

tracking 模块

参数：

以下测试结果使用了控制变量法。

TrackerBoosting：

```
//set the parameters
cv::TrackerBoosting::Params params;
// params.iterationInit = 1;
// params.numClassifiers = 20;
// params.samplerOverlap = 5;
// params.samplerSearchFactor = 5;

//a Boosting tracker object
Ptr<TrackerBoosting> tracker = cv::TrackerBoosting::create(params);
```

成员名	类型	作用	测试
iterationInit	int	初始化迭代次数	测试时此参数对算法跟踪效果没有明显影响，但是数值比较大时会增加跟踪器初始化时间。设置值为50时，能感觉到秒级别的延迟。
numClassifiers	int	分类器数目	测试时此参数对算法跟踪效果没有明显影响，但是数值比较大时可能会导致内存错误，以至于初始化失败。
samplerOverlap	float	搜索区域	测试时调节此参数可以提高算法跟踪效果，并且对时间影响不大。
samplerSearchFactor	float	搜索区域	测试时调节增大此参数可以很大程度上提高算法跟踪效果，但是值过大时对时间及内存消耗影响很大，甚至导致死机。测试时设置值为50导致死机（8G内存占用率大约50%）。

TrackerCSRT：

```
//set parameters
cv::TrackerCSRT::Params params;
params.window_function = "kaiser"; // "hann", "cheb", "kaiser"
params.use_channel_weights = true;
params.use_color_names = true;
params.use_gray = true;
params.use_hog = true;
params.use_rgb = true;
params.use_segmentation = true;

//create the CSRT object
Ptr<TrackerCSRT> tracker = cv::TrackerCSRT::create(params);
```

成员名	类型	作用	测试
window_function	string	设置窗函数	可选项为："hann", "cheb", "kaiser"。在测试中三种窗函数表现没有明显差别。
use_channel_weights	bool	\	单独使用（设置为true）会报错。
use_color_names	bool	使用cn特征	单独使用无明显差别。
use_gray	bool	使用灰度特征	单独使用无明显差别，但速度最快。
use_hog	bool	使用hog特征	单独使用无明显差别。
use_rgb	bool	使用rgb特征	单独使用无明显差别。
use_segmentation	bool	\	单独使用（设置为true）会报错。
admm_iterations	int	\	值越大算法运行速度越慢，但是测试中，跟踪效果没有明显差别
background_ratio	int	\	测试中，值越大（测试值1，10，100）效果越好，但是提升不是很明显，同时运行速度稍微降低（不明显）。
cheb_attenuation	float	\	无明显作用。
filter_lr	float	\	值比较大时（测试值>1.5），跟踪框漂移严重，导致无法跟踪。
gsl_sigma	float	\	值比较大时（测试值>7.0），跟踪框漂移严重，导致无法跟踪。
histogram_bins	int	\	值比较大时（测试值100），会降低算法运行速度，对算法性能提升不明显。
histogram_lr	float	\	无明显作用。
hog_clip	float	\	无明显作用。
hog_orientations	float	\	无明显作用。
kaiser_alpha	float	\	无明显作用。
num_hog_channels_used	int	\	值比较大时算法效果降低（测试值>10）,值过大时（测试值100）会导致运行错误。

成员名	类型	作用	测试
number_of_scales	int	\	值较大时（测试值100）可以提升跟踪效果，但是帧率略微降低（21FPS，值为1时帧率44FPS）。
padding	float	\	值较小（测试值<1）可以提升算法运行速度和跟踪效果，值较大时（测试值>10）跟踪效果变差（跟踪框漂移）。
scale_lr	float	\	值过大（测试值>1）时会导致运行时程序崩溃。
scale_model_max_area	float	\	值较小（测试值<10）时会导致程序无法运行。
scale_sigma_factor	float	\	值较大（100）的跟踪效果不如较小（0.1）。
scale_step	float	\	设置测试值（0.1,10,100）时，代码无法运行。
template_size	float	\	值较小（测试值0.1）无法运行，值较大（测试值50,100）时可以提升算法运行速度，算法运行效果没有明显变化。

TrackerKCF：

```

//set parameters
cv::TrackerKCF::Params params;
// params.compress_feature = true;
// params.compressed_size = 10;
// params.pca_learning_rate = 0.9;
// params.resize = false;
// params.desc_pca = cv::TrackerKCF::CN;
// params.detect_thresh = 0.5;
// params.lambda = 0.2;
// params.sigma = 1.5;
// params.split_coeff = true;
// params.wrap_kernel = true;
// params.interp_factor = 10.5;
// params.max_patch_size = 100;
// params.output_sigma_factor = 0.4;

//create a KCF tracker
Ptr<TrackerKCF> tracker = cv::TrackerKCF::create(params);

```

成员名	类型	作用	测试
compress_feature	bool	启用PCA	无明显差别
compressed_size	int	降维维度	值过大时（测试值100），会导致内核错误以致运行时崩溃。
desc_npca	int	不采用PCA的特征描述符	可选值为：cv::TrackerKCF::CN、cv::TrackerKCF::GRAY、cv::TrackerKCF::CUSTOM。RGB图像使用cv::TrackerKCF::CN，否则运行出错。
desc_pca	int	采用PCA的特征描述符	可选值为：cv::TrackerKCF::CN、cv::TrackerKCF::GRAY、cv::TrackerKCF::CUSTOM。RGB图像使用cv::TrackerKCF::CN，否则运行出错。
detect_thresh	float	检测器的置信度阈值	适当调低（增大过检率）可以提高跟踪效果。
interp_factor	float	自适应线性差值因子	值过大时（测试值10.5）会导致算法基本失效，无法跟踪。
lambda	float	正则化参数	值过大时（测试值0.2以上）导致算法无法使用。
max_patch_size	int	roi区域大小	测试使用时，无明显差别，但是可以加快算法速度（测试值100）。
output_sigma_factor	float	空间带宽	为0时跟踪框迅速飘逸到视野外，值较大时（0.5以上）跟踪框飘逸严重导致无法跟踪。
pca_learning_rate	float	PCA学习率	测试使用时，无明显差别。
resize	bool	减小特征维度	测试使用时，无明显差别。
sigma	float	高斯核带宽	测试使用时，无明显差别。
split_coeff	bool	将训练系数转化为两个矩阵	测试使用时，无明显差别。
wrap_kernel	bool		测试使用时，无明显差别。

TrackerMedianFlow:

```

//set parameters
cv::TrackerMedianFlow::Params params;
params.maxLevel = 10;
params.maxMedianLengthOfDisplacementDifference = 80.0;
params.pointsInGrid = 10;
params.winSize = cv::Size(20,20);
params.winSizeNCC = cv::Size(20,20);

//create a MedianFlow tracker
Ptr<TrackerMedianFlow> tracker = cv::TrackerMedianFlow::create(params);

```

成员名	类型	作用	测试
maxLevel	int	光流法中金字塔层数	测试中调节此参数无明显影响。
maxMedianLengthOfDisplacementDifference	double	丢失目标的惩罚	测试中调节此参数无明显影响。但是值过低时（测试中小于1.0）算法无效。
pointsInGrid	int	\	测试中调节此参数无明显影响。但是值过大时（测试中100）算法帧率降低（6FPS）。
termCriteria	TermCriteria	光流法终止标准	\
winSize	cv::Size	光流法窗口	测试中调节此参数无明显影响。
winSizeNCC	cv::Size	NCC窗口	测试中调节此参数无明显影响。

TrackerMIL:

```
//set parameters
cv::TrackerMIL::Params params;
// params.featureSetNumFeatures = 1;
params.samplerInitInRadius = 5.0;
params.samplerInitMaxNegNum = 100;
params.samplerSearchWinSize = 10.0;
params.samplerTrackInRadius = 10.0;
params.samplerTrackMaxNegNum = 100;
params.samplerTrackMaxPosNum = 100;

//create a MIL tracker
Ptr<TrackerMIL> tracker = cv::TrackerMIL::create(params);
```

成员名	类型	作用	测试
featureSetNumFeatures	int	\	测试中使用了几组数据都无法使用
samplerInitInRadius	float	初始化过程的 正样本搜索半径	数值太小时（测试设置为1.0）无法追踪目标，算法基本失效；数值太大时（测试设置为10.0）严重影响算法耗时。设置一个合适的值（测试值为5.0）可以提升算法性能。
samplerInitMaxNegNum	int	初始化过程的 最大负样本数	值较大时（测试值100）可以提升跟踪效果，但是影响算法耗时（15FPS）；值较小时（测试值10）可以减少算法耗时（25FPS），但是会降低跟踪效果。
samplerSearchWinSize	float	搜索框大小	值较大时（测试值50.0）降低跟踪效果（跟踪框容易跑偏），同时增加耗时（7FPS）；值较低时（测试值10.0）跟踪效果比较好，帧率为17FPS；值过低时（测试5.0）算法基本失效。
samplerTrackInRadius	float	正样本搜索半径	测试时无明显影响。
samplerTrackMaxNegNum	int	最大负样本数	测试时无明显影响。
samplerTrackMaxPosNum	int	最大正样本数	测试时无明显影响。

TrackerMOSSE/TrackerTLD/TrackerGOTURN：

无可调参数。

耗时（帧率）：

以下各算法为默认参数。测试视频大小 768×576 ，一共794帧，初始化boundingBox为(500, 160, 525, 235)，取平均帧率：

算法	帧率/FPS	备注
TrackerBoosting	54.540891	与CSRT差不多
TrackerCSRT	38.112262	与Boosting差不多
TrackerKCF	369.234893	与MedianFlow差不多
TrackerMedianFlow	165.631738	与KCF差不多
TrackerMIL	13.997373	\
TrackerMOSSE	4036.727679	\
TrackerTLD	24.032678	\
TrackerGOTURN	\	\

说明：

上述时间具有参考价值。实际运行时间与boundingBox选取有关，与跟踪效果有关（有些跟踪器不能一直跟踪到目标）。

在同样视频不同boundingBox的另外一些测试中，Boosting与CSRT在耗时表现上差不多，并且基本能保证>30FPS；KCF与MedianFlow方法在耗时表现上也相差无几，并且基本能保证>100FPS，上表KCF达到300多帧率是因为大部分帧KCF没有找到目标。

跟踪效果：

算法	效果
TrackerBoosting	目标被遮挡时会将遮挡物当做目标；目标快速移动时容易跟错目标；boundingBox会一直存在。
TrackerCSRT	可以解决一些目标被遮挡的问题；但是容易跟错目标；跟踪过程中，boundingBox大小可以自动变化。
TrackerKCF	速度快；但是很容易丢失目标，且目标丢失后算法基本失效。
TrackerMedianFlow	目标被遮挡时会将遮挡物当做目标；目标运动过快时算法基本失效；boundingBox比目标大很多时算法基本失效；跟踪过程中，boundingBox大小可以自动变化；boundingBox会一直存在。
TrackerMIL	速度慢；目标被遮挡时会将遮挡物当做目标；boundingBox会一直存在。
TrackerMOSSE	速度很快；但是容易丢失目标，且目标丢失后算法基本失效。
TrackerTLD	容易跟错目标；跟踪过程中，boundingBox大小可以自动变化；boundingBox比目标大很多时算法基本失效；boundingBox会一直存在。
TrackerGOTURN	\

说明：

不同场景下同一算法的表现也有一定的差异。在满足时间要求的情况下，不同场景需要用不同的算法测试，因地制宜。