



(12) 发明专利申请

(10) 申请公布号 CN 104751487 A

(43) 申请公布日 2015. 07. 01

(21) 申请号 201510134601. 6

(22) 申请日 2015. 03. 26

(71) 申请人 太原理工大学

地址 030024 山西省太原市万柏林区迎泽西大街 79 号

(72) 发明人 耿蒲龙 刘旭飞 宋渊 刘媛  
雷志鹏 宋建成 田慕琴

(74) 专利代理机构 太原科卫专利事务所(普通合伙) 14100

代理人 戎文华

(51) Int. Cl.

G06T 7/20(2006. 01)

权利要求书1页 说明书4页

(54) 发明名称

一种基于彩色 RGB 三平面色变帧差的运动目标检测方法

(57) 摘要

一种基于彩色 RGB 三平面色变帧差的运动目标检测方法,所述方法是获取待测场景下的相邻两帧彩色图像,提取彩色图像中红色、绿色和蓝色平面并存储;根据提取的六个平面,使用帧差法将红色、绿色和蓝色平面分别进行差运算,生成三幅差值图像;再针对三幅差值图像的像素点依次计算;最后对获得的新差值图像中每个像素点做二值化处理,从而获得运动目标的轮廓。本方法通过对 RGB 三个平面分别进行帧差计算,判断同一坐标像素点色差变化是否同向,对色差变化进行叠加计算,使得运动目标轮廓像素值与其他区域像素值形成较大反差,从而实现了对运动目标的准确提取,最大限度地保留了运动目标轮廓的完整信息。

1. 一种基于彩色 RGB 三平面色变帧差的运动目标检测方法, 它由以下步骤实现:

(1) 根据待测场景下的视频内容获得第  $k-1$  帧和第  $k$  帧场景彩色图像,  $k$  为大于 1 的整数;

(2) 对步骤(1)获取的第  $k-1$  帧彩色图像, 提取该彩色图像红色平面  $f_{k-1}(x, y, r)$ 、绿色平面  $f_{k-1}(x, y, g)$  和蓝色平面  $f_{k-1}(x, y, b)$ , 存储于记忆体内,  $x, y$  表示像素点坐标,  $r, g, b$  分别表示彩色图像的红色平面、绿色平面和蓝色平面, 下标  $k-1$  表示视频帧数;

(3) 对步骤(1)获取的第  $k$  帧彩色图像, 提取该彩色图像红色平面  $f_k(x, y, r)$ 、绿色平面  $f_k(x, y, g)$  和蓝色平面  $f_k(x, y, b)$ , 存储于记忆体内, 下标  $k$  表示视频帧数;

(4) 根据步骤(2)和步骤(3)中提取的六个平面, 使用帧差法将第  $k$  帧和第  $k-1$  帧图像的红色平面、绿色平面和蓝色平面分别进行差运算, 生成对应于 RGB 三平面的三幅差值图像, 分别为  $D_k(x, y, r)$ 、 $D_k(x, y, g)$ 、 $D_k(x, y, b)$ ; 运算表达式为:

$$\begin{cases} D_k(x, y, r) = f_k(x, y, r) - f_{k-1}(x, y, r) \\ D_k(x, y, g) = f_k(x, y, g) - f_{k-1}(x, y, g) \\ D_k(x, y, b) = f_k(x, y, b) - f_{k-1}(x, y, b) \end{cases}$$

(5) 对满足  $D_k(x, y, r)$ 、 $D_k(x, y, g)$ 、 $D_k(x, y, b)$  同时大于 0 或者同时小于 0 的像素点计算  $D_k(x, y)$ ;  $C_k(x, y)$  的运算表达式为:

$$C_k(x, y) = \text{Min}\{|D_k(x, y, r)|, |D_k(x, y, g)|, |D_k(x, y, b)|\}$$

式中,  $||$  表示取绝对值,  $\text{Min}\{\}$  表示  $\{\}$  括号内各项值中的最小值;

(6) 根据是否满足  $D_k(x, y, r)$ 、 $D_k(x, y, g)$ 、 $D_k(x, y, b)$  同时大于 0 或者同时小于 0 的条件, 对各像素点计算相应的  $D_k(x, y)$ ; 对于符合条件的像素点,  $D_k(x, y)$  的运算表达式为:

$$D_k(x, y) = |D_k(x, y, r)| + |D_k(x, y, g)| + |D_k(x, y, b)| - \delta \cdot C_k(x, y)$$

