文章编号:1005－0523(2005)02－0095－05

侵入图像侦测的程序设计技术研究

詹学峰,蒋先刚,胡晓燕

(华东交通大学 信息与控制工程研究所,江西 南昌 330013)

摘要:对侵入图像侦测技术及其程序设计方法进行了探讨,主要讲述了对图像侵入进行判断的几种设计方法并对它们的运算效率做出比较,并用 Delphi 7 对相应侦测方式进行了相关的程序验证.

关 键 词:图像相减;侵入图像;RGB 颜色空间
中图分类号:TP391　　**文献标识码**:A

1 概 述

在录像监控系统中,我们需要对视频图像的变化进行自动侦测.一般的监控系统都是通过肉眼来观察监视器,辨别监视器中的图像变化情况.这就不可避免地加大了监控人员的工作量,影响了工作效率.而且对于图像变化很少的地方和夜间的一些特殊时空需要对监视器里的情况进行自动分析判断,使监控人员能够做出及时的应对,避免产生不必要的后果.我们可以把录像中的一幅正常图像作为监控的原始标准图像,监控过程中如果有侵入图像发生的话,那么这幅作为标准的图像就必然地在局部发生一些变化(比如画面中多了一个人).如何高效准确地判断侵入图像关系到实时监视系统的功效,本文主要探讨了侵入图像的各种侦测技术和运行效率.

本系统软件运行环境为 Windows 2000,软件开发采用 Delphi 7 编程工具.系统主要硬件为奔腾 4 主频为 2.40 GHz,256 兆内存的 DELL 品牌机,基于 PCI 总线的图像输入卡 FlyVideo EZ.

2 侵入图像侦测的程序实现

根据实时捕捉的图像与原始图像的变化,计算变化部分在整个图像中的占有率而确定其侦测结果.当系统实时计算出的图像变化率大于预定值的时候,系统发出报警声,提示有侵入现象.图像侦测的方式为侦测图像面中需要比较的像素点的分布格式.这些方式实施的要求是在满足客观分布地反映图像变化趋势的前提下力求最小的运算时间.下面我们将详细说明四种侵入图像的侦测方式和程序设计技术.

2.1 图像相减侦测方式

要判断一幅图片中是否有变化,可以通过图像的相减运算来进行.图像的代数运算是指对两幅图像进行点对点的加、减、乘、除四则运算而得到输出图像的运算.其图像减法运算为:$C(x,y)=A(x,y)-B(x,y)$,其中 $A(x,y)$和 $B(x,y)$为输入图像,而 $C(x,y)$为输出图像.

图像相减方法可用于去除一幅图像中不需要的加性图案,加性图案可能是缓慢变化的背景阴影、周期性的噪声或是在图像上每一个像素处均已

收稿日期:2004－06－10
基金项目:江西省教育厅 2004 年科研基金资助项目.
作者简介:詹学峰(1981－),男,江西九江人,华东交通大学在读研究生,主要研究领域:工业监控、数字图像处理.

知的附加污染等.因此,图像减法也可以用于检测同一场景的两幅图像之间的变化.

对于彩色图像,它的显示必须服从三原色 RGB 概念.自然界中所有的颜色都是由红绿蓝(R、G、B)3 种原色组成的.而在 YUV 颜色空间,Y 分量的物理含义就是亮度,它包含了灰度图的所有信息,所以只用 Y 分量就能够表示出一幅灰度图来.YUV 和 RGB 之间有着如下的对应关系:

$$Y=0.299R+0.587G+0.114B$$

将 R、G、B 值转换成 Y,就能表示出灰度图来.因此,图像的相减效果可以根据两幅图像(原始图像和实时图像)的灰度值相减得到,动态图像的假性变化可由一阀值而除去.

图像灰度处理的根本在数字图像处理中属于图像的点运算,可以利用 Delphi 7 的 Pixels 方法,该方法可以在画布上直接访问某一点的像素值,也可以使用每行像素值内存直接访问的 ScanLine 方法.为区分两种方法,我们称前者的模式为点像素处理模式,而后者的模式为像素内存操作模式.通过 ScanLine 方法,可以获取某一行所有像素点的颜色值(即各点像素的 R、G、B 分量值),返回的是一个指向位图像素数据的指针.

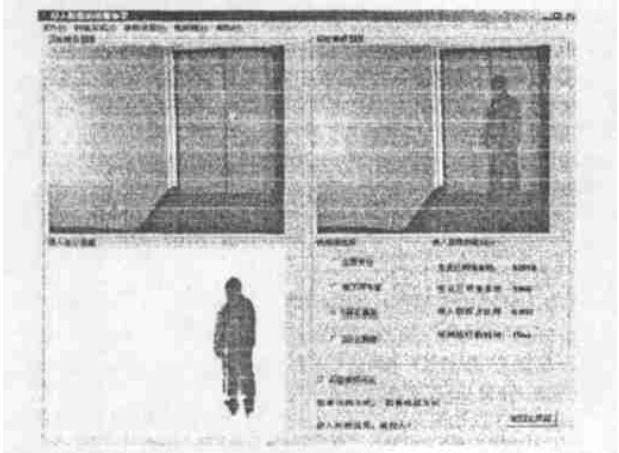

图 1 侵入图像侦测统计子窗口

图像相减侦测方式的部分程序如下:

```
Procedure TImageDetectionForm.ImageSubtractionClick(Sender: TObject);
Var P1,P2,P3: pByteArray; //定义获取每行像素点颜色值的变量
SourceShow,TargetShow,ResultShow: Byte; //定义灰度值变量
B1,B2,G1,G2,R1,R2: Byte; //定义 R、G、B 分量变量
Begin
...
SourceBmp.PixelFormat:=pf24bit; //设定图像显示格式
For Y:=0 To SourceBmp.Height-1 Do Begin
P1:=SourceBmp.ScanLine[Y]; //获取原始图像单行所有像素点颜色值
P2:=TargetBmp.ScanLine[Y]; //获取实时图像单行所有像素点颜色值
For X:=0 To SourceBmp.Width-1 Do Begin
B1:=P1[X*3];G1:=P1[X*3+1];R1:=P1[X*3+2]; //获取原始图像像素点的各分量值
B2:=P2[X*3];G2:=P2[X*3+1];R2:=P2[X*3+2]; //获取实时图像像素点的各分量值
SourceShow:=Round(0.299*R1+0.587*G1+0.114*B1); //计算原始图像像素点灰度值
TargetShow:=Round(0.299*R2+0.587*G2+0.114*B2); //计算实时图像像素点灰度值
ResultShow:=Abs(SourceShow-TargetShow); //灰度值相减,实现图像相减运算
End; End; ... End;
```

从图 1 可以看到,两幅图像基本可以"相减"出来,但是由于光线会自然变化,实时监控图像不可能像静态图像一样完完整整地相减,所以出现了某些本来没有变化的区域中有一些随机分布的小点.在此,我们根据实际情况,预先设定了一个阀值.灰度差低于阀值的,认为是光线等环境的自然变化;反之,则认为是图像中实际侵入现象造成的变化.这样,可以得到"相减结果"所在的区域在整个监控画面中所占的比例,就可以判断出原始图像中是否有侵入.

在程序中,我们使用的 ScanLine 方法是每次扫描一行.如果要处理的是一幅很大的图像(比如 640*480 和 800*600)的话,可以选择性地使用每 2 行或者相隔更多行扫描一次,这样就可以减少程序的运行时间,节省系统资源.在表 1 中可以看出:不论是每行还是每 5 行扫描一次,图像变化率的变化仅在 0.1%;且随着扫描间隔增大,程序运行的时间也相应减小.所以,对于处理多路大尺寸图像,程序还是能在保证数据准确性的前提下实现实时监控.

由于可以得到比较直观的"相减"后侵入图像,因此用这种方式得到的结果应该是非常准确的.所以,在监控要求比较高的场合下,可以采用这种方式对监控的结果加以确认核对,保证监控结果正确无误.

2.2　直接比较像素分量侦测方式

通过图像相减方式，我们不但可以得到所需要的数据(所占比例)，而且还可以得到相减的图像.但在很多情况下，我们监控的目的不是为了看到发生侵入现象前后相减的图像，而是判断侵入图像所占背景图像的像素比进而确定其报警判断结果.因此，在有异常情况发生时，我们关心的是此侵入现象是不是要引起我们的关注.所以，只要计算出发生异常区域所占比例就可以了，我们只以此为依据进行判断.

在这种方式下，还是通过 ScanLine 方法来获得各点的像素的 R、G、B 分量值.根据实际情况，设定一个区分光线自然变化和异常事件的阀值，通过比较不同画面每行像素的 R、G、B 分量，得出大概的污染区域，再直接计算图像变化率.

直接比较像素分量侦测方式的例程如下：

```
Procedure TImageDetectionForm.CompareRowClick(Sender: TObject);
Type
pRGBTripleArray = ^TRGBTripleArray;
TRGBTripleArray = Array[WORD] of TRGBTriple; //定义行扫描数据类型
Var RowA,RowB: pRGBTripleArray; //定义扫描行的类型
Begin
...
For Y:=0 To SourceBmp.Height-1 Do Begin
RowA:=SourceBmp.ScanLine[Y]; RowB:=TargetBmp.ScanLine[Y]; //扫描两幅图像的某行
For X:=0 to SourceBmp.Width-1 do Begin
JudgeRed:=Abs(RowA[X].rgbtRed-RowB[X].rgbtRed); //比较每行像素点红色分量信息
JudgeGreen:=Abs(RowA[X].rgbtGreen-RowB[X].rgbtGreen);//比较每行像素点绿色分量信息
JudgeBlue:=Abs(RowA[X].rgbtBlue-RowB[X].rgbtBlue); //比较每行像素点蓝色分量信息
... End; End; End;
```

在表 1 中可以看出，虽然是每行都进行比较，但即使是在处理较大尺寸图像的时候，该方式始终能保持很快的速度(≤16ms)，因此是一种非常快捷的方法.但是在相同情况下，这种方式的误差较之图像相减方式要大.从表 1 中可以看出，该方式与图像相减方式的结果相差在 5% 之内，为此我们可以通过调整阀值来获得最佳效果.在一些监视要求并不需要太高的场合(如在白天)可以采用这种侦测方式.

采用 BMP 格式的 RGB 三原色空间图像虽然可调用低层次的处理函数进行快速图像处理，但三原色空间中空间两点间的欧式距离与颜色距离不成线性比例，这样的图像空间不适宜于快速图像比较，合理的处理方法是采用 CIE 的均匀色度空间，将 RGB 模式的彩色图像转化为 HIS 模式的彩色图像，而 HIS 空间的图像的彩色值只与 H 有关，图像比较的判断处理完成后，再将处理后的各像素的 HIS 返回到 RGB 空间，这样将 RGB 空间图像要素处理的快捷性和 HIS 空间图像颜色判断的节度性有机地结合起来.

2.3　交叉扫描侦测方式

所谓交叉扫描，就是图像中各探测行与各探测列相互交错，把整个图像交叉划分，在整个图像监控区域形成一张栅格形监控网.因此，侦察扫描的部分只是有一定间隔的行和列的像素段参与比较，这就比前面两种侦测方式所计算的数据要少.如果图像中某几个地方有变化，就相当于这张“网”的某些敏感线有触动，也就是实时图像与原始图像相比较在那几段的像素值发生了灰度变化.根据发生了变化的像素段的数目，就可以得出我们所需要的图像变化判断结果.

