

基于 Mean Shift 的自适应尺度变化跟踪算法研究

何小诚, 黄 凯, 谭毅华, 田金文

(华中科技大学 多谱信息处理技术国防科技重点实验室, 湖北 武汉 430074)

摘 要: 在实时跟踪系统中, 要求在跟踪过程中跟踪窗口的大小实时地适应运动目标的外观变化, 这对应着 Mean-shift 的尺度变化. 针对跟踪这种尺度变化的问题, 在跟踪框内检测角点并进行匹配, 将所有得到的匹配角点对建立仿射模型, 并采用最小二乘法求解得到尺度变化系数, 进而更新跟踪框尺度, 使得 Meanshift 算法可自适应地变化尺度并跟踪到大小不断变化的目标. 实验结果表明, 提出的算法具有较好的准确性、鲁棒性和实时性.

关键词: Mean Shift; 目标跟踪; 角点; 仿射模型

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2010)04-0069-06

Adaptive Scale Object Tracking with Based on Mean Shift

HE Xiao-cheng, HUANG kai, TAN Yi-hua, TIAN Jin-wen

(The State Key Laboratory for Multi-spectra Information Processing Technology,
Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: In real time tracking system, the size of tracking window should be updated when the scale of the moving object is being varied, which is corresponding to the scale change of mean shift. In order to solve the problem, this paper proposed a method to catch the adaptive scale change. Firstly, the corner points are detected in the tracking window. Secondly, the detected points are matched with the preceding frame, and the affine model is constructed based on the matching results. Thirdly, least square method is taken to induce the scale coefficients. Lastly, the obtained scale is applied to the mean shift tracking framework. Experimental results show the improved algorithm could track objects precisely when scale of the object changes progressively, as well as is robust and operated real time.

Key words: Mean Shift; objects tracking; corner points; affine model

1 引言

目标跟踪一直是计算机视觉研究领域的热点问题, 一个实用的跟踪系统要求能够实时适应目标因各种运动导致的外观变化, 然而现在大多数的跟踪算法在跟踪存在明显尺度变化的目标(尤其是当目标尺寸逐渐增大以至超出跟踪窗口范围)时, 常常会导致目标的丢失. 目前, 还没有系统的方法来解决跟踪窗口尺寸的自动调整问题.

Mean Shift 算法作为一种高效的模式匹配算法, 已经成功地应用在对实时性要求较高的目标跟踪应用中^[1]. 文献[2-4]中对 Mean Shift 算法核窗宽

(即跟踪窗口尺寸)的选取提出了改进. 文献[5]中基于连续帧中刚性运动物体满足仿射模型的假设, 利用相邻帧跟踪窗口内的匹配特征点建立仿射模型来估算伸缩幅值. 该方法对刚体目标的大小变化有较好适应性, 但特征点的匹配算法存在不稳定性, 文中也对该算法提出了进一步的改进算法.

2 Mean Shift 跟踪算法

在当前帧中定位目标的问题可以采用如下方法解决: 设 z 表示目标的某种特征, 如颜色或纹理等, 其密度分布为 q_z , 同时, 在位置 y 处的候选目标, 其特征 z 的分布为 $p_z(y)$, 那么跟踪问题转化为求位

收稿日期: 2009-03-18; 修回日期: 2009-05-18

基金项目: 国家自然科学基金项目(60572048); 华中科技大学科技基金项目

置 y , 使得 $p_z(y)$ 与 q_z 最相似. 相似度采用 Bhattacharyya 系数表征. 其定义为

$$p(y) = \int \sqrt{p_z(y)q_z} dz \quad (1)$$

取目标像素的特征值(如灰度)为特征, 分别建立目标模型和候选模型, 并引入 kernel 计算直方图 $\{q_u\}$ 和 $\{p_u(y)\}$ 如下:

$$q_u = C \sum_{i=1}^n k(\|x_i^*\|^2) \delta[b(x_i^*) - u] \quad (2)$$

$$p_u(y) = C_h \sum_{i=1}^n k\left(\frac{\|y - x_i^*\|^2}{h^2}\right) \delta[b(x_i^*) - u] \quad (3)$$

其中

$$C = \frac{1}{\sum_{i=1}^n k(\|x_i^*\|^2)}$$

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\frac{\|y - x_i^*\|^2}{h^2}\right)}$$

是归一化常数. x_i^* 为以 0 为中心的目标像素坐标, x_i 为以 y 为中心的候选目标像素坐标. $b(x)$ 表示 x 点像素在原始直方图(未经 kernel 加权)中对应值.

假设目标的初始位置为 y_0 , 由 Taylor 公式可知

$$\rho(y) \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(y_0)q_u} + \frac{1}{2} \sum_{u=1}^m p_u(y) \sqrt{\frac{q_u}{p_u(y_0)}}$$

将式(2)代入上式, 得

$$\rho(y) \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(y_0)q_u} + \frac{1}{2} \sum_{u=1}^m \omega_u k\left(\frac{\|y - x_i\|^2}{h^2}\right)$$

其中

$$\omega_i = \delta[b(x_i) - u] \sqrt{\frac{q_u}{p_u(y_0)}} \quad (4)$$

由式(3)可知, 第 1 项独立于 y_0 , 因此要使 $\rho[\cdot]$ 极大, 只需使第 2 项极大. 而第 2 项有式(1)的形式, 可用 Mean Shift 算法取得使第 2 项极大时的 y . 迭代过程如下:

$$y_{j+1} = \frac{\sum_{i=1}^n x_i \omega_i g\left(\frac{\|y_j - x_i\|^2}{h^2}\right)}{\sum_{i=1}^n g\left(\frac{\|y_j - x_i\|^2}{h^2}\right)}, j = 1, 2, \dots \quad (5)$$

3 跟踪窗尺度变化算法

跟踪窗口的大小在目标跟踪的迭代过程中起关键作用. 跟踪窗口过大, 跟踪窗内过多的背景区域会影响跟踪结果; 另外, 在多目标跟踪中, 一个跟踪窗内就可能包含多个跟踪目标等等. 跟踪窗口过小, 跟踪窗口内过少的目标信息不能有效跟踪目标, 从而使得目标丢失. 因此, 在跟踪过程中更新跟踪窗口是改进跟踪算法的关键.

文献[5]中基于连续帧中刚性运动物体满足仿射模型的假设, 利用相邻帧跟踪窗口内的匹配特征点建立仿射模型来估算伸缩幅值, 从而更新跟踪窗的尺度. 但文中角点匹配的方法存在一些问题. 一方面, 角点匹配的方法存在问题, 在匹配的过程中采用单个点的灰度最相似原则会使匹配产生误差; 另一方面, 在求解仿射模型的时候, 在所有匹配的角点中挑选固定个数的匹配角点, 若挑选的角点不在目标上或是不能反映目标的尺度特征, 则利用仿射模型得到的伸缩幅值误差很大.

文中针对上述两个问题, 主要对角点匹配的方法和仿射模型的求解方法进行了改进, 提出了基于模板的角点匹配方法和仿射模型的最小二乘求解方法, 增加了匹配的准确性, 进一步提高了跟踪窗尺度变化的准确性.

3.1 基于目标的角点匹配方法

对于刚性物体, 角点能很好地刻画其空间结构, 而且易于检测. 因此, 采用两帧中匹配的角点作为样本来对仿射模型参数进行估算^[5]. 角点是二维图像亮度变化剧烈的点或图像边缘曲线上具有曲率极大值的点. 利用角点的这种特性, 可以用角点来标记目标上的特征点.

首先对连续两帧跟踪框内进行角点检测, 将所有检测得到的角点坐标记录下来. 接着对两帧的角点逐个进行判断匹配, 判断方法如图 1 所示. 逐个扫描当前帧跟踪框内检测得到的每个角点的位置, 并在上一帧中该位置的邻域内进行搜索, 判断邻域内角点情况. 若无候选角点, 则说明两帧之间没有匹配的角点, 直接将该角点忽略; 若有一个候选角点, 则直接将该角点作为匹配的角点; 若有两个或两个以上的候选角点, 则按角点模板的方法进行匹配.

当有两个或两个以上的候选角点时, 必须对候选的角点进行筛选, 选择与目标角点最相似的角点作为匹配角点. 在一般的角点匹配中, 会直接以两个角点的灰度最相似原则将两个点作为匹配的角点,

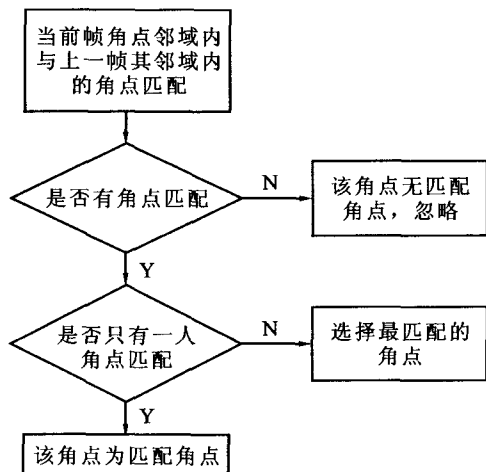


图1 判断匹配角点流程图

这种方法简单易行,但也存在匹配误差的问题。

在角点检测的过程中,我们希望检测得到的角点能全部在被跟踪目标的边缘或者能刻画其形状特征的位置上,这样角点就能很好地表示被跟踪目标的形状大小变化,再利用仿射模型得到的跟踪框尺度变化系数才能准确反映目标大小变化。但在角点检测的过程中不可避免的会检测得到一些干扰点。如图2所示,在检测得到的角点中,大部分的角点是在被跟踪目标的边缘或者内部能较好刻画目标特征的点,但也有一部分角点是非目标上的点,这就对角点匹配形成了干扰。

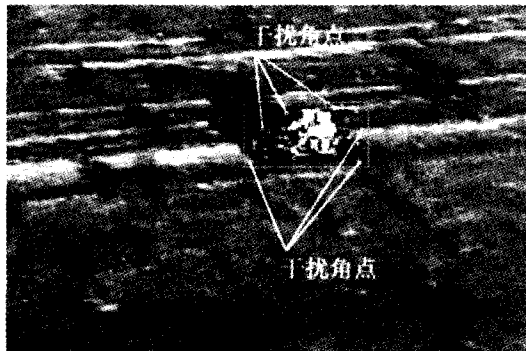


图2 干扰角点示意图

在进行角点匹配的过程中,如果只以简单的灰度最相似原则进行匹配,当目标角点其邻域内存在与其灰度相似的干扰角点时,则很有可能出现将目标角点与干扰角点匹配的错误结果结果,如图3所示。

图3为两帧图像跟踪框内角点匹配的示意图,前一帧图像检测得到的角点作为目标角点与后一帧检测得到的多个候选角点在匹配过程中会有上述的筛选过程。前一帧目标角点的灰度为97,而两个候选角点的灰度为94和101(其中干扰角点的灰度为

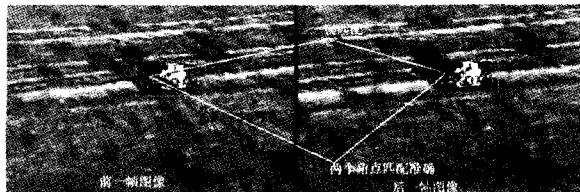


图3 角点匹配示意图

94)。目标角点与实际相匹配角点的灰度差值为4,与干扰角点的灰度差值为3,若以灰度值最相似原则匹配则将干扰角点与目标角点相匹配,则会产生如图中误匹配的情况。我们所希望的是图中下面两个角点匹配准确,但结果是图中上面标注的目标角点与干扰角点匹配产生误匹配的情况。因此,仅用灰度最相似原则匹配角点容易出现匹配误差,导致仿射模型的尺度变化系数不准确。

当有两个或两个以上的候选角点时,采用如下角点模板的方法进行匹配:

首先,分别记录当前帧角点的坐标和上一帧其邻域内可能与其匹配的角点的坐标。以当前帧角点为中心,建立一个邻域模板(例如 7×7 、 9×9)作为目标模板;同样的方法在上一帧以可能匹配的角点为中心建立相应的模板,作为匹配模板。将目标模板与匹配模板依次比较,比较的方法是将模板每个像素点的灰度值对应相减取绝对值并累加,累加和最小的那个模板作为与匹配模板,相应的角点作为匹配角点。设当前帧角点坐标为 (x, y) ,其对应的灰度为 $I(x, y)$,上一帧中候选角点坐标为 (x', y') ,其对应的灰度为 $I'(x', y')$,则分别以这两个角点为中心按图4所示建立模板(以 7×7 模板为例):

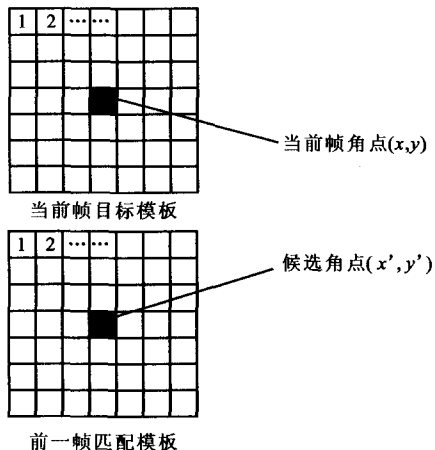


图4 角点模板匹配示意图

匹配时将两个模板对应的位置1, 2, ... 的灰度值分别相减并取绝对值再求和,如下式:

$$\Delta I_k = \sum_{j=-M}^M \sum_{i=-M}^M |I(x_i, y_i) - I'(x'_i, y'_i)| \quad (k=1, 2, \dots, K) \quad (6)$$

$$\Delta I_{\min} = \min\{\Delta I_k \mid k=1, 2, \dots, K\} \quad (7)$$

式中, M 为模板宽度的一半(如 7×7 模板则 $M=3$), K 为候选角点的个数. 取 $\Delta I_{\min}$ 对应的角点作为匹配角点.

采用上述角点模板匹配的方法, 充分利用角点本身和其领域信息, 进一步限制了匹配的条件, 避免仅使用单个角点的灰度信息导致匹配不准确的问题, 提高了匹配的准确性.

3.2 仿射模型的最小二乘求解方法

如果目标本身是刚体, 或者它的运动在图像上的投影可以近似为一个平面的运动, 就可以用仿射变换来表示两帧之间目标的平移、旋转和比例缩放的变化, 如果能得到这些仿射变换的参数, 就可以知道目标尺寸的缩放比例, 进而确定 Mean Shift 算法中核函数的带宽了. 四参数仿射模型公式如下:

$$\begin{aligned} x'_i &= k \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} x_i + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \\ &= \begin{bmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{bmatrix} x_i + \begin{bmatrix} a_3 \\ a_4 \end{bmatrix} \end{aligned} \quad (8)$$

式中, x'_i 和 x_i 分别是同一个目标特征点在帧 i 和 $i+1$ 中的位置, k 表示目标大小的伸缩幅度, θ 表示目标的旋转角度, Δx 和 Δy 分别表示该目标特征点的水平和垂直位移.

在四参数仿射模型中, 我们需要四个输入参数便可利用式(8)计算得到目标大小的伸缩幅度 k , 即需要输入前一帧图像两个角点的坐标以及这两个角点在一帧中与它们匹配角点的坐标. 但如果单独以两对角点坐标进行仿射变换, 当选择的角点不能反映被跟踪目标尺度变化特征时, 仿射模型得到的尺度变换系数也不准确.

图 5 和图 6 分别显示了两种角点选择的情况. 图 5 的两个角点都是目标上的角点, 也能反映目标的形状尺度变化, 但这两个角点距离比较近, 其运动变化趋势也比较相似, 从而不能很好反映目标大小伸缩的情况, 通过仿射模型得到的尺度变化系数也相应的不准确. 图 6 的两个角点都属于干扰角点, 利用干扰角点得到的尺度变化系数显然是不准确的. 这两种情况下选择的角点都对利用仿射模型得到尺度变化系数造成了很大的影响并会带来尺度变化的错误. 因此, 仅用两个角点对建立仿射模型容易产生较大误差.

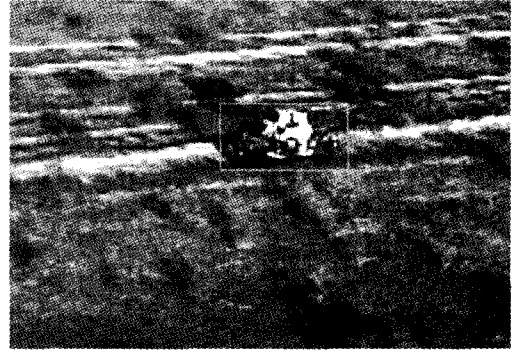


图 5 角点选择示意图

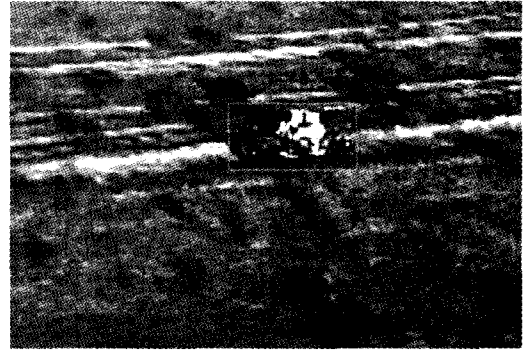


图 6 角点选择示意图

文献[5]中建立仿射模型的方法是选择一对匹配的角点利用式(8)建立仿射模型, 如果选择的角点不在物体上或者不能反映目标尺度的变化, 则会产生很大的误差. 为了避免上述问题, 我们提出利用上一节中得到的两帧图像跟踪框内所有的匹配角点对, 利用这些角点对采用最小二乘法建立仿射模型, 由仿射变换得到尺度变化系数. 一方面, 利用所有的匹配角点对计算尺度变化系数能够利用所有的表征被跟踪目标的特征变化信息, 避免了仅用两个角点产生的误差, 进一步提高了仿射模型的准确性; 另一方面, 所有的匹配角点对里可能会有部分干扰角点对, 但由于绝大部分的匹配角点对都是能表征目标的大小特性变化的角点对, 可以减小干扰角点对产生的误差影响, 特别是能够避免如图 6 中选择了两个干扰角点对而产生较大误差的情况^[6-7].

式(8)中四参数仿射模型采用最小二乘求解方法得到的系数可由以下公式得出:

$$a_1 = \frac{M \sum_{i=1}^M (x_i x'_i + y_i y'_i) - (\sum_{i=1}^M x_i)(\sum_{i=1}^M x'_i) - (\sum_{i=1}^M y_i)(\sum_{i=1}^M y'_i)}{M \sum_{i=1}^M (x_i^2 + y_i^2) - (\sum_{i=1}^M x_i)^2 - (\sum_{i=1}^M y_i)^2} \quad (9)$$

$$a_2 = \frac{M \sum_{i=1}^M (x'_i y_i + x_i y'_i) - (\sum_{i=1}^M x'_i)(\sum_{i=1}^M y_i) + (\sum_{i=1}^M x_i)(\sum_{i=1}^M y'_i)}{M \sum_{i=1}^M (x_i^2 + y_i^2) - (\sum_{i=1}^M x_i)^2 - (\sum_{i=1}^M y_i)^2} \quad (10)$$

式中, M 为所有匹配角点的个数, x_i, y_i 为当前帧角点坐标, x'_i, y'_i 为上一帧中与角点 (x_i, y_i) 匹配的角点坐标. 由式(8)可以得出变框系数 k 可由下式得到:

$$k = \sqrt{a_1^2 + a_2^2} \quad (11)$$

由变框系数 k 更新跟踪窗口的大小如下式:

$$H = H \cdot k \quad (12)$$

$$W = W \cdot k \quad (13)$$

式中, H 和 W 分别为跟踪窗的高和宽.

3.3 算法描述

在通过回归计算得到目标仿射模型以后,可以利用式(12)、式(13)对当前跟踪窗进行更新,更新后的跟踪窗一方面用于修正当前跟踪窗口的尺寸从而减小尺度定位偏差;另一方面,在下一帧跟踪中用来决定 Mean Shift 迭代过程中样本的数量. 这样,系统就能很好地适应目标尺度的变化,克服了固定窗的局限性. 算法步骤如下^[8]:

(1) 初始化帧数 $n = 0$, 变框系数 $k = 0$, 设定跟踪窗口尺度更新间隔帧数 N ;

(2) 读取一帧图像, $n = n + 1$, 如果 $n = 1$, 手动画定跟在窗口, 并计算初始框高 H 和宽 W , 否则进入下一步;

(3) 如果帧数 n 是 N 的整数倍时, 进入步骤(4), 否则 $k = 0$, 进入步骤(5);

(4) 对第 n 帧图像和当前帧图像的跟踪窗口对应区域分别提取角点, 并按上述方法进行匹配由式(9)得到变框系数 k ;

(5) 按式(10)和(11)更新跟踪窗的高和宽. 若被跟踪视频未结束返回步骤(2); 否则结束算法.

4 实验结果

文中提出的算法对目标尺寸的增大和减小都有较好的跟踪效果. 选取视频作为实验对象, 跟踪窗口都选择矩形窗, 且初始跟踪窗口都在第一帧手动选择.

实验证明文中方法在跟踪过程中根据目标的视觉细节信息的变化来更新跟踪窗口, 从而也更新了目标模型的核函数直方图, 得到了有效的跟踪结果. 文中方法能对逐渐增大的跟踪目标有很好的适应

性. 另外, 文中算法只对跟踪框内的范围内进行角点检测和匹配, 相应的计算量不大, 算法复杂度也不高.

图7给出了跟踪结果与目标的真实位置偏差的关系图. 偏差值如表1所示.

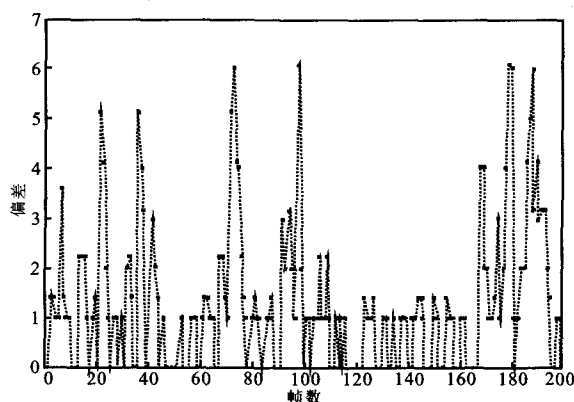


图7 跟踪结果与目标真实位置偏差关系图

表1 目标准确位置与跟踪结果偏差值比较

	最大偏差	最小偏差	平均偏差	平均迭代次数
方法一	7.23	0	3.37	1.45
方法二	跟踪丢失	1.36	4.52	1.52
方法三	6.08	0	1.45	1.48

从图7和表1中可以看出, 文中算法的跟踪结果与真实位置之间的偏差很小, 初始跟踪窗的长和宽分别为42和26, 平均偏差1.45带来的误差较小, 实验结果表明文中的改进算法跟踪的准确性较高, 能满足实时跟踪的准确性要求. 这里仅给出了运动的车为实验对象的结果, 大量实验结果表明文中算法对各种刚体实验对象都有较好的跟踪结果.

对分辨率为 $M \times N$ 的图像, 其算法复杂度为 $O(MN)$, 相当于进行 MN 次加法, 计算复杂度低. 实验测试中, 对连续的400帧 352×240 的彩色图像进行了测试, 总耗时为17s, 处理速度为24帧/s, 能达到实时跟踪的要求. 由于两帧之间目标的大小变化并不会十分明显, 实验中每隔5帧进行一次变框, 也避免了跟踪窗大小更新过频产生的误差问题, 同时也提高了算法的效率.

5 结束语

文中针对刚体目标跟踪, 对角点匹配的方法进行了改进, 利用前后帧匹配角点建立仿射模型, 用最小二乘法计算出仿射变化系数, 用该系数更新跟踪窗口的大小. 实验结果表明, 改进的跟踪算法对目标大小的变化有较好的适应性.

文中在检测得到的角点中有部分是背景上的非

目标特征点,这些点的存在也会对仿射模型的准确性产生影响.从检测得到的角点中去除这部分角点.

参考文献:

- [1] Comaniciu D, Ramesh V, Meer P. Real - Time tracking of non - rigid objects using mean shift[C]// IEEE International Proceeding on Computer Vision and Pattern Recognition. Stoughton: Printing House, 2000(2):142 - 149.
- [2] Comaniciu D, Ramesh V, Meer P. Kernel - based object tracking[J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2003, 25(5):64 - 575.
- [3] Bradski G R. Real time face and object tracking as a component of a perceptual for user interface[C]// IEEE Workshop on Applications of Computer Vision. Princeton, 1998:214 - 219.
- [4] 钱惠敏,茅耀斌,王执铨. 自动选择跟踪窗尺度的 Mean - Shift 算法[J]. 中国图像图形学报,2005,12(2):245 - 249.
- [5] 彭宁嵩,杨杰,刘志,等. Mean - Shift 跟踪算法中核函数

带宽的自动选取[J]. 软件学报,2005,16(9):1542 - 1550.

- [6] 王长军,朱善安. 融合梯度特征的灰度目标跟踪[J]. 微电子学与计算机,2009,26(2):69 - 71.
- [7] 柯海峰,刘伟宁. 基于变形模板技术的目标轮廓跟踪算法[J]. 微电子学与计算机,2008, 25(6):197 - 200.
- [8] Comaniciu D, Meer P. Mean shift: a robust approach toward feature space analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999, 24(5): 603 - 619.

作者简介:

何小城 男,(1974 -),博士后. 研究方向为视频处理和分析.

黄 凯 男,(1982 -),硕士研究生. 研究方向为目标检测和跟踪.

谭毅华 男,(1975 -),博士,副教授. 研究方向为遥感图像处理和分析、计算机视觉.

田金文 男,(1960 -),博士,教授. 研究方向为图像处理和分、制导与导航.

(上接第 68 页)

延时优化的影响,给出了理论上最佳的优化方案,并结合实际的物理设计条件得出了切实可行的优化方案,进而提出了优化算法. 经过证明,所给出的算法可有效的解决长互联线的延时问题.

参考文献:

- [1] 郝冬艳,张明,郑伟. 低功耗 VLSI 芯片的设计方法[J]. 微电子学与计算机,2007,24(6):137 - 142.
- [2] 王伟,冯哲,侯立刚,等. 深亚微米 VLSI 物理设计中天线效应的预防及修复[J]. 微电子学与计算机,2007,24(8):42 - 48.
- [3] Jan M Rabaey, Anantha chandrakasan, Borivoje Nikolic.

Digital integrated circuits - a design perspective[M]. 2nd ed. Beijing: Publishing House of Electronics Industry, 2004.

- [4] David A Hodges, Horace G Jackson, Resve A Saleh. Analysis and design of digital integrated circuit - In deep submicron technology[M]. 3rd ed. Beijing: Publishing House of Electronics Industry, 2005.
- [5] E v. The transient response of damped liner networks with particular regard to wideband amplifiers[J]. Journal of Applied Physics, 1948: 55 - 63.

作者简介:

孔 昕 男,(1985 -),硕士研究生. 研究方向为集成电路与系统集成、VLSI 物理设计.