式中,  $\delta$  为  $(0, 3]$  之间的实数; 对于不符合判断条件的像素点,  $D_k(x, y)$  的运算表达式为:

$$D_k(x, y) = |D_k(x, y, r)| + |D_k(x, y, g)| + |D_k(x, y, b)|$$

(7) 对步骤(6)中  $D_k(x, y)$  值计算得到新的差值图像  $E_k(x, y)$ ;  $E_k(x, y)$  的运算表达式为:

$$E_k(x, y) = \begin{cases} 255 & D_k(x, y) \geq 255 \\ \text{floor}\{D_k(x, y)\} & D_k(x, y) < 255 \end{cases}$$

式中,  $\text{floor}[\ ]$  函数表示对中括号  $[\ ]$  内的数值进行向下取整运算;

(8) 对步骤(7)获得的新差值图像  $E_k(x, y)$  中每个像素点做二值化处理, 从而获得运动目标的轮廓, 二值化处理运算表达式为:

$$A_k(x, y) = \begin{cases} 1 & \text{Foreground } E_k(x, y) \geq T \\ 0 & \text{Background } E_k(x, y) < T \end{cases}$$

式中,  $A_k(x, y)$  为二值化后图像,  $T$  为阈值。

## 一种基于彩色 RGB三平面色变帧差的运动目标检测方法

### 技术领域

[0001] 本发明涉及一种运动目标的检测方法,尤其是一种基于彩色 RGB 三平面色变帧差的运动目标检测方法。

### 背景技术

[0002] 视频技术在科学研究和工程应用领域有着广阔的发展前景。在视频图像处理过程中,运动目标的检测和提取是一项关键技术。运动目标检测与提取的目的是为了把运动目标从背景图像中分离出来,是运动目标与背景的分割问题。

[0003] 视频图像处理的首要任务是从视频图像序列中检测出运动的目标,如何快速有效的对背景和目标进行分离是当前研究的热点和重点。目前常用的检测算法有光流法、背景差法和帧间差法。其中,光流法运算量较大,不适合实时处理;背景差法需要建模,而且对动态场景变化较为敏感,参数(比如学习率)的不恰当设置将直接影响整体效果;帧间差法是一种在时间序列上进行差分的方法,由于不需要建模,因此实时性较其他方法是最好的。例如 A. Lipton 发表的论文《Moving target classification and tracking from real time video》对灰度平面帧差进行了详细的阐述;专利文献《一种运动目标的识别方法、装置》(CN103826102A)以帧差法为基础,与背景差法相结合,通过对当前帧建模并与下一帧图像进行差运算来确定运动目标区域。但是,上述文献中所使用的帧差法只对灰度平面进行帧差计算,由于单个平面每个像素点取值范围小且易受外部环境光线亮度变化的干扰,使得运动目标图像难以与背景图像准确区分,容易造成有效信息的丢失,增加了完整提取运动目标轮廓的难度,影响了后期视频图像处理的准确性。

[0004] 视频序列中运动目标检测是视频图像处理领域中的难点,在智能视频监控应用中,计算机需要对所采集的图像进行快速和准确处理,这一过程对软件算法提出了更高要求。尤其是对监控区域小、光线较暗或者亮度变化频繁的视频监控场景,现有算法的准确性就很难满足实际要求。

### 发明内容

[0005] 针对上述现有技术存在的问题,本发明提供一种基于彩色 RGB 三平面色变帧差的运动目标检测方法。

[0006] 实现上述运动目标检测方法的技术方案如下。

[0007] 一种基于彩色 RGB 三平面色变帧差的运动目标检测方法,它由以下步骤实现:

(1)根据待测场景下的视频内容获得第  $k-1$  帧和第  $k$  帧场景彩色图像, $k$  为大于 1 的整数;

(2)对步骤(1)获取的第  $k-1$  帧彩色图像,提取该彩色图像红色平面  $f_{k-1}(x, y, r)$ 、绿色平面  $f_{k-1}(x, y, g)$  和蓝色平面  $f_{k-1}(x, y, b)$ ,存储于记忆体内, $x, y$  表示像素点坐标, $r, g, b$  分别表示彩色图像的红色平面、绿色平面和蓝色平面,下标  $k-1$  表示视频帧数;

(3)对步骤(1)获取的第  $k$  帧彩色图像,提取该彩色图像的红色平面  $f_k(x, y, r)$ 、绿色

平面  $f_k(x, y, g)$  和蓝色平面  $f_k(x, y, b)$ , 存储于记忆体内, 下标  $k$  表示视频帧数;

(4) 根据步骤(2)和步骤(3)中提取的六个平面, 使用帧差法将第  $k$  帧和第  $k-1$  帧图像  
的红色平面、绿色平面和蓝色平面分别进行差运算, 生成对应于 RGB 三平面的三幅差值图  
像, 分别为  $D_k(x, y, r)$ 、 $D_k(x, y, g)$ 、 $D_k(x, y, b)$ ; 运算表达式为:

$$\begin{cases} D_k(x, y, r) = f_k(x, y, r) - f_{k-1}(x, y, r) \\ D_k(x, y, g) = f_k(x, y, g) - f_{k-1}(x, y, g) \\ D_k(x, y, b) = f_k(x, y, b) - f_{k-1}(x, y, b) \end{cases}$$

(5) 对满足  $D_k(x, y, r)$ 、 $D_k(x, y, g)$ 、 $D_k(x, y, b)$  同时大于 0 或者同时小于 0 的像素点  
计算  $D_k(x, y)$ ;  $C_k(x, y)$  的运算表达式为:

$$C_k(x, y) = \text{Min}\{|D_k(x, y, r)|, |D_k(x, y, g)|, |D_k(x, y, b)|\}$$

式中,  $||$  表示取绝对值,  $\text{Min}\{ \}$  表示  $\{ \}$  括号内各项值中的最小值;

(6) 根据是否满足  $D_k(x, y, r)$ 、 $D_k(x, y, g)$ 、 $D_k(x, y, b)$  同时大于 0 或者同时小于 0 的  
条件, 对各像素点计算相应的  $D_k(x, y)$ ; 对于符合条件的像素点,  $D_k(x, y)$  的运算表达式  
为:

$$D_k(x, y) = |D_k(x, y, r)| + |D_k(x, y, g)| + |D_k(x, y, b)| - \delta \cdot C_k(x, y)$$

式中,  $\delta$  为 (0, 3] 之间的实数; 对于不符合判断条件的像素点,  $D_k(x, y)$  的运算表达式  
为:

$$D_k(x, y) = |D_k(x, y, r)| + |D_k(x, y, g)| + |D_k(x, y, b)|$$

(7) 对步骤(6)中  $D_k(x, y)$  值计算得到新的差值图像  $E_k(x, y)$ ;  $E_k(x, y)$  的运算表达  
式为:

$$E_k(x, y) = \begin{cases} 255 & D_k(x, y) \geq 255 \\ \text{floor}[D_k(x, y)] & D_k(x, y) < 255 \end{cases}$$

式中,  $\text{floor}[ \ ]$  函数表示对中括号  $[ \ ]$  内的数值进行向下取整运算;

(8) 对步骤(7)获得的新差值图像  $E_k(x, y)$  中每个像素点做二值化处理, 从而获得运  
动目标的轮廓, 二值化处理运算表达式为:

$$A_k(x, y) = \begin{cases} 1 & \text{Foreground } E_k(x, y) \geq T \\ 0 & \text{Background } E_k(x, y) < T \end{cases}$$

式中,  $A_k(x, y)$  为二值化后图像,  $T$  为阈值。

[0008] 实现上述本发明所提供的一种基于彩色 RGB 三平面色变帧差的运动目标检测方  
法的技术方案, 与现有技术相比, 本发明通过对 RGB 三个平面分别进行帧差计算, 以同一像  
素点色差变化的大小确定是否为前景运动图像; 同时, 通过判断同一坐标像素点色差变化  
是否同向, 消除了外部环境因亮度变化而造成的影响; 最后, 对色差变化进行叠加计算, 使  
得运动目标轮廓像素值与其他区域像素值形成较大反差, 从而对运动目标进行完整、准确  
提取, 最大限度地保留了运动目标轮廓的完整信息。

## 具体实施方式

[0009] 下面将详细阐述本发明的具体实施方式。

[0010] 实施本发明上述所提供的一种基于彩色 RGB 三平面色变帧差的运动目标检测方法的技术方案,它是由以下步骤实现的。

[0011] 步骤一、根据待测场景下的视频内容获得第  $k-1$  帧和第  $k$  帧场景彩色图像,  $k$  为大于 1 的整数;

步骤二、对步骤一获取的第  $k-1$  帧彩色图像,提取该彩色图像红色平面  $f_{k-1}(x, y, r)$ 、绿色平面  $f_{k-1}(x, y, g)$  和蓝色平面  $f_{k-1}(x, y, b)$ , 存储于记忆体内,  $x, y$  表示像素点坐标,  $r, g, b$  分别表示彩色图像的红色平面、绿色平面和蓝色平面, 下标  $k-1$  表示视频帧数;

步骤三、对步骤一获取的第  $k$  帧彩色图像,提取该彩色图像红色平面  $f_k(x, y, r)$ 、绿色平面  $f_k(x, y, g)$  和蓝色平面  $f_k(x, y, b)$ , 存储于记忆体内, 下标  $k$  表示视频帧数;

步骤四、根据步骤二和步骤三中提取的六个平面,使用帧差法将第  $k$  帧和第  $k-1$  帧彩色图像的红色平面、绿色平面和蓝色平面分别进行差运算,生成对应于 RGB 三平面的三幅差值图像,分别为  $D_k(x, y, r)$ 、 $D_k(x, y, g)$ 、 $D_k(x, y, b)$ ; 运算表达式为:

$$\begin{cases} D_k(x, y, r) = f_k(x, y, r) - f_{k-1}(x, y, r) \\ D_k(x, y, g) = f_k(x, y, g) - f_{k-1}(x, y, g) \\ D_k(x, y, b) = f_k(x, y, b) - f_{k-1}(x, y, b) \end{cases}$$

$f_k$  图像为彩色图像,  $f_k(x, y, r)$ 、 $f_k(x, y, g)$  和  $f_k(x, y, b)$  为  $f_k$  图像的红绿蓝三个平面,其中每个像素点的值均为大于零的整数;  $D_k(x, y, r)$ 、 $D_k(x, y, g)$ 、 $D_k(x, y, b)$  分别表示彩色 RGB 三平面的差值图像,差值图像每个像素点的值可能为正,也可能为负值。下面以第 2 帧与第 1 帧图像(即  $k=2$  时)中取 3 个坐标  $(0, 0)$   $(1, 1)$   $(2, 2)$  对应像素点具体数值计算进行说明。假设坐标  $(0, 0)$  对应的 R 平面  $f_1(0, 0, r)=1$ 、 $f_2(0, 0, r)=200$ , G 平面  $f_1(0, 0, g)=10$ 、 $f_2(0, 0, g)=160$ , B 平面  $f_1(0, 0, b)=8$ 、 $f_2(0, 0, b)=100$ ; 坐标  $(0, 0)$  对应的 R 平面  $D_2(0, 0, r)=f_2(0, 0, r)-f_1(0, 0, r)=200-1=199$ , G 平面  $D_2(0, 0, g)=f_2(0, 0, g)-f_1(0, 0, g)=160-10=150$ , B 平面  $D_2(0, 0, b)=f_2(0, 0, b)-f_1(0, 0, b)=100-8=92$ 。坐标  $(1, 1)$  对应的 R 平面  $f_1(1, 1, r)=5$ 、 $f_2(1, 1, r)=0$ , G 平面  $f_1(1, 1, g)=20$ 、 $f_2(1, 1, g)=1$ , B 平面  $f_1(1, 1, b)=40$ 、 $f_2(1, 1, b)=3$ ; 坐标  $(1, 1)$  对应的 R 平面  $D_2(1, 1, r)=f_2(1, 1, r)-f_1(1, 1, r)=0-5=-5$ , G 平面  $D_2(1, 1, g)=f_2(1, 1, g)-f_1(1, 1, g)=1-20=-19$ , B 平面  $D_2(1, 1, b)=f_2(1, 1, b)-f_1(1, 1, b)=3-40=-37$ 。坐标  $(2, 2)$  对应的 R 平面  $f_1(2, 2, r)=50$ 、 $f_2(2, 2, r)=0$ , G 平面  $f_1(2, 2, g)=5$ 、 $f_2(2, 2, g)=20$ , B 平面  $f_1(2, 2, b)=3$ 、 $f_2(2, 2, b)=30$ ; 坐标  $(2, 2)$  对应的 R 平面  $D_2(2, 2, r)=f_2(2, 2, r)-f_1(2, 2, r)=0-50=-50$ , G 平面  $D_2(2, 2, g)=f_2(2, 2, g)-f_1(2, 2, g)=20-5=15$ , B 平面  $D_2(2, 2, b)=f_2(2, 2, b)-f_1(2, 2, b)=30-3=27$ 。

[0012] 步骤五,对满足  $D_k(x, y, r)$ 、 $D_k(x, y, g)$ 、 $D_k(x, y, b)$  同时大于 0 或者同时小于 0 的像素点计算  $D_k(x, y)$ ;  $C_k(x, y)$  的运算表达式为:

$$C_k(x, y) = \text{Min}\{|D_k(x, y, r)|, |D_k(x, y, g)|, |D_k(x, y, b)|\}$$

式中,  $||$  表示取绝对值,  $\text{Min}\{\}$  表示  $\{\}$  括号内各项值中的最小值;

对于坐标  $(0, 0)$  点,  $D_2(0, 0, r)=199$ 、 $D_2(0, 0, g)=150$ 、 $D_2(0, 0, b)=92$  均大于 0, 符合判断条件,  $0 < 92 < 150 < 199$ , 故  $C_k(0, 0)$  取值为 92。对于坐标  $(1, 1)$  点,  $D_2(1, 1, r)$

$=-5$ 、 $D_2(1, 1, g)=-19$ 、 $D_2(1, 1, b)=-37$  均小于 0, 符合判断条件,  $-37 < -19 < -5 < 0$ , 故  $C_k(1, 1)$  取值为 5。对于坐标(2, 2)点,  $D_2(2, 2, r)=-50$ 、 $D_2(2, 2, g)=15$ 、 $D_2(2, 2, b)=27$ , 不符合判断条件。

[0013] 步骤六, 根据是否满足  $D_k(x, y, r)$ 、 $D_k(x, y, g)$ 、 $D_k(x, y, b)$  同时大于 0 或者同时小于 0 的条件, 对各像素点计算相应的  $D_k(x, y)$ ; 对于符合条件的像素点,  $D_k(x, y)$  的运算表达式为:

$$D_k(x, y) = |D_k(x, y, r)| + |D_k(x, y, g)| + |D_k(x, y, b)| - \delta \cdot C_k(x, y)$$

式中,  $\delta$  为 (0, 3] 之间的实数; 对于不符合判断条件的像素点,  $D_k(x, y)$  的运算表达式为:

$$D_k(x, y) = |D_k(x, y, r)| + |D_k(x, y, g)| + |D_k(x, y, b)|$$

对于符合步骤五中判断条件的像素点, 假设  $\delta$  取值为 2, 坐标(0, 0)点,  $D_2(0, 0) = 199 + 150 + 92 - 2 \times 92 = 461 - 184 = 257$ ; 坐标(1, 1)点,  $D_2(1, 1) = 5 + 19 + 37 - 2 \times 5 = 51$ 。对于不符合判断条件的像素点,  $D_2(2, 2) = 50 + 15 + 27 = 92$ 。

[0014] 步骤七, 对步骤六中  $D_k(x, y)$  值计算得到新的差值图像  $E_k(x, y)$ ;  $E_k(x, y)$  的运算表达式为:

$$E_k(x, y) = \begin{cases} 255 & D_k(x, y) \geq 255 \\ \text{floor}[D_k(x, y)] & D_k(x, y) < 255 \end{cases}$$

式中, floor[ ] 函数表示对中括号 [ ] 内的数值进行向下取整运算;

对于坐标(0, 0)点,  $D_2(0, 0) = 257 > 255$ , 故  $E_2(0, 0) = 255$ ; 坐标(1, 1)点,  $D_2(1, 1) = 51 < 255$ , 故  $E_2(1, 1) = 51$ ; 坐标(2, 2)点,  $D_2(2, 2) = 92 < 255$ , 故  $E_2(2, 2) = 92$ 。

[0015] 步骤八, 对步骤七获得的新差值图像  $E_k(x, y)$  中每个像素点做二值化处理, 从而获得运动目标的轮廓, 二值化处理运算表达式为:

$$A_k(x, y) = \begin{cases} 1 & \text{Foreground } E_k(x, y) \geq T \\ 0 & \text{Background } E_k(x, y) < T \end{cases}$$

式中,  $A_k(x, y)$  为二值化后图像, T 为阈值。

[0016] 假设 T 为 100, 对于坐标(0, 0)点,  $E_2(0, 0) = 255 > 100$ , 故  $A_2(0, 0) = 1$ ; 坐标(1, 1)点,  $E_2(1, 1) = 51 < 100$ , 故  $A_2(1, 1) = 0$ ; 坐标(2, 2)点,  $E_2(2, 2) = 92 < 100$ , 故  $A_2(2, 2) = 0$ 。经过前述步骤的处理, 运动目标轮廓像素值与其他区域像素值形成较大反差, 阈值 T 将很容易确定, 从而保证了对运动目标进行完整、准确提取, 最大限度地保留了运动目标轮廓的完整信息。