

### 6.2.2 视频图像

视频文本对象与单个图像文本对象的区别在于，它包括同一文本行的多个而不仅是一个图像文本表示（例如位图）。在一些实施例中，使用下面的方法来利用该冗余性，以移去包围实际字符的复杂背景。但是，本发明不限于这些细节。该方法不但可以应用于静止文本，也可以应用于移动文本，因为我们已经解决了子像素精确文本行对准的问题。

可以以灰度格式重新装载原始图像。但是，可以如下所述用矢量量化的版本来确定哪个灰度颜色与估计文本颜色相同。

在一些实施例中，它如下工作。假设你将一个文本对象的各个位图堆叠起来，使得字符彼此精确地对准。属于文本的像素随着时间的推移仅仅有微小的改变，而属于非文本（背景）的像素经常随着时间的推移有很大的改变。由于文本位置因对准而成为静止的，其像素应该不会改变。（应注意，尽管文本应该是静止的，但各帧之间可能有微小的改变）。背景像素很可能因为背景中的运动或文本行的运动而改变。

我们对于每个文本对象导出一个代表性文本行位图。给定精确对准的位图堆，在一段时间内对于常规/反向文本的灰度图像进行最大化/最小化运算。应注意，不必使用文本对象的每个位图，因为在两个连续的帧中背景通常不会显著改变。结果是，选择大约 40 个在时间上均匀间隔的帧就足以获得很好的结果。例如，如果选择 40 帧并且共有 200 帧，则这 40 帧的间隔为 5。如果有 150 帧，则这 40 帧的间隔为 15/4，这表明该间隔可以四舍五入为一整数，或者该间隔可以不是恒定的，有时是 3，但更多时候是 4，以使平均值为 15/4。还应注意，在文本对象的开始和结束时的一些帧可以被跳过，以避免渐强和渐弱效应带来的潜在问题。如上面所述，对某些帧使用基于图像的定位技术，以避免表征性文本颜色在渐强或渐弱中缓慢改变。仅基于特征谱的跟踪会导致在这样的情况下破坏分割。

下面描述了如何基本上精确地对准这些位图。首先，就象对图像和网页那样，可以扩展一个文本对象的所有界定文本方框，例如，在水平方向上扩展 20%，在垂直方向上扩展 40%。接着，可以将所有位图转换为灰度，因为灰度对于颜色压缩伪影更稳定。几乎所有的视频压缩算法所表示的亮度比例如著名的 4:2:0 采样方案中的颜色有更高的分辨率。

令  $B_0(x,y), \dots, B_{N-1}(x,y)$  指代所考虑的  $N$  个位图， $B^r(x,y)$  表示要导出的代表性位图，并被初始化为  $B^r_0(x,y) = B_0(x,y)$ 。作为一个例子， $N$  可以为 40，于是有来自 40 帧的 40 个位图。然后，对于每个位图  $B_i(x,y)$ ,  $i \in \{1, \dots, 39\}$ ，我们可以搜索最佳位移  $(dx, dy)$ ，该最佳位移使得对于文本颜色来说， $B^r(x,y)$  与  $B_i(x,y)$  之间的差异最小，例如，

$$(dx_i^{opt}, dy_i^{opt}) = \arg \min \sqrt{\sum_{(x,y) \in B^r \wedge B^r_{i-1}(x,y) \subseteq textColor} (B^r_{i-1}(x-y) - B_i(x+dx, y+dy))^2}$$

这种块匹配搜索能见效的原因是，仅考虑具有文本颜色的像素，其中文本颜色可以是来自小节 4.4.3 的估计文本颜色。当且仅当一个像素与为文本对象确定的文本颜色的差别不大于一个特定量的时候，将该像素定义为具有该文本颜色。应注意，该距离是基于 RGB 值计算的。在每次迭代中，将  $B^r(x,y)$  从前面列出的公式更新为：

对于常规文本

$$B^r_i(x,y) = \max(B^r_{i-1}(x,y), B_i(x+dx_i^{opt}, y+dy_i^{opt}))$$

对于反向文本

$$B^r_i(x,y) = \min(B^r_{i-1}(x,y), B_i(x+dx_i^{opt}, y+dy_i^{opt}))$$

应注意，如果一个文本对象已经在小节 4.4.3 中被识别为静止，我们不必搜索精确的转换。代之以，将各位图之间的转换均设置为无。

通过小节 6.2.2 的处理，对于常规文本来说，背景可能倾向于变得越来越亮，而对于反向文本来说，背景可能变得越来越暗。但是，第一帧可能分别是最亮和最暗的。

### 6.3 二值化（见图 1 中的块 38）

现在准备文本位图  $B^r_i(x,y)$  以便由标准 OCR 工具识别。这里，可以将灰度文本位图转换为白色背景上的黑色文本。下面描述了一种找到合适阈

值的方法，该值是区分文本和背景的一种很好甚至是最佳的值。从小节 4.4.3，我们知道估计的文本颜色，最主要的背景颜色，以及我们必须处理常规文本还是反向文本。由于在小节 6.2 中已经移去大部分背景，我们决定，对于反向文本，将背景颜色设置为黑色，而对于常规文本，将背景颜色设置为白色。然后，将文本颜色亮度与背景颜色亮度中间的亮度选择为二值化阈值是比较好的。对于常规文本，将文本位图中高于该二值化阈值的每个像素设置为白色，而对于反向文本，则将其设置为黑色。对于常规文本，将文本位图中低于该二值化阈值的每个像素设置为黑色，而对于反向文本，则将其设置为白色。最后，我们建议，通过以小节 6.2.1 中所述的方式丢弃较小的区域（设置为背景颜色）来清理二元位图。

### 其他信息

对于上面的每个小节，本发明不限于其中提到的特定细节。

本发明的一些实施例不仅能定位文本的出现并将其分割为较大的二元图像，还能将图像或视频内的每个像素分为属于或不属于文本。因而，我们的文本定位和分割技术可以用于基于对象的视频编码。众所周知，与现有的压缩技术相比，基于对象的视频编码在固定比特率下能获得好得多的视频质量。但是，在大多数情况下，自动提取对象的问题尚未得到解决。对于视频中出现的文本，我们的文本定位和文本分割算法解决了该问题。

本发明的一些实施例涉及一种多分辨率方法，其中，文本定位和文本分割算法能成功地处理 MPEG-1 视频序列直到 HDTV MPEG-2 视频序列（1980x1280），而无需任何参数调整。作为一个例子，字符大小可以在 8 像素和帧高度的一半之间变化。

图 10 示出了具有处理器 184 和存储器 188 的计算机系统 180。存储器 188 表示一个或多个各种类型的存储装置，包括 RAM，硬盘驱动器，CD ROM，和视频存储器等，这里只列举出了几种。存储器 188 包括机器可读的介质，可以在其上存储指令来完成上述的各种功能。存储器 188 也可以存储要处理的数据（例如数字视频信号）和处理的中间及最终结果。可以理解，图 10 是非常简略的，实际上可以包括很多其他公知的元件。

术语“帧”具有较宽的含义。例如，它并不限制于交错的或是非交错的帧。同样，术语“图像”和“视频”也应做较宽的解释。不要求任何特定的格式。

如果说说明书中提到“可以”、“可”或“可能”包括一个元件、特征、结构或特性，则不要求必须包括该特定元件、特征、结构或特性。说明书或权利要求书中提到“一个”元素时，并不是指仅有这一个元素。说明书或权利要求书中提到“一个额外的”元素时，并不排除可以有多于一个所述额外的元素。

本领域的技术人员阅读了此处公开的内容后可以理解，可以在本发明的范围内对前面的说明书和附图的内容做出许多其他的改变。事实上，本发明不限于上述的细节。后附的权利要求限定了本发明的范围，这些权利要求包括了对本发明的任何修改。

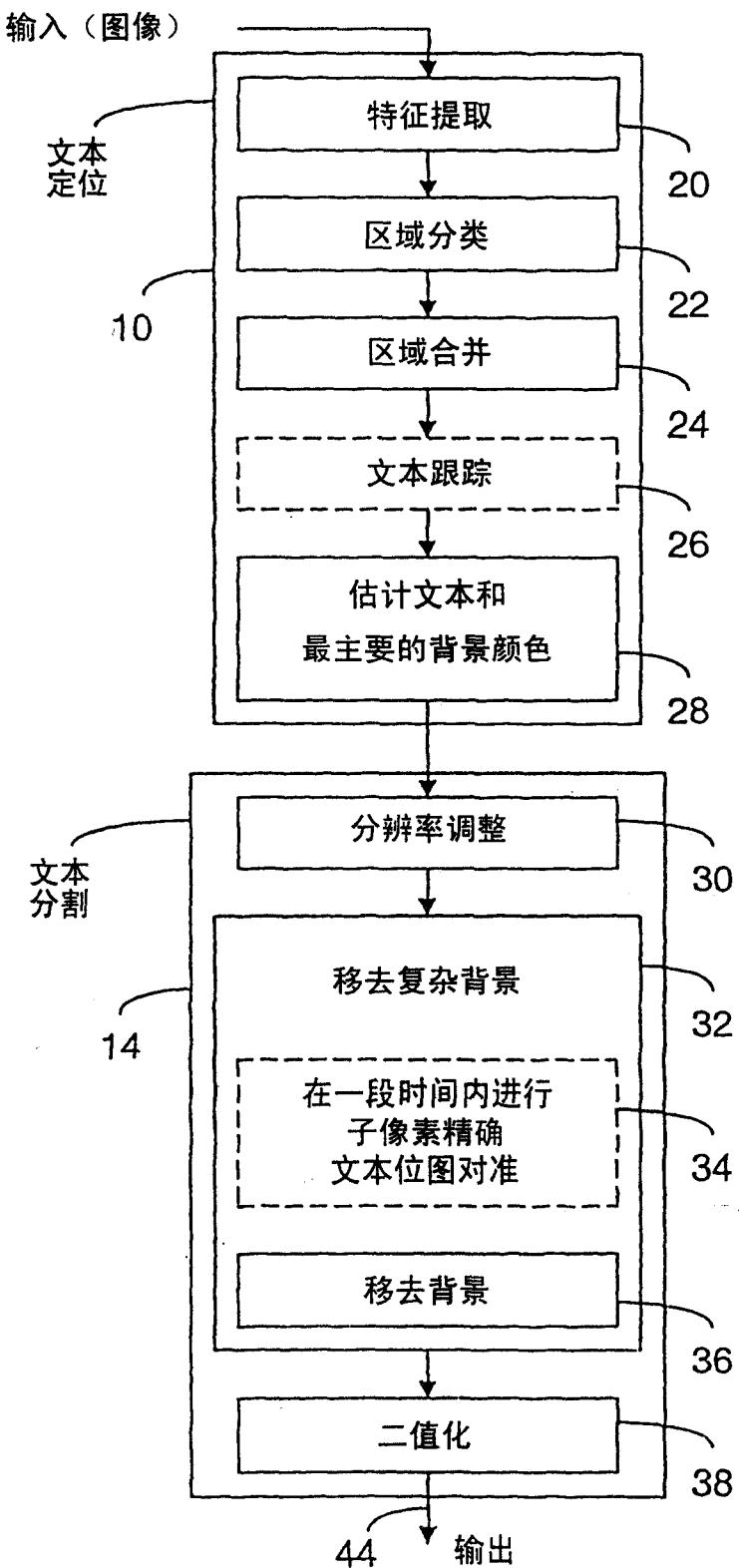
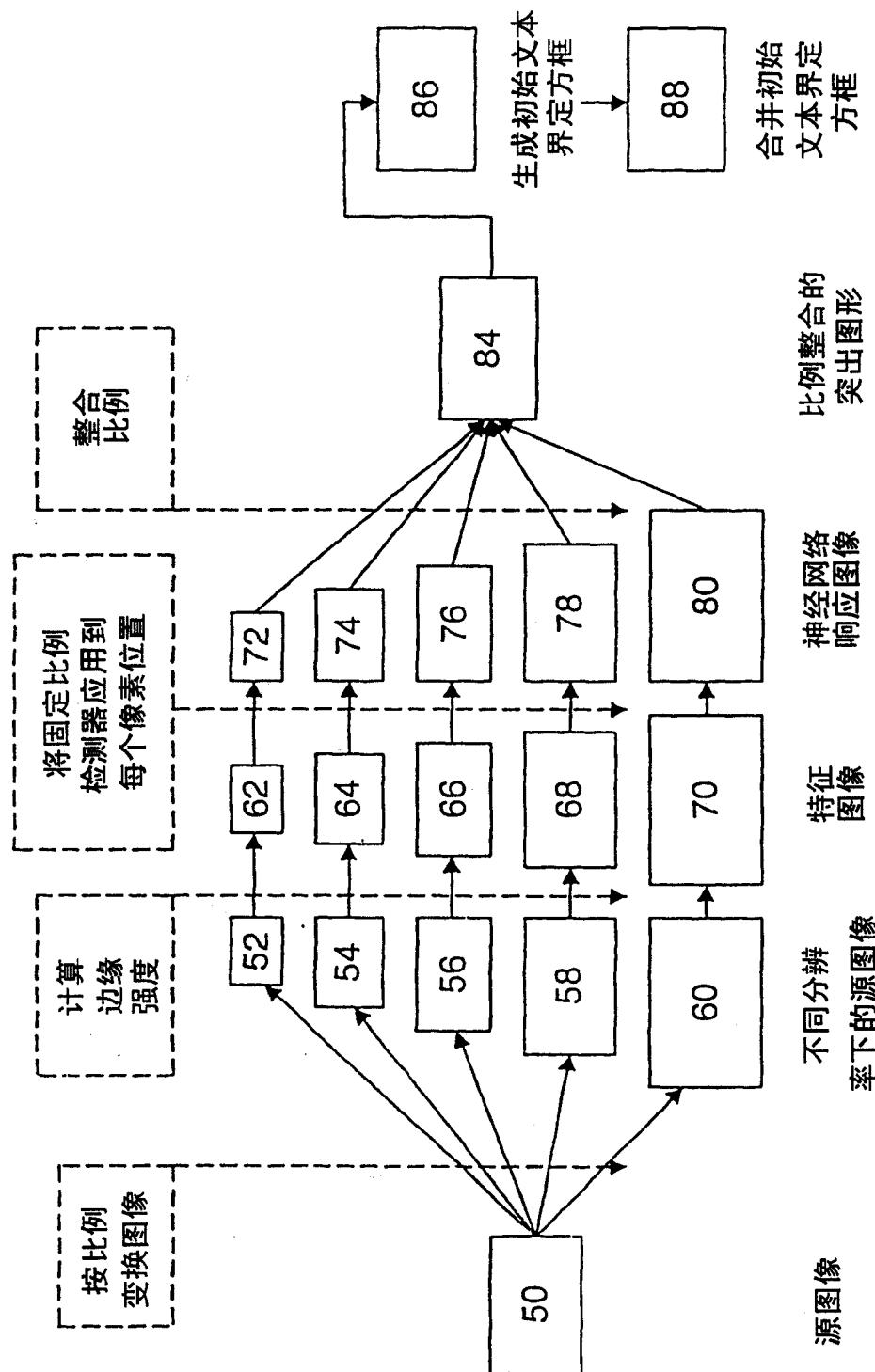


图 1

**图2**

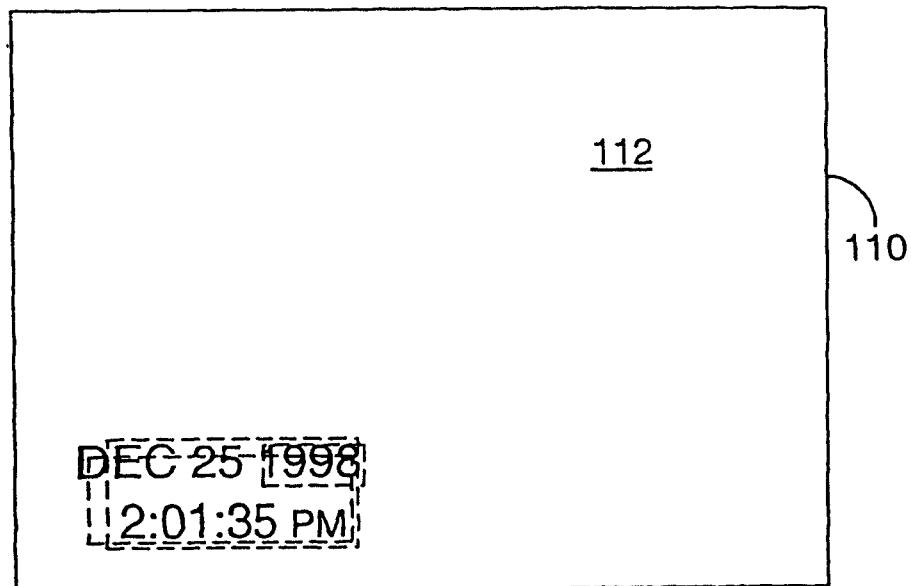


图3

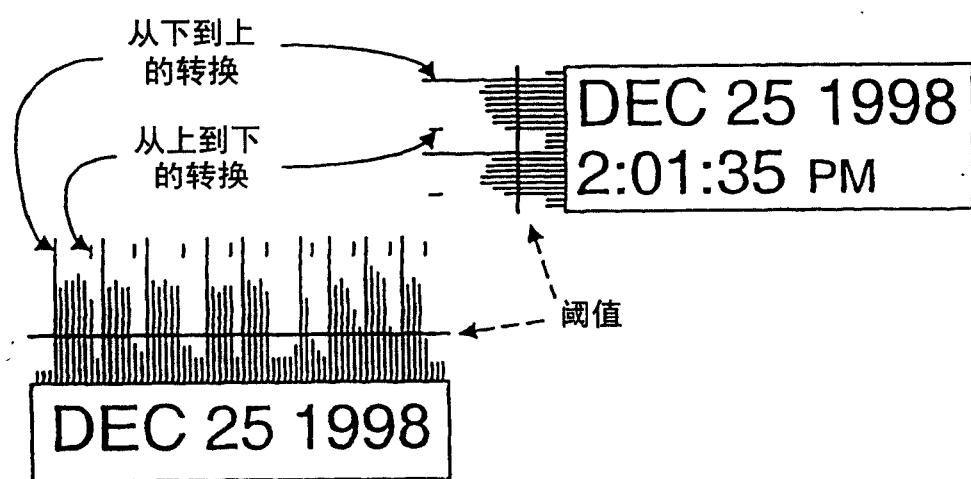


图4

DEC 25 1998  
2:01:35 PM

图5

DEC 25 1998  
2:01:35 PM

图6

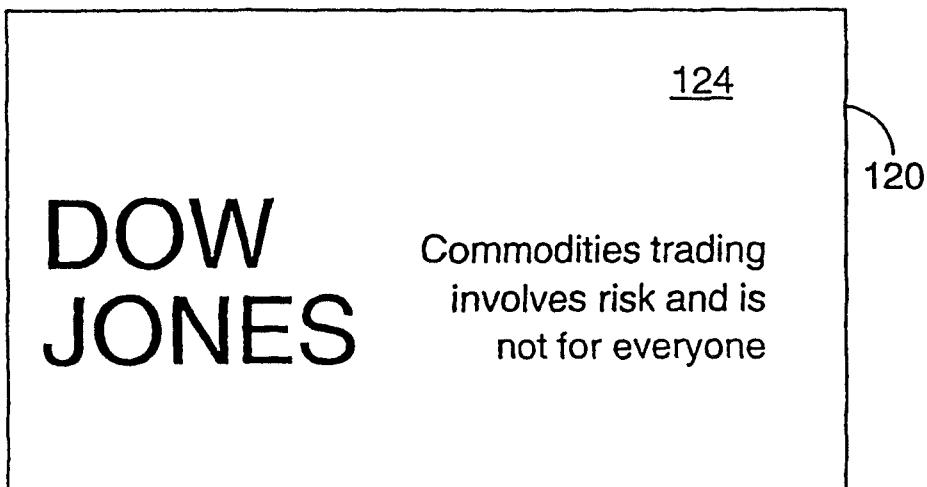


图7

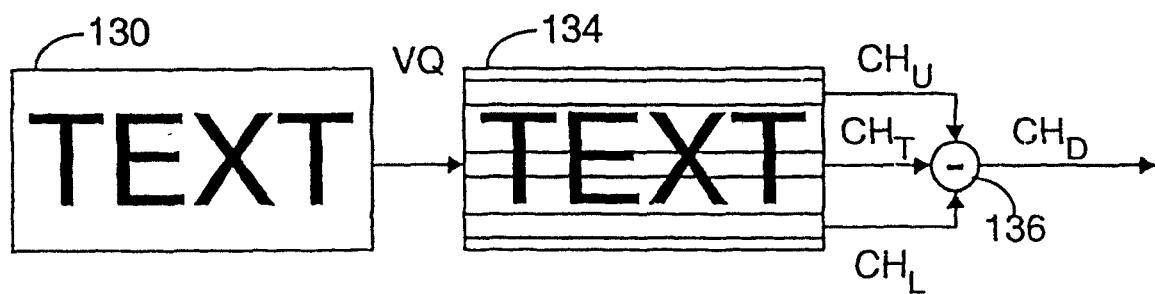


图8

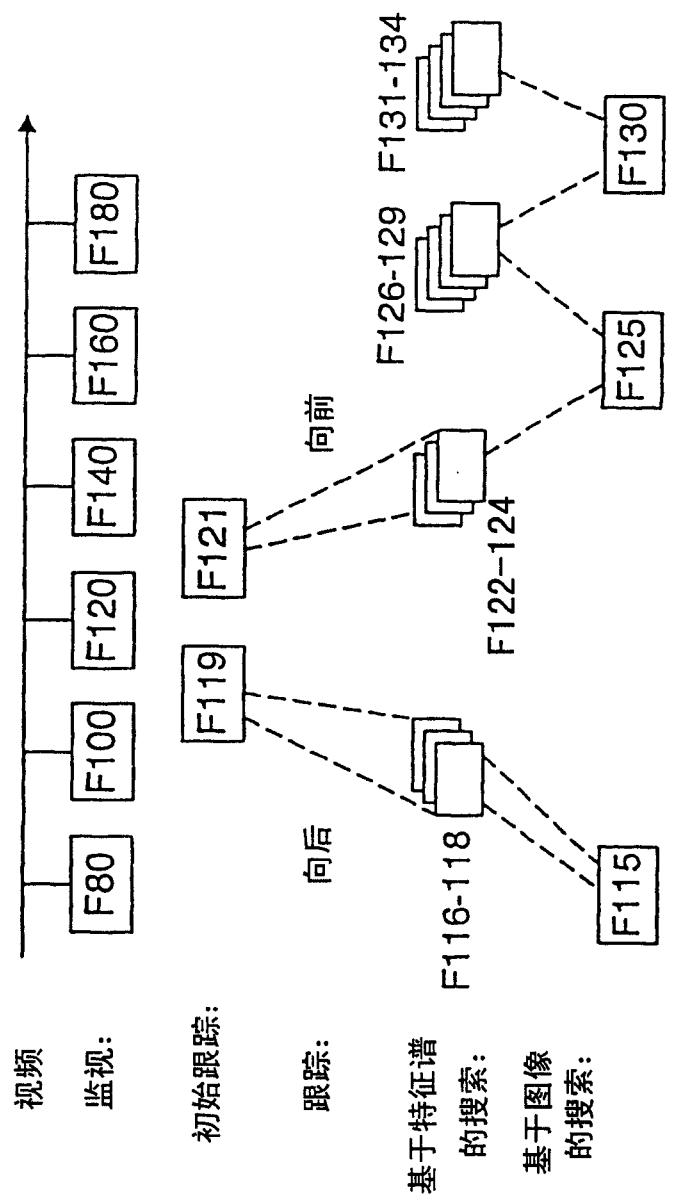


图9

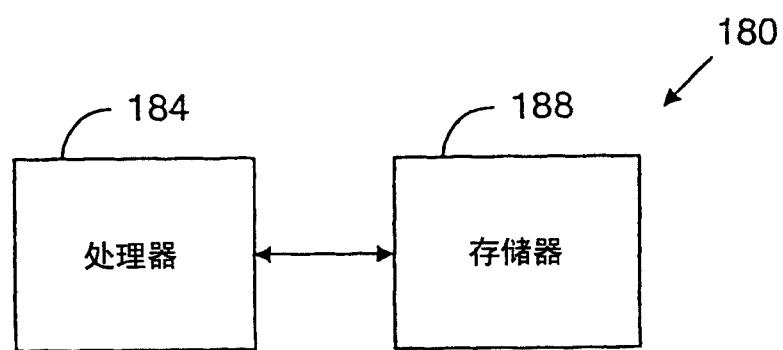


图10