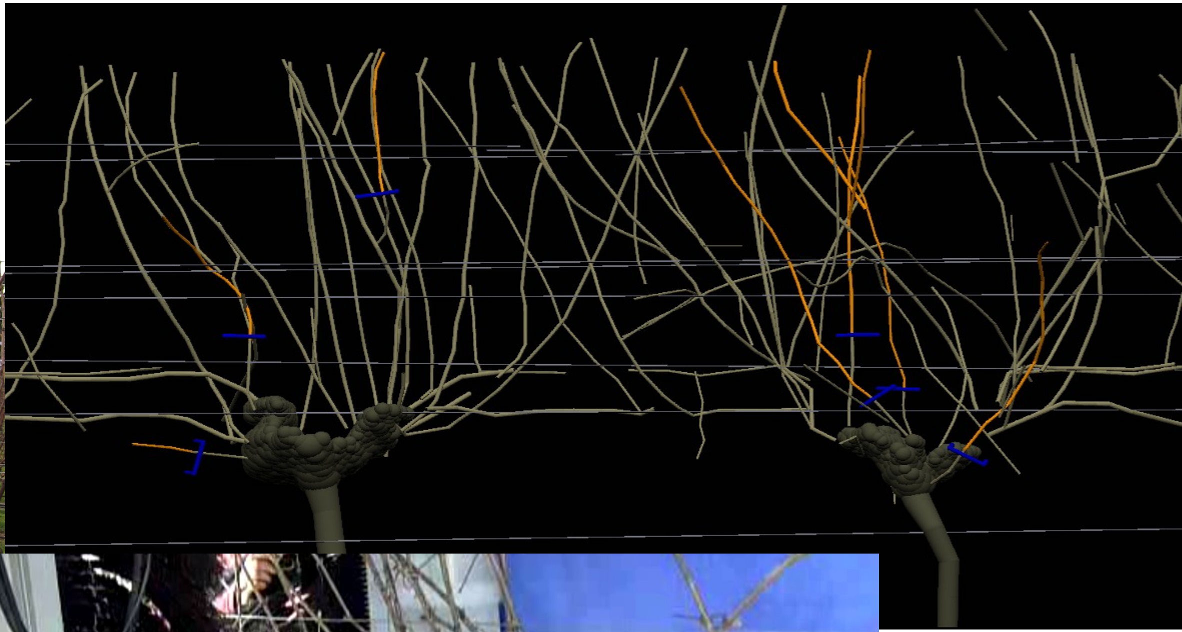
Computer vision for a grape vine pruning robot



Vision Based Automated Pruning
Computer Vision team (past + present)

PI: Richard Green

Post-doc: Tom Botterill

PhD: Davide Floriello, Ricardo David Castañeda Marin, Scott Paulin

Honour's + 3rd pro: Jarred Klopper, Joshua McCulloch, Oliver Fisher, Will Gittoes, Simon Flowers, Sam Corbett-Davies

Computer vision consultant: Steven Mills (University of Otago)

Robotics expertise: XiaoQi Chen **AI expertise:** Tanja Mitrovic

Viticulture expertise: Val Saxton (Lincoln University)

Software + hardware engineers: Samuel Williams, Ants Field, Jess Lin + Mech and COSC support staff

Pruning: Remove old wood and most new canes

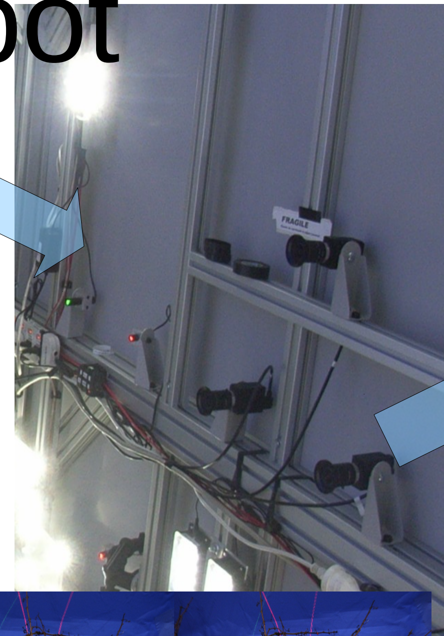
Spring – Summer – Autumn – **Winter**

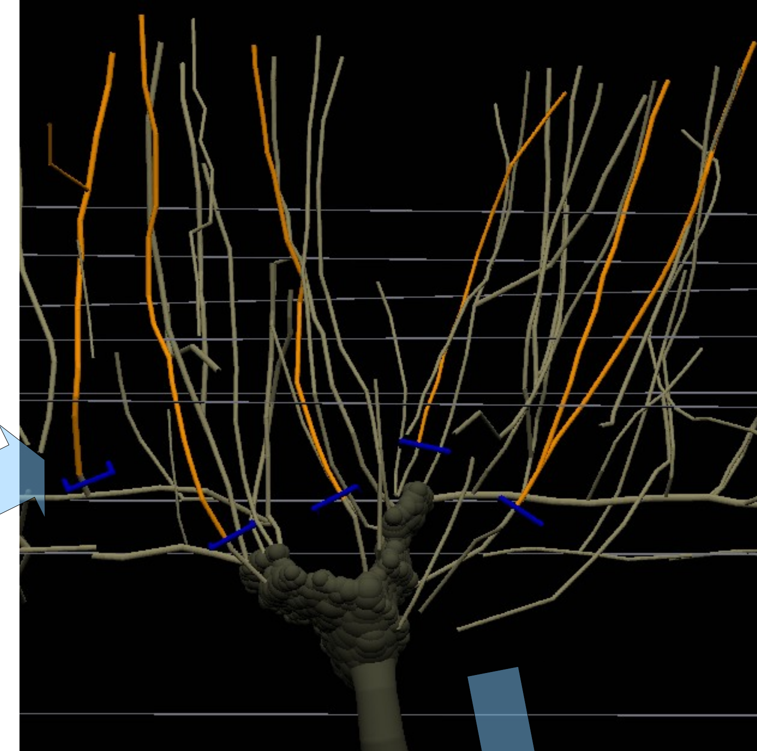
NZ

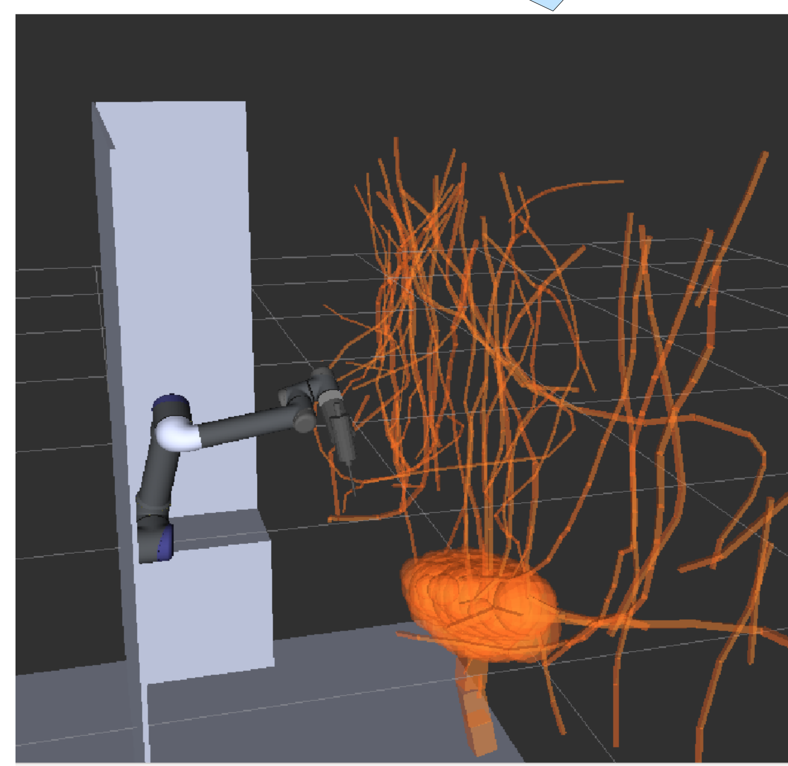
- 90 million vines, mostly Sav blanc
- Hand pruned!
  - 2 minutes per vine = 400 000 days work/year

