

[19] 中华人民共和国国家知识产权局



[12] 发明专利申请公布说明书

[21] 申请号 200710193525.1

[51] Int. Cl.

G06K 9/00 (2006.01)

G06K 9/46 (2006.01)

G06K 9/36 (2006.01)

[43] 公开日 2008年5月28日

[11] 公开号 CN 101187981A

[22] 申请日 2007.12.11

[21] 申请号 200710193525.1

[71] 申请人 中国科学院长春光学精密机械与物理研究所

地址 130033 吉林省长春市东南湖大路16号

[72] 发明人 邸 男

[74] 专利代理机构 长春菁华专利商标代理事务所
代理人 赵炳仁

权利要求书 2 页 说明书 8 页 附图 3 页

[54] 发明名称

对航拍图像中机场跑道的实时检测方法

[57] 摘要

本发明涉及图像末制导领域目标的实时检测方法，尤其是一种对航拍图像中机场跑道的实时检测方法。本发明采用霍夫变换与方向编码相结合的方法实现了航拍图像中机场跑道的实时检测。首先采用方向编码法对图像进行方向扫描，得到直线段的位置、长度、方向等信息，通过统计所有直线段的主要方向，确定跑道的粗略方向，然后在该方向的邻域内进行霍夫变换运算，从而大大缩小了霍夫变换的搜索范围，提高了算法的准确率和运算速度。通过大量的实验表明，当图像发生任意角度的旋转、缩放、遮挡、光照变化、模糊干扰时，该方法均可以有效克服图像的各种变化，稳定检测出直线目标。

1. 一种对航拍图像中机场跑道的实时检测方法，其特征在于，在 32bit 定点，1GHz 时钟的 TMS320C6416 定点数字信号处理器系统下按以下步骤进行：

a. 图像预处理：

首先对图像做模板大小为 3×3 的均值滤波剔除噪声干扰，然后使用索贝尔算子对图像进行边缘提取，通过设定适当的域值将图像二值化；

b. 对二值化图像进行方向编码：

采用链表结构数组存储边缘点信息，将像素P的 8个相邻像素的方向按逆时针顺序编码为0, 1, 2, 3, 4, 5, 6, 7；并将编码以1、5为对角线分成上下两个部分，上部分由2、3、4、5组成，在上半部分做搜索时的搜索方向取为+1，下半部分由6、7、0、1组成，搜索方向取为-1，当在下半部分搜索过程中未找到不为零并且未被标记的像素点，则将搜索方向改为1，继续进行上半部分搜索，否则将该点记入链表，继续在其邻域内保持原有的搜索方向搜索下一个节点；同样如果在上半部分搜索未找到不为零并且未被标记的像素点，则将搜索方向改为-1，继续进行下半部分搜索，如果8个像素点均不满足条件则链表结束；

c. 确定跑道的粗略方向 α ：

对于步骤b中得到的链表，首先剔除长度小于域值的链表，在剩余的链表中，统计每条链表中的出现次数最多的编码，则为该链表的主方向，统计所有链表的主方向中出现次数最多的方向作为跑道的粗略方向 α ，其中编码0、4对应 0° 方向；1、5对应 45° 方向；2、6对应 90° 方向；3、7对应 135° 方向；

d. 在粗略方向 α 邻域内使用霍夫变换检测平行线：

按照霍夫变换参数方程 $\rho = x \cos \theta + y \sin \theta$ 建立 $H(\rho, \theta)$ 的数组，在 $[\alpha - 30^\circ, \alpha + 30^\circ]$ 区间内进行直线的精确定位，先每隔 10° 计算 $H(\rho, \theta)$ 值，选择

最大的 $H(\rho, \theta)$ 值对应的方向，然后在该方向的 $\pm 10^\circ$ 区间内每隔 1° 计算 $H(\rho, \theta)$ 值精确定位直线，在定位一条直线之后，将该直线对应的 ρ 的 ± 9 邻域内， θ 的 ± 2 邻域内的 $H(\rho, \theta)$ 值置为 0，然后选择最大的 $H(\rho, \theta)$ 值精确定位第二条直线。

对航拍图像中机场跑道的实时检测方法

技术领域

本发明涉及图像末制导领域目标的实时检测方法，尤其是一种对航拍图像中机场跑道的实时检测方法。

背景技术

航拍图像中的目标检测是国内外普遍关注的研究热点问题。机场作为重要的运输场地，是一类特殊的目标，对机场跑道的检测无论从民用还是军用角度都具有重要的意义。观察来自互联网的高空航拍机场图像可以看到，机场航拍图像的背景比较复杂，把机场从背景中分离出来具有一定的难度，并且实际应用需要满足实时性要求。但从图像中亦可以看到机场跑道在机场的航拍图像中具有较明显的特征：机场跑道的长宽具有一定的范围，跑道本身的灰度较为平稳且跑道呈长方形状。采用传统的相关匹配算法，匹配效率很低，原因如下：

1. 由于跑道可能被一些交通工具如飞机、卡车等物遮挡，当使用主副跑道连接处等特征进行相关匹配时，匹配效率很低。
2. 由于跑道内部具有灰度和纹理的同一性，容易造成误匹配，产生爬竿现象。
3. 当跑道部分地区已经被破坏时，使用相关匹配方法难以准确定位跑道位置。
4. 当图像发生旋转、缩放变化时，相关匹配方法失效。

因此，针对机场跑道的特殊性质，常用基于直线检测的方法识别跑道目标。目前，直线检测算法主要有两大类：①是通过对图像的处理得到目标的

边界点集合，然后利用霍夫变换(Hough Transform，以下简称HT)提取目标边界上的直线；②是在图像预处理后，直接获取目标的边界线集合，然后在该集合中进行直线段识别。

方法①是最经典的直线检测算法，1962年由Paul Hough设计提出，不久被IBM申请了专利。该方法的优点是对图像中的噪声不敏感，个别非边界像素点不影响直线检测结果。缺点是计算量比较大，不利于实时应用。方法②一般是先对目标边界进行方向编码，然后在得到的链码串集合中进行直线段提取。该方法的优点是计算量小，并且能同时得到直线段的位置、长度、方向等信息。不足之处是算法性能受目标边界的跟踪算法制约，对噪声比较敏感。

发明内容

本发明的目的是为了解决目前航拍机场跑道传统检测处理方法存在的缺陷，提出一种对航拍图像中机场跑道的实时检测方法，以提高对机场跑道位置的识别精度和速度，实现方法的工程应用。

本发明对航拍图像中机场跑道的实时检测方法，是在32bit定点，1GHz时钟的TMS320C6416定点数字信号处理器系统下按以下步骤进行：

a. 图像预处理：

首先对图像做模板大小为 3×3 的均值滤波剔除噪声干扰，然后使用索贝尔算子对图像进行边缘提取，通过设定适当的域值将图像二值化。

b. 对二值化图像进行方向编码：

在边缘跟踪时，为了便于记录边缘点的信息(如图像坐标、搜索标记等等)，我们采用链表的方式来存储边缘点信息。由于数字信号处理器(DSPs)中存储空间非常小，如果以动态分配内存方式生成链表，无法控制越界问题。本发明为读入DSPs的图像建立链表结构数组，在读入图像后就固定了需要的内存空间大小，从而有效控制了越界问题。链表结构为

```
struct node
{
    unsigned short x;
    unsigned short y;
    unsigned char flag;
    unsigned char gray;
    struct node *next;
}
```

其中，x、y为当前点的坐标值；flag = 0, 1, 2, ...7分别代表对应的编码方向；flag = 8表示当点未被标记过，flag = 9表示当前点已经标记；gray代表当前点的灰度值；next是指向node类型结构的指针变量。

本发明采用著名的Freeman链码表示法，它不是逐点记录像素坐标，而是记录相邻像素之间连接方向的方法。这种编码能够压缩数据并且为后续处理提供方便。图1显示了像素P邻域的编码方向，其中像素P到相邻像素的方向按逆时针顺序编码为0, 1, 2, ...7。

为了不必每次都搜索邻域的8个像素点，将编码以1、5为对角线分成上下两个部分。上部分由2, 3, 4, 5组成，在上半部分做搜索时的搜索方向取为+1；下半部分由6, 7, 0, 1组成，搜索方向取为-1，当在下半部分搜索过程中未找到不为零并且未被标记的像素点，则将搜索方向改为1，继续进行上半部分搜索，否则将该点记入链表，继续在其邻域内保持原有的搜索方向搜索下一个节点；同样如果在上半部分搜索未找到不为零并且未被标记的像素点，则将搜索方向改为-1，继续进行下半部分搜索，如果8个像素点均不满足条件则链表结束。

c. 确定跑道的粗略方向 α

对于步骤b中得到的链表，首先剔除长度小于10的链表，在剩余的链表中，

统计每条链表中的出现次数最多的编码，则为该链表的主方向。统计所有链表的主方向中出现次数最多的方向作为跑道的粗略方向 α 。其中编码0, 4对应 0° 方向；1, 5对应 45° 方向；2, 6对应 90° 方向；3, 7对应 135° 方向。

d. 在粗略方向 α 邻域内使用霍夫变换（HT）检测平行线：

HT提取直线的原理是在二维坐标平面中，所有经过点(x,y)的直线可以由 ρ 和 θ 参数来描述， ρ 表示坐标原点到直线的距离， θ 表示该直线的垂线与X轴的夹角，它的参数方程为 $\rho = x \cos \theta + y \sin \theta$ 。

图像空间的一条直线上的点，对应着一个参数空间 (ρ, θ) 累加器 $H(\rho, \theta)$ ， $H(\rho, \theta)$ 事实上就是某方向上目标点的数目。传统的HT方法对每个像素点均做从0到 179° 的坐标空间变换，计算量非常大。本发明仅在 $[\alpha - 30^\circ, \alpha + 30^\circ]$ 区间内进行直线的精确定位。同时采用分层思想，先每隔 10° 计算 $H(\rho, \theta)$ 值，选择最大的 $H(\rho, \theta)$ 值对应的方向，然后在该方向的 $\pm 10^\circ$ 区间内每隔 1° 计算 $H(\rho, \theta)$ 值精确定位直线，大大降低了直接检测的计算量。在定位一条直线之后，将该直线对应的 ρ 的 ± 9 邻域内， θ 的 ± 2 邻域内的 $H(\rho, \theta)$ 值置为0，选择最大的 $H(\rho, \theta)$ 值精确定位第二条直线。

本发明是采用了HT与方向编码相结合的方法实现机场跑道直线检测。由于HT计算量较大，我们首先采用方向编码法对图像进行方向扫描，得到直线段的位置、长度、方向等信息，通过统计所有直线段的主要方向，确定跑道的粗略方向，然后在该方向的邻域内进行HT运算，从而大大缩小了HT的搜索范围，提高了算法的准确率和运算速度。

由于HT对图像中的噪声不敏感，个别非边界像素点不影响直线检测结果，即使在跑道存在遮挡或多处受损的情况下，仍能准确地检测出跑道位置。在检测到跑道的位置与方向后，通过仅计算垂直于跑道方向的脱靶量，剔除了由于跑道内部具有灰度和纹理的同一性，容易造成误匹配，产生爬竿现象的问题。当图像发生任意角度的旋转、缩放、遮挡、光照变化、模糊干扰时，

图像中的主要直线信息不会改变，因此基于直线检测的跑道识别跟踪算法可以有效克服图像的各种变化，稳定检测出目标，解决了当图像发生旋转、缩放变化时，相关匹配方法失效的问题。

附图说明

图1为本方法像素P邻域的编码方向示意图；

图2为所给出的实施例跑道检测的实验结果图；

其中(a)为原始灰度图像，(b)为方向编码后的二值图像，(c)为HT精确定位的平行直线，(d)为检测算法确定跑道位置（+表示）；

图3为跑道上存在遮挡的实验结果图；

其中(a)为加遮挡的灰度图像，(b)为方向编码后的二值图像，(c)为HT精确定位平行直线，(d)为检测算法确定跑道位置（+表示）；

图4为模糊图像的实验结果图；

其中(a)为模糊干扰图像；(b)为3×3均值滤波后的图像；(c)方向编码后的二值图像；(d) HT精确定位平行直线，+表示攻击目标点。

具体实施方式

以下结合给出的实施例对本发明方法作进一步详细阐述。

对来自互联网的航拍跑道图像进行跑道的实时检测，确定跑道的位置坐标。

采用的硬件环境：32bit定点，1GHz时钟的TMS320C6416定点数字信号处理器。

软件配置：Detect_Runway.c

对航拍图像中机场跑道的实时检测方法，按如下步骤进行：

- a. 调入实时图像，存储图像灰度值，大小范围为0~255。
- b. 对图像进行预处理。首先对图像做模板大小为3×3的均值滤波，然后使用索贝尔算子对图像进行边缘提取，通过设定适当的域值将图像二值化。

c. 对二值化图像进行方向编码，具体作法如下：

1) 建立二值图像的链表结构数组，

```
struct node
{
    unsigned short x;
    unsigned short y;
    unsigned char flag;
    unsigned char gray;
    struct node *next;
}
```

其中， x 、 y 为当前点的坐标值； $flag = 0, 1, 2, \dots, 7$ 分别代表对应的编码方向； $flag = 8$ 表示当点未被标记过， $flag = 9$ 表示当前点已经标记； $gray$ 代表当前点的灰度值； $next$ 是指向 $node$ 类型结构的指针变量。设置 $flag$ 初值为8， $next$ 初值为NULL。

2) 对图像从上至下，从左至右扫描，如果发现灰度值不为0，并且 $flag = 8$ 的像素点A，则建立一个新链表。头指针指向A， $flag$ 置为9，置A为当前点C，置搜索方向 $s = -1$ 。

3) 寻找下一个节点B，计数器 n 初值为0。如果 $s = -1$ ，跳至4；否则跳至5。

4) 按照逆时针的方向搜索C的邻域内编码为6, 7, 0, 1的像素点。每搜索一个点B，计数器 n 值加1。如果 $flag$ 不为8，则继续搜索其余像素点；否则如果B的灰度值为0，则 $flag$ 置为9，继续搜索其余的点，如果B的灰度值不为0，则 $flag$ 置为当前对应的编码，C的 $next$ 指针指向B，将B置为当前点C，跳至3。如果 $n = 8$ ，则C的 $next$ 指针指向NULL，链表结束，跳至2；否则改变搜索方向 $s = 1$ ，跳至5。

5) 按照逆时针方向搜索C的邻域内编码为2, 3, 4, 5的像素点。每搜索

一个点B，计数器n值加1。如果flag不为8，则继续搜索其余像素点；否则如果B的灰度值为0，则flag置为9，继续搜索其余的点，如果B的灰度值不为0，则flag置为当前对应的编码，C的next指针指向B，将B置为当前点C，跳至3。如果n = 8，则C的next指针指向NULL，链表结束；否则改变搜索方向s = -1，跳至4。

d. 确定跑道的粗略方向。首先剔除长度小于10的链表，在剩余的链表中，统计每条链表中的出现次数最多的编码，则为该链表的主方向。统计所有链表的主方向中出现次数最多的方向作为跑道的粗略方向 α 。

e. 在 α 的邻域内使用HT检测平行线。建立 $H(\rho, \theta)$ 的数组，每个元素初值为0。 θ 在 $[\alpha - 30^\circ, \alpha + 30^\circ]$ 区间内每隔 10° 变化，计算每个不为0的像素点的 ρ 值，同时 $H(\rho, \theta)$ 值加1。选择最大的 $H(\rho, \theta)$ 值对应的 θ ，然后 θ 的 $\pm 10^\circ$ 区间内每隔 1° 计算每个不为0的像素点的 ρ 值，同时 $H(\rho, \theta)$ 值加1，最后选择最大的 $H(\rho, \theta)$ 值精确定位直线。在定位一条直线之后，将该直线对应的 ρ 的 ± 9 邻域内， θ 的 ± 2 邻域内的 $H(\rho, \theta)$ 值置为0，选择最大的 $H(\rho, \theta)$ 值精确定位第二条直线。

为了验证该算法的有效性和实时性，本发明进行了大量实验，下面给出实验结果。

图2显示了跑道检测的实验结果。(a)为原始灰度图像；(b)为方向编码后的二值图像，得出图像的直线粗略方向为 $\theta = 45^\circ$ ，可以看出方向编码后的图像突出了其中直线的信息；(c)为HT精确定位的平行直线；(d)为检测算法确定跑道位置（+表示）。

图3显示了跑道上存在遮挡的实验结果。(a)为加遮挡的灰度图像；(b)为方向编码后的二值图像，可以看出遮挡并不会影响方向编码后跑道的主要方向；(c)为HT精确定位平行直线；(d)为检测算法确定跑道位置（+表示）。

图4显示了模糊图像的实验结果。(a)为模糊干扰图像；(b)为 3×3 均值滤波后的图像；(c)方向编码后的二值图像；(d) HT精确定位平行直线，+表示攻击

目标点，可见HT准确检测出直线，并没有受到模糊干扰的影响。

在时间性能方面，实验得出在方向编码阶段，本发明使用的方法运算时间不超过5ms，比传统的8邻域全部搜索算法节约了近1倍时间；在HT阶段，本发明的分层思想，大大降低了HT的运算时间，保证精确确定直线的时间不超过10ms。因此总的算法实现时间不超过 $5+10=15\text{ms}$ ，对于每秒25帧的图像采样速率，能够满足运算时间小于20ms的工程实际要求。

从上述实验结果可以看出，当图像发生任意角度的旋转、缩放、遮挡、光照变化、模糊干扰时，图像中的主要直线信息不会改变，因此基于直线检测的跑道识别跟踪算法可以有效克服图像的各种变化，稳定检测出目标。目前算法已经应用于工程硬件平台，可以满足实时处理要求。

3	2	1
4	P	0
5	6	7

图 1

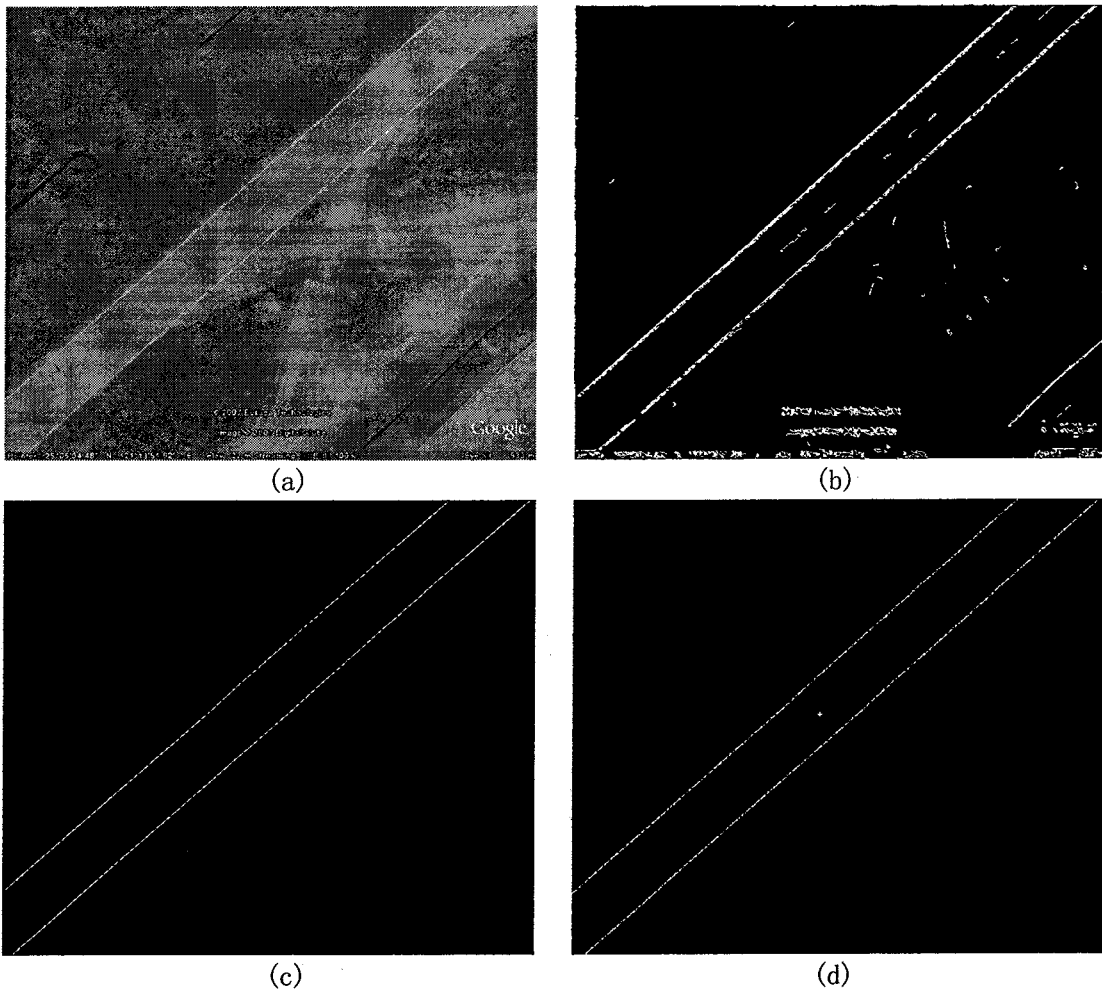


图 2

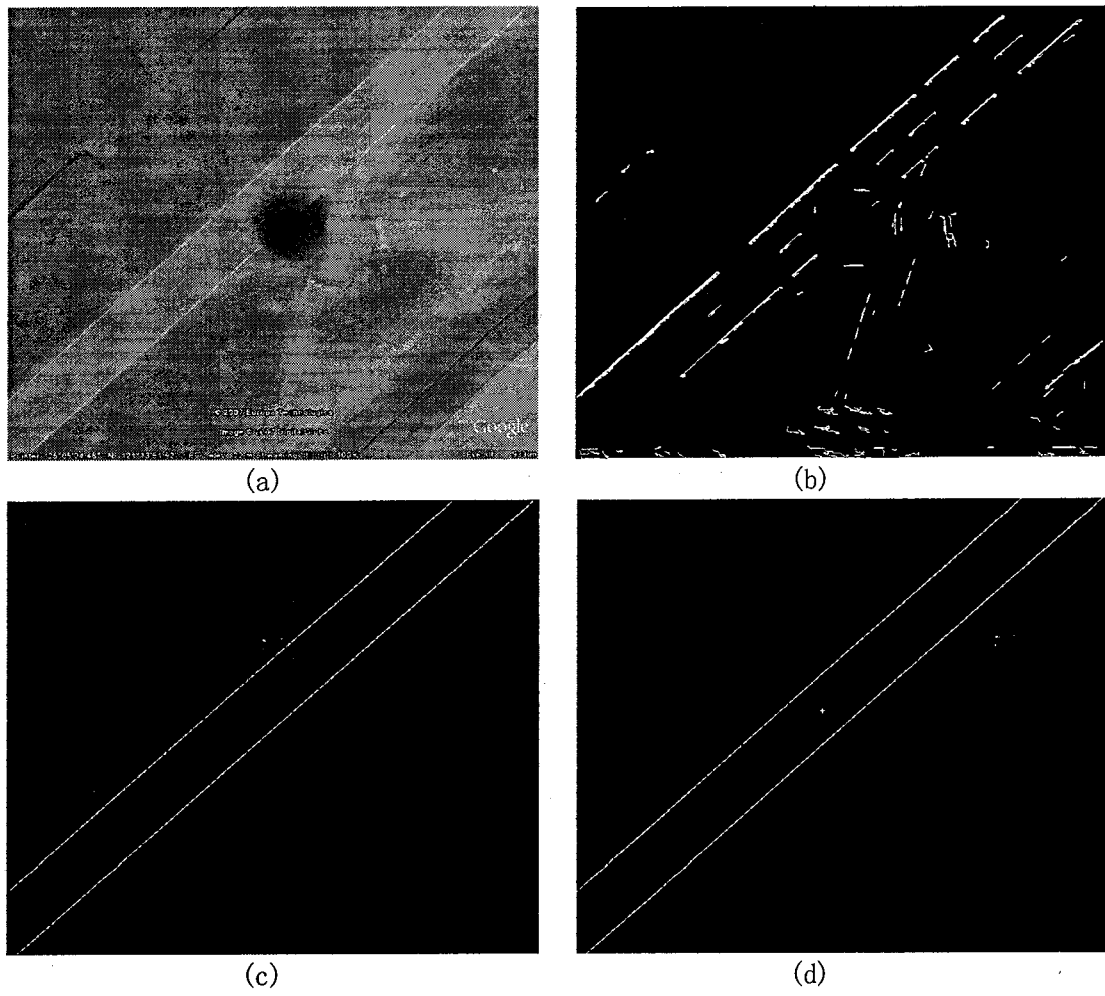


图 3

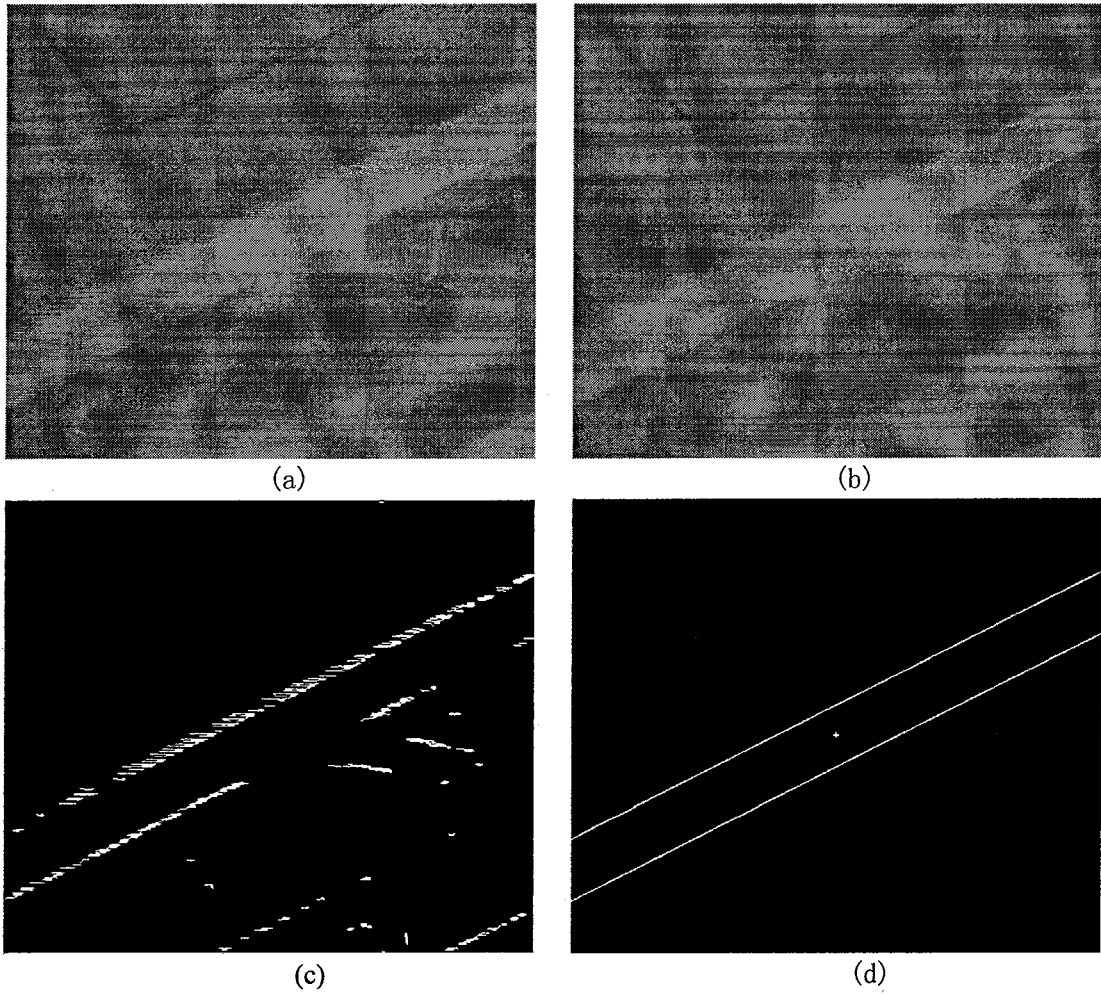


图 4