2D Visual Object Tracking summary

I. Karakostas, V. Mygdalis, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 3.8.4

Artificial Intelligence &
Information Analysis Lab

2D Object Tracking

- **Introduction**
- Prediction in object tracking
- Feature Point based trackers
- Region similarity trackers
- Correlation trackers
- Object Detection Performance Metrics

2D Object Tracking

- Video tracking is the process of locating a moving object (or multiple objects) over time using a camera
- Variety of uses:
  - Human-computer interaction;
  - Security and surveillance;
  - Video communication and compression;
  - Traffic control;
  - Medical imaging.

Target/object examples

- Athletes, boats, bicycles.

2D Visual Object Tracking

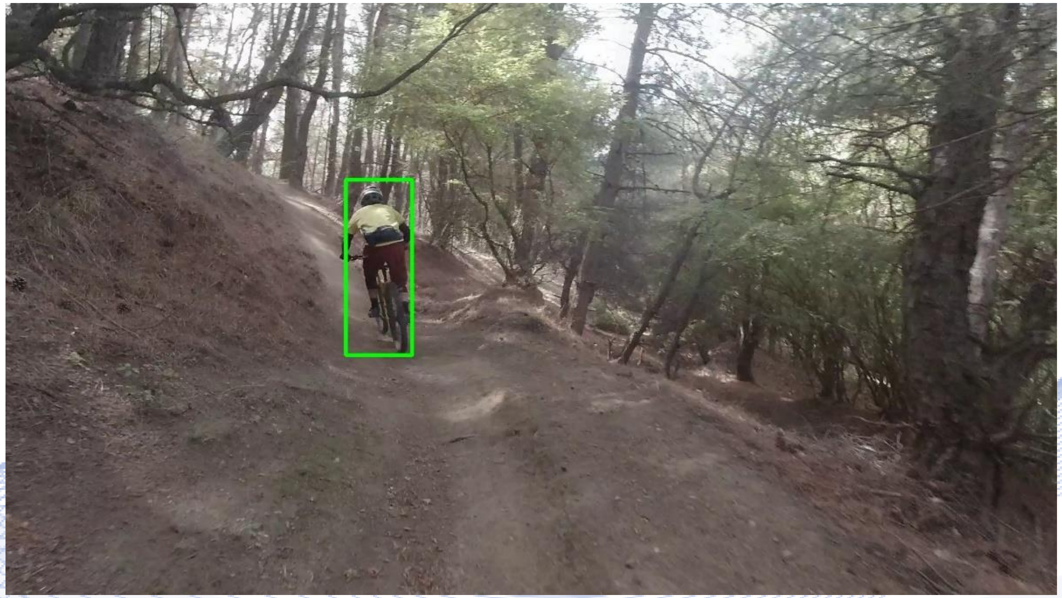
- Problem statement:
  - To track a target/object (e.g. human face) image in each video frame and localize its ***Region-Of-Interest*** (***ROI***).
  - To track the detected object over the video frames.

1st frame | 6th frame | 11th frame | 16th frame

2D Object Tracking

- ROI is typically a bounding box at time t defined by:
  - its center/size parameter vector $[x_c, y_c, w, h]^T$ or
  - the upper left lower right rectangle coordinates $[x_l, y_l, x_r, y_r]$.
- ***Object ROI center***: $\mathbf{c} = [x_c, y_c]^T$.
- ***Object trajectory***: object ROI center coordinates over time.
- ***Moving region***: a series of tracked object ROIs over time.
- ***Object instance***: object region ROI plus other info.

2D Object Tracking



Object Region-Of-Interest (ROI).

2D Object Tracking

- ***2D visual object tracking*** is performed ***on the image plane***:
  - object ROI coordinates and trajectory are defined in (x, y, t) image plane coordinates (typically in pixels, sec).
- 2D visual object racking associates each detected object ROI in the current video frame with one in the next video frame.

2D Object Tracking

- ***3D object tracking*** is performed on a world coordinate system: (X, Y, Z, t) (in meters, sec).
- ***3D object following*** is a control problem, ensuring a vehicle follows a physical object moving in a world coordinate system: (X, Y, Z, t).

2D Object tracking requirements

In order to track a moving object, a tracker has to confront:

- 3D geometric solid object motion (3D translations, rotations) causing 2D object image transformations:
  - notably 2D translation, rotation, scaling or projective transformations object scaling.

- Effects of camera motion and/or parameter change:
  - zooming, global motion field;

2D Object tracking requirements

- In order to track a moving object, a tracker has to confront:
  - Partial occlusion,
  - Object image deformation,
  - Motion blur,
  - Fast object image motion,
  - Illumination variations,
  - Background clutter.

2D Object Tracking

2D Object Tracking

Detection-tracking loop:

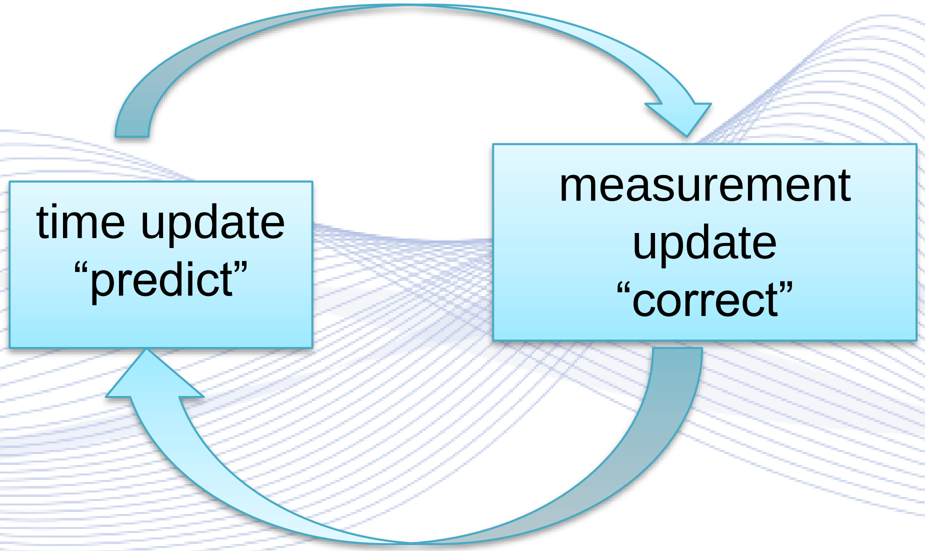
- ***Object (re)detection***.
- ***Object position prediction*** in the next frame and search region initialization (optional).
- ***Object localization*** in the next video frame:
  - Feature/Similarity/Correlation matching.
  - Handling tracking failure (optional):
    - Occlusion detection and handling;
    - Object/background model update;
    - Background discrimination.

2D Object Tracking

- Introduction
- **Prediction in object tracking**
- Feature Point based trackers
- Region similarity trackers
- Correlation trackers
- Object Detection Performance Metrics

Object position prediction

Kalman filter for object position/velocity parameter prediction.



Object position prediction

Kalman filter for object position prediction:

- ***Object state vector***: $\mathbf{x}_t = [x, y, v_x, v_y]^T$ (2D image plane position, velocity).
- Motion state estimation model: $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{n}_t$
- Position prediction: $\hat{\mathbf{x}}_{t+1} = \mathbf{A}\hat{\mathbf{x}}_t,$

$$\widehat{\mathbf{P}}_{t+1} = \mathbf{A}\widehat{\mathbf{P}}_t\mathbf{A}^T + \mathbf{Q}_s$$

Object position prediction

- Measurement model:

$$\mathbf{z}_{t+1} = \mathbf{H}\mathbf{x}_{t+1} + \mathbf{v}_{t+1},$$

$$\mathbf{K}_{t+1} = \widehat{\mathbf{P}}_t\mathbf{H}^T\left(\mathbf{H}\widehat{\mathbf{P}}_t\mathbf{H}^T + \mathbf{Q}_m\right)^{-1}.$$

- Adjustment of $\mathbf{P}_{t+1}$:

$$\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t + \mathbf{K}_{t+1}(\mathbf{z}_{t+1} - \mathbf{H}\hat{\mathbf{x}}_{t+1}),$$

$$\mathbf{P}_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H})\mathbf{P}_{t+1}.$$

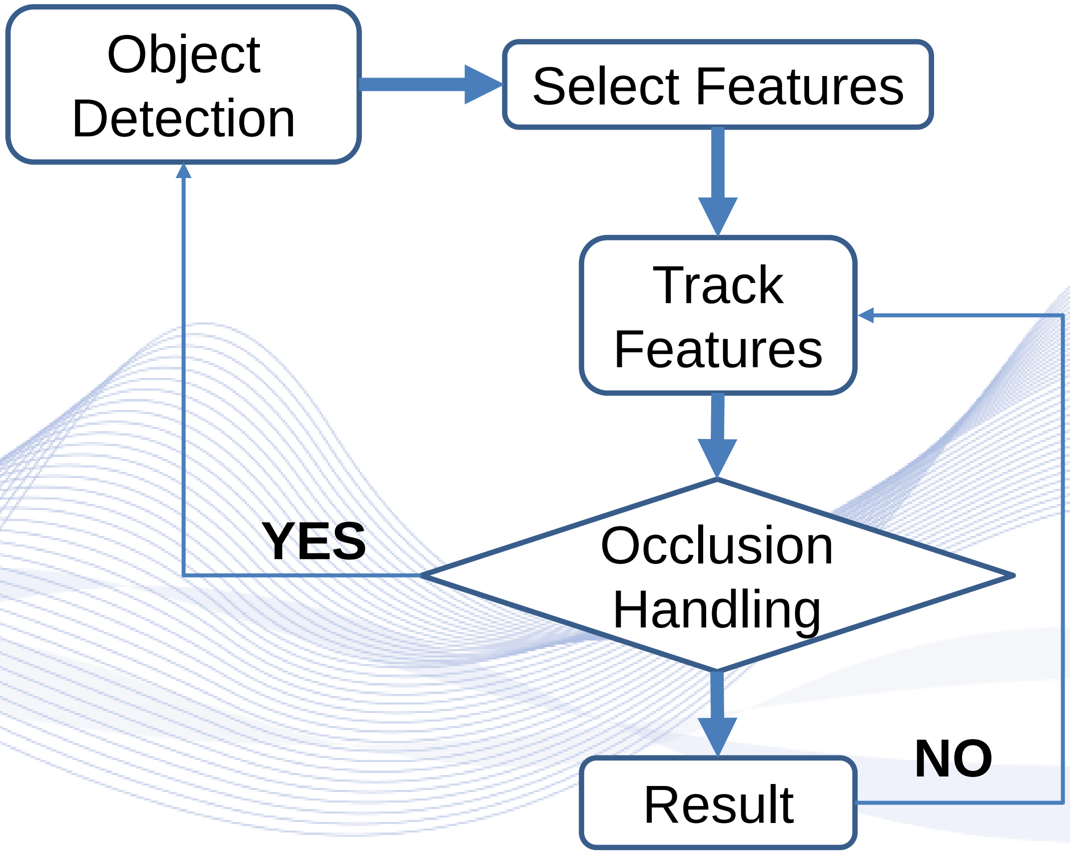
2D Object Tracking

- Introduction
- Prediction in object tracking
- **Feature Point based trackers**
- Region similarity trackers
- Correlation trackers
- Object Detection Performance Metrics

Feature Point based Tracking

Feature point based face tracking:

- Sparse image feature generation.
- Various image features can be used.
- Features may have semantic meaning (e.g., mouth corner).
- Feature points (landmarks) are individually tracked.



2D Object Tracking

- Introduction
- Prediction in object tracking
- Feature Point based trackers
- **Region similarity trackers**
- Correlation trackers
- Object Detection Performance Metrics

Artificial Intelligence & Information Analysis Lab

Region Similarity Tracking

- An object ROI is detected.
- For this ROI, feature vectors are calculated:
    - image color similarity features, e.g., color histogram;
    - structure similarity features (e.g., LSK).
  - They form the object image model.
- A search region is defined around the predicted object position.
    - The search region is divided in overlapping patches.

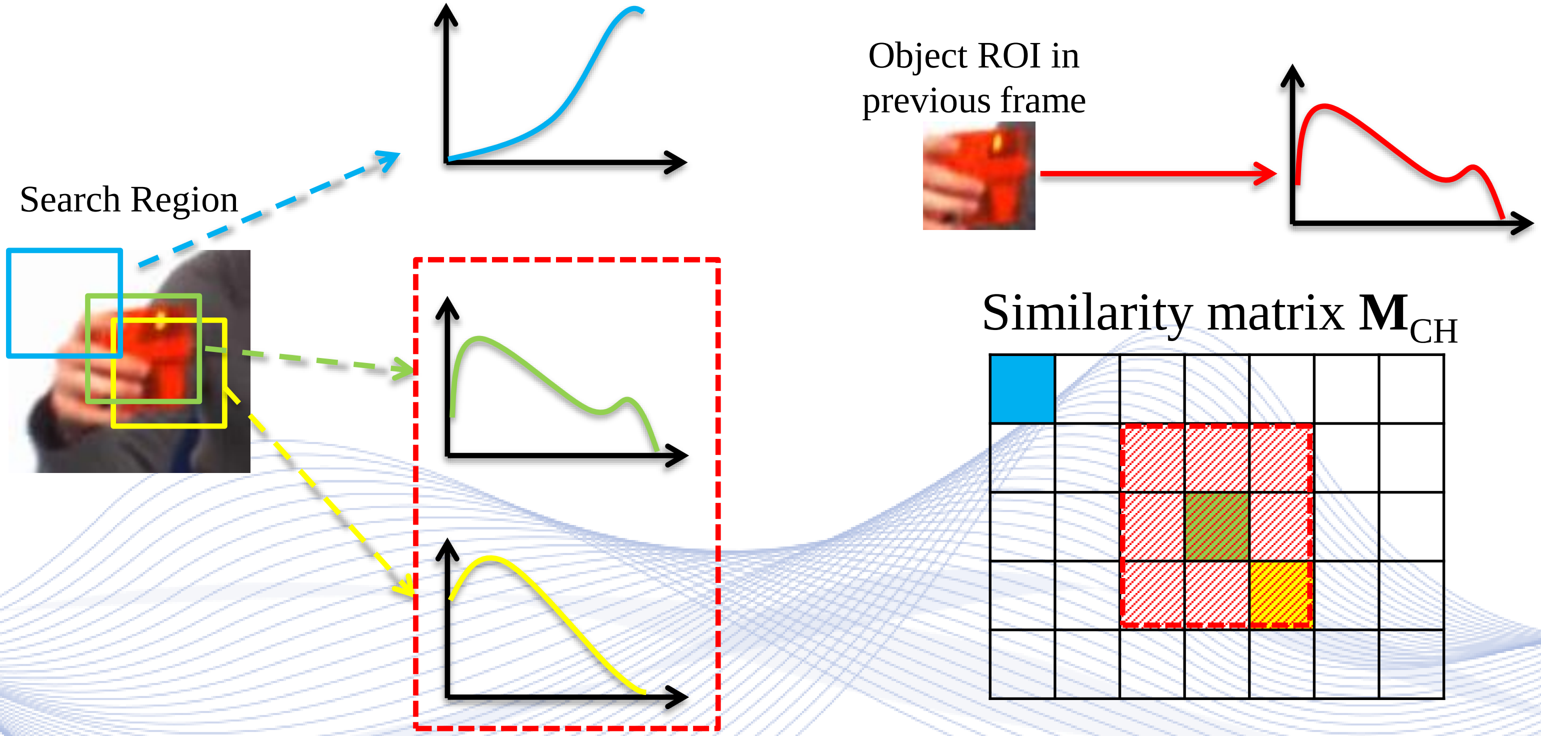
Region Similarity Tracking

- Feature vector similarity between:
  - the ones of image patches
  - the ones of the image model.
- If feature vector similarity is big:
  - tracking is successful.

- When an object appearance change is detected, the object model is updated.

Region Similarity Tracking

- Color-histogram (CH) similarity

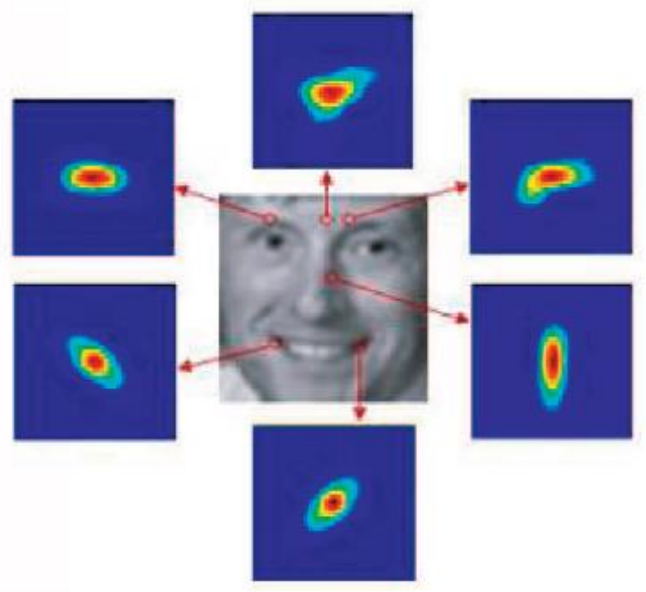


- Only ROIs with the most similar color histograms are retained.

Region Similarity Tracking

Local Steering Kernels (***LSKs***)

- They are a non-linear combination of weighted distances between a pixel and its surrounding pixels.
- They exploit both spatial and edge detection information.
- One LSK vector per pixel is derived.
- LSKs are invariant to brightness & contrast variations and noise.



LSKs [SEO2010].

Region Similarity Tracking

- ***Cosine similarity*** between histogram vectors $\mathbf{h}_1$ and $\mathbf{h}_2$:

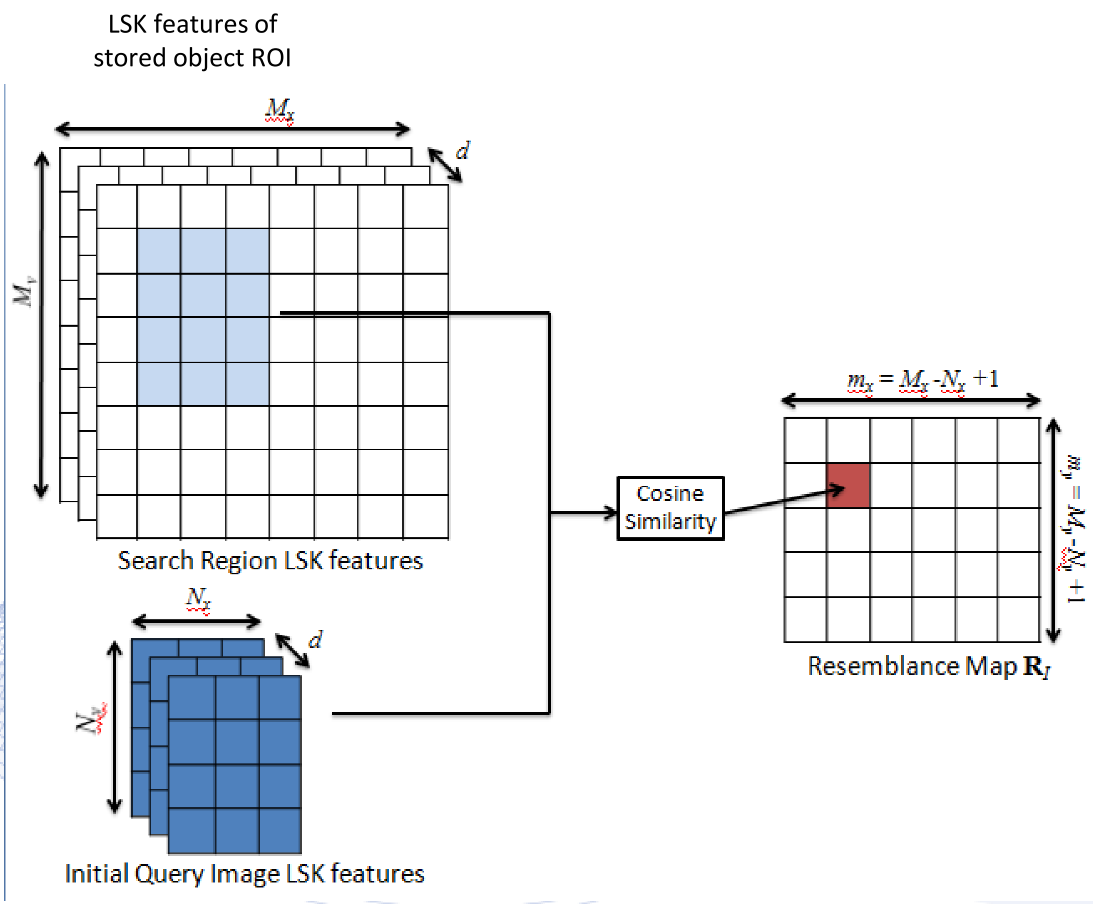
$$S = \frac{s^2}{1-s^2},$$

$$s(\mathbf{h}_1, \mathbf{h}_2) = \cos(\theta) = \frac{\mathbf{h}_1^T \mathbf{h}_2}{\|\mathbf{h}_1\| \|\mathbf{h}_2\|}.$$

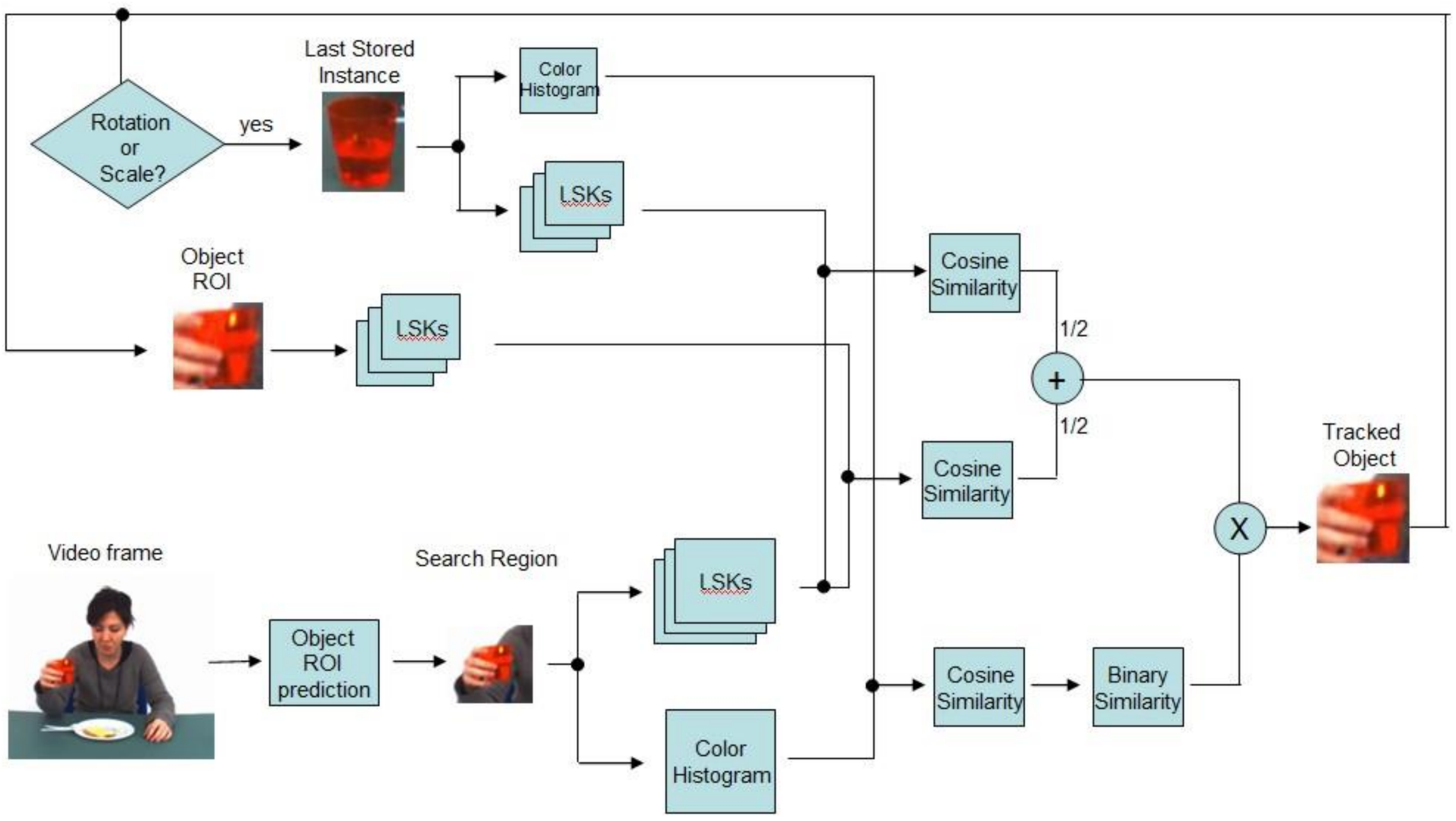
- It can be used both for image color and structure feature vectors.

Region Similarity Tracking

- Extract LSKs resemblance map $\mathbf{R}_I$ to a stored object ROI in a previous frame (***object appearance model***).

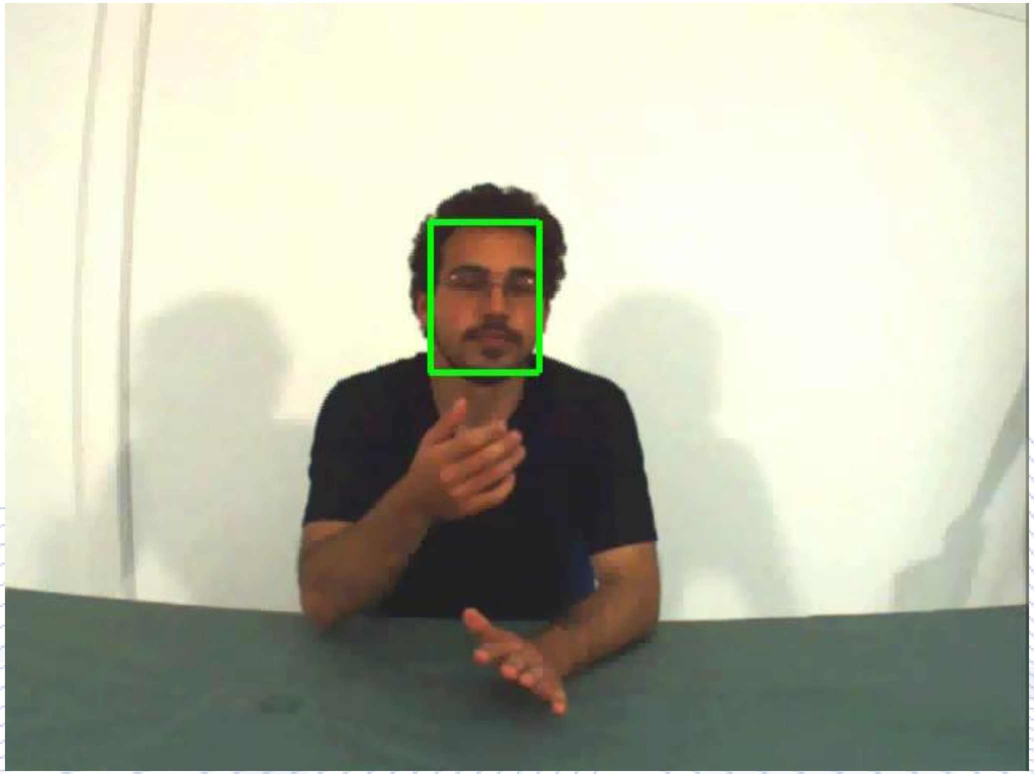


Region Similarity Tracking



Overall LSK tracker block diagram.

Region Similarity Tracking

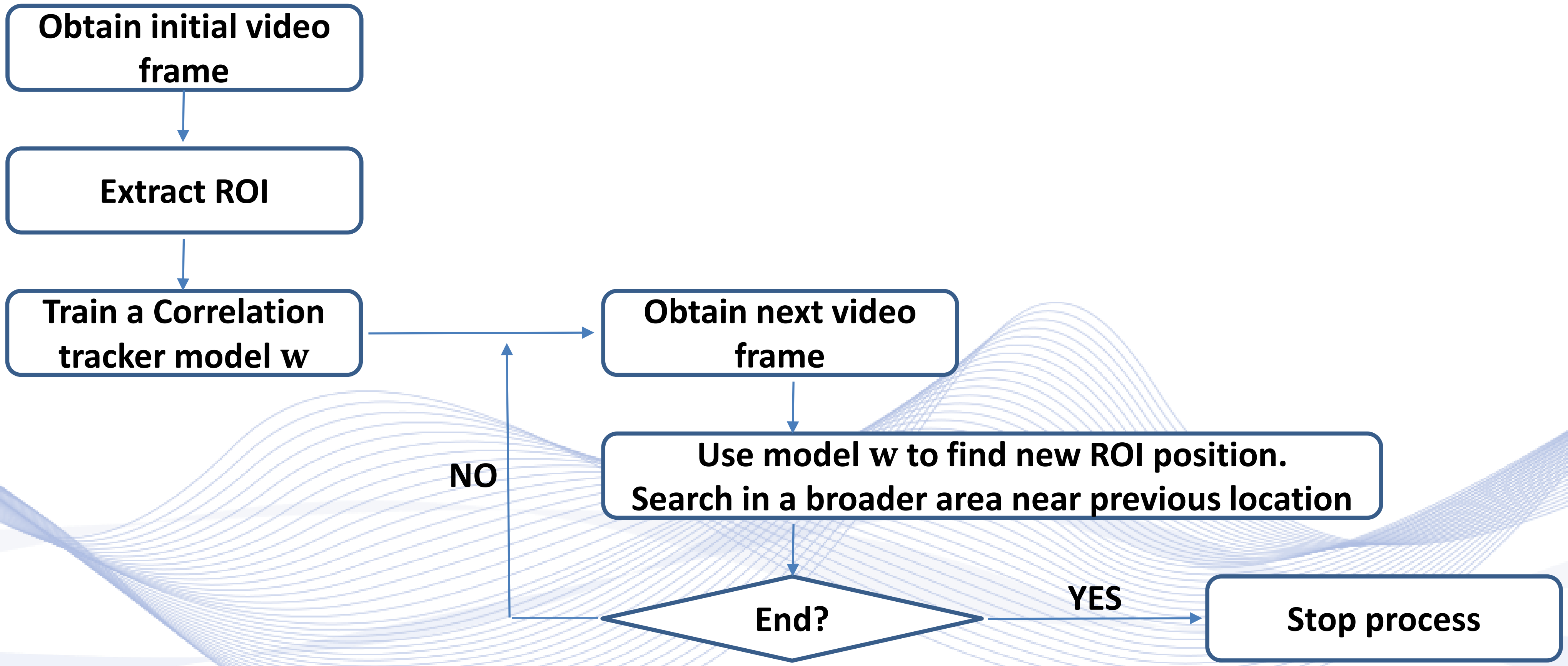


LSK tracker results.

2D Object Tracking

- Introduction
- Prediction in object tracking
- Feature Point based trackers
- Region similarity trackers
- **Correlation trackers**
- Object Detection Performance Metrics

Correlation trackers



Correlation trackers

Kernelized Correlation Filter (***KCF***) ***tracker***:

- KCF is a very fast video tracker. Ideal for embedded system applications.
- It can be adapted to use various features (pixel intensity, HOG, etc.) or even use deep features provided by Convolutional Neural Networks (CNNs).
- Standard KCF has no scaling adaptation mechanism but can be modified to this end.

KCF Tracker

- Various image descriptors can be used with KCF.

- Current implementations deploy:
  - Grayscale features
  - HOG
  - Features calculated with deep neural networks

KCF Tracker

Linear regression

- Goal: training a linear function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ that minimizes the objective function:

$$\min_{\mathbf{w}} \sum_{i=1}^{N} (f(\mathbf{x}_i) - y_i)^2 + \lambda \|\mathbf{w}\|^2.$$

- $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \dots, N$: object ROI feature vectors
- y_i, $i = 1, \dots, N$: regression targets
- $\mathbf{w}$: ***KCF tracker model*** (unknown KCF weight vector).
- λ: regularization parameter.

KCF Tracker

Linear regression solution:

$$\mathbf{w} = (\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{y},$$

- $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N]$: $N \times n$ ***data matrix***.
- $\mathbf{y} = [y_1, \dots, y_N]^T$: regression target vector.
- $\mathbf{w}$: unknown weight vector.
- Regularized pseudoinversion is used.
- It forms the theoretical basis for KCF object tracking.
- In the following, $\mathbf{X}^T$ will be used in the place of $\mathbf{X}$, as is the typical notation in the literature.

KCF Tracker

- Mapping the linear regression solution:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y},$$

to kernel space $\mathbb{R}^L$ produces the kernel coefficient vector $\mathbf{a} = [a_1, \dots a_N]^T$:

$$\mathbf{a} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}.$$

- If $\mathbf{K}$ is circulant, this solution can be diagonalized to obtain a fast solution:

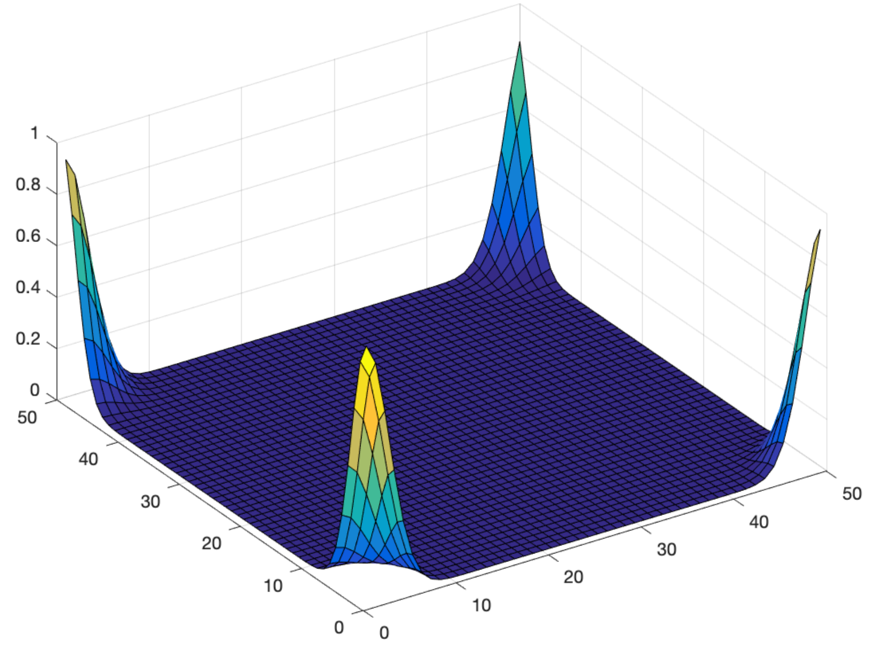
$$\hat{\mathbf{a}} = \hat{\mathbf{y}} \oslash (\hat{\mathbf{k}}_{\mathbf{xx}} + \lambda).$$

- $\hat{\mathbf{k}}_{\mathbf{xx}}$ is the 1D Fourier transform of the N elements $\mathbf{k}_{\mathbf{xx}} = \kappa(\mathbf{x}, \mathbf{x})$.

KCF Tracker

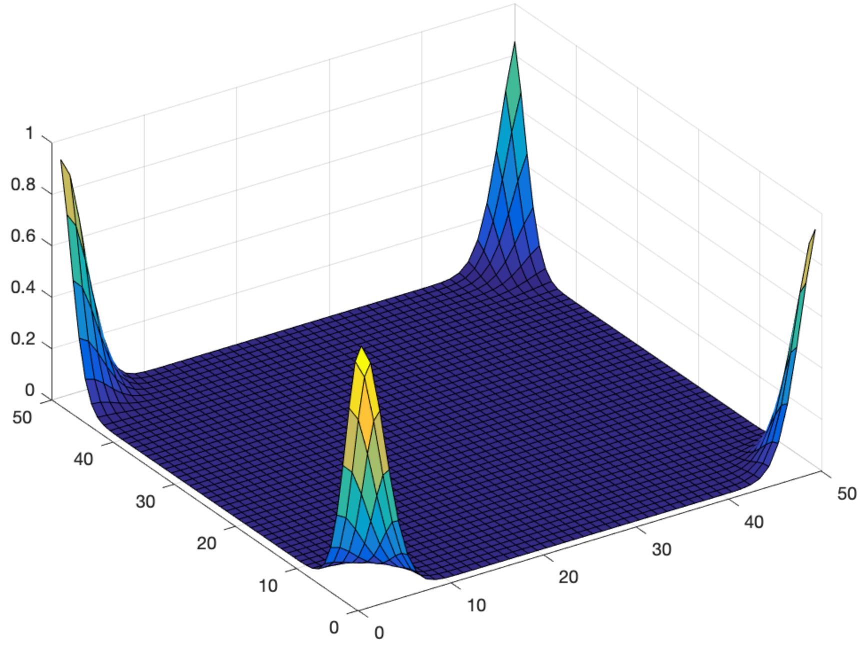
Definition of the regression target $\mathbf{y}$:

- A 2D Gaussian distribution is used.
- Higher value corresponds to the non permuted data vector $\mathbf{x}$.



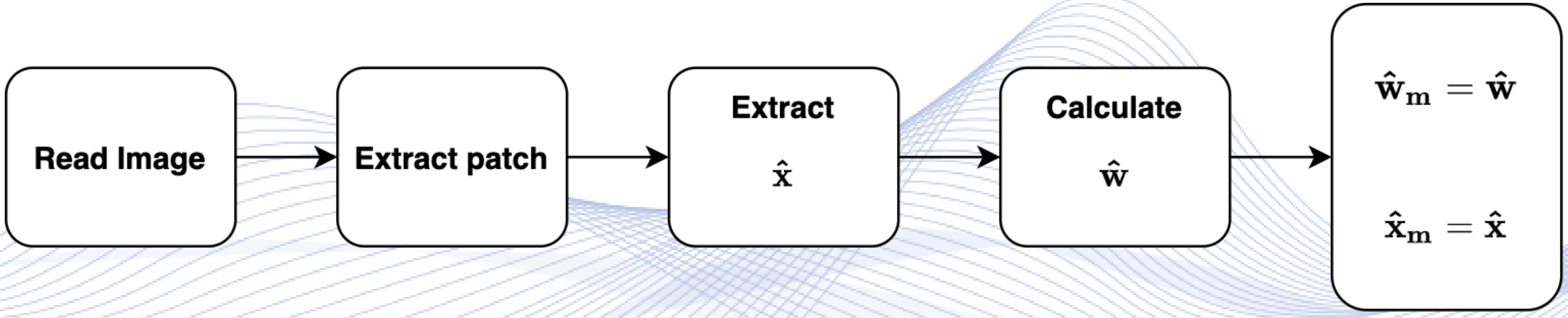
KCF Tracker

- Placing the peak in the middle will unnecessarily cause the detection output to be shifted by half a window.
- Placing the peak at the top-left element (and wrapping around) correctly centers the detection output.



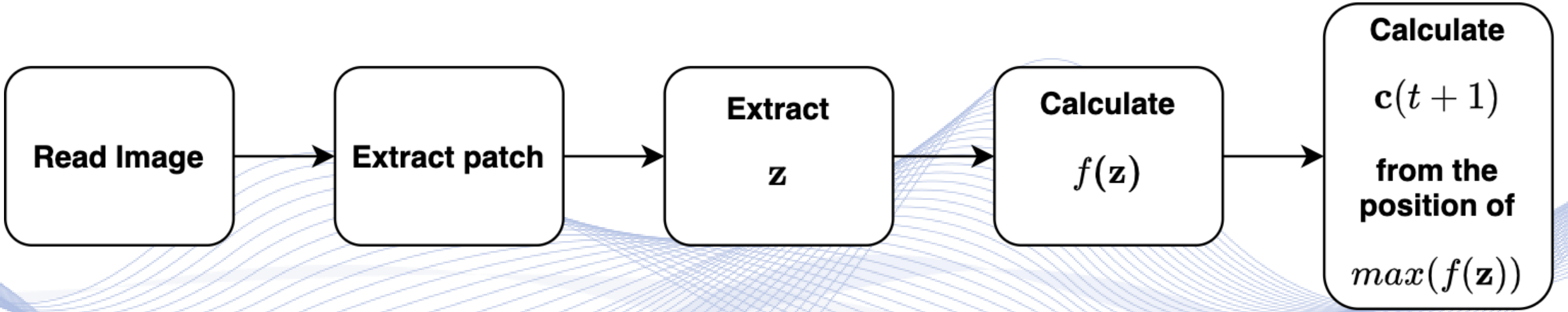
KCF Algorithm implementation

- Initial frame ($t = 0$):
  - Initialize $\hat{\mathbf{w}}$,
  - Initialize $\hat{\mathbf{w}}_m$, $\hat{\mathbf{x}}_m$: interpolated model and target features.



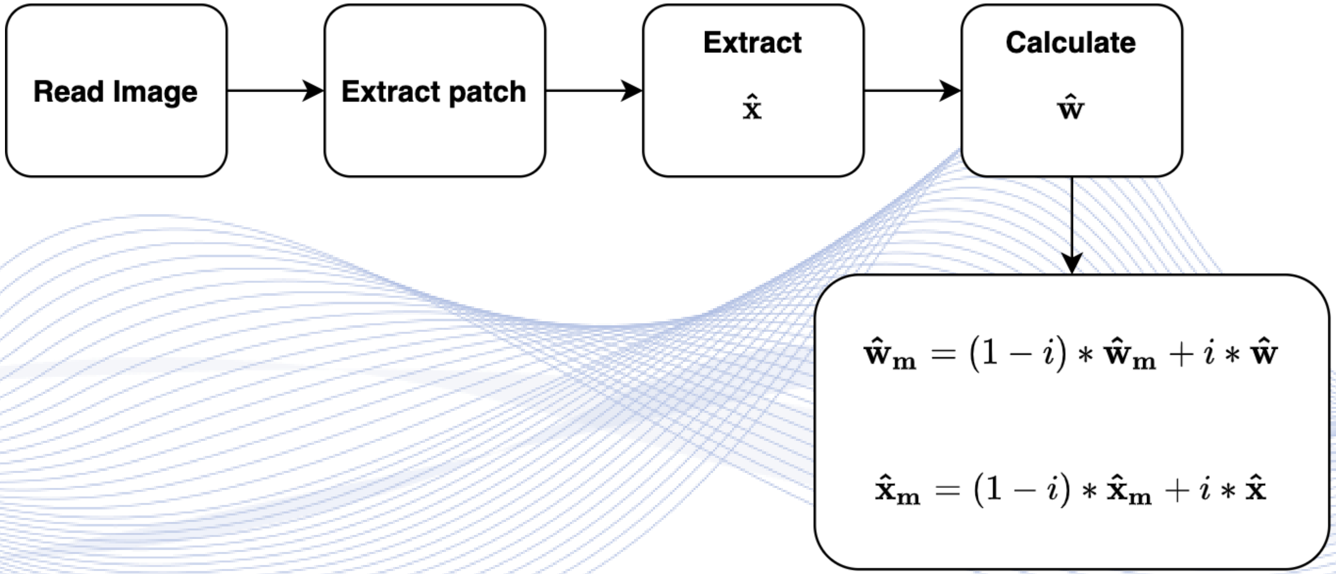
KCF Algorithm implementation

- Next frame ($t = t + 1$)
  - update target position $\mathbf{c}(t+1)$:

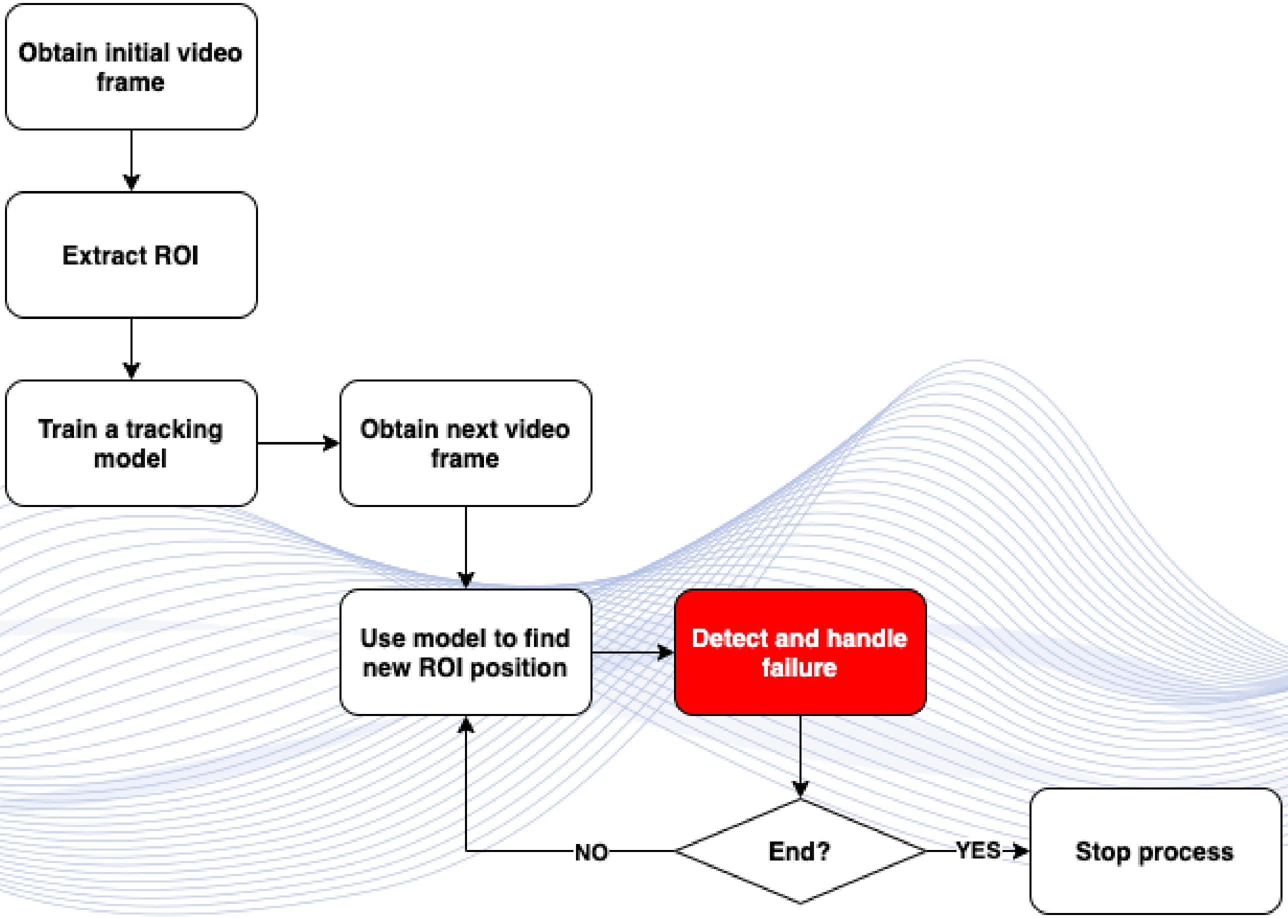


KCF Algorithm implementation

- Next frame ($t = t + 1$)
  - interpolate $\hat{\mathbf{w}}_m, \hat{\mathbf{x}}_m$:



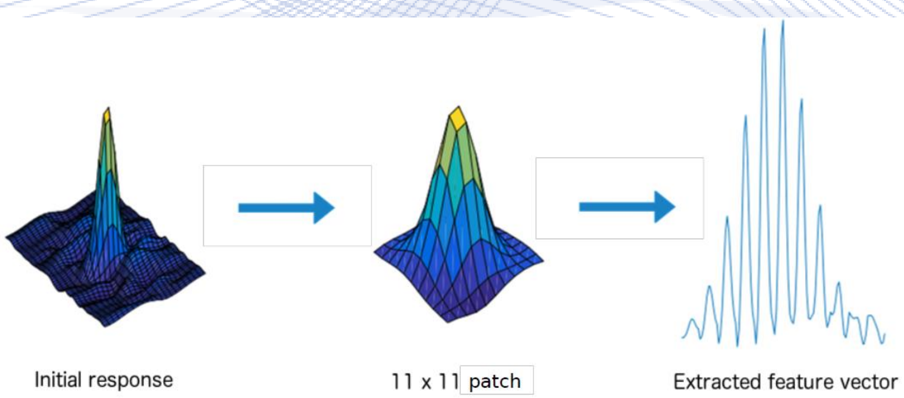
Handling Tracking Failure



Handling Tracking Failure

Occlusion handling (features for SVM classifier):

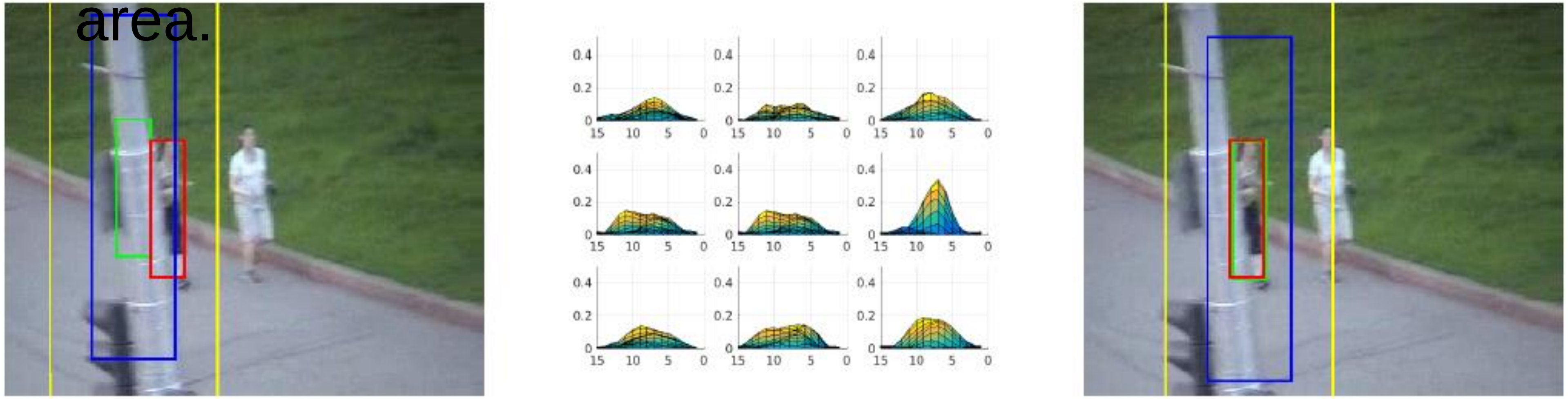
- Obtain the tracker response $f(\mathbf{z})$ on a search region, crop and vectorize the central area (11×11 patch), corresponding to low frequencies.
- When no occlusions or heavy translation occur, response values $f(\mathbf{z})$ are larger in this patch.



Handling Tracking Failure

Object re-detection:

- If occlusion is detected, the tracker model is employed at the ***re-detection area***.
- The re-detection area is larger than the standard search area.



Tracker outputs (green), Window size area (blue), Re-detection area (yellow), target position (red).

Correlation trackers

Baseline Correlation Trackers:

- Minimum Output Sum of Squared Errors (MOSSE).
- Circulant Structure Kernels (CSK).
- Spatio-Temporal Context (STC).
- Kernelized Correlation Filters (KCF) / Dual Correlation Filters (DCF).

Correlation trackers

Scaling Handling Correlation Trackers:

- Discriminative Scale Space Tracker (DSST).
- Scalable Kernel Correlation Filter (SKCF).
- Scale Adaptive with Multiple Features (SAMF).
- Kernelized Correlation Filter with Detection Proposal (KCFDP).

2D Object Tracking

- Introduction
- Prediction in object tracking
- Feature Point based trackers
- Region similarity trackers
- Correlation trackers
- **Object Detection Performance Metrics**

Object Detection Performance Metrics

Intersection over Union (***IoU***):

$$J(\mathcal{A},\mathcal{B}) = |\mathcal{A}\cap\mathcal{B}|/|\mathcal{A}\cup\mathcal{B}|.$$

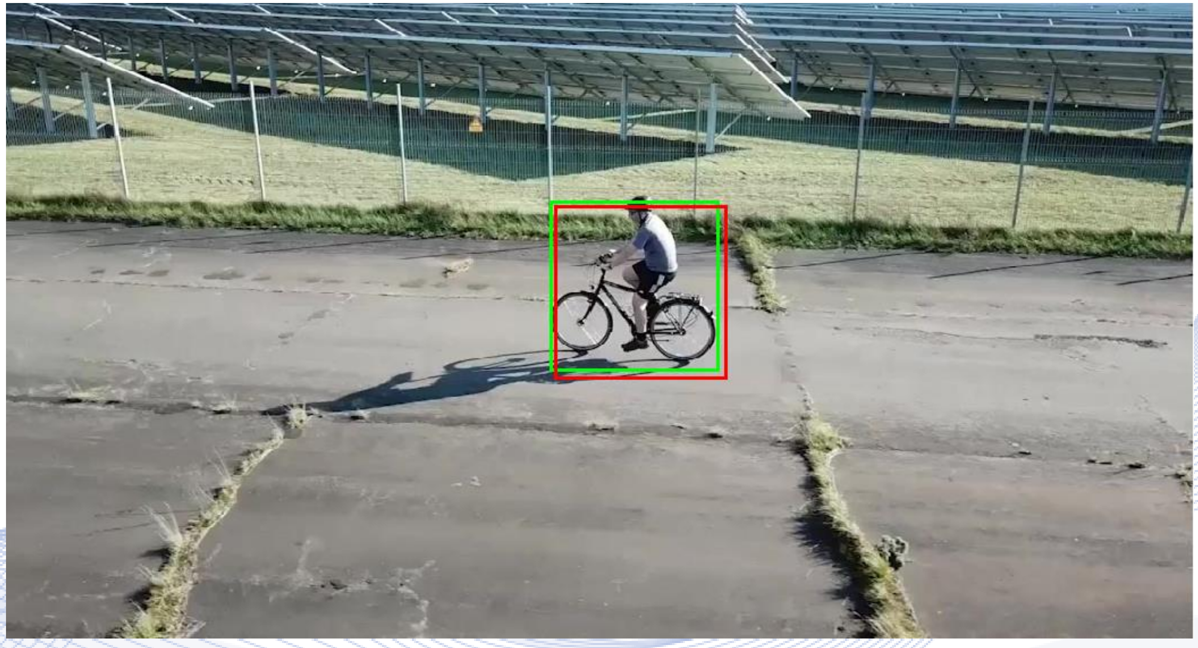
- $\mathcal{A},\mathcal{B}$: estimated, ground truth ROIs (sets, bounding boxes).
- $|\mathcal{A}|$: set cardinality (area counted in pixels)
- Also called ***Jaccard Similarity Coefficient*** or ***Overlap Score.***

Object Detection Performance Metrics

Object detection: a) $J(\mathcal{A},\mathcal{B}) = 0.67$; b) $J(\mathcal{A},\mathcal{B}) = 0.27$.

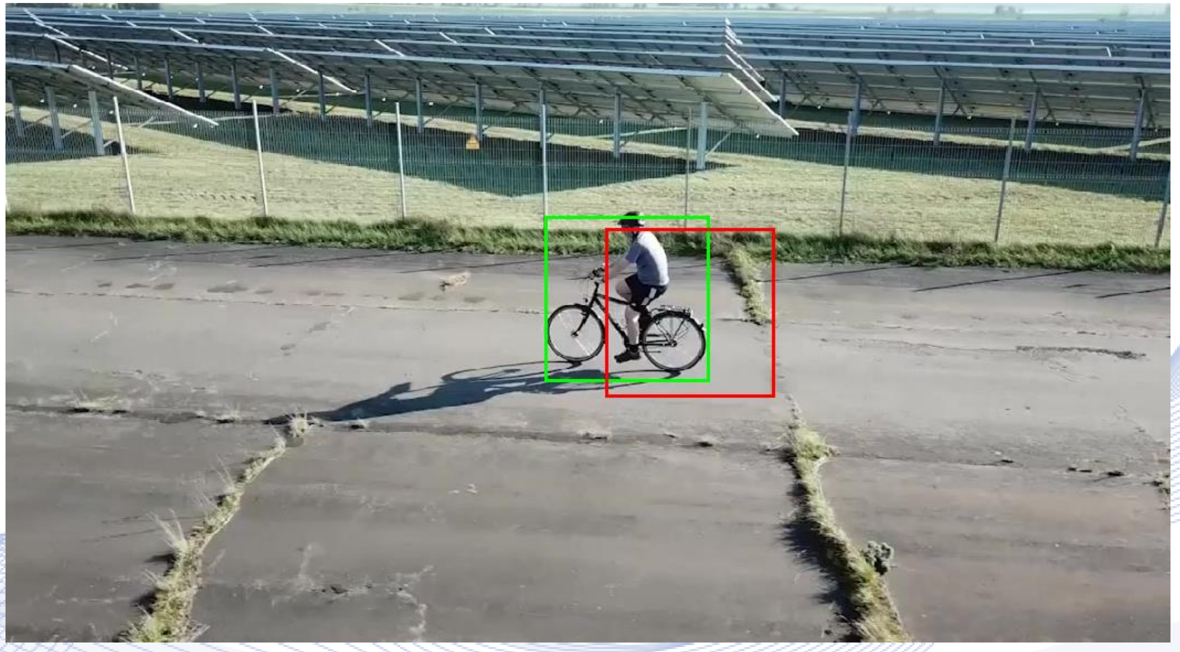
2D Tracking Performance metrics

- Red box: tracking results.
- Green box: ground-truth.
- Euclidean distance of central points: 6 pixels.
- $J(\mathcal{A}, \mathcal{B}) = 0.9$
- Tracking: ***SUCCESS***.



2D Tracking Performance metrics

- Red box: tracking results.
- Green box: ground-truth.
- Euclidean distance of central points: 61 pixels.
- $J(\mathcal{A},\mathcal{B}) = 0.4$
- Tracking: ***FAILURE***.



Bibliography

[PIT2017] I. Pitas, "Digital video processing and analysis" , China Machine Press, 2017 (in Chinese).

[PIT2013] I. Pitas, "Digital Video and Television" , Createspace/Amazon, 2013.

[PIT2021] I. Pitas, "Computer vision", Createspace/Amazon, in press.

[NIK2000] N. Nikolaidis and I. Pitas, "3D Image Processing Algorithms", J. Wiley, 2000.

[PIT2000] I. Pitas, "Digital Image Processing Algorithms and Applications", J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr