RTip: A Fully Automated Root Tip Tracker for Measuring Plant Growth With Intermittent Perturbations

ICIP 2020 - 1067:(ARS-02.14) Biomedical and Biological Image Analysis

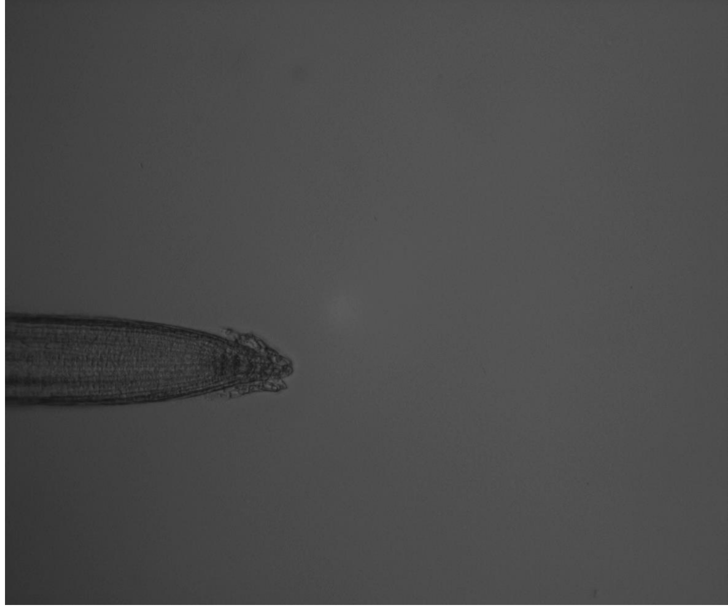
Presented by: Deniz Kavzak Ufuktepe

Deniz Kavzak Ufuktepe[1], Kannappan Palaniappan[1], Melissa Elmali[2], Tobias I. Baskin[2]

1 Electrical Engineering and Computer Science Dept., University of Missouri, Columbia, MO, USA

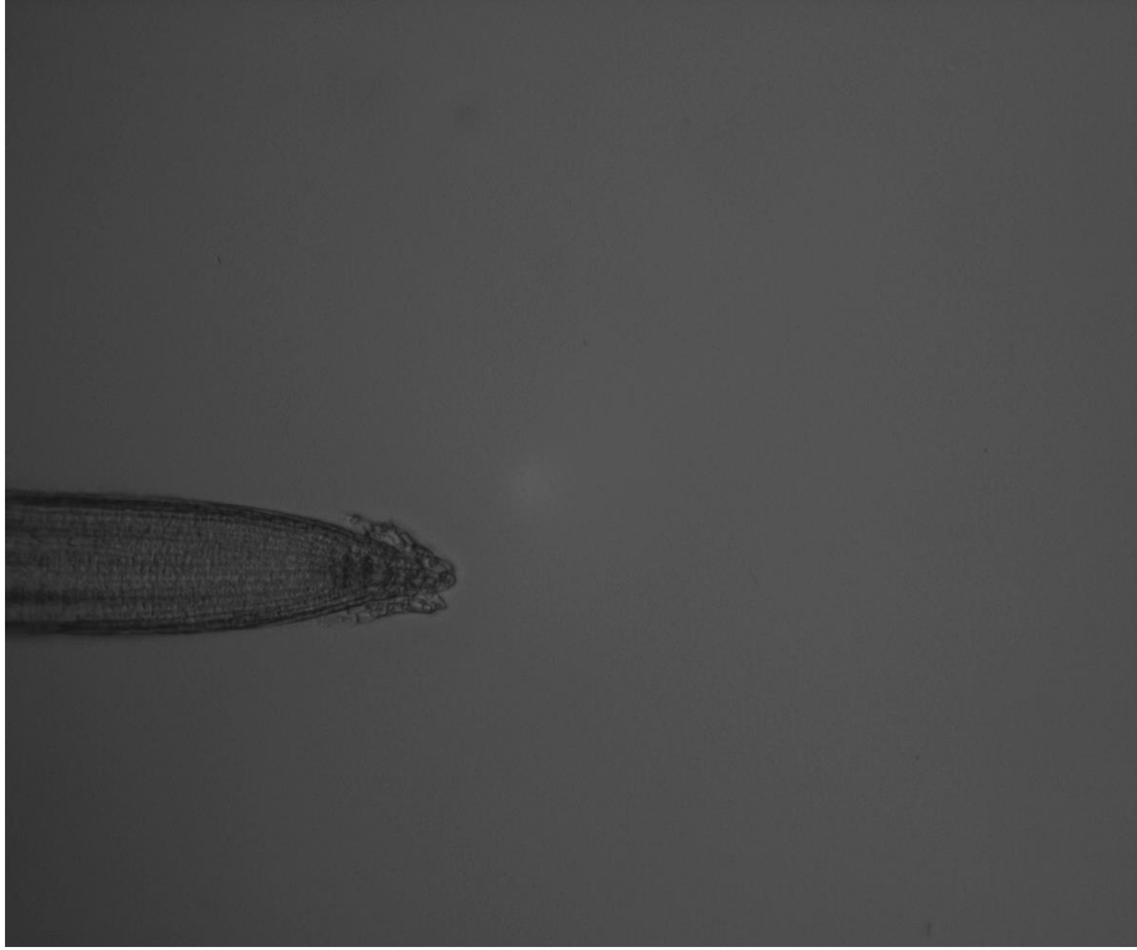
2 Biology Department, University of Massachusetts, Amherst, MA, USA

Background



- Observe growth kinematics for plant physiology studies
- Root tip tracking for analysis of plant phenotyping
  - Growth rate
- Measure effects of different manipulations on root - perturbations
  - e.g. cut plant shoot, apply substance

Root Tip Tracking



- High resolution root tip microscopy
image sequence (2448x2048)

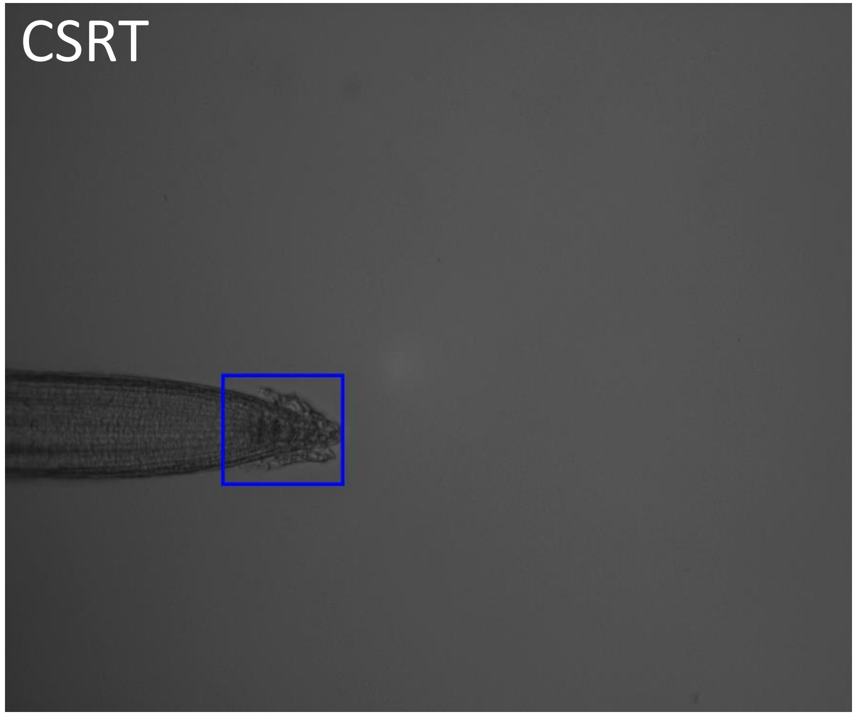
- Tracking the root tip for velocity estimation
    - Manually too error prone,
    even when generating ground truth

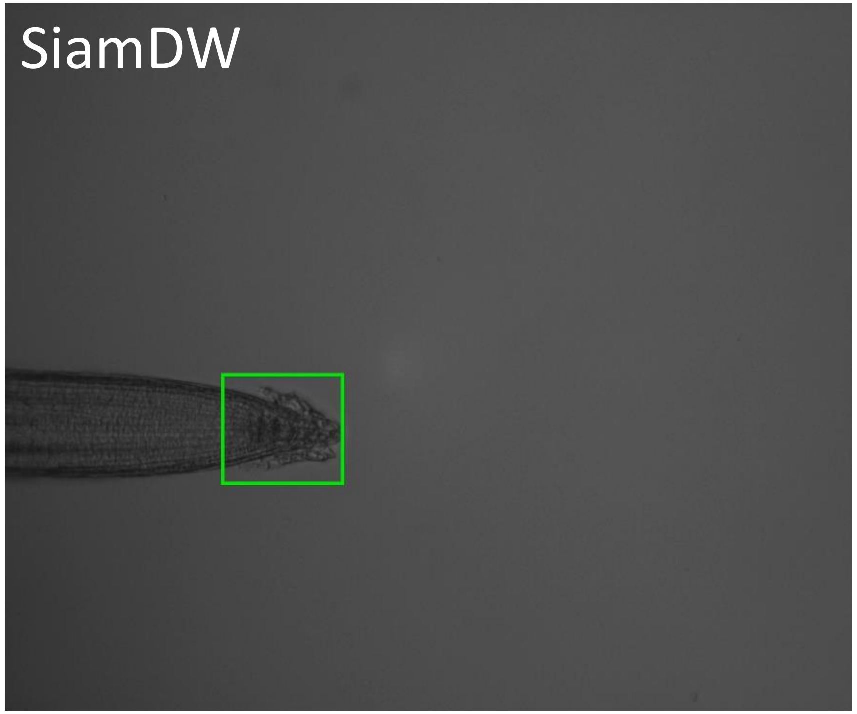
- The sequence is perturbed for an unknown number of frames
    - Cutting the root shoot – analyze effects

- Many collected videos
    - Manually too long, too many videos to be analyzed

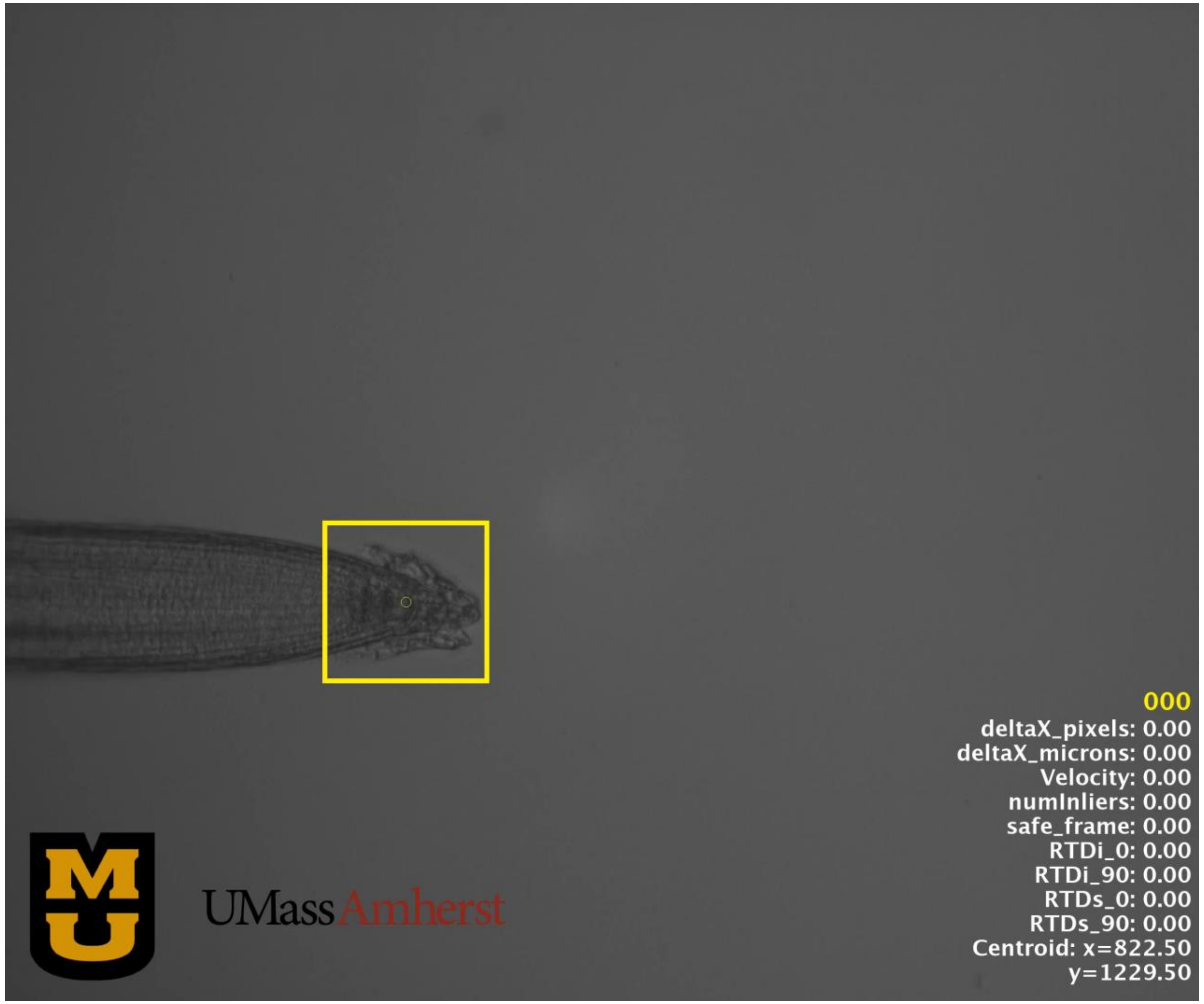
Root Tip Tracking

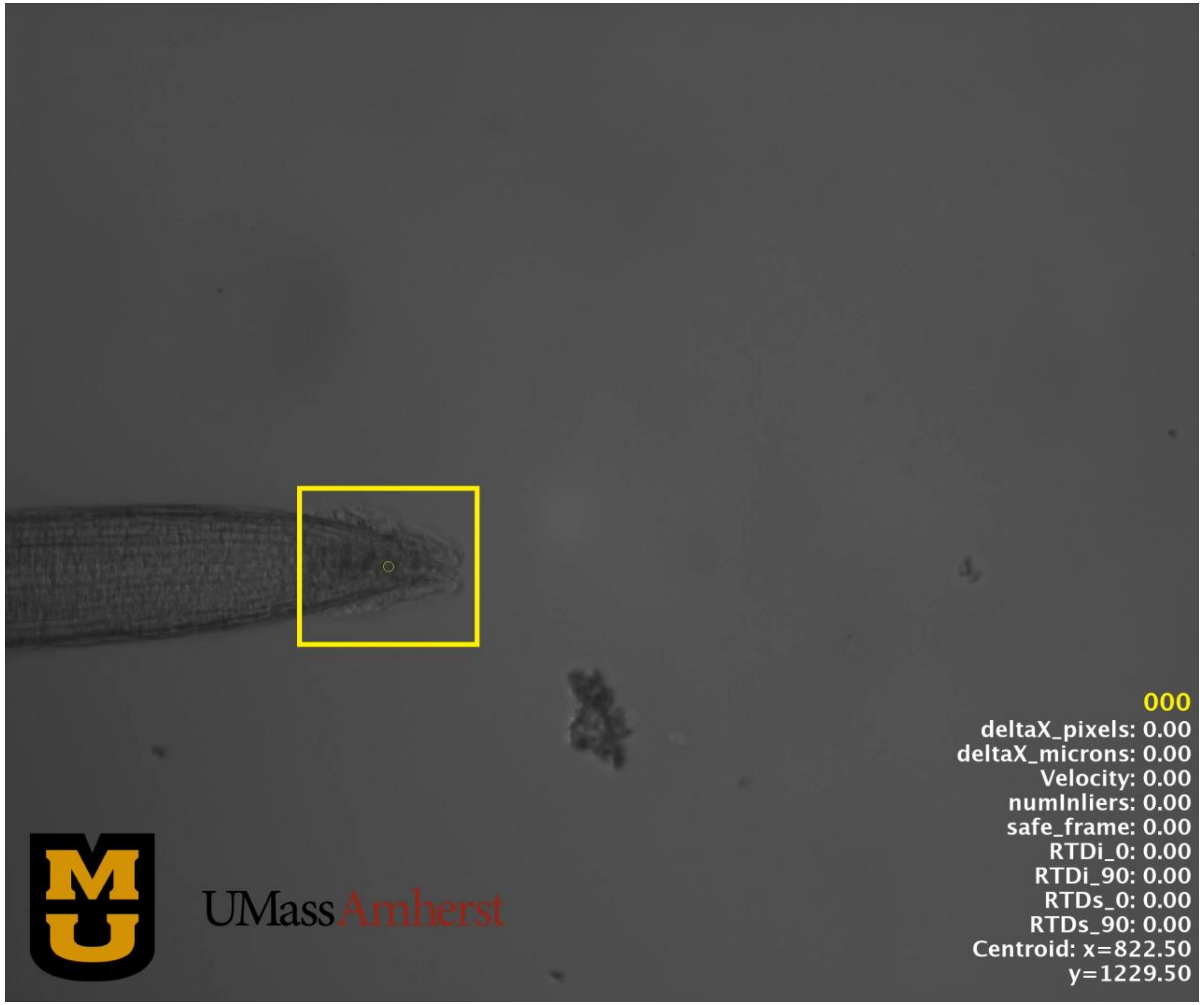
- Subpixel accuracy in high-resolution
  - Smooth transition between (bounding box) BBoxes
  - No jitter, no size changes in BBoxes
- Time and duration of perturbation is unknown
  - Automatic identification of root + root tip
    - Root tip might be gone, or blurry
  - Automatic recovery
- Current well-performing trackers:
  - fail to track accurately
  - needs manual re-initialization





Rtip Results





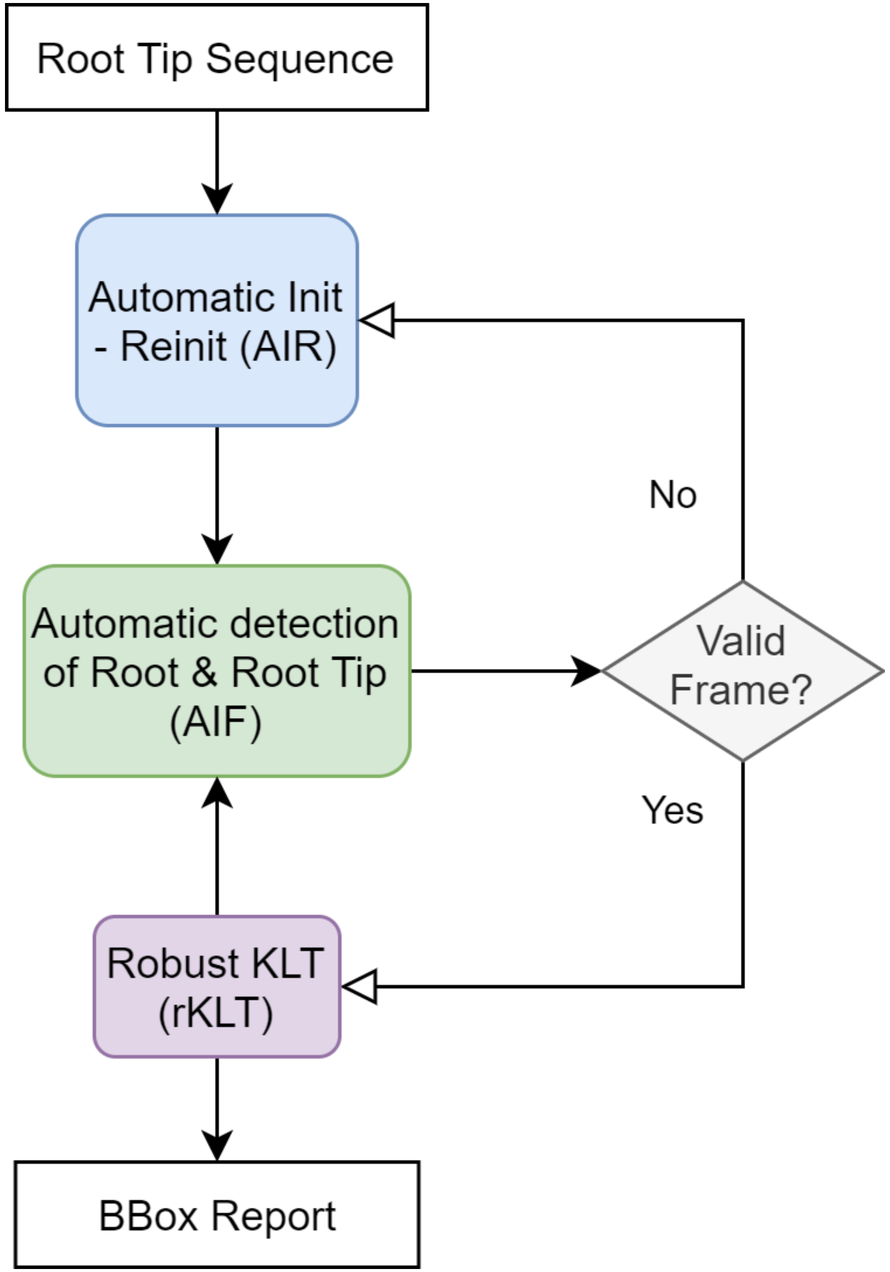
RTip Algorithm

Fully Automated root tip tracker

- Automatic Init and Reinit (AIR) with BBox improvement
  - Automatic localization of root tip
  - Identify blurry images with root tip
  - Improved BBox – best fit in neighborhood

- Automatic Invalid Frame (AIF) Detection-Recovery
  - Automatic identification:
    - Invalid frames – no root
    - Valid frames - root

- Robust (Kanade–Lucas–Tomasi feature tracker) KLT (rKLT) Tracking with (M-estimator sample consensus) MSAC Outlier Filtering
  - Smooth BBox transition



Kanade–Lucas–Tomasi feature tracker (KLT)

- Shi-Tomasi corner detector -> feature points
  - Eigen values of structure tensor for each pixel p: λ_1, λ_2, threshold λ
  - Pixel p is corner <--> $\min(\lambda_1 \lambda_2) > \lambda$

- Track feature points $\boldsymbol{x}$ in time $t \rightarrow t + \tau$:
  - $I(x, y, t + \tau) = I(x - \xi, y - \eta, t)$

- Model displacement $\boldsymbol{d} = (\xi, \eta)$ of the point $\boldsymbol{x} = (x, y)$ between t and $t + \tau$
  - Motion vectors -> Gradient of Hessian matrix at $\boldsymbol{x}$

- Minimize the error of noise η

- Carlo Tomasi and T Kanade, “Detection and Tracking of point features,” Tech. Rep., Tech. Rep. CMU-CS-91- 132, Carnegie Mellon University, 1991.
- Jianbo Shi and Tomasi, "Good features to track," CVPR, Seattle, WA, USA, 1994, pp. 593-600.

Automatic Init and Reinit (AIR) with bbox improvement

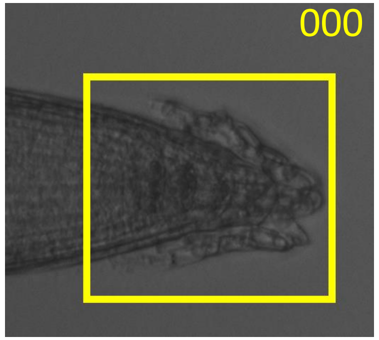
- Template + Normalized Cross Correlation (NCC) → initial candidate BBox

- Neighbor BBoxes of initial candidate
  - KLT* to track points, count # of inliers (Forward-Backward error)
  - Chose the BBox with max inliers – (re)start rKLT tracker

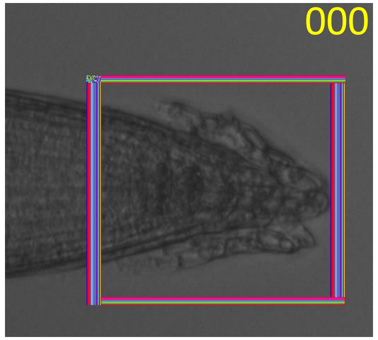
- Blurry, corrupted, perturbed frame with root
  - maximum # of inliers < threshold (i.e. 1500 points)
  - Frame is not good to restart tracker
  - Run AIR for next frame

* Carlo Tomasi and T Kanade Detection, “Tracking of point features,” Tech. Rep., Tech. Rep. CMU-CS-91- 132, Carnegie Mellon University, 1991.

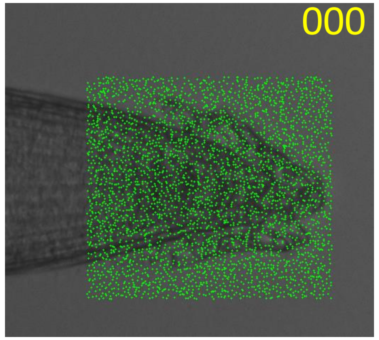
Automatic Init and Reinit (AIR) with bbox improvement



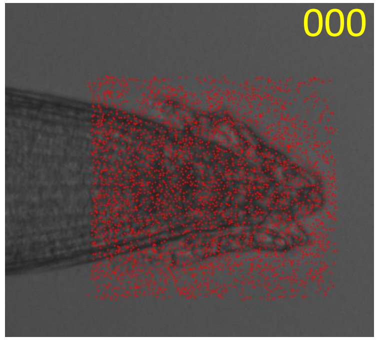
init candidate



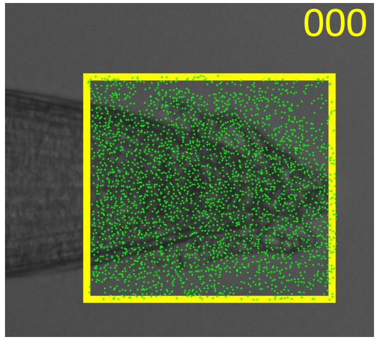
all candidates



Features of
init candidate

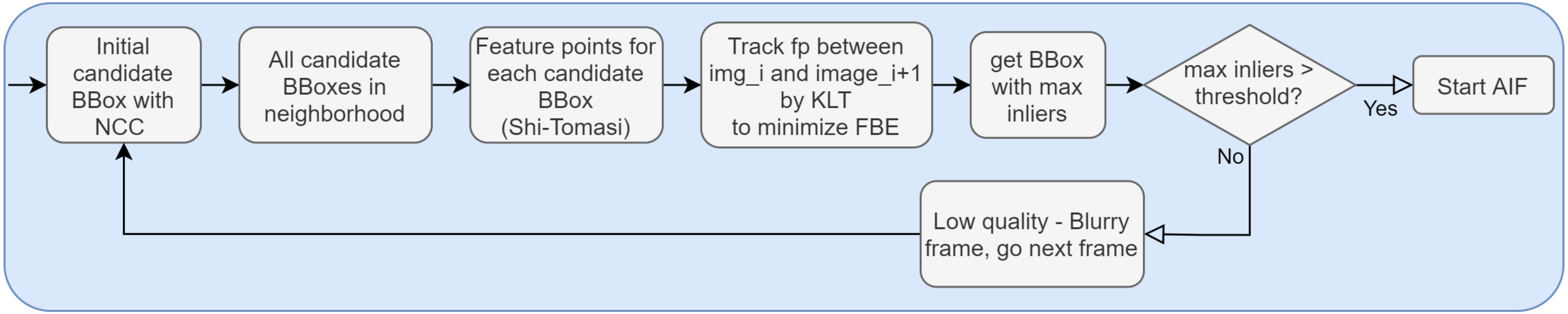


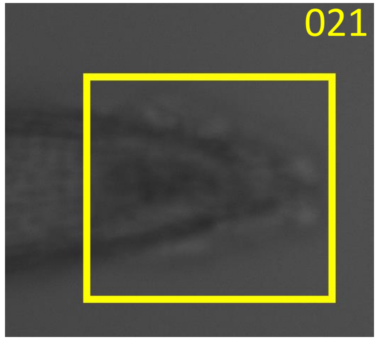
Features after KLT
with FBE



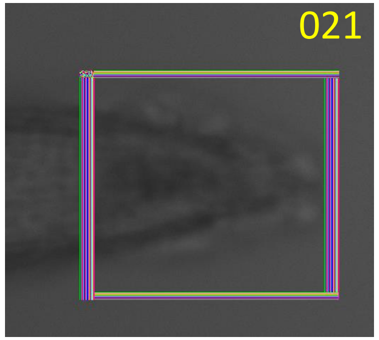
Best bbox
inliers = 2907

Automatic Init - Reinit (AIR)

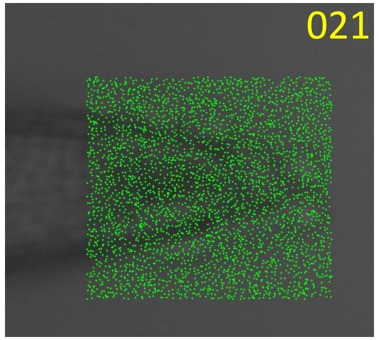




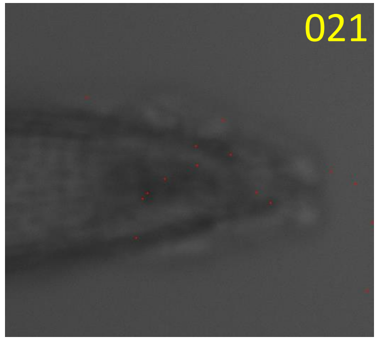
init candidate



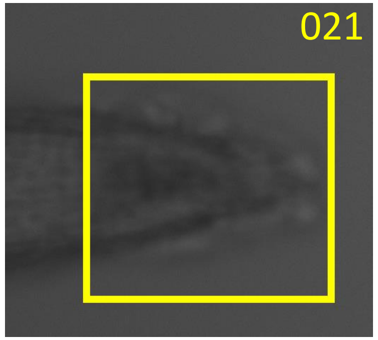
all candidates



Features of init candidate



Features after KLT with FBE



Best bbox # inliers = 18

Simplified Radon Transform (RT)

- RT operator calculates projections of an object along specified angles by line integrals

- Coordinates are rotated by each $\theta \in [0,180]$

- A set of parallel lines are integrated that are perpendicular to the rotated axis

$$R_\theta(x') = \int_{-\infty}^{\infty} f(x'cos\theta - y'sin\theta, x'sin\theta + y'cos\theta)dy' \,,$$

$$where \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Simplified RT: $\theta = 0$ and $\theta = 90$ for simpler images

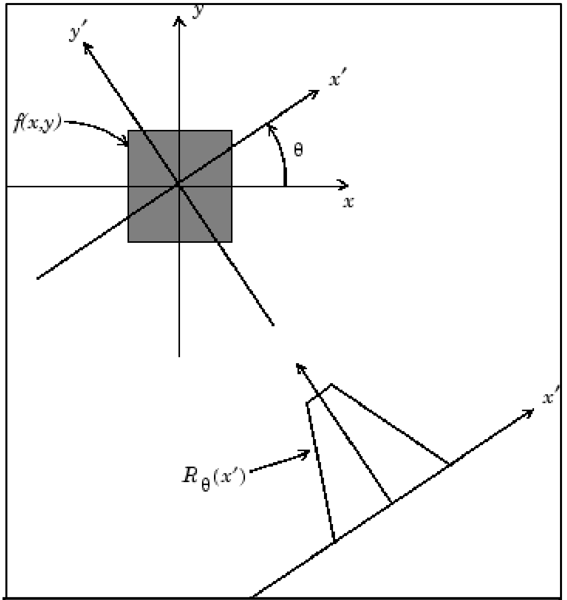


Fig. 1.* RT of an object over θ

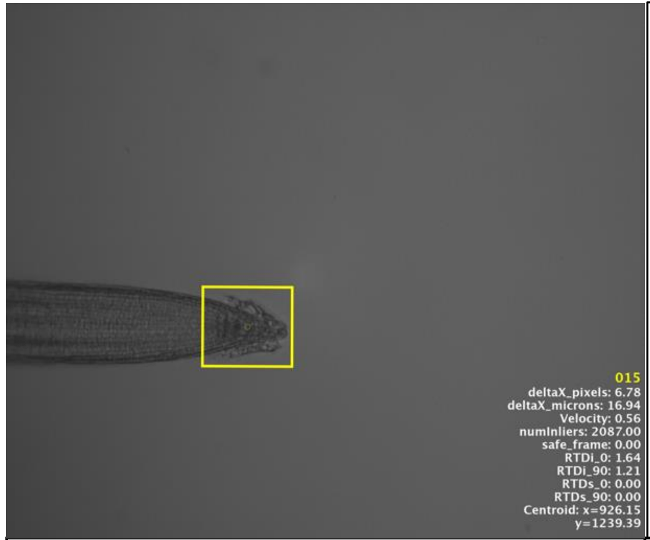


Fig. 2. Image frame 015 from sequence

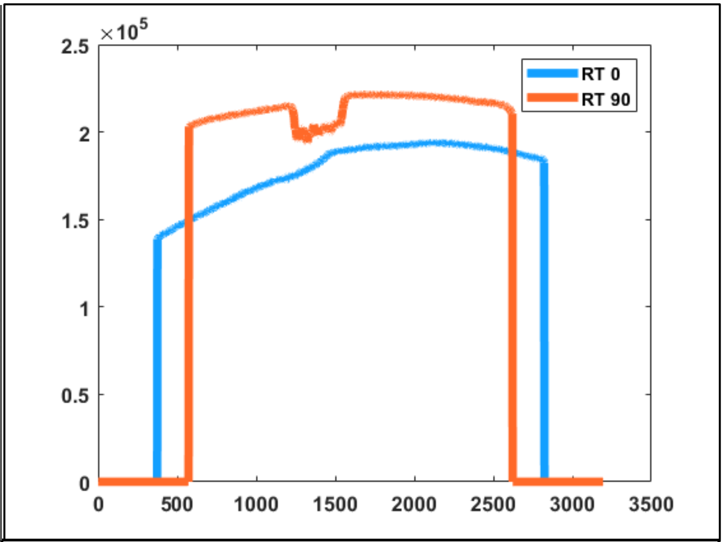
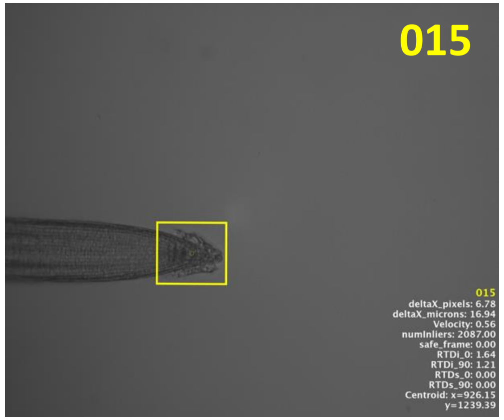
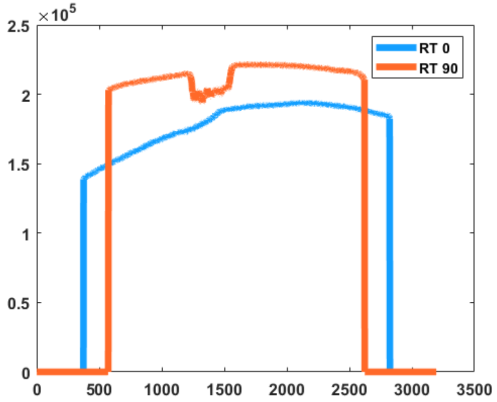


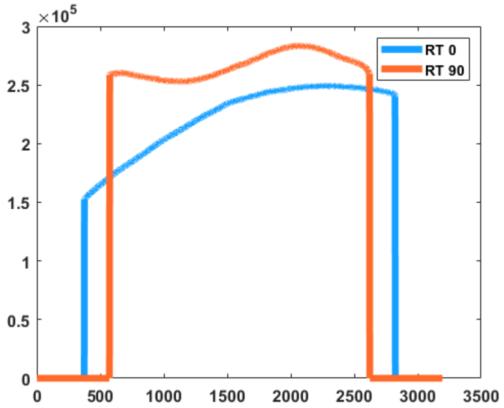
Fig. 3. RT for 0 and 90 degrees of frame 015

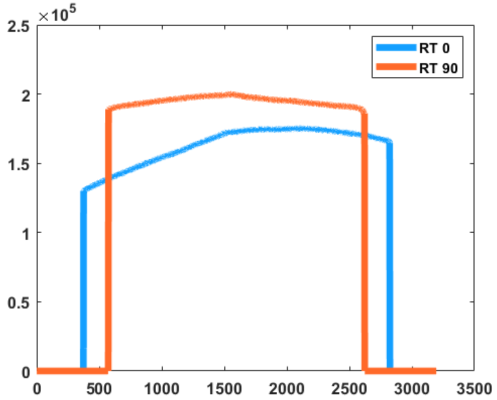
* Picture from MATLAB web pagehttps://www.mathworks.com/help/images/radon-transform.html .

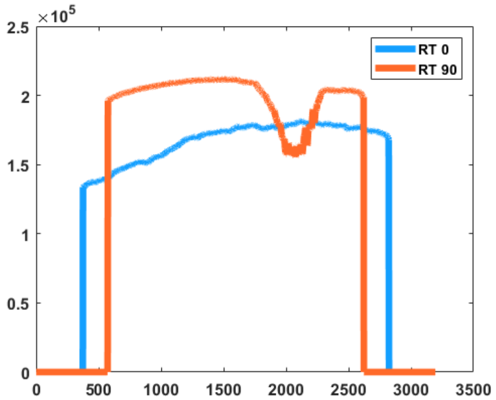
Simple root + root tip identification using radon transform

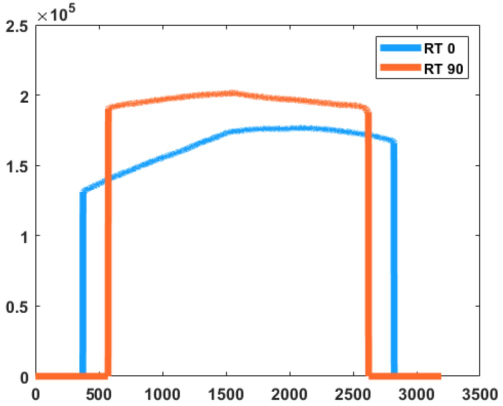


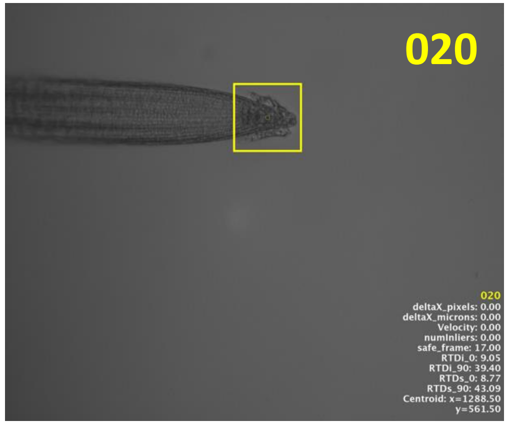


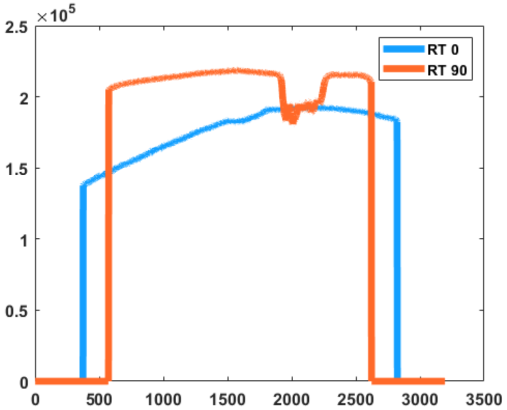










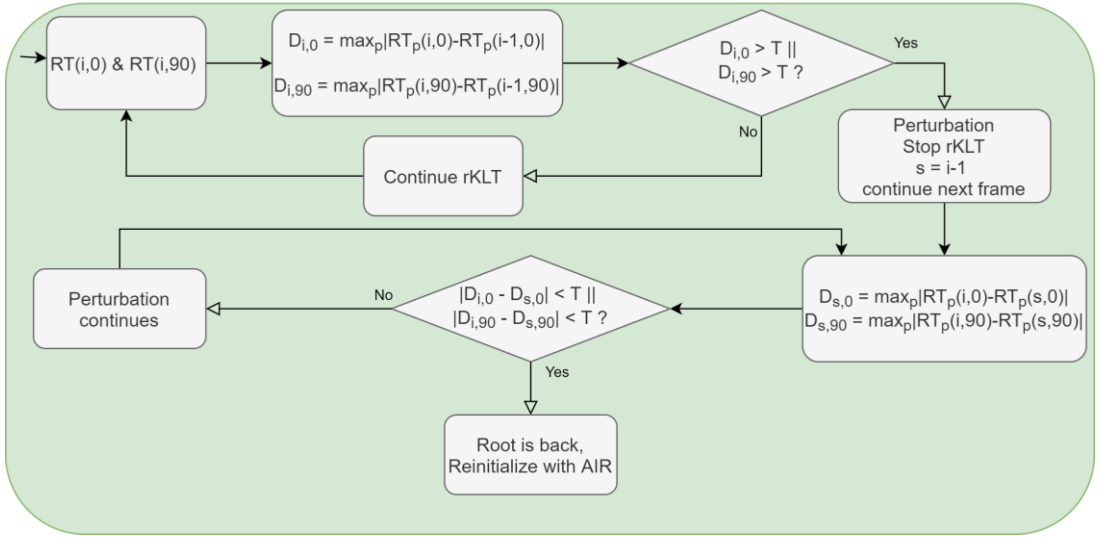


Automatic Invalid Frame (AIF) Detection-Recovery

- Apply Simplified Radon Transform (RT) on 0 and 90 degrees

- Find Chebyshev distance between RTs of each degree
- $D^{RT}_{i,\theta}(\boldsymbol{RT}_i, \boldsymbol{RT}_{i-1}) = \max_p |RT_{i,p} - RT_{i-1,p}|$ and $D^{RT}_{s,\theta}(\boldsymbol{RT}_i, \boldsymbol{RT}_s) = \max_p |RT_{i,p} - RT_{s,p}|$
  - p: projection vector axis, i: current frame, s: safe frame

Automatic Invalid Frame (AIF) Detection & Recovery for presence of root tip



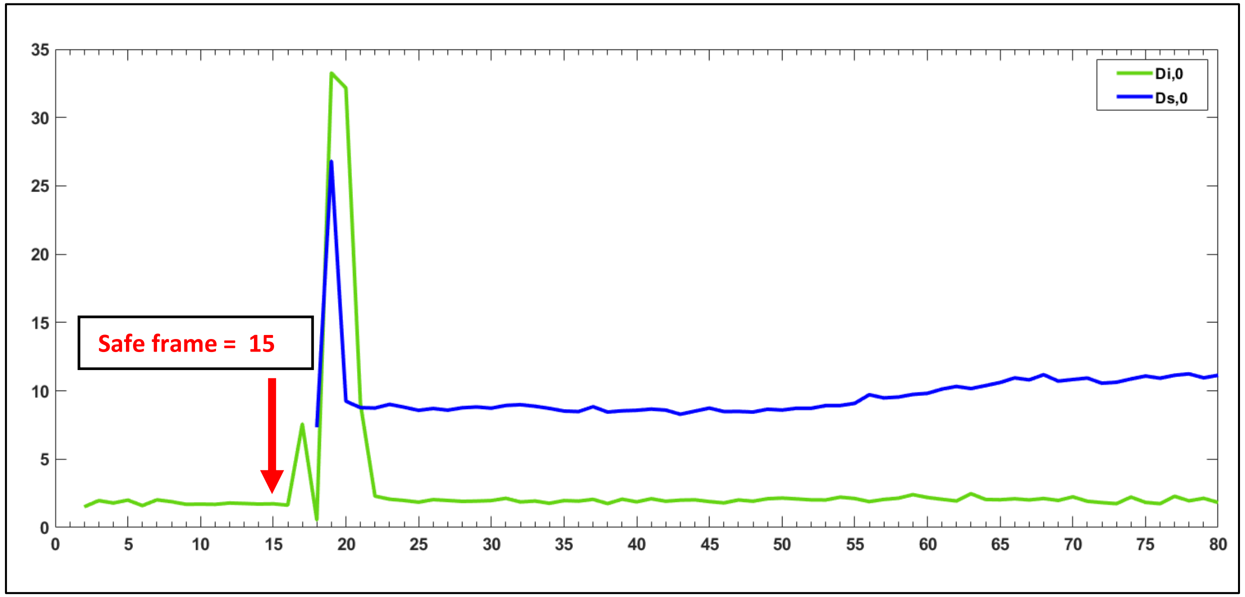


Fig. 1. $D^{RT}_{i,0}$ and $D^{RT}_{s,0}$ distance values for the whole sequence

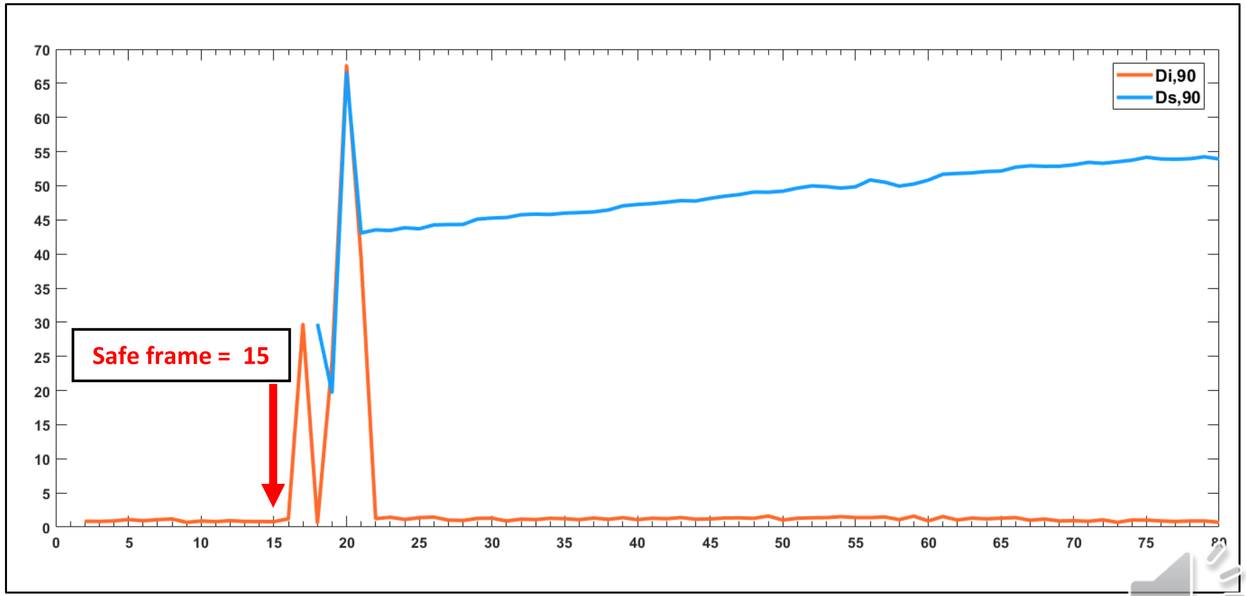
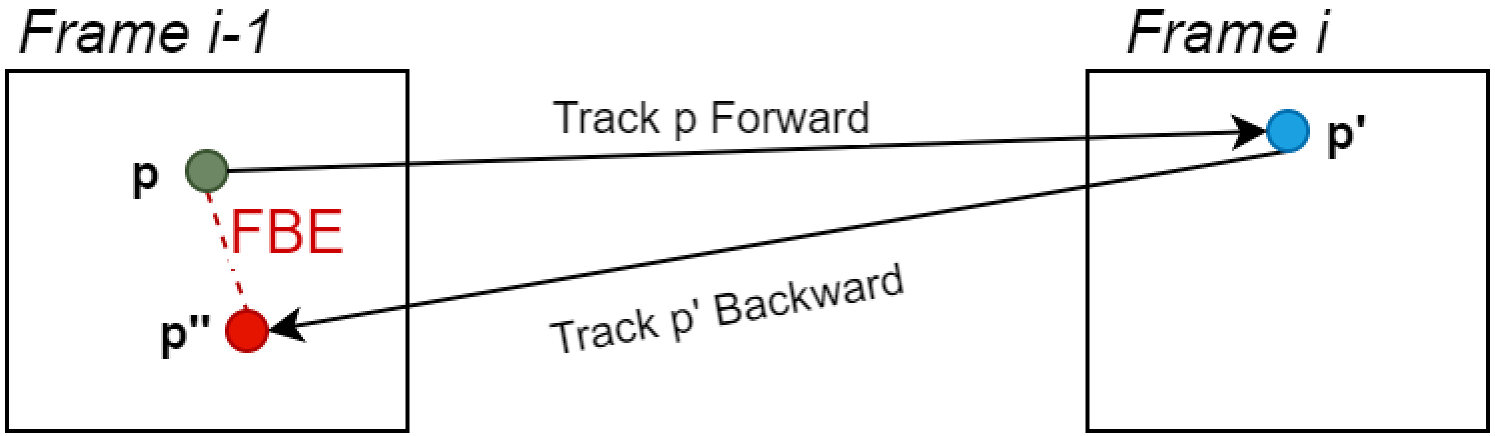


Fig. 2. $D^{RT}_{i,90}$ and $D^{RT}_{s,90}$ distance values for the whole sequence

Robust KLT (rKLT) Tracking with MSAC Outlier Filtering

- Track Shi-Tomasi feature points from the Bbox with KLT

- KLT tracks feature points – rKLT: Minimize Forward-Backward Error (FBE*)



- Estimate Similarity transformation between feature points and tracked points
  - Use Forward-Backward Error again to eliminate outliers
  - M-estimator sample consensus (MSAC**) - outlier elimination in trans. Estimation

- Transform the Bbox to the next frame by the transformation

(*) Z. Kalal, K. Mikolajczyk and J. Matas, "Forward-Backward Error: Automatic Detection of Tracking Failures," ICPR, Istanbul, 2010, pp. 2756-2759
(**) Torr, Philip HS, and Andrew Zisserman. "MLESAC: A new robust estimator with application to estimating image geometry." Computer vision and image understanding 78.1 (2000): 138-156

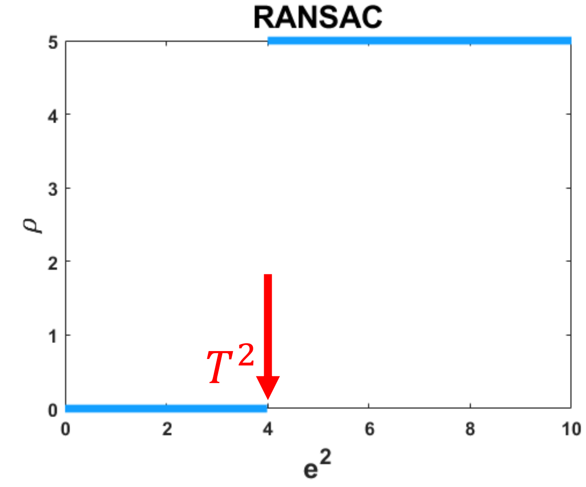
RANSAC vs. M-estimator sample consensus (MSAC)

- RANSAC is effective in finding the minimum cost for the function

$c = \sum_i \rho(e_i^2)$, where *ρ()* is $\rho(e^2) = \begin{cases} 0 & e^2 < T^2 \\ constant & e^2 \geq T^2 \end{cases}$

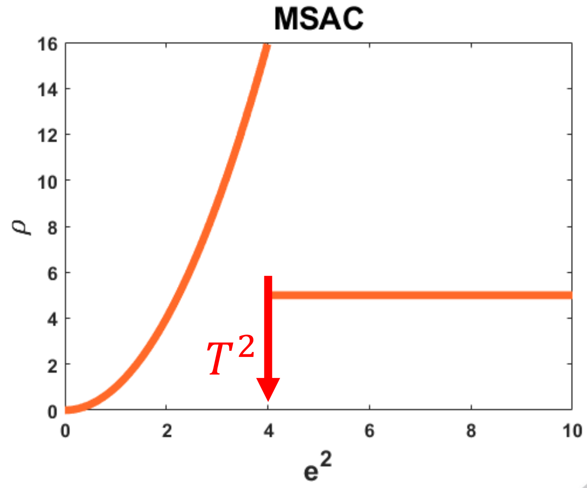
and T is the threshold.

  - inliers don't have a score and outliers score a constant penalty
  - higher T causes more solutions to have same cost, leads poor estimation



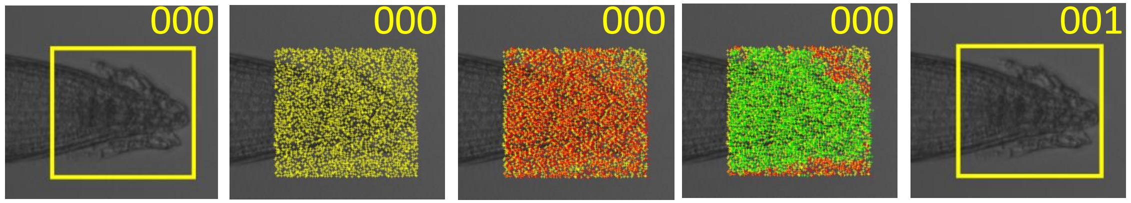
- MSAC* minimizes same cost func. with robust error term *ρ()* is
- $\rho(e^2) = \begin{cases} e^2 & e^2 < T^2 \\ constant & e^2 \geq T^2 \end{cases}$

  - outliers are still given a fixed penalty, but
  - inliers are scored on how well they fit the data.

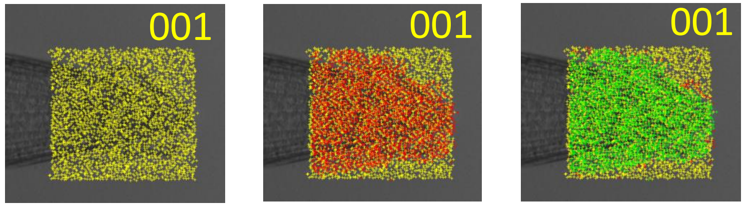


(*) Torr, Philip HS, and Andrew Zisserman. "MLESAC: A new robust estimator with application to estimating image geometry." Computer vision and image understanding 78.1 (2000): 138-156

Robust KLT (rKLT) Tracking with MSAC Outlier Filtering



Bbox to track | Extracted features # features = 3004 | Inliers after KLT with FBE # inliers = 2663 | Inliers after MSAC # inliers = 2265 | New BBox

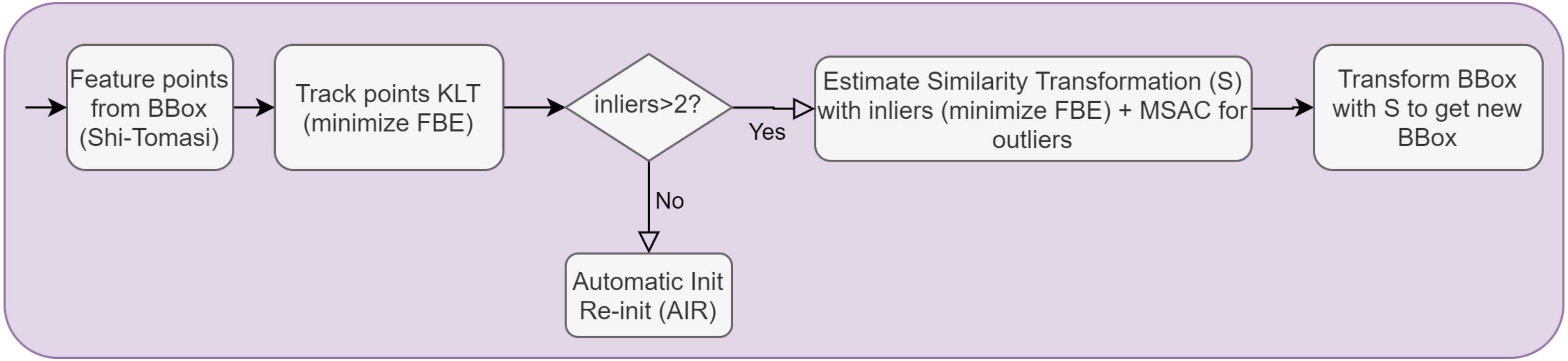


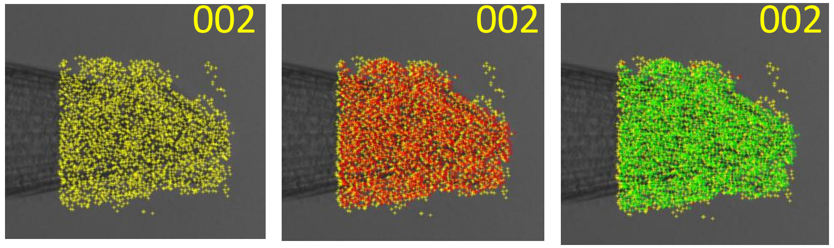
Tracked features # features = 3004 | Inliers after KLT with FBE # inliers = 2177 | Inliers after MSAC # inliers = 2122

$$S = \begin{bmatrix} 0.9996448 & 0.0013135 & 0 \\ -0.0013135 & 0.9996448 & 0 \\ 8.1890259 & 0.0521240 & 1 \end{bmatrix}$$

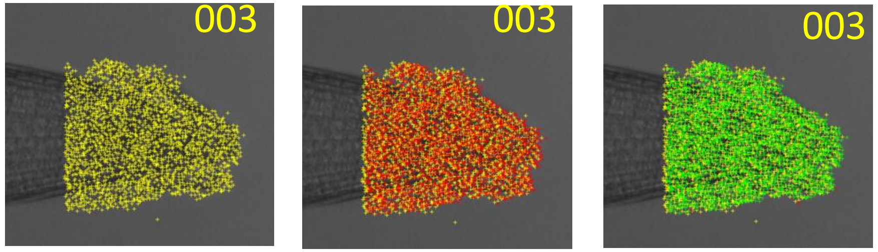
$$displacement = (8.1890259, 0.0521240)$$

Robust KLT (rKLT)





Tracked features # features = 2265 | Inliers after KLT with FBE # inliers = 2103 | Inliers after MSAC # inliers = 2096



Tracked features # features = 2122 | Inliers after KLT with FBE # inliers = 2087 | Inliers after MSAC # inliers = 2084

Experiments

- Experimented on 2 datasets
  - 2048x2448
  - 80 frames each – 160 in total
  - Sampling rate is 30 seconds per frame – total elapsed time is $\approx$ 40 min

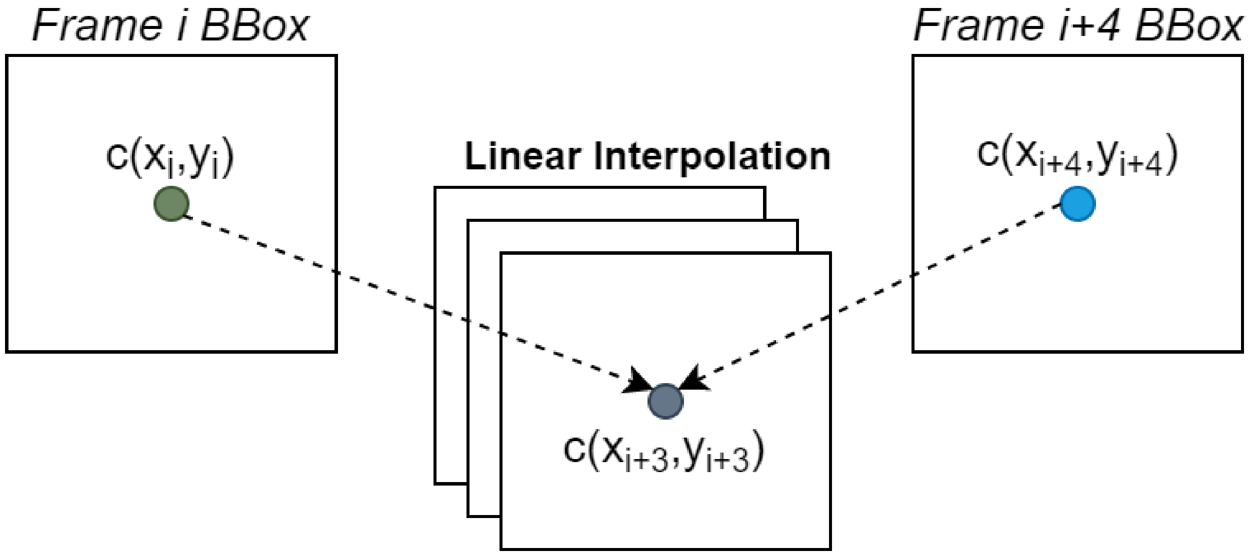
- 6 other trackers are also run on the same datasets for comparison

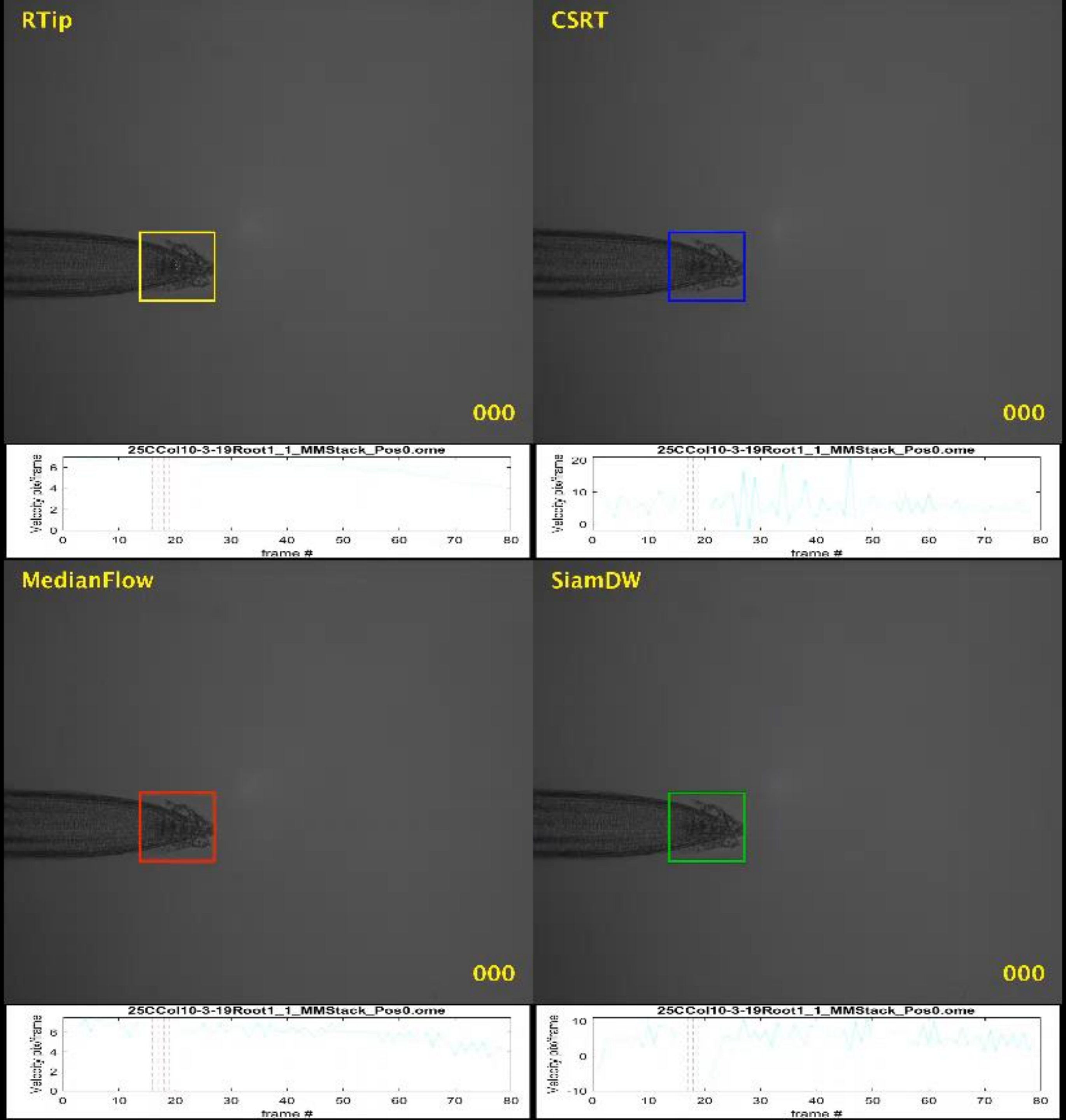
Trackers

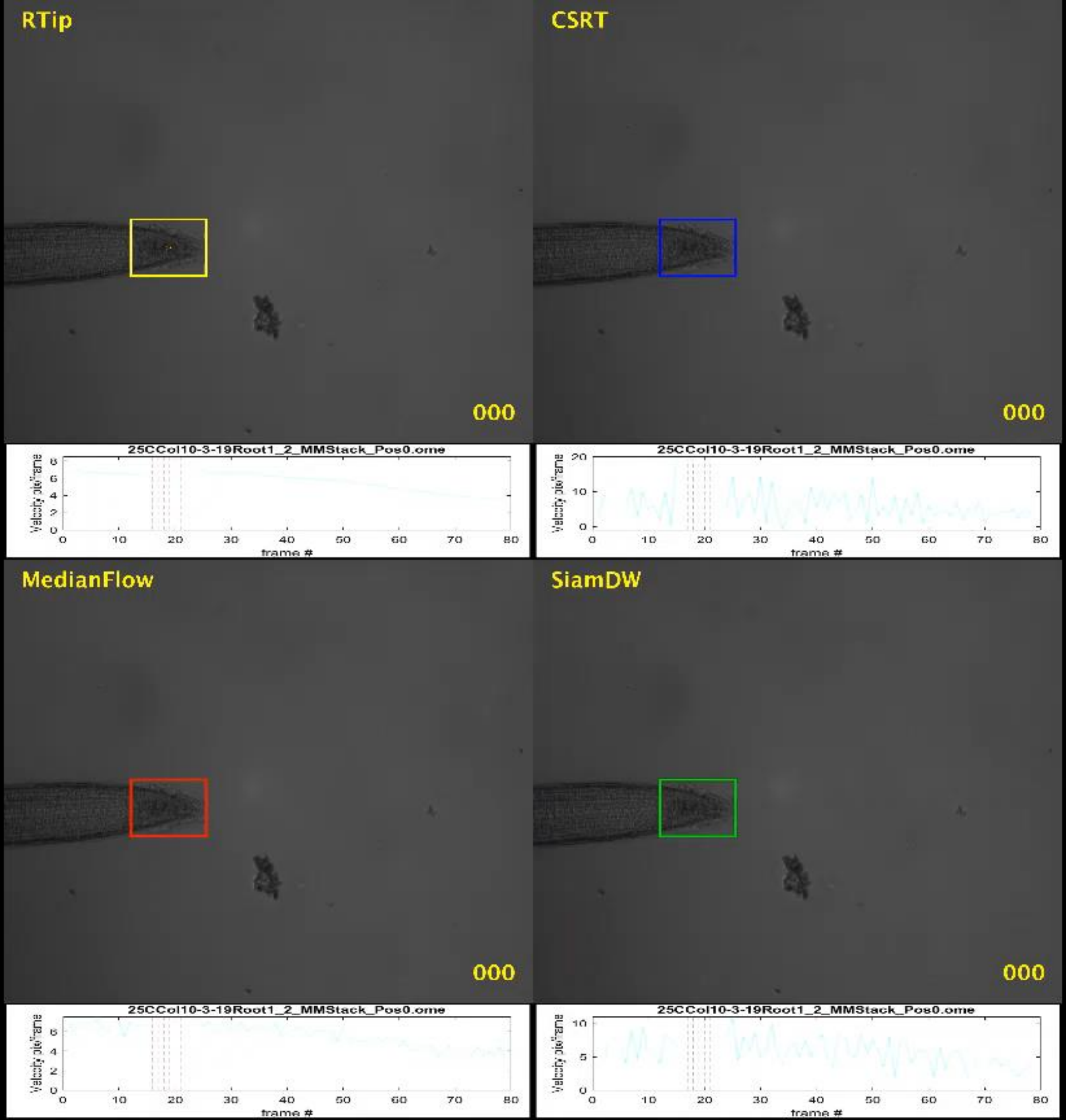
- Discriminative Correlation Filter Tracker (CSRT) – multiple feature channels
  - Lukezic, Alan, et al. "Discriminative correlation filter with channel and spatial reliability." CVPR 2017.
- SiamDW – deep learning, given default weights are used
  - Zhang, Zhipeng, and Houwen Peng. "Deeper and wider siamese networks for real-time visual tracking." CVPR 2019.
- Multiple Instance Learning (MIL) – discriminative classifier separate background-foreground
  - Babenko, Boris, et al. "Visual tracking with online multiple instance learning." CVPR 2009.
- MedianFlow – Forward-Backward error to detect tracking failures
  - Kalal, Zdenek, et al. "Forward-backward error: Automatic detection of tracking failures." ICPR 2010.
- Kernelized Correlation Filters (KCF) – color features – tracking by detection
  - Danelljan, Martin, et al. "Adaptive color attributes for real-time visual tracking." CVPR 2014.
- Tracking, learning and detection (TLD) – learn detection errors
  - Kalal, Zdenek, et al. "Tracking-learning-detection." IEEE transactions on pattern analysis and machine intelligence 34.7 (2011): 1409-1422.

Ground Truth Generation

- Start from initial bounding box of first frame
- Manually locate BBox in each 5th frame
- Generate Bbox in between by interpolating centroids of i and i+4




RTip
CSRT
MedianFlow
SiamDW
000
25CCol10-3-19Root1_1_MMStack_Pos0.ome
frame #


RTip
000
CSRT
000
MedianFlow
000
SiamDW
000
25CCol10-3-19Root1_2_MMStack_Pos0.ome
frame #

Results

Tracker	V_{err}	RMSE	SSIM	Adapt
GT	NA	6.23±1.68	0.81±0.05	NA
RTip	**0.49±0.34**	**6.37±1.58**	**0.80±0.05**	✓
MIL	0.56±0.61	6.41±1.90	0.80±0.06	×
MedFl	0.57±0.48	**6.35±1.88**	**0.80±0.05**	×
KCF	0.97±1.17	8.55±1.41	0.69±0.05	×
Siam	1.92±1.72	8.94±1.61	0.68±0.05	×
CSRT	2.64±2.66	8.04±1.87	0.71±0.07	×
TLD	13.01±30.09	10.23±1.78	0.68±0.05	×

- V_{err}
  - RTip 0.49 ± 0.34
- RMSE
  - GT 6.23±1.68
  - RTip 6.37±**1.58**
  - MedianFlow 6.35±1.88
- SSIM
  - GT 0.81 ± 0.05
  - RTip 0.80 ± 0.05
  - MedianFlow 0.80 ± 0.05
- RTip: Automatic reinit - Adaptive

- V_{err} = Difference between ground truth and tracker velocity estimation
- *RMSE* = Root mean squared error between the Bboxes and initial template
- *SSIM* = Structural Similarity Index between Bboxes and initial template
- *Adapt* = Adaptation to perturbed frames
- GT = Ground Truth (with linear interpolation)

Conclusion

- Simplified Radon Transform can be used to identify root and root tip
- KLT with FBE + similarity transformation estimation with FBE + MSAC eliminate outliers - robust BBox transition
- NCC with KLT+FBE to find best BBox can be used to identify root tip - good recovery

Future Work

- Machine learning to select distance threshold value intelligently that will work with all sequences without manual tuning
- Using contour information for tracking
- Use a deep learning detector for faster and accurate recovery
  - YOLO – retrain with variety of root species
- Extend RTip to detect medial line of the root
  - DeepFlux – deep learning algorithm finds skeleton of objects

Thank you for listening..

Questions?