上海大学无人艇工程研究院

——环境感知组

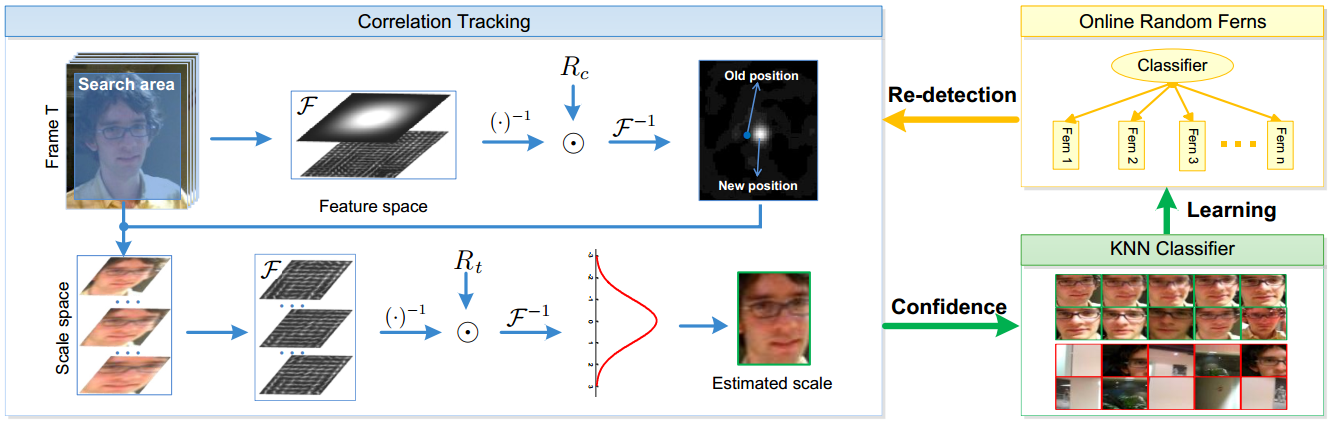
# Long-term Correlation Tracking--LCT

作者：**Chao Ma** Xiaokang Yang Chongyang Zhang Ming-Hsuan Yang上海交大

主页：<https://sites.google.com/site/chaoma99/cvpr15_tracking>

出处：2015年CVPR

源码：matlab



注：**加粗**的作者为重点关注研究者

图注：本算法的核心示意图

Date：2017.03.27

## 版本更新记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日 期** | **更新人** | **主要更新内容描述** | **版本号** |
| 2017年03月28日 | 陈加宏 | 完成大致框架搭建 | V1.0.0 |
| 2017年09月26日 | 陈加宏 | 完成算法细节的总结 | V1.0.1 |

目 录

[Long-term Correlation Tracking--LCT 1](#_Toc494212891)

[版本更新记录 2](#_Toc494212892)

[1、概述 3](#_Toc494212893)

[1.1 前言——研究背景及意义、该领域存在的问题 3](#_Toc494212894)

[1.2 创新点——本文算法要解决的问题以及具体的解决方法 5](#_Toc494212895)

[2、细节 5](#_Toc494212896)

[2.1 主要流程 5](#_Toc494212897)

[2.2 数学模型 6](#_Toc494212898)

[2.3 模型求解 7](#_Toc494212899)

[3、实验 7](#_Toc494212900)

[3.1 代码框架 7](#_Toc494212901)

[3.2 实验结果及分析 7](#_Toc494212902)

[3.3 优缺点总结 8](#_Toc494212903)

[3.4 今后工作 8](#_Toc494212904)

## 1、概述

该部分主要讲述的是在视觉目标跟踪领域相对短时跟踪short-term的长时long-term目标跟踪，两者的主要区别在于短时目标跟踪在跟踪失败后没有再检测模块，反之，长时间目标跟踪有跟踪失败后再检测模块。主流的视觉目标跟踪研究领域是在短时目标跟踪上，但是随着对目标跟踪鲁棒性、精度的要求的提升，特别是对跟踪鲁棒性的要求的提高，对长时目标跟踪的研究热情也日趋高涨，本文主要讲解在目标跟踪领域表现优秀的国内上海交大的学者马超的LCT，还是在相关滤波框架内同时加入了再检测模块。

### 1.1 前言——研究背景及意义、该领域存在的问题

以前提到的很多CF算法，也包括VOT竞赛，都是针对short-term的跟踪问题，即短期跟踪，我们只关注短期内(如100~500帧)跟踪是否准确。但在实际应用场合，我们希望正确跟踪时间长一点，如几分钟或十几分钟，如果直接让前面介绍的那些看起来还不错的short-term的算法，如KCF, Staple等跟踪较长一段时间会怎么样呢？