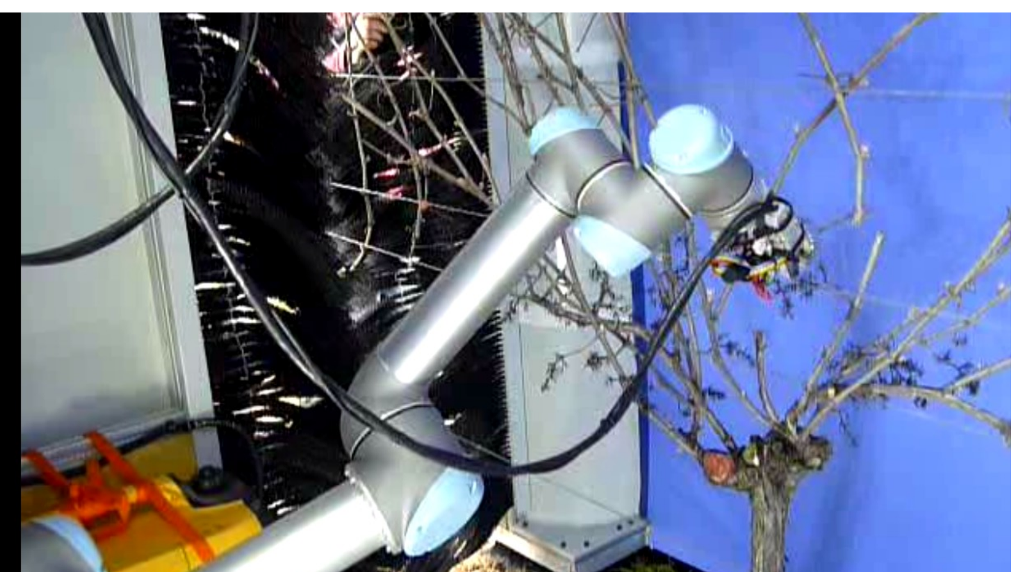
- NZ = 0.5% of world production (37 billion bottles per year!)

Vine pruning robot







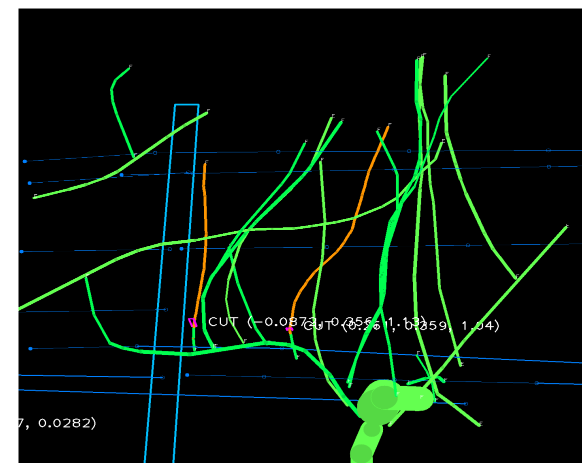


Overview

1) System overview

2) Videos of our first cuts

3) Lessons learned



System design: capturing images for 3D reconstruction

Imaging challenges:

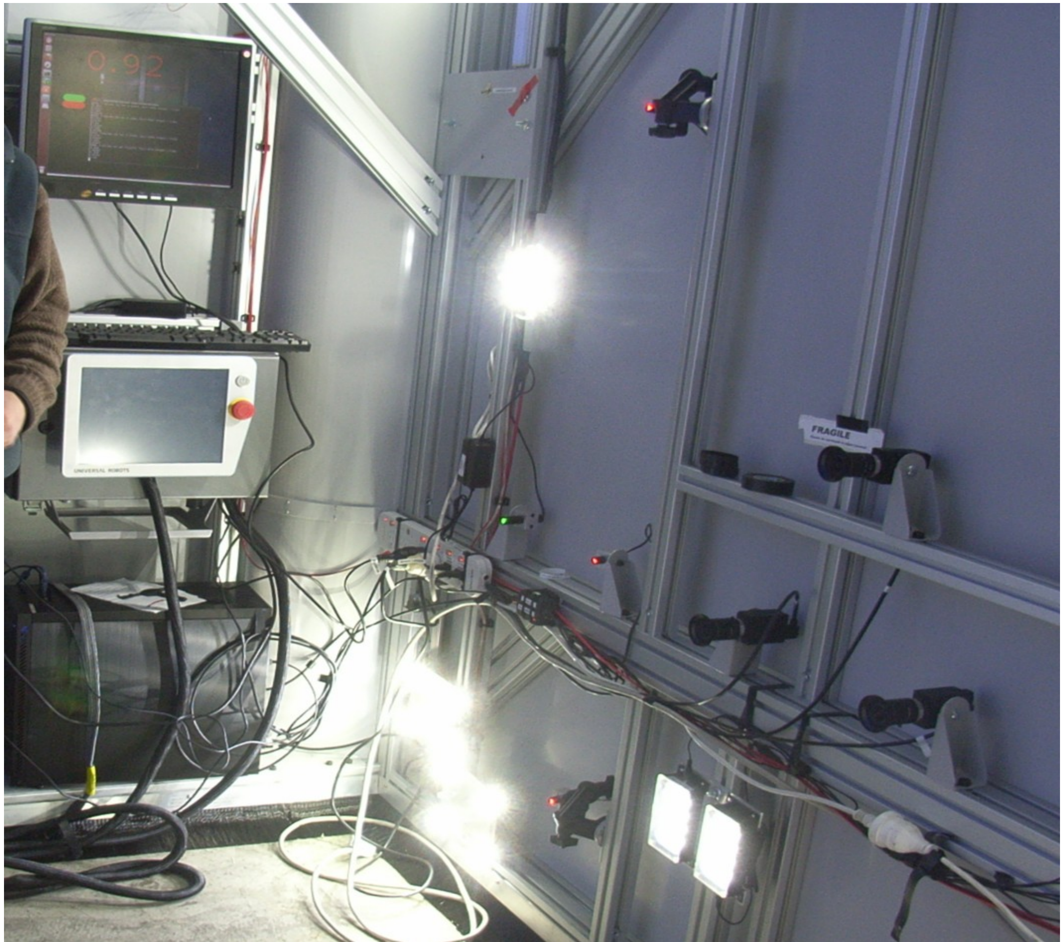
- Lighting – massive dynamic range in sun
- Separating vines from background

Canopy design

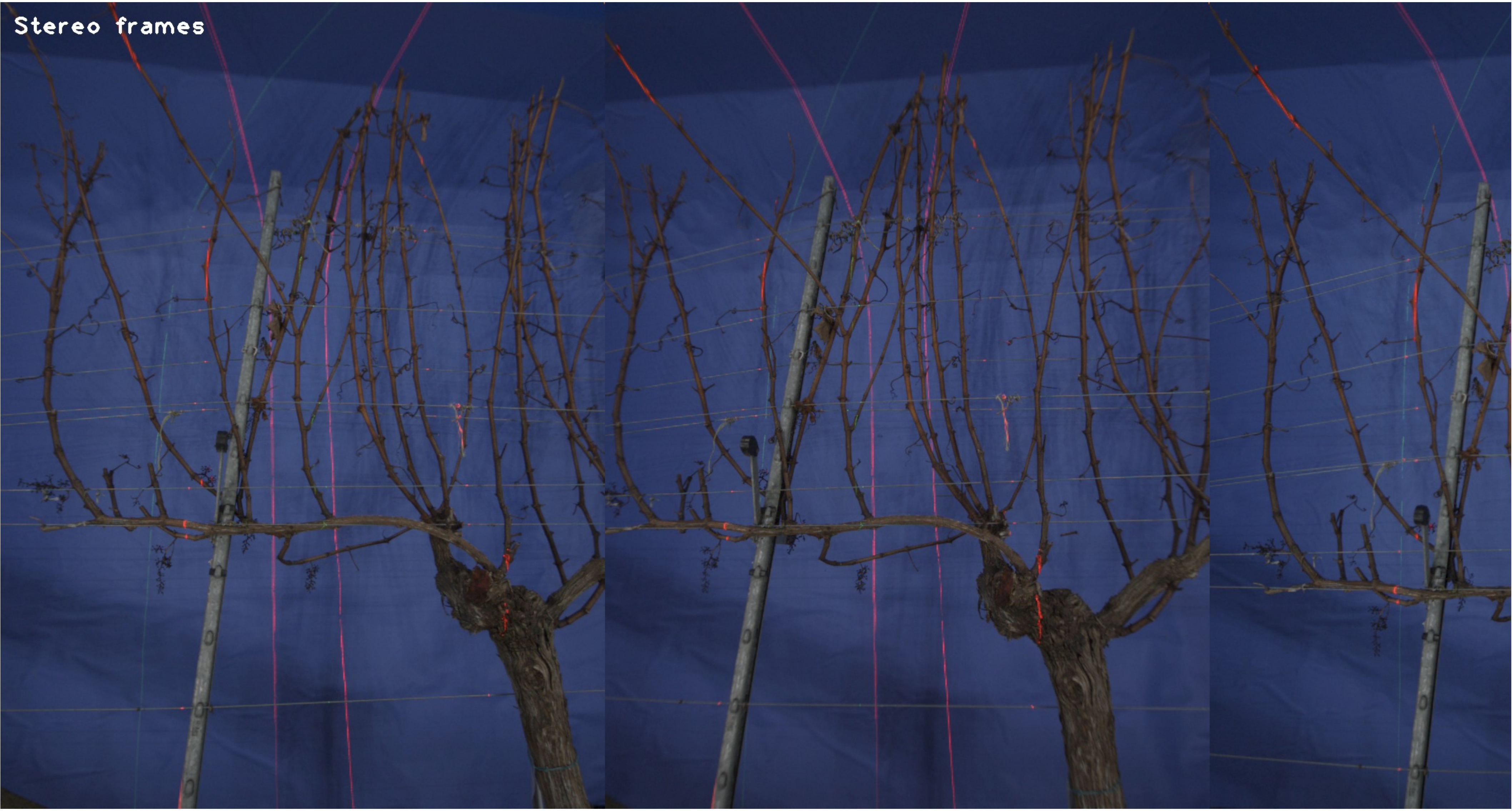
- Eliminate sunlight
- Uniform blue background

Camera rig design

- High resolution (1280 x 960)
- Large sensors (2/3”)
- Wide angle lenses (82 degree FOV rectilinear)
- 3 cameras = well conditioned reconstruction in all directions
- 30Hz = 100MB/s



Data collection at Lincoln University


Stereo frames

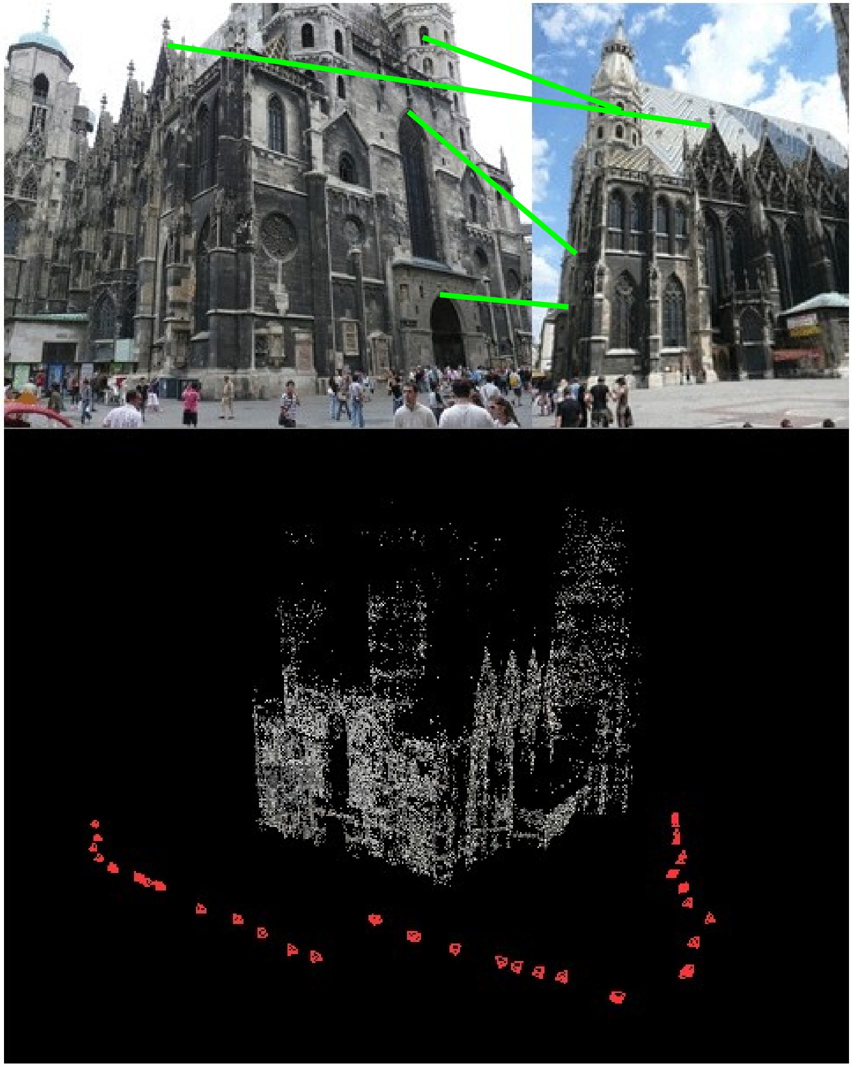
How to make a 3D model
Feature matching/bundle adjustment based 3D reconstruction?

Widely used, e.g. Bundler, Photomodeler, used to make Google Earth, etc.

(1) Match features between views

(2) Triangulate → 3D point cloud

(3) Jointly optimise structure and camera poses (Bundle Adjustment)



From Havlena et al. http://cmp.felk.cvut.cz/

Model-based 3D reconstruction

Challenges: occlusions, depth discontinuities, self similarity

Model-based 3D reconstruction

Challenges: occlusions, depth discontinuities, self similarