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 Ufuktepe¹, Kannappan Palaniappan¹, Melissa Elmali², Tobias I. Baskin²

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

deltaX_pixels: 0.00
deltaX_microns: 0.00
Velocity: 0.00
numInliers: 0.00
safe_frame: 0.00
RTDI_0: 0.00
RTDi_90: 0.00
RTDs_0: 0.00
RTDs_90: 0.00
Centroid: x=822.50
y=1229.50

UMass Amherst
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: $\lambda_1, \lambda_2$, threshold $\lambda$
  • Pixel p is corner $\iff \min(\lambda_1 \lambda_2) > \lambda$

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

• Model displacement $d = (\xi, \eta)$ of the point $x = (x, y)$ between $t$ and $t + \tau$
  • Motion vectors $\rightarrow$ Gradient of Hessian matrix at $x$

• Minimize the error of noise $\eta$

- 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

Automatic Init and Reinit (AIR) with bbox improvement

Best bbox

# inliers = 2907

Features after KLT with FBE

Features of init candidate

Features of all candidates

init candidate

Automatic Init - Reinit (AIR)

Initial candidate BBox with NCC

All candidate BBoxes in neighborhood

Feature points for each candidate BBox (Shi-Tomasi)

Track fp between img_i and image_i+1 by KLT to minimize FBE

get BBox with max inliers

max inliers > threshold?

Yes

Start AIR

No

Low quality - Blurry frame, go next frame

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.

$$R_\theta(x') = \int_{-\infty}^{\infty} f(x' \cos \theta - y' \sin \theta, x' \sin \theta + y' \cos \theta) dy', \quad \text{where } \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

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

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

\[ D_i^{RT}(\mathbf{RT}_i, \mathbf{RT}_{i-1}) = \max_p |\mathbf{RT}_{i,p} - \mathbf{RT}_{i-1,p}| \quad \text{and} \quad D_s^{RT}(\mathbf{RT}_i, \mathbf{RT}_s) = \max_p |\mathbf{RT}_{i,p} - \mathbf{RT}_{s,p}| \]

- \( p \): projection vector axis, \( i \): current frame, \( s \): safe frame

\[ D_{i,0} = \max_p |\mathbf{RT}_{i,0} - \mathbf{RT}_{i-1,0}| \]
\[ D_{s,0} = \max_p |\mathbf{RT}_{i,0} - \mathbf{RT}_{0,0}| \]

\[ D_{i,90} = \max_p |\mathbf{RT}_{i,90} - \mathbf{RT}_{i-1,90}| \]
\[ D_{s,90} = \max_p |\mathbf{RT}_{i,90} - \mathbf{RT}_{0,90}| \]

Fig. 1. \( D_i^{RT} \) and \( D_s^{RT} \) distance values for the whole sequence

Fig. 2. \( D_{i,90}^{RT} \) and \( D_{s,90}^{RT} \) distance values for the whole sequence

Safe frame = 15
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

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 \( \rho() \) is \( \rho(e^2) = \begin{cases} 
  0 & e^2 < T^2 \\
  \text{constant} & e^2 \geq T^2 
\end{cases} \)  
  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 \( \rho() \) is
  \[ \rho(e^2) = \begin{cases} 
  e^2 & e^2 < T^2 \\
  \text{constant} & e^2 \geq T^2 
\end{cases} \]

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

Robust KLT (rKLT) Tracking with MSAC Outlier Filtering

```
\begin{bmatrix}
0.9996448 & 0.0013135 & 0 \\
-0.0013135 & 0.9996448 & 0 \\
8.1890259 & 0.0521240 & 1
\end{bmatrix}
```

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

\[ \text{displacement} = (8.1890259, 0.0521240) \]

Robust KLT (rKLT)

1. Feature points from BBox (Shi-Tomasi)
2. Track points KLT (minimize FBE)
3. \( \text{inliers} > 2? \)
   - Yes: Estimate Similarity Transformation (S) with inliers (minimize FBE) + MSAC for outliers
   - No: Transform BBox with S to get new BBox
4. Automatic Init Re-init (AIR)

```
\begin{bmatrix}
000 & 000 & 000 & 001 \\
\end{bmatrix}
```

```
\begin{bmatrix}
001 & 001 & 001 \\
\end{bmatrix}
```

```
\begin{bmatrix}
002 & 002 & 002 \\
\end{bmatrix}
```

```
\begin{bmatrix}
003 & 003 & 003 \\
\end{bmatrix}
```
Experiments

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

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

- **Discriminative Correlation Filter Tracker (CSRT) – multiple feature channels**

- **SiamDW – deep learning, given default weights are used**

- **Multiple Instance Learning (MIL) – discriminative classifier separate background-foreground**

- **MedianFlow – Forward-Backward error to detect tracking failures**

- **Kernelized Correlation Filters (KCF) – color features – tracking by detection**

- **Tracking, learning and detection (TLD) – learn detection errors**
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
Results

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

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

- $V_{err}$
  - 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
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?