基于卡尔曼滤波器组的 Mean Shift 模板更新算法

朱胜利 朱善安
(浙江大学电气工程学院, 杭州 310027)

摘 要 针对 Mean Shift 算法缺乏必要的模板更新方法的缺陷,提出了一种基于卡尔曼滤波器组的 Mean Shift 模板更新算法。该算法首先将目标在特征空间中的特征值的概率作为模板信息;然后设计了一个滤波器组,其中每个滤波器用于估计特征子空间中一个子特征值概率的变化;最后将这些子特征值概率对应相乘就可以得到整个模板的更新值。由于滤波器的噪声参数是随着输入数据的变化随时动态确定的,因此,根据滤波器残差的变化就可以确定模板的更新策略。实验证明,该新算法不仅能够增强 Mean Shift 算法在目标姿态变化、光照变化下的跟踪效果,而且对阻挡时的鲁棒性也较好。

关键词 Mean Shift 算法 卡尔曼滤波器 模板更新
中图法分类号: TP391.41 **文献标识码**: A **文章编号**: 1006-8961(2007)03-0460-06

An Algorithm of Mean Shift Template Update Based a Group of Kalman Filters

ZHU Sheng-li, ZHU Shan-an
(*College of Electrical Engineering, Zhejiang University, Hangzhou* 310027)

Abstract To improve the limitation of Mean Shift lacks method of template update, an algorithm of template update based a group of Kalman filters is proposed. Probability of eigenvalue in feature space is taken as the template information. A group of filters are devised, where each filter is used to estimate the change of probability of sub-eigenvalue. All update value of template can be received by multiplying these corresponding probabilities in sub-feature space. The noise parameter of each filter would change with input data, so a novel strategy of template update according to change of residual of filters could be proposed. Experimental results show that the proposed algorithm can successfully track target under condition of changeable gesture of target and changeable illumination, and is robust to occlusion.

Keywords Mean Shift, Kalman filter, template update

1 引 言

作为计算机视觉的一个重要分支,目标跟踪在视频监控、物体识别和人机界面等领域得到广泛应用。Mean Shift 算法是一种基于密度梯度的无参数估计方法[1],其于 1975 年由 Fukunaga 提出,1999 年 Cheng 将其引入到计算机视觉领域[2],但直到近几年才引起国外学者们的广泛关注。在目标跟踪领域,Mean Shift 算法有一些很好的性质,如:(1)算法实时性好;(2)是一个单参数算法,容易作为一个模块和别的算法集成;(3)它采用核函数直方图建模,对边缘阻挡、目标的旋转、变形以及背景运动都不敏感。Comaniciu 和 Meer 对 Mean Shift 算法在图像滤波、分割和跟踪中的应用都做了较多的论述[3~5]。文献[6]论述了 Mean Shift 算法中核函数带宽的选择问题,但计算繁琐。Yang 对多维图像的 Mean Shift 方法进行了讨论[7],并使用改进的快速高斯变换来提高算法的速度;Collins 将尺度空间[8]和 Mean Shift 算法相结合解决了核函数带宽实时变化时的

收稿日期:2005-09-20;**改回日期**:2005-12-20

第一作者简介:朱胜利(1970~),男。2002 年 7 月于新疆大学机械工程学院获机械电子工程硕士学位,现为浙江大学电气学院博士研究生。研究方向为图像处理、图像目标跟踪。E-mail: qiurankee@sohu.com

目标跟踪问题,然而这两种算法都没有涉及模板更新问题。另外,Nummiaro 研究了粒子滤波器和 Mean Shift 方法相结合的情况[9],但是粒子滤波器本身的复杂计算却降低了跟踪的实时性。

Mean Shift 算法的不足之处有:(1)缺乏必要的模板更新算法;(2)由于跟踪时带宽的大小保持不变,因此当目标有尺寸变化时,可能跟踪失败;(3)由于直方图是一种比较弱的特征描述方法,因此,当背景和目标的颜色分布较相近时,算法效果欠佳。

本文针对第 1 点不足,提出了一种带有模板更新的 Mean Shift 算法,该算法首先将目标模型的颜色特征空间中的特征值的概率作为模板的信息,并设计了一个滤波器组,其中每个滤波器用于估计特征子空间中一个子特征值概率的变化,然后将这些子特征值概率对应相乘就可以得到整个模板的更新值。

2　Mean Shift 算法颜色子空间的分解

Mean Shift 算法是一种半自动跟踪方法,该算法首先在起始帧通过手工确定搜索窗口来选定运动目标;然后在起始帧和其后的当前帧同时计算核函数加权下的搜索窗口的直方图分布;最后,以两个分布的相似性最大为原则,使搜索窗口沿密度增加最大的方向移动来得到目标的真实位置。下面在对 Mean Shift 算法的起始帧、当前帧的目标模型进行简要描述的基础上,论述颜色子空间的分解和子特征值概率的计算,这些值将作为模板更新的输入数据。

2.1　初始帧的目标模型

包含目标的搜索窗口中的像素构成目标模型,也就构成了初始模板。如果令目标模型中像素的颜色值构成特征空间,那么,特征空间即为 RGB 颜色空间,其包含 R、G 和 B 3 个颜色子空间,令 $\boldsymbol{R}$、$\boldsymbol{G}$ 和 $\boldsymbol{B}$ 为 3 个子特征空间。如果每个子空间取值范围均用 0 ~ 255 表示,将每个颜色子空间分为 16 份,每份称为这个子特征空间的一个子特征值,而所有子空间的子特征值就构成了整个特征空间的特征值,共 $M=16^3=4\,096$ 个,那么目标模型中的第 m 个特征值 $u_m(m=1,\cdots,M)$ 的估计概率密度为

$$\hat{q}_{u_m}=C\sum_{i=1}^{n}k\left(\left\|\frac{\boldsymbol{X}_0^c-\boldsymbol{X}_{0,i}}{h}\right\|^2\right)\delta[b(\boldsymbol{X}_{0,i})-u_m] \quad (1)$$

式中,C 是一个常量系数,使得 $\sum_{u_m=1}^{M}\hat{q}_{u_m}=1$,有

$$C=\frac{1}{\sum_{i=1}^{n}k\left(\left\|\frac{\boldsymbol{X}_0^c-\boldsymbol{X}_{0,i}}{h}\right\|^2\right)} \quad (2)$$

令初始帧中搜索窗口共有 n 个像素,$\boldsymbol{X}_0^c$(上角标 c 代表 center)表示搜索窗口中心位置的像素坐标(下标 0 表示初始帧数,上标 c 标识此为中心像素);$\boldsymbol{X}_{0,i}$ 表示初始帧搜索窗口中第 i 个像素的坐标;$k(\|\boldsymbol{X}\|^2)$ 是核函数,h 表示核函数的带宽;函数 b 和 δ 的作用是用于判断 $\boldsymbol{X}_{0,i}$ 处的颜色值是否属于特征值 u_m。

2.2　当前帧模型

假定当前时刻为 t,类似式(1),则当前 t 帧中搜索窗口的第 m 个特征值 u_m 的概率的计算式为

$$\hat{p}_{u_m}(\boldsymbol{X}_t^c)=C_h\sum_{i=1}^{n_h}k\left(\left\|\frac{\boldsymbol{X}_t^c-\boldsymbol{X}_{i,t}}{h}\right\|^2\right)\delta[b(\boldsymbol{X}_{i,t})-u_m] \quad (3)$$

式中,$\boldsymbol{X}_t^c$ 表示当前帧搜索窗口中心像素的坐标,C_h 对应于式(1)中的 C。

2.3　颜色子空间的分解

下面在 R、G 和 B 3 个颜色子空间 $\boldsymbol{R}$、$\boldsymbol{G}$、$\boldsymbol{B}$ 中计算每个子空间中各个子特征值的概率值。令颜色子空间 $\boldsymbol{R}$ 中的第 m 个子特征值为 $u_m^{\boldsymbol{R}}$,$m=1,\cdots,16$,并将 $u_m^{\boldsymbol{R}}$ 的概率记为 $\hat{p}_{u_m^{\boldsymbol{R}}}$,则类似式(3)有

$$\hat{p}_{u_m^{\boldsymbol{R}}}(\boldsymbol{X}_t^c)=C_h\sum_{i=1}^{n_h}k\left(\left\|\frac{\boldsymbol{X}_t^c-\boldsymbol{X}_{i,t}}{h}\right\|^2\right)\delta[b(\boldsymbol{X}_{i,t})-u_m^{\boldsymbol{R}}] \quad (4)$$

式中的各个参数如式(3)所示。

同样,颜色子空间 $\boldsymbol{G}$、$\boldsymbol{B}$ 中的第 m 个子特征值分别记为 $u_m^{\boldsymbol{G}}$、$u_m^{\boldsymbol{B}}$,$m=1,\cdots,16$,它们的概率记为 $\hat{p}_{u_m^{\boldsymbol{G}}}$、$\hat{p}_{u_m^{\boldsymbol{B}}}$,则有

$$\hat{p}_{u_m^{\boldsymbol{G}}}(\boldsymbol{X}_t^c)=C_h\sum_{i=1}^{n_h}k\left(\left\|\frac{\boldsymbol{X}_t^c-\boldsymbol{X}_{i,t}}{h}\right\|^2\right)\delta[b(\boldsymbol{X}_{i,t})-u_m^{\boldsymbol{G}}] \quad (5)$$

$$\hat{p}_{u_m^{\boldsymbol{B}}}(\boldsymbol{X}_t^c)=C_h\sum_{i=1}^{n_h}k\left(\left\|\frac{\boldsymbol{X}_t^c-\boldsymbol{X}_{i,t}}{h}\right\|^2\right)\delta[b(\boldsymbol{X}_{i,t})-u_m^{\boldsymbol{B}}] \quad (6)$$

如果颜色空间中某一特征值 u_m 的颜色分量在子空间 $\boldsymbol{R}$ 中属于 $u_m^{\boldsymbol{R}}$,在子空间 $\boldsymbol{G}$ 中属于 $u_m^{\boldsymbol{G}}$,在子空间 $\boldsymbol{B}$ 中属于 $u_m^{\boldsymbol{B}}$,则做如下变换:当 $\hat{p}_{u_m^{\boldsymbol{R}}}$、$\hat{p}_{u_m^{\boldsymbol{G}}}$ 和 $\hat{p}_{u_m^{\boldsymbol{B}}}$ 等于 0 时,则令它等于 1,否则保持不变,即有

$$\hat{p}_{u_m}(\boldsymbol{X}_t^c)=\hat{p}_{u_m^{\boldsymbol{R}}}(\boldsymbol{X}_t^c)\times\hat{p}_{u_m^{\boldsymbol{G}}}(\boldsymbol{X}_t^c)\times\hat{p}_{u_m^{\boldsymbol{B}}}(\boldsymbol{X}_t^c) \quad (7)$$

即特征值的概率等于相应子特征值的概率的乘积。

3　卡尔曼滤波器组的设计

3.1　模板更新的意义

模板更新对目标跟踪算法而言,不算新问题,因

为在目标的运动中,目标的姿态及环境的照度等会发生变化,模板更新有利于跟踪。以往的模板更新方法都是将模板图像看作是一个整体,最普遍的就是先对初始帧中的整个模板和当前帧中的整个匹配区域赋予不同的权值,然后通过对二者求和来获得新的模板,即对模板中所有像素做一视同仁的修正。但在目标运动时,模板内有些像素值的变化是缓慢的,而有些则是突变的,这种更新方法显然是粗糙的。Nguyen 首次将卡尔曼滤波器用于跟踪中的模板更新[10],即通过对模板中的每个像素赋予一个卡尔曼滤波器来修正像素值的变化,这样每个像素都能独立更新,这样就获得了一个更柔性、合理的新模板。该方法虽能对目标姿态和光照变化、遮挡等表现出较好的鲁棒性,却无法应对目标在运动的同时发生尺寸变化的情况,因为这要求滤波器的数目必须不断增加或减少。为克服这个缺陷,本文进行了如下设计。

3.2 卡尔曼滤波器组的设计

由 Mean Shift 算法可知:特征空间中每两个特征值的概率值是相互独立的。如果给模板中的每个特征值赋予一个卡尔曼滤波器来预测该特征值概率的变化情况,则用一个滤波器组就可获得所有特征值概率的变化,而且无论目标的大小如何变化,特征值的个数不变。由前所述,对于彩色图像来说,需要用4 096个滤波器构成的滤波器组来估计所有特征值的变化,可以称它为虚拟滤波器组,然而由于它包含的滤波器数目太多,既显笨拙,又势必影响算法的实时性,为此,本文试图设计更好的方法来实现同样的功能。

本文在当前帧的计算中,先将颜色特征子空间 $\boldsymbol{R}$ 中 16 个子特征值赋予 16 个滤波器,然后将子空间 $\boldsymbol{R}$ 中第 m 个子特征值 $u_m^{\boldsymbol{R}}$ 的实际概率 $\hat{p}_{u_m^{\boldsymbol{R}}}$ 作为滤波器的观测值输入,这样就可以获得子空间 $\boldsymbol{R}$ 中每个子特征值概率的估计值。同样,将特征子空间 $\boldsymbol{G}$、$\boldsymbol{B}$ 中的共 32 个子特征值赋予 32 个独立的卡尔曼滤波器,就可以获得各个子特征值概率的估计值。由式(7)可知,如果进行一定的变换,由这 48 个滤波器就可以最终获得整个模板每个特征值的概率估计。这样,用这 48 个滤波器组成的实际滤波器组就可以实现与虚拟滤波器组相同的功能了,且可极大地减少滤波器的数目。

下面仅介绍子空间 $\boldsymbol{R}$ 中第 $u_m^{\boldsymbol{R}}$ 个子特征值的滤波器的设计,其余的滤波器设计进行同样操作。令 $u_m^{\boldsymbol{R}}$ 在第 k 帧搜索窗口中的理想概率为 $p_{u_m^{\boldsymbol{R}}}(k)$,设它为滤波器的待估计状态量。若将用 Mean Shift 算法在每帧搜索到的最优窗口中计算得到的 $u_m^{\boldsymbol{R}}$ 的概率作为滤波器的观测值,记为 $o_{u_m^{\boldsymbol{R}}}(k)$,则滤波器的状态模型和观测模型分别为

$$p_{u_m^{\boldsymbol{R}}}(k) = p_{u_m^{\boldsymbol{R}}}(k-1) + w(k-1) \tag{8}$$

$$o_{u_m^{\boldsymbol{R}}}(k) = p_{u_m^{\boldsymbol{R}}}(k) + v(k) \tag{9}$$

式中,$w(x)$、$v(x)$分别是状态噪声和观测噪声,均设为零均值的高斯噪声,其方差分别为 σ_{w}^2(下角标 w 代表 state)、σ_{v}^2(下角标 v 代表 observation),由于滤波器组在同样的场景中,所以对滤波器组中每个滤波器的这两个噪声的方差都设为相等。设 $\hat{p}^{\mathrm{f}}_{u_m^{\boldsymbol{R}}}(k)$ 是 k 帧 $p_{u_m^{\boldsymbol{R}}}(k)$ 的预测值,$\hat{p}^{\mathrm{E}}_{u_m^{\boldsymbol{R}}}(k)$ 为滤波后 $p_{u_m^{\boldsymbol{R}}}(k)$ 的估计值,$\hat{\sigma}_{\mathrm{E}}^2(k)$(下角 E 代表 estimation)、$\hat{\sigma}_{\mathrm{f}}^2(k)$(下角 f 代表 forecast)分别是 $\hat{p}^{\mathrm{E}}_{u_m^{\boldsymbol{R}}}(k)$、$\hat{p}^{\mathrm{f}}_{u_m^{\boldsymbol{R}}}(k)$ 的方差,且有

$$\hat{p}^{\mathrm{f}}_{u_m^{\boldsymbol{R}}}(k) = \hat{p}_{u_m^{\boldsymbol{R}}}(k-1) \tag{10}$$

$$\hat{\sigma}_{\mathrm{f}}^2(k) = \hat{\sigma}_{\mathrm{E}}^2(k-1) + \sigma_{\mathrm{w}}^2 \tag{11}$$

定义测量值和预测值的差为滤波器的残差为

$$r_{u_m^{\boldsymbol{R}}}(k) = o_{u_m^{\boldsymbol{R}}}(k) - \hat{p}_{u_m^{\boldsymbol{R}}}(k-1) \tag{12}$$

则滤波器的状态量 $p_{u_m^{\boldsymbol{R}}}(k)$ 的估计值为

$$\hat{p}_{u_m^{\boldsymbol{R}}}(k) = \hat{p}_{u_m^{\boldsymbol{R}}}(k-1) + g_{u_m^{\boldsymbol{R}}}(k) \times r_{u_m^{\boldsymbol{R}}}(k) \tag{13}$$

式中,

$$g_{u_m^{\boldsymbol{R}}}(k) = \frac{\hat{\sigma}_{\mathrm{E}}^2(k) + \sigma_{\mathrm{w}}^2}{\hat{\sigma}_{\mathrm{E}}^2(k-1) + \sigma_{\mathrm{w}}^2 + \sigma_{\mathrm{v}}^2} \tag{14}$$

均方误差为

$$\hat{\sigma}_{\mathrm{E}}^2(k) = \frac{\hat{\sigma}_{\mathrm{E}}^2(k-1)\sigma_{\mathrm{v}}^2}{\hat{\sigma}_{\mathrm{E}}^2(k-1) + \sigma_{\mathrm{v}}^2} \tag{15}$$

这样,第 k 帧中颜色子空间 $\boldsymbol{R}$ 中第 m 个子特征值 $u_m^{\boldsymbol{R}}$ 的概率值就可以依据式(13)和式(14)进行更新。同样可以设计滤波器组中其他的滤波器。

在计算得到每一帧 48 个滤波器的估计值后,如式(7)所示,就可以计算出在颜色特征空间中虚拟滤波器组的4 096个特征值的概率估计值了。

3.3 滤波器参数的设定

3.3.1 滤波器残差方差的计算

如果某滤波器残差(如式(12)的计算)远远大于前一帧统计的滤波器组的平均残差(如下面式(17)的计算),则认为这个滤波器没有正常工作。

RGB 颜色空间中的每个特征值 u_m 对应一个虚拟滤波器组中的虚拟滤波器(共 $m=4\,096$个)。下面分析滤波器残差的计算,首先统计每帧中虚拟滤波器组中正常工作的滤波器的平均残差(如下式所

示),即

$$\bar{r}^2(k) = \frac{1}{N}\sum_{u_m \in \boldsymbol{D}} r_{u_m}^2(k) \tag{16}$$

其中,$u_m \in \boldsymbol{D}$ 表示虚拟滤波器组中属于正常工作的滤波器,N 表示在正常工作的滤波器的数目,k 表示帧数;然后,从当前帧起,向前连续取 L 帧的平均残差再进行平均,即

$$\bar{r}^2 = \frac{1}{L}\sum_{t=k+1-L}^{k} \bar{r}^2(t) \tag{17}$$

即,滤波器残差的方差是对前 L 帧每个滤波器残差平方的平均。根据上一帧计算得到的 $\bar{r}^2$ 进行判断,如果第 k 帧第 m 个滤波器 u_m 的残差 $r_{u_m}(k) > 3 \times \bar{r}^2$,则认为这个滤波器没有正常工作。

3.3.2　滤波器参数的设定

上面对滤波器的设计中,两个噪声的参数 σ_w^2、σ_v^2 非常重要,而且设定这两个参数在所有特征值的滤波器中均相等。由于在动态场景中,它们常常随时间变化,所以需要随着状态的变化同时估计这两个参数。如果知道一个参数,那么就可以通过对比滤波器残差的估计方差 $\bar{r}^2$ 和它们的理论方差[11]来求得另一个的参数。注意这里对滤波器噪声参数的求取与实际滤波器组相关,而非与虚拟滤波器组相关。

滤波器参数设定时,首先定义 σ_w^2、σ_v^2 和 $\hat{\sigma}_E^2(k)$ 的初始值,即

$$\begin{cases} \sigma_v^2 = 0.5\bar{r}^2(1) \\ \sigma_w^2(0) = 0 \\ \hat{\sigma}_E^2 = 0.5\bar{r}^2(1) \end{cases} \tag{18}$$

假设观测噪声的参数 σ_v^2 已知,则由式(12)、式(16)、式(17)可得

$$\bar{r}^2 = \sigma_v^2 + \hat{\sigma}_f^2(\hat{k}) \tag{19}$$

将式(11)代入上式,得

$$\sigma_w^2 = \bar{r}^2 - \sigma_v^2 - \hat{\sigma}_E^2(k-1) \tag{20}$$

通过式(18)和式(20),就可以自动设定滤波器的噪声参数了。这样设计的好处是,当模板内像素值变化较快时,$\bar{r}^2$ 的值增加,由式(20)可知,σ_w^2 也增加。在颜色子空间 $\boldsymbol{R}$ 中,若式(14)中的 $g_{u_m^R}(k)$ 将变大,则如式(13)所示,$g_{u_m^R}(k)$ 可给 $r_{u_m^R}(k)$ 一个更大的权值,使得 $\hat{p}_{u_m^R}(k)$ 更大,这样,更能反映模板像素值的变化。

3.4　模板更新策略

虽然 3.2 节已给出了滤波器组更新模板的方法,但这仅仅适用于比较理想的情况,即光照和目标方向的变化均不大的情况,由于此时各个滤波器的估计值和测量值很接近,因而滤波器的残差都较小。如果某几个滤波器的残差变得很大,就需要判断它们所对应的特征值是否需要更新。这里模板更新采取的策略是:(1)如果连续两帧间,模板发生了大比例的变化,则不更新;(2)如果由背景物体阻挡了目标而引起滤波器残差变大,则不更新。除此两种情况外,均更新。

3.4.1　虚拟滤波器组的计算

为论述方便起见,将由滤波器估计的子特征值的概率通过式(7)换算得到的特征值的概率记为 $\hat{p}_t$。由式(12)可以得到每个子特征值的残差值,同样类似于式(7),相应子特征值的残差相乘就可以得到虚拟滤波器组中每个特征值的残差 r_t;然后,保存最后 3 帧的虚拟滤波器组的残差,并对所有这些残差取平均即可得 $\bar{r}_t$。

若不进行颜色空间的分解,则如式(3)所示直接计算当前帧中搜索窗口中的每个特征值的概率 $\hat{p}_{u_m}$。若当前帧某个特征值的残差 $\hat{p}_{u_m} - \hat{p}_t > 3 \times \bar{r}_t$,则认为与这个特征值对应的虚拟滤波器没有正常工作,进而即可找出所有没有正常工作的虚拟滤波器。

3.4.2　模板更新策略

(1)模板大比例变化

设在虚拟滤波器组中有 n 个滤波器没有正常工作,如果它们所涉及的像素点的数目大于模板所有像素点数的 40%,则认为模板变化太大,而且不论什么原因,这些特征值的概率值都不更新。

(2)边界相邻判断

当背景中某物体部分阻挡了运动目标时,则在搜索窗口的边界上,窗口中部分区域和背景中部分区域颜色是相同的。当虚拟滤波器组中某个滤波器的残差没有正常工作时,则在区域的边界上逐点向内向外各搜索一个像素,如果内外各有 S 个以上的连续点,且其颜色值属于这个滤波器所对应的颜色值,则判断为是阻挡引起了该滤波器残差变大,此时滤波器所对应的特征值的概率保持不变。如果没有出现这两种情况,则特征值的值就更新。

4　实验结果与分析

为验证本文算法的跟踪效果,从电影中截取了一段序列进行实验,图像大小 352 × 240。本文算法是在 AMD2100 + 的 CPU,256M 内存配置的电脑上,在 Windows XP 系统下用 VC6.0 编程实现。

4.1 跟踪效果

图 2 是用原始 Mean Shift 算法对序列进行处理的结果，第 69 帧是起始帧，并设定目标区域为男演员头部的正面，从 69 帧到 235 帧，演员的头部虽然变为侧面，但由于搜索窗口中仍有部分为面部区域，所以算法仍可以很好地跟踪目标。从 249 帧起，到 345 帧的前几帧，由于演员头部完全转为头后脑部，致使搜索区域搜索到与人面部颜色相近的窗户右面的柱子上，因此跟踪失败。在第 345 帧，由于头部又转为正面，所以又可以正确地跟踪了。在第 367、386 帧，头部又转为后脑部，这次，窗口搜索到了与面部颜色相近的墙面。图 3 是用本文算法对视频序列进行处理的结果，由于模板能够不断更新，因此可以看到不论头部怎么转动，搜索窗口都能很好地跟踪目标，由此可见，模板更新是必要的。

4.2 滤波器组的相关参数分析

噪声参数 σ_w^2、σ_v^2 对卡尔曼滤波器的性能是很重要的。若固定测量噪声方差 σ_v^2，则可通过式(20)确定状态噪声方差 σ_w^2。显然，由于噪声方差 σ_w^2 是随着像素值的幅度变化而不断变化的，从而使估计的状态变量更能反映像素值的变化。滤波器的均方误差用于反映状态变量的真实值和估计值的偏差，当滤波器运行平稳时，这个值应较小。图 1 反映的是采用本文算法对电影序列进行处理时，从 75 帧起 50 帧内滤波器状态噪声和均方误差的变化情况。因为所有滤波器的噪声参数均相同，所以对所有滤波器来说，图 1 是相同的。由于电影序列背景较复杂，演员头部不断转动，所以搜索窗口内的子特征值的概率变化较大，这可以反映在状态噪声方差 σ_w^2 上，从图 1 中可以看出，其值变化幅度较大，但图中的均方误差 $\hat{\sigma}_f^2(\hat{k})$ 的值则很小而且很稳定。这说明滤波器用于子特征值的估计效果很好。

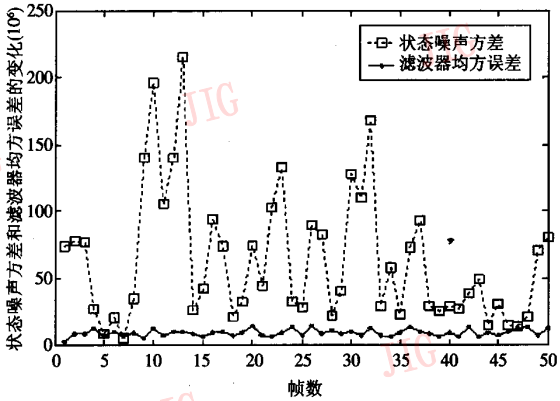
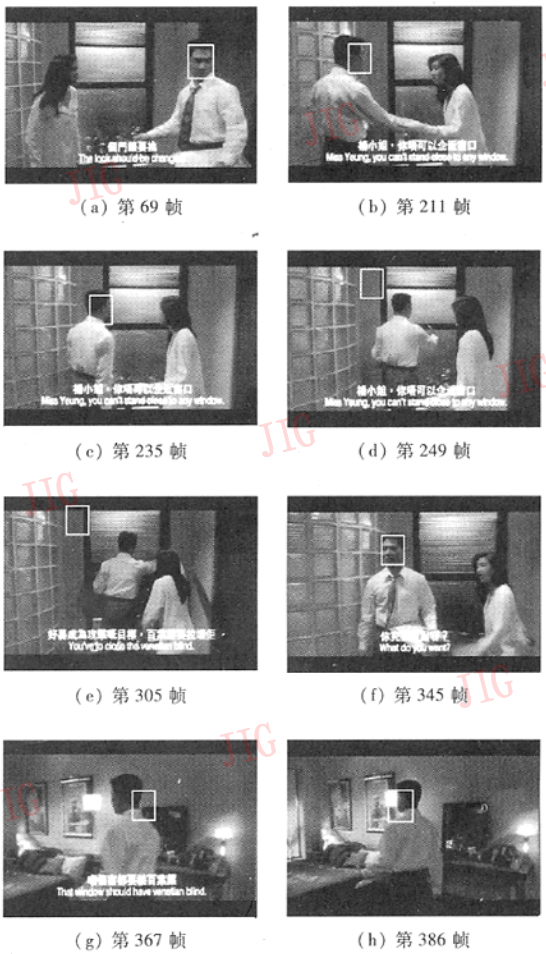


图 1 滤波器状态噪声方差和均方误差

Fig. 1 Variance of state filter of filter and variance of filter error

4.3 实时性分析

本文算法中使用的滤波器组是由 48 个标量卡尔曼滤波器组成，而且噪声参数和状态量 $\hat{p}_{u_m}(k)$ 的预测值的方差 $\hat{\sigma}_f^2(\hat{k})$ 对每个滤波器是相等的，这就使得整个滤波器组的计算可以相当简化，由于滤波器残差的计算、噪声参数的设定和滤波器的计算都可以统一规划，因而计算量很小，而且 Mean Shift 颜色子空间中子特征值的计算也并不增加多少计算量。对图像大小为 352 × 240 的视频序列(如图 2、图 3 所示的电影序列)，经典 Mean Shift 算法的处理速度为 62.5fps，如果 3.4.2 节模板更新策略中的边界相邻判断中的连续点个数 S 设为 10，那么本文算法的处理速度为 125fps 左右，可满足一般的实时性要求。



(a) 第 69 帧 (b) 第 211 帧

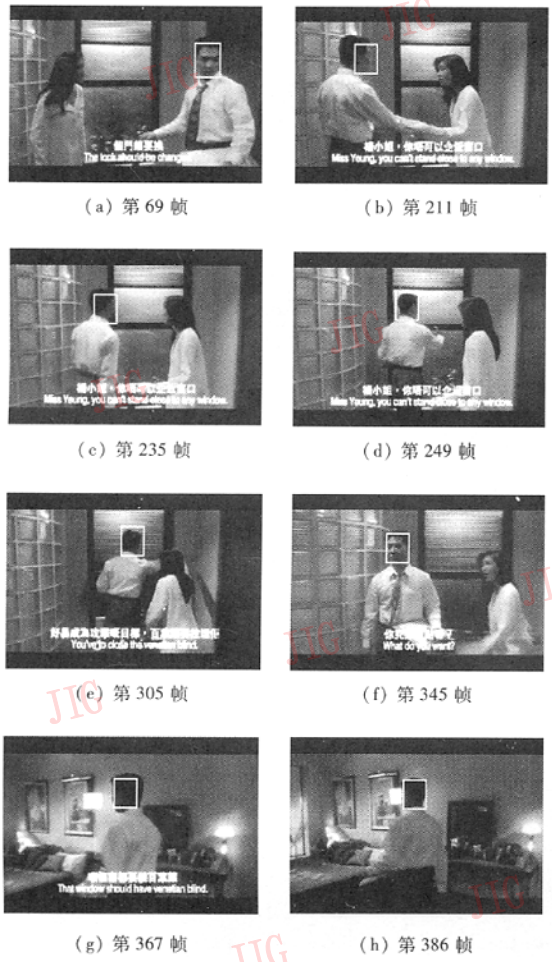
(c) 第 235 帧 (d) 第 249 帧

(e) 第 305 帧 (f) 第 345 帧

(g) 第 367 帧 (h) 第 386 帧

图 2 原始 Mean Shift 算法对序列的处理效果

Fig. 2 Processing effect of original Mean Shift algorithm



(a) 第 69 帧　(b) 第 211 帧　(c) 第 235 帧　(d) 第 249 帧　(e) 第 305 帧　(f) 第 345 帧　(g) 第 367 帧　(h) 第 386 帧

图 3　本文算法对序列的处理效果

Fig. 3　Processing effect of algorithm in this text

5　结　论

本文提出了一种基于卡尔曼滤波器组的 Mean Shift 模板更新算法,该算法是先将目标在特征空间中的特征值的概率作为模板信息来设计一个滤波器组,然后在子特征空间中计算目标模型的子特征值的概率值,并将它作为滤波器组中滤波器的输入数据,而每个滤波器则用于估计特征子空间中一个子特征值概率的变化。这些子特征值概率对应相乘就可以得到整个模板的更新值。滤波器的噪声参数是随着输入数据的变化而随时动态确定的。通过对滤波器残值的分析,就可制定相应的模板更新策略。试验证明,本算法能够保证在目标姿态变化、光照变化情况下获得较好的跟踪效果,不仅对阻挡也有很好的鲁棒性,而且具有较好的实时性。

参考文献(References)

1　Fukunage K, Hostetler L D. The estimation of the gradient of a density function with application in pattern recognition[J]. IEEE Transactions of Information Theory, 1975, **21**(1): 32 ~ 40.

2　Cheng Y. Mean shift, mode seeking, and clustering[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, **17**(8): 790 ~ 799.

3　Comaniciu D, Meer P. Mean shift analysis and application[A]. In: Proceedings of the Seventh IEEE International Conference on Computer Vison[C], Washingtan, DC, USA, 1992, **2**: 1197 ~ 1203.

4　Comaniciu D, Meer P. Mean shift: A robust application toward feature space analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, **24**(5): 603 ~ 619.

5　Comaniciu D, Meer P. robust analysis of feature spaces: color Image Segmentation[A]. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)[C], San Juan, Puerto Rico, 1997: 750 ~ 755.

6　Comaniciu D, Ramesh V, Meer P. The variable bandwidth Mean shift and data-driven scale selection[A]. In: Proceedings of IEEE International Conference on Computer Vision (ICCV'01)[C], Vancouver, Canada, 2001, **1**:438 ~ 445.

7　Yang Changjiang. Efficent Mean-Shift tracking via a new similarity measure[A]. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)[C], San Diego, California, USA, 2005, **1**: 176 ~ 183.

8　Collins R T. Mean-Shift blob tracking through scale space[A]. In: Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'03)[C], Madison, Wisconsin, USA, 2003, **2**: 234 ~ 240.

9　Nummiaro K, Koller-Meier E, Van Goo L. Color features for tracking non-rigid objects. Special Issue on Visual Surveillance[J]. Chinese Journal of Automation, 2003, **29**(3): 345 ~ 355.

10　Nguyen H T, Worring M, Rein van den Boomgaard. Occlusion robust adaptive template tracking [A]. In: Proceedings of Eighth International Conference on Computer Vision (ICCV'01)[C], Vancouver, British Columbia, Canada, 2001, **1**:678 ~ 683.

11　Maybeck P. Stochastic Models, Estimation and Control[M]. New York: Academic Press, 1982.