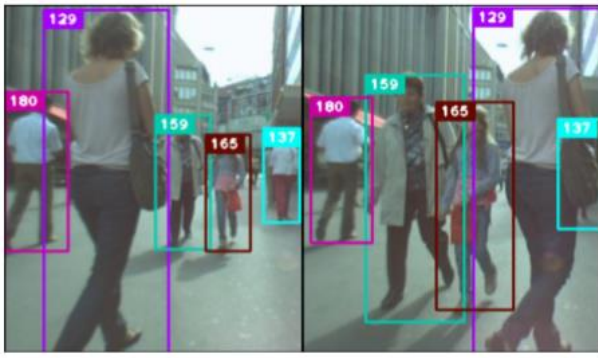
ABSTRACT

Simple Online and Realtime Tracking (SORT) is a pragmatic approach to multiple object tracking with a focus on simple, effective algorithms. In this paper, we integrate appearance information to improve the performance of SORT. Due to this extension we are able to track objects through longer periods of occlusions, effectively reducing the number of identity switches. In spirit of the original framework we place much of the computational complexity into an offline pre-training stage where we learn a deep association metric on a large-scale person re-identification dataset. During online application, we establish measurement-to-track associations using nearest neighbor queries in visual appearance space. Experimental evaluation shows that our extensions reduce the number of identity switches by 45%, achieving overall competitive performance at high frame rates.



Listing 1 Matching Cascade

Input: Track indices $\mathcal{T} = \{1, \ldots, N\}$, Detection indices $\mathcal{D} = \{1, \ldots, M\}$, Maximum age $A_{\max}$

Compute cost matrix $C = [c_{i,j}]$ using Eq. 5
Compute gate matrix $B = [b_{i,j}]$ using Eq. 6
Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
Initialize set of unmatched detections $\mathcal{U} \leftarrow \mathcal{D}$
for $n \in \{1, \ldots, A_{\max}\}$ **do**
  Select tracks by age $\mathcal{T}_n \leftarrow \{i \in \mathcal{T} \mid a_i = n\}$
  $[x_{i,j}] \leftarrow \text{min\_cost\_matching}(C, \mathcal{T}_n, \mathcal{U})$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
end for
return $\mathcal{M}, \mathcal{U}$

- Matching Algorithm

Name	Patch Size/Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
Batch and ℓ_2 normalization		128

■ CNN Architecture

Purpose: Improving Online Detecting && Tracking using SORT with Deep Learning.

Method: Using SORT with CNN while Tracking

Problem: SORT is a pragmatic approach to MOT with a focus on simple, effective algorithms. However, there's a problem with Identity switches.

Suggestion: **improving SORT algorithm using Deep Learning.** Overcome this issue by replacing the association metric with a more informed metric that combines motion and appearance information. (association metric을 동작 및 모양 정보를 결합한 보다 유용한 metric으로 대체함...?) => **Applying CNN that has been trained to discriminate pedestrians on a large-scale person re-identification dataset.**

문제점: tracking에서 여전히 occlusion이 많이 발생. FPS가 현저하게 떨어짐. 약 7~8fps.

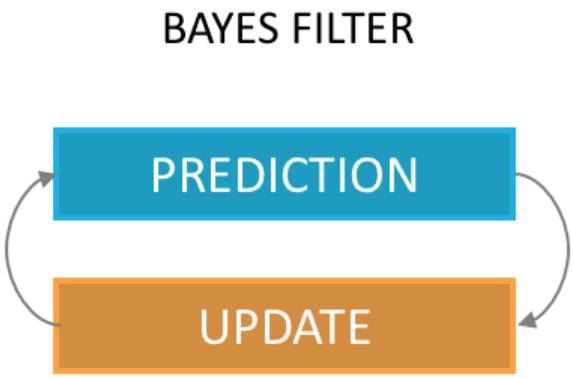
TODO: fps를 높이면서 occlusion을 조금 더 개선할 방법?

SORT: Kalman Filter + Hungarian Algorithm

DeepSORT = DeepLearning + SORT

Kalman Filter: 기존에 추적하던 물체의 속도를 반영하여 다음 상황을 예측함.

1. 과거의 값을 이용해 현재 값을 예측한다.
2. 예측 값과 측정 값에 각각 노이즈를 반영한 후, 실제 값을 예측함.
3. 실제 값을 다시 다음 측정에 사용



SORT: 칼만필터 + 헝가리안 알고리즘

- 헝가리안 알고리즘: 최저비용의 할당을 하려고 하는 최적화 문제를 해결하기 위한 알고리즘, scipy의 linear_assignment라는 함수로 구현되어 있음
- Non Maximum Suppressions(NMS): 여러 bbox가 겹쳐 있을 때, 어떤 것을 선택하고 어떤 것을 버릴지 판단하는 알고리즘. 모든 bbox에 대해, 가장 높은 confidence score를 가진 box를 선택하고, 해당 박스와의 IOU가 threshold이상이면 제외(suppression)함.
  - 딥소트에서 YOLOv3가 detecting한 bbox를 tracker로 넘기기 전에, preprocessing으로 사용함
- MARS: Re-id dataset: Deep SORT에서는 이전의 Tracking 결과와 현재 Detecting결과의 bbox를 매칭하는데 헝가리안 알고리즘을 사용함. 이때 사용하는 최적화 factor는 3가지(KNN, deeplearning feature, IOU).
  - 여기서 deeplearning feature는 bbox내의 이미지간 유사도를 나타내며, 해당 모델을 학습시킬 때 사용되는 dataset은 Market1501과 MARS가 있다. 둘 다 re-id를 위해서 만들어진 dataset이며, MARS는 비디오와 같은 time-series dataset에 특화시켜 Market1501을 확장한 버전임.
  - DeepSORT는 주로 CCTV에서 보행자를 tracking하는 등, 비디오 데이터를 이용하여 tracking을 하는데 많이 사용되어, MARS로 학습된 딥러닝 모델로 re-id 점수를 매겼을 때 성능이 더 잘나옴.
- DeepSORT
  - 딥소트는 칼만필터 기본으로 딥러닝 피쳐(Re-Id)를 추가로 반영하

여 헝가리안 알고리즘을 수행하는 것.

- 딥러닝 피쳐는 칼만필터의 한계때문에 도입된 것으로 생각하는데, 칼만필터는 이전 속도를 기반으로 예측하기 때문에 실제로 SORT나 칼만필터만 써서 tracking하게 되면 둘이 겹치는(occlusion)scene에서 오류가 많이 남.
- 둘이 겹치는 부분에서 갑자기 서로 반대로 가거나, 한명이 멈춰있다가 나타나는 경우 tracking id를 반대로 바뀌어 지거나 새로운 id를 부여하는 경우가 많이 발생
- DeepSORT는 이러한 경우를 상당부분 보정해줌. 즉 Kalman gain이 높더라도, 이미지 feature이 서로 유사하면 아이디를 유지해주고 새로운 아이디를 부여하지 않음.

참고😊

http://blog.haandol.com/2020/02/27/deep-sort-with-mxnet-yolo3.html#fn:2