

# Advanced Computer Vision

## HOMEWORK 4

Lab : VPILab

Advisor : Cheng-Ming Huang(黃正民)

Student : Yu Cho(卓諭)

Student ID : 106318025

Data : 2017.12.28

## I. Problem

1. Find the target in the image sequence "1.avi"
2. Find the target in the image sequence "2.avi"
3. Find the target in the image sequence "3.avi"

## II. Method

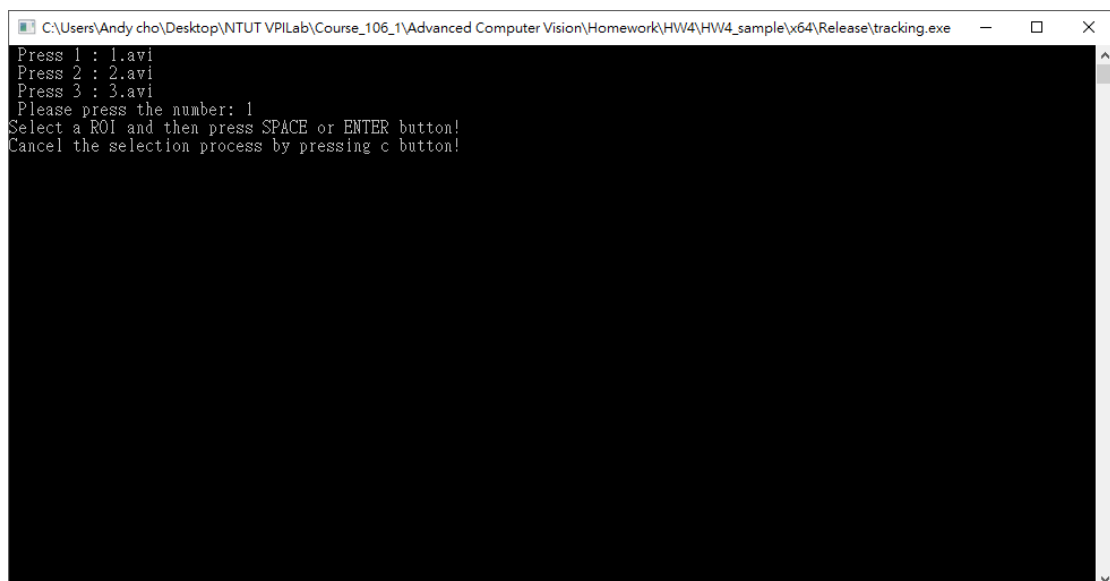
1. Program interface

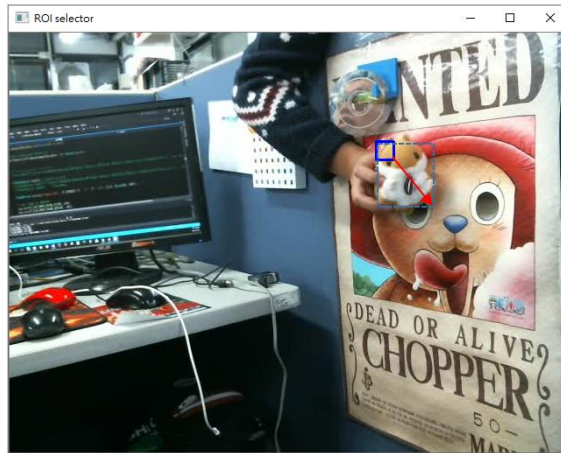
Step1. 路徑:

開啟資料夾 HW4>>開啟資料夾 x64>>開啟資料夾 Release>>tracking.cpp



Step2. 輸入 1 讀取 1.avi，輸入 2 讀取 2.avi，輸入 3 讀取 3.avi，輸入後會進入選擇 ROI 視窗，請用滑鼠標示出物體位置

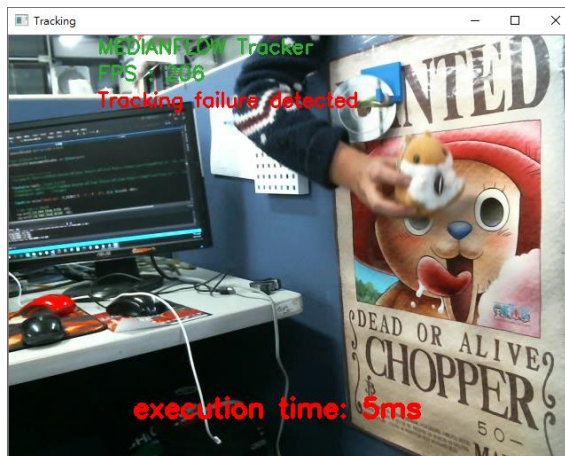




Step3. 開始追蹤，並且每 30 frame 存一張影像，每一張 frame 計算執行時間，最後會在 Command Window 顯示計算過的平均執行時間。



備註:若追蹤失敗，則會顯示”Tracking failure detected”



## 1. MeanShift

Mean-Shift 追蹤演算法是一個疊代過程，它是基於特徵追蹤算法的一種。在初始化的時候要手動標定欲跟蹤的目標區域，然後計算該區域內的顏色直方圖分布，作為目標模板；在後續 frame 中以上一個 frame 的目標位置為起始中心位置，目標長寬的一半分別作為搜索窗口的長寬，計算窗口區域內的顏色直方圖分布，作為候選目標。利用相似度量 Bhattacharyya 係數評價目標模型和當前 frame 候選目標的相似程度，該係數越大代表候選目標與目標模型越相似。而均值偏移向量就是指向 Bhattacharyya 係數增大的方向，通過不斷疊代計算均值偏移向量，在當前 frame 中，當該係數達到最大時，目標中心即收斂到目標的真實位置，從而達到跟蹤的目的。

優點:

1. 算法簡潔，即時性好，算法框架化
2. 採用核函數加權處理，能夠解決目標部分遮擋等情況，對目標的旋轉、變形以及背景變化都不敏感，具有很大的實用價值。

缺點:

1. 目標模型僅採用顏色特徵，缺乏目標空間資訊，對目標的描述不充分，當目標附近有顏色特徵相近的干擾物存在時，易發生錯誤定位。
2. 跟蹤過程中搜索窗口大小保持不變，當目標的尺寸在跟蹤過程中發生變化時，定位目標的矩形框大小並不會依據目標的大小改變，造成定位不準確

## 2. Median Flow

Median Flow 跟蹤方法利用目標框來表示被跟蹤目標，並在連續的相鄰視頻 frame 之間估計目標的運動。

## 3. TLD Tracker

TLD(Tracking-Learning-Detection)算法是一種名為中值流跟蹤 (Median-Flow tracker) 的跟蹤方法基礎上增加了跟蹤失敗檢測算法的新的跟蹤方法。一個用於針對視頻中未知物體長期跟蹤的架構。簡單來說，TLD 算法由三部分組成：跟蹤模塊、檢測模塊、學習模塊。跟蹤模塊是觀察 frame 與 frame 之間的目标的動向。檢測模塊是把每張圖看成獨立的，然後去定位。學習模塊根據跟蹤模塊的結果對檢測模塊的錯誤進行評估，生成訓練樣本來對檢測模塊的目標模型進行更新，避免以後出現類似錯誤。

優點:

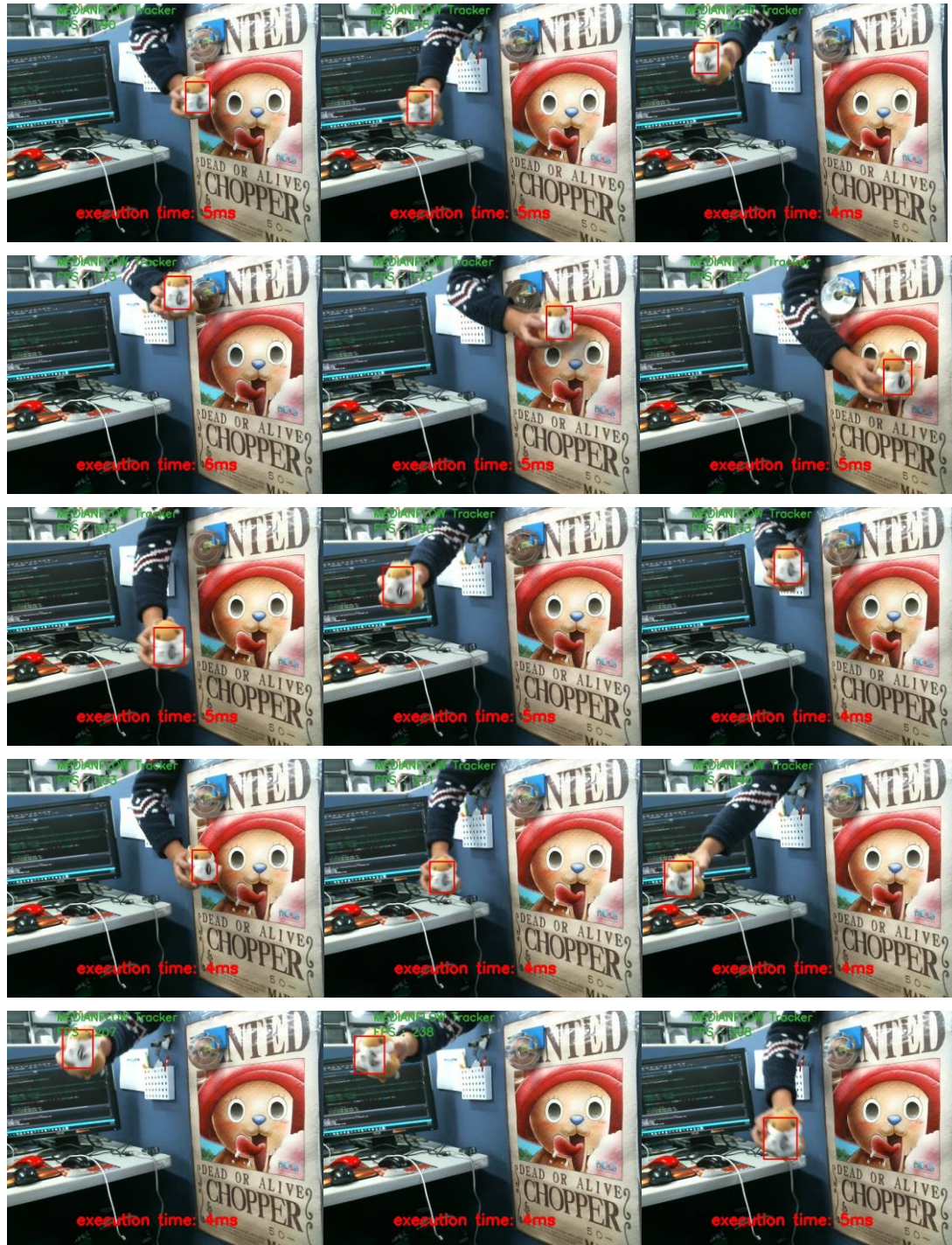
能對鎖定的目標進行不斷的學習，以獲取目標最新的外觀特徵，從而及時完美跟蹤，以達到最佳的狀態。



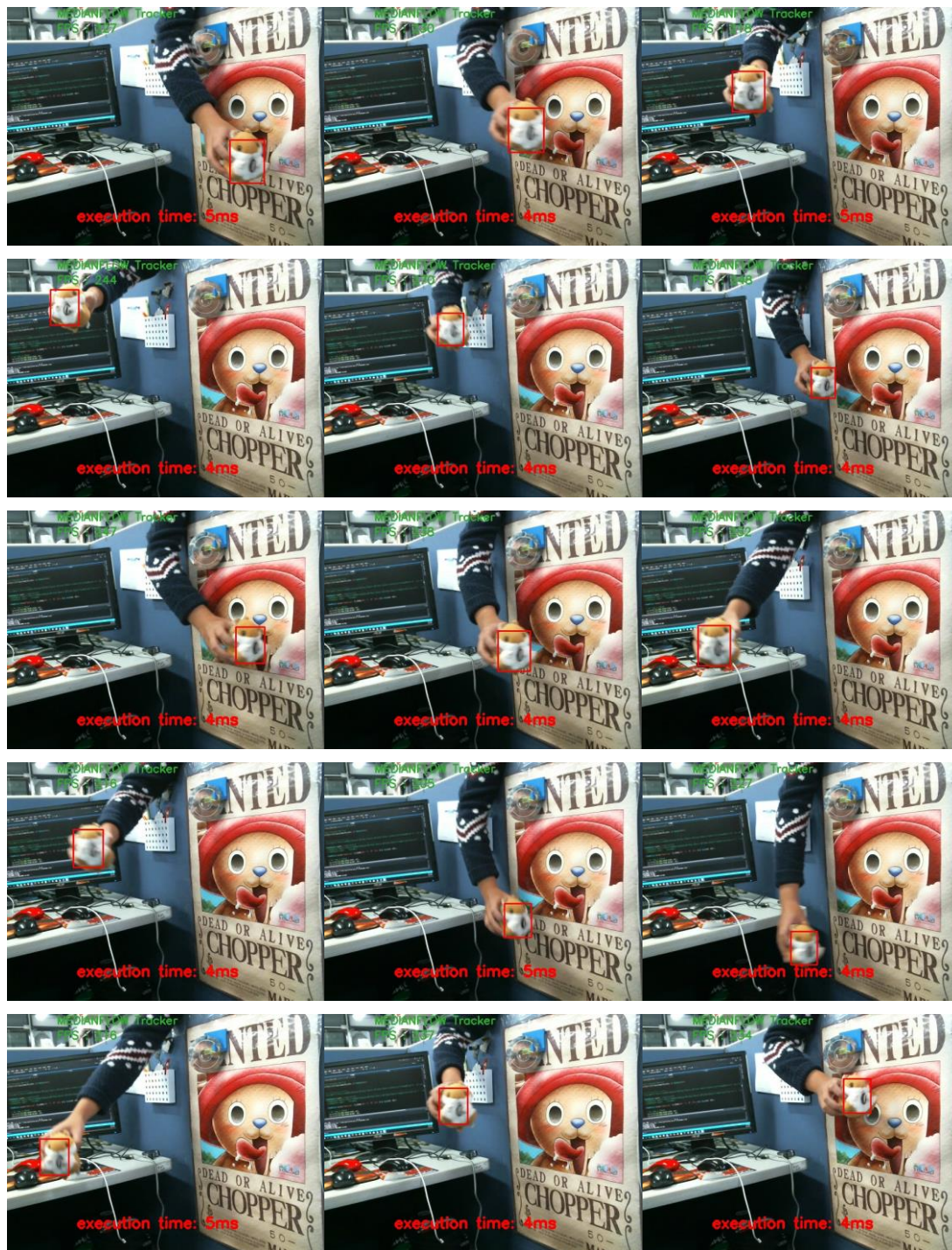
### III. Result

#### 1. Find the target in the image sequence "1.avi"

每 30 個 frame 取樣一張影像，共取樣 30 張，其中 30 張影像正確追蹤







```

C:\Users\Andy.cho\Desktop\NTUT_VPLab\Course_106_T\Advanced Computer Vision\Homework\HW4\HW4_sample\y4\Release\tracking.exe
save: SAVE_IMG_Frame0090.jpg
save: SAVE_IMG_Frame0120.jpg
save: SAVE_IMG_Frame0180.jpg
save: SAVE_IMG_Frame0210.jpg
save: SAVE_IMG_Frame0240.jpg
save: SAVE_IMG_Frame0270.jpg
save: SAVE_IMG_Frame0300.jpg
save: SAVE_IMG_Frame0330.jpg
save: SAVE_IMG_Frame0360.jpg
save: SAVE_IMG_Frame0390.jpg
save: SAVE_IMG_Frame0420.jpg
save: SAVE_IMG_Frame0450.jpg
save: SAVE_IMG_Frame0480.jpg
save: SAVE_IMG_Frame0510.jpg
save: SAVE_IMG_Frame0540.jpg
save: SAVE_IMG_Frame0570.jpg
save: SAVE_IMG_Frame0600.jpg
save: SAVE_IMG_Frame0630.jpg
save: SAVE_IMG_Frame0660.jpg
save: SAVE_IMG_Frame0690.jpg
save: SAVE_IMG_Frame0720.jpg
save: SAVE_IMG_Frame0750.jpg
save: SAVE_IMG_Frame0780.jpg
save: SAVE_IMG_Frame0810.jpg
save: SAVE_IMG_Frame0840.jpg
save: SAVE_IMG_Frame0870.jpg
save: SAVE_IMG_Frame0900.jpg
the average = 4.640551 ms

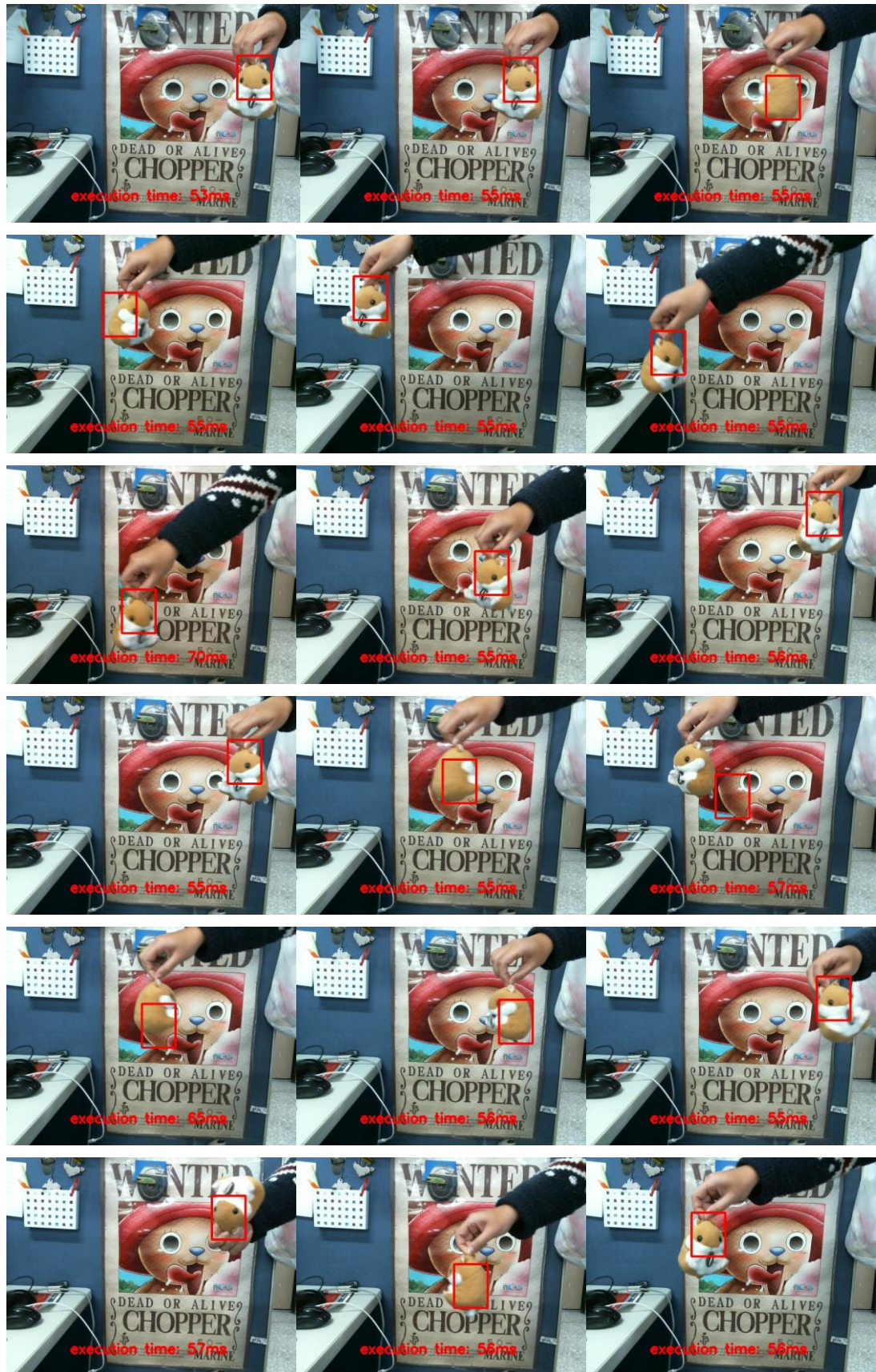
```

每張 frame 平均執行時間

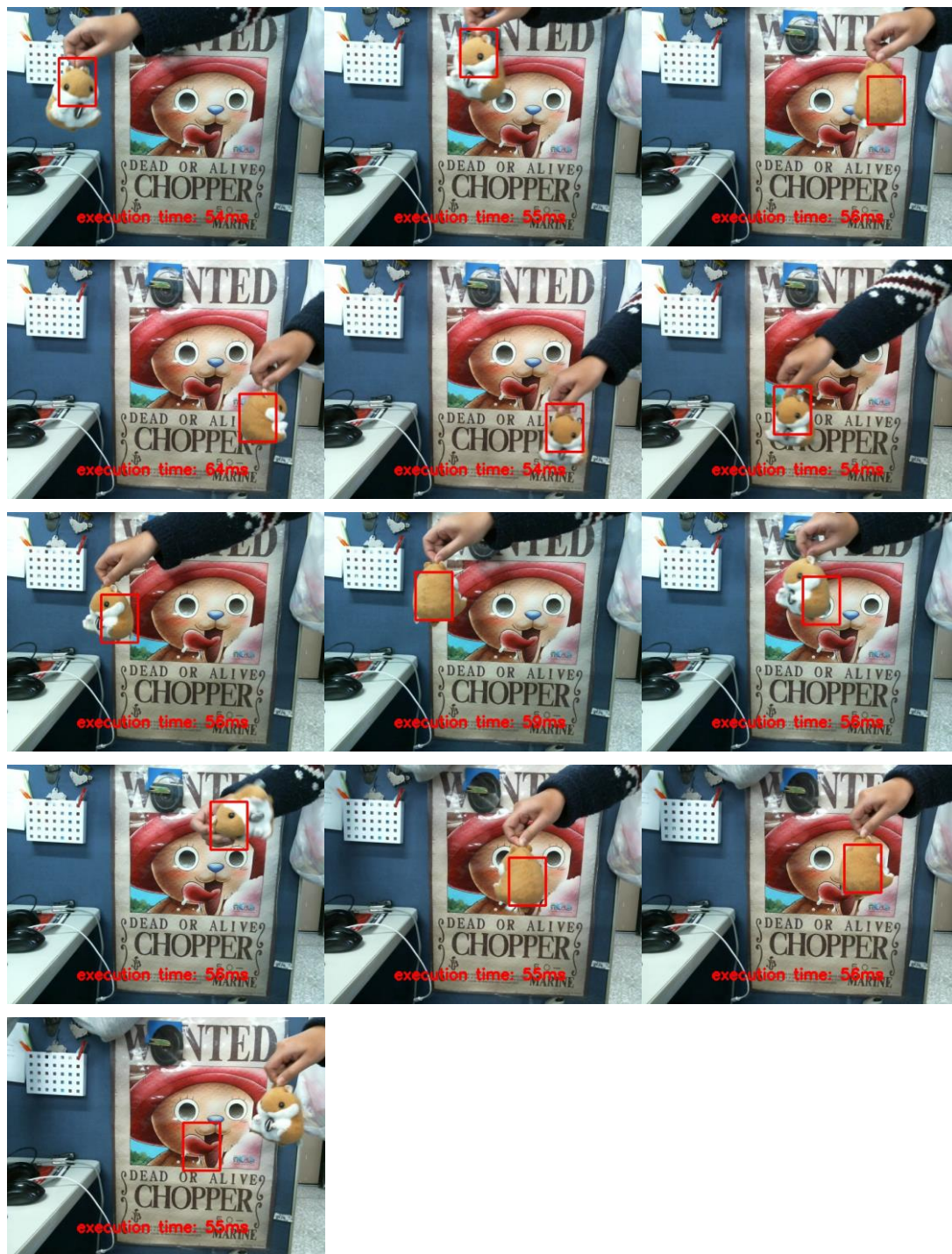


2. Find the target in the image sequence "2.avi"

每 30 個 frame 取樣一張影像，共取樣 31 張，其中 29 張影像正確追蹤







```

C:\Users\Andy\cho\Desktop\NTUT_VPLab\Course_106_1\Advanced Computer Vision\Homework\HW4\HW4_sample\vx64\Release\tracking.exe
save: SAVE_IMG Frame0120.jpg
save: SAVE_IMG Frame0150.jpg
save: SAVE_IMG Frame0180.jpg
save: SAVE_IMG Frame0210.jpg
save: SAVE_IMG Frame0240.jpg
save: SAVE_IMG Frame0270.jpg
save: SAVE_IMG Frame0300.jpg
save: SAVE_IMG Frame0330.jpg
save: SAVE_IMG Frame0360.jpg
save: SAVE_IMG Frame0390.jpg
save: SAVE_IMG Frame0420.jpg
save: SAVE_IMG Frame0450.jpg
save: SAVE_IMG Frame0480.jpg
save: SAVE_IMG Frame0510.jpg
save: SAVE_IMG Frame0540.jpg
save: SAVE_IMG Frame0570.jpg
save: SAVE_IMG Frame0600.jpg
save: SAVE_IMG Frame0630.jpg
save: SAVE_IMG Frame0660.jpg
save: SAVE_IMG Frame0690.jpg
save: SAVE_IMG Frame0720.jpg
save: SAVE_IMG Frame0750.jpg
save: SAVE_IMG Frame0780.jpg
save: SAVE_IMG Frame0810.jpg
save: SAVE_IMG Frame0840.jpg
save: SAVE_IMG Frame0870.jpg
save: SAVE_IMG Frame0900.jpg
save: SAVE_IMG Frame0930.jpg
time average = 55.205755 ms

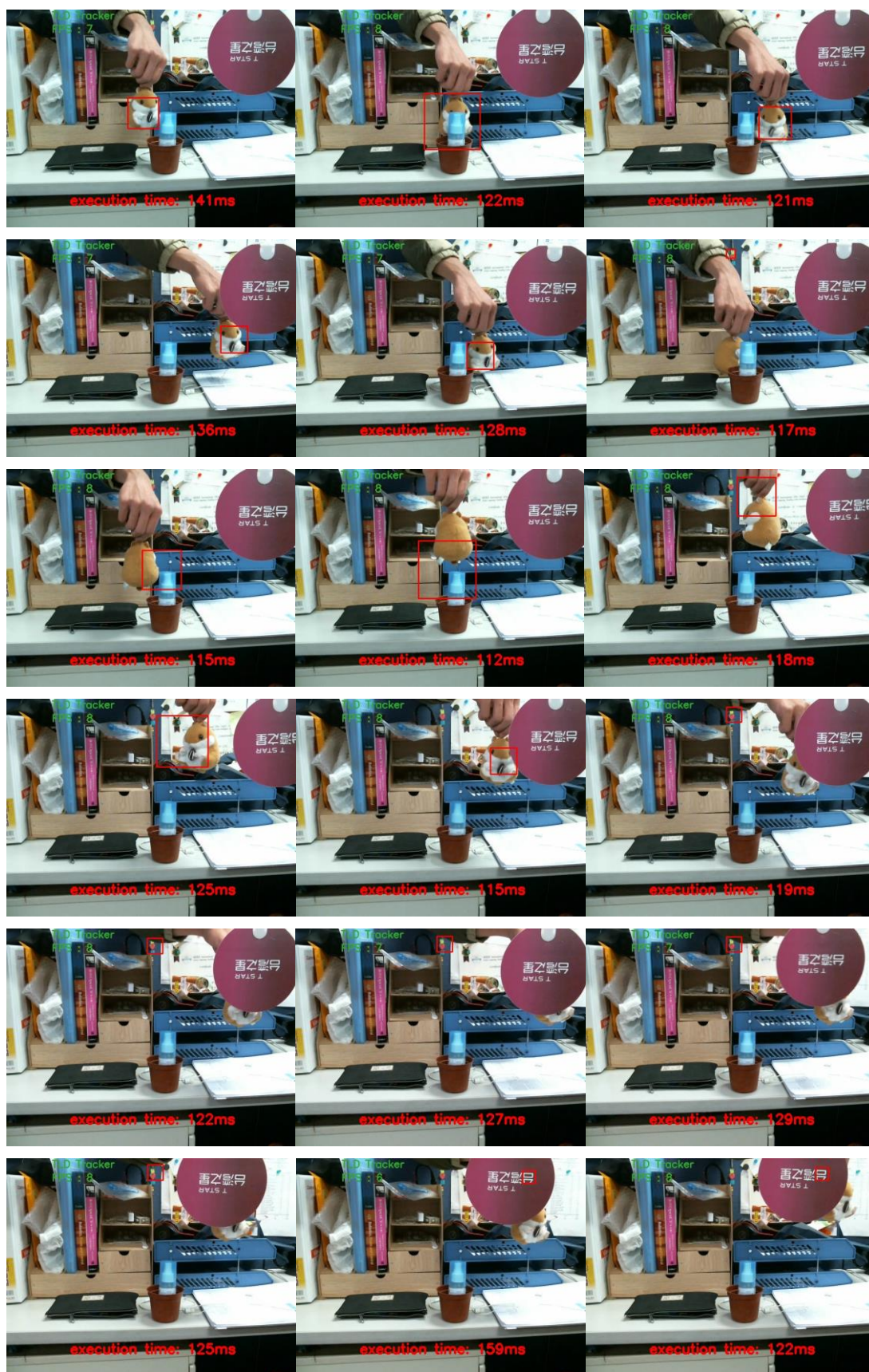
```

每張 frame 平均執行時間

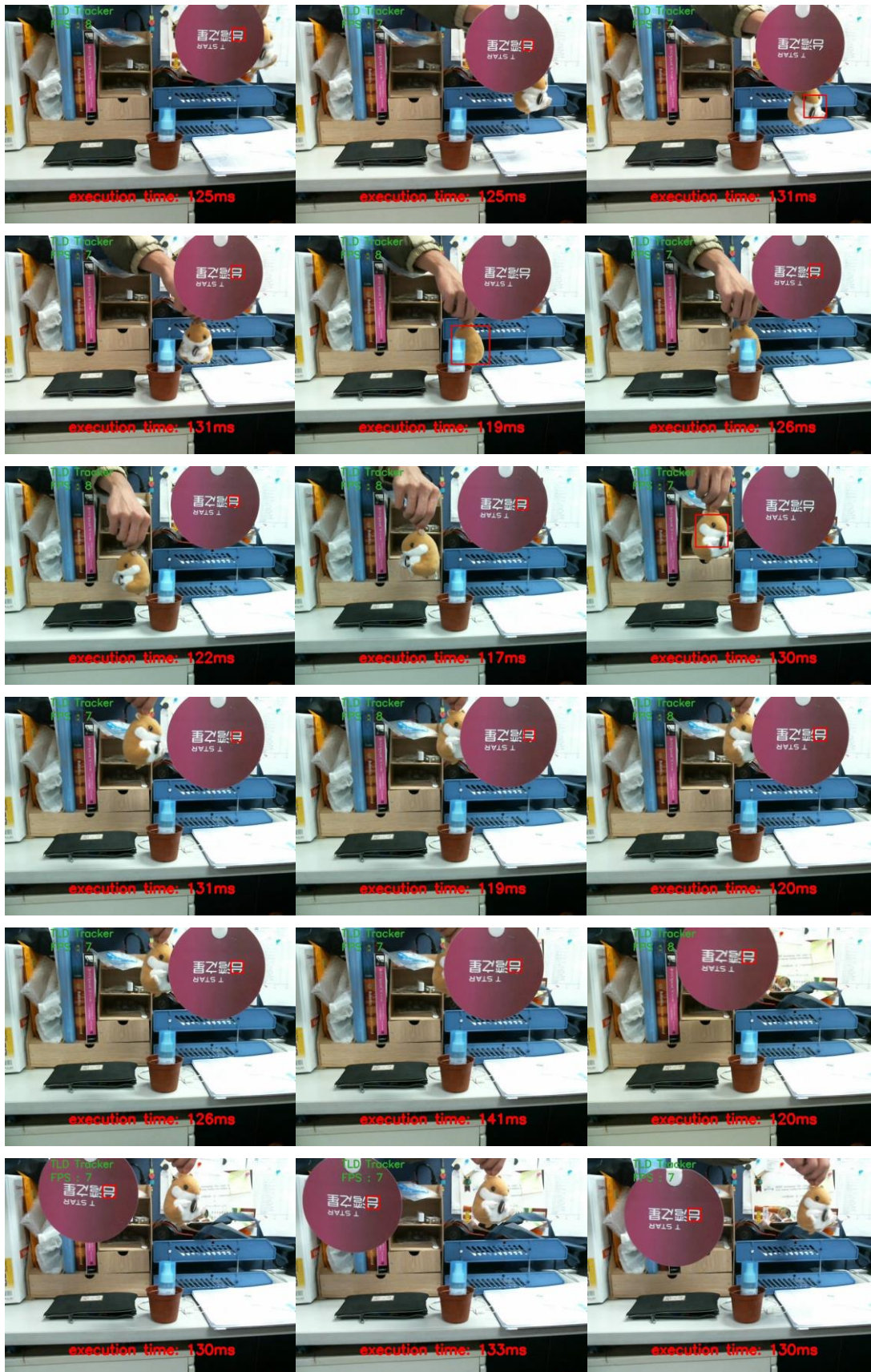


### 3. Find the target in the image sequence "3.avi"

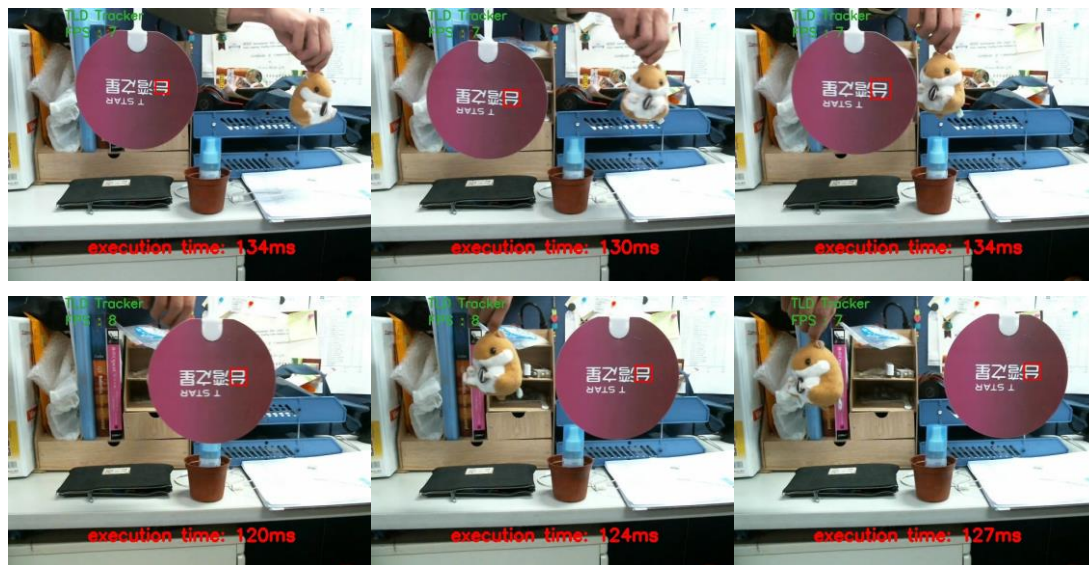
每 30 個 frame 取樣一張影像，共取樣 42 張，其中 13 張影像正確追蹤











```

C:\Users\Andy cho\Desktop\NTUT VPLab\Course_106_1\Advanced Computer Vision\Homework\HW4\HW4_sample\v64\Release\tracking.exe
save: SAVE_IMG_Frame0450.jpg
save: SAVE_IMG_Frame0480.jpg
save: SAVE_IMG_Frame0510.jpg
save: SAVE_IMG_Frame0540.jpg
save: SAVE_IMG_Frame0570.jpg
save: SAVE_IMG_Frame0600.jpg
save: SAVE_IMG_Frame0630.jpg
save: SAVE_IMG_Frame0660.jpg
save: SAVE_IMG_Frame0690.jpg
save: SAVE_IMG_Frame0720.jpg
save: SAVE_IMG_Frame0750.jpg
save: SAVE_IMG_Frame0780.jpg
save: SAVE_IMG_Frame0810.jpg
save: SAVE_IMG_Frame0840.jpg
save: SAVE_IMG_Frame0870.jpg
save: SAVE_IMG_Frame0900.jpg
save: SAVE_IMG_Frame0930.jpg
save: SAVE_IMG_Frame0960.jpg
save: SAVE_IMG_Frame0990.jpg
save: SAVE_IMG_Frame1020.jpg
save: SAVE_IMG_Frame1050.jpg
save: SAVE_IMG_Frame1080.jpg
save: SAVE_IMG_Frame1110.jpg
save: SAVE_IMG_Frame1140.jpg
save: SAVE_IMG_Frame1170.jpg
save: SAVE_IMG_Frame1200.jpg
save: SAVE_IMG_Frame1230.jpg
save: SAVE_IMG_Frame1260.jpg
time average = 123.010212 ms

```

每張 frame 平均執行時間

## IV. Discussion

### 1. Find the target in the image sequence"1.avi"

目標物沒有任何的旋轉或者遮蔽，故我使用 Median Flow 追蹤，可以增加追蹤的效率，大約一個 frame 處理時間約為 4.6ms，並且，並且每 30 個 frame 取樣一張影像，總共取樣 30 張，其中有 30 張影像正確追蹤，可以達到完美追蹤，準確率達 100%。

### 2. Find the target in the image sequence"2.avi"

目標物增加了旋轉，故我使用 MeanShift 追蹤，此演算法對目標的旋轉、變形以及背景變化都不敏感，而為何不使用 CamShift 是因為，CamShift 會變換搜索窗口的大小，其目的為當物體有遠近時，依然可以準確地追蹤物體，但這部影片沒有物體遠近，我發現使用 CamShift 時會造成誤判，使搜索視窗拉大，最終失去追蹤能力，故改回原本的 MeanShift 使用，兼顧執行準確度以及效率，大約一個 frame 處理時間為 56.3ms，並且每 30 個 frame 取樣一張影像，總共取樣 30 張，其中有 28 張影像正確追蹤，準確率達 93%。

### 3. Find the target in the image sequence"3.avi"

目標物除了有旋轉還增加了遮蔽，雖然 MeanShift 可以在部分遮蔽時維持穩健性，但是此影片的遮蔽物會將本體蓋過，我使用 TLD 追蹤法能對鎖定的目標進行不斷的學習，以獲取目標最新的外觀特徵，從而及時完美跟踪，以達到最佳的狀態。但卻不是最佳的方法，可以看到因為此方法增加了不斷學習的演算，處理時間大幅增加，無法應用於即時影像，而且在目標物完全被遮蔽的情況下，會偵測錯誤，我認為可以使用 Kalman Filter 改進在物體遇到遮蔽時的追蹤。一個 frame 處理時間約為 128ms，並且每 30 個 frame 取樣一張影像，總共取樣 42 張，其中有 13 張影像正確追蹤，準確率只有 31%。

	1.avi	2.avi	3.avi
Method	Median Flow	MeanShift	TLD
Execution time	4.6ms	56.3ms	128ms
Accuracy rate	100%	93%	31%

(表一)各影片追蹤方法與時間準確度比較