这种方式下，从表 1 中可以看出，随着行与行(列与列)之间的间隔逐渐增大，变化率也在增大.这是由于间隔的增大，会忽略一些发生变化的像素点.交叉扫描方式虽然扫描的像素点减少，但网络分布的均衡性反映了图像实际的变化.从表 1 可以看出，每隔 4 行扫描的结果与每行都扫描的结果相差只有 2%，因此侦测结果满足技术要求.因此，选定合适的扫描间隔和阀值，可以获得足够的准确度，又可以比较快速地处理数据.

2.4　马赛克方格侦测方式

相对于前面的 3 种方法，该方式是一种在概率统计概念上来说比较平均的估算方法.其思路是，先把整个侦察区域分成多块交错小方格，每个小方格代表一块探测区域.当以小方格为扫描的最小单位时，在探测奇数方格行的时候，每个方格行探测奇数列的小方格；在探测偶数方格行的时候，每行探测偶数列的小方格.这就像国际象棋的棋盘一样，以类似马赛克格子的方式来侦测整个监控区域.而每个像素块的灰度在概率统计上来讲，代表了这一小块正方形区域内灰度的相对平均值.通过

设置合适的阀值,就消除了一些因光线自然变化问题而产生的不规则灰度变化的像素点对监控产生的影响.

从表 1 中可以看出,在相同阀值下,由于光线自然变化而发生改变的像素点分布的均衡性,该侦测方式测得的图像变化率随着扫描间隔的增大逐渐减小.当每 5 行扫描一次(即设置成 5 * 5 的像素块方格)时,图像变化率和图像相减侦测方式的结果达到一致.因此,当阀值和扫描间隔取值合适的话,就能得出非常实际的结果.

马赛克方格侦测方式的部分源码如下:

```
Procedure TImageDetectionForm.MosaicDetectionClick(Sender: TObject);
Var pSource,pTarget: Array of pByteArray; //定义获取行像素信息的变量
SourceGraySum,TargetGraySum: Array of Integer; //定义灰度统计计算的动态数组
Begin {方格大小为 Step * Step}
For Y:=0 To SourceBmp.Height-1 Do Begin
If (Y Mod (2 * Step))=0 Then Begin //判断每次扫描的起点,每 2 * Step 行才扫描一次
For I:=Y To Y+2 * Step-1 Do Begin //每次循环扫描 2 * Step 行像素信息
If I=SourceBmp.Height Then Begin
Temp:=((I div (2 * Step))+1) * (2 * Step); //设定临边界的循环标志点
Break; End;
pSource[I]:=SourceBmp.ScanLine[I]; pTarget[I]:=TargetBmp.ScanLine[I];
Temp:=I+1; End; End; //设定一般位置的循环标志点
...
For J:=0 To SourceBmp.Width-1 Do Begin
If (Y<=Temp-Step) And (Y>=Temp-(2 * Step)) Then Begin //以方格为单位,计算灰度统计值
If (J Mod (2 * Step))=0 Then Begin //判断方格起点
For X:=J To J+Step Do Begin //计算 step 个像素(一格)的灰度值和
If X=SourceBmp.Width Then Break; //计算像素点到达边界,跳出循环
B1:=pSource[Y][X * 3]; G1:=pSource[Y][X * 3+1]; R1:=pSource[Y][X * 3+2]; //获取颜色分量
B2:=pTarget[Y][X * 3]; G2:=pTarget[Y][X * 3+1]; R2:=pTarget[Y][X * 3+2]; //获取颜色分量
SourceScanArray[X,Y]:=Round(0.299 * R1+0.587 * G1+0.114 * B1); //原始图像灰度数组
TargetScanArray[X,Y]:=Round(0.299 * R2+0.587 * G2+0.114 * B2); //实时图像灰度数组
//以下两个数组分别计算原始图像和实时图像奇数行灰度值和
SourceGraySum[OddCount]:=SourceGraySum[OddCount]+SourceScanArray[X,Y];
TargetGraySum[OddCount]:=TargetGraySum[OddCount]+TargetScanArray[X,Y]; End;
OddCount:=OddCount+2; End; End; //设定扫描方格索引号
If (Y>Temp-Step) And (Y<Temp) Then Begin //以方格为单位计算偶数行灰度统计值
...(过程与奇数行灰度计算例程基本相同) //计算偶数行灰度值和
End; End; End;
```

表 1 两种模式下 4 种侦测方式图像侦测效率比较表

扫描间隔	图像侦测方式	点像素处理模式	像素内存操作模式			
		图像相减方式	图像相减方式	直接比较方式	交叉扫描方式	马赛克方格方式
1	运行时间	3 156 ms	31 ms	≤16 ms	78 ms	47 ms
	图像变化率	0.158	0.158	0.190	0.155	0.132
2	运行时间	1 594 ms	16 ms	≤16 ms	47 ms	31 ms
	图像变化率	0.158	0.158	0.190	0.158	0.169
3	运行时间	1 063 ms	16 ms	≤16 ms	31 ms	31 ms
	图像变化率	0.158	0.158	0.190	0.166	0.162
4	运行时间	797 ms	16 ms	≤16 ms	31 ms	31 ms
	图像变化率	0.159	0.159	0.193	0.173	0.159
5	运行时间	625 ms	≤16 ms	≤16 ms	31 ms	31 ms
	图像变化率	0.158	0.158	0.189	0.176	0.158

注：消除图像伪变化阀值为 20，图像大小为 640 * 480

从表 1 中可以看出，点像素处理模式的速度比较慢，而采用像素内存操作模式的几种方式速度比之快 50～90 倍. 图像相减侦测方式和直接比较方式在不同扫描间隔下的结果最为一致；交叉扫描侦测方式由于其算法会忽略一些像素变化使扫描结果随着扫描间隔的增大而稍微增大，马赛克方格侦测方式则因其块状区域灰度值计算的平均性会使扫描结果随着扫描间隔增大而逐渐趋于平缓.

为了获得最佳的侦察效果，在不同环境里应当选取适当的阀值，而阀值的取值又取决于光线的明暗程度. 在光线比较暗的环境中，图像灰度值较之正常情况下减小，阀值可以取得相对小些（比如取 15 左右）；在光线比较强的环境中，图像灰度值也相应变大，阀值可以取得相对大些（比如取 25 左右）. 也可根据经验对一天中的不同时段进行分析总结，得出大致的阀值取值变化曲线.

3　结束语

通过实际测试表明，根据不同的监控场合和监控需求，我们可以选择不同的图像侦测方式. 图像侦测之图像相减方式比较准确直观；直接比较方式算法快速；交叉扫描方式效率和环境适应性比较适中；马赛克方格扫描方式侦测结果反映图像变化的区域均衡性. 因此，根据实际情况调整阀值和扫描间隔，四种侦测方式可分别用于不同的场合和时间段，也可以根据摄像环境、侦测的时效性而选择最佳方式以实现高可靠性和实时性的视频监控.

参考文献：

[1] 刘　骏. Delphi 数字图像处理及高级应用[M]. 北京：科学出版社，2003.

[2] 蒋先刚. 医学显微图像管理系统研究[J]. 计算机应用，2001.6：37－39.

[3] 赵雪春. 基于彩色分割的车牌自动识别技术[J]. 上海交通大学学报，1998.10：4－9.

[4] 何　斌，等. Visual C＋＋数字图像处理[M]. 北京：人民邮电出版社，2001.

Research on Program Technology of Invaded Image's Detection

ZHAN Xue-feng，JIANG Xian-gang，HU Xiao-yan

(Institute of Information and Control Engineering，East China Jiaotong University，Nanchang 330013，China)

Abstract：This pager discusses the technologies of invaded image's detection and program designing，mainly introduces some kinds of methods of judging invaded image and has a comparison with their efficiency，and gives some source codes of them by Delphi 7.

Key words：image subtraction；invaded image；RGB colour space