很长时间过去了，这些算法或许还跟的很好，或许早已经跟丢了。那我们会有疑问：前面不是说这些算法综合性能都算是顶尖水平了，怎么还会跟丢呢？原因大概可以总结为以下两点：

1、算法没有100%。自从有了深度学习方法，很多计算机视觉问题都有了很大突破，然而，在IMAGENET上的图像分类或LFW的人脸识别准确率还是没有到达100%，再退一步说，即使到了100%，也只能说在某个数据集上达到了100%，换个数据集可能就不行了，再退一步说，即使在某几个数据集都达到了100%，实际应用中可能又不行了，因为现实情况非常复杂，有很多不可控因素，不是几个数据集就能囊括的。跟踪算法也是算法，算法就是算法，没有最好，只有更好。

2、错误累积。即使VOT2016上排第一的C-COT，各项指标离满分还是存在着很大距离的，任何一个算法都不是100%靠谱，如CF对快速运动比较弱，某个算法遇到它不擅长的难题就容易跟丢跑飞。即使某个算法在它擅长的情况下，每次检测的结果也都有一定误差，这个误差会随着跟踪时间慢慢积累，量变到质变，最终跟丢。

这就是一开始提到的，tracker要配合detecter一起用，隔一段时间重新初始化一次tracker，跟丢的找回来，有误差的清除误差，这样才能正常跟踪下去。

Long-term就是希望tracker能长期正确跟踪，我们分析了前面介绍的方法不适合这种应用场合，必须是short-term tracker + detecter配合才能实现正确的长期跟踪。那如果应用场合没有detecter，怎么才能实现Long-term呢？简单啊，给tracker配一个detecter就可以了。用一句话介绍Long-term，就是给普通tracker配一个detecter，在发现跟踪出错的时候调用自带detecter重新检测并矫正tracker。

对于tracker有以下要求：

1、这个tracker必须尽可能地好，不然跟一帧重新检测一次，肯定是不行的；

2、detecter必须尽可能简单，它也是跟踪算法的一部分，而且检测算法是整帧进行检测，太慢了影响速度；

3、detecter还必须尽可能好，检测错误了还不如不检测；

4、tracker必须知道什么时候跟丢了，然后再去调用detecter去重新检测，保证以跟踪为主，提高帧率。

总结起来就是跟踪器和检测器都最好又快又准，同时需要跟踪器感知到什么时候来激活检测器的使用，以上就是Long-term的基本情况，忘了介绍检测算法的训练样本：负样本要多少有多少，跟踪如果正确，那跟踪目标就是正样本，正样本数量有限，正确跟踪帧数越多，正样本数量越多。为了方便，检测算法也可以用online learning，每一帧更新一次。

### 1.2 创新点——本文算法要解决的问题以及具体的解决方法

这篇文献名为长时间的相关跟踪，解决的主要问题是目标形变、遮挡和失踪等情况。首先将跟踪器**分为定位跟踪和尺度跟踪**两个方面。为了提高算法的鲁棒性，提出一种结合上下文信息的目标定位方法，用来确定目标的位置。目标的尺度则是根据对尺度的仿射变换，再利用相关滤波器对其作用，响应越大的尺度就是当前帧的尺度。这篇文章比较重要是提出了目标被完全遮挡或者目标被跟丢后如何**再检测**重新跟上目标的方法。

## 2、细节

VOT2015中最亮眼的相关滤波方法当属Martin Danelljan大牛的SRDCF，主要思路：既然边界效应发生在边界附近，那就忽略所有移位样本的边界部分像素，或者说让边界附近滤波器系数为接近零。该部分主要讲述本文算法的核心细节，包括算法的主要流程、数学模型的建立以及模型的求解方法。要完全的理解跟踪算法必须从最基础的问题本质出发，借助数学模型对问题进行抽象，最后通过优化求解方法得到解决方案。

### 2.1 主要流程

先说说这个算法的思想来源。由于在跟踪领域内存在这样一个进退两难的问题：如果对训练样本的建模要求十分严格，这样的模型就不易受到噪声的影响而产生偏移和遮挡的影响，但是当目标有些许形变就会出现问题，模型自适应能力差；那么如果模型自适应能力强的话，它就更容易使目标漂移。**针对这种问题就可以采用两个回归模型来结合上下文信息和尺度信息来克服**。另外一个问题就是采样二义性问题，那么基于高斯加权标签的样本采样可以很好的克服这个问题。LCT跟踪算法的整体流程图：

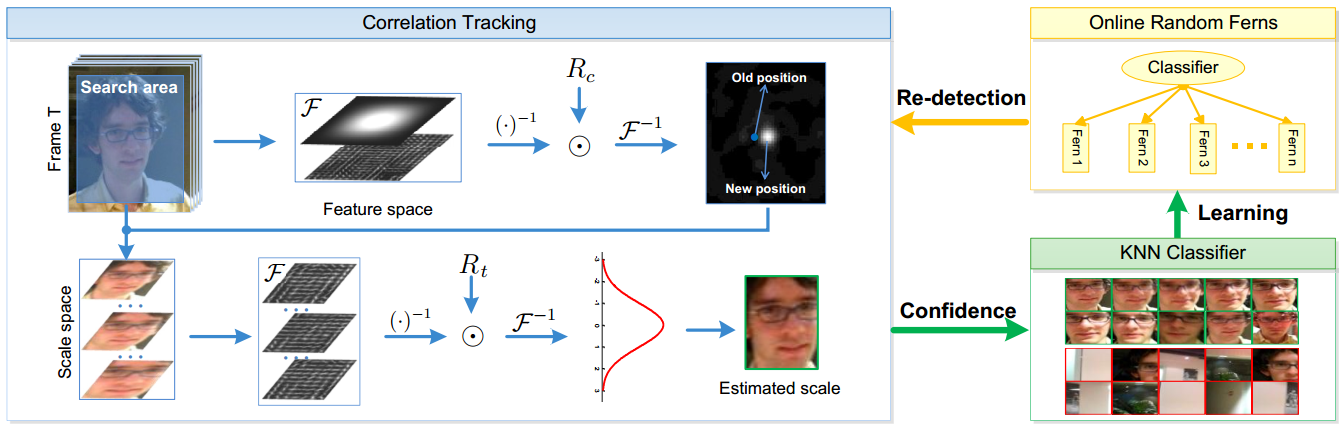


图2.7.2 LCT算法的运行流程图

整个跟踪器由三部分组成：目标定位、尺度估计、在线再检测模块。

再看跟踪器的目标定位和尺度估计方案：

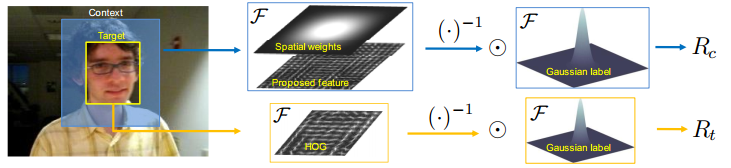


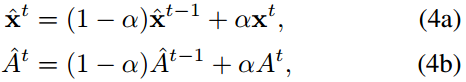
图2.7.3 目标定位和尺度估计的具体流程

### 2.2 数学模型

再次回到相关滤波跟踪器的优化函数：

在对偶空间中求解得：







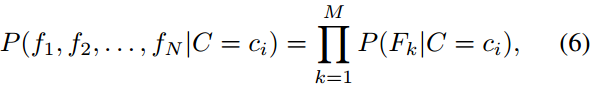
那么对于模板的更新，为了避免错误信息对模型的影响，要满足一定的条件才能进行模型的更新，即响应的最大值要大于某一阈值。





对于尺度更新问题，是采用一种类似于DSST的关于尺度框的仿射变换，然后再做一个响应，从而根据响应的大小选择响应最大的那个尺度为当前的目标的尺度。

接来下说明下检测器的构成：



### 2.3 模型求解

## 3、实验

该部分主要讲述算法实现代码的主要流程、实验环境及效果分析、算法优缺点的总结，最后提出后续可改进的方面。实验是检验真理的唯一标准，那么对实验结果详细的分析以及结合算法的原理对算法本质上的一些思考有利于之后研究工作的开展，也是今后工作的一个研究突破点。

### 3.1 代码框架

### 3.2 实验结果及分析

这篇文献主要关注的两个点就是1、如何在模型精确度和模型自适应性上做一个折中的选择；2、要实现长时间的目标跟踪就必须要有跟丢后重新检测的模块；针对前一个问题本文将跟踪过程分为定位和尺度确定两个跟踪子模块。对于目标位置的估计采用的特征是47个通道的特征（31个HOG特征和8个6乘6的强度直方图和8个变换）。对于尺度的确定类似于DSST，对目标框进行仿射变换后选取响应最大的框为当前帧目标的尺度。针对后一个问题提出了一种再检测模块的方案。

### 3.3 优缺点总结

### 3.4 今后工作