

# Tutorial of Using Visual Tracker Benchmark

Qiang Wang

National Laboratory of Pattern Recognition,  
Institute of Automation, Chinese Academy of Sciences  
No. 95, Zhongguancun East Road, Beijing 100190, P. R. China

November 29, 2016

The tutorial is a step by step tutorial to tell you how to use Visual Tracker Benchmark<sup>1</sup> toolkit. If you are beginner in the filed of visual tracking. I think you should read the paper to get a comprehensive perspective of this filed.

## 1 Download

At the beginning, you should download all need files to save time in the after.

### 1.1 Download codebase

Go to the official website of **OTB**. Download the codebase.

How to get the tracker benchmark codebase.

The tracker codes used in this benchmark can be download ([tracker\\_benchmark\\_v1.0.zip](#) 229MB).  
If you suffer from slow download speed, try [this file](#) on Google Drive.  
The benchmark results using the above code is available also : [tracker\\_benchmark\\_v1.0\\_results.zip](#) (222MB, or Google Drive version).

The results zip-file needs to be unzipped in the 'tracker\_benchmark\_v1.0' directory.

- We are building a new Python-based test suite [https://github.com/jwilm/tracker\\_benchmark](https://github.com/jwilm/tracker_benchmark).



### 1.2 Download videos

Go to the **Datasets** subpage. Download the videos. I don't know why they don't provide a packed file. So if you have enough patient, you can download the videos one by one and unzip them one by one. Inspired by

---

<sup>1</sup>[http://cvlab.hanyang.ac.kr/tracker\\_benchmark/index.html](http://cvlab.hanyang.ac.kr/tracker_benchmark/index.html)



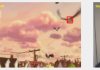




Home **Datasets** Benchmark v1.0 Community

## Datasets

The full benchmark contains 100 sequences from recent literatures.

- The sequence names are in CamelCase without any blanks or underscores (\_).
- When there exist multiple targets each target is identified as dot+id\_number (e.g. Jogging.1 and Jogging.2).
- Each row in the ground-truth files represents the bounding box of the target in that frame, (x, y, box-width, box-height).
- In most sequences the first row corresponds to the first frame and the last row to the last frame, except the following sequences:
  - David(300:770), Football1(1:74), Freeman3(1:460), Freeman4(1:283).

### TB-50 Sequences.

						
<b>Basketball</b> TV, OCC, DEF, OPR, BC	Biker SV, OCC, MB, FM, OPR, OV, LR	Bird1 DEF, FM, OV	BlurBody SV, DEF, MB, FM, IPR	BlurCar2 SV, MB, FM	BlurFace MB, FM, IPR	BlurOwl SV, MB, FM, IPR

João F. Henriques, I provide another automatic download method to save your time.

```
% This script downloads and extracts all videos to the path specified
% below.
%
% Joao F. Henriques, 2014
% http://www.isr.uc.pt/~henriques/

%local path where the videos will be located.
%note that if you change it here, you must also change it in RUN_TRACKER.
base_path = './data/Benchmark/';

%list of videos to download
videos = {'basketball', 'bolt', 'boy', 'car4', 'carDark', 'carScale', ...
          'coke', 'couple', 'crossing', 'david2', 'david3', 'david', 'deer', ...
          'dog1', 'doll', 'dudek', 'faceocc1', 'faceocc2', 'fish', ...
          'fleetface', 'football', 'football1', 'freeman1', 'freeman3', ...
          'freeman4', 'girl', 'ironman', 'jogging', 'jumping', 'lemming', ...
          'liquor', 'matrix', 'mhyang', 'motorRolling', 'mountainBike', ...
          'shaking', 'singer1', 'singer2', 'skating1', 'skiing', 'soccer', ...
          'subway', 'suv', 'sylvestre', 'tiger1', 'tiger2', 'trellis', ...
          'walking', 'walking2', 'woman'};

if ~exist(base_path, 'dir'), %create if it doesn't exist already
    mkdir(base_path);
end

if ~exist('matlabpool', 'file'),
    %no parallel toolbox, use a simple 'for' to iterate
    disp('Downloading_videos_one_by_one, _this_may_take_a_while.')
    disp('_')

    for k = 1:numel(videos),
        disp(['Downloading_and_extracting_' videos{k} '...']);
        unzip(['http://cvlab.hanyang.ac.kr/tracker_benchmark/seq/' ...
              videos{k} '.zip'], base_path);
    end
else
    %download all videos in parallel
    disp('Downloading_videos_in_parallel, _this_may_take_a_while.')
    disp('_')

    if matlabpool('size') == 0,
        matlabpool open;
    end
end
```

```

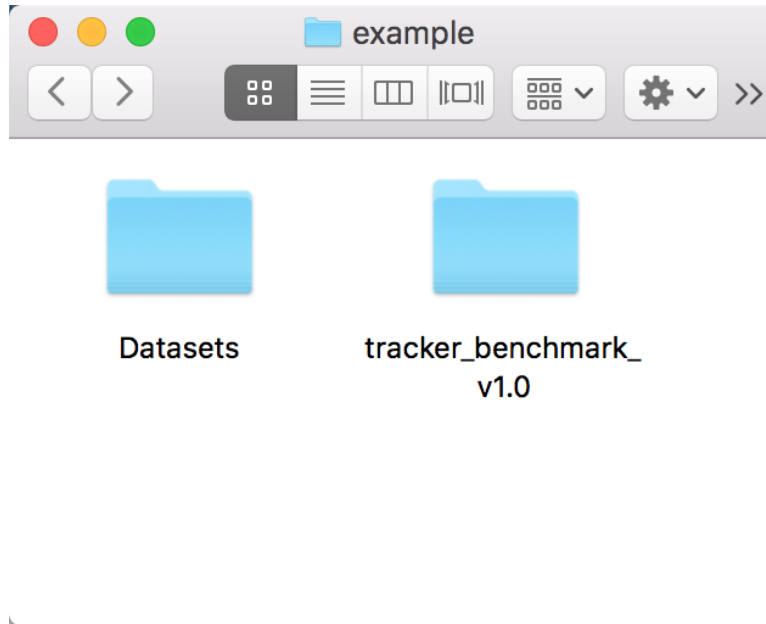
        parfor k = 1:numel(videos),
            disp(['Downloading_and_extracting_' videos{k} '...']);
            unzip(['http://cvlab.hanyang.ac.kr/tracker_benchmark/seq/' ...
                videos{k} '.zip'], base-path);
        end
    end
end

```

You can download this file in the github of my website.

## 2 Configuration perfPlot.m

Now, I assume you have download all files mentioned above. The dataset in the <Datasets> directory and the toolkit in the <tracker\_benchmark.v1.0> directory. The file struct look like this way:



Open the file

**tracker\_benchmark\_v1.0/perfPlot.m**

If your operating system was Windows. you can run without any bug.

If your operating system was OS X or Linux. you should change the \to  
/ in

```

attPath = './anno/att/'; % The folder that contains the annotation files for sequence attributes
figPath = './figs/overall/';
perfMatPath = './perfMat/overall/';

```

genPerfMat.m

```

switch evalType
case 'SRE'
    rpAll=['./results/results_SRE-CVPR13/'];
case {'TRE', 'OPE'}
    rpAll=['./results/results_TRE-CVPR13/'];
end

```

Now you can run the perfPlot.m.

But there is a little different from the normal paper. The ranktype is just AUC. Generally speaking, the threshold of the precision of the evaluation is the use of 20 pixels as a sort of standards. So you can just change the line 105. Be careful it comment has a spelling mistakes.

```

%rankingType = 'AUC';%AUC, threshod
if strcmp(metricTypeSet,'overlap')
    rankingType = 'AUC';%AUC, threshod
else
    rankingType = 'threshold';%AUC, threshod
end

```

### 3 Advanced Topics

After we have run the file to show perfPlot. You must want to evaluate your own algorithms and the 29 algorithms the benchmark provided. But there is a lot of bug you should change. So I don't recommend to run the 29 algorithm for beginners.

Let's just congiguration a the simplest "tracker"-NONE. This is just to show the most basic elements.

#### 3.1 Configuration your own tracker

You must know the process of the main\_running.m. In this section, We just want to run our own tracker. So there is no means to add some line. Just comment these line.

```

% addpath(['D:\vlfeat-0.9.14\toolbox']);
% vl_setup

% addpath(['./rstEval']);
% addpath(['./trackers/VIVID_Tracker'])

```

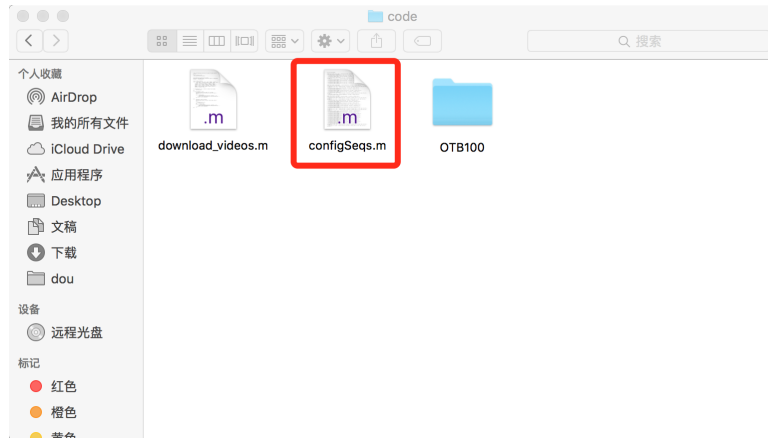
By now, you can run main\_running.m there will jump out of a mistake.

```

Error using imread (line 349)
File "d:\data_seq\carDark_c\0001.jpg" does not exist.

Error in main_running (line 58)
    img = imread(s.s_frames{1});

```



Simple change the dataset path to your own.

And now, you can run main\_running.m If all configuration work well, it will pop up a window showing nothing and the command lines look like this :

```

Command Window
11/29 15:42:12
>> main_running
K>> |

```

Now you can create your own tracker. We create here a tracker name NONE.

1. fix configTrackers.m

```

trackersMY={ struct('name','NONE','namePaper','NONE')};

trackers = [trackersVIVID,trackers1,trackersEXE,trackersMY];

```

2. add a file named NONE in trackers
3. add a run\_NONE.m file in NONE

```

function res = run_NONE(subS, rp, bSaveImage)
init_rect = subS.init_rect;
img_files = subS.s_frames;

res.type = 'rect';
res.res = repmat(init_rect,[numel(img_files),1]);
res.fps = numel(img_files) / time;

```

```
end
```

4.run main\_running.m you can get your own result.  
Set a breakpoint in line 144.

```
eval(funcName);
```

You will see what actually been worked.

### 3.2 OTB100 and OTB50

In this section, We will introduce how to evaluate your tracker on OTB100 and OTB50.

The office website don't provide the configuration file. So you should download in my website.

