

基于改进码本模型的视频运动目标检测算法

姜 柯, 李艾华, 苏延召

(第二炮兵工程大学502教研室 西安 710025)

【摘要】针对经典码本模型对动态背景适应能力不足及更新算法效率不高的问题,提出了一种用快速冒泡排序和短时滑动窗口改进的码本模型。为了优化码本结构,提高活动码字首次匹配成功概率,设计了一种快速冒泡排序算法对模型码本中码字位置进行快速排序;为了实现像素的均值及偏差的快速跟踪,设计了一种短时滑动窗口算法对像素变化信息进行存储,解决了动态背景的背景自适应问题。实验表明,改进后的算法能够有效适应复杂环境下的背景变化,且具有良好的检测精度和实时性能。

关键词 码本模型; 快速排序; 短时滑动窗口; 目标检测

中图分类号 TP391

文献标识码 A

doi:10.3969/j.issn.1001-0548.2012.06.022

Moving Targets Detecting Algorithm in Video Based on Improved Codebook Model

JIANG Ke, LI Ai-hua, and SU Yan-zhao

(502 Faculty, The Second Artillery Engineering University Xi'an 710025)

Abstract In order to improve the adaptive ability and update efficiency of the classic codebook background model when detecting moving targets in complex dynamic scene, an improved codebook model algorithm is proposed. Rapid bubble sorting and short sliding window are used in the new algorithm to sort the position of the code words according to their hit times, which can improve the probability of matching the active code word at the first time. And a short sliding time window is used to buffer the change of the pixel and trace the average and deviations, which can effectively solve the adaptive problem of the model in the case of dynamic background. The result of experiment shows that the improved algorithm provides better detection precision and real-time performance in complex environment conditions.

Key words codebook model; rapid sorting; short sliding window; target detection

视频运动目标检测是将视频序列中的前景运动目标从背景中检测并分离出来^[1-2]。目前较流行的方法主要有帧差法、光流法及背景减除法。其中帧差法^[3-4]的基本思想是对相邻帧间的像素差值进行阈值判断得到目标像素,其算法逻辑简单、计算量小,但目标检测精度较差,运动目标的分割缺乏完整性;光流法^[5-6]是根据运动目标光流场信息进行检测,不需要场景的先验知识,可实现运动场景的检测,但计算量大且容易受到噪声的干扰;背景减除法原理与帧差法原理类似,它通过建立背景模型的方法构造虚拟的背景,并将待检测帧像素与该背景像素进行比较,由于背景像素可随场景的变化而不断更新,所以背景减除法对背景的动态变化具有一定的适应能力,该算法也是目前研究的热点。

背景减除法的关键是如何准确地构建及更新背

景模型。比较经典的背景减除法是混合高斯模型算法^[7-8]。该算法以多个高斯分布的加权形式来表示背景像素的概率分布,对于树叶晃动、光线渐变等动态场景有一定的适应能力。由于模型建立在概率函数的基础之上,涉及大量的开方及平方运算,算法复杂度高、运算量较大且参数调整困难,只适用于低分辨率及低帧率的情况。码本模型算法^[9-10]也是背景减除法的一种,是一种非参数化的运动目标检测方法,该算法利用每个像素点的颜色扭曲度及亮度信息构建背景码本模型,具有运算量小,检测精度高等特点。与混合高斯模型类似,码本模型算法利用每个像素码本中的多个码字来存储背景的动态信息,并通过迭代更新的方法进行更新。文献[11]通过计算区域归一化的秩和统计量自适应地调整运动目标检测阈值,并使用Mean-Shift进行码本中码字和方

差的更新, 检测精度有一定的提高, 但运算量也随之大幅提升; 文献[12]结合像素的领域信息来构建码本, 在纹理丰富的场景有较好的检测结果; 但在背景平滑以及目标同样存在大量平滑区域的场景中会产生误检; 文献[13]提出在YUV空间的码本模型算法, 码字描述更加简单高效, 但其更新算法仍有待进一步改进。

传统的码本模型算法中, 每个像素码本中的多个码字的位置在码字创建之初就已固定, 然而, 复杂环境下像素码本可能包含不只一个的码字, 当活动码字没有位于码本最顶端时, 经典模型检测过程中就需要对该码字之前的所有码字进行遍历才能正确定位到该码字上。本文提出了一种快速冒泡排序方法, 将命中数次最高的码字上浮到码本顶端, 增加码字匹配时首次成功的概率。传统码本模型的更新算法将背景模型逐帧进行更新, 当场景中的运动目标具有和背景相似的像素分布时, 更新算法造成的背景码字的错误, 本文提出了一种改进的背景更新算法, 设计一个可以抑制前景运动目标干扰的短时滑动时间窗, 存储时间窗口内的像素累加值和最大偏差, 在窗口时间结束时进行均值和偏差的更新。另外, 本文还通过改进更新公式并利用移位操作代替乘除运算的方法来减少算法运算时间, 提高算法的实时性能。

1 改进的码本模型

1.1 基于YUV空间的码本模型

由于网络监控、DV、图像采集卡等设备的视频数据多采用基于YUV空间的编码格式^[13], 且YUV空间更符合人眼对色彩的感知, 本文设计了基于YUV色彩空间的码本模型以避免从YUV空间往RGB空间转换, 码字结构如下:

$$cw_i = \{ \bar{Y}_i, y_i, \bar{U}_i, u_i, \bar{V}_i, v_i, f_i, \lambda_i, p_i, q_i \} \quad (1)$$

式中, $\bar{Y}_i, \bar{U}_i, \bar{V}_i$ 分别为像素在YUV色彩空间的YUV分量的聚类均值; y_i, u_i, v_i 分别为其匹配容许偏差; f_i 为该码字出现的次数; λ_i 为码字在两次命中的最大时间间隔; p_i, q_i 分别表示该码字出现后第一次和最后一次匹配时间。

为了更全面地记录模型检测过程中的历史信息, 本文设计了以下码本结构:

$$CB = \{ CW, H, i_{last}, c_{temp}, t_{last} \} \quad (2)$$

$$H = \{ Y_{add}, y_{bias}, U_{add}, v_{bias}, V_{add}, v_{bias} \} \quad (3)$$

式中, $CW = \{ cw_1, cw_2, \dots, cw_L \}$ 为码字集合; 增加 H 以记录一个窗口时间内像素在3个分量上的累加值

及最大偏差值, i_{last} 为上一次该码本命中的码字序号; c_{temp} 为该码本检测过程中连续命中某个特定码字的次数; t_{last} 为上一次访问该码本并匹配成功的全局时间。

1.2 码字的快速冒泡排序

复杂场景中, 随着背景的演变, 当前背景所对应的可能并非首个码字, 即其序号 $i \neq 1$ 时, 每次匹配就得遍历该码字之前的所有码字。为了提高匹配搜索速度, 本文对码字采取了快速排序操作, 在每个码字新建时定义 $f_i = 1$, 并将其置于码本的最末端, 如果 cw_i 成为活动码字而被连续命中, 则通过命中后不断递增的 f_i 来将 cw_i 不断上浮至最顶端, 其流程如图1所示。

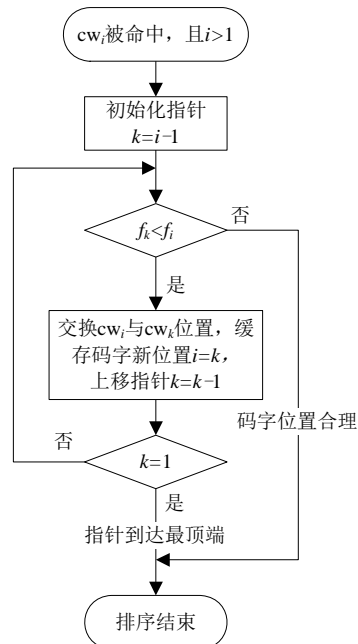


图1 快速冒泡排序流程图

相比传统的遍历排序算法, 本算法只需要将命中码字与位于它之前的码字进行比较和交换, 且命中第一个码字的情况不需要重新排序而只需要更新它的命中次数。该排序算法是冒泡排序的一个简化, 可以在线快速地将码字进行重新排序, 增加活动码字一次匹配成功概率。

2 改进码本模型的训练及检测算法

2.1 码本模型的训练

在码本模型的训练期间, 将 t 时刻的像素点 x_t 的采样值与当前的码本做比较, 如果有码字 cw_m 与其匹配, 则以码字 cw_m 作为该采样点的编码近似值, 表示该码字出现一次。训练步骤如下:

- 1) 初始化每个像素的码本。

2) 对于训练视频的每个像素的序列值:

$$X = \{x_1, x_2, \dots, x_n\}, \quad x_t = (y_t, u_t, v_t), \quad t = 1, \dots, n$$

① 如果码本为空, $L = 0$, 则创建一个码字:

$$\begin{cases} L = L + 1, \quad \bar{E}_i = x_t(e), \quad e_i = \varepsilon_{\text{set}} \\ f_i = 1, \quad \lambda_i = t - 1, \quad p_i = q_i = t \\ e \in (y, u, v), \quad E \in (Y, U, V) \end{cases} \quad (4)$$

其中, $\varepsilon_{\text{set}} \in (y_{\text{set}}, u_{\text{set}}, v_{\text{set}})$, 定义了训练阶段的初始容许偏差。

② 如果码本不为空, 则根据下面条件从码本中找出与 x_t 匹配的码字 cw_m :

$$\bar{E}_i - e_i \leq x_t(e) \leq \bar{E}_i + e_i \quad (5)$$

如果找不到, 即码本中没有码字满足上述条件, 则利用式(4)为该像素创建一个码字。

③ 如果有码字 cw_m 满足式(5), 则更新:

$$\begin{cases} \bar{E}_i = \frac{f_m \bar{E}_i + x_t(e)}{f_m + 1} \\ e_i = \max(e_i, |x_t(e) - \bar{E}_i|) \end{cases} \quad (6)$$

3) 训练结束后, 计算该像素每个码字命中的最大时间间隔, 即对于 cw_i , $i = 1, 2, \dots, L$, 有:

$$\lambda_i = \max(\lambda_i, (N - q_i + p_i - 1)) \quad (7)$$

4) 最后利用 λ 消除冗余码字, 精简得到最能代表真实背景的码本:

$$M = \{\text{cw}_k \mid \text{cw}_k \in \text{CB}, \lambda_k \leq T_M\} \quad (8)$$

T_M 取训练帧数的一半, 即 $N/2$ 。

步骤②中式(5)的条件满足情况是 x_t 和 cw_m 的颜色分量及亮度分量的偏差都在 cw_m 的容许范围内。这里只要求找到第一个满足条件的码字即可。式(8)可以剔除训练过程中得到的冗余码本, 它们是表征前景运动目标和噪声的码字, 将这些码字剔除, 从而允许初始训练过程中存在运动目标。

2.2 短时滑动窗口在检测更新过程中的应用

训练结束后, 利用得到的码本对后续帧中各像素进行前景检测, 其过程与训练过程中的步骤2)相似, 利用式(5)进行匹配。

当有像素值命中某个背景码字时, 需要对码字进行更新。经典模型中的更新算法将背景模型逐帧进行更新, 当场景中的运动目标具有和背景相似的像素分布时, 过于频繁且不加区别的更新会将对应码字中的容许区间更新到一个更接近于运动目标像素的区间上, 当目标离开后背景像素就会被已经偏离真实区间的码字拒识, 从而造成误检。本文提出了一种改进的背景更新算法, 设计一个可以抑制前

景运动目标干扰的短时滑动时间窗存储时间窗口内的像素累加值和最大偏差, 在窗口时间结束时进行均值和偏差的更新, 窗口长度为 W_{length} 。

偏差更新策略中使用了偏差可靠区间概念, 即当窗口内各分量与码字中分量均值的最大偏差 e_{bias} 位于式(9)所定义的区间时, 认为该码字所构造的容许区间能够有效地进行分类识别, 无需进行更新; 如不满足, 则对码字中的偏差量进行更新。 α 为区间控制参数, 满足 $0 < \alpha_i < 1$, \bar{E}_i 、 e_i 和 α 构成的码字容许区间及偏差可靠区间的关系如图2所示。

$$\alpha e_i \leq e_{\text{bias}} \leq (1 - \alpha)e_i, \quad e \in (y, u, v) \quad (9)$$

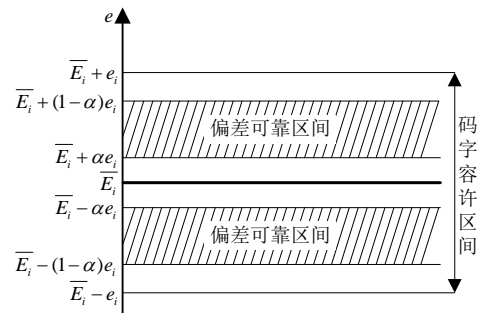


图2 码字容许区间与偏差安全区间的关系

以背景模型中的 Y 分量为例, 更新算法描述如下:

1) 当像素分量 $P_{(x,y)}$ 第一次落入码字 cw_i 的容许区间, 初始化滑动窗口及相关变量;

$$i_{\text{last}} = i, \quad c_{\text{temp}} = 1, \quad Y_{\text{add}} = y, \quad y_{\text{bias}} = y - \bar{Y}_i \quad (10)$$

2) 进行一个窗口长度的连续帧检测。如果在一个窗口内, 像素落入同一个码字 cw_i , 则计算式(11), 否则销毁窗口并转到步骤1), 启动新一轮窗口;

$$\begin{cases} c_{\text{temp}} = c_{\text{temp}} + 1 \\ Y_{\text{add}} = Y_{\text{add}} + y \\ y_{\text{bias}} = \max(y_{\text{bias}}, |y - \bar{Y}_i|) \end{cases} \quad (11)$$

当到达窗口末端时, 即 $c_{\text{temp}} = W_{\text{length}}$ 时, 启动更新, 转到步骤3)。

3) 更新均值:

$$\bar{Y}_i = (Y_{\text{add}} / W_{\text{length}} + \bar{Y}_i) / 2 \quad (12)$$

4) 更新偏差:

① 如果满足 $y_{\text{bias}} > (1 - \alpha)y_i$, 窗口内偏差大于安全区间上限, 需扩大安全区间, 则做如下更新:

$$y_i = ((1 - \alpha)y_i + y_{\text{bias}} / 2) / 2 \quad (13)$$

② 如果满足 $y_{\text{bias}} < \alpha y_i$, 窗口内偏差小于安全区间下限, 需缩小安全区间, 作如下更新:

$$y_i = y_i / 2 \quad (14)$$

5) 更新完成, 销毁窗口:

$$c_{\text{temp}} = 0, \quad Y_{\text{add}} = 0, \quad y_{\text{bias}} = 0 \quad (15)$$

为了减少浮点运算, 提高更新速度, 本文取 $W_{\text{length}} = 2^n$, n 取3~5, 即窗口长度分别为8、16、32, $Y_{\text{add}} / W_{\text{length}}$ 就可用对 Y_{add} 的移位操作实现, 即将 Y_{add} 右移 n 位, 实现除以 2^n 操作, 相应的数学表达式为 $Y_{\text{add}} \gg n$ (Y_{add} 为无符号整数)。当取 $\alpha = 1/4$ 时, 有 $\alpha y_i = y_i \gg 2$ 。整个更新中的浮点和乘除运算将大大减少。

本文选择了PETS2000中的一个视频文件对算法进行测试, 选择场景中两个具有代表性的像素进行了分析, 并将训练及检测过程中 Y 分量的码字容许区间和实时像素值记录下来, 通过波形图的形式直观地对算法效果进行观察, 如图3所示。图3a中A点所在汽车玻璃上的光照变化比较剧烈, B点的路面上共出现了3次运动前景。

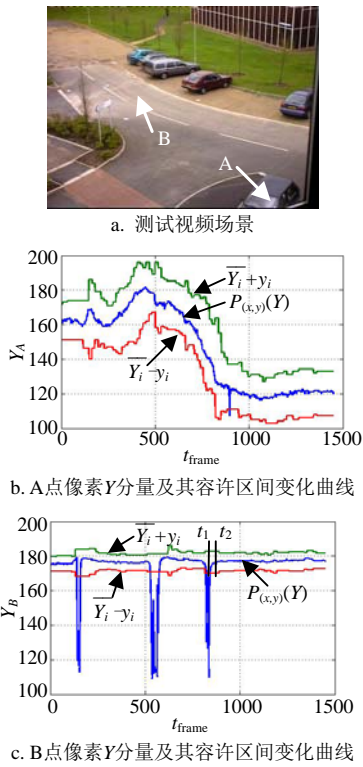


图3 光照变化情况下的像素点序列数据分析 ($\alpha = 1/4$, $W_{\text{length}} = 32$)

图3b是A点的像素波形及码字容许区间, 可以看出码字容许区间稳定地跟踪到光照变化给像素亮度分量带来的影响, 始终将该点识别为背景, 并有效的检测出了第897帧的一个前景噪点。图3c是B点的像素波形及码字容许区间, 在3次目标出现前后都给像素带来了一个剧烈变化的过程, 例如第817帧附近前景点出现后, 算法临时关闭了缓存像素信息的短时滑动窗口, 当 t_1 时刻像素重新回到码字的匹配区间时, 算法并没有立即进行参数更新而是延时一个窗口长度, 直到 t_2 时刻再重新开始更新, 有效地

滤除了从 t_1 到 t_2 区间快速变化的像素值, 提高了算法的抗干扰能力。

3 实验结果

本文选取3组典型视频文件分别和文献[13]的改进码本算法和混合高斯模型算法(MOG)进行比较。图4a为原始视频序列, 图4b为文献[13]算法检测结果, 图4c为MOG算法检测结果, 图4d为本文改进码本算法检测结果。实验条件为CPU酷睿2 E7500, 内存2 GB, Visual C++ 2008及OpenCV 2.2.1开发环境, 视频文件为RGB格式, 先进行RGB到YUV空间的转换, 再进行参数为3×3的高斯平滑, 前景检测结果没有做任何形态学处理, 直接显示和保存图片。

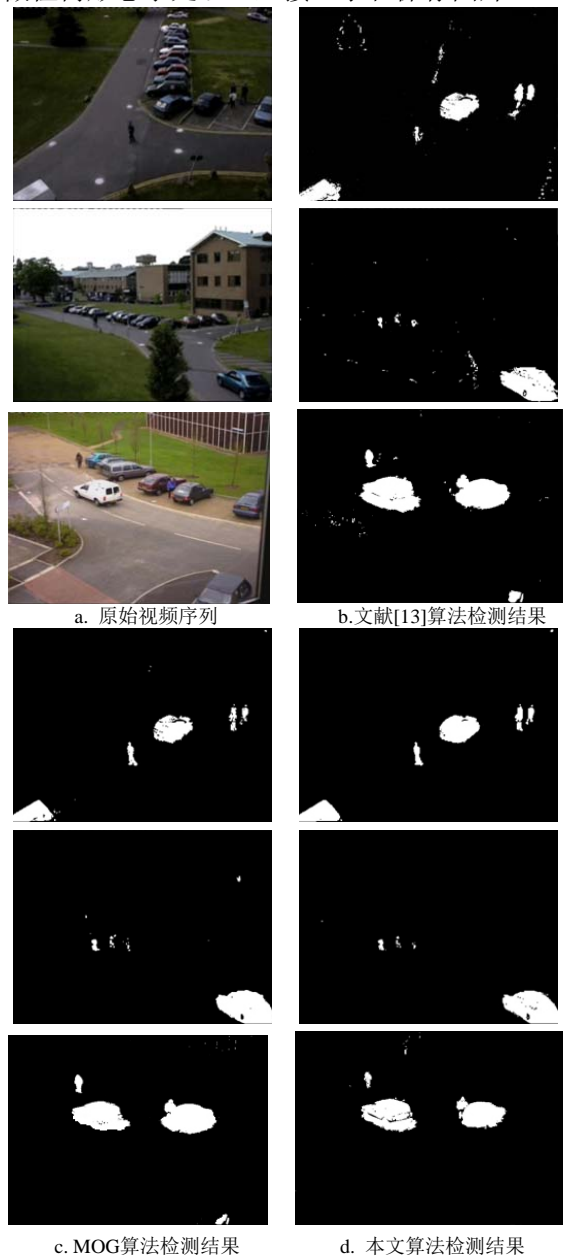


图4 3种算法的测试结果对比

通过3组视频的实测效果对比,可以看出,3种算法在训练阶段对复杂背景如轻微的树叶晃动均具有一定的学习能力,MOG和本文的改进算法性能接近,但本文的改进算法能更有效地跟踪光照变化下的背景区域(第3个视频右下角汽车玻璃)。

3种算法在训练及检测时平均每帧所消耗的时间如表1所示。表中TPT(time passed at training)和TPD(time passed at detecting)分别表示每帧训练时间和检测时间。

表1 3种算法实时性能对比

Video No	分辨率	ms					
		文献[13]		MOG		本文算法	
		TPT	TPD	TPT	TPD	TPT	TPD
1	384×288	26.5	10.6	42.8	30.6	25.3	10.1
2	384×288	27.7	12.8	45.3	33.0	26.4	11.5
3	768×576	50.3	25.6	72.3	50.6	50.5	24.2

第3组高分辨率(768×576)的视频测试结果显示,MOG在高分辨下只能达到20 帧/s,经典码本算法和本文改进的码本算法帧率都能达到40 帧/s(每帧25 ms)左右,满足实时检测30 帧/s的要求。

4 结 论

本文提出了一种基于YUV空间的改进码本模型算法,设计了YUV三分量的改进码字结构。针对经典算法中的不足,设计了码字快速冒泡排序算法,提高了码字匹配速度;利用短时滑动窗口方法改进了更新算法,提高了算法抗干扰能力。实验表明,本文改进的码本算法在精度和实时性方面都有一定的提高,适用于复杂环境下的运动目标检测。

参 考 文 献

- [1] FRIEDMAN N, RUSSELL S. Image segmentation in video sequences: a probabilistic approach[C]//13th Conference on Uncertainty in Artificial Intelligence. Providence, USA: Morgan Kaufmann Publishers Inc, 1997: 175-181.
- [2] BENEZETH Y, JODOIN P M, EMILE B, et al. Review and evaluation of commonly-implemented background subtraction algorithms[C]//19th International Conference on Pattern Recognition. Tampa, FL: IEEE, 2008: 1-4.
- [3] 李刚, 邱尚斌, 林凌, 等. 基于背景差法和帧间差法的运动目标检测方法[J]. 仪器仪表学报, 2006, 27(8): 961-964.
- [4] RUI L, LEI Y, PENGQI G, et al. Aviation video moving-target detection with inter-frame difference[C]//3rd International Congress on Image and Signal Processing (CISP). Yantai, China: IEEE, 2010.
- [5] ADIV G. Determining three-dimensional motion and structure from optical flow generated by several moving objects[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1985, 7(4): 384-401.
- [6] PARRA E G A, LOPEZ P V E. Three dimensional dynamic measurements using a stereo vision system and optical flow algorithms for high speed video applications[C]//22nd International Conference on Electrical Communications and Computers (CONIELECOMP). Cholula, Mexico: IEEE, 2012.
- [7] STAUFFER C, GRIMSON W E L. Adaptive background mixture models for real-time tracking[C]//The 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Fort Collins, USA: IEEE, 1999.
- [8] JAE K S, HO I J, GEN L, et al. Mixture of gaussians- based background subtraction for bayer-pattern image sequences[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2011, 21(3): 365-370.
- [9] KIM K, CHALIDABHONGSE T H, HARWOOD D, et al. Background modeling and subtraction by codebook construction[C]//The 2004 International Conference on Image Processing. Singapore: IEEE, 2004.
- [10] KIM K, CHALIDABHONGSE T H, HARWOOD D, et al. Real-time foreground-background segmentation using codebook model[J]. Real-Time Imaging, 2005, 11(3): 172-185.
- [11] 郭春生, 王盼. 一种基于码本模型的运动目标检测算法[J]. 中国图象图形学报A, 2010, 15(7): 1079-1083.
GUO Chun-sheng, WANG Pan. An algorithm based on codebook model to moving objects detection[J]. Journal of Image and Graphics, 2010, 15(7): 1079-1083.
- [12] MARIE R, POTELLE A, MOUADDIB E M. Dynamic background subtraction using moments[C]//The 2011 IEEE International Conference on Image Processing (ICIP). Brussels, Belgium: IEEE, 2011.
- [13] 齐美彬, 杨爱丽, 蒋建国, 等. 一种基于改进码本的车辆检测与跟踪方法[J]. 中国图象图形学报A, 2011, 16(3): 406-412.
QI Mei-bin, YANG Ai-li, JIANG Jian-guo, et al. A vehicles detection and tracking algorithm based on improved codebook[J]. Journal of Image and Graphics, 2011, 16(3): 406-412.

编辑 税红