Contour Detection in Multi-Angle Time-Lapse Images of Growing Plants

Merel te Hofsté
Supervisor: Neil Budko

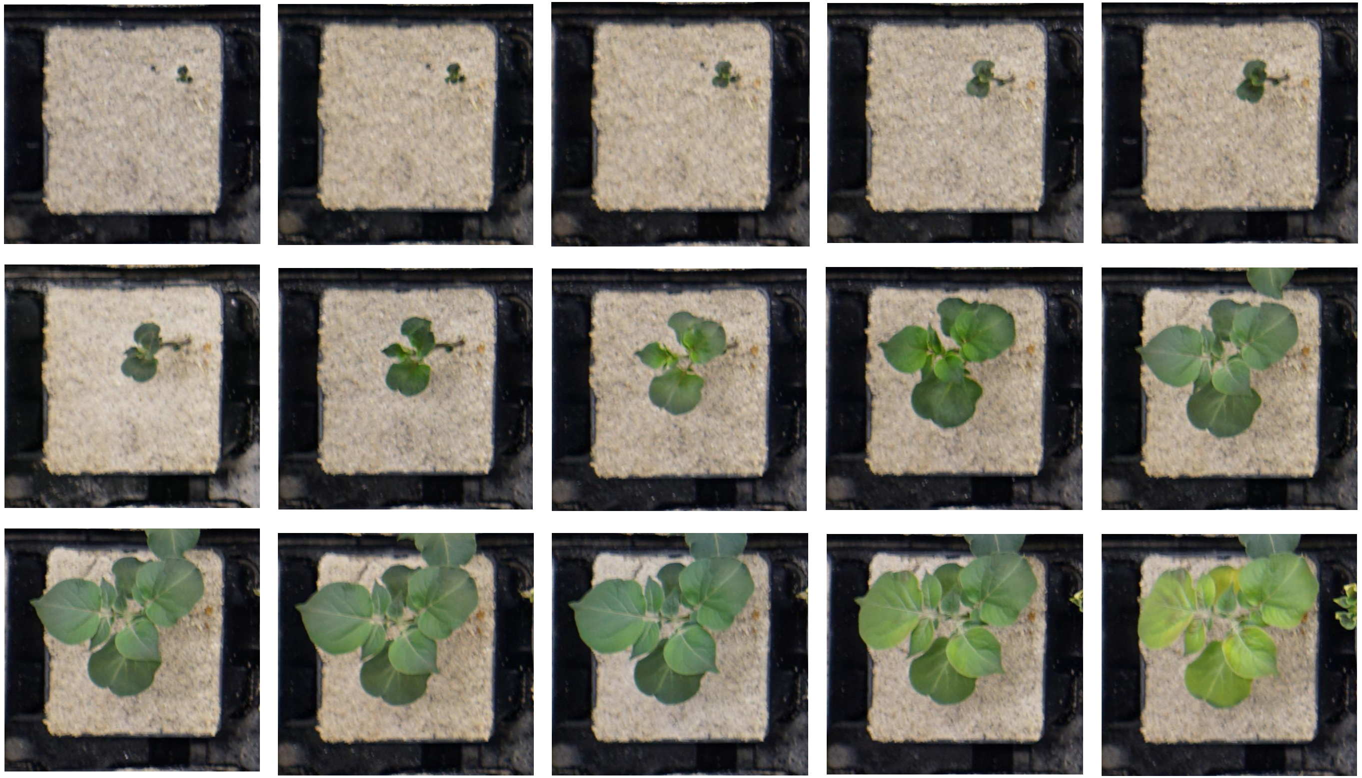
TU Delft

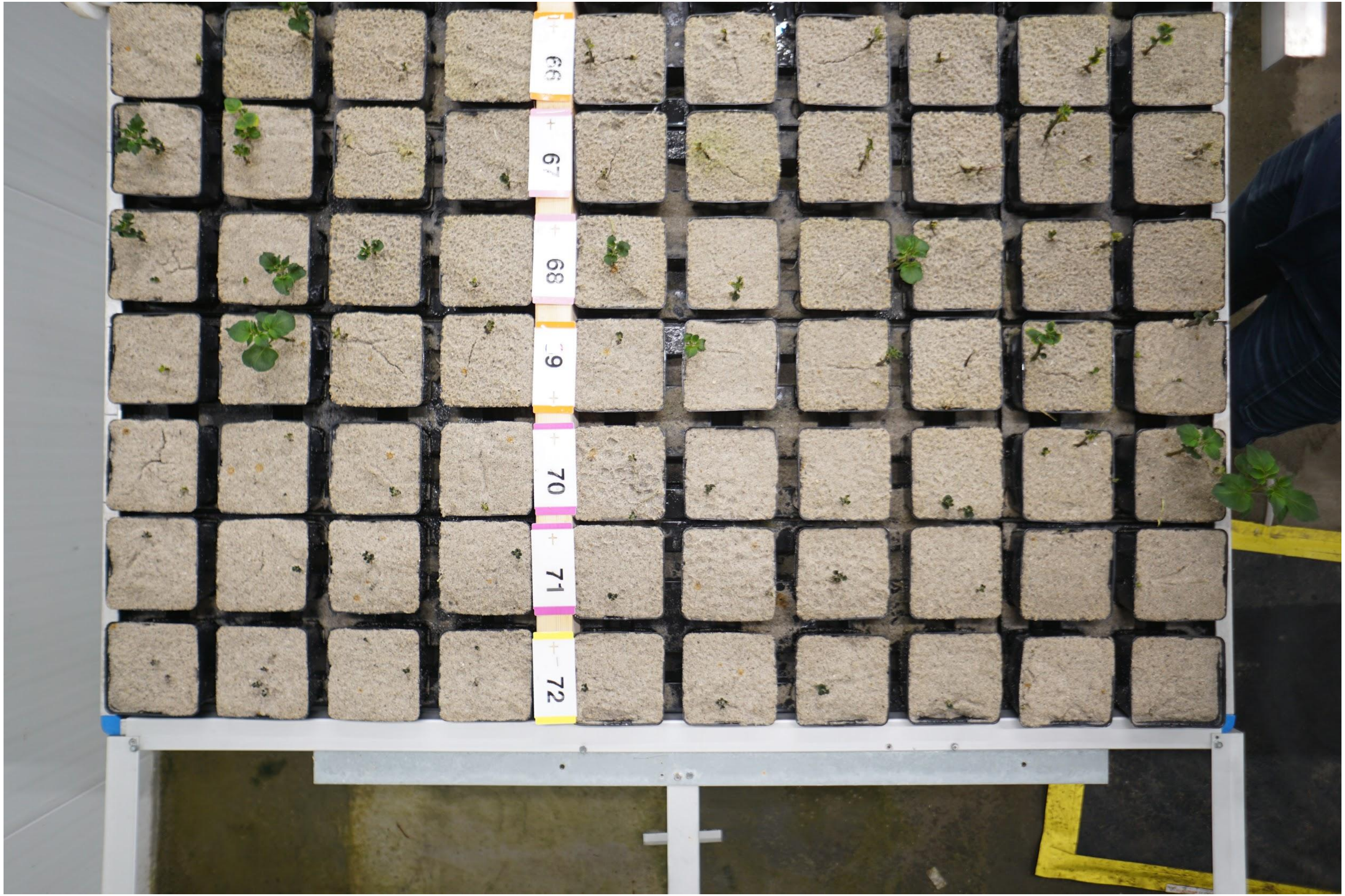
Quantifying growth

- Six potato varieties
- 2 Climate rooms
- Dry and wet sections

HZPC
keeps you growing

Problem




66
67
68
70
71
72

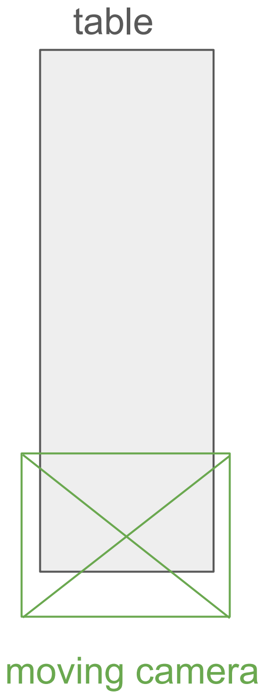
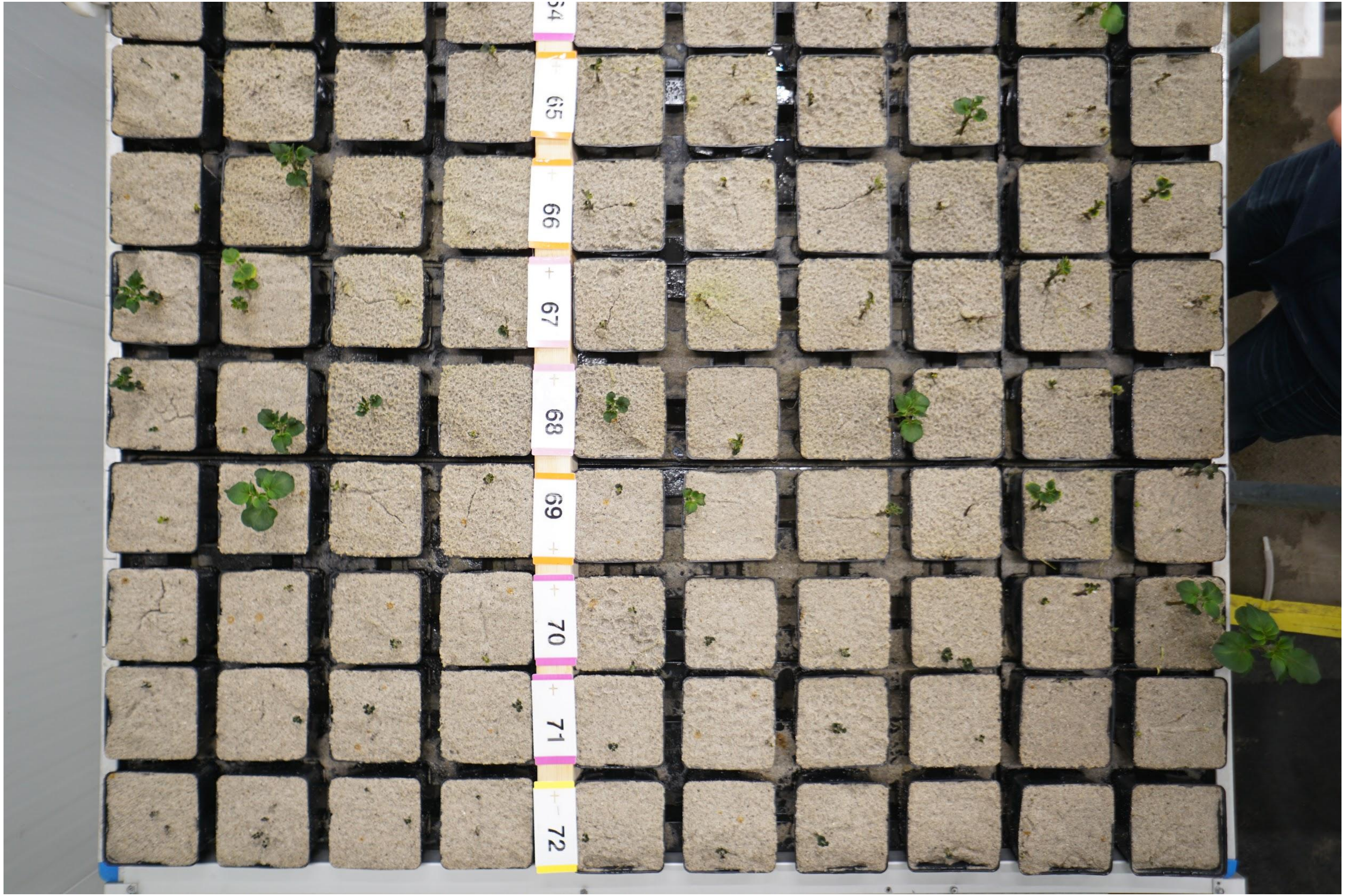
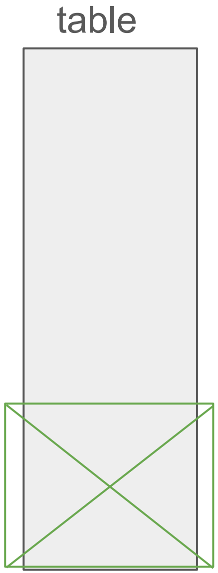


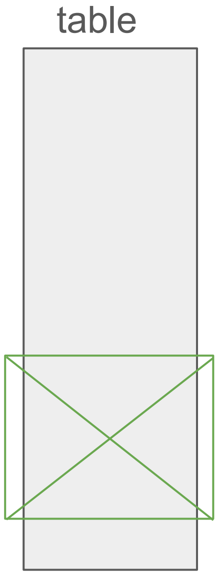
table
moving camera


65
66
67
68
69
70
71
72



table


63
64
65
66
67
68
69
70



table

Overview

Contour detection

To detect the outline of the plant

3D Contour Reconstruction

Using a pinhole camera model

Double Snakes

Combination of previous problems

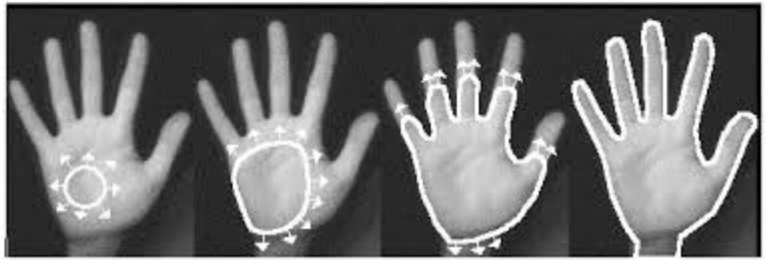
Conclusion

Conclusion and recommendations

Traditional Snakes

Contour detection

- Active contours to detect outline in image



Traditional Snakes

- Energy functional of a traditional snake parameterized by $\mathbf{q}(s) = [u(s), v(s)]$

$$E^*_{\text{snake}} = \int_0^1 E_{\text{snake}}(\mathbf{q}(s))\,\mathrm{d}s$$
$$= \int_0^1 E_{\text{int}}(\mathbf{q}(s)) + E_{\text{ext}}(\mathbf{q}(s))\,\mathrm{d}s$$

- α and β control parameters for amount of stretch and curvature

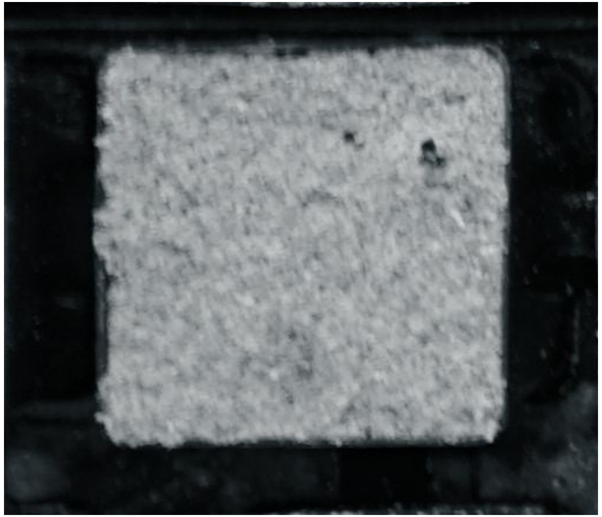
$$E[\mathbf{q}] = \int_0^1 \frac{1}{2}\left(\alpha \left\|\mathbf{q}'(s)\right\|^2 + \beta \left\|\mathbf{q}''(s)\right\|^2\right) + E_{\text{ext}}(\mathbf{q}(s))\,\mathrm{d}s$$

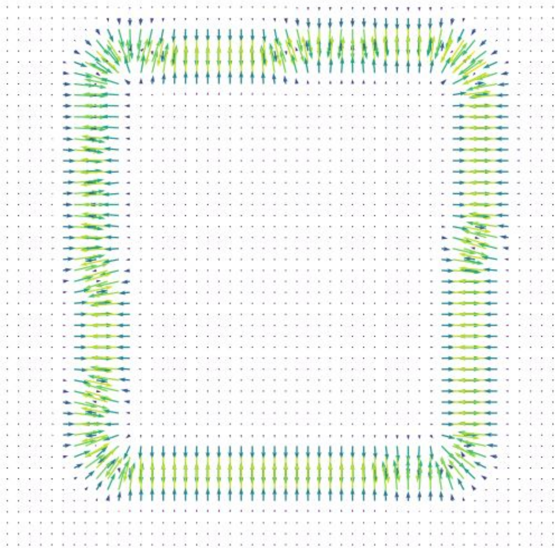
- Assume local minimum of E in $\mathbf{q}$ to derive the Euler-Lagrange equation

$$\alpha\mathbf{q}'' - \beta\mathbf{q}'''' - \nabla E_{\text{ext}}(\mathbf{q}) = 0$$

Traditional Snakes

- Typical external energy $E_{\text{ext}}(\mathbf{q}) = -\left|\nabla I(\mathbf{q})\right|^2$ which points towards regions of interest in image I





Gradient Vector Flow

- New external force field $\mathbf{w}(u,v) = [\phi(u,v), \psi(u,v)]$ which minimizes the functional

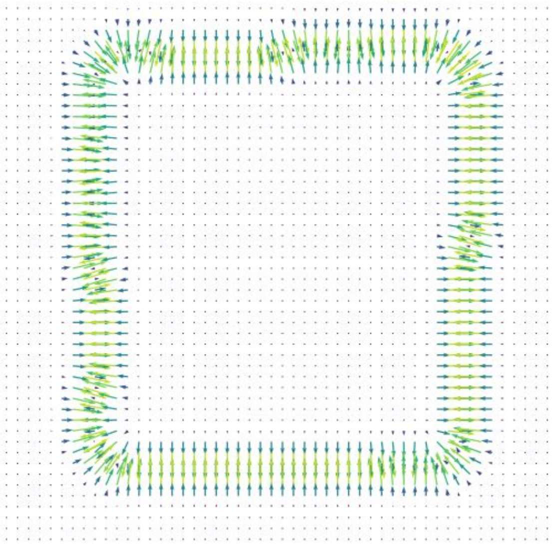
$$\mathcal{E}[\mathbf{w}] = \iint \mu \left(\phi_u^2 + \phi_v^2 + \psi_u^2 + \psi_v^2\right) + |\nabla f|^2 |\mathbf{w} - \nabla f|^2 \, \mathrm{d}u \, \mathrm{d}v$$

μ smoothing term and edge map f derived from image

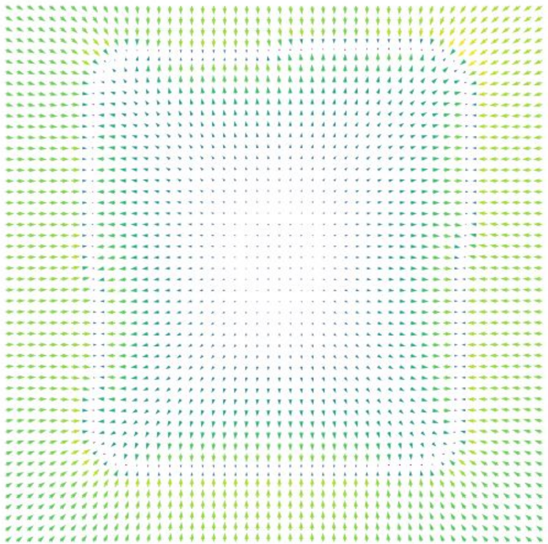
- GVF Snake

$$\alpha \mathbf{q}'' - \beta \mathbf{q}'''' + \mathbf{w} = 0$$

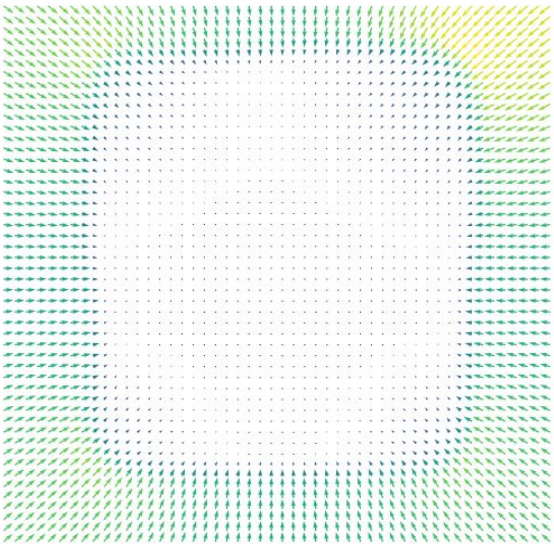
Gradient Vector Flow



μ=0



μ=1e-3



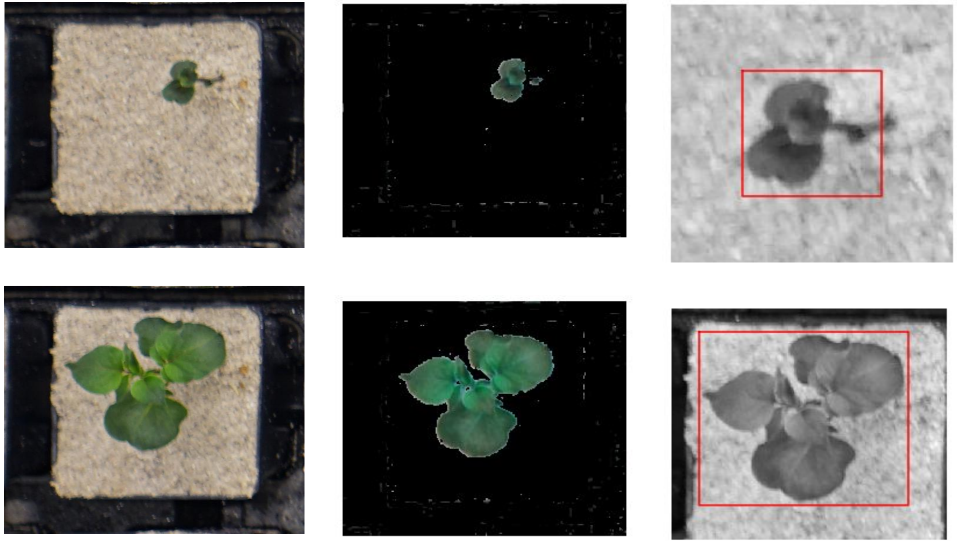
μ=0.1

GVF-Snakes

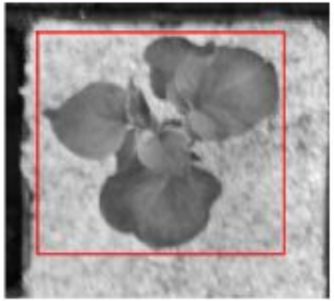
- Solve for $\frac{\partial \mathbf{q}}{\partial t} = \alpha \frac{\partial^2 \mathbf{q}}{\partial s^2} - \beta \frac{\partial^4 \mathbf{q}}{\partial s^4} + \mathbf{w}(\mathbf{q})$

  with time-integration method

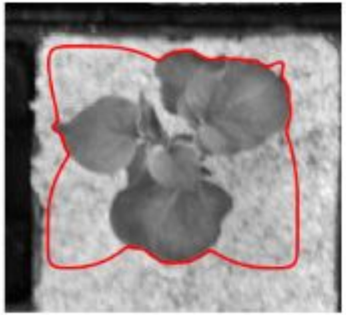
- Initial contour



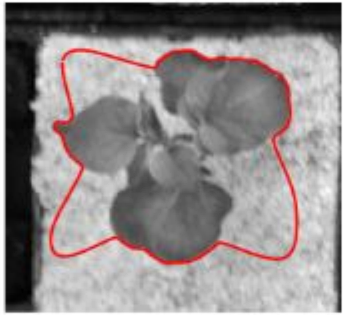
Results



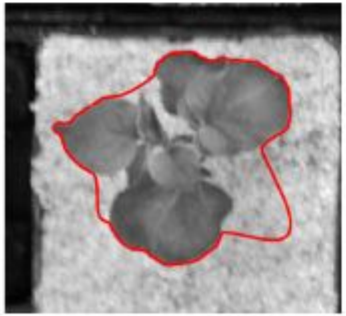
(a) $t = 0$



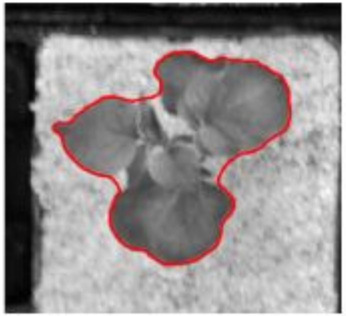
(b) $t = 1$



(c) $t = 5$



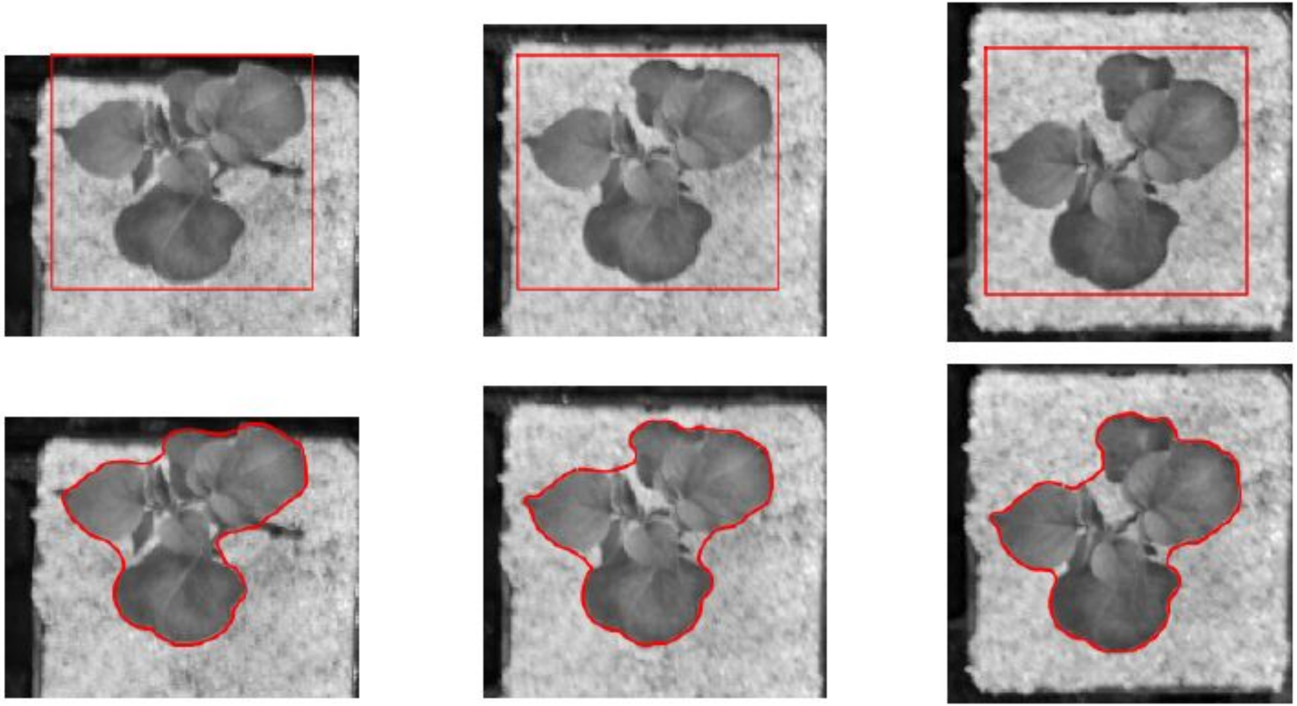
(d) $t = 10$



(e) $t = 50$

Results

Contour detection

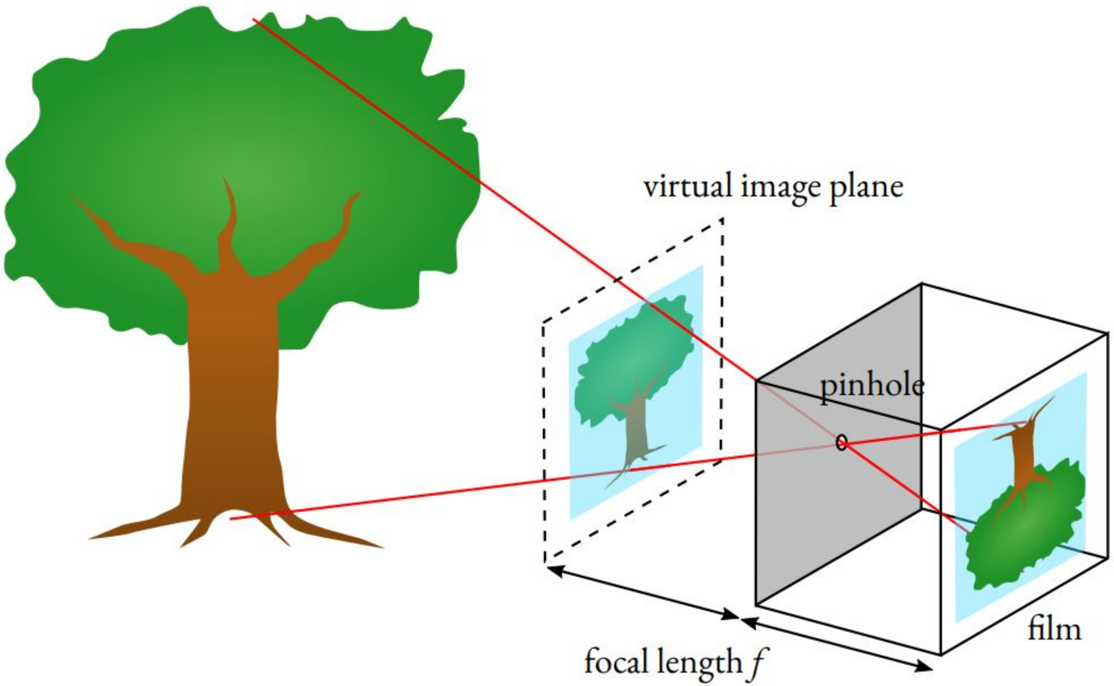


(a) Angle -1 (b) Angle 0 (c) Angle +1

GVF-Snakes

- Algorithm to detect outline
- Success ensured by initial contour and preprocessing steps

Pinhole camera model



Pinhole camera model

- Point in camera reference frame:

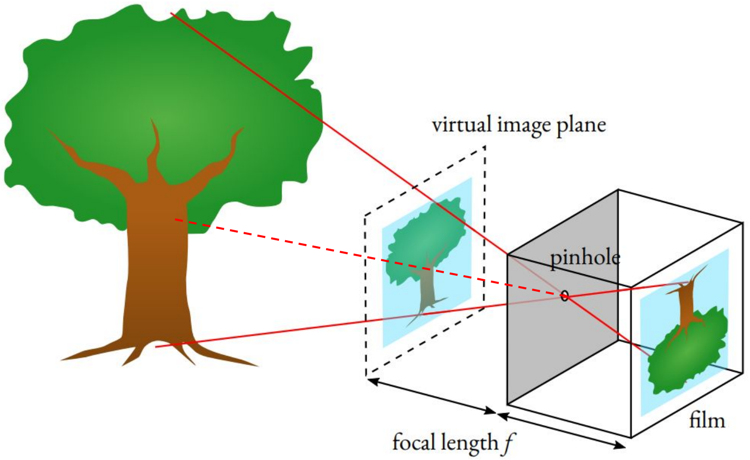
$$\mathbf{r}' = [x', y', z']$$

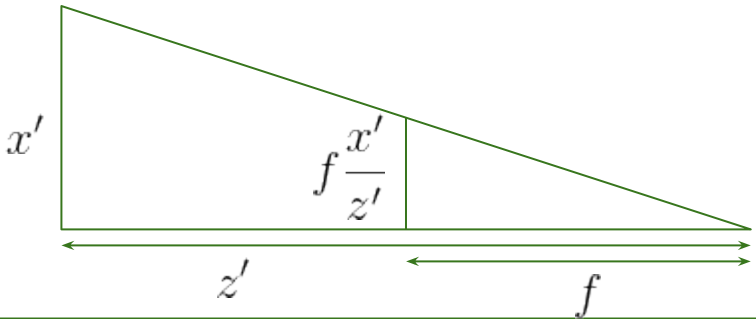
- Projection equations:

$$f\frac{x'}{z'} = u - u_0$$

$$f\frac{y'}{z'} = v - v_0$$

(u_0, v_0) center of image plane





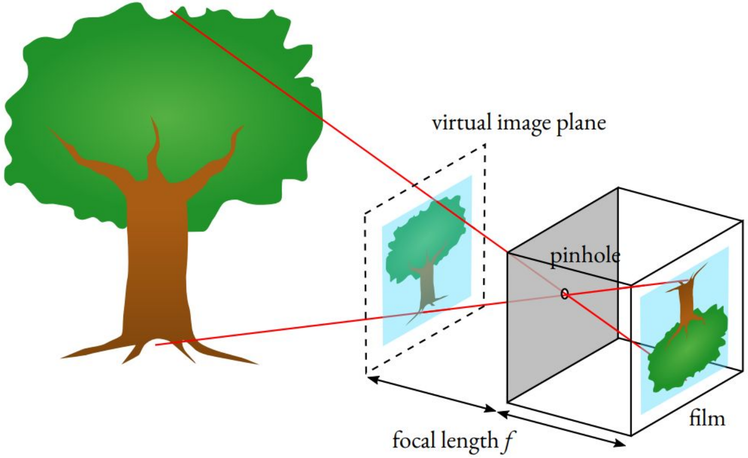
Pinhole camera model

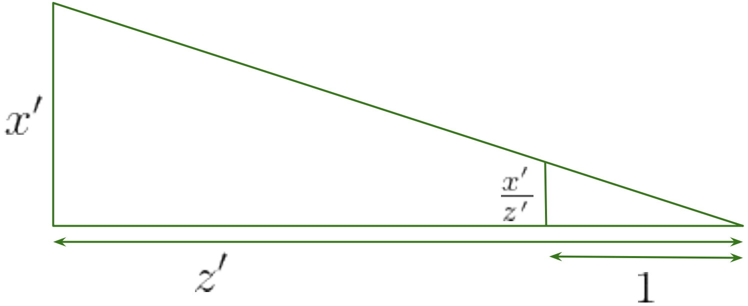
- Point in camera reference frame:

$$\mathbf{r}' = [x', y', z']$$

- Normalized image coordinates:

$$\frac{1}{z'}\mathbf{r}' = \begin{bmatrix} x'/z' \\ y'/z' \\ z'/z' \end{bmatrix} = \mathbf{y}$$





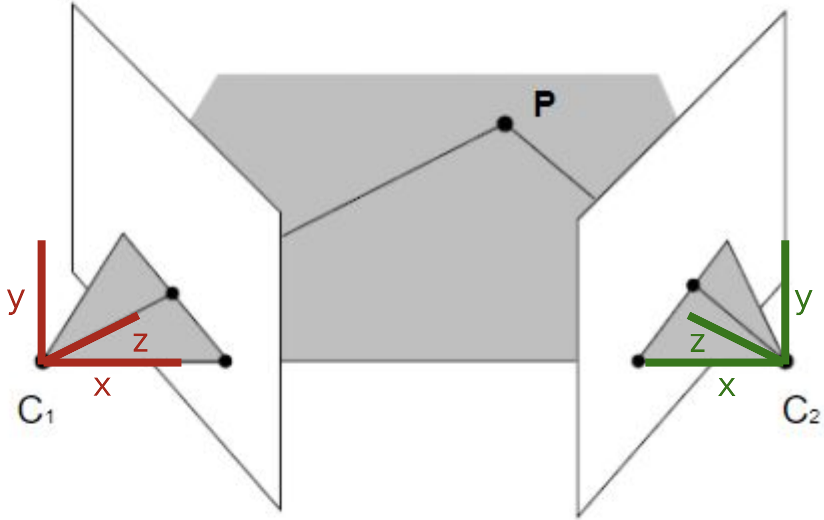
3D Contour Reconstruction

Two-view geometry

- Two camera's aiming at the same point $\mathbf{P}$

$$\mathbf{r}' = [x', y', z']$$
$$\mathbf{r}'' = [x'', y'', z'']$$
$$\mathbf{t} = [t_1, t_2, t_3]$$

$$\mathbf{r}'' = R\mathbf{r}' + \mathbf{t}$$



Essential Matrix

- $$\mathbf{r}'' = R\mathbf{r}' + \mathbf{t}$$
  $$z''\mathbf{y}'' = z'R\mathbf{y}' + \mathbf{t}$$
  $$0 = (\mathbf{y}'')^T E\mathbf{y}'$$

3D Contour Reconstruction

Essential Matrix

- $$\begin{aligned} \mathbf{r}'' &= R\mathbf{r}' + \mathbf{t} \\ z''\mathbf{y}'' &= z'R\mathbf{y}' + \mathbf{t} \\ 0 &= (\mathbf{y}'')^T E\mathbf{y}' \end{aligned}$$

- Where E is a 3x3 matrix called the Essential Matrix defined by $E = [\mathbf{t}]_\times R$

$$\mathbf{t} \times R = [\mathbf{t}]_\times R, \quad [\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

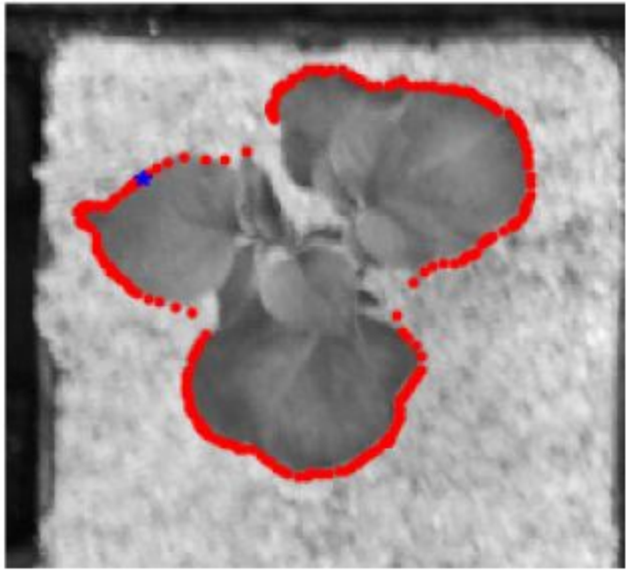
3D Contour Reconstruction

Essential Matrix

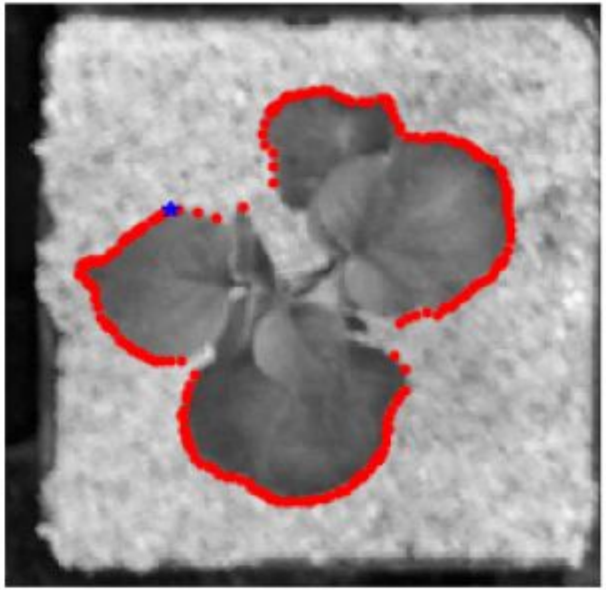
- Determine E from $(\mathbf{y}'')^T E \mathbf{y}' = 0$
- $Y\mathbf{e} = \mathbf{0}$, nx9 matrix Y
- Minimum of 8 point correspondences
- Find nullspace of Y using Singular Value Decomposition
  $U, D, V^T = \mathrm{svd}(Y)$
  in column of V

3D Contour Reconstruction

Point matching



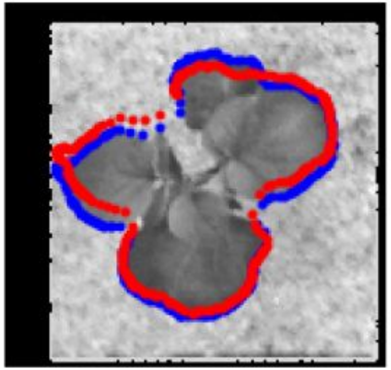
(a) Image and contour 1



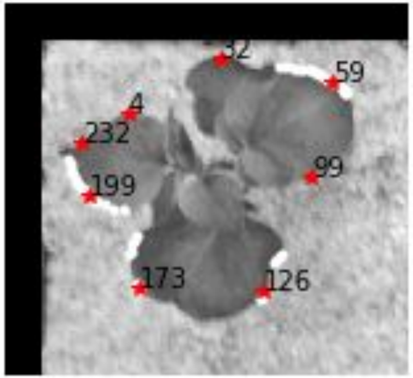
(b) Image and contour 2

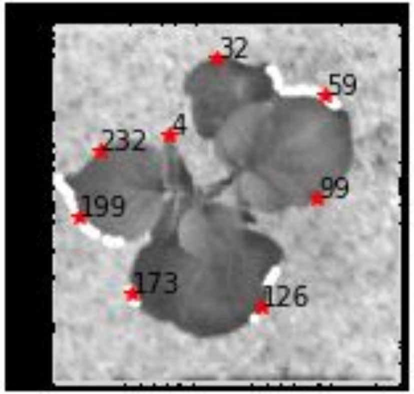
3D Contour Reconstruction

Point matching



- Rough alignment
- Choose 8 random points on contour
  and find the corresponding closest points
- Compute E and check
  $(\mathbf{y}'')^T E \mathbf{y}' < \varepsilon$
- Continue until satisfied





3D Contour reconstruction

- Obtain rotation and translation via another Singular Value Decomposition of E

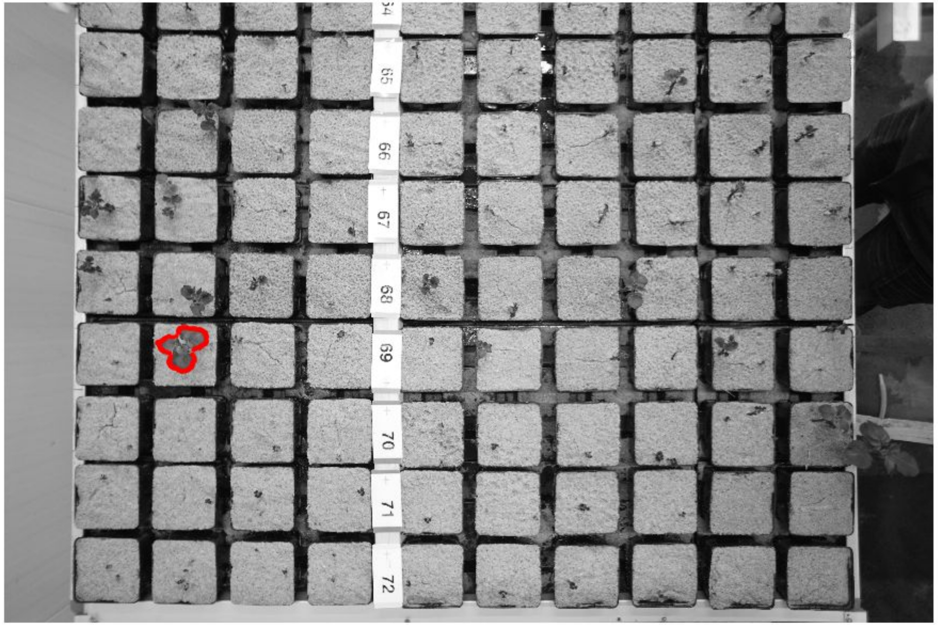
- Retrieve depth coordinate

$$\mathbf{y}'' = \frac{1}{z''}\mathbf{r}'' = \frac{R\mathbf{r}' + \mathbf{t}}{\mathbf{e}_3^T(R\mathbf{r}' + \mathbf{t})}$$

$$\mathbf{y}'' = \frac{Rz'\mathbf{y}' + \mathbf{t}}{\mathbf{e}_3^T(Rz'\mathbf{y}' + \mathbf{t})}$$

$$z' = -\frac{\mathbf{y}'^T R^T \left(\mathbf{y}''\mathbf{e}_3^T - I\right)^T \left(\mathbf{y}''\mathbf{e}_3^T - I\right)\mathbf{t}}{\mathbf{y}'^T R^T \left(\mathbf{y}''\mathbf{e}_3^T - I\right)^T \left(\mathbf{y}''\mathbf{e}_3^T - I\right) R\mathbf{y}'}$$

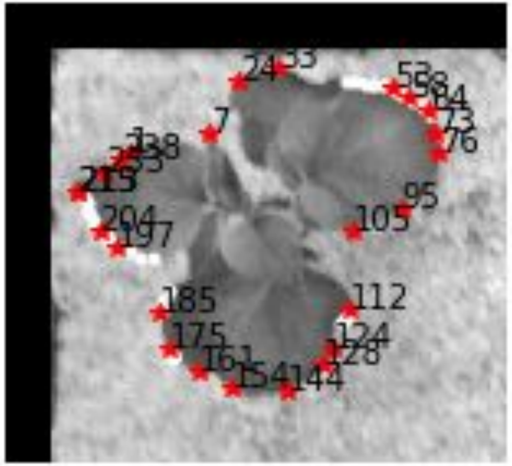
3D Contour reconstruction

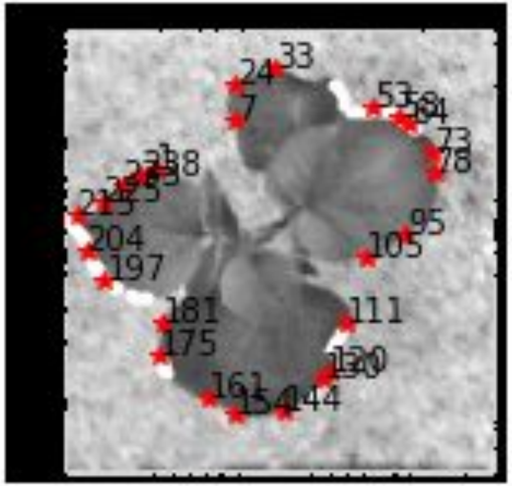


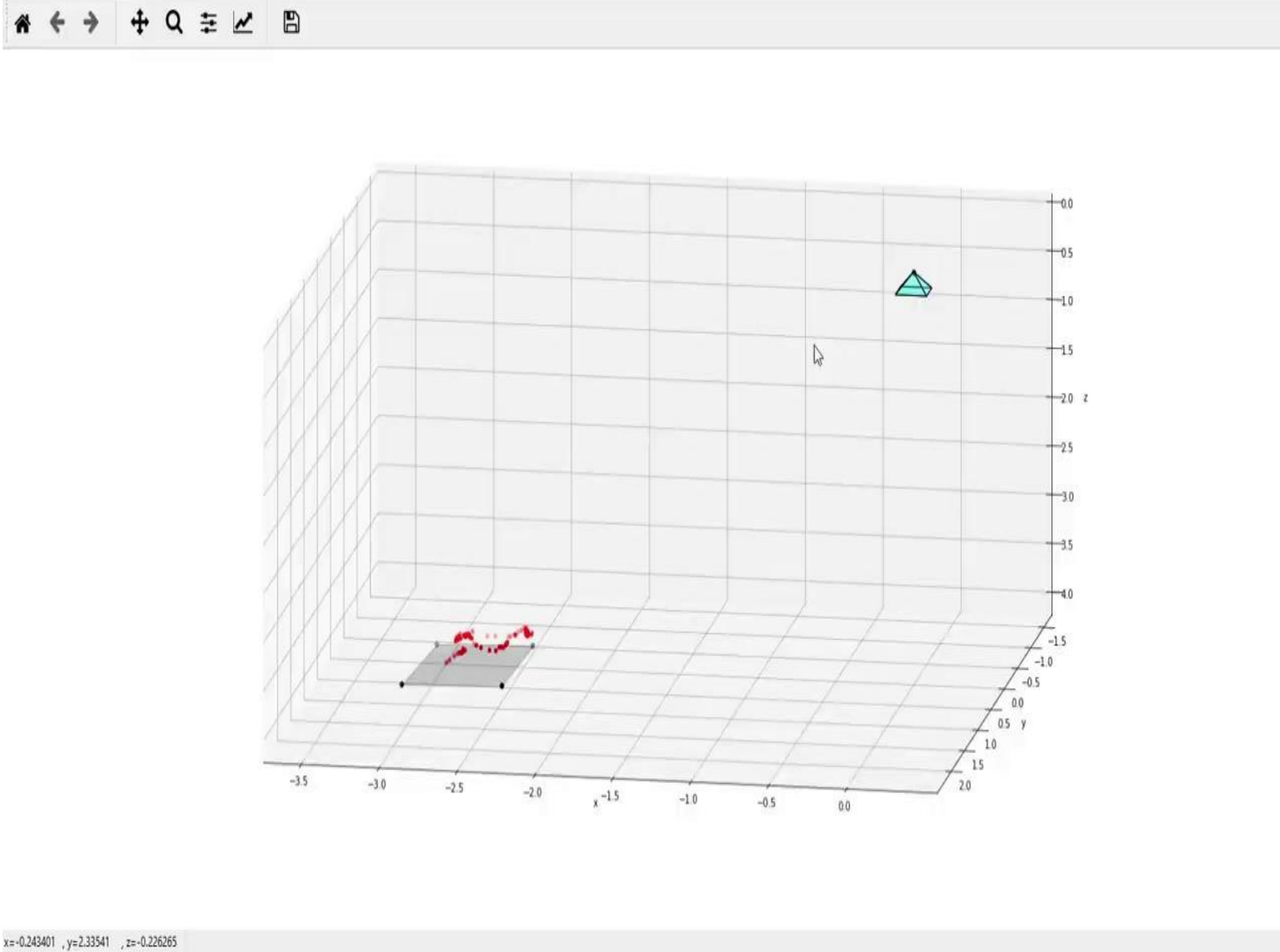
3D Contour reconstruction

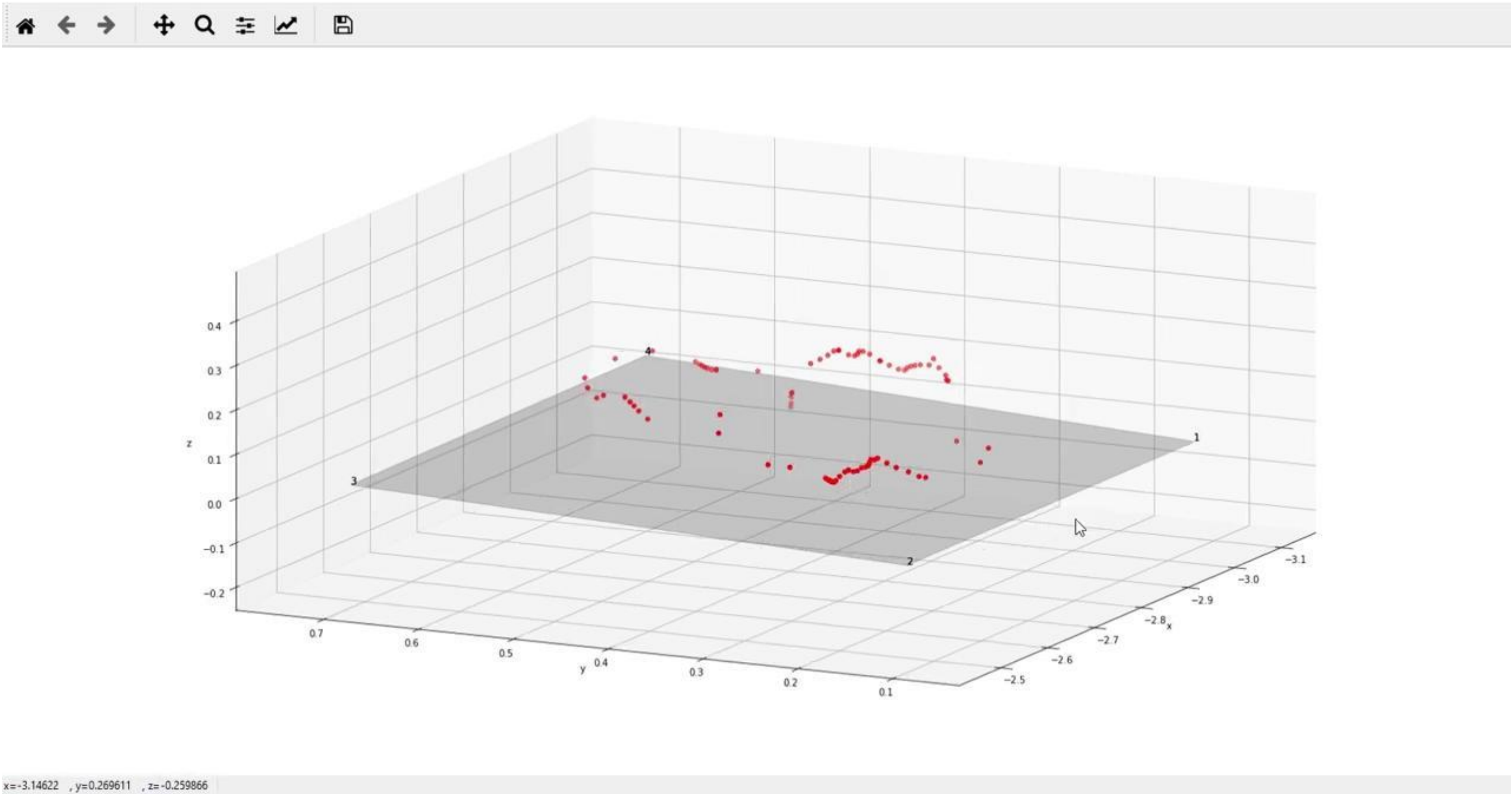
$$R = \begin{bmatrix} 1 & 0.015 & -0.010 \\ -0.014 & 1 & 0.025 \\ 0.011 & -0.025 & 1 \end{bmatrix}$$

$$\mathbf{t} = \begin{bmatrix} 0.08 \\ 0.99 \\ -0.11 \end{bmatrix}$$



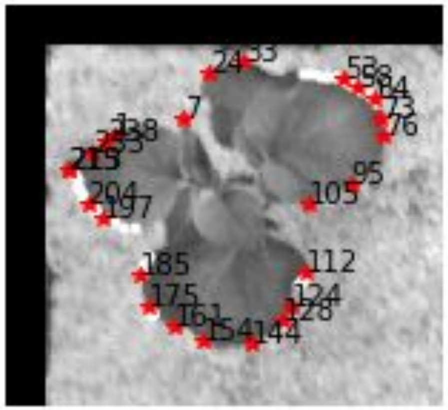


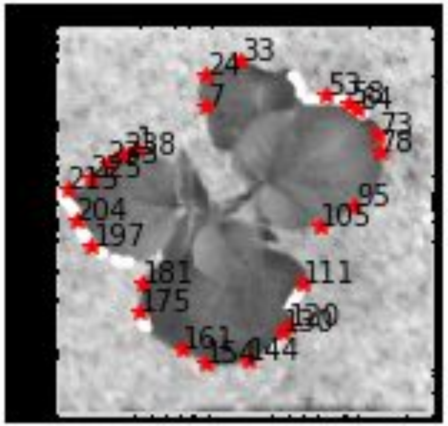

0.0
0.5
1.0
1.5
2.0 z
2.5
3.0
3.5
4.0
-1.5
-1.0
-0.5
0.0
0.5 y
1.0
1.5
2.0
-3.5
-3.0
-2.5
-2.0
x -1.5
-1.0
-0.5
0.0
x=-0.243401 , y=2.33541 , z=-0.226265

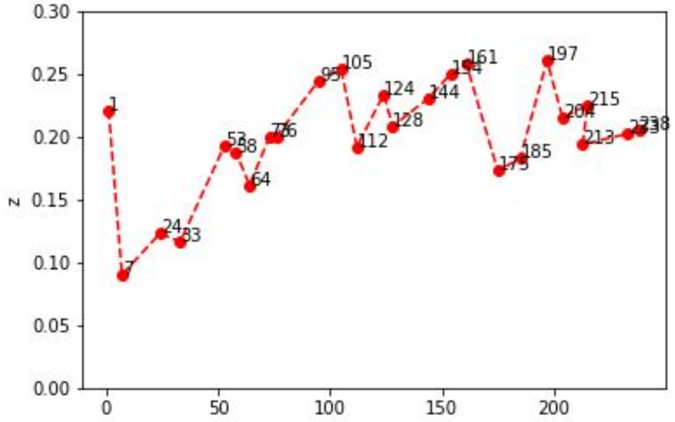


0.4
0.3
0.2
z
0.1
0.0
−0.1
−0.2
0.7
0.6
0.5
y
0.4
0.3
0.2
0.1
−2.5
−2.6
−2.7
−2.8
x
−2.9
−3.0
−3.1
1
2
3
4
x=-3.14622 , y=0.269611 , z=-0.259866

3D Contour reconstruction





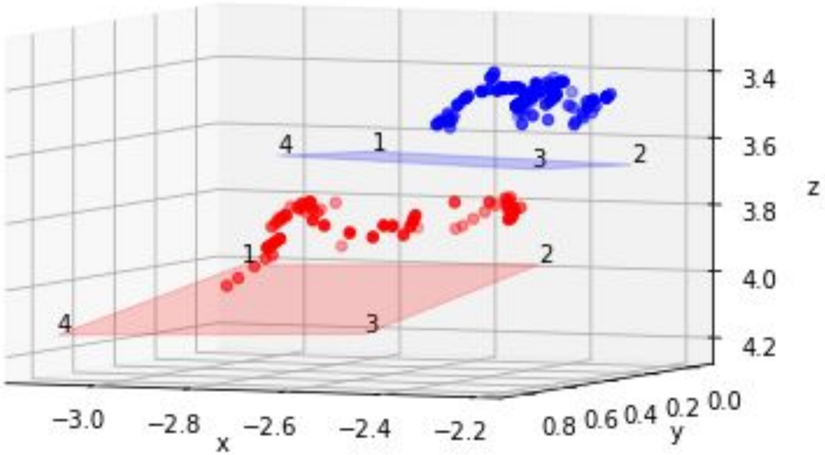


Observation of growth

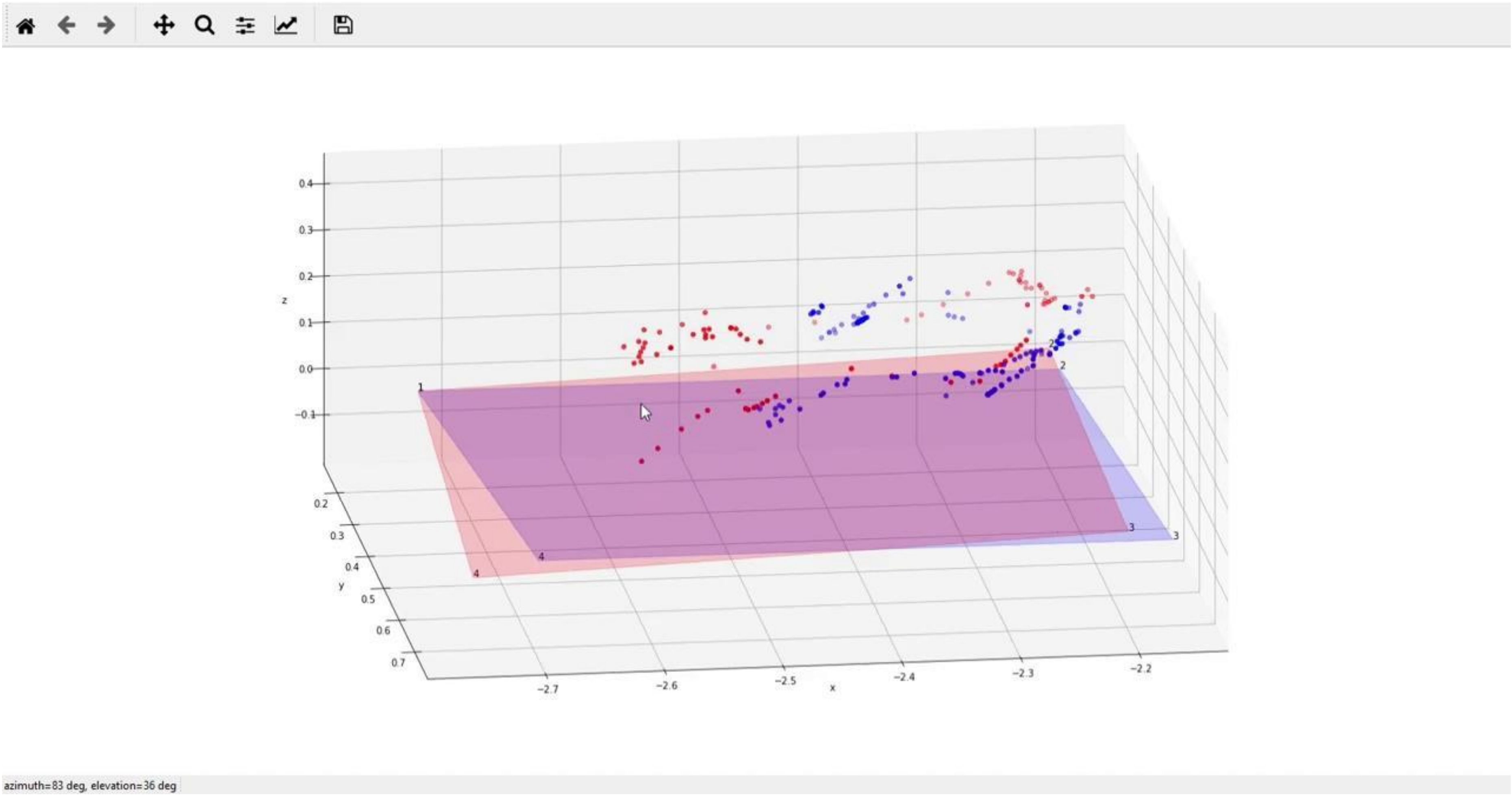
3D Contour Reconstruction

Misalignment

- $\mathbf{r}'(\tau_1) = aR\mathbf{r}'(\tau_2) + \mathbf{t}$
- Alignment based solely on corner points

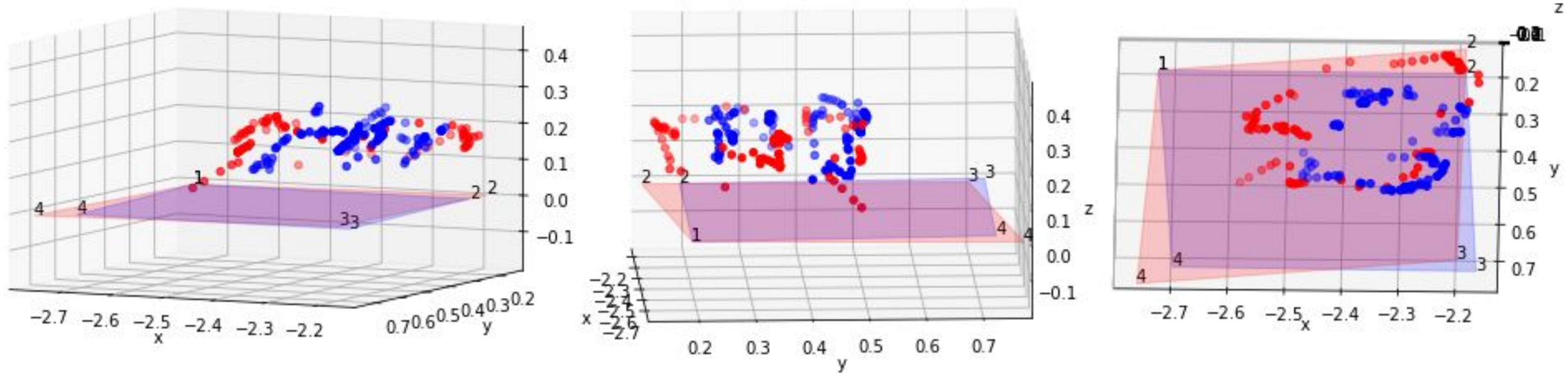


3D Contour Reconstruction



z
0.4
0.3
0.2
0.1
0.0
-0.1
y
0.2
0.3
0.4
0.5
0.6
0.7
x
-2.7
-2.6
-2.5
-2.4
-2.3
-2.2
1
2
2
3
3
4
4
azimuth=83 deg, elevation=36 deg

Observation of growth



3D Reconstruction

- 3D reconstruction up to scale
- In camera reference frame
- SVD
- Alignment of plant’s container

Double Snakes

- Combine previous problems
- Evolve both snakes simultaneously and match points on contour

- Minimize the functional

$$\Phi[\mathbf{y}_1, \mathbf{y}_2] = \int_0^1 \left[\frac{\alpha}{2} \left(\|\mathbf{y}_1'\|^2 + \|\mathbf{y}_2'\|^2 \right) + \frac{\beta}{2} \left(\|\mathbf{y}_1''\|^2 + \|\mathbf{y}_2''\|^2 \right) + \frac{1}{2} \left(\mathbf{y}_2^T E \mathbf{y}_1 \right)^2 \right] \mathrm{d}s$$

$$\frac{\partial \mathbf{y}_1}{\partial t} = \alpha \frac{\partial^2 \mathbf{y}_1}{\partial s^2} - \beta \frac{\partial^4 \mathbf{y}_1}{\partial s^4} - E^T \mathbf{y}_2 \mathbf{y}_2^T E \mathbf{y}_1 + \mathbf{F}(\mathbf{y}_1),$$

$$\frac{\partial \mathbf{y}_2}{\partial t} = \alpha \frac{\partial^2 \mathbf{y}_2}{\partial s^2} - \beta \frac{\partial^4 \mathbf{y}_2}{\partial s^4} - E \mathbf{y}_1 \mathbf{y}_1^T E^T \mathbf{y}_2 + \mathbf{F}(\mathbf{y}_2).$$

Convexity

- When does $(\mathbf{y}'')^T E \mathbf{y}' = 0$ hold?

- $(\mathbf{y}'')^T E = \mathbf{0}$
  $E\mathbf{y}' = \mathbf{0}$

  The left and right nullspace of E

Double Snakes

Double Snakes

- Simultaneously updating both contours
- Convexity of the essential matrix

Conclusion

- GVF-Snakes algorithm to detect contour
- 3D contour reconstruction up to scale
- Observation of growth due to alignment of plant's container
- Introduction of Double Snakes

Recommendations

- Stabilizing the estimation of the essential matrix
- More advanced camera model
- Improve alignment of plant's container
- Explore convexity of the coplanarity constraint

Questions?