Human Action Recognition via Keypoint Tracking

Yunus Emre Kara

Department of Computer Engineering
Boğaziçi University

MS Thesis Presentation, 17.01.2011

Action Recognition vs. Action Detection

- The two concepts are usually confused.
- Detection answers the question:
  "Is there an action in this sequence?"
- Recognition answers the question:
  "What is the action, given that there exists an action in this sequence?"
- In this work, we deal with the recognition of actions.

Action variations

- Actions can have large variations in performance
  - An action can differ in speed
  - Stride length can differ
- Antropometric differences
- Avoiding obstacles
- Temporal variations
  - Segmenting the actions in time is difficult
  - The length of an action can differ for different people

Environment and recording settings

- Person localization: Harder in cluttered or dynamic environments
  - Dynamic backgrounds
  - Moving camera
- Parts of the person can be occluded
- Lighting conditions can affect appearance
- An action has different observations for different viewpoints

Global Representations

- Person is localized first
- The region of interest is encoded as a whole
- Bobick and Davis[1] use MEI and MHI for recognizing actions
- Efros et al.[2] calculate optical flow measurements
- Gorelick et al.[3] stack silhouettes of the consecutive frames to form spacetime volumes.

[1] Bobick, A. and J. Davis, "The recognition of human movement using temporal templates", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 3, pp. 257-267, Mar. 2001.

[2] Efros, A. A., A. C. Berg, G. Mori and J. Malik, "Recognizing action at a distance", Proceedings Ninth IEEE International Conference on Computer Vision, Vol. 2, No. October, pp. 726-733, 2003.

[3] Gorelick, L., M. Blank, E. Shechtman, M. Irani and R. Basri, "Actions as spacetime shapes.", IEEE transactions on pattern analysis and machine intelligence, Vol. 29, No. 12, pp. 2247-2253, Dec. 2007.

Local Representations

- Some patches in the images are encoded separately
- Patches are extracted from the neighborhoods of keypoints
- The representation is found by combining the information of the patches
- Laptev [4] proposed space-time interest points by extending the Harris interest point detector to 3D
- Willems et al.[5] use integral videos to find salient points.
- Sun et al.[6] find SIFT descriptors and track interest points using these descriptors.

[4] Laptev, I., "On Space-Time Interest Points", International Journal of Computer Vision, Vol. 64, No. 2-3, pp. 107-123, Sep. 2005.

[5] Willems, G., T. Tuytelaars and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector", Computer Vision ECCV 2008 , pp. 650-663, 2008.

[6] Sun J, Wu X, Yan S, et al. "Hierarchical spatio-temporal context modeling for action recognition", Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. 2009:2004-2011.

KTH

- 2391 sequences, 25 people, 4 scenarios, 3-4 repeats, low resolution
- 6 actions: walking, running, jogging, boxing, hand waving, hand clapping

URADL

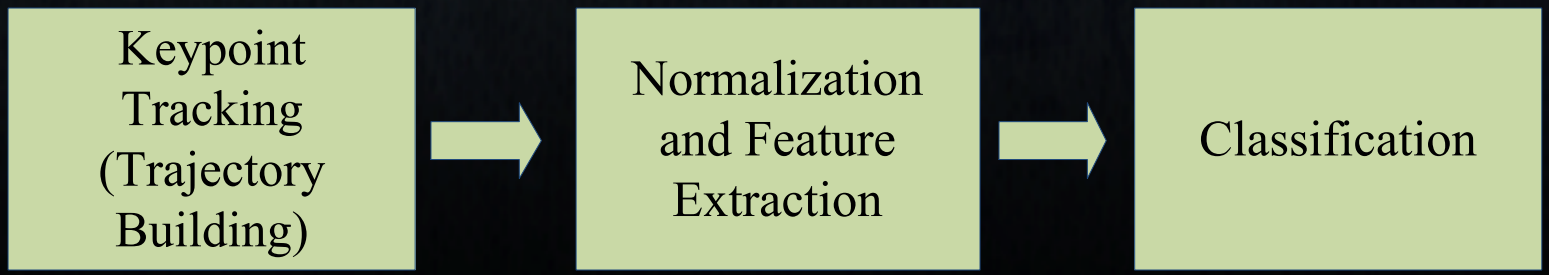
- 150 sequences, 5 people, 3 repeats, high resolution
- 10 actions: Answering a phone, Chopping a banana, Dialing a phone, Drinking water, Eating banana, Eating snack chips, Looking up a phone number in a phone book, Peeling a banana, Using silverware, Write a phone number on a white board

WeCare

- 720 sequences, 6 people, 10 repeats, moderate resolution
- 8 actions: Walking, Jumping, Sitting on the armchair, Standing up from the armchair, Lying on the gym mat, Standing up from the gym mat, Falling onto the armchair, Falling onto the gym mat

Outline

- A local approach
- Tracks keypoints
- Uses their trajectories



What is a keypoint?

- Keypoint: Important points; such as corners, blobs, t-junctions
- Usage: object detection, image registration, camera calibration, image mosaicing, ...
- Methods: Harris, FAST, SIFT, SURF, ...
- Keypoint descriptor: Gives information about the neighborhood of a keypoint

Why do we use trajectories?

- Keypoints are robust for tracking objects

Why do we use trajectories?

- Keypoints are robust for tracking objects
- Localization and background subtraction are not required

Why do we use trajectories?

- Keypoints are robust for tracking objects
- Localization and background subtraction are not required
- Trajectories of tracked keypoints give information about the motion

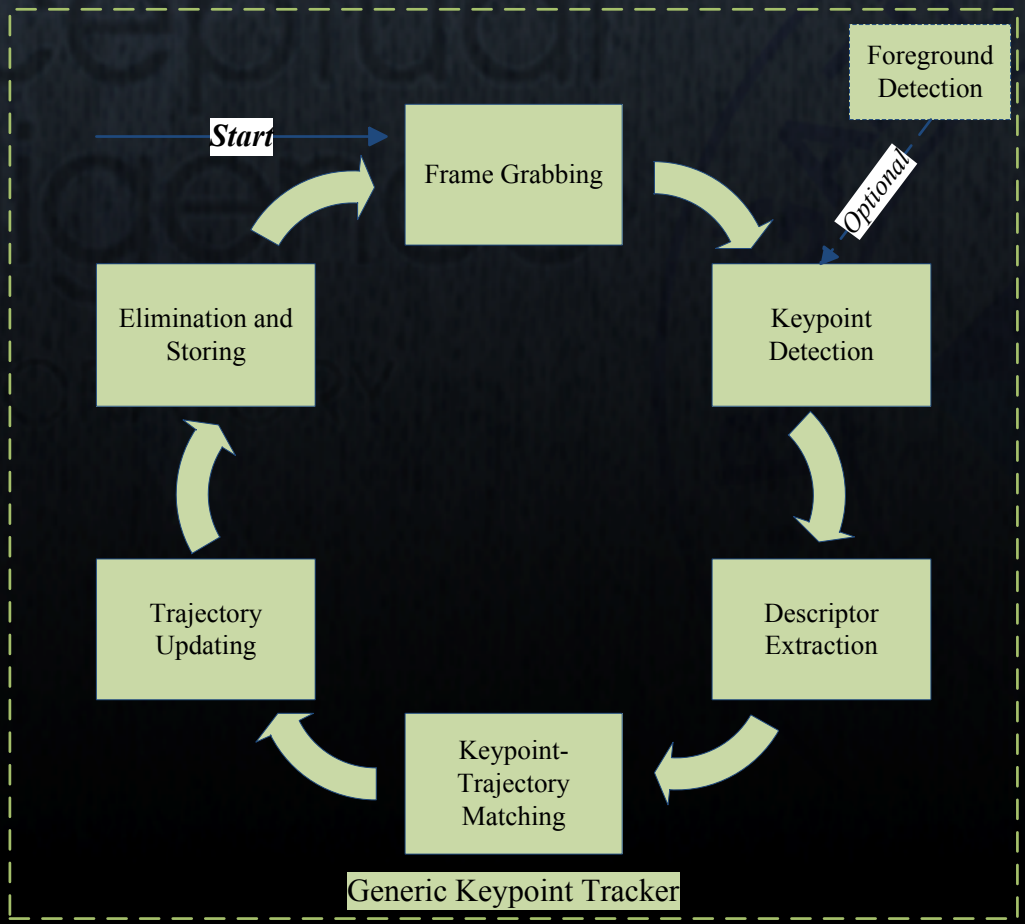
Why do we use trajectories?

- Keypoints are robust for tracking objects
- Localization and background subtraction are not required
- Trajectories of tracked keypoints give information about the motion
- The motion information is invariant to the appearance

Some advantages of our approach

- Invariant to some viewpoint changes
- Invariant to partial occlusions
- Robust to the temporal length variations.
- Invariant to the location of the action.

Outline of the Generic Keypoint Tracker



Keypoint detection

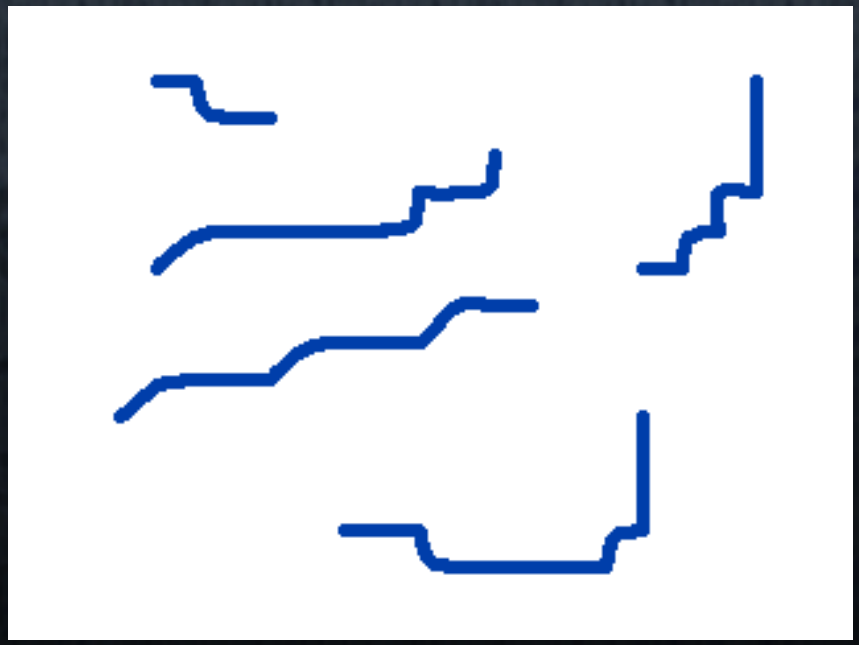
- Finding point correspondences between two images:
  - Keypoint detection
  - Descriptor extraction
  - Keypoint matching
- Our algorithm allows the use of external keypoint detector and descriptor extraction modules
- Built-in "Keypoint Matching" module
- We use SURF method[7] both for keypoint detection and descriptor extraction steps.
- SURF keypoint descriptor: 64 dimensional vector extracted from the keypoint's local patch

[7] Bay, H., A. Ess, T. Tuytelaars and L. V. Gool, "Speeded-Up Robust Features (SURF)", Computer Vision and Image Understanding, Vol. 110, No. 3, pp. 346-359, 2008.

Keypoint-Trajectory Matching

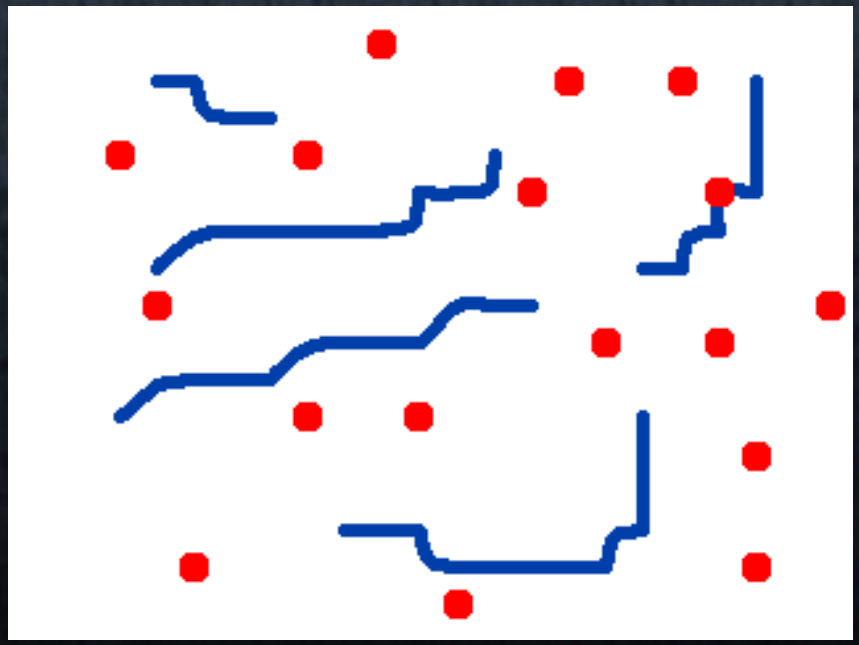
- Gaps in time are allowed ⇒ robust to occlusions
- Missing keypoints are interpolated linearly
- Trajectory descriptor: Calculated using the descriptors of member keypoints
- We compare the descriptors of keypoints with the descriptors of trajectories
- The descriptor distance of the best match is divided by the descriptor distance of the second best match.
  Small ratio ⇒ Mark for matching
- For matching, we must mark in both ways. The matched trajectory of a keypoint must also be matched with that keypoint.

Keypoint-Trajectory Matching



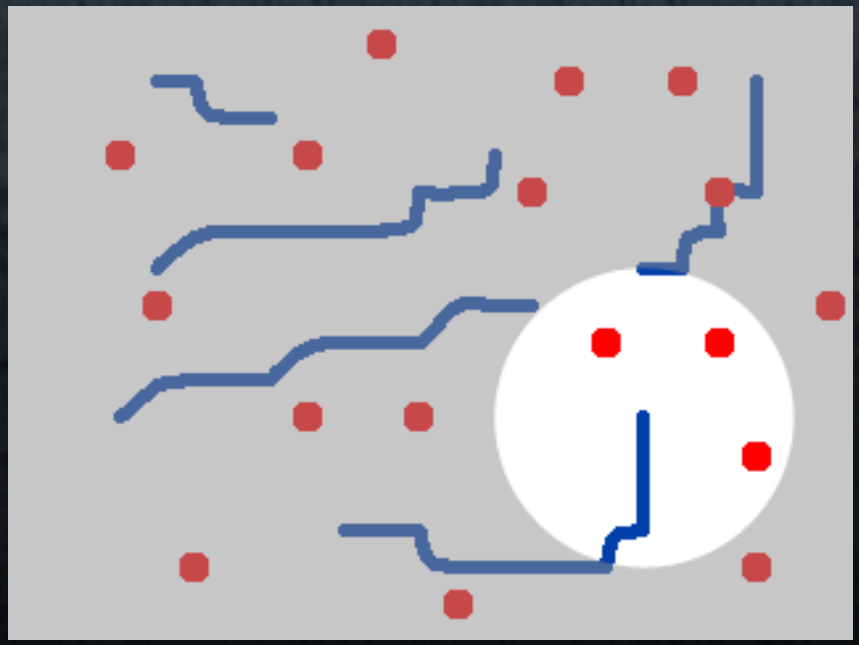
- Current trajectories (blue curves)

Keypoint-Trajectory Matching



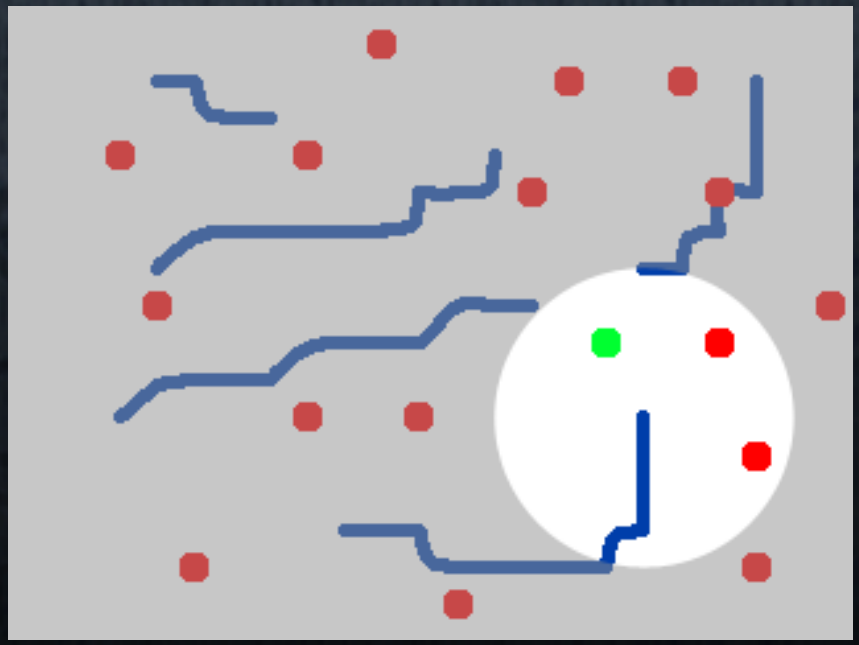
- Current trajectories (blue curves)
- Newly detected keypoints (red points)

Keypoint-Trajectory Matching



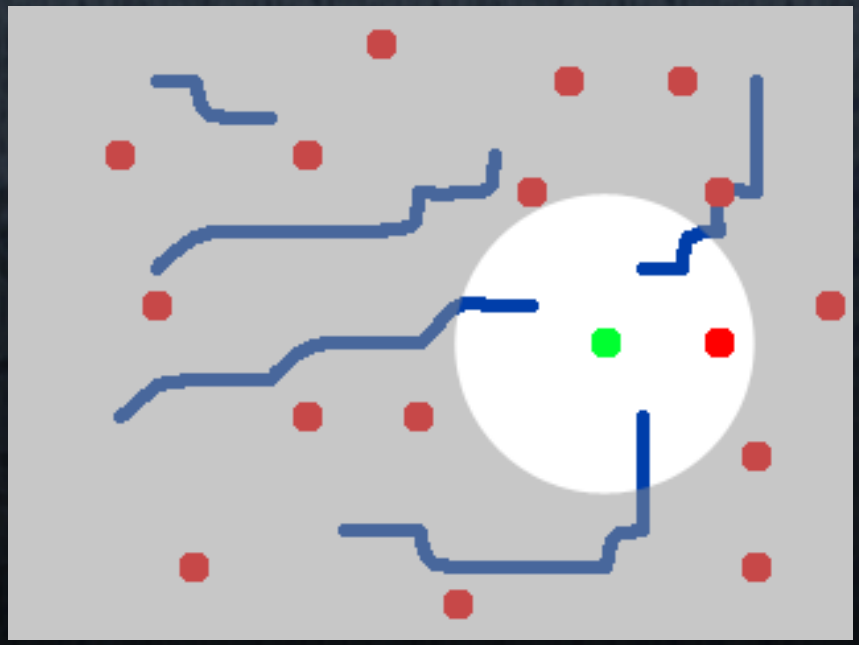
- Current trajectories (blue curves)
- Newly detected keypoints (red points)
- Spatial neighborhood of a trajectory

Keypoint-Trajectory Matching



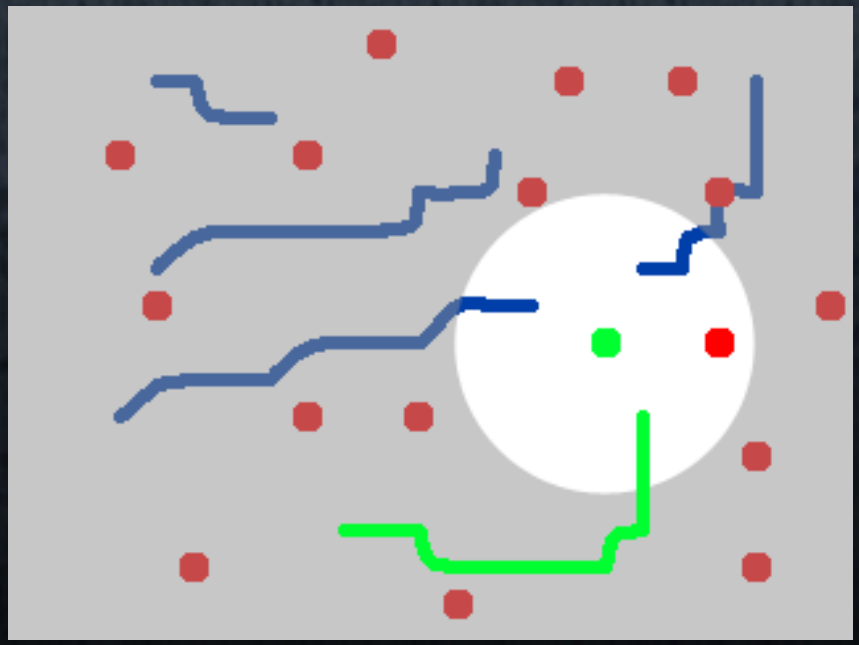
- Current trajectories (blue curves)
- Newly detected keypoints (red points)
- Spatial neighborhood of a trajectory
- Best matching keypoint (green point)

Keypoint-Trajectory Matching



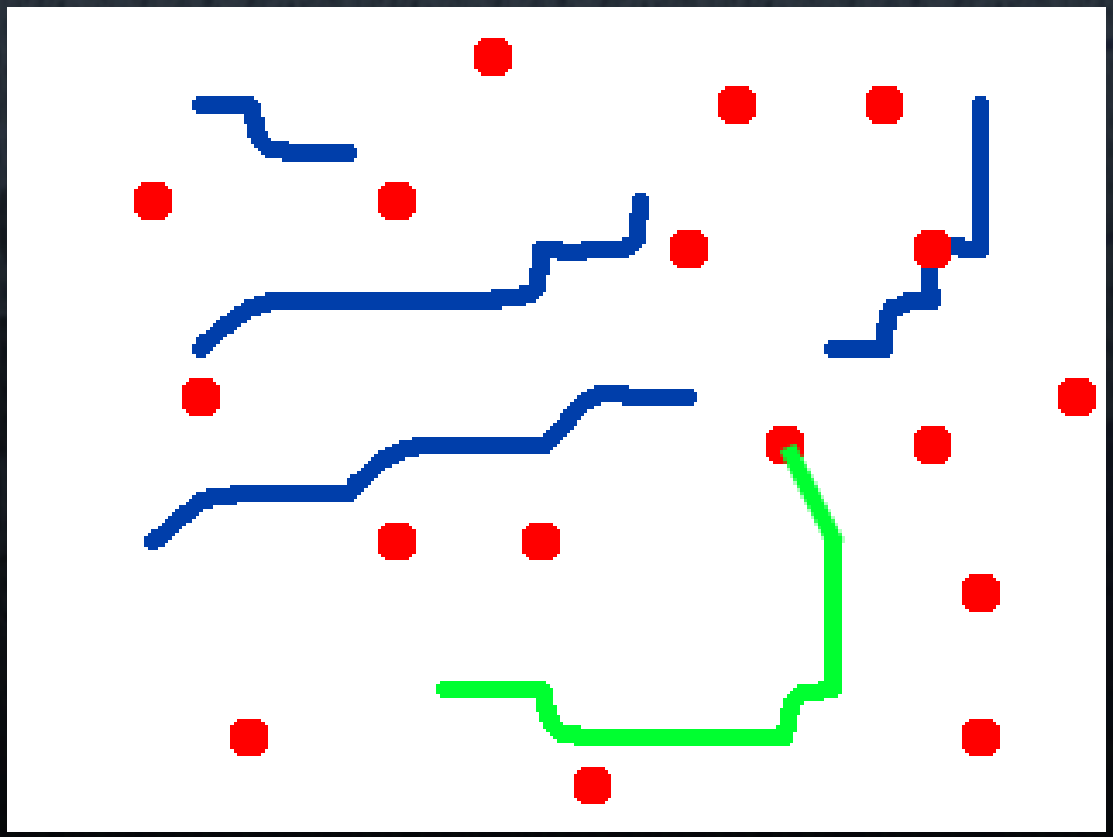
- Current trajectories (blue curves)
- Newly detected keypoints (red points)
- Spatial neighborhood of a trajectory
- Best matching keypoint (green point)
- Spatial neighborhood of the best matching keypoint

Keypoint-Trajectory Matching



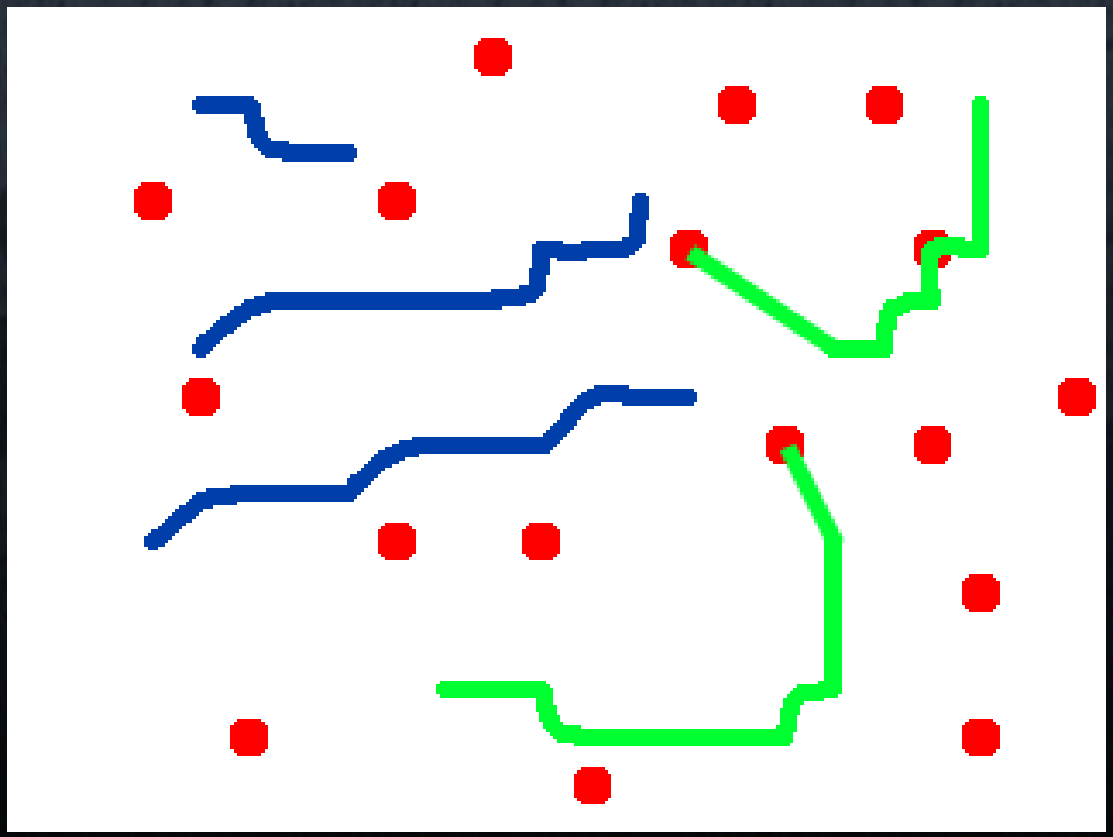
- Current trajectories (blue curves)
- Newly detected keypoints (red points)
- Spatial neighborhood of a trajectory
- Best matching keypoint (green point)
- Spatial neighborhood of the best matching keypoint
- Best matching trajectory (green curve)

Trajectory Updating



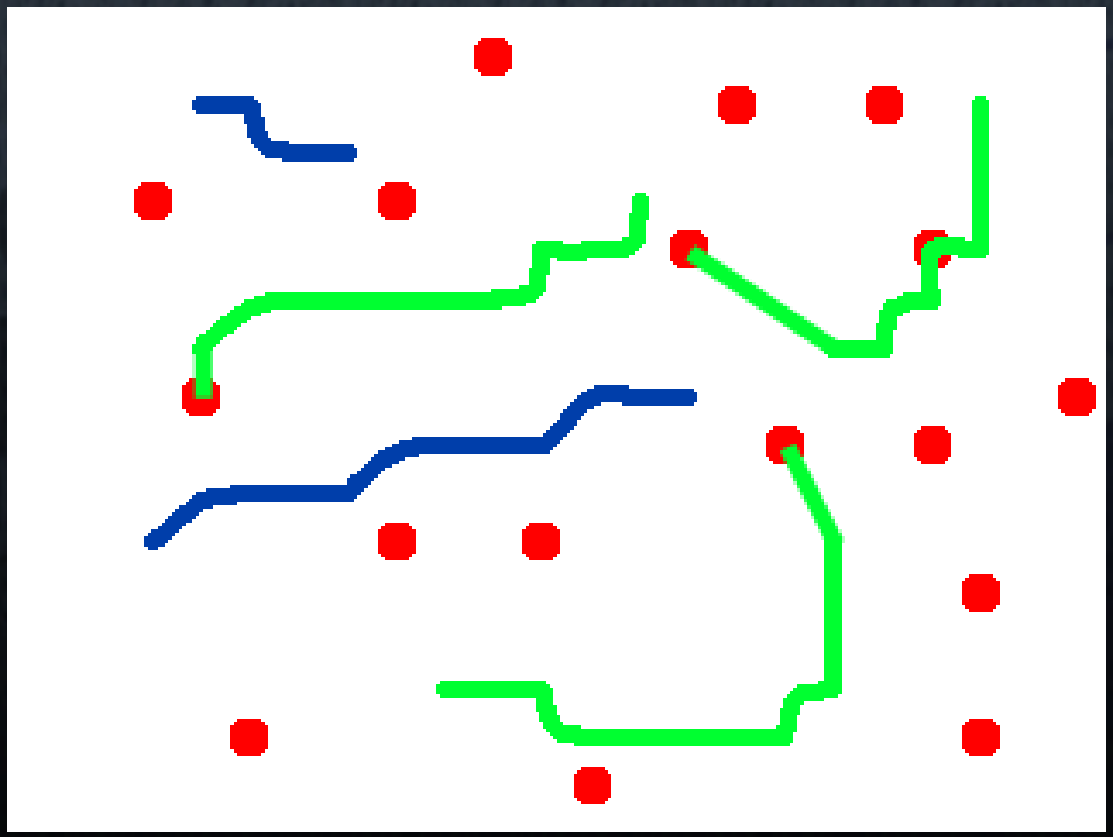
- Matching is done for each trajectory (green curves)

Trajectory Updating



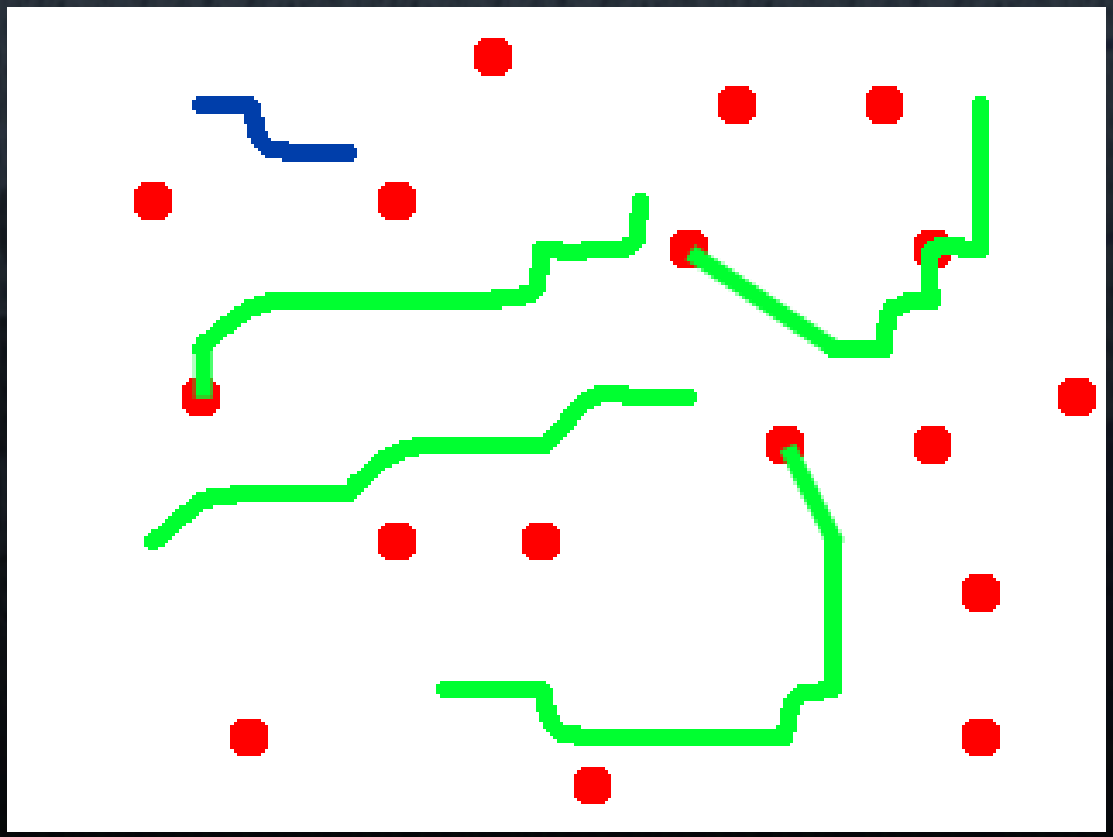
- Matching is done for each trajectory (green curves)

Trajectory Updating



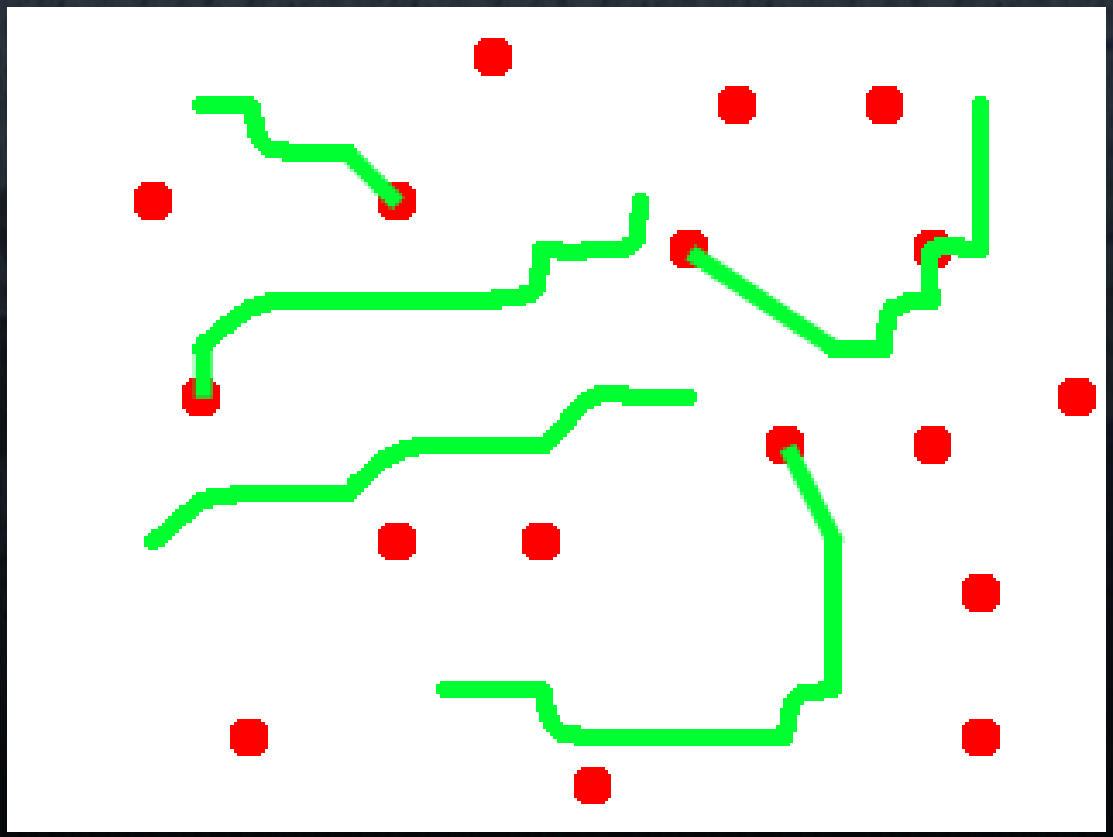
- Matching is done for each trajectory (green curves)

Trajectory Updating



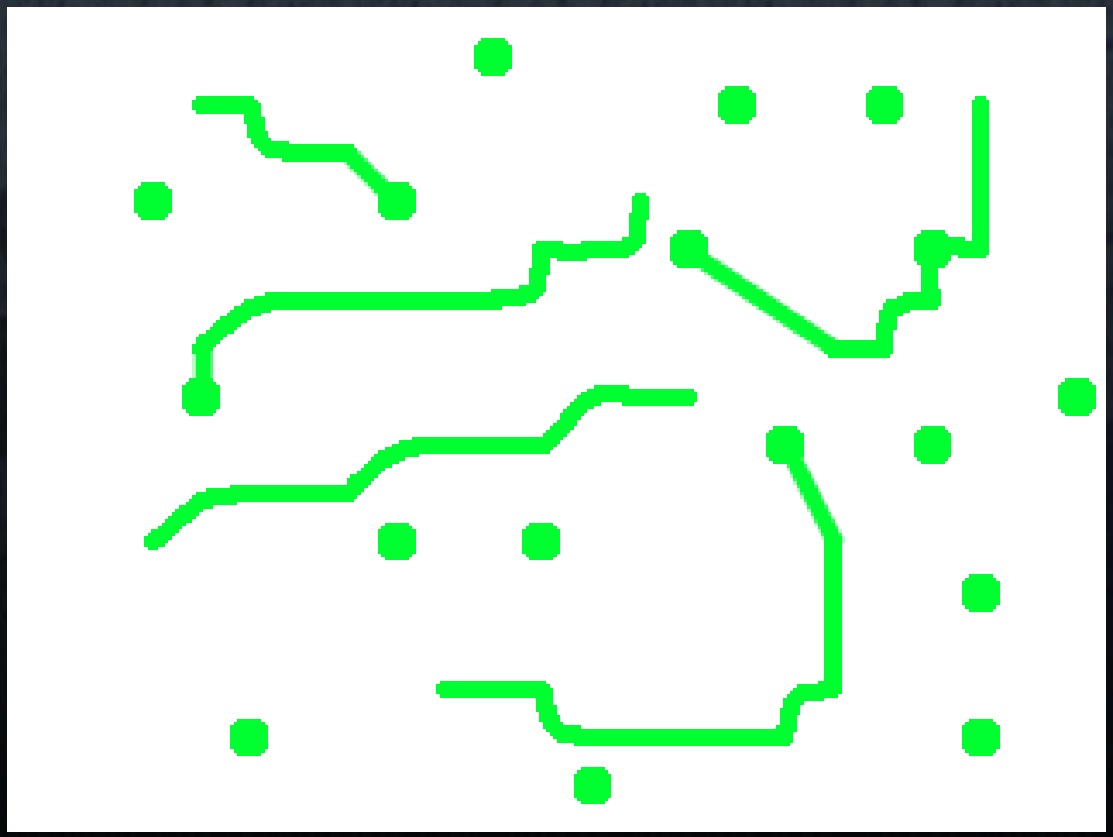
- Matching is done for each trajectory (green curves)
- Some trajectories may not be matched with a keypoint

Trajectory Updating



- Matching is done for each trajectory (green curves)
- Some trajectories may not be matched with a keypoint

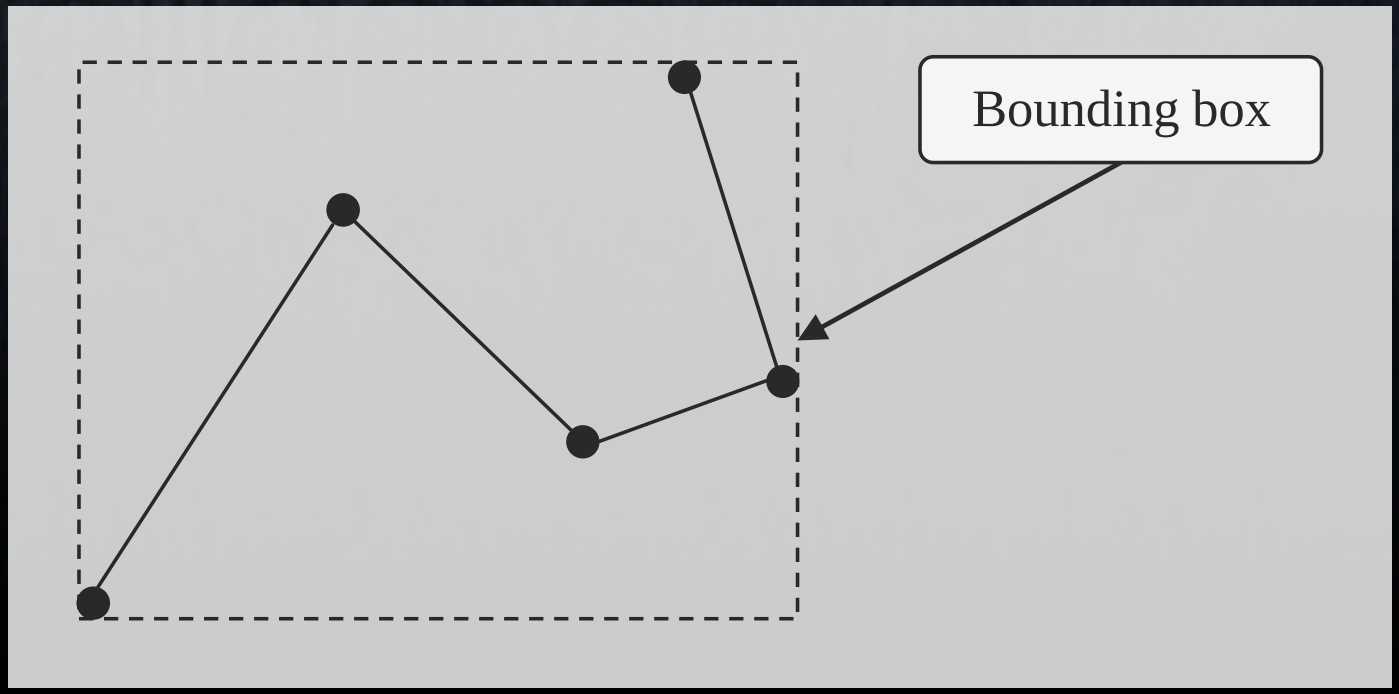
Trajectory Updating



- Matching is done for each trajectory (green curves)
- Some trajectories may not be matched with a keypoint
- Unmatched keypoints initiate new trajectories (green points)

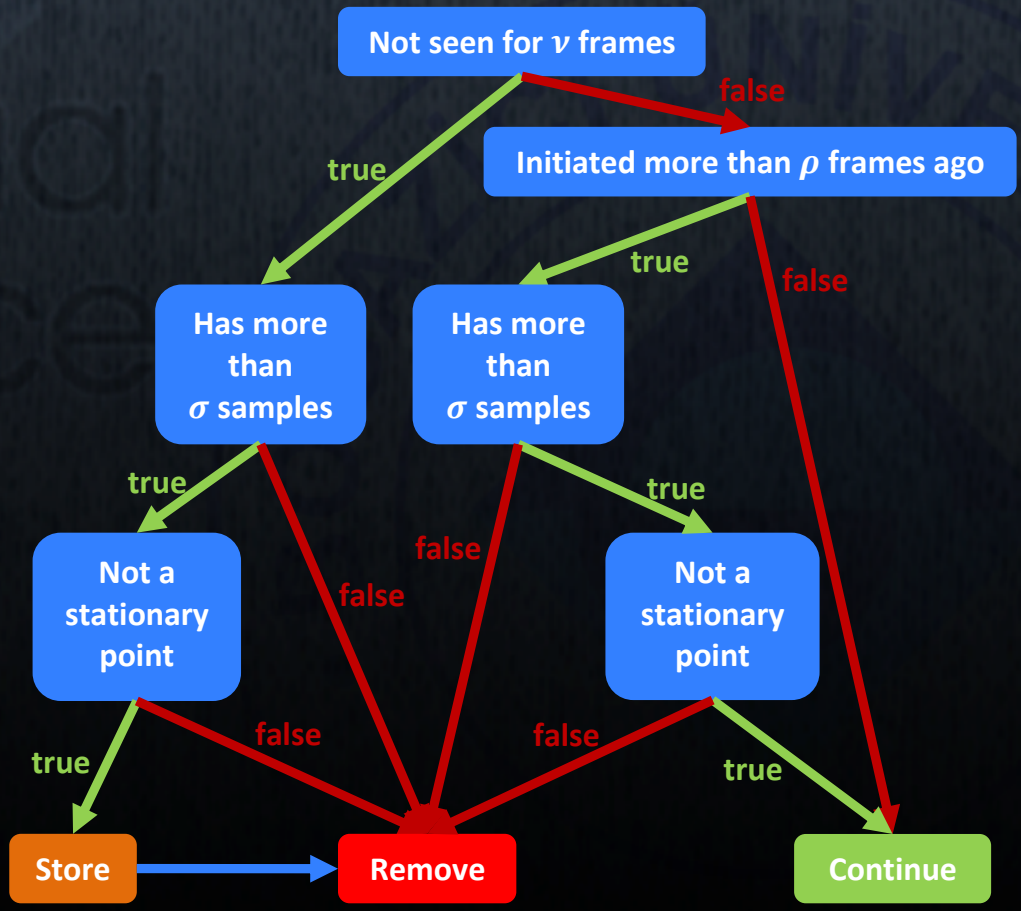
Stationary points

The stationary points are found by looking at the diagonal length of the trajectory's bound box.

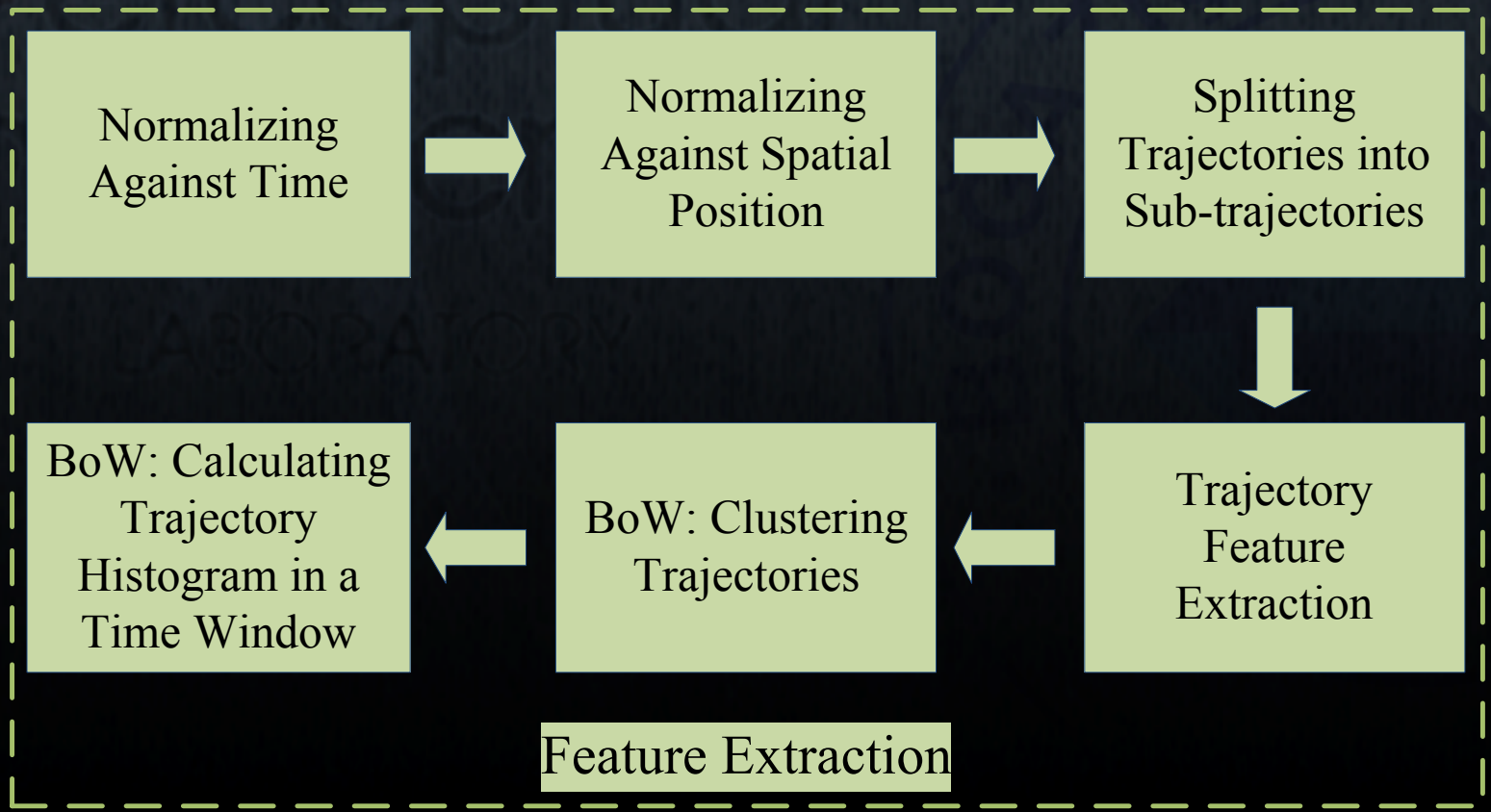


Elimination and Storing

- We eliminate trajectories
  - for narrowing down the search space
  - and removing noise
- We store trajectories if
  - they are ready for feature extraction
  - and not modified for some time
- We do these operations after each frame



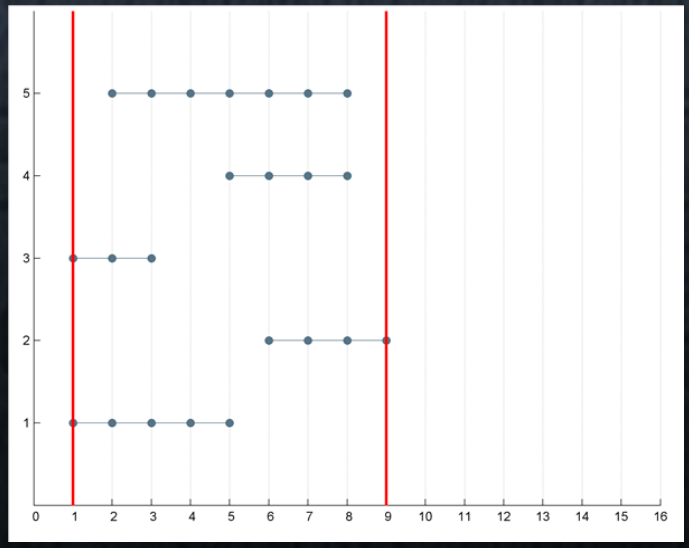
Feature extraction outline



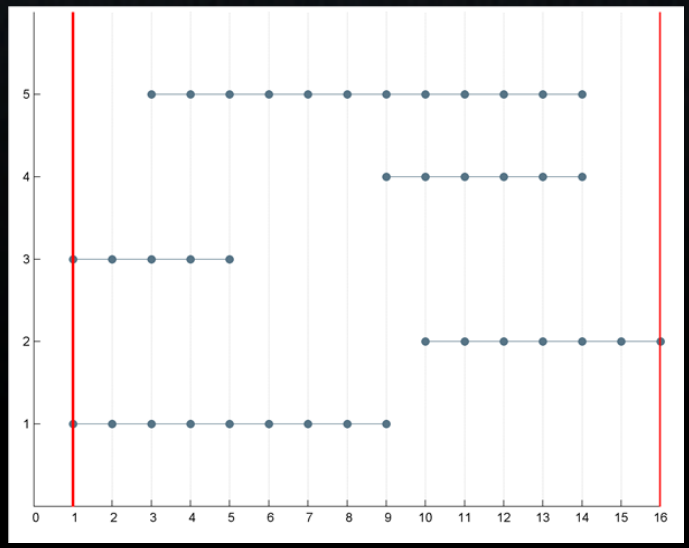
Normalizing Against Time

Original

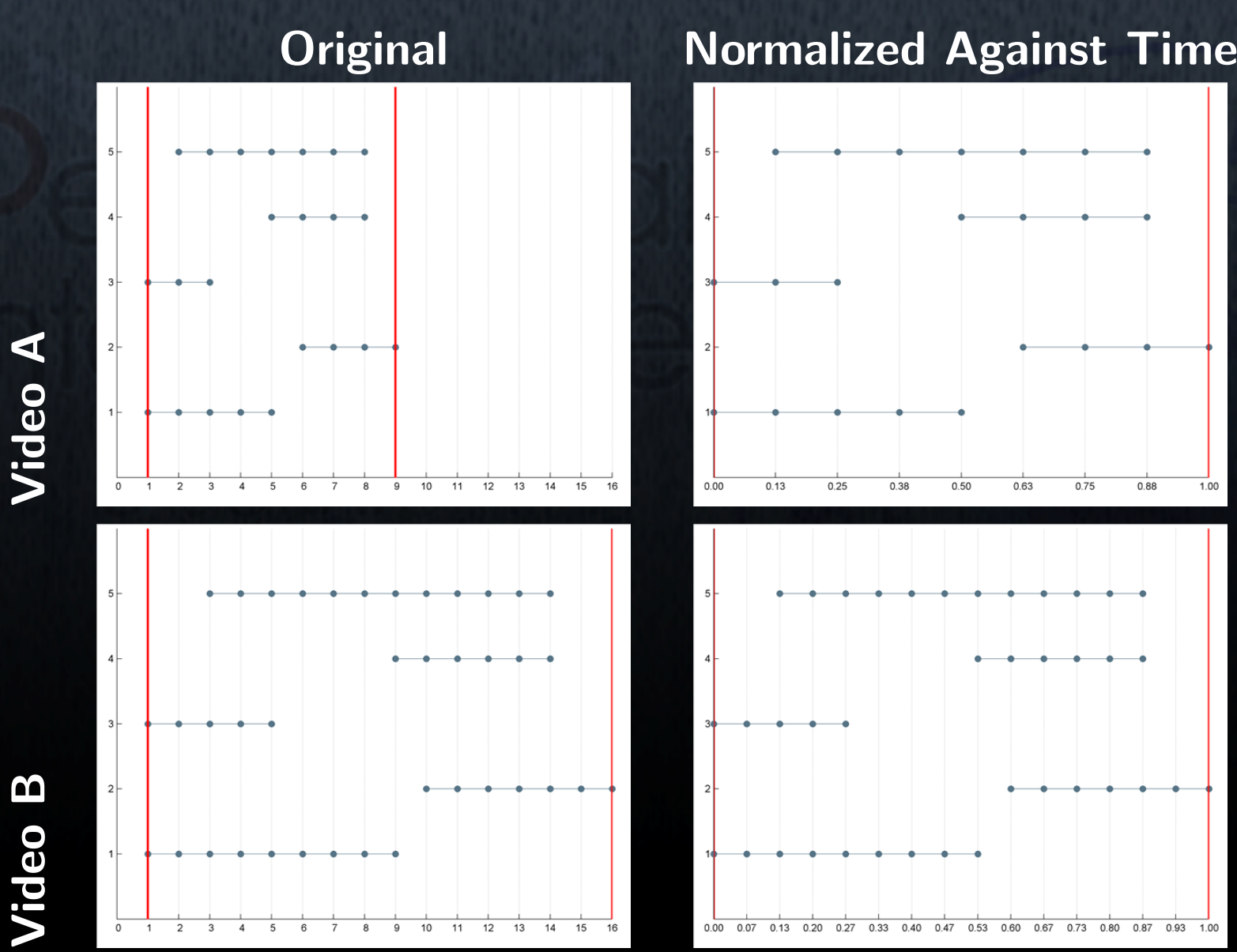
Video A



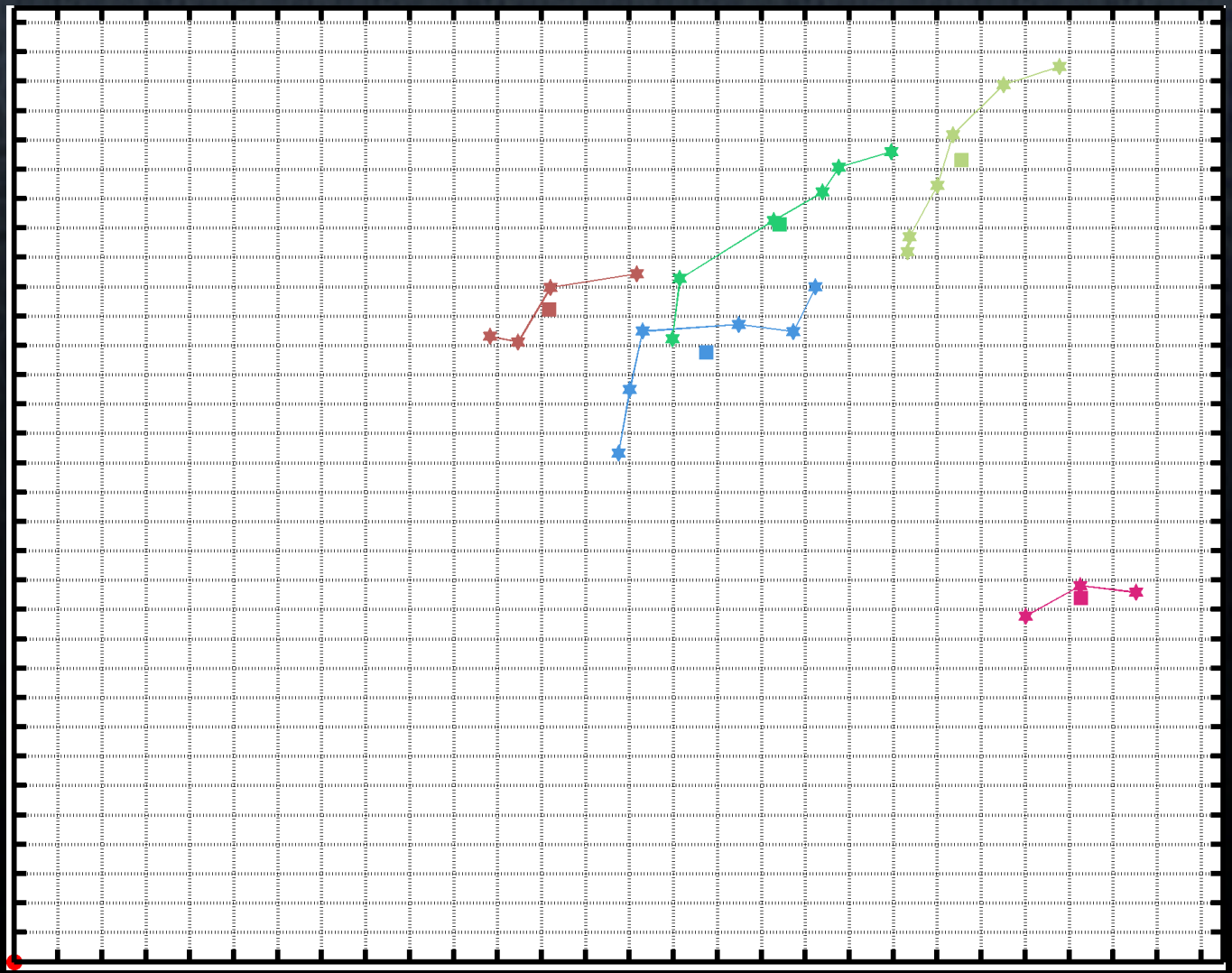
Video B



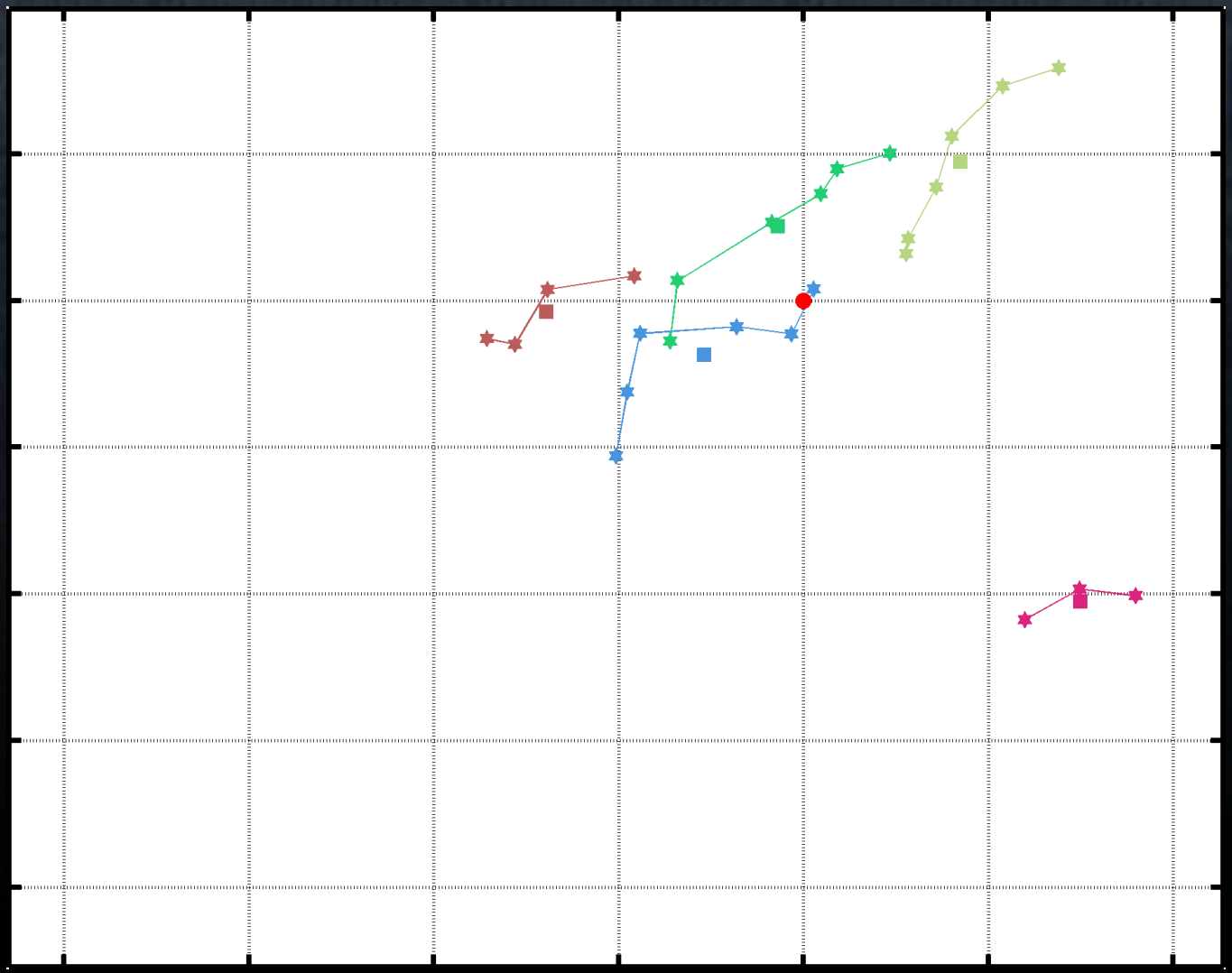
Normalizing Against Time



Normalizing Against Spatial Position

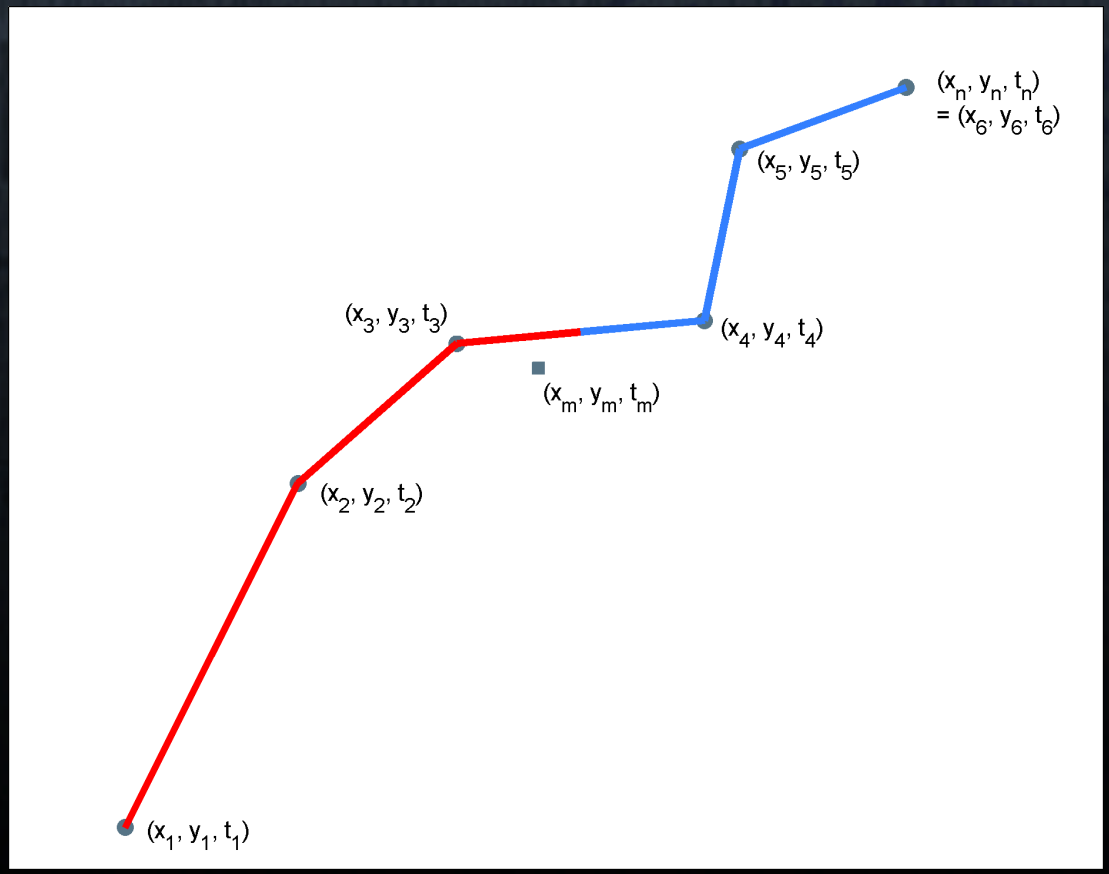


Normalizing Against Spatial Position



Trajectory Feature Extraction

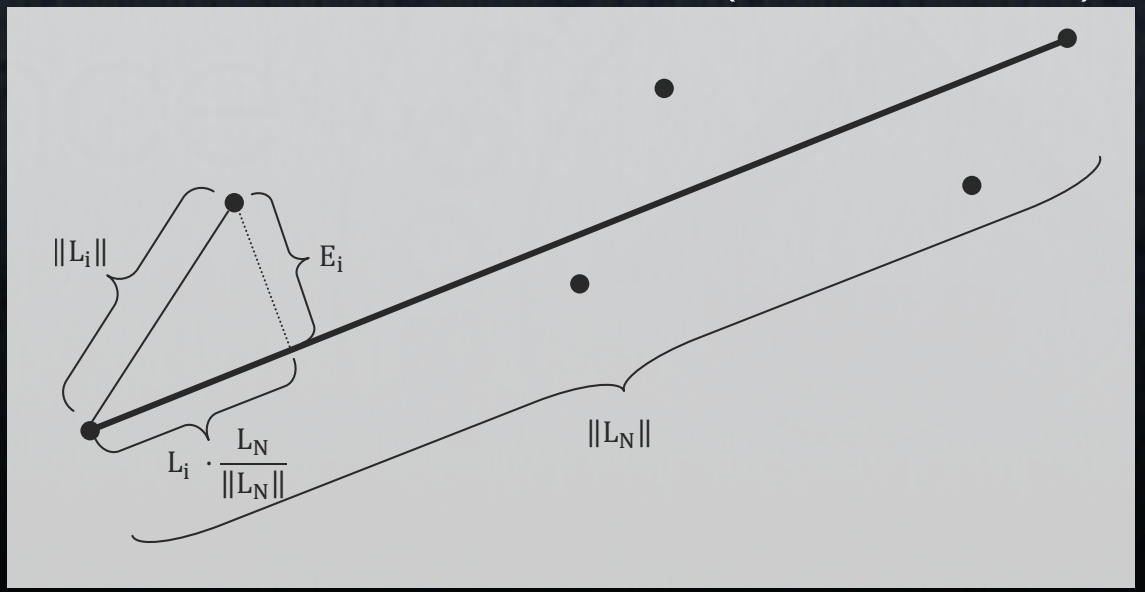
- Start time: t_1
- End time: t_n
- Mean x coordinate of keypoints (x_m)
- Mean y coordinate of keypoints (y_m)
- Length of the path until halftime (red path)
- Length of the path since halftime (blue path)



Trajectory Feature Extraction

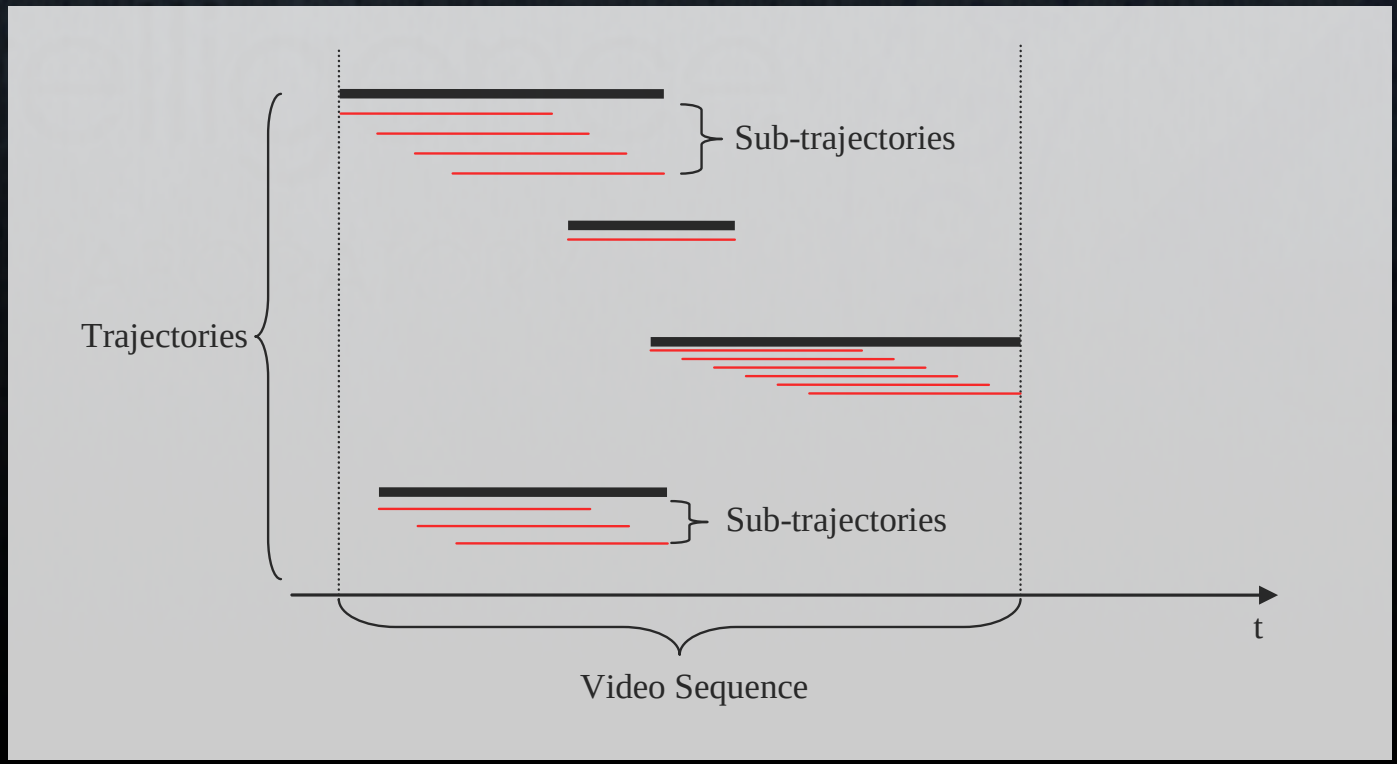
- x and y components of the vector passing through the first and the last keypoints ($L_{N,x}$ and $L_{N,y}$)
- Maximum $|E_i|$, with sign
- $$\frac{L_i \cdot L_N}{||L_N||^2}$$

The vector passing through the first and i^{th} keypoints: $L_i = (x_i - x_1, y_i - y_1)$



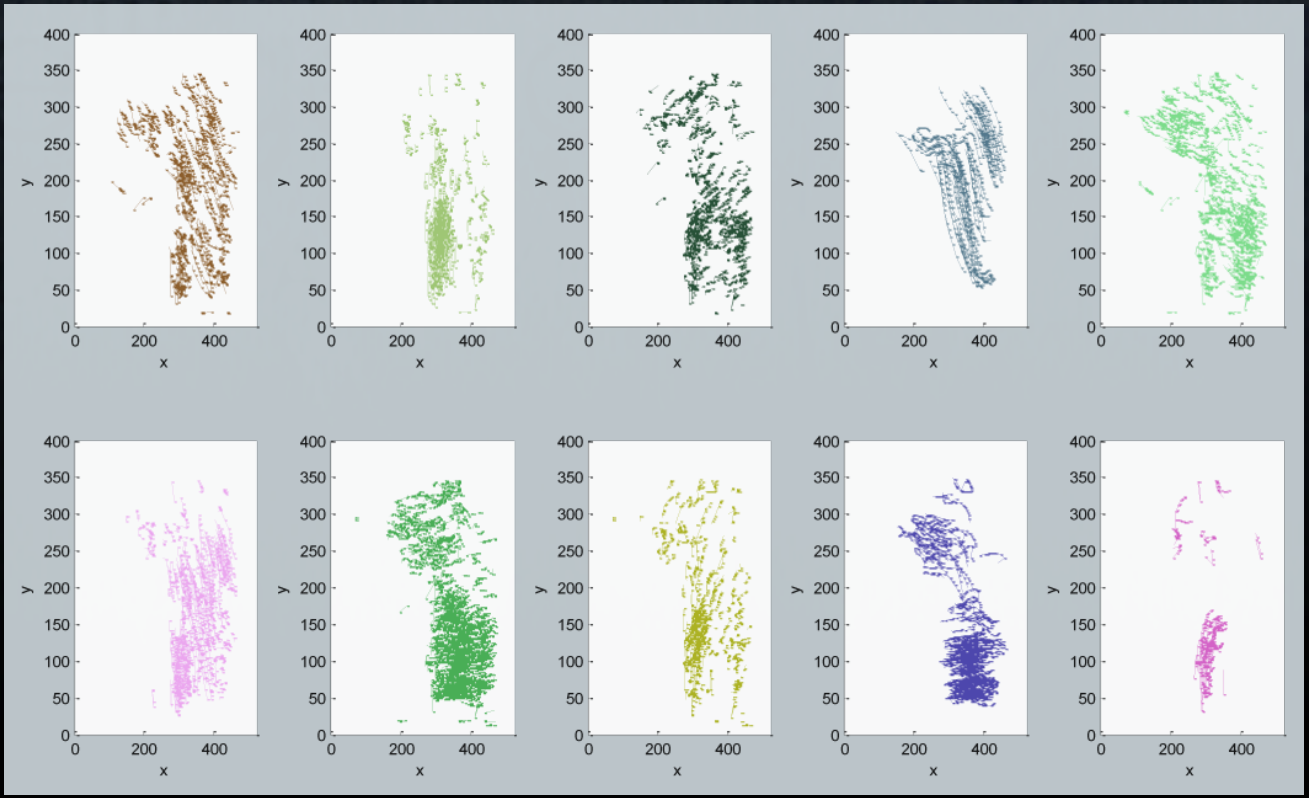
Extracting Sub-trajectories

- Trajectories differ in length
- We divide them into smaller subparts
- These subparts are used in describing videos



Bag-of-trajectories

- Trajectory features are normalized (t-statistic or min-max)
- Normalized features are clustered using k-means
- Image sequence descriptor: Normalized histogram of the clusters of trajectories



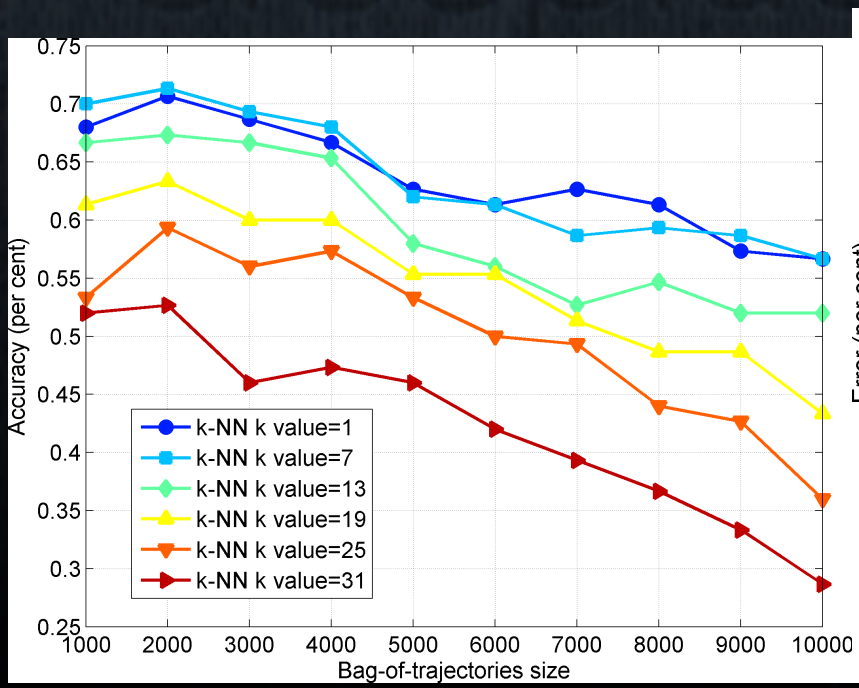
Parameters

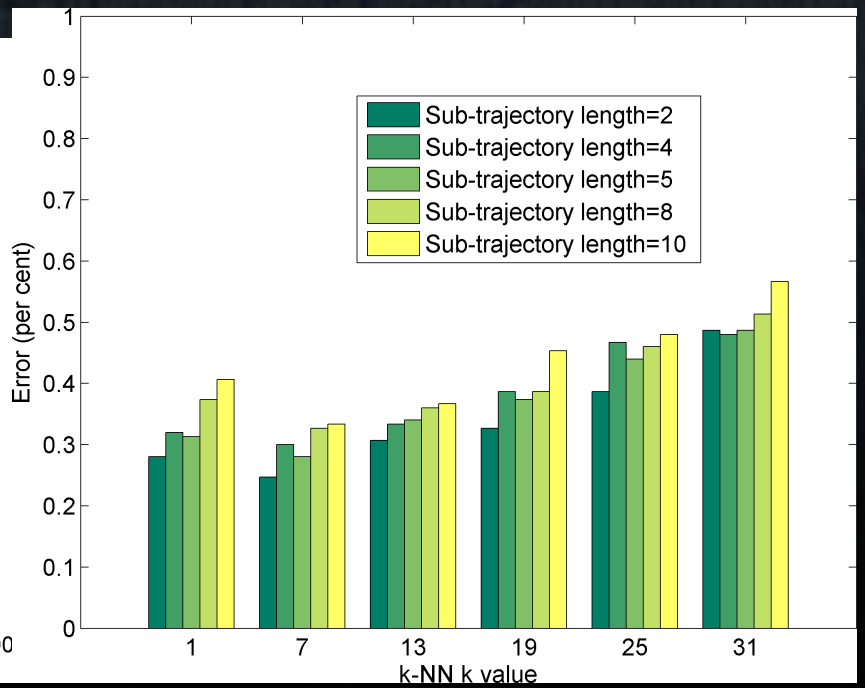
σ	the minimum required sample count
ν	the maximum allowed time since last observation
ρ	the minimum age to be eliminated
ω	the maximum number of keypoints for a sub-trajectory
n	normalization method before bag-of-trajectories
K	cluster count for K-means of bag-of-trajectories
C	SVM cost parameter
k	k-NN k value depending on the method

Experiment Setup

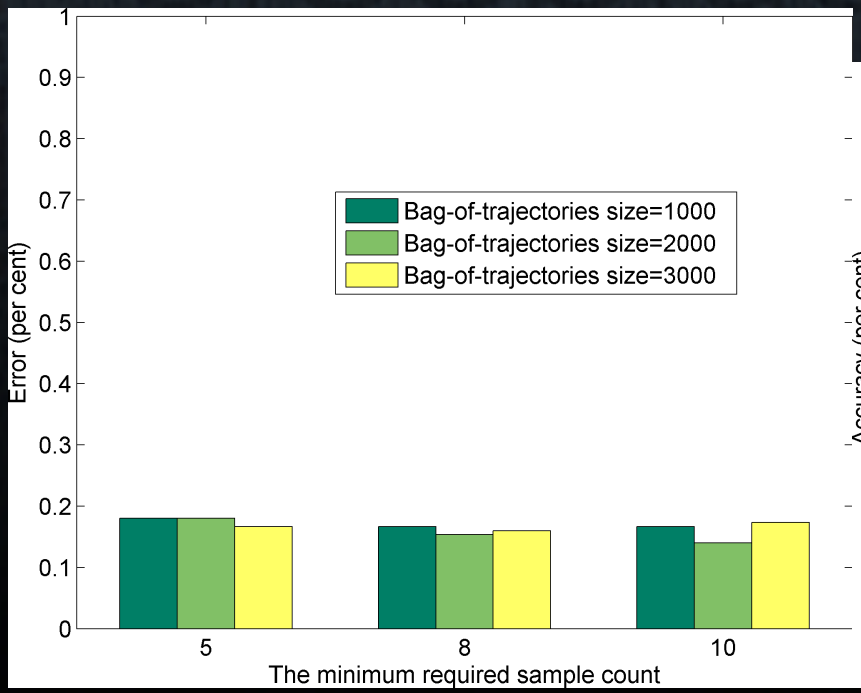
- A total of 23994 experiments
  - 4300 experiments on the KTH dataset
  - 15394 experiments on the URADL dataset
  - 4300 experiments on the WeCare dataset
- 10-fold cross validation on KTH and WeCare datasets
- 5-fold cross validation on the URADL datasets
- SVM with χ^2 kernel is tried with 11 different cost parameter values
- k-NN is tried with all k's in the range 1-32

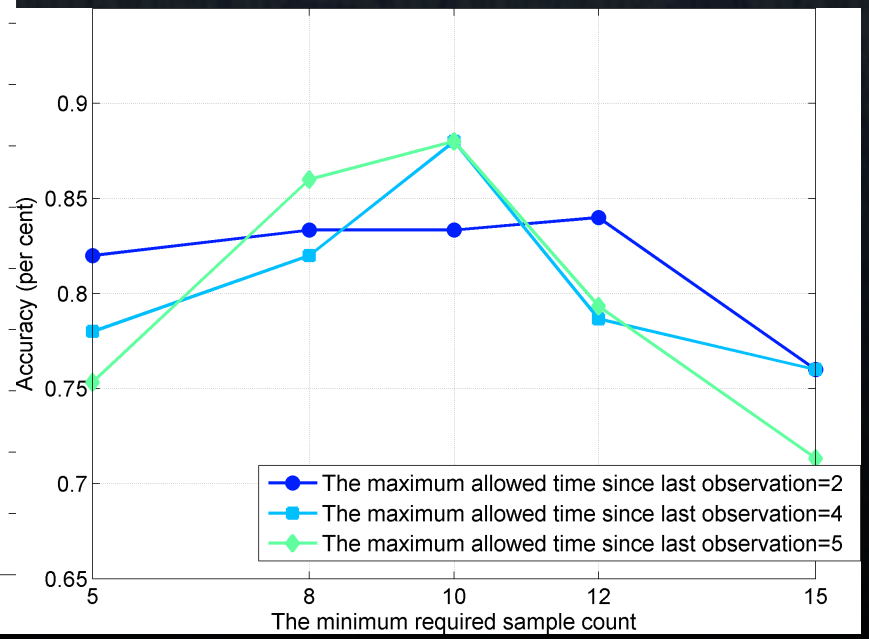
Validation Results (1)



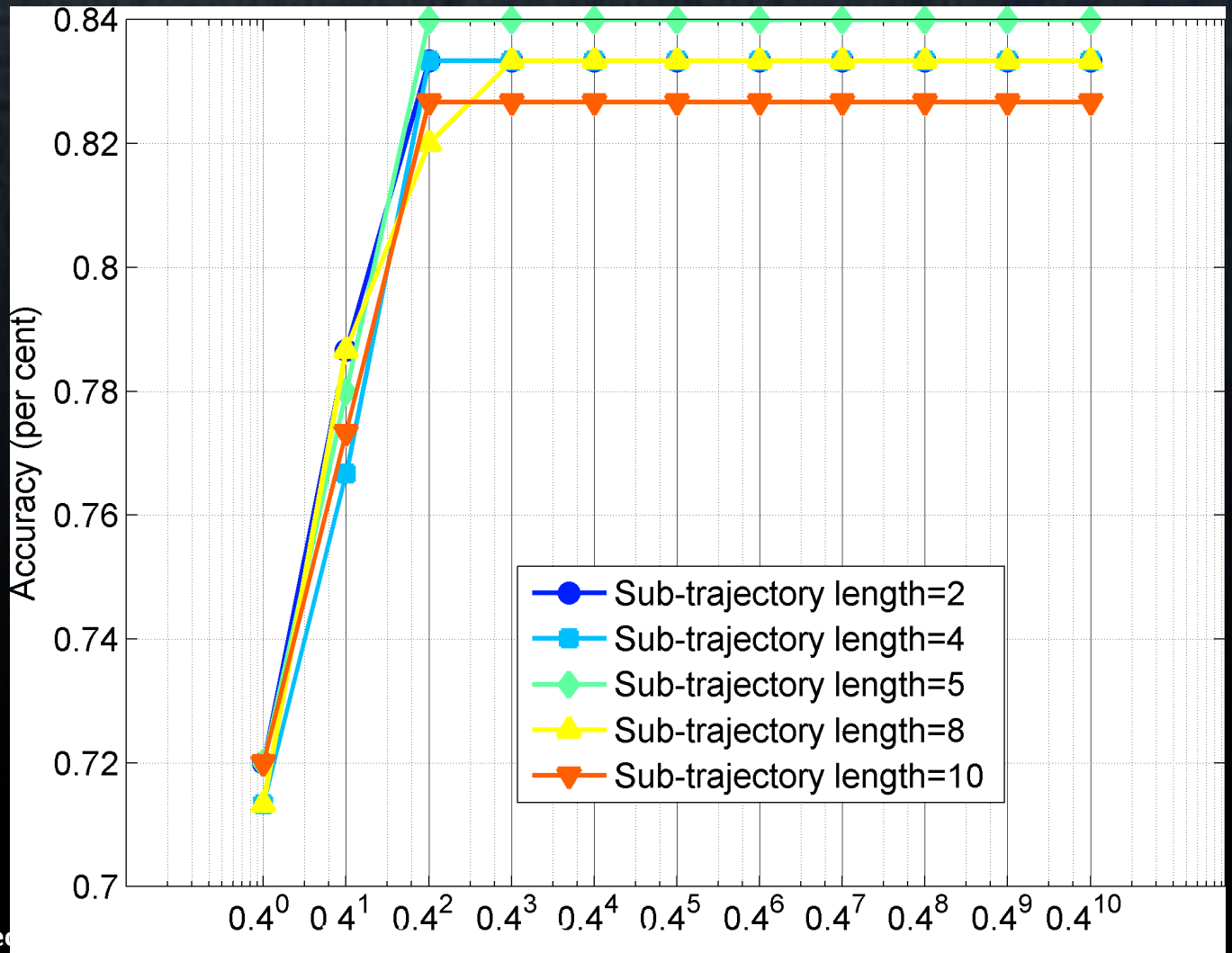


Validation Results (2)





Validation Results (3)



KTH dataset

- Accuracy: 87.25%
- Classes: C_1: walking, C_2: jogging, C_3: running, C_4: boxing, C_5: hand clapping, C_6: hand waving

		Actual Class					
		C_1	C_2	C_3	C_4	C_5	C_6
Prediction	C_1	130	8	0	0	0	0
	C_2	14	126	26	0	0	0
	C_3	0	10	118	0	0	0
	C_4	0	0	0	139	15	5
	C_5	0	0	0	4	129	28
	C_6	0	0	0	0	0	111

URADL dataset

- Accuracy: 88%
- Classes: C_1: answer phone, C_2: chop banana, C_3: dial phone, C_4: drink water, C_5: eat banana, C_6: eat snack, C_7: lookup in phonebook, C_8: peel banana, C_9: use silverware, C_{10}: write on whiteboard

		Actual Class									
		C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Prediction	C_1	15	0	1	0	0	0	0	0	0	0
	C_2	0	14	0	0	0	0	0	0	0	0
	C_3	0	1	14	0	2	0	0	1	0	0
	C_4	0	0	0	14	0	0	0	1	0	0
	C_5	0	0	0	0	12	0	0	2	0	0
	C_6	0	0	0	0	0	12	0	2	0	0
	C_7	0	0	0	0	0	2	14	0	1	0
	C_8	0	0	0	1	1	1	0	8	0	0
	C_9	0	0	0	0	0	0	1	1	14	0
	C_{10}	0	0	0	0	0	0	0	0	0	15

WeCare dataset

- Accuracy: 98.75%
- Classes: C_1: walking, C_2: jumping, C_3: sitting on the armchair, C_4: standing up from the armchair, C_5: lying on the gym mat, C_6: standing up from the gym mat, C_7: falling onto the armchair, C_8: falling onto the gym mat

		Actual Class							
		C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
Prediction	C_1	80	0	0	0	1	0	0	0
	C_2	0	20	0	0	0	0	0	0
	C_3	0	0	20	0	0	0	1	0
	C_4	0	0	0	40	0	0	0	0
	C_5	0	0	0	0	18	0	0	0
	C_6	0	0	0	0	0	20	0	0
	C_7	0	0	0	0	0	0	19	0
	C_8	0	0	0	0	1	0	0	20

Conclusions (1)

- A novel method for tracking keypoints is proposed.
- A feature set for describing trajectories is proposed.
- Performance is comparable to the methods in the literature.
- The KTH dataset
  - Our approach: 87.25%
  - Schuldt et al.[8]: 71.72%
  - Niebles et al.[9]: 81.5%
  - Laptev et al.[10]: 91.8%
  - Messing et al.[11]: 74%

[8] Schuldt, C., I. Laptev and B. Caputo, "Recognizing Human Actions: A Local SVM Approach", Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., Vol. 3, pp. 32-36 Vol.3, IEEE, 2004.

[9] Niebles, J. C., H. Wang and L. Fei-Fei, "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words", International Journal of Computer Vision, Vol. 79, No. 3, pp. 299-318, Mar. 2008.

[10] Laptev, I., M. Marszalek, C. Schmid and B. Rozenfeld, "Learning realistic human actions from movies", Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1-8, IEEE, 2008.

[11] Messing, R., C. Pal and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints", IEEE 12th International Conference on Computer Vision, pp. 104-111, Sep. 2009.

Conclusions (2)

- The URADL dataset
  - Our approach: 88%
  - Messing et al. (Velocity Histories): 63%
  - Messing et al. (Latent Velocity Histories): 67%
  - Messing et al. (Augmented Velocity Histories): 89%
  - However, augmented velocity histories method of Messing et al. is highly dependent on the position of the action on the frame and it requires the position of the face.
- A new dataset is introduced: WeCare
- Our performance of the WeCare dataset is 98.75%
- More challenging scenarios should be added to the WeCare dataset for testing our work.

Future directions

- More robust elimination module
- New trajectory features
- Improved recognition performance
- Human action detection system
- Trajectories in 3D space
- Real time recognition support

THANK YOU