

基于自适应跟踪窗尺度的人脸探测

雷 震¹ 王青海¹ 吴玲达² 薛廷梅¹

(装甲兵工程学院信息工程系 北京 100072)¹ (国防科技大学多媒体实验室 长沙 410073)²

摘 要 作为一种有效的迭代算法, Mean-shift 具有良好的特性, 在目标跟踪、图像平滑和其他计算机视觉领域得到了广泛应用。鉴于标准 Mean-shift 算法缺乏尺度自适应机制, 而 Camshift 算法每次探测前需要人工选定人脸区域样本才能进行准确的探测, 提出了一种用于视频中人脸探测的自适应跟踪窗算法。该算法在跟踪框内采用光照补偿和肤色分割来校正跟踪窗尺度和位置。实验表明, 该算法不但具有良好的实时性, 而且能较好地减少传统算法中的定位误差, 更加准确地探测出视频中的人脸。

关键词 人脸探测, 目标跟踪, 自适应尺度, Mean-shift

Face Detection Based on Adaptive Tracking Window Scale

LEI Zhen¹ WANG Qing-hai¹ WU Ling-da² XUE Ting-mei¹

(Department of Information Engineering, Academy of Armored Force Engineering, Beijing 100072, China)¹

(Multimedia Laboratory of National University of Defense Technology, Changsha 410073, China)²

Abstract As an effective iterative algorithm, Mean-shift is widely used in object tracking, image smoothing and other computer vision areas. In view of the fact that the standard Mean-shift tracking algorithm is lack of scale adaptation mechanism and Camshift algorithm needs choosing face local sample manually to track accurately before detecting, an adaptive tracking window scale algorithm was proposed for detecting face in video. The algorithm adopts illumination compensation and skin color segmentation to rectify the size and location of tracking window. The results of experiments show that the proposed algorithm not only has good real-time capability, but also reduces the location error of the traditional algorithm and achieves more accurate face detection in video.

Keywords Face detection, Object tracking, Scale adaptation, Mean-shift

对于运动目标的实时跟踪而言, 如何有效地提取运动目标的特征并简化运算, 是目标跟踪的关键。Mean-shift^[1-4]作为一种有效的迭代算法, 依靠样本分布进行分析, 由于具有较快的处理速度和稳定的性能, 已经广泛应用在运动目标跟踪、图像分割与平滑等多个计算机视觉领域。本质上, Mean-shift 是一种通过局部搜索实现对象检测与跟踪的算法, 能对非刚性目标进行实时跟踪, 其涉及到的一些概念, 有文献已经做了较详细的介绍, 在此仅介绍该算法的基本定义并对其进行简要描述。

定义 1 设 X 为 d 维欧氏空间 R_d , $x \in X$ 。称函数 $K: X \rightarrow R$ 为核函数, 若存在一个剖面函数 $k: [0, \infty] \rightarrow R$, 使得 $K(x) = k(\|x\|^2)$, 且满足

$$(1) k(t) \geq 0, \text{ 对 } t \in [0, \infty]$$

$$(2) k(a) \geq k(b), \text{ 若 } a < b$$

$$(3) k \text{ 分段连续, 且 } \int_0^{\infty} k(r) dr < \infty$$

在 Mean-shift 算法中, 跟踪窗的大小应该能够随着探测对象实际尺寸的变化而变化, 但由于缺乏内在的尺度自调节

机制, 在一定程度上限制了这种算法的应用。为弥补这一不足, 有研究者^[1]提出在上一帧尺度及其基础上增/减 10% 后的这 3 个尺度下执行 3 次 Mean-shift 跟踪算法, 然后找出 3 次执行结果中使相似性系数最大者对应的带宽为最优带宽, 再用 IIR 滤波器对最优带宽进行平滑处理, 进而得到本帧带宽的估计值。这种方法被称为尺度增减自适应算法。该算法由于计算量小, 执行速度快, 因而很受研究者推崇。但文献^[4]通过实现发现, 尺度增减算法所作出的“对象的真实尺度一定发生在相似性系数取最大值处”的假设未必在任何情况下都成立, 这就可能导致尺度定位出现误差。

在 OpenCV^[5]中实现了一种连续自适应 Mean-shift 算法, 但遗憾的是该算法的跟踪过程是半自动的, 在每次跟踪前都需要手工标定人脸区域样本, 如果每次跟踪都使用相同的人脸区域样本, 常常会导致跟踪框不能随人脸对象的远近变化及时更新等问题。

针对以上问题, 提出了一种 AWV-Shift (Adaptive Window Video Mean-shift) 算法, 即一种自适应窗口的视频对象跟踪算法, 该算法主要通过视频帧中运动物体的颜色信息来

到稿日期: 2009-05-14 返修日期: 2009-07-05 本文受军内科研项目基金, 装甲兵工程学院创新基金资助。

雷 震 (1976-), 男, 博士, 讲师, 研究方向为多媒体信息处理, E-mail: williamjohnmail@126.com; 王青海 (1964-), 男, 教授, 研究方向为装甲装备信息处理; 吴玲达 (1962-), 女, 博士, 教授, 研究方向为多媒体信息系统和虚拟现实技术; 薛廷梅 (1978-), 男, 讲师, 研究方向为信息处理和无线网络通信。

达到跟踪的目的。

1 跟踪窗尺度徘徊问题及其解决方案

1.1 跟踪窗尺度徘徊问题

Collins 指出尺度增减自适应算法有时候不能防止尺度在小于真实尺度下游荡,当对象尺寸增大时,这种可造成尺度定位偏差进而引起空间定位偏差的现象尤为严重。Collins 虽指出了小尺度游荡问题,但并未对其作进一步分析,并且,尺度徘徊引起的尺度定位误差还可能会进一步影响到空间的定位精度。本文通过实验发现尺度增减自适应算法在某些情况下不但不能防止尺度在小于真实尺度下游荡,而且还可能出现跟踪窗尺度在大于真实尺度时不能及时收敛的情况。通过分析发现,其严重程度不仅与对象的运动形式有关,更重要的是与对象颜色特征在尺度空间上分布的均匀程度有关。文献[4]通过实验得到相似性系数随尺度变化的3维图(如图1所示),由此图可以看出,当尺度大于真实尺度(图中白色“星号”处)时,背景像素的融入使相似性系数迅速减小;当尺度小于真实尺度时,相似性系数不但没有减小,反而增大了,并且在某小尺度处达到峰值。事实上,在某些情况下文献[4]的结论并不完全正确,当尺度大于真实尺度时,背景像素的融入尤其是一些和肤色特征相近的背景像素的融入也会使相似性系数比真实尺度下的大。这就会导致在目标人脸变小时跟踪框的面积并未出现明显减小甚至还会增大,我们将这个问题连同小尺度游荡问题一并称为跟踪窗尺度徘徊问题。

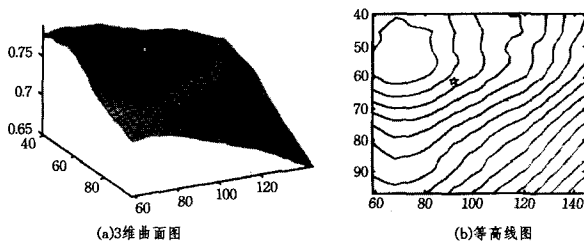


图1 人脸序列 Bhattacharyya 系数随尺度变化的3维图

1.2 尺度徘徊问题的解决方案

尺度增减算法认为3种尺度中使相似性系数最大的尺度为最优尺度,这意味着它总是假设对象的真实尺度发生在相似性系数取最大值处。绝大多数情况下这种假设看起来似乎很合理,但是否在任何情况下都成立呢?文献[4]通过实验发现,在有些情况下,当尺度小于或大于真实尺度时的相似性系数反而比真实尺度时的还要大,并且在某些非真实尺度处达到峰值。事实上,对于颜色特征在尺度空间上分布较均匀的对象,在跟踪过程中由于光照等因素的影响,很可能出现非真实尺度下的系数比真实尺度下的还要大的情形,这就可能导致尺度增减算法所依赖的基于相似性系数最大化的最优尺度判别准则有时不成立,因此出现尺度定位误差也就不可避免了。

本文提出了一种自适应跟踪窗尺度的 AWV-Shift 算法,选择在多种尺度下分别执行 Mean-shift 算法,求得对应的相似性系数,并认为多种迭代结果中应将 Bhattacharyya 系数最大的带宽作为候选最优带宽。考虑到光照等因素的影响,采用了光照补偿和肤色探测等措施来提高人脸检测精度。

1.2.1 光照补偿

进行光照补偿主要是因为肤色等色彩信息受到光源颜色、图像采集设备的色彩偏差等因素的影响,在整体上会偏本质色彩而向某个方向移动,出现色彩偏冷、偏暖,图像偏黄、偏蓝等现象。由于各新闻故事的背景、发生地点、时间等条件各异,造成光照条件各不相同。过亮、过暗等现象都会使人脸探测的正确率急剧下降,所以有必要对探测窗口进行光照补偿。

光照补偿分为彩色图像和灰度图像中的光照补偿两种。彩色图像中可采用 Rein-Lien Hsu 等提出的“参考白”方法^[6],而灰度图像中可以通过直方图均衡化、非线性变换等方法进行光照补偿。本文直接对彩色图像进行光照补偿,计算过程的参考代码如下。

```
.....
//遍历跟踪框内图像
for(i=0;i<H;i++) //H为跟踪窗高度
for(int j=0;j<W;j++) //W为跟踪窗宽度
{
//得到像素数据的偏移
IOffset = this->PixelOffset(i,j,wBytesPerLine);
//得到视频帧的三原色值
m_b = *(lpData+IOffset++);
m_g = *(lpData+IOffset++);
m_r = *(lpData+IOffset++);
//计算出灰度值
int m_gv = (m_r*299 + m_g*587 + m_b*114)/1000;
histogram[m_gv]++;
}
int calnum = 0;
int total = width * height;
int num;
//下面的循环得到满足系数 thresholdco 的临界灰度级
for(i = 0; i < 256; i++)
{
if((float)calnum/total < thresholdco)
{
calnum += histogram[255-i];
num = i;
}
else
break;
}
int averagegray = 0;
calnum = 0;
//得到满足条件的像素总的灰度值
for(i = 255; i >= 255 - num; i--)
{
averagegray += histogram[i] * i;
calnum += histogram[i];
}
averagegray /= calnum;
//得到光线补偿的系数
float m_p = 255.0 / (float)averagegray;
//下面的循环对图像进行光线补偿
for(i = 0; i < H; i++)
for(int j = 0; j < W; j++)
{
IOffset = this->PixelOffset(i,j,wBytesPerLine);
//蓝
```

```

m_b = *(lpData+IOffset);
m_b *= m_p;
//临界点处理
if(m_b > 255) m_b = 255;
*(lpData+IOffset) = m_b;
//绿
m_b = *(lpData+IOffset+1);
m_b *= m_p;
if(m_b > 255) m_b = 255;
*(lpData+IOffset+1) = m_b;
//红
m_b = *(lpData+IOffset+2);
m_b *= m_p;
if(m_b > 255) m_b = 255;
*(lpData+IOffset+2) = m_b;
}
.....

```

1.2.2 肤色探测

本文将肤色判断作为 AWV-Shift 定位的后期验证。由于人脸探测过程中可能存在误判或漏判,因此可以将肤色信息作为验证探测结果的手段,排除不是人脸的候选脸,最大限度地降低探测算法的误判率。对多幅图像进行统计后发现,在 HSI 空间中,大部分肤色色调集中在 $[0, 0.7]$ 范围内,而且传统的方法是在一幅图像中仅判别严格意义上的肤色,其后果是人的眼睛、嘴巴被排除在外。为此,将选取范围扩大到:色调 $[0, 0.7] \cup [6.10, 6.28]$,并且 $R > G, R > B$ 。这样,嘴巴等区域中的很多像素都被判别为人脸肤色,再经过滤波处理,基本上组成了一个人脸的整体。实验证明亮度对肤色选取影响比较大,为此将亮度限制在 $[80, 230]$ 范围内。

由于彩色图像中的色彩信息容易受到光照明暗、光源色彩等外在因素的影响,鲁棒性弱,很难找到一种普适性良好的肤色模型,导致某些图像中的人脸区域没有被判断为肤色,而非人脸区域却被判断为肤色,同时还会出现一些孤立点。

为了提高探测率,减少探测时间,有必要将这些孤立点、误判点去除。采用的解决方法为利用 55×77 滤波,并对肤色与非肤色点分别进行处理,以减少发生这种情况的可能性。当某非肤色点周围的肤色点个数大于 25 时,认为该点为肤色区域;当某肤色点周围的肤色点个数小于 15 时,认为该点为非肤色点。

1.2.3 AWV-Shift 算法

AWV-Shift 算法的基本思想是对所有视频帧作 Mean-shift 运算,并将上一帧中跟踪窗的中心和大小作为下一帧 Mean-shift 算法的跟踪窗的初始值,如此进行迭代运算,以实现视频中人脸对象的跟踪。整个算法的具体步骤如下。

Step1 初始化跟踪窗的大小和位置。

Step2 对跟踪窗内的图像进行光照补偿。

Step3 在 HSI 空间中根据其中的 H 分量计算跟踪窗内的色彩直方图,并根据获得的色彩直方图计算跟踪窗内的色彩概率分布图像。

Step4 运行 Mean-shift 算法,获得跟踪窗新的中心和位置。

其中,Mean-shift 算法又可归结为以下几个步骤:

(1)初始化当前图像候选对象的中心位置 y_0 , 分别计算

$\hat{p}_u(y_0)$ 以及 $\rho[\hat{p}_u(y_0), \hat{q}_u]$;

(2)更新权值 w_i ;

(3)确定候选对象中心的新位置:

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left|\frac{y_0 - x_i}{h}\right|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}$$

并根据结果计算 $\rho[\hat{p}_u(y_1), \hat{q}_u]$;

(4)当 $\rho[\hat{p}_u(y_1), \hat{q}_u] < \rho[\hat{p}_u(y_0), \hat{q}_u]$ 时,则 $y_1 = 0.5(y_0 + y_1)$;

(5)若 $\|y_0 - y_1\| < \epsilon$, 则迭代结束, y_1 即为候选对象窗口中心;否则令 $y_1 = y_0$, 返回到步骤(1)。

Step5 在 HSI 颜色空间中对跟踪窗内的区域进行肤色探测,找出其中的肤色区域,然后在粗糙的分辨率下进行区域增长以连接各肤色区域,将连通后肤色区域的外接矩形作为新的跟踪窗。

Step6 间隔一帧后的下一帧视频图像中,用 Step5 获得的值来更新跟踪窗的位置和大小,然后跳转到 Step2 继续运行。

由于以上的光照补偿和肤色探测仅在跟踪框内进行,因此大大降低了算法的计算量,提高了算法的执行速度。

2 实验结果

本文在实验的基础上对跟踪窗尺度徘徊问题进行了深入分析,从而进一步佐证了本文所提出的 AWV-Shift 算法。我们发现,如果没有光照变化并且每次跟踪前都手工标定人脸区域样本,OpenCV 中采用的连续自适应 Mean-shift 算法的跟踪效果还比较令人满意,但是一旦出现光照变化或每次跟踪都使用相同的人脸区域样本,就会产生如图 2 所示的跟踪结果。从图 2(b),图 2(c)可以看出,跟踪框并没有随人脸目标的尺寸同步变化,在目标人脸变小时跟踪框的面积并未出现明显减小。

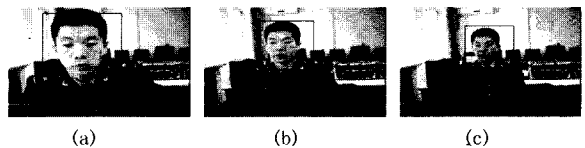


图 2 跟踪框尺度徘徊现象

我们将 AWV-Shift 方法分别用于监控视频后得到的实验结果,如图 3—图 5 所示。

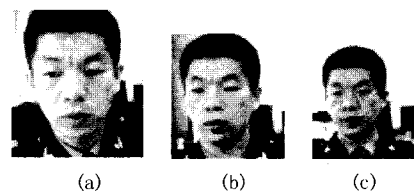


图 3 跟踪窗内的视频帧进行光照补偿的结果

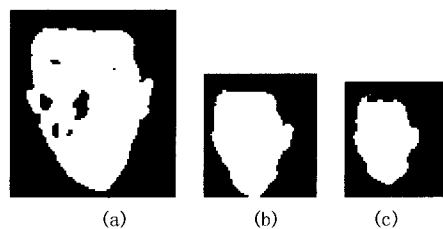


图 4 跟踪窗内的视频帧进行肤色探测的结果

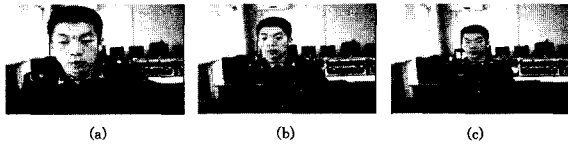


图5 AWV-Shift算法的人脸探测结果

实验结果表明,AWV-Shift算法能够快速准确地探测出视频中的人脸目标,并且跟踪窗的尺寸能够随着人脸目标尺寸的变化自适应地进行调整。该算法在探测速度方面与目前广泛采用的 Adaboost 算法接近,并且在一定程度上降低了 Adaboost 算法的误判率。

结束语 本文将光照补偿和肤色探测作为修正人脸位置的后期手段。光照补偿和肤色探测由于仅在跟踪框内进行,因此能够以很小的计算成本提高传统 Mean-shift 算法的探测精度,在探测速度上也能够满足实时性要求。并且,本算法探测过程完全不需要人工参与,不像 OpenCV 中的连续自适应 Mean-shift 算法那样,在每次跟踪前还得手工标定人脸区域样本。

参考文献

- (上接第 254 页)
- [12] Zahara, Erwie, Hu, et al. Solving constrained optimization problems with hybrid particle swarm optimization [J]. *Engineering Optimization*, 2008, 40(11): 1031-1049
- [13] Cao Y J, Wu Q H. Convergence analysis of adaptive genetic algorithm, genetic algorithms in engineering systems [J]. *Innovations and applications*, September 1997, 446-450
- [14] Powell D, Skolnick M. Using genetic algorithms in engineering design optimization with non-linear constraints [C]// *Proceedings of the fifth international conference on genetic algorithms*. 1993: 270-271
- [15] Kuri-Morales A F, Gutierrez-Garia J. Penalty function methods for constrained optimization with genetic algorithms; a statistical analysis [M]. Berlin Heidelberg: Springer-Verlag, 2002: 187-200
- [16] Li Ai-guo. Particle swarms cooperative optimizer [J]. *Journal of Fudan University, Natural Science*, 2004, 43(5)
- [17] 许艳. 基于改进遗传算法的自动组卷研究 [J]. *计算机与信息技术*, 2008, 7(2)
- (上接第 257 页)
- [4] Fei Sheng-wei, Sun Yu. Forecasting dissolved gases content in power transformer oil based on support vector machine with genetic algorithm [J]. *Electric Power Systems Research*, 2008, 78(3): 507-514
- [5] Pai Ping-feng. System reliability forecasting by support vector machines with genetic algorithms [J]. *Mathematical and Computer Modelling*, 2006, 43(3): 262-274
- [6] 张袅娜, 张德江, 冯勇. 基于混沌遗传算法的柔性机械手滑模控制器优化设计 [J]. *控制理论与应用*, 2008, 25(3): 451-455
- [7] Schölkopf B, Smola A. *Learning with kernels: support vector machines, regularization, and beyond* [R]. Cambridge, MA: MIT Press, 2002
- [8] Wang Wenjian, Xia Zongben, Lu Weizhen, et al. Determination of the spread parameter in the Gaussian kernel for classification and regression [J]. *Neurocomputing*, 2003, 55(3): 643-663
- [9] 杜京义, 侯媛彬. 基于遗传算法的支持向量回归机参数选取 [J]. *系统工程与电子技术*, 2006, 28(9): 1430-1433
- [10] 姚俊峰, 梅焱, 彭小奇, 等. 混沌遗传算法及其应用 [J]. *系统工程*, 2001, 19(1): 70-74
- [11] Tavazoei M S, Haeri M. Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms [J]. *Applied Mathematics and Computation*, 2007, 187(2): 1076-1085
- [12] Yang Dixiong, Li Gang, Cheng Gengdong. On the efficiency of chaos optimization algorithms for global optimization [J]. *Chaos, Solitons & Fractals*, 2007, 34(4): 1366-1375
- (上接第 260 页)
- 参考文献**
- [1] Samad T, Bay J S, Godbole D. *Network-Centric Systems for Military Operations in Urban Terrain: The Role of UAVs* [J]. *Proceedings of the IEEE*, 2007, 95(1): 92-107
- [2] Nolle L, Wong K C P, Hopgood A A. DARBS: A Distributed Blackboard System [M]// Bramer, Coenen, Preece, eds. *Research and Development in Intelligent Systems XVIII*. Springer, 2001: 161-170
- [3] A Hovercraft Testbed for Decentralized and Cooperative Control [C]// *Proceeding of the 2004 American Control Conference*. Boston, Massachusetts, 2004
- [4] Popp R. MTE Ground Station Testbed - A Battlefield Awareness Asset for GMTI Exploitation
- [5] Shields J F. The Formation Control Testbed Celestial Sensor: Overview, Modelling, and Calibrated Performance
- [6] King E, Kuwata Y, Alighanbari M, et al. Coordination and Control Experiments on a Multi-vehicle Testbed
- [7] McLain T W, Randal W. Beard Unmanned Air Vehicle Testbed for Cooperative Control Experiments
- [8] Niland W M. The Migration of a Collaborative UAV Testbed into the Flames Simulation Environment [C]// *Proceedings of the 2006 Winter Simulation Conference*
- [9] Wise R, Rysdyk R, Seattle, et al. Multi-vehicle Cooperative Control Flight Test [C]// *Anawat Pongpunwattana, MO. IEEE 25th Digital Avionics Systems Conference*. October 2006
- [10] Hoppe H. Terrain Rendering Using GPU-Based Geometry Clipmaps Arual Asirvatham, Microsoft Research, 2004