UUCT - HyMP: Towards Tracking Dispersed Crowd Groups from UAVs

Tonmoay Deb, Mahieyin Rahmun, Shahriar Ali Bijoy, Mayamin Hamid Raha, and Dr. Mohammad Ashrafuzzaman Khan

Department of Electrical & Computer Engineering, North South University

International Joint Conference on Neural Networks 2021
Paper ID: 1827

Outline

- Introduction and Motivation

Outline

- Introduction and Motivation
- Simulator
  - UUCT - dataset generation

Outline

- Introduction and Motivation
- Simulator
  - UUCT - dataset generation
  - Ground truth generation and evaluation

Outline

- Introduction and Motivation
- Simulator
  - UUCT - dataset generation
  - Ground truth generation and evaluation
- Proposed tracking algorithm: HyMP

Outline

- Introduction and Motivation
- Simulator
  - UUCT - dataset generation
  - Ground truth generation and evaluation
- Proposed tracking algorithm: HyMP
- Results

Outline

- Introduction and Motivation
- Simulator
  - UUCT - dataset generation
  - Ground truth generation and evaluation
- Proposed tracking algorithm: HyMP
- Results
- Conclusion
- Simulator Demo

Outline

- Introduction and Motivation
- Simulator
  - UUCT - dataset generation
  - Ground truth generation and evaluation
- Proposed tracking algorithm: HyMP
- Results
- Conclusion
- Simulator Demo

Introduction and Motivation

Why We Are Interested in New Tracking Approach?

- A challenging problem: *how to track groups of dispersed crowds from aerial viewpoint?*

Why We Are Interested in New Tracking Approach?

- A challenging problem: *how to track groups of dispersed crowds from aerial viewpoint?*
- Existing Object Trackers may not be able to adapt to our problem.

Why We Are Interested in New Tracking Approach?

- A challenging problem: *how to track groups of dispersed crowds from aerial viewpoint?*
- Existing Object Trackers may not be able to adapt to our problem.
- Issues with Single Object Trackers (SOT)

Why We Are Interested in New Tracking Approach?

- A challenging problem: *how to track groups of dispersed crowds from aerial viewpoint?*
- Existing Object Trackers may not be able to adapt to our problem.
- Issues with Single Object Trackers (SOT)
  - It can not adapt to crowd group deformations or reformations because they might move sparse.

Why We Are Interested in New Tracking Approach?

- A challenging problem: *how to track groups of dispersed crowds from aerial viewpoint?*
- Existing Object Trackers may not be able to adapt to our problem.
- Issues with Single Object Trackers (SOT)
  - It can not adapt to crowd group deformations or reformations because they might move sparse.
  - The tracker memory can not scale according to the crowd shape changes

Why We Are Interested in New Tracking Approach?

- A challenging problem: *how to track groups of dispersed crowds from aerial viewpoint?*
- Existing Object Trackers may not be able to adapt to our problem.
- Issues with Single Object Trackers (SOT)
  - It can not adapt to crowd group deformations or reformations because they might move sparse.
  - The tracker memory can not scale according to the crowd shape changes
- Issues with Multi-Object Trackers (MOT)

Why We Are Interested in New Tracking Approach?

- A challenging problem: *how to track groups of dispersed crowds from aerial viewpoint?*
- Existing Object Trackers may not be able to adapt to our problem.
- Issues with Single Object Trackers (SOT)
  - It can not adapt to crowd group deformations or reformations because they might move sparse.
  - The tracker memory can not scale according to the crowd shape changes
- Issues with Multi-Object Trackers (MOT)
  - Our dataset will have a number of people to track. MOTs will fail to scale tracking each person simultaneously.

Why We Are Interested in New Tracking Approach?

- A challenging problem: *how to track groups of dispersed crowds from aerial viewpoint?*
- Existing Object Trackers may not be able to adapt to our problem.
- Issues with Single Object Trackers (SOT)
  - It can not adapt to crowd group deformations or reformations because they might move sparse.
  - The tracker memory can not scale according to the crowd shape changes
- Issues with Multi-Object Trackers (MOT)
  - Our dataset will have a number of people to track. MOTs will fail to scale tracking each person simultaneously.
  - It may not be able to handle crowd dispersion, i.e., split into several groups suddenly because it can not connect the relation between 'multiple targets.'

Why We Are Interested in New Tracking Approach?

- A challenging problem: *how to track groups of dispersed crowds from aerial viewpoint?*
- Existing Object Trackers may not be able to adapt to our problem.
- Issues with Single Object Trackers (SOT)
  - It can not adapt to crowd group deformations or reformations because they might move sparse.
  - The tracker memory can not scale according to the crowd shape changes
- Issues with Multi-Object Trackers (MOT)
  - Our dataset will have a number of people to track. MOTs will fail to scale tracking each person simultaneously.
  - It may not be able to handle crowd dispersion, i.e., split into several groups suddenly because it can not connect the relation between 'multiple targets.'

To tackle these issues, we propose a novel dataset and benchmark, followed by a tracking algorithm that learns end-to-end.

A Quick Discussion About Our Dataset and Benchmark

- Dataset

A Quick Discussion About Our Dataset and Benchmark

- Dataset
  - Existing datasets deal with object tracking in a generalized manner e.g. not crowd specific

A Quick Discussion About Our Dataset and Benchmark

- Dataset
  - Existing datasets deal with object tracking in a generalized manner e.g. not crowd specific
  - Lack of crowd group tracking benchmarks from aerial viewpoint

A Quick Discussion About Our Dataset and Benchmark

- Dataset
  - Existing datasets deal with object tracking in a generalized manner e.g. not crowd specific
  - Lack of crowd group tracking benchmarks from aerial viewpoint
- The proposed dataset and benchmark: UUCT

A Quick Discussion About Our Dataset and Benchmark

- Dataset
  - Existing datasets deal with object tracking in a generalized manner e.g. not crowd specific
  - Lack of crowd group tracking benchmarks from aerial viewpoint
- The proposed dataset and benchmark: UUCT
  - Unreal UAV Crowd Tracking

A Quick Discussion About Our Dataset and Benchmark

- Dataset
  - Existing datasets deal with object tracking in a generalized manner e.g. not crowd specific
  - Lack of crowd group tracking benchmarks from aerial viewpoint
- The proposed dataset and benchmark: UUCT
  - Unreal UAV Crowd Tracking
  - Built with Unreal Engine

A Quick Discussion About Our Dataset and Benchmark

- Dataset
  - Existing datasets deal with object tracking in a generalized manner e.g. not crowd specific
  - Lack of crowd group tracking benchmarks from aerial viewpoint
- The proposed dataset and benchmark: UUCT
  - Unreal UAV Crowd Tracking
  - Built with Unreal Engine
  - Baseline benchmark

A Quick Discussion About Our Dataset and Benchmark

- Dataset
  - Existing datasets deal with object tracking in a generalized manner e.g. not crowd specific
  - Lack of crowd group tracking benchmarks from aerial viewpoint
- The proposed dataset and benchmark: UUCT
  - Unreal UAV Crowd Tracking
  - Built with Unreal Engine
  - Baseline benchmark
  - Photorealistic synthetic images

A Quick Discussion About Our Dataset and Benchmark

- Dataset
  - Existing datasets deal with object tracking in a generalized manner e.g. not crowd specific
  - Lack of crowd group tracking benchmarks from aerial viewpoint
- The proposed dataset and benchmark: UUCT
  - Unreal UAV Crowd Tracking
  - Built with Unreal Engine
  - Baseline benchmark
  - Photorealistic synthetic images
  - Automated ground truth generation

A Quick Discussion About Our Dataset and Benchmark

- Dataset
  - Existing datasets deal with object tracking in a generalized manner e.g. not crowd specific
  - Lack of crowd group tracking benchmarks from aerial viewpoint
- The proposed dataset and benchmark: UUCT
  - Unreal UAV Crowd Tracking
  - Built with Unreal Engine
  - Baseline benchmark
  - Photorealistic synthetic images
  - Automated ground truth generation
  - Rich in crowd-specific attributes

A Quick Discussion About Our Dataset and Benchmark

- Dataset
  - Existing datasets deal with object tracking in a generalized manner e.g. not crowd specific
  - Lack of crowd group tracking benchmarks from aerial viewpoint
- The proposed dataset and benchmark: UUCT
  - Unreal UAV Crowd Tracking
  - Built with Unreal Engine
  - Baseline benchmark
  - Photorealistic synthetic images
  - Automated ground truth generation
  - Rich in crowd-specific attributes
  - Evaluated on existing tracking algorithms and HyMP

Simulator

Development of The Simulator Pipeline

- Built with Unreal Engine 4

Development of The Simulator Pipeline

- Built with Unreal Engine 4
- Uses Microsoft AirSim plugin for UAV deployment, control and image capturing

Development of The Simulator Pipeline

- Built with Unreal Engine 4
- Uses Microsoft AirSim plugin for UAV deployment, control and image capturing
- AI controlled NPCs for directed/randomized movement around the levels

Development of The Simulator Pipeline

- Built with Unreal Engine 4
- Uses Microsoft AirSim plugin for UAV deployment, control and image capturing
- AI controlled NPCs for directed/randomized movement around the levels
- A baseline benchmark

Development of The Simulator Pipeline

- Built with Unreal Engine 4
- Uses Microsoft AirSim plugin for UAV deployment, control and image capturing
- AI controlled NPCs for directed/randomized movement around the levels
- A baseline benchmark
  - No environmental clutters/props

Development of The Simulator Pipeline

- Built with Unreal Engine 4
- Uses Microsoft AirSim plugin for UAV deployment, control and image capturing
- AI controlled NPCs for directed/randomized movement around the levels
- A baseline benchmark
  - No environmental clutters/props
  - The complexity of the level design can be increased to make the benchmark more difficult in the future

Level Design and Dataset Generation

- Divided into two components

Level Design and Dataset Generation

- Divided into two components
  - Setting up the paths

Level Design and Dataset Generation

- Divided into two components
  - Setting up the paths
  - Placing the NPCs on the level

Level Design and Dataset Generation

- Divided into two components
  - Setting up the paths
  - Placing the NPCs on the level
- Images captured using AirSim Python API

Level Design and Dataset Generation

- Divided into two components
  - Setting up the paths
  - Placing the NPCs on the level
- Images captured using AirSim Python API
  - RGB

Level Design and Dataset Generation

- Divided into two components
  - Setting up the paths
  - Placing the NPCs on the level
- Images captured using AirSim Python API
  - RGB
  - Segmentation

Level Design and Dataset Generation

- Divided into two components
  - Setting up the paths
  - Placing the NPCs on the level
- Images captured using AirSim Python API
  - RGB
  - Segmentation
  - Depth Perspective

Level Design and Dataset Generation

- Divided into two components
  - Setting up the paths
  - Placing the NPCs on the level
- Images captured using AirSim Python API
  - RGB
  - Segmentation
  - Depth Perspective
- Segmentation images used to compute bounding boxes around crowd groups

Level Design and Dataset Generation

- Divided into two components
  - Setting up the paths
  - Placing the NPCs on the level
- Images captured using AirSim Python API
  - RGB
  - Segmentation
  - Depth Perspective
- Segmentation images used to compute bounding boxes around crowd groups
- Individual bounding boxes processed to obtain a single Ground Truth Bounding Box which should encompass the largest crowd

Level Design and Dataset Generation

- Divided into two components
  - Setting up the paths
  - Placing the NPCs on the level
- Images captured using AirSim Python API
  - RGB
  - Segmentation
  - Depth Perspective
- Segmentation images used to compute bounding boxes around crowd groups
- Individual bounding boxes processed to obtain a single Ground Truth Bounding Box which should encompass the largest crowd
- The frames are combined into a single video

Level Design and Dataset Generation

- Divided into two components
  - Setting up the paths
  - Placing the NPCs on the level
- Images captured using AirSim Python API
  - RGB
  - Segmentation
  - Depth Perspective
- Segmentation images used to compute bounding boxes around crowd groups
- Individual bounding boxes processed to obtain a single Ground Truth Bounding Box which should encompass the largest crowd
- The frames are combined into a single video
- We will demonstrate the entire process at the end, stay tuned!

Level Design and Image Capturing (continued)

- Time Dilation is employed while capturing images

Level Design and Image Capturing (continued)

- Time Dilation is employed while capturing images
  - Time dilation implies “slowing down” the simulation world

Level Design and Image Capturing (continued)

- Time Dilation is employed while capturing images
  - Time dilation implies "slowing down" the simulation world
  - 1 second elapsed in simulation world clock $= n$ seconds elapsed in real world clock; $n > 1$

Level Design and Image Capturing (continued)

- Time Dilation is employed while capturing images
  - Time dilation implies "slowing down" the simulation world
  - 1 second elapsed in simulation world clock $= n$ seconds elapsed in real world clock; $n > 1$
  - The AirSim API cannot maintain stable frame rates if this is not done

Level Design and Image Capturing (continued)

- Time Dilation is employed while capturing images
  - Time dilation implies "slowing down" the simulation world
  - 1 second elapsed in simulation world clock $= n$ seconds elapsed in real world clock; $n > 1$
  - The AirSim API cannot maintain stable frame rates if this is not done
- UAV controlled via RC or the API

Level Design and Image Capturing (continued)

- Time Dilation is employed while capturing images
  - Time dilation implies "slowing down" the simulation world
  - 1 second elapsed in simulation world clock $= n$ seconds elapsed in real world clock; $n > 1$
  - The AirSim API cannot maintain stable frame rates if this is not done
- UAV controlled via RC or the API
  - Move around the world to capture images of crowd groups

Level Design and Image Capturing (continued)

- Time Dilation is employed while capturing images
  - Time dilation implies "slowing down" the simulation world
  - 1 second elapsed in simulation world clock $= n$ seconds elapsed in real world clock; $n > 1$
  - The AirSim API cannot maintain stable frame rates if this is not done
- UAV controlled via RC or the API
  - Move around the world to capture images of crowd groups
  - Possible to maintain stable viewpoint at static height as well

Ground Truth Generation Pseudo code

```
define compute_ground_truth(crowd_bounding_boxes, overlap_threshold, max_iterations):

     filtered_boxes = {}

     while current_iteration < max_iterations:
          foreach box in crowd_bounding_boxes:
               foreach other_box ≠ box in crowd_bounding_boxes:
                    if overlap ≥ overlap_threshold:
                         combine box and other_box into one box
                    else:
                         pick the box with the highest area
                         discard the other box with smaller area
                         add the picked box to filtered_boxes

          crowd_bounding_boxes = filtered_boxes
          current_iteration += 1

     if filtered_boxes.length() is 1:
          return filtered_boxes[0]
     else:
          return the box in filtered_boxes with the largest area

end compute_ground_truth
```

Generated Ground Truth Evaluation

- We hire individuals to annotate the dataset

Generated Ground Truth Evaluation

- We hire individuals to annotate the dataset
  - They should place bounding box around the crowd group they believe to be the largest

Generated Ground Truth Evaluation

- We hire individuals to annotate the dataset
  - They should place bounding box around the crowd group they believe to be the largest
  - This is subjective by nature

Generated Ground Truth Evaluation

- We hire individuals to annotate the dataset
  - They should place bounding box around the crowd group they believe to be the largest
  - This is subjective by nature
- We measure the Intersection-over-Union of:

Generated Ground Truth Evaluation

- We hire individuals to annotate the dataset
  - They should place bounding box around the crowd group they believe to be the largest
  - This is subjective by nature
- We measure the Intersection-over-Union of:
  - The generated ground truth bounding box of each frame

Generated Ground Truth Evaluation

- We hire individuals to annotate the dataset
  - They should place bounding box around the crowd group they believe to be the largest
  - This is subjective by nature
- We measure the Intersection-over-Union of:
  - The generated ground truth bounding box of each frame
  - The human annotated bounding box of each frame

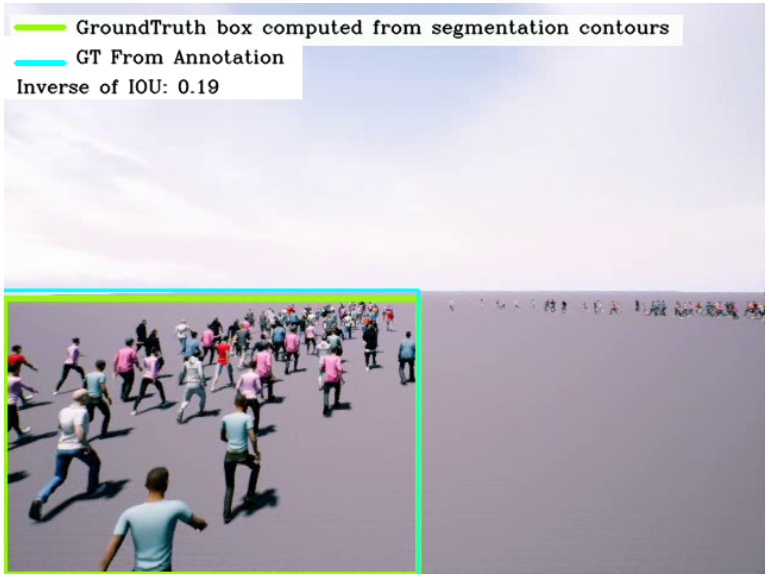
Generated Ground Truth Evaluation

- We hire individuals to annotate the dataset
  - They should place bounding box around the crowd group they believe to be the largest
  - This is subjective by nature
- We measure the Intersection-over-Union of:
  - The generated ground truth bounding box of each frame
  - The human annotated bounding box of each frame
- Compute 1 - IOU (Inverse of IOU / Overlap Error)

Generated Ground Truth Evaluation

- We hire individuals to annotate the dataset
  - They should place bounding box around the crowd group they believe to be the largest
  - This is subjective by nature
- We measure the Intersection-over-Union of:
  - The generated ground truth bounding box of each frame
  - The human annotated bounding box of each frame
- Compute 1 - IOU (Inverse of IOU / Overlap Error)
- Compute the average of errors in each sequence of frames

Ground Truth Results



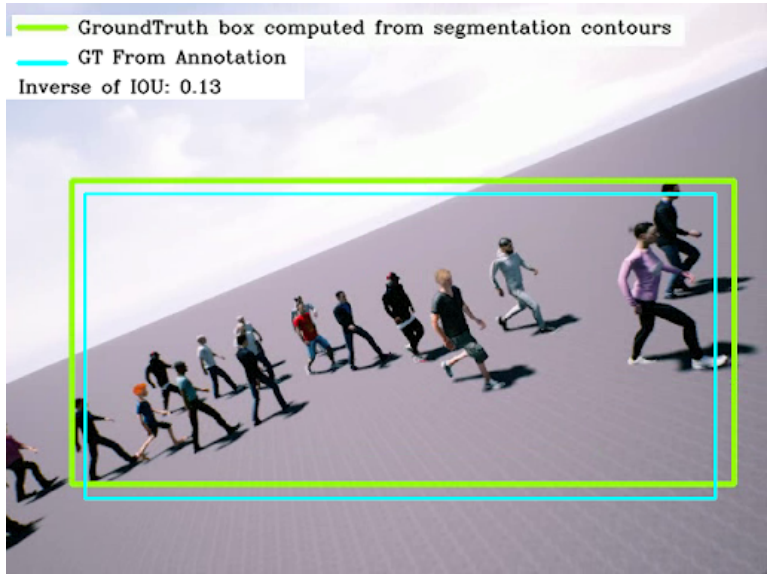


Figure: Good cases where human annotation and computed ground truth agree

Ground Truth Results (continued)

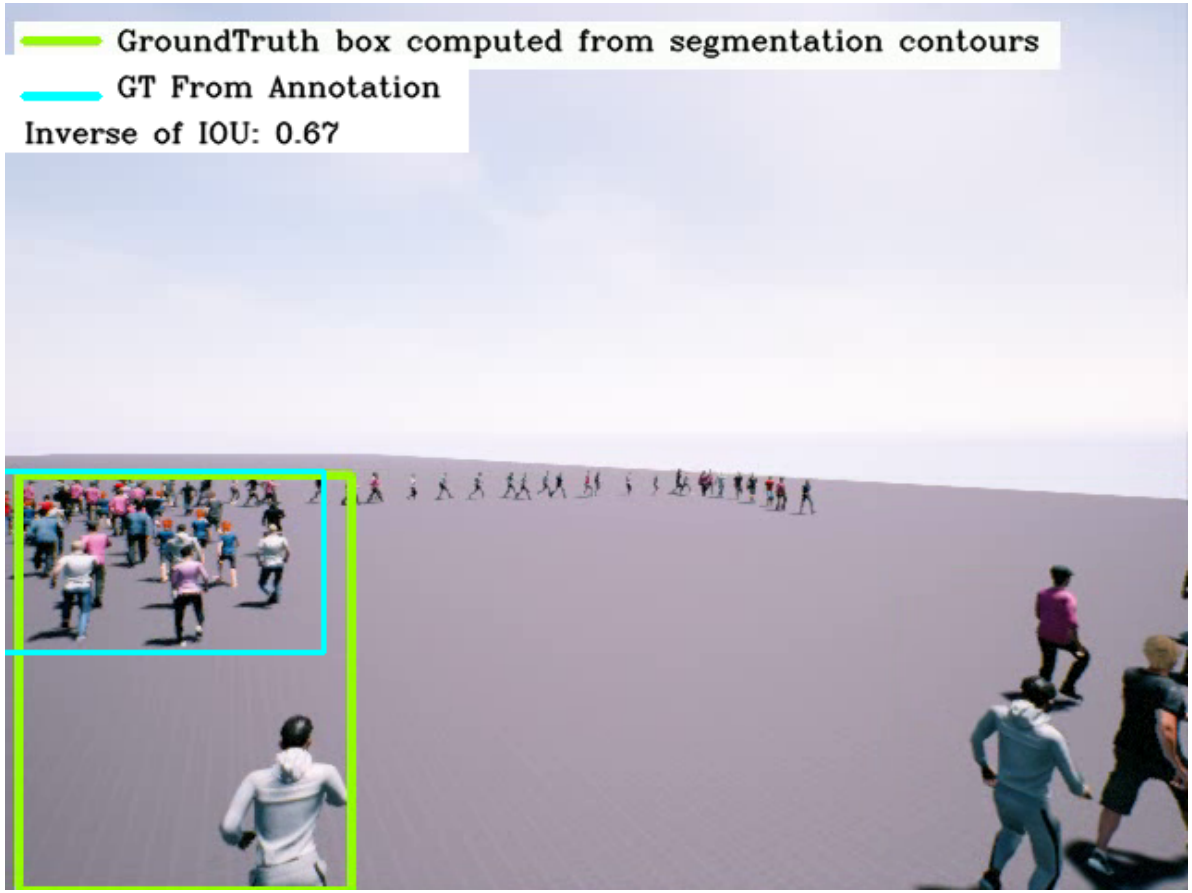


Figure: The algorithm picks a larger box, where human annotator believes the person at the bottom should be skipped for a proper box

Ground Truth Results (continued)

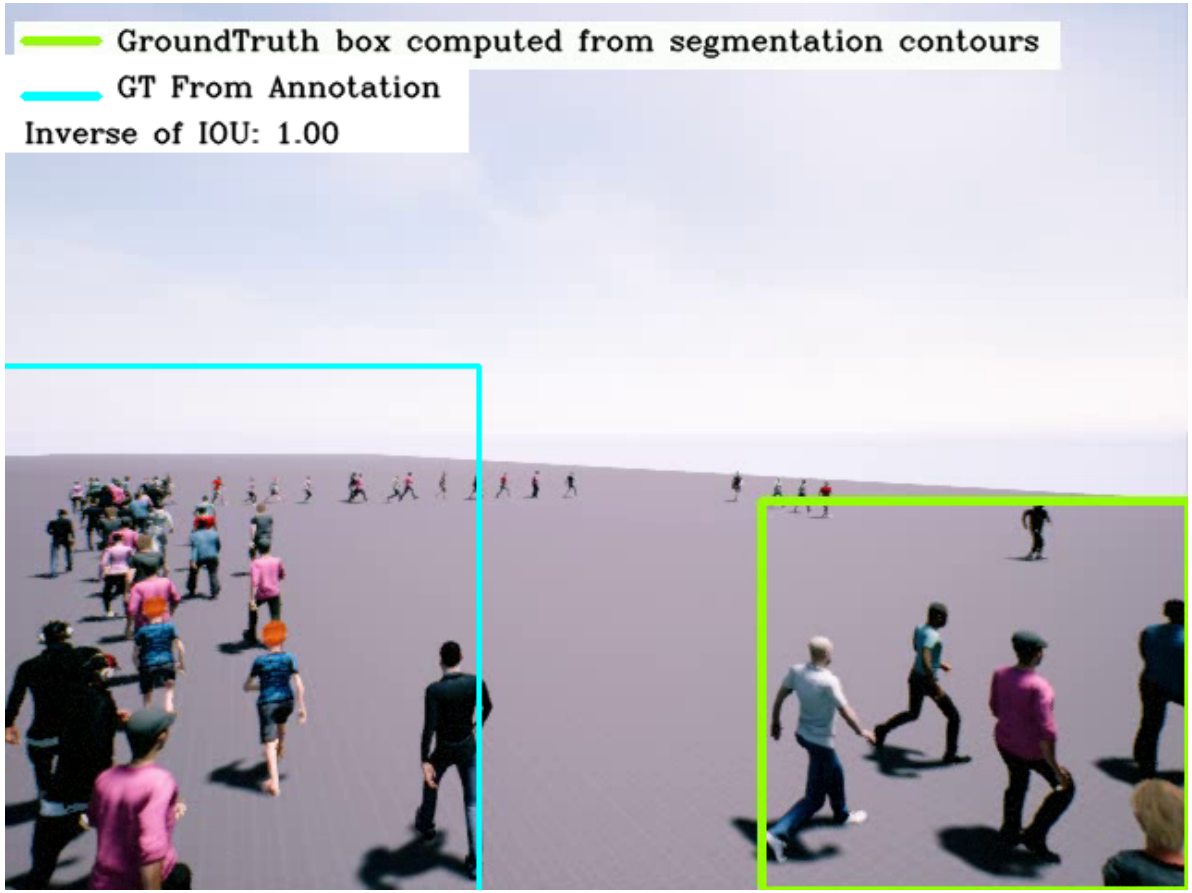


Figure: The algorithm fails to pick the proper groups of crowds and ends up selecting a smaller crowd group

If Ground Truth can be computed, why a tracking algorithm is required?

- A valid question

If Ground Truth can be computed, why a tracking algorithm is required?

- A valid question
- Vision data based ground truth

If Ground Truth can be computed, why a tracking algorithm is required?

- A valid question
- Vision data based ground truth
- Erratic at times due to changes in vision properties

If Ground Truth can be computed, why a tracking algorithm is required?

- A valid question
- Vision data based ground truth
- Erratic at times due to changes in vision properties
- By contrast, a tracking algorithm

If Ground Truth can be computed, why a tracking algorithm is required?

- A valid question
- Vision data based ground truth
- Erratic at times due to changes in vision properties
- By contrast, a tracking algorithm
  - can learn the changes in the patterns

If Ground Truth can be computed, why a tracking algorithm is required?

- A valid question
- Vision data based ground truth
- Erratic at times due to changes in vision properties
- By contrast, a tracking algorithm
  - can learn the changes in the patterns
  - can also maintain a stable bounding box on the target for a longer time

If Ground Truth can be computed, why a tracking algorithm is required?

- A valid question
- Vision data based ground truth
- Erratic at times due to changes in vision properties
- By contrast, a tracking algorithm
  - can learn the changes in the patterns
  - can also maintain a stable bounding box on the target for a longer time
  - visual servoing techniques are benefitted from stability of the bounding box

Our Tracking Algorithm: Hybrid Motion Pooling

Baseline: Discriminative Model Prediction (DiMP)

- Has two fundamental components: model initializer and optimizer.

Baseline: Discriminative Model Prediction (DiMP)

- Has two fundamental components: model initializer and optimizer.
- Trained model weight: $f = \rho(\chi_{sample})$

Baseline: Discriminative Model Prediction (DiMP)

- Has two fundamental components: model initializer and optimizer.
- Trained model weight: $f = \rho(\chi_{sample})$
- Training sample pairs: $\chi_{sample} = \{(v_j, c_j)\}_{j=1}^{m}$

Baseline: Discriminative Model Prediction (DiMP)

- Has two fundamental components: model initializer and optimizer.
- Trained model weight: $f = \rho(\chi_{sample})$
- Training sample pairs: $\chi_{sample} = \{(v_j, c_j)\}_{j=1}^{m}$
- v_j: extracted visual features. $c_j \in \mathbb{R}^2$: centroid of target.

Baseline: Discriminative Model Prediction (DiMP)

- Has two fundamental components: model initializer and optimizer.
- Trained model weight: $f = \rho(\chi_{sample})$
- Training sample pairs: $\chi_{sample} = \{(v_j, c_j)\}_{j=1}^{m}$
- v_j: extracted visual features. $c_j \in \mathbb{R}^2$: centroid of target.
- Loss is calculated as:

$$\ell(f) = \frac{1}{|\chi_{sample}|} \sum_{j=1}^{m} \|\nabla(v_j * f, c_j)\|^2 + \|\tau f\|^2$$

Baseline: Discriminative Model Prediction (DiMP)

- Has two fundamental components: model initializer and optimizer.
- Trained model weight: $f = \rho(\chi_{sample})$
- Training sample pairs: $\chi_{sample} = \{(v_j, c_j)\}_{j=1}^{m}$
- v_j: extracted visual features. $c_j \in \mathbb{R}^2$: centroid of target.
- Loss is calculated as:

$$\ell(f) = \frac{1}{|\chi_{sample}|} \sum_{j=1}^{m} \|\nabla(v_j * f, c_j)\|^2 + \|\tau f\|^2$$

- f is iteratively updated offline based on m samples.

Baseline: Discriminative Model Prediction (DiMP)

- Has two fundamental components: model initializer and optimizer.
- Trained model weight: $f = \rho(\chi_{sample})$
- Training sample pairs: $\chi_{sample} = \{(v_j, c_j)\}_{j=1}^{m}$
- v_j: extracted visual features. $c_j \in \mathbb{R}^2$: centroid of target.
- Loss is calculated as:

$$\ell(f) = \frac{1}{|\chi_{sample}|} \sum_{j=1}^{m} \|\nabla(v_j * f, c_j)\|^2 + \|\tau f\|^2$$

- f is iteratively updated offline based on m samples.
- Issue: Can not scale to groups or multiple human interactions.

Baseline: Discriminative Model Prediction (DiMP)

- Has two fundamental components: model initializer and optimizer.
- Trained model weight: $f = \rho(\chi_{sample})$
- Training sample pairs: $\chi_{sample} = \{(v_j, c_j)\}_{j=1}^{m}$
- v_j: extracted visual features. $c_j \in \mathbb{R}^2$: centroid of target.
- Loss is calculated as:

$$\ell(f) = \frac{1}{|\chi_{sample}|} \sum_{j=1}^{m} \|\nabla(v_j * f, c_j)\|^2 + \|\tau f\|^2$$

- f is iteratively updated offline based on m samples.
- Issue: Can not scale to groups or multiple human interactions.
- Solution: Incorporate Spatio-Temporal Graphs!

Spatial Graph Learning

- Goal: Capture spatial interaction between humans by building a graph Φ^S.

Spatial Graph Learning

- Goal: Capture spatial interaction between humans by building a graph Φ^S.
- At time step t: $\gamma_t = \{(\vartheta_t^1, b_t^1), (\vartheta_t^2, b_t^2), ..., (\vartheta_t^n, b_t^{n_t})\}$, where $\vartheta_t^i \in \mathbb{R}^{1\times d}$

Spatial Graph Learning

- Goal: Capture spatial interaction between humans by building a graph Φ^S.
- At time step t: $\gamma_t = \{(\vartheta_t^1, b_t^1), (\vartheta_t^2, b_t^2), ..., (\vartheta_t^n, b_t^{n_t})\}$, where $\vartheta_t^i \in \mathbb{R}^{1\times d}$
- Build graph based on Intersection over Union (IoU) weights:

$$\Phi_{t_{ij}}^S = \frac{e^{\zeta_t^{ij}}}{\sum_{j=1}^{n_t} e^{\zeta_t^{ij}}}$$

Spatial Graph Learning

- Goal: Capture spatial interaction between humans by building a graph Φ^S.
- At time step t: $\gamma_t = \{(\vartheta_t^1, b_t^1), (\vartheta_t^2, b_t^2), ..., (\vartheta_t^n, b_t^{n_t})\}$, where $\vartheta_t^i \in \mathbb{R}^{1\times d}$
- Build graph based on Intersection over Union (IoU) weights:

$$\Phi_{t_{ij}}^S = \frac{e^{\zeta_t^{ij}}}{\sum_{j=1}^{n_t} e^{\zeta_t^{ij}}}$$

- $\Phi_{t_{ij}}^S \in \mathbb{R}^{n\times n}$ is spatial adjacency matrix. $\zeta_t^{ij} = IoU(b_t^i, b_t^j)$

Adaptive Crowd Group Learning

- Connect the frames with pair-wise cosine similarity as:

$$\Phi^{T}_{t_{ij}} = \frac{e^{\cos(\vartheta^{i}_{t}, \vartheta^{j}_{t+1})}}{\sum_{j=1}^{n_t+1} e^{\cos(\vartheta^{i}_{t}, \vartheta^{j}_{t+1})}}$$

Adaptive Crowd Group Learning

- Connect the frames with pair-wise cosine similarity as:

$$\Phi_{t_{ij}}^{T} = \frac{e^{\cos(\vartheta_t^i, \vartheta_{t+1}^j)}}{\sum_{j=1}^{n_t+1} e^{\cos(\vartheta_t^i, \vartheta_{t+1}^j)}}$$

- $\Phi_{t_{ij}}^{T} \in \mathbb{R}^{n \times n+1}$ is temporal adjacency matrix element at (i, j)

Adaptive Crowd Group Learning

- Connect the frames with pair-wise cosine similarity as:

$$\Phi^{T}_{t_{ij}} = \frac{e^{\cos(\vartheta^{i}_{t}, \vartheta^{j}_{t+1})}}{\sum_{j=1}^{n_t+1} e^{\cos(\vartheta^{i}_{t}, \vartheta^{j}_{t+1})}}$$

- $\Phi^{T}_{t_{ij}} \in \mathbb{R}^{n \times n+1}$ is temporal adjacency matrix element at (i, j)
- Merge spatial and temporal graph as:

$$\Phi^{ST} = \begin{bmatrix} \Phi^{S}_{1} & \Phi^{T}_{1} & 0 & \cdots & 0 \\ 0 & \Phi^{S}_{2} & \Phi^{T}_{2} & \cdots & 0 \\ 0 & 0 & \Phi^{S}_{3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & \Phi^{S}_{t} \end{bmatrix}$$

Adaptive Crowd Group Learning

- Connect the frames with pair-wise cosine similarity as:

$$\Phi^{T}_{t_{ij}} = \frac{e^{\cos(\vartheta^{i}_{t}, \vartheta^{j}_{t+1})}}{\sum_{j=1}^{n_t+1} e^{\cos(\vartheta^{i}_{t}, \vartheta^{j}_{t+1})}}$$

- $\Phi^{T}_{t_{ij}} \in \mathbb{R}^{n \times n+1}$ is temporal adjacency matrix element at (i, j)
- Merge spatial and temporal graph as:

$$\Phi^{ST} = \begin{bmatrix} \Phi^{S}_{1} & \Phi^{T}_{1} & 0 & \cdots & 0 \\ 0 & \Phi^{S}_{2} & \Phi^{T}_{2} & \cdots & 0 \\ 0 & 0 & \Phi^{S}_{3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & \Phi^{S}_{t} \end{bmatrix}$$

- $\Phi^{ST} \in \mathbb{R}^{N \times N}$ is composed of Φ^{S}_{t} and Φ^{T}_{t}

Adaptive Crowd Group Learning

- Connect the frames with pair-wise cosine similarity as:

$$\Phi^{T}_{t_{ij}} = \frac{e^{\cos(\vartheta^{i}_{t}, \vartheta^{j}_{t+1})}}{\sum_{j=1}^{n_t+1} e^{\cos(\vartheta^{i}_{t}, \vartheta^{j}_{t+1})}}$$

- $\Phi^{T}_{t_{ij}} \in \mathbb{R}^{n \times n+1}$ is temporal adjacency matrix element at (i, j)
- Merge spatial and temporal graph as:

$$\Phi^{ST} = \begin{bmatrix} \Phi^{S}_{1} & \Phi^{T}_{1} & 0 & \cdots & 0 \\ 0 & \Phi^{S}_{2} & \Phi^{T}_{2} & \cdots & 0 \\ 0 & 0 & \Phi^{S}_{3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & \Phi^{S}_{t} \end{bmatrix}$$

- $\Phi^{ST} \in \mathbb{R}^{N \times N}$ is composed of Φ^{S}_{t} and Φ^{T}_{t}
- Update using GCN and get $\Lambda \in \mathbb{R}^{T \times dm}$:

$$\mathbb{C}^{l+1} = \sigma(\mathbb{C}^{l} + \lambda^{-\frac{1}{2}} \Phi^{ST} \lambda^{-\frac{1}{2}} \mathbb{C}^{l} W^{l})$$

Hybrid Motion Pooling

- Combine Λ with f from DiMP at each timestep using low-rank bilinear pooling as:

$$\beta_i = \sigma\left(W_f f\right) \odot \sigma\left(W_\Lambda \Lambda_i\right)$$

Hybrid Motion Pooling

- Combine Λ with f from DiMP at each timestep using low-rank bilinear pooling as:

$$\beta_i = \sigma\left(W_f f\right) \odot \sigma\left(W_\Lambda \Lambda_i\right)$$

- Calculate final filter ∂_T by performing temporal max-pooling:

$$\partial_T = MaxPool_T\left([\beta_1, \beta_2, \ldots, \beta_{T-1}]\right) \in \mathbb{R}^{1 \times D_\Lambda}$$

Hybrid Motion Pooling

- Combine Λ with f from DiMP at each timestep using low-rank bilinear pooling as:

$$\beta_i = \sigma\left(W_f f\right) \odot \sigma\left(W_\Lambda \Lambda_i\right)$$

- Calculate final filter ∂_T by performing temporal max-pooling:

$$\partial_T = MaxPool_T\left([\beta_1, \beta_2, \ldots, \beta_{T-1}]\right) \in \mathbb{R}^{1 \times D_\Lambda}$$

- dimension of D_Λ does not depend on training video length T

Hybrid Motion Pooling

- Combine Λ with f from DiMP at each timestep using low-rank bilinear pooling as:

$$\beta_i = \sigma\left(W_f f\right) \odot \sigma\left(W_\Lambda \Lambda_i\right)$$

- Calculate final filter ∂_T by performing temporal max-pooling:

$$\partial_T = MaxPool_T\left([\beta_1, \beta_2, \ldots, \beta_{T-1}]\right) \in \mathbb{R}^{1 \times D_\Lambda}$$

- dimension of D_Λ does not depend on training video length T
- T: temporal information for HyMP. m: for DiMP

Offline Training

- Initially, pre-train the DiMP weight model f from the UUCT dataset for $T = m = 10$.

Offline Training

- Initially, pre-train the DiMP weight model f from the UUCT dataset for $T = m = 10$.
- For test window pairs, loss is calculated as:

$$\ell_{t\,\mathrm{arg}\,et} = \frac{1}{\mathbb{N}} \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{m} \left\| \kappa \left(v_j * \partial_T^i, g_{c_j} \right) \right\|^2$$

Offline Training

- Initially, pre-train the DiMP weight model f from the UUCT dataset for $T = m = 10$.
- For test window pairs, loss is calculated as:

$$\ell_{target} = \frac{1}{\mathbb{N}} \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{m} \left\| \kappa \left(v_j * \partial_T^i, g_{c_j} \right) \right\|^2$$

- $$\kappa(v, c) = \begin{cases} v - c, & \text{if } c > \varepsilon \\ \max(0, c), & \text{otherwise} \end{cases}$$
- $\ell_{final} = \eta \ell_{target} + \ell_{bb}$

Offline Training

- Initially, pre-train the DiMP weight model f from the UUCT dataset for $T = m = 10$.
- For test window pairs, loss is calculated as:

$$\ell_{target} = \frac{1}{\mathbb{N}} \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{m} \left\| \kappa \left(v_j * \partial_T^i, g_{c_j} \right) \right\|^2$$

- $$\kappa(v, c) = \begin{cases} v - c, & \text{if } c > \varepsilon \\ \max(0, c), & \text{otherwise} \end{cases}$$
- $\ell_{final} = \eta \ell_{target} + \ell_{bb}$
- ℓ_{bb} is learned according to IoU overlap

Online Tracking and Algorithm

Algorithm 1 Online HyMP Tracking Algorithm

Input : Initial Trained Model weight ∂_T, Initial Test sample $\chi_{test} = (v_0, c_0)$. New frames v_q where $q > 0$

Output: Estimated new positions with updated model

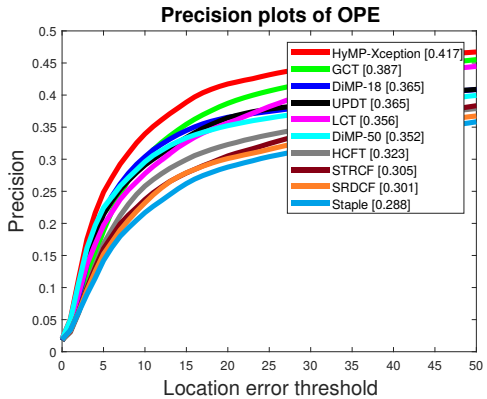
$\chi_{sample} \leftarrow$ augment $\chi_{test} = (v_0, c_0)$ pair into 10 samples
$\gamma_0 \leftarrow$ apply detector and fetch feature, box pair from χ_{test}
$\gamma \leftarrow$ repeat γ_0 10 times
$\partial_T \leftarrow$ forward pass χ_{sample} and γ to the model ρ and perform 10 training iterations
repeat
    `/* bounding box prediction stage */`
    $y^{tc} \in \mathbb{R}^2 \leftarrow$ Perform convolution on $\delta(v_q) * \partial_T$
    $y^{bb} \in \mathbb{R}^4, p_q \leftarrow$ Regress candidate boundinng boxes and choose one with the highest confidence score.
    plot y^{bb} in the frame v_q
    `/* model update step */`
    if $p_q > \varepsilon$ **then**
        `/* append frame (+) to memory */`
        $\chi_{sample} \leftarrow \chi_{sample} + \delta(v_q)$
        $\gamma_t \leftarrow$ feature, box pair for v_q
        $\gamma \leftarrow \gamma - \gamma_t$
    end
    if $length(\chi_{sample}) > 50$ **then**
        `/* remove frame (-) from memory */`
        $\chi_{sample} \leftarrow \chi_{sample} - v_0$
        $\gamma \leftarrow \gamma + \gamma_0$
    end
    if 30 *new frames are appended* **then**
        $\partial_T \leftarrow$ forward pass χ_{sample} and γ to the model ρ and perform 1 training iteration
    end
until *Until novel* v_q *without bounding boxes is received*

Experimental Evaluation

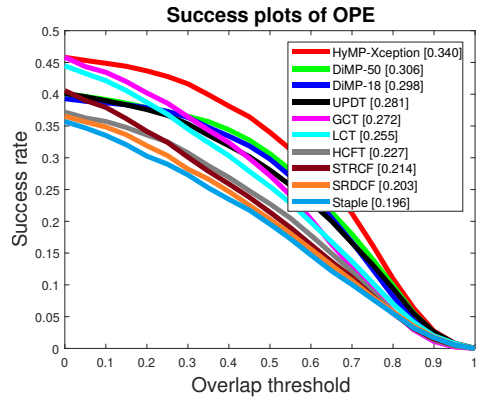
Table: Comparison of algorithms' AUC on the benchmark datasets. For VOT2018, we compared EAO score [KLM+18]. HyMP-X is HyMP with Xception [Cho17] as backbone feature extractor

Tracker	VOT-2018	OTB100	UAV123	UAV20L	DTB70	UUCT
Correlation Filter Trackers						
Staple [BVG+16]	16.9	58.6	45.0	33.1	35.1	19.6
SRDCF [DHSKF15]	11.9	59.8	46.4	34.3	36.3	20.3
STRCF [LTZ+18]	14.3	64.1	48.1	35.4	40.7	20.3
Tracekrs applied Correlation Filter on deeply learned features						
HCFT [MHYY18b]	19.9	64.7	48.6	36.8	41.5	22.7
LCT [MHYY18a]	22.1	65.2	49.4	35.9	43.1	25.5
Tracekrs exploited end-to-end deep learning pipeline						
GCT [GZX19]	27.6	64.8	50.8	46.1	44.2	27.2
UPDT [BJD+18]	38.3	**70.2**	54.5	49.5	45.7	28.1
DiMP-18 [BDGT19]	40.2	66	64.3	51.7	46.9	29.8
DiMP-50 [BDGT19]	43.1	68.4	65.4	52.8	47.5	30.6
End-to-end HyMP Tracker (ours)						
HyMP-18	40.1	66.8	66.2	52.5	47.1	31.7
HyMP-50	42.8	69.1	67.6	52.9	48.6	32.9
HyMP-X	**43.6**	69.4	**68.2**	**53.7**	**49.2**	**34.0**

Algorithm Performance Visualization on UUCT Dataset

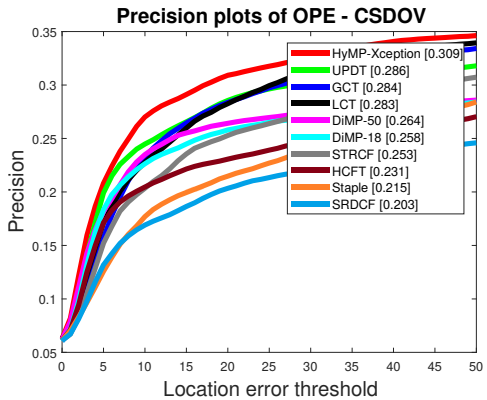


(a) Precision

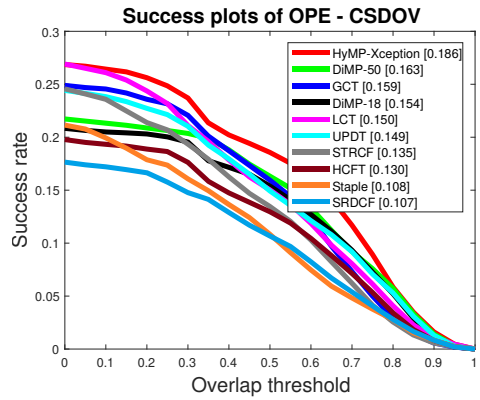


(b) Overlap Success

Figure: Overall benchmark of SOTA trackers on UUCT



(a) Precision



(b) Overlap Success

Figure: Benchmark on Crowd Split and Dispersed Out of View (CSDOV)

Thank You!

References


Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte, *Learning discriminative model prediction for tracking*, Proceedings of the IEEE International Conference on Computer Vision, 2019.


Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg, *Unveiling the power of deep tracking*, Proceedings of the European Conference on Computer Vision (ECCV), 2018.


Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr, *Staple: Complementary learners for real-time tracking*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1401–1409.


François Chollet, *Xception: Deep learning with depthwise separable convolutions*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.


Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, *Learning spatially regularized correlation filters for visual tracking*, Proceedings of the IEEE international conference on computer vision, 2015.


Junyu Gao, Tianzhu Zhang, and Changsheng Xu, *Graph convolutional tracking*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 4649–4659.


Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka ˇCehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al., *The sixth visual object tracking vot2018 challenge results*, Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 0–0.


Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang, *Learning spatial-temporal regularized correlation filters for visual tracking*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.


Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, *Adaptive correlation filters with long-term and short-term memory for object tracking*, International Journal of Computer Vision **126** (2018), no. 8.


———, *Robust visual tracking via hierarchical convolutional features*, IEEE transactions on pattern analysis and machine intelligence (2018) (en).