

# 一种基于运动矢量分析的 Mean shift 目标跟踪算法

田 纲<sup>1), 2)</sup> 胡瑞敏<sup>1), 2)</sup> 王中元<sup>2)</sup>

<sup>1)</sup> (武汉大学计算机学院, 武汉 430079) <sup>2)</sup> (国家多媒体软件工程技术研究中心(武汉大学), 武汉 430079)

**摘 要** Mean shift 算法作为一种非参密度估计算法, 目前已被广泛应用于视频运动目标的跟踪。该算法具有运算效率高, 对目标变形、旋转不敏感, 在部分遮挡的情况下有一定鲁棒性等特点, 但该算法在运动目标速度过快的情况下, 由于没有考虑利用目标的运动方向和速度信息, 因此在跟踪快速运动目标时容易造成跟踪丢失。针对此问题, 提出了一种基于运动矢量分析与 Mean shift 跟踪算法相结合的新方法, 即首先对视频编码过程中产生的运动矢量进行概率统计分析, 以获取目标运动方向与运动速度的估计值, 再以此修正 Mean shift 运动候选区域的中心位置, 使每次搜索开始时, 候选中心位置更接近实际目标中心位置。通过与传统的 Mean shift 算法的跟踪实验比较可见, 新算法不仅提高了快速运动目标跟踪的精度, 而且减少了算法的搜索迭代次数, 从而提高了运算效率。该算法可适用于智能视频监控设备中的视频编码与目标跟踪同时计算的情况, 实验结果表明, 该算法是有效可行的。

**关键词** Mean shift 目标跟踪, 运动矢量

中图法分类号: TP391.41 文献标志码: A 文章编号: 1006-8961(2010)01-0085-06

## A Mean shift Target Tracking Algorithm Based on Motion Vector Analysis

TIAN Gang<sup>1), 2)</sup>, HU Ruimin<sup>1), 2)</sup>, WANG Zhongyuan<sup>2)</sup>

<sup>1)</sup> (School of Computing, Wuhan University, Wuhan 430079)

<sup>2)</sup> (National Multimedia Software Engineering Research Center of Wuhan University, Wuhan 430079)

**Abstract** Mean shift can make rapidly optimal matching during target tracking. But Mean shift algorithm doesn't use the target's motion direction and speed information. When target's moving speed is so fast that it's easily fails to track the target accurately. We proposed a new target tracking algorithm combining motion vector analysis with Mean shift algorithm. By statistical analysis of the motion vector, we adjust the central point of the motion candidate region of Mean shift, which makes the starting position for search more close to the centre of the actual objective. New method not only improves the accuracy of fast moving tracking, but also reduces the iterative times during the center searching.

**Keywords** Mean shift, target tracking, motion vector

## 0 引 言

Mean shift 算法是一种非参数密度估计算法<sup>[1]</sup>, 由 Fukunaga 于 1975 年首先提出。其作为一种高效

的模式匹配算法, 已被成功地应用于实时性要求较高的目标跟踪系统中<sup>[2]</sup>。Dom Comaniciu 首先将 Mean shift 算法应用于图像滤波、分割与目标跟踪等领域<sup>[3-4]</sup>。Bradski 提出以颜色直方图为目标模式的 Mean shift 目标跟踪算法<sup>[5]</sup>。该算法先利用颜色直

基金项目: 国家重点基础研究发展计划(973)项目(2009CB320906)

收稿日期: 2008-06-16 改回日期: 2009-01-16

第一作者简介: 田纲(1977—), 男, 教师。武汉大学计算机学院博士研究生。并在武汉大学国家多媒体软件工程技术研究中心从事图像处理、视频编解码和智能视频分析相关研究工作。E-mail: tiangang\_comput@163.com

方图得到每帧图像的颜色投影图,然后自适应调整搜索窗口的位置和大小,并通过不断收敛将得到的最优中心位置作为目标的中心。另外,Nummiaro等人采用粒子滤波结合 Mean shift的方法进行目标跟踪,但是粒子滤波器本身的复杂计算降低了跟踪的实时性<sup>[6]</sup>。

与其他的运动目标跟踪算法相比较,Mean shift算法具有如下特点:1)计算量不大,可以满足实时跟踪要求;2)对运动目标变形、旋转、背景变化不敏感;3)对于部分遮挡情况下的目标跟踪具有一定的鲁棒性;4)Mean shift算法作为一个无参估计算法,很容易作为一个模块与其他算法集成。Mean shift算法也存在以下一些不足:1)缺乏必要的模板更新算法,由于跟踪过程中窗宽的大小保持不变,因此当目标有尺度变化时,可能跟踪失败;2)直方图是一种对目标特征进行描述较弱的方法,当背景和目标的分布较相似时,算法效果欠佳;3)当场景中目标的运动速度很快时,或者当视频图像帧率较低,帧间时间间隔较长时,目标区域在相邻两帧间会出现没有重叠区域的情况,此时目标往往收敛于背景中与目标颜色分布比较相似的物体,而不是场景中的目标。

本文针对 Mean shift算法在进行快速运动目标跟踪时所表现出的不足,提出了一种改进的算法,目前一般采用的解决方法是结合卡尔曼滤波或粒子滤波来对运动目标空间运动位置进行预测,然后结合基于颜色直方图的 Mean shift算法,同时采用这两种方法进行跟踪,并采用不同的比例因子将两个跟踪结果进行线性加权,即可得到目标的最终位置<sup>[7-8]</sup>。此类算法的思路是考虑到因目标运动速度过快而导致运动目标超过 Mean shift的收敛范围,如果先通过预测来初步定位运动目标在下一帧的位置,并将其作为 Mean shift搜索的中心位置的参考,然后再在中心点位置进行更精细的 Mean shift搜索,以进一步精确定位运动目标。但由于这类方法需要对图像进行复杂的滤波预测计算,因而使跟踪效率降低。考虑到目前视频分析技术在数字信号处理(DSP)的应用中多需要在同一块芯片中同时实现视频编码、目标跟踪等算法,据此,即可以利用视频编码在运动估计时得到的宏块运动矢量信息来进行概率统计计算目标的运动方向和速度估计值,并预测运动目标在下一帧图像中的空间位置,同时以此来修正 Mean shift算法搜索的中心位置。

## 1 Mean shift算法概述

Mean shift算法是一种非参数概率密度估计算法,可以通过迭代快速的收敛于概率密度函数的局部极大值。算法的跟踪过程就是不断寻找概率密度局部最大值的过程,其步骤如下:

### 1.1 目标模型的描述

关于目标模型的描述,首先是目标的初始化,即在首帧图像中确定被跟踪的目标区域,其可以通过手工选择的方式来确定目标区域,也可以根据运动检测的结果自动选取目标区域。如果目标区域的中心为  $\mathbf{x}_0$ ,那么目标模型可以描述为在目标区域上的所有特征值的概率值,该特征值一般指 HSV 颜色特征或其他特征。用目标模型的特征值  $u = 1, \dots, m$  估计的概率密度为

$$\hat{q}_u = C \sum_{i=1}^n K \left\| \frac{\mathbf{x}_0 - \mathbf{x}_i}{h} \right\|^2 \delta[b(\mathbf{x}_i) - u] \quad (1)$$

其中,  $K(\mathbf{x})$  为核函数的轮廓函数,由于目标模型中心附近的像素比外围像素更可靠,因此  $K(\mathbf{x})$  对于中心像素给一个大的权值,而对于远离中心的像素则给一个小的权值。 $b(\mathbf{x}_i)$  表示像素  $\mathbf{x}_i$  属于哪种特征值,  $\delta(\mathbf{x})$  是 Delta 函数,  $\delta[b(\mathbf{x}_i) - u]$  的作用是用于判断目标区域中任一像素  $\mathbf{x}_i$  的颜色值是否等于第  $u$  个特征值,若等于,则颜色值为 1,否则为 0。 $C$  是一个标准化的常量系数,用于使得  $\sum_{u=1}^m q_u = 1$ 。

### 1.2 候选模型的描述

运动目标在第 2 帧及以后的每一帧中,可能包含目标的区域称为候选区域,其中心坐标为  $\mathbf{y}$ ,候选模型的像素特征值  $u = 1, \dots, m$  的概率密度为

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} K \left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \delta[b(\mathbf{x}_i) - u] \quad (2)$$

其中,  $h$  为带宽参数, Mean shift 的跟踪窗口大小依赖于带宽  $h$ ,  $C_h$  为归一化常数,该常数为  $C_h = \left[ \sum_{i=1}^{n_h} K \left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \right]^{-1}$ 。

### 1.3 相似性函数

相似性函数用于描述目标模型和候选目标之间的相似程度,可使用 Bhattacharyya 系数作为相似性函数,即

$$\rho(\mathbf{y}) \equiv \rho(\hat{p}(\mathbf{y}), \hat{q}) = \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}) \hat{q}_u} \quad (3)$$

其值在 0~ 1 之间。 $\hat{\rho}(\mathbf{y})$  的值越大, 表示两个模型越相似。

### 1.4 目标定位

为使  $\hat{\rho}(\mathbf{y})$  最大, 应当先将当前帧的目标中心定位为前一帧中目标中心的位置  $\mathbf{y}_0$ , 然后从这一点开始寻找最优匹配的目标, 其中心为  $\mathbf{y}_1$ 。定位时, 先在  $\hat{\rho}(\mathbf{y}_0)$  处对函数 (式 (3)) 进行泰勒展开, 相似性函数可近似为

$$\rho(\hat{p}(\mathbf{y}), \hat{q}) = \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y}_0) \hat{q}_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k \left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \quad (4)$$

其中,

$$w_i = \sum_{u=1}^m \frac{\sqrt{\hat{q}_u}}{N_{\hat{p}_u(\mathbf{y}_0)}} \delta[b(\mathbf{x}_i) - u] \quad (5)$$

令

$$f_{n,k} = \sum_{i=1}^{n_h} \frac{C_h}{2} w_i k \left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2 \quad (6)$$

这就是在有权值  $w_i$  时的核密度估计, 可以看出, 求相似性函数最大就是要求式 (6) 最大, 通过对相似性函数求最大值, 即可计算 Mean shift 向量  $\mathbf{y}_1 - \mathbf{y}_0$  (如式 (7) 所示)。在每次的 Mean shift 迭代中, 若  $\|\mathbf{y}_1 - \mathbf{y}_0\| < \epsilon$  则停止迭代, 此时目标区域的中心位置都会由  $\mathbf{y}_0$  移动到新的位置  $\mathbf{y}_1$ 。

$$\mathbf{y}_1 - \mathbf{y}_0 = \frac{\sum_{i=1}^{n_h} \mathbf{x}_i w_i g \left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2}{\sum_{i=1}^{n_h} w_i g \left\| \frac{\mathbf{y} - \mathbf{x}_i}{h} \right\|^2} - \mathbf{y}_0 \quad (7)$$

其中,  $g(\mathbf{x}) = -k'(\mathbf{x})$  这样目标区域即可由初始位置逐步移动到真实的目标位置。

根据式 (4) 可知, 相似性函数在  $\hat{\rho}(\mathbf{y}_0)$  处先进行泰勒展开, 泰勒展开要求是在邻域内展开, 这就限制了起始点  $\mathbf{y}_0$  和  $\mathbf{y}$  的距离不能太大, 如果运动速度过快, 则 Mean shift 算法的跟踪效果不好。

## 2 运动矢量的统计特征

由于视频监控图像序列中一般不会出现场景突变的情况, 因此运动矢量具有连贯特性, 采用对运动矢量进行统计分析的方法可计算出运动目标的运动方向、速度等重要参数。

对于视频监控图像来说, 一般造成目标运动的原因主要有摄像头的运动、现场物体的运动, 以及摄像头和现场物体同时运动几种。本文将“摄像头的运动”分成左移运动、右移运动、上移运动和右移运动 (图 1) 4 种。将“现场物体的运动”分成 8 个方向上的运动、运动目标的远离和靠近 (图 2) 10 种。

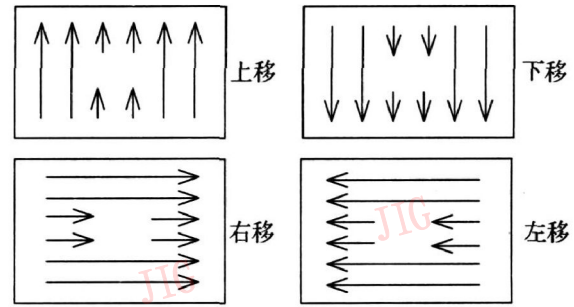


图 1 摄像头的运动矢量

Fig. 1 Motion vector of camera

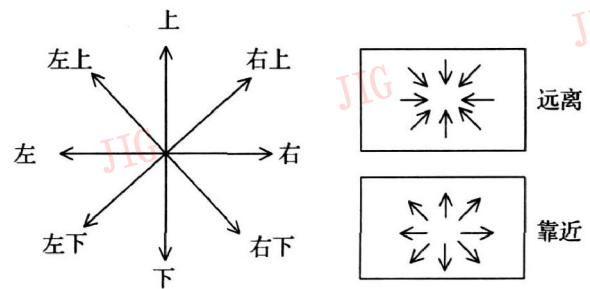


图 2 运动目标运动矢量

Fig. 2 Motion vector of object

在摄像头运动、现场物体运动、摄像头和现场物体同时运动 3 种不同情况下, 通过对图像运动矢量进行统计分析可知, 当摄像头静止时, 背景区域的运动矢量几乎全为 0, 运动目标区域才会有运动矢量产生; 当摄像头运动时, 由于整幅图像都属于背景, 因此所有运动矢量几乎一致。对于摄像头和物体一起运动的情况, 背景区域运动矢量大部分是一致的, 但运动目标区域的运动矢量与背景区的运动矢量在方向上不一致, 运动目标区域存在自己局部的统计特性。

### 2.1 运动矢量预处理

一般视频编码主要考虑的首先是码流的压缩效率而非估计的准确性, 由于其得到的运动矢量并不能完全代表真实的运动情况, 因此运动目标区域宏块的所有运动矢量并不一定朝向同一个方向; 其次, 外界环境的变化、摄像头移动以及其他噪声的影响,

都会在视频编码过程中产生相应的运动矢量, 这些运动矢量不利于运动目标的检测, 需要进行运动矢量的预处理。对于小于一定阈值的运动矢量, 其为噪声的可能性很大, 可以用以下方法滤除。

$I_{x,y} = |V_x^{\text{motion}}| + |V_y^{\text{motion}}|$ , 其中  $V_x^{\text{motion}}$  是运动矢量在  $x$  轴方向上的分量,  $V_y^{\text{motion}}$  是运动矢量  $V^{\text{motion}}$  在  $y$  轴方向上的分量。取滤波阈值为  $T$ , 如果  $I_{x,y} < T$ , 则  $I_{x,y} = 0$ 。

另外, 可采用向量中值滤波器<sup>[9]</sup>来去除随机噪声。

## 2.2 运动矢量方向编码

运动矢量的方向编码如图 3 所示, 运动矢量方向角的范围为  $0 \sim 2\pi$ , 将其划分成 8 个区间 ( $0 \sim 7$  区间), 其量化系数为  $\pi/4$ 。

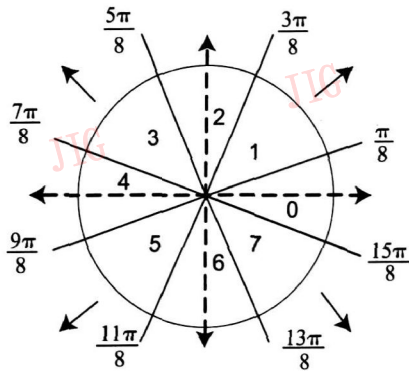


图 3 运动矢量方向编码示意图

Fig 3 Director coding of motion vector

运动矢量的垂直分量与水平分量的夹角即为运动矢量的方向角  $\theta$  可求得  $\text{tg}\theta = V_y^{\text{motion}} / V_x^{\text{motion}}$ , 方向角  $\theta$  属于图中的哪个区间, 即可生成这个区间的方向编码。图像中第  $i$  行,  $j$  列宏块的运动矢量的方向编码为

$$C_{ij} = \begin{cases} \left\lfloor \arctg \left| \frac{V_y^{\text{motion}}}{V_x^{\text{motion}}} \right| + \frac{\pi}{8} \right\rfloor / \Delta\theta & (\theta > \frac{\pi}{8} \parallel \frac{\pi}{8} < \frac{16\pi}{8}) \\ 0 & \text{其他} \end{cases} \quad (8)$$

$C_{ij} \in (0, 7)$ ,  $\Delta\theta = \pi/4$  为量化系数。

## 2.3 估计目标的运动方向与速度

对于静止摄像头来说, 背景区域宏块的运动矢量基本上都为 0 仅运动目标区域存在运动矢量, 对运动目标的运动方向和运动速度进行估计的步骤如下:

1) 求运动目标区域中所有运动矢量的方向编码概率密度分布  $\hat{P}_i$ ,  $i \in (0, 7)$ , 其为运动矢量方向编码在 8 个方向上的概率, 因此可用于建立运动矢量方向编码直方图。

2) 根据运动矢量的方向概率密度分布, 即可得到运动矢量方向编码最大概率密度  $P_{\max} = \max(\hat{P}_i)$ , 如果  $P_{\max}$  超过一定阈值, 即表示在运动区域内运动方向指向同一个方向的运动矢量的个数超过一定的比例, 则表示运动目标具有整体位移的特性, 此时可以对其运动方向和运动速度进行估计, 否则认为该目标区域的运动方向和速度全为 0。

3) 取  $\hat{P}_i = P_{\max}$  的所有运动矢量  $\bar{V}_i^{\text{motion}}$  ( $1 \leq i \leq N$ )

4) 定义平均运动矢量  $\bar{V}^{\text{motion}}$ , 其在  $x$  方向和  $y$  方向分量的均值为

$$\bar{V}_x^{\text{motion}} = \frac{1}{N} \sum_{i=1}^N V_{ix}^{\text{motion}}$$

$$\bar{V}_y^{\text{motion}} = \frac{1}{N} \sum_{i=1}^N V_{iy}^{\text{motion}}$$

运动速度的估计值为

$$\hat{v} = \sqrt{(\bar{V}_x^{\text{motion}})^2 + (\bar{V}_y^{\text{motion}})^2} \quad (9)$$

运动方向的估计值为

$$\hat{\theta} = \arctg \left| \frac{\bar{V}_y^{\text{motion}}}{\bar{V}_x^{\text{motion}}} \right| \quad (10)$$

对于摄像机运动的情况, 整幅图像的大多数的运动矢量会朝向同一个方向, 当整帧图像的同方向的运动矢量概率密度大于一定的阈值, 则认为有摄像头运动。此时, 仍只需考虑当前运动目标区域的运动矢量方向编码概率密度的统计特性, 即可根据运动矢量方向编码直方图来获取运动目标的运动方向与速度。

## 3 算法流程

本文算法先根据运动矢量估计得到的目标运动方向与速度, 用以修正候选区域的中心坐标; 然后结合 Mean shift 搜索来得到搜索结果。本文算法步骤如下:

1) 确定第 1 帧图像的中心点  $y_0$ 。

2)从第 2 帧开始,根据运动搜索得到的目标区域的运动矢量分析来得到跟踪目标的预测运动方向  $\hat{\theta}$  和速度  $\hat{v}$ ,根据中心点  $y_0$  修正的当前候选区域的中心点为

$$\tilde{y}_0 = \alpha y_0 + (1 - \alpha) \bar{V}^{\text{motion}} \quad (11)$$

其中,  $\alpha (0 \leq \alpha \leq 1)$  为中心点修正因子,可表示为中心点与运动速度的相关性大小,若运动速度越快,则平均运动矢量的对中心点的修正作用越大,即当速度  $\hat{v}$  越大时,则  $\alpha$  越小;反之,如果运动速度越慢,则平均运动矢量对中心点的修正作用越小,即当速度  $\hat{v}$  越小时,则  $\alpha$  越大;如果运动速度为 0 则不需要修正,可直接采用  $y_0$  作为中心点。

3)根据修正的中心点  $\tilde{y}_0$  进行 Mean shift 搜索,每次收敛即可得到运动目标的中心点  $y_1$ ,这样不断地迭代收敛,若  $\|y_1 - y_0\| < \epsilon$  则停止迭代,同时将目标区域的中心位置由  $y_0$  移动到新的位置  $y_1$ ,并将  $y_1$  作为下一帧的中心位置  $y_0$ ,继续步骤 2),先得到新的  $\tilde{y}_0$ ,然后再进行下一帧的 Mean shift 搜索。

## 4 实验结果及分析

本实验采用基于 HSV 颜色特征概率密度的 Mean shift 算法进行运动跟踪,并采用 MPEG-4 编码运动估计算法获取图像宏块的运动矢量,原始图像大小为  $320 \times 240$  帧率 30 fps(帧/s),通过大量的序列图像对本算法进行了测试,在快速运动目标场景下跟踪的准确度有明显提高。

图 4 显示了在智能视频监控中采用传统的 Mean shift 算法因出现运动目标速度太快而丢失跟踪的情况,而对算法进行修正后,则可以很快得到运动目标,且算法跟踪快速运动目标的准确度得到明显提高。图 4(a)~图 4(c)对运动目标在编码过程中的运动矢量进行了显示,根据运动矢量的统计结果基本上可预测运动目标的下一帧的中心位置。图 4(d)~图 4(f)是传统的 Mean shift 算法的跟踪丢失结果。图 4(g)~图 4(i)是本文算法的跟踪结果。利用间隔抽帧可以模拟运动目标的速度加快,原始帧率为 30 fps 通

过抽帧帧率变成 10 fps 等于是在每 30 帧中去掉 20 帧,仅对剩余的 10 帧进行处理。此时,传统的 Mean shift 算法已经不能跟踪到场景中的车辆,而本文的组合算法在视频帧率低至 5 fps 时仍然可以有效跟踪运动目标。

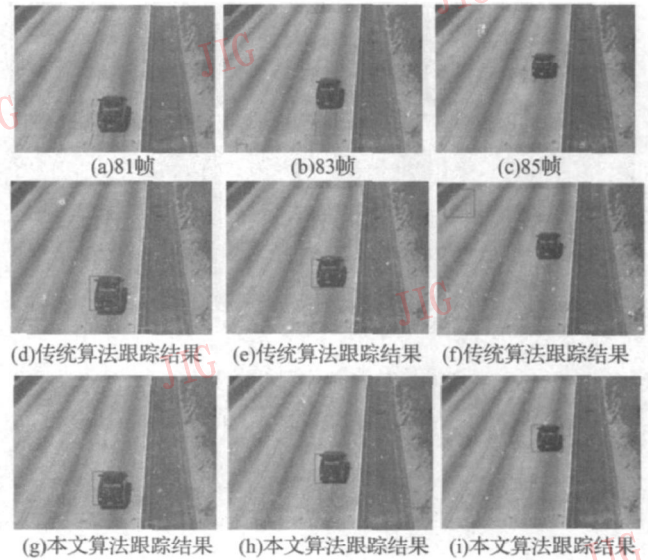


图 4 算法的跟踪效果图

Fig 4 Tracking result of new algorithm

另外,算法跟踪效率提高的情况如图 5 所示,相比传统的 Mean shift 算法,基于运动矢量分析的 Mean shift 算法,其迭代次数明显减少,一般最多迭代 3 次就可以找到运动目标所在的中心位置,从而提高了 Mean shift 算法的运算效率。

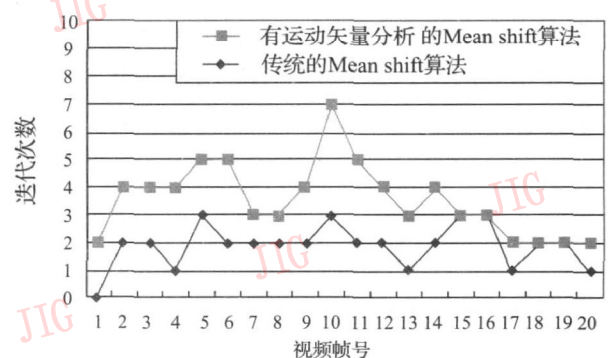


图 5 算法的迭代次数

Fig 5 The Mean shift iteration times

图 6 给出了新算法进行室内奔跑人体跟踪的效果与原始 Mean shift 算法跟踪效果的比较。图 6(d)~图 6(f)是传统 Mean shift 算法的跟踪效果,图 6(g)~图 6(i)是本文改进算法的跟踪效果。

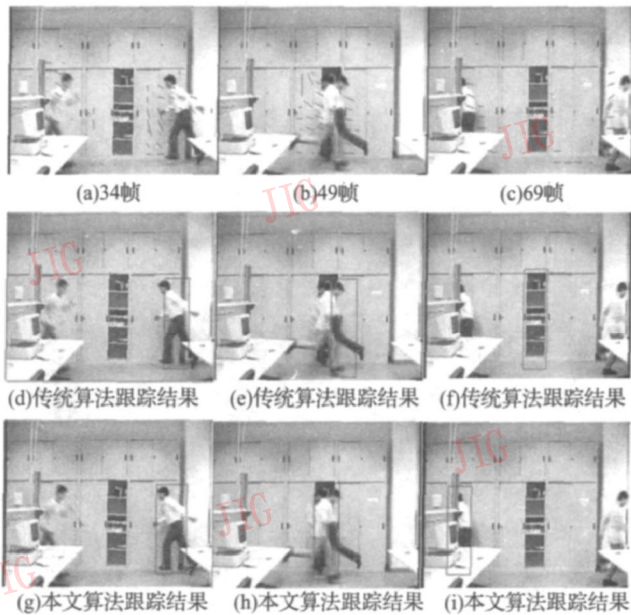


图 6 室内人体跟踪效果图

Fig 6 Human running tracking result of new algorithm

## 5 结 论

本文提出了一种基于运动矢量分析的 Mean shift 算法, 用于进行运动目标的跟踪, 由于 Mean shift 算法在进行运动目标跟踪的时候没有考虑目标运动的宏观信息, 即没有利用运动目标在图像中的运动方向与运动速度等有利因素, 而目前已有的卡尔曼滤波或粒子滤波对运动目标预测的方法又存在计算量大的问题, 在实际的智能视频监控 DSP 前端设备的应用中, 往往需要将视频编码与视频分析结合在同一片处理器上运算。基于此, 本文提出了一种利用视频编码过程中得到的运动矢量与 Mean shift 相结合的跟踪算法, 不仅可以解决快速运动目标的跟踪丢失问题, 而且可减少 Mean shift 算法的

收敛次数和提高 CPU 的运算效率。

## 参考文献 (References)

- [1] Fukunaga K, Hostetler L. The estimation of the gradient of a density function, with applications in pattern recognition [J]. IEEE Transactions on Information Theory, 1975, 21(1): 32
- [2] Cheng Y. Mean shift mode seeking and clustering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(8): 790
- [3] Comaniciu D, Meer P. Mean shift analysis and application [C] // Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, 2: 1197-1203
- [4] Comaniciu D, Ramesh V, Meer P. Real-time Tracking of Non-rigid Objects Using Mean Shift [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Los Alamitos, CA, USA: IEEE Press, 2000: 142-149
- [5] Bradski G. R. Computer Vision Face Tracking for Use in Perceptual User Interface [EB/OL]. <http://www.cse.psu.edu/~rcollins/CSE598G/papers/canshift.pdf> 1998
- [6] Nummiaro Katja, Koller-Meier Esther, Van Gool Luc. Color features for tracking non-rigid objects [J]. Chinese Journal of Automation, Special Issue on Visual Surveillance, 2003, 29(3): 345-355
- [7] Zhu Sheng-li, Zhu Shan-an, Li Xu-chao. Algorithm for tracking of fast motion objects with Mean shift [J]. Opto-Electronic Engineering, 2006, 33(5): 66-70 [朱胜俐, 朱善安, 李旭超, 快速运动目标的 Mean shift 跟踪算法 [J]. 光电工程, 2006, 33(5): 66-70].
- [8] Wei Kun, Zhao Yong-qiang, Pan Quan, et al. IR target tracking based on mean shift and particle filter [J]. Journal of Optoelectronics Laser, 2008, 19(2): 213-217. [魏坤, 赵永强, 潘泉, 等, 基于均值漂移和粒子滤波的红外目标跟踪 [J]. 光电子激光, 2008, 19(2): 213-217].
- [9] Astola J, Haavisto P, Neuvo I. Vector median filters [J]. Proceedings of the IEEE, 1990, 78(4): 678-679