

# 分块快速压缩追踪算法<sup>①</sup>

张彬彬, 王福龙

(广东工业大学 应用数学学院, 广州 510520)

**摘要:** FCT 作为一种新的追踪算法, 具有简单、高效、实时的优点, 但是该算法依旧存在缺点. 在 FCT 中, 由于压缩测量矩阵的稀疏性, 忽略了样本的空间信息, 使得提取的特征不能准确的表征样本; 当追踪错误时没有补救措施. 本文提出一种改进的快速压缩追踪算法, 该方法充分利用样本图像的空间信息, 分块提取样本的 Haar-like 特征; 利用目标运动估计法矫正分类错误时追踪到的目标. 通过调整压缩测量矩阵中行向量的稀疏度以及朴素贝叶斯分类器的阈值可以实现目标的准确追踪. 实验结果表明, 与快速压缩追踪算法(FCT)相比, 本文改进后的算法, 无论是在追踪相似度、追踪成功率还是主观视觉效果上都有所提高.

**关键词:** 快速压缩追踪(FCT); Haar-like 特征; 压缩感知; 朴素贝叶斯分类器; 稀疏性

## Block Fast Compressive Tracking Algorithm

ZHANG Bin-Bin, WANG Fu-Long

(Department of Applied Mathematics, Guangdong University of Technology, Guangzhou 510520, China)

**Abstract:** FCT is a simple yet effective and efficient tracking algorithm, despite much success has been demonstrated, numerous issues remain to be addressed. In FCT, because of the sparsity of the compression measurement matrix, the spatial information of the sample is neglected, so the feature cannot represent the tracking target correctly and there is no remedy when tracking error. In this paper, we propose an improved fast compressive tracking algorithm considering the sample space information and extracts generalized Haar-like features randomly in block; target motion estimation method is used to correct target location, as the classifier is wrong. Adjusting the sparse degree of vector in compression measurement matrix and threshold of naive Bayes classifier can realize accurate target tracking. The experimental results show that compared with FCT, the improved algorithm achieves much better results in terms of both similarity and success rate and subjective visual perception.

**Key words:** fast compressive tracking(FCT); Haar-like feature; compressive sensing; naive Bayes classifier; sparse

## 1 引言

近年来, 针对动态追踪问题提出很多有效的算法, 已被广泛接受的一种方法是低维子空间追踪学习法(例如基于特征子空间的算法<sup>[1,2]</sup>). 该算法的数据相互依赖, 且在特征分解时, 计算复杂度会显著增加. 当有噪声或者错误样本时, 噪声或者错误样本会改变特征空间的基, 导致追踪目标渐渐漂移. 另一种成功的方法是从高维空间提取具有识别力的目标特征. 该方法把动态追踪问题转化为二值分类问题, 即把目标从背景中分离出来. 也有一些算法将两者结合, 把高维

特征投影到低维空间. 压缩感知(CS)<sup>[3]</sup>理论已经证明: 当把高维特征投影到低维空间时, 在一定条件下低维空间的特征包含了高维特征的足够信息. 因此可以从压缩后的信息高概率的恢复出原始信息, 并保证压缩后的信息是相互独立的.

尽管已经有很多高效的动态追踪算法, 但光照变化、物体形变、环境复杂、动态模糊等因素容易引起外观模型变化. 这些问题的存在使得建立一种高效的追踪算法仍旧是一项艰巨且富有挑战性的任务.

CT<sup>[4]</sup>算法对目标的搜索只局限在帧间, 忽略了样

<sup>①</sup> 收稿时间:2015-08-20;收到修改稿时间:2015-10-19

本在当前帧内的空间信息. 文献[5]的算法首先在 CT<sup>[4]</sup>的基础上对 Haar-like 特征增加权重, 选出具有相对分辨度的特征, 其次通过协同训练算法, 加入样本的 LBP 特征, 最后采用与 CT<sup>[4]</sup>相同的更新以及追踪策略追踪目标.

在不考虑样本的空间信息时, 无论是 Haar-like 特征还是 LBP 特征, 都只能代表样本的一般特征, 区分度小. 并且在追踪错误时都没有补救措施<sup>[4,5]</sup>, 以至于误差不断积累, 甚至导致追踪失败.

为了进一步提高追踪效果, 本文针对 FCT<sup>[6]</sup>模型不能充分利用样本图像空间信息以及充分考虑分类错误的情况, 采用分块以及目标运动估计法实现更好的目标追踪.

## 2 快速压缩追踪算法(FCT)

快速压缩追踪算法(FCT<sup>[6]</sup>)通过非自适应的随机投影保护原始像素空间的结构. FCT 首先通过满足有限等距性质(RIP)的稀疏测量矩阵把 Haar-like 特征投影到低维空间; 然后, 根据朴素贝叶斯分类器把在压缩域的追踪问题转化为二值分类问题, 分类器最大响应值所对应的样本就是当前帧追踪到的目标.

### 2.1 提取特征

压缩感知指出大多数情况下, 信息是冗余的<sup>[2]</sup>, 即信号在某一变换域是稀疏的或者说是可压缩的, 那么就可用一个与变换基不相关的观测矩阵将变换所得高维信号投影到一个低维空间, 然后通过求解一个优化问题就可以从这些少量的投影中高概率的重构出原信号. 可以证明这样的投影包含了重构信号的足够信息(如图 1 所示).

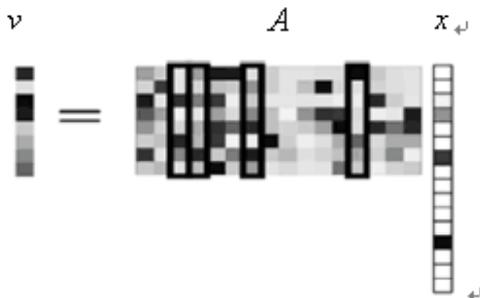


图 1  $x$  是信号的稀疏表示,  $A$  是 CS 信息算子,  $v$  是压缩后的信息

为了保证算法的收敛性, 即使得  $k$  稀疏信号  $x$  映

射到  $v$  时信息得到保护, 假设  $x_1$  和  $x_2$  都是  $k$  稀疏信号, 并且由相同的  $k$  个基向量线性组成, 则

$$(1-\varepsilon)\|x_1-x_2\|_{l_2}^2 \leq \|Rx_1-Rx_2\|_{l_2}^2 \leq (1+\varepsilon)\|x_1-x_2\|_{l_2}^2 \quad (1)$$

FCT<sup>[6]</sup>选择的压缩测量矩阵是非常稀疏矩阵  $R$ <sup>[5]</sup>, 其定义如下:

$$r_{i,j} = \sqrt{\rho} \begin{cases} +1, & p = 1/2\rho \\ 0, & p = 1 - 1/\rho \\ -1, & p = 1/2\rho \end{cases} \quad (2)$$

令  $p = o(m) = \frac{m}{a \log 10} = \frac{m}{10a} \sim \frac{m}{6a}$ , 其中  $a$  是一个常数,  $m$  在  $10^6 \sim 10^{10}$  之间. 矩阵  $R$  的每一行只有  $c = (\frac{1}{2\rho} + \frac{1}{2\rho}) \times m$  个非零值, 只需要计算和保存这些非零元素的位置和值.

FCT<sup>[6]</sup>提取样本的 Haar-Like 特征. Haar-Like 特征指的是样本中随机选择的矩形框中的像素和(如图 3 所示). 不同尺度的矩形窗口对样本卷积得到向量  $x$ , 因此  $x$  的维度很高, 在  $10^6 \sim 10^{10}$  之间. 测量矩阵  $R$  把高维的  $x$  投影到低维的  $v$ , 避免了维数灾难(如图 2 所示).

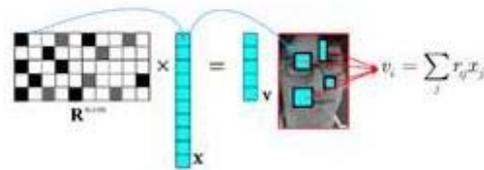


图 2 把高维的信号  $x$  投影到低维的  $v$ . 在矩阵  $R^{m \times n}$  中, 黑色、灰色、白色矩形框分别代表  $-1$ 、 $+1$ 、 $0$

### 2.2 样本分类

FCT<sup>[6]</sup> 根据朴素贝叶斯分类器把追踪问题转化为二值分类问题:

$$H(v) = \log \left( \frac{\prod_{i=1}^n p(v_i | y=1) p(y=1)}{\prod_{i=1}^n p(v_i | y=0) p(y=0)} \right) \quad (3)$$

$$= \sum_{i=1}^n \log \left( \frac{p(v_i | y=1)}{p(v_i | y=0)} \right)$$

其中  $p(y=1) = p(y=0)$ ,  $y \in \{0,1\}$  是一个二值变量, 是样本的标记,  $y=1$  代表正样本,  $y=0$  代表负样本. Diaconis 和 Freedman<sup>[8]</sup>证明高维空间投影到低维空间时几乎都满足高斯分布, 因此, 条件分布  $p(v_i | y=1)$

和  $p(v_i | y = 0)$  满足高斯分布  $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$ , 即

$$\begin{cases} p(v_i | y = 1) \sim N(\mu_i^1, \sigma_i^1) \\ p(v_i | y = 0) \sim N(\mu_i^0, \sigma_i^0) \end{cases} \quad (4)$$

分类器最大响应值  $\max(H(v))$  所对应的样本就是追踪到的目标. 找到当前帧的目标后, 根据 coarse-to-fine 采样策略, 同时采集正负样本用来更新朴素贝叶斯分类器, 为下一帧追踪做准备. 以上尺度参数递增的更新如下:

$$\begin{aligned} \mu_i^1 &\leftarrow \lambda \mu_i^1 + (1-\lambda) \mu^1 \\ \sigma_i^1 &\leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2} \end{aligned} \quad (5)$$

其中  $\lambda > 0$  是一个学习参数, 用来权衡前一帧和当前帧. 当目标外观变化比较快时,  $\lambda$  要小一些. 其中

$$\sigma^1 = \sqrt{\frac{1}{n} \sum_{k=0}^{n-1} \sum_{y=1} v_i(k) - \mu^1}^2} \quad (6)$$

$$\mu^1 = \frac{1}{n} \sum_{k=0}^{n-1} \sum_{y=1} v_i(k)$$

### 2.3 算法流程

输入: 第  $t$  帧视频;

Step1. 设置初始参数;

Step2. 首先采样得到正样本集:

$$D^c = \{z \mid \|I(z) - I_{t-1}\| < r_c\},$$

其中  $I_{t-1}$  是  $t-1$  帧中目标的位置,  $\Delta_c$  为采样步长. 最后得到低维的特征值  $V(z)$ ;

Step3. 根据(3)式, 计算每一个样本对朴素贝叶斯分类器的响应值.  $\max(H(v))$  所对应样本就是找到的粗略目标  $I_t^r$ ;

Step4. 首先在第3步追踪到的粗略位置  $I_t^r$  附近采样, 得到正样本集:

$$D^f = \{z \mid \|I(z) - I_t^r\| < r_f\},$$

其中步长为  $\Delta_f$ . 最后得到低维的特征值  $V(z)$ ;

Step5. 根据(3)式, 计算每一个样本对朴素贝叶斯分类器的响应值,  $\max(H(v))$  所对应样本就是找到的当前帧的准确目标  $I_t$ ;

Step6. 根据上一步得到的  $I_t$  以及 Coarse-to-fine 采样策略采样得到正负样本集:

$$D^a = \{z \mid \|I(z) - I_t\| < \alpha\},$$

$$D^{\xi, \beta} = \{z \mid \xi < \|I(z) - I_t\| < \beta\},$$

并且提取正负样本的特征, 其中  $\alpha < \xi < \beta$ ;

Step7. 根据(5)式更新分类器参数;

输出: 追踪到的目标  $I_t$  和更新后的分类器参数.

## 3 算法改进

动态目标追踪结果经常会受到姿势变化、光照、运动模糊、噪声等的影响, 因此建立高效的目标外观模型对动态目标追踪起到至关重要的作用.

### 3.1 划分区域增加样本的空间信息

FCT 由稀疏压缩测量矩阵提取样本的 Haar-like 特征(如图3所示). 由于矩阵的稀疏性使得选取的特征不仅数量少, 而且随机. 在最不理想的情况下, 矩形框的位置和大小都很接近或者完全相同, 那么获得的特征就不能很好的代表样本, 从而导致追踪不准确(如图5(b)所示).

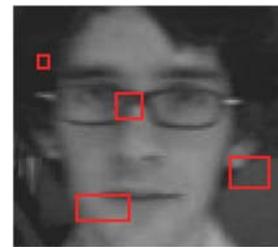


图3 FCT 提取特征的方法: 提取样本的 Haar-Like 特征. 首先在每一个样本区域随机得到 2 到 4 个矩形框, 每一个矩形框的位置和大小是随机的, 其次根据积分图得到每一个矩形框内的像素和.

为了解决这个问题, 本文提出一种分块快速压缩追踪算法, 增加样本的空间信息. 首先把  $N_1 \times N_2$  大小的样本分为  $N_1/C \times N_2$  大小的  $C$  个的互不相交的区域(如图4); 然后在每个区域内随机选择一个矩形框, 提取样本的 Haar-like 特征(如图4), 使得即使压缩矩阵在最稀疏的情况下, 所提取的特征也能分散在整个样本区域, 准确的表征要追踪的目标. 其中  $C$  为稀疏压缩测量矩阵的每一行向量中非零元素的个数, 是一个在矩形框最小个数和最大个数之间的随机数.

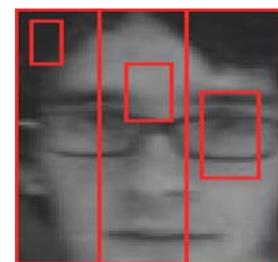


图4 样本被分割成随机个数的互不相交的区域

本文改进后的算法,不同的视频序列选择不同的目标框最大和最小个数.

### 3.2 矫正追踪目标

FCT 算法把追踪问题转化为二值分类问题.当朴素贝叶斯分类器的最大响应值  $\max(H(v))$  小于等于零时,追踪到的目标更应该被认为是负样本,而不是作为正样本成为追踪到的目标.随着误差的积累,FCT 算法会导致追踪不准确等问题(如图 5(b)所示).

为解决该问题,本文提出:当朴素贝叶斯分类器的最大响应值  $\max(H(v))$  小于或等于设定的阈值  $\text{threshold}$  时,对当前帧的目标位置矫正.

根据第  $i-1$  帧和  $i-4$  帧的目标位置差估计目标的平均移动速度  $\vec{v}$

$$\vec{v} = \frac{I_{i-1} - I_{i-4}}{3} \quad (7)$$

其中  $I_{i-1}$  和  $I_{i-4}$  分别是第  $i-1$  帧和  $i-4$  帧的目标位置.

当前帧的目标位置  $I_i$  等于前一帧的目标位置加上速度:

$$I_i = I_{i-\Delta t} + \vec{v} \Delta t \quad (8)$$

其中  $\Delta t$  等于 1.  $\Delta t$  过大过小都会引入误差.  $\Delta t$  等于 1 是经过调试实验,根据实验效果确定的,是非自适应的.

## 4 实验结果

实验基于 matlab2014a 和 C++ 环境.

FCT<sup>[6]</sup>中压缩测量矩阵过于稀疏以及分类的不准确性,使得追踪误差不断积累,最终导致追踪漂移.在视频序列 david 中,追踪目标运动缓慢,且外观变化缓慢.在不考虑样本的空间信息以及分类错误情况时,FCT<sup>[6]</sup>虽然能追踪到目标,但是追踪误差不断的积累,以至于 FCT<sup>[6]</sup>算法追踪准确率不高,视觉效果差(如图 5(b)所示).本文改进后的算法一直能稳定的追踪到目标(如图 5(a)所示),提高了追踪的准确率以及视觉效果.

FCT<sup>[6]</sup>中压缩测量矩阵对不同的视频序列非自适应的使用相同的参数,比如矩形框的个数最大为 4,最小为 2.在视频序列 biker 中,FCT<sup>[6]</sup>能准确的追踪到目标(如图 6(b)所示),但在视频序列 chasing 中,目标快速运动,并且伴随着明显的姿态变化.FCT<sup>[6]</sup>算法在目标发生明显姿态变化后,只能追踪到目标的部分内容,以至于在采集和更新样本信息时,会不断地加入错误的样本,使得误差不断积累.本文改进的算法,通过对样本

分块和改变相应的参数以及不断矫正目标,提升了信息对样本的表达能力和追踪的精度(如图 6(a)、7(a)所示).

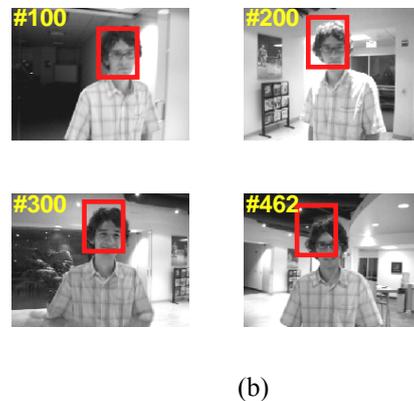


图 5 (a)是本文算法追踪到的目标,(b)是 FCT 追踪到的目标.且目标框的最大个数为 7,最小个数为 5,  $\text{threshold}_c = 10$ ,  $\text{threshold}_f = 0$ .

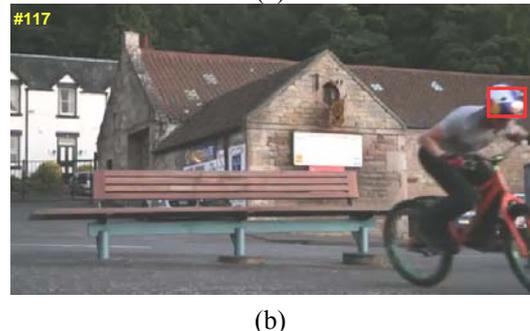
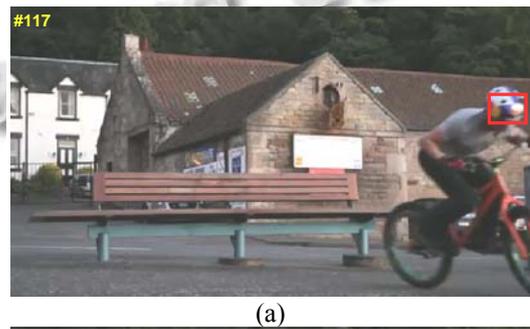


图 6 (a)是本文算法追踪的结果,(b)是 FCT 追踪的结果.

果. 其中目标框的最大个数为 5, 最小个数为 3,  
 $threshold_c = 20$ ,  $threshold_f = 0$

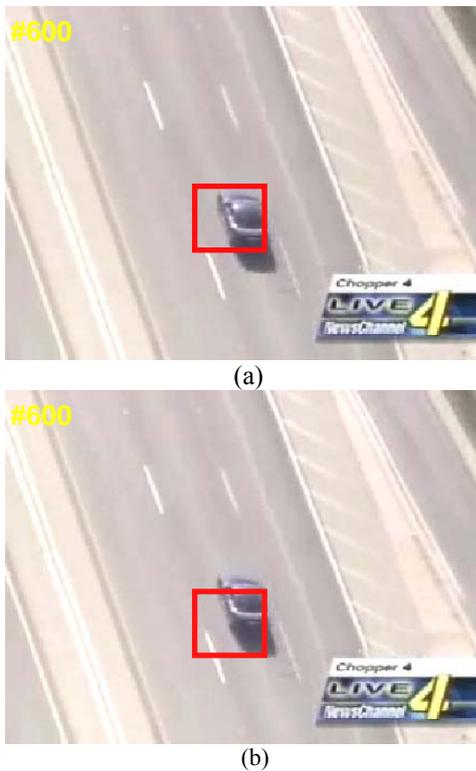


图 7 (a)是本文算法追踪的结果, (b)是 FCT 追踪的结果. 其中目标框的最大个数为 5, 最小个数为 2,  
 $threshold_c = 20$ ,  $threshold_f = 0$ .

本文改进后的算法, 加入了样本的时间和空间信息, 提高了压缩后信息对样本的表达能. FCT<sup>[6]</sup>对目标的弱表达能力, 使得在相同的视频序列中改变被追踪的目标时, FCT<sup>[6]</sup>追踪错误(如图 8(b)所示). 本文改进后的算法, 总能找到相应的分块和矫正参数, 实现准确追踪(如图 8(a)). 在视频序列 david 中, 把追踪的目标改为衣领部分, FCT 算法在 300 帧之后目标漂移, 追踪失败, 本文改进后的算法却能稳定的追踪到目标.

在本文算法中, 阈值以及分块个数的选择是自适应的. 阈值以及分块个数的选择是根据实验效果调试得到的, 其中要求大于, 逐步提高分类的准确性.

定义追踪的平均相识度为  $\frac{1}{n-1} \sum B \cap C$  平均追踪

成功率  $\frac{1}{n-1} \sum \frac{B \cap C}{B \cup C}$ , 本文改进的算法和 FCT<sup>[6]</sup>算法的平均相识度和平均追踪成功率如表 1 和表 2.

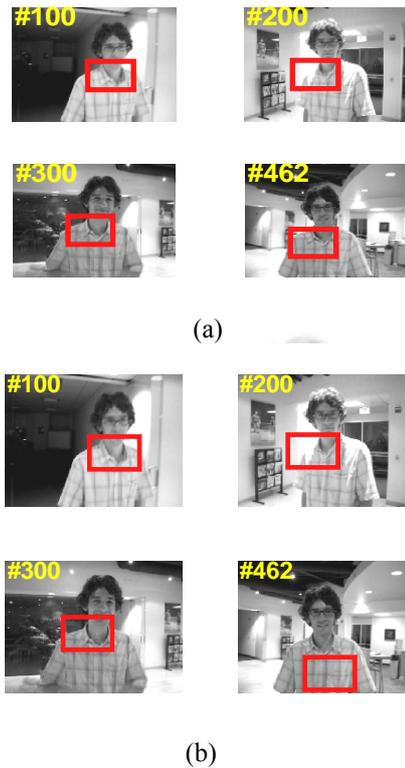


图 8 追踪的目标是衣领. (a)是本文算法追踪到的目标, (b)是 FCT<sup>[6]</sup>追踪到的目标. 其中目标框的最大个数为 7, 最小个数为 4,  $threshold_c = 5$ ,  $threshold_f = -10$ .

表 1 本文算法

|         | 平均相识度    | 平均追踪成功率 |
|---------|----------|---------|
| biker   | 186.2821 | 0.8267  |
| bolt    | 188.7440 | 0.8144  |
| chasing | 171.4900 | 0.8811  |

表 2 FCT 算法

|         | 平均相识度    | 平均追踪成功率 |
|---------|----------|---------|
| biker   | 185.2308 | 0.8229  |
| bolt    | 188.6416 | 0.8200  |
| chasing | 170.9567 | 0.8772  |

### 5 结语

本文算法在视频目标追踪上, 对比 FCT<sup>[6]</sup>, 在提取样本 Haar-like 特征时, 加入样本的空间信息, 增强了特征对样本的表达能; 当朴素贝叶斯分类器分类不准确时, 通过增加阈值, 加入样本时间信息: 根据临近帧得到的目标移动速度矫正当前帧目标. 实验表明, 本文提出的分块快速压缩追踪, 无论是在追踪相似度、追踪成功率还是视觉效果上都有所提高.

## 参考文献

- 1 Ross D, Lim J, Lin R, Yang MH. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 2008, 77(1): 25-141.
- 2 Ho J, Lee K, Yang MH, Kriegman D. Visual tracking using learned linear subspaces. *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 2004, (1). 1-178.
- 3 Donoho D. Compressed sensing. *IEEE Trans. on Information Theory*, 2006, 52: 1289-1306.
- 4 Zhang KH, Zhang L, Yang MH. Real-time compressive tracking. *Proc. of the 12th European Conference on Computer Vision*. 2012, 3. 866-879.
- 5 Zhu JZ, Ma Y, Qin QQ, Zheng C. Adaptive weighted real-time compressive tracking. *IET Computer Vision* Received, 2014, 8: 740-752.
- 6 Zhang KH, Zhang L, Yang MH. Fast compressive tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014, 36(10): 2002-2015.

[www.c-s-a.org.cn](http://www.c-s-a.org.cn)

[www.c-s-a.org.cn](http://www.c-s-a.org.cn)