

(12) 发明专利申请

(10) 申请公布号 CN 102236784 A

(43) 申请公布日 2011. 11. 09

(21) 申请号 201010165224. X

(22) 申请日 2010. 05. 07

(71) 申请人 株式会社理光

地址 日本东京都

(72) 发明人 于海龙 李东来 张文波 潘萍萍

宫卫涛

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 黄小临

(51) Int. Cl.

G06K 9/00 (2006. 01)

G06T 7/00 (2006. 01)

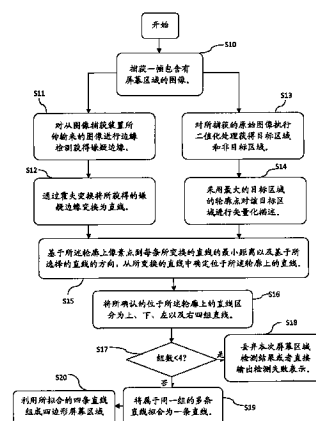
权利要求书 2 页 说明书 9 页 附图 5 页

(54) 发明名称

屏幕区域检测方法及系统

(57) 摘要

本发明提供了一种屏幕区域检测的方法和系统,所述方法包括:捕获含有屏幕的图像;基于所捕获的图像中的每个像素的亮度值的变化,获取屏幕区域的嫌疑边缘;通过霍夫变换,将所获得的嫌疑边缘变换为直线;对所捕获的图像执行二值化处理,其中将图像划分为目标区域和非目标区域;获取最大目标区域的边界点,并将所有边界像素点集合作为相应目标区域的轮廓;基于所述轮廓上像素点到每条所变换的直线的最小距离以及基于所选择的直线的方向,从所变换的直线中确定位于所述轮廓上的直线;基于所选择的直线的坐标位置关系,将所选择的直线区分为上、下、左以及右四组直线;以及将同一组直线中的多条直线拟合为一条直线,从而获得屏幕区域的四个边缘。



1. 一种屏幕区域检测的方法,包括:
捕获含有屏幕的图像;
基于所捕获的图像中的每个像素的亮度值的变化,获取屏幕区域的嫌疑边缘;
通过霍夫变换,将所获得的嫌疑边缘变换为直线;
对所捕获的图像执行二值化处理,其中将图像划分为目标区域和非目标区域;
获取最大目标区域的边界点,并将所有边界像素点集合作为相应目标区域的轮廓;
基于所述轮廓上像素点到每条所变换的直线的最小距离以及基于所选择的直线的方向,从所变换的直线中确定位于所述轮廓上的直线;
基于所选择的直线的坐标位置关系,将所选择的直线区分为上、下、左以及右四组直线;以及
将同一组直线中的多条直线拟合为一条直线,从而获得屏幕区域的四个边缘。
2. 根据权利要求1所述的屏幕区域检测的方法,其中,所述获取屏幕区域的嫌疑边缘通过 Canny、Sobel、或 Harris 边缘检测法实现。
3. 根据权利要求1所述的屏幕区域检测的方法,其中,所述从所变换的直线中确定位于所述轮廓上的直线包括:
计算所述轮廓上每一像素点到相应直线的距离,并从所计算获得距离中选取最小值作为所述相应直线与轮廓的最小距离;
比较所计算获得的最小距离与一距离阈值,将最小距离小于所述距离阈值的直线确定为位于所述轮廓上的直线。
4. 根据权利要求1或3所述的屏幕区域检测的方法,其中,所述从所变换的直线中确定位于所述轮廓上的直线包括:
基于每条直线上个点的横向和纵向的亮度的梯度,计算相应直线的方向;
计算基于距离所述直线最近的轮廓点的横向和纵向的亮度的梯度,计算所述轮廓点的方向;以及
比较所述直线的方向与所述轮廓点的方向,并将两者方向的夹角小于一角度阈值的直线确定为位于所述轮廓上的直线。
5. 根据权利要求3所述的屏幕区域检测的方法,其中,所述距离阈值为5、6、7、8、9或10个像素距离。
6. 根据权利要求3所述的屏幕区域检测的方法,其中,所述角度阈值为20度、30度、40度或45度。
7. 一种屏幕区域检测系统,包括:
图像获取模块,用于捕获含有屏幕的图像;
边缘检测模块,基于所捕获的图像中的每个像素的亮度值的变化,获取屏幕区域的嫌疑边缘;
直线变换模块,通过霍夫变换,将所获得的嫌疑边缘变换为直线;
二值化处理模块,对所捕获的图像执行二值化处理,其中将图像划分为目标区域和非目标区域;
轮廓提取模块,获取最大目标区域的边界点,并将所有边界像素点集合作为相应目标区域的轮廓;

直线选择模块,基于所述轮廓上像素点到每条所变换的直线的最小距离以及基于所选择的直线的方向,从所变换的直线中确定位于所述轮廓上的直线;

直线分组模块,基于所选择的直线的坐标位置关系,将所选择的直线区分为上、下、左以及右四组直线;以及

直线拟合模块,将同一组直线中的多条直线拟合为一条直线,从而获得屏幕区域的四个边缘。

8. 根据权利要求7所述的屏幕区域检测系统,其中,所述边缘检测模块获取屏幕区域的嫌疑边缘过 Canny、Sobel、或 Harris 边缘检测法来机型边缘检测。

9. 根据权利要求7所述的屏幕区域检测系统,其中,所述直线选择模块包括:

距离计算模块,计算所述轮廓上每一像素点到相应直线的距离,并从所计算获得距离中选取最小值作为所述相应直线与轮廓的最小距离;

距离比较模块,比较所计算获得的最小距离与一距离阈值,将最小距离小于所述距离阈值的直线确定为位于所述轮廓上的直线。

10. 根据权利要求7或9所述的屏幕区域检测系统,其中,所述直线选择模块包括:

方向计算模块,基于每条直线上个点的横向和纵向的亮度的梯度,计算相应直线的方向,并且基于距离所述直线最近的轮廓点的横向或纵向的亮度的梯度,计算所述轮廓点的方向;以及

角度比较模块,比较比较所述直线的方向与所述轮廓点的方向,并将两者方向的夹角小于一角度阈值的直线确定为位于所述轮廓上的直线。

屏幕区域检测方法及系统

技术领域

[0001] 本发明涉及图像处理领域,具体而言,涉及图像分割领域,尤其涉及一种从所捕获的图像中检测屏幕区域的方法以及使用这种方法的系统。

背景技术

[0002] 在各种研讨会中,采用投影仪作为辅助工具对于演讲的进行非常有帮助。但是在演讲的过程中,演讲人需要频繁地操作计算机来切换投影屏幕上的内容,或者需要助手根据演讲人员演讲的进度来操作与投影仪连接的计算机来切换投影屏幕上的内容。这种利用投影仪的方式虽然为演讲人带来了极大的方便,但是其操作过程依然非常费时或者费力。

[0003] 为此,人们提出了一种能够通过直接点击投影屏幕上的内容来操作计算机的方法,即通过成像装置获取包含投影屏幕的图像,通过计算机提取所获取的图像中的投影屏幕的屏幕区域位置信息,并利用屏幕区域的位置信息以及对投影屏幕中内容的操作来切换所需投影的内容。

[0004] 为了提取所获取的图像中的特定区域,美国专利申请公开号 US20040165786A 披露了一种用于提取图像中的白板内容并将所提取的白板内容转换为电子文档的方法和装置。该专利申请为了能够自动识别所拍摄图像中的白板,采用了一种四边形检测算法来识别白板区域。但是,根据所述专利申请描述的方法,往往会在白板边缘检测到多条平行直线,因此很难精确确定这些直线中哪一条是白板边缘,由此导致所检测到的白板与实际白板之间存在较大的误差。

[0005] 另一美国专利申请公开号 US20080266253A 披露了一种在计算机投影区域追踪激光点的系统。在该专利所述的系统中,采用了一种屏幕区域检测方法,该方法通过二值化捕获图像,并从二值化图像中筛选四边形来获取投影区域。该方法很大程度上依赖于所捕获图像的质量,因此,为了提高所捕获的图像的质量,需要为图像捕获设备加装特殊滤镜。

[0006] 另一美国专利 US6618076 披露了一中校准投影仪摄像头系统的装置。该专利所披露的技术方案通过投影仪投射特殊图案来进行特征提取。

[0007] 因此,人们希望能够有一种方法和设备能够在不增加硬件设备的情况下,更精确地检测屏幕区域,从而使得演讲者在操作投影系统时能够更准确地切换投影内容。

发明内容

[0008] 本发明提供了一种解决上述现有技术中问题的方法。具体而言,提供了一种如何在采用交互式虚拟白板的系统中,精确地获取虚拟白板的边缘,从而获得投射在投影屏幕上的指示点相对于投影屏幕的精确坐标位置的方法,从而能够向采用交互式虚拟白板的系统输出指示点在虚拟白板中的精确坐标点信息,以便能够使用指示点与计算机进行交互。但是,为了精确地检测指示点(例如激光点)在捕获图像中的位置,捕获图像的亮度需要很低。这就使得采用现有的技术在捕获图像中对屏幕区域的检测相对困难,很难用一般的角点检测算法获得。

[0009] 为克服上述技术问题,本发明提供了一种屏幕区域检测的方法,包括:捕获含有屏幕的图像;基于所捕获的图像中的每个像素的亮度值的变化,获取屏幕区域的嫌疑边缘;通过霍夫变换,将所获得的嫌疑边缘变换为直线;对所捕获的图像执行二值化处理,其中将图像划分为目标区域和非目标区域;获取最大目标区域的边界点,并将所有边界像素点集合作为相应目标区域的轮廓;基于所述轮廓上像素点到每条所变换的直线的最小距离以及基于所选择的直线的方向,从所变换的直线中确定位于所述轮廓上的直线;基于所选择的直线的坐标位置关系,将所选择的直线区分为上、下、左以及右四组直线;以及将同一组直线中的多条直线拟合为一条直线,从而获得屏幕区域的四个边缘。

[0010] 根据本发明的屏幕区域检测的方法,所述获取屏幕区域的嫌疑边缘通过 Canny、Sobel、或 Harris 边缘检测法实现。

[0011] 根据本发明的屏幕区域检测的方法,其中,所述从所变换的直线中确定位于所述轮廓上的直线包括:计算所述轮廓上每一像素点到相应直线的距离,并从所计算获得距离中选取最小值作为所述相应直线与轮廓的最小距离;以及比较所计算获得的最小距离与一距离阈值,将最小距离小于所述距离阈值的直线确定为位于所述轮廓上的直线。

[0012] 根据本发明的屏幕区域检测的方法,所述从所变换的直线中确定位于所述轮廓上的直线还包括:基于每条直线上个点的横向和纵向的亮度的梯度,计算相应直线的方向;计算基于距离所述直线最近的轮廓点的横向和纵向的亮度的梯度,计算所述轮廓点的方向;以及比较比较所述直线的方向与所述轮廓点的方向,并将两者方向的夹角小于一角度阈值的直线确定为位于所述轮廓上的直线。

[0013] 根据本发明的屏幕区域检测的方法,其中,所述距离阈值为 5、6、7、8、9 或 10 个像素距离。

[0014] 根据本发明的屏幕区域检测的方法,其中,所述角度阈值为 20 度、30 度、40 度或 45 度。

[0015] 根据本发明的另一个方面,还提供了一种屏幕区域检测的系统,包括:图像获取模块,用于捕获含有屏幕的图像;边缘检测模块,基于所捕获的图像中的每个像素的亮度值的变化,获取屏幕区域的嫌疑边缘;直线变换模块,通过霍夫变换,将所获得的嫌疑边缘变换为直线;二值化处理模块,对所捕获的图像执行二值化处理,其中将图像划分为目标区域和非目标区域;轮廓提取模块,获取最大目标区域的边界点,并将所有边界像素点集合作为相应目标区域的轮廓;直线选择模块,基于所述轮廓上像素点到每条所变换的直线的最小距离以及基于所选择的直线的方向,从所变换的直线中确定位于所述轮廓上的直线;直线分组模块,基于所选择的直线的坐标位置关系,将所选择的直线区分为上、下、左以及右四组直线;以及直线拟合模块,将同一组直线中的多条直线拟合为一条直线,从而获得屏幕区域的四个边缘。

[0016] 通过上述方法,获得屏幕区域的四个边缘,从而能够输出屏幕区域各角点坐标。

[0017] 根据本发明的屏幕区域检测的系统,所述边缘检测模块获取屏幕区域的嫌疑边缘通过 Canny、Sobel、或 Harris 边缘检测法来机边缘检测。

[0018] 根据本发明的屏幕区域检测的系统,所述直线选择模块包括:距离计算模块,计算所述轮廓上每一像素点到相应直线的距离,并从所计算获得距离中选取最小值作为所述相应直线与轮廓的最小距离;距离比较模块,比较所计算获得的最小距离与一距离阈值,将最

小距离小于所述距离阈值的直线确定为位于所述轮廓上的直线。

[0019] 根据本发明的屏幕区域检测的系统,所述直线选择模块包括:方向计算模块,基于每条直线上个点的横向和纵向的亮度的梯度,计算相应直线的方向,并且基于距离所述直线最近的轮廓点的横向或纵向的亮度的梯度,计算所述轮廓点的方向;以及角度比较模块,比较比较所述直线的方向与所述轮廓点的方向,并将两者方向的夹角小于一角度阈值的直线确定为位于所述轮廓上的直线。

[0020] 本发明通过提供上述屏幕区域检测方法,获得屏幕的四个顶点精确位置,从而能够将检测出指示点在所捕获图像中的坐标正确转换为所检测的屏幕区域中的坐标。因此,所转换获得的指示点在所检测的屏幕区域中的坐标的精确度很大程度上依赖于屏幕区域检测的精确度,因此,由于本发明能够提供幕的四个顶点精确位置,因此指示点在所检测的屏幕区域中的坐标也就比较精确。

附图说明

[0021] 图 1 所示的是根据本发明的屏幕区域检测方法的流程图;

[0022] 图 2 所示的是一张所捕获的包含屏幕的原始图像的实例;

[0023] 图 3A 所示的是一张对图 2 所示的原始图像执行屏幕区域边缘检测处理后的图像的实例;

[0024] 图 3B 所示的是一张对图 3A 所示的图像执行直线变换后的图像的实例;

[0025] 图 3C 所示的是一张对图 2 所示的原始图像执行二值化处理后的图像的实例;

[0026] 图 3D 所示的是一张是基于图 3C 所示的图像执行确定屏幕区域边缘的处理后的图像的实例;

[0027] 图 4 所示的是将图 3B 所示的图像中的直线进行标注处理后的图像的实例;

[0028] 图 5 所示的是采用本发明的屏幕区域检测方法的系统的示意图;

[0029] 图 6 所示的是构成图 5 所示的系统的方框图。

具体实施方式

[0030] 以下,将参照附图来详细说明本发明的具体实施例。

[0031] 图 1 所示的是根据本发明的屏幕区域检测方法的流程图。

[0032] 如图 1 所示,首先,在步骤 S10 中,通过图像捕获装置捕获一帧包含有屏幕区域的图像,如图 2 所示,并将所捕获的图像传输给图 5 中所示的计算机 51,以便进行随后的处理。这种图像捕获装置如图 5 中所示的摄像头 53。图 2 所示的示例性图像为带有计算机屏幕的 256 阶灰度图像。这仅仅是一种距离说明,可以根据具体的需要来确定图像的灰度阶。同样,如果采用附图 5 中的摄像头 53 来拍摄,其所拍摄的一帧图像中的屏幕则不是计算机屏幕,而是投影屏幕 54。在实际应用情况下,所拍摄的图像中的屏幕区域占所拍摄图像的面积的一半以上。

[0033] 随后,在步骤 S11 中,计算机对从图像捕获装置所传输来的图像进行边缘检测,即在所捕获的图像中获取屏幕区域的嫌疑边缘。关于边缘检测,目前本发明所属技术领域存在多种方法,这些方法大致有两类:基于搜索和基于零交叉。所谓基于搜索的边缘检测,首先要计算边缘强度,通常用一阶导数表示,例如梯度模,然后,通常采用梯度的方向估计边

缘的局部方向,并利用此方向找到局部梯度模的最大值。所谓基于零交叉的方法,就是找到由图像得到的二阶导数的零交叉点来定位边缘,通常用拉普拉斯算子或非线性微分方程的零交叉点。

[0034] 本发明所举的一个例子中的方法为基于搜索的边缘检测。下面大致描述一下基于搜索的边缘检测方法。

[0035] 首先,计算一阶导数。许多边缘检测操作都是基于亮度的一阶导数,由此获得所捕获的图像的原始数据亮度的梯度。使用所获得的梯度,可以在图像的亮度梯度中搜寻峰值。如果 $I(x)$ 表示像素点 x 的亮度, $I'(x)$ 表示像素 x 的一阶导数(即亮度梯度),这样我们就会发现:

$$I'(x) = -1/2 \cdot I(x-1) + 0 \cdot I(x) + 1/2 \cdot I(x+1), \quad (1)$$

[0037] 对于更高性能的图像处理来说,一阶导数能够通过带有掩码的原始数据(1维)卷积计算得到。

[0038] 其次,确定阈值。一旦计算出亮度的一阶导数之后,下一步要做的就是给出一个阈值来确定那里是边缘位置。

[0039] 在基于搜索的边缘检测方法,所采用边缘检测算子有 Roberts Cross 算子, Prewitt 算子, Sobel 算子, Canny 算子。目前, Canny 算子(或者这个算子的变体)是最常用的边缘检测方法。在采用 Canny 算子边缘检测方法中,采用了一种用于边缘检测最优预平滑滤波器,该滤波器能够很好地被一阶高斯导数核优化。此外,采用 Canny 算子的边缘检测方法中引入了非最大抑制概念,将边缘定义为在梯度方向具有最大梯度值的点。在一种采用 Canny 算子边缘检测方法中,使用了两个阈值,两个阈值比使用一个阈值更加灵活。通常,阈值越低,能够检测出的边线越多,结果也就越容易受到图像中噪声信息的影响,并且越容易从图像中挑出不相关的特性。与此相反,一个过高的阈值将会遗失较细的或者较短的边缘。

[0040] 在此,具体描述一种基于图像中的像素的亮度值来进行的边缘检测,即,基于所捕获的图像中的每个像素的亮度值的变化,获取屏幕区域的嫌疑边缘。

[0041] 目前在计算机图像处理领域中,边缘检测的目的都是检测图像中明暗急变的特征。如图 2 所示的图像,图像明暗急变这一特征符合屏幕区域亮度与周边亮度变化特征,这是由于屏幕区域或者投影区域往往较周围环境要亮。

[0042] 在发明的一种实施方式中,采用了如上所述的 Canny 边缘检测算法。这是因为 Canny 边缘检测算法是一种多阶段的检测算法,能够在图像处理中检测较宽范围的边缘信息,这使得在调试过程中能够更容易的适应不同的使用环境。所述边缘检测方法的具体步骤为:

[0043] 首先,消除噪声。Canny 边缘检测一般使用高斯平滑消除噪声。通过使用高斯模板对原始图像进行卷积运算来平滑图像中的噪声。发明中使用如下高斯模板对图像进行平滑。

$$[0044] \quad B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A \quad (2)$$

[0045] 其中, A 是原图像, B 是平滑后图像。

[0046] 其次, 计算图像的亮度梯度及方向。图像中的边缘可能会是不同方向的, 因此, Canny 边缘检测使用了四个模板分别检测平滑图像中水平、垂直及对角线方向的边缘。通过使用边缘检测算子 (如 Roberts, Prewitt, Sobel 等) 计算得到图像中各像素点的水平亮度梯度 (G_y) 与垂直亮度梯度 (G_x), 通过如下公式计算图像各像素点亮度梯度 (G) 和亮度梯度方向 (θ) :

$$[0047] \quad G = \sqrt{G_x^2 + G_y^2} \quad (3)$$

$$[0048] \quad \theta = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (4)$$

[0049] 第三, 在图像中跟踪边缘。较高的亮度梯度比较有可能是边缘, 但是没有确切的价值来限定多大的亮度梯度是边缘多大又不是, 所以 Canny 使用了滞后阈值。滞后阈值需要两个阈值——高阈值 (T_1) 与低阈值 (T_2)。假设图像中的重要边缘都是连续的曲线, 这样我们就可以跟踪给定曲线中模糊的部分, 并且避免将没有组成曲线的噪声像素当成边缘。所以我们从一个较大的阈值开始, 这将标识出我们比较确信的真正边缘, 使用前面导出的方向信息, 我们从这些真正的边缘开始在图像中跟踪整个的边缘。在跟踪的时候, 我们使用一个较小的阈值, 这样就可以跟踪曲线的模糊部分直到我们回到起点。一旦这个过程完成, 我们就得到了一个二值图像, 每点表示是否是一个边缘点。

[0050] 该发明中使用了一种自动阈值计算方法, 以使得该发明在不同的环境中能够自动适应。其自动阈值计算方法如下:

[0051] 首先, 计算所捕获的图像的所有像素点的平均亮度 L_{Avg} 。其计算公式为:

$$[0052] \quad L_{Avg} = \frac{\sum L_{x,y}}{N} \quad (4)$$

[0053] 其中 $L_{x,y}$ 是在 (x, y) 处像素点的亮度值, N 是图像中像素点的数量。

[0054] 然后, 根据平均亮度 L_{Avg} , 将所有像素点分为大于平均亮度的像素点 (P_h) 与小于平均亮度 L_{Avg} 的像素点 (P_l) 两类。

[0055] 其次, 对这两类像素点再分别使用公式 (4) 计算各类像素点的平均亮度, 并将两类的平均亮度分别记为 $L_{h_{Avg}}$ 和 $L_{l_{Avg}}$ 。最后, 根据如下公式计算两个阈值 $T1$ 与 $T2$:

$$[0056] \quad T1 = L_{h_{Avg}} - L_{l_{Avg}} \quad (5)$$

$$[0057] \quad T2 = T1 \times 0.8 \quad (6)$$

[0058] 根据如上描述, 从而将图 2 中所示的示例性图像中的嫌疑边缘检测出来。显然, Canny 边缘检测算法由于使用了两个阈值 $T1$ 与 $T2$, 因此人们在实际应用过程中可以通过调节这两个阈值使得在捕获图像中能够检测出更多屏幕边界上的边缘, 同时排除更多其他的边缘。根据本发明的实施方式, 由于这两个阈值 $T1$ 与 $T2$ 源于所捕获的图像的所有像素点的平均亮度 L_{Avg} , 因此, 两个阈值 $T1$ 与 $T2$ 是通过自动计算方法获得, 因此, 提高了边缘检测

的效率。

[0059] 随后,在步骤 S12 处,计算机对在步骤 S11 处所处理获得的图像(例如图 3A 中的图像)中的嫌疑边缘执行直线变换处理,通过霍夫变换,将所获得的嫌疑边缘变换为直线。霍夫曼变换通常用于图像分析、计算机视觉及数字图像处理等领域的一种技术,其目的是在图像处理中识别几何形状。

[0060] 在本发明的实施方式中,使用了一种本领域技术人员所常用的在图像中识别直线的简单霍夫变换,即,利用 OpenCV 中已有的函数 HoughLinesP,通过基于几率的霍夫变换算法进行直线检测。函数 HoughLinesP 中有 5 项参数,它们分别是:与像素相关单位的距离精度(Rho)、弧度测量的角度精度(Theta)、阈值参数(Threshold)、最小线段长度(minLineLength)、以及在同一条直线上进行碎线段连接的最大间隔值(maxLineGap),其中,当相应的累计值大于阈值参数 threshold,则函数 HoughLinesP 返回的这个线段。作为本发明的一个具体实施方式,可以给以上参数赋予如下对应值:

[0061] $\text{Rho} = 1$

[0062] $\text{Theta} = \pi / 90$

[0063] $\text{Threshold} = 20$

[0064] $\text{minLineLength} = 20$

[0065] $\text{maxLineGap} = 5$

[0066] 以上值仅为参考,具体参数值与实现环境有关,受到如拍摄图片大小、图片边缘检测结果等影响。因此具体参数值可以根据具体情况进行调整。

[0067] 在执行步骤 S11 和 S12 的同时,计算机在步骤 S13 中对所捕获的原始图像(例如图 2 所示的图像)执行二值化处理。图像的二值化是一个基本的像素分割问题。其目的是把一幅给定的图像分割成两类:属于目标的像素和属于背景的像素。阈值化是图像分割中一个最常用的工具。多年来,已有各种各样的阈值选择方法提出来,如 Otsu 算法、最大熵阈值等。其中 Otsu 算法就是一种用来转换灰度图像为二值图像的简单高效方法。

[0068] 一般而言,由于屏幕区域亮度大于非屏幕区域亮度,而屏幕区域面积占整个图像面积一半以上,因此在以下本发明的一种实施方式中,可以直接使用在边缘检测步骤 S11 中计算所得平均亮度 L_{Avg} 作为图像二值化的阈值。具体而言,首先,计算所捕获的图像的所有像素点的平均亮度 L_{Avg} ,随后,以所计算平均亮度值作为阈值,对所捕获的图像执行二值化处理,即,将亮度大于所述平均亮度值 L_{Avg} 的像素点设置为白色,其余像素设置为黑色。具体到本发明的所涉及的屏幕区域检测来说,二值化转换的目的是根据所捕获的图像的像素的亮度特征将所捕获的图像中的像素粗略分为屏幕区域(白色)与非屏幕区域(黑色),从而获得图 3C 所示的图像。二值化阈值的设定在该步骤中非常重要,本发明能够根据图像的具体情况来自动确定二值化阈值,因此能够提高本发明屏幕区域检测的效果。

[0069] 尽管上面所举例的实施方式中采用了 Otsu 算法来执行图像二值化过程,但是,采用最大熵阈值同样可以执行图像二值化过程。因此上述举例并不对本发明并不具有限制性。

[0070] 接着,在步骤 S14 处,计算机获取由被设置为白色的像素点构成的最大的连通白色区域的边界点,并将所有边界像素点集合作为相应连通白色区域的轮廓,即检测在步骤 S13 中所获得的二值化图像中的最大的连通白色区域,并通过将最大的连通白色区域边界点集合作为轮廓的矢量来对该最大的连通白色区域进行矢量化描述。

[0071] 在本发明的一种实施方式中,通过 OpenCV 中 cvFindContours 函数来检测所述二值化图像中的最大白色连通区域的轮廓。如图 3C,二值化图像中有多个互补连通的白色区域,因此通过轮廓检测也会得到多个轮廓信息。由于屏幕区域在图像中是最大区域,我们可以选择面积最大的轮廓作为屏幕区域。

[0072] 接着,在步骤 S15 处,基于所述轮廓上像素点到每条所变换的直线的最小距离以及基于所选择的直线的方向,从所变换的直线中确定位于所述轮廓上的直线。具体而言,就是确定图 3B 中的那些直线中的那些直线属于屏幕区域的轮廓线。

[0073] 由于在边缘检测步骤 S11 以及直线变换步骤 S12 中的噪声干扰,所变换获得的直线很多都与屏幕区域边界无关,这些无关的直线我们称之为干扰直线。因此在步骤 S15 处所进行的处理就是消除这些干扰直线,以降低屏幕区域检测的误差。如图 4 所示,在步骤 S12 的直线变换处理后所获得的直线有很多与屏幕区域无关(例如图 4 中直线 7~18),这些无关直线也并不在屏幕边缘上。在本发明的一个具体实施方式中,通过计算每条直线与屏幕区域轮廓的位置关系及相应直线的方向来确定该直线是否属于屏幕区域的轮廓。

[0074] 首先,选取任意一条直线,计算该直线与屏幕区域的轮廓的最小距离。具体而言,如前所述,屏幕区域的轮廓是通过屏幕区域的轮廓上的像素点构成的矢量来描述的,因此在计算一条直线与屏幕区域的轮廓的最小距离时,需要计算屏幕区域轮廓上每一像素点到该直线的距离,并从针对该直线的所有距离值中选取最小值作为该直线与屏幕区域轮廓的最小距离。通过设定一个距离阈值,可以将所述有直线初步区分为属于屏幕边缘的直线与干扰直线。该距离阈值的大小取决于所捕获的图像的大小及屏幕区域在图像中所占比例的大小。在根据本发明的实施方式中,所捕获的图像(如图 2 所示)的大小为 320*240,屏幕区域面积占图像的 60%以上,可以将该距离阈值设为 5、6、7、8、9、或 10 个像素距离。如果选择阈值距离为 6 个像素,则将大于 6 个像素的最小距离所对应的相知确定为干扰直线,反之则确定为属于屏幕区域轮廓的直线。以图 4 所示的情况而言,图中标号为 1~8 的直线被认作在属于屏幕区域边缘的直线,而标号为 9~18 的直线被认作干扰直线。

[0075] 其次,通过计算直线的方向进一步确认哪些直线属于屏幕区域边缘的直线以及哪些直线是干扰直线。具体而言,就是计算每条直线的方向以及与该条直线距其最近的轮廓上像素点的方向,计算两个方向的差值(也就是两个方向的夹角),如果该差值小于或等于一预定角度阈值(例如 30°、35°、40°、或 45°),则认为该直线属于屏幕区域边缘的直线,否则为干扰直线。

[0076] 在本发明的一个具体实施方式中,直线方向是通过使用某一特定算子,计算直线上每一像素点的亮度梯度得到。这个算子使用两个 3×3 矩阵卷积图像中的点,一个用来计算纵向亮度梯度,另一个用来计算横向亮度梯度。设 A 为原图像,Gx 和 Gy 分别表示横向亮度梯度与纵向亮度梯度,则 Gx 与 Gy 的计算公式如下:

$$[0077] \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (7)$$

$$[0078] \quad \text{以及 } G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad (8)$$

[0079] 其中 * 号表示二维卷积计算。

[0080] 通过以上公式计算得出的 G_x 与 G_y , 直线的方向可以通过下面的公式计算得出角度 θ :

$$[0081] \quad \theta = \sin^{-1} \left(\frac{G_x}{\sqrt{G_x^2 + G_y^2}} \right) \quad (9)$$

[0082] 采用同样的方式, 可以计算屏幕边缘轮廓上各点的方向。通过比较直线的方向与距该直线最近的轮廓像素点的方向, 可以确认该直线是否属于屏幕边缘轮廓。通过比较, 可以将图 4 中标号为 7、8 的两条直线认为不属于屏幕区域轮廓的干扰直线。而标号为 1 ~ 6 的直线最终被确认属于屏幕区域轮廓的直线。

[0083] 接着, 在步骤 S16 中, 基于在步骤 S15 所确认的属于屏幕区域轮廓的直线的坐标位置关系, 将所确认的直线区分为上、下、左以及右四组直线。一般来讲, 屏幕区域为四边形, 因而根据各直线的位置关系分成屏幕的上边缘、下边缘、左边缘及右边缘四组。对于屏幕区域轮廓, 可以分别计算其横向与纵向位置坐标最大与最小值。对于屏幕区域轮廓上的各条直线, 可以利用已计算出的直线方向 θ 及其位置将其分为上、下、左、右四组, 并进入步骤 S17。

[0084] 在步骤 S17 处, 计算机判断所分的直线的组数是否小于四组, 如果小于四组, 则计算机在步骤 S18 处执行操作丢弃本次屏幕区域检测结果而不对用户的操作作出任何反应, 或者直接输出检测失败表示。在本发明的具体实施方式中, 图中标号为 1 ~ 6 的直线分组如下: 直线 1 是左边缘直线, 直线 2、3 是上边缘直线, 直线 4 是右边缘直线, 直线 5、6 是下边缘直线。

[0085] 如果在步骤 S17 处, 计算机判断所分的直线的组数等于四组, 则在步骤 S19 处将同一组直线中的多条直线拟合为一条直线, 从而获得屏幕区域的四个边缘。通常, 如上面以图 4 中的带标号的直线为例所描述的那样, 在步骤 S16 中所分组获得的屏幕区域轮廓的上、下、左、右各组中可能含有多条直线。为了提高屏幕监测区域的精确度, 可以将各边缘上属于同一组的多条的直线进行拟合, 从而获得更准取得屏幕区域轮廓边缘。对于直线拟合有很多的已知算法可以使用。

[0086] 常用的直线拟合方法是最小二乘法直线拟合。

[0087] 假设所给数据点 $(x_i, y_i), i = 1, 2, \dots, N$ 的大致分成一直线, 构造拟合直线 $y = a + bx$, 并不要求其严格地通过所有的数据点 (x_i, y_i) , 希望它尽可能地从所给数据点附近通过。设 $\hat{y}_i = a + bx_i, i = 1, 2, \dots, N$ 表示按拟合直线 $y = a + bx$ 求得的近似值, 称两者之差 $e_i = y_i - \hat{y}_i$ 为残差。残差的大小是衡量拟合好坏的重要标志, 具体有三种准则:

[0088] (i) 使残差的最大绝对值为最小: $\min(\max_i |e_i|)$

[0089] (ii) 使残差的绝对值之和为最小: $\min \sum_i |e_i|$

[0090] (iii) 使残差的平方和为最小: $\min \sum_i e_i^2$

[0091] 基于准则 (iii) 来选取拟合直线的方法称作直线拟合的最小二乘法, 解方程:

$$[0092] \quad \begin{cases} aN + b \sum x_i = \sum y_i \\ a \sum x_i + b \sum x_i^2 = \sum x_i y_i \end{cases} \quad (10)$$

[0093] 在本发明的具体实施方式中, 还可以使用提取同一组中各条直线的端点, 通过对

所有端点的拟合求得一条近似直线,并以该近似直线作为最终屏幕区域的边缘。

[0094] 最后,在步骤 S20 中,利用在步骤 S19 中拟合处的四条直线组成构成屏幕区域的四边形。

[0095] 通过上述根据本发明的屏幕区域检测方法,在交互式虚拟白板中,能够自动实时地检测屏幕区域位置,因此对于不同的使用条件具有一定的鲁棒性,由此大大提高系统的用户体验。

[0096] 图 5 所示的是采用本发明的屏幕区域检测方法的系统的示意图。该系统包括:计算机 501、投影仪 502、摄像头 503 以及投影屏幕 504。计算机 501 运行应用程序,并将所需的内容输出到投影仪 502。投影仪 502 将相关的内容投射到投影屏幕 504 上。摄像头 503 按照预定的规则拍摄包含投影屏幕 54 的图像并将图像发送到计算机 501,计算机 501 在接收到该图像之后执行上述步骤 S11-S20,进行屏幕区域检测,从而精确地获得所拍摄的图像中的屏幕区域。

[0097] 图 6 所示的是构成图 5 所示的系统 600 的方框图。所述系统 600 包括:图像捕获模块 601,用于捕获含有屏幕的图像;边缘检测模块 602,基于所捕获的图像中的每个像素的亮度值的变化,获取屏幕区域的嫌疑边缘;直线变换模块 603,通过霍夫变换,将所获得的嫌疑边缘变换为直线;图像二值化模块 604,用于对所捕获的图像执行二值化处理,其中将图像划分为目标区域和非目标区域,具体而言,就是基于所捕获的图像中的每个像素的亮度值的平均亮度值,对所捕获的图像执行二值化处理,其中将亮度大于所述平均亮度值的像素点设置为白色,其余像素设置为黑色;轮廓提取模块 605,获取由被设置为白色的像素点构成的最大的连通白色区域的边界点,并将所有边界像素点集合作为相应连通白色区域的轮廓;轮廓直线选择模块 606,基于所述轮廓上像素点到每条所变换的直线的最小距离以及基于所选择的直线的方向,从所变换的直线中确定位于所述轮廓上的直线;选择直线分组模块 607,基于所选择的直线的坐标位置关系,将所选择的直线区分为上、下、左以及右四组直线;组数判定模块 608,用于判定所分组数是否为 4 组;故障显示模块 609,用于在组数判定模块 608 判定所分组数小于 4 组时输出表示屏幕区域检测故障的信息;同组直线拟合模块 610,将属于同一组的多条直线拟合为一条直线,从而获得屏幕区域的四个边缘;以及屏幕区域构成模块 611,其基于所获得的四条直线形成屏幕区域。

[0098] 上面已经参考附图描述了根据本发明实施例的代表性屏幕区域检测方法和系统。应当注意的是,上述实施例仅是示例性的,而并非限制性的。本领域技术人员完全可以对本发明实施例进行各种修改和替换,而不背离本发明的范围。

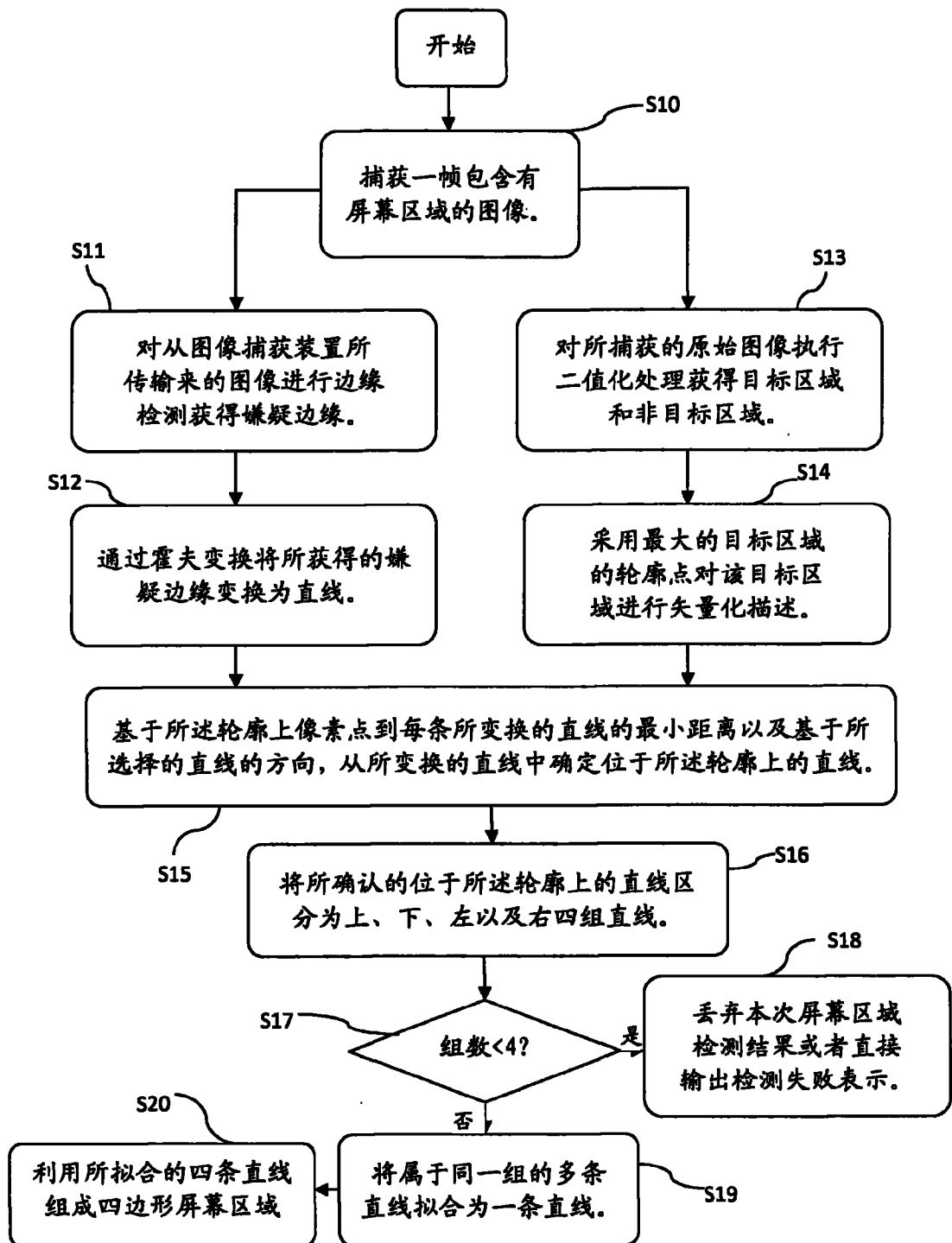


图 1

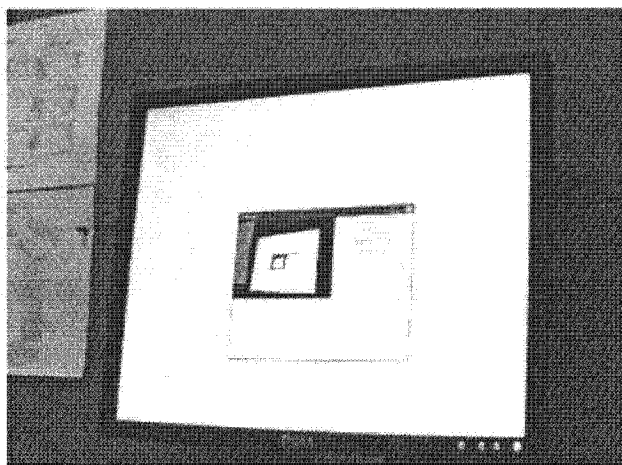


图 2

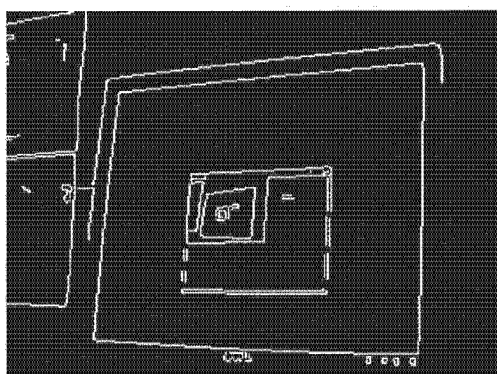


图 3A

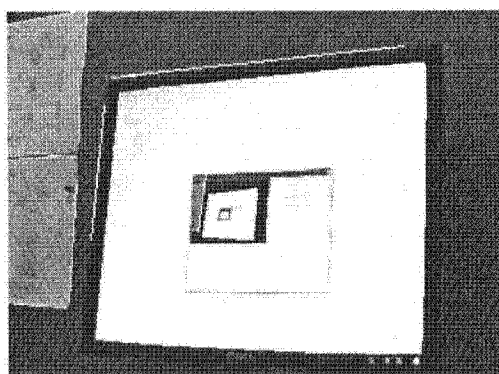


图 3B

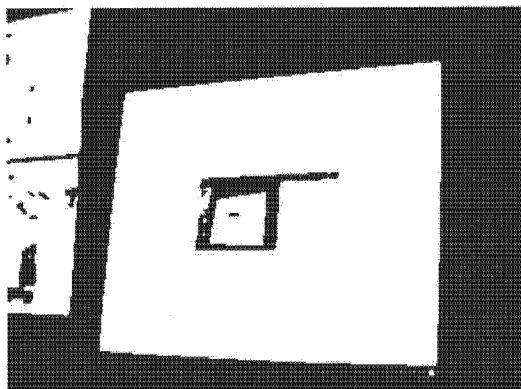


图 3C

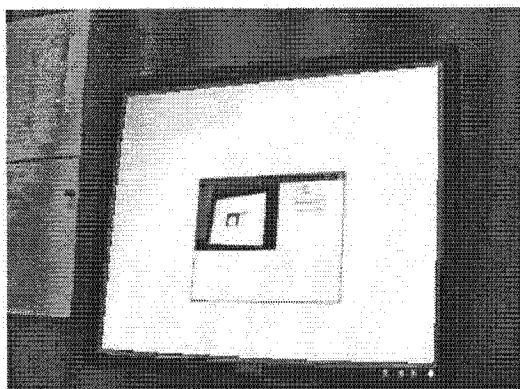


图 3D

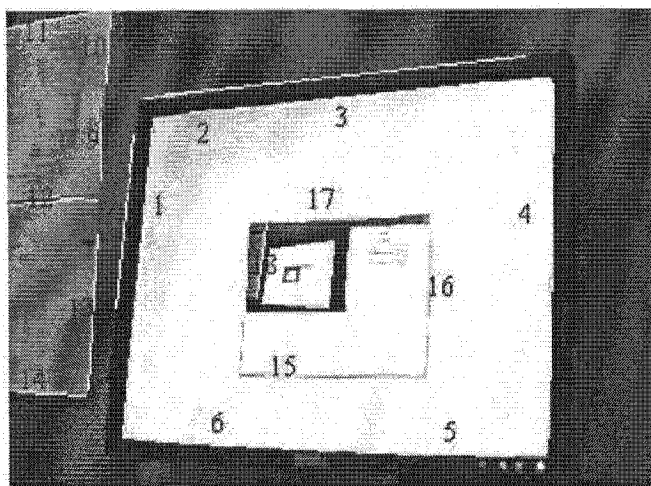


图 4

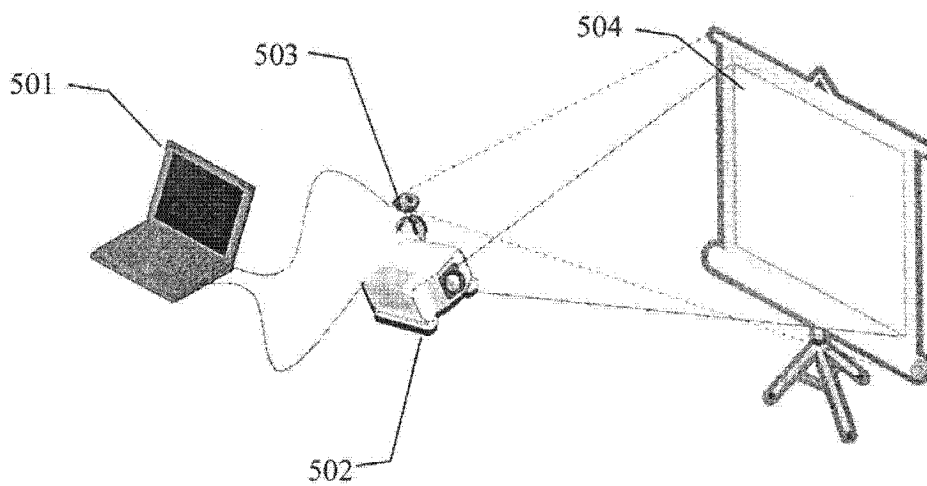


图 5

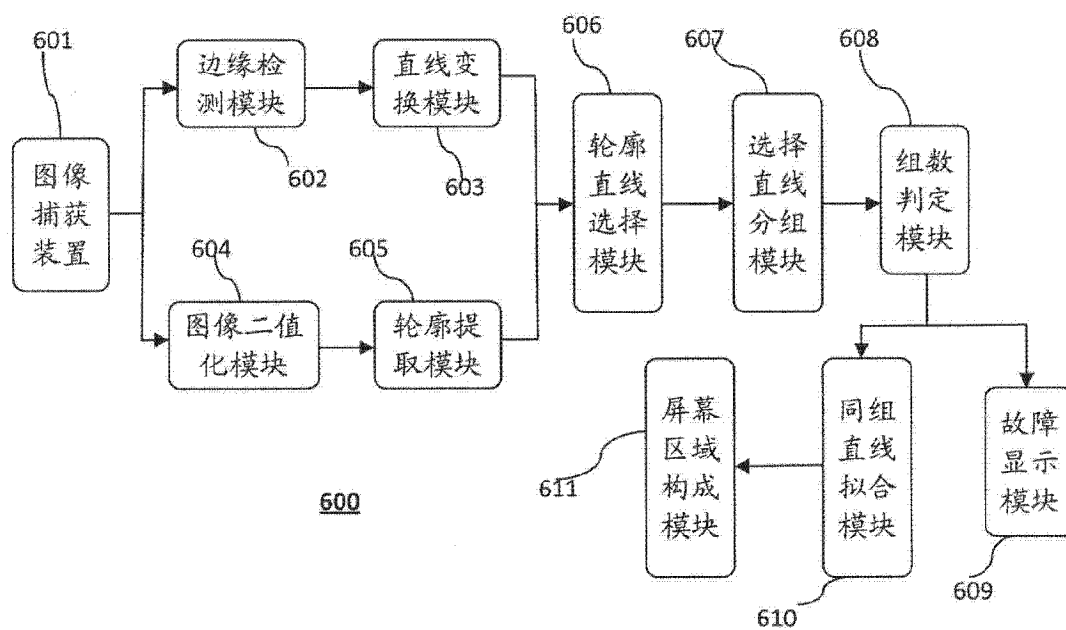


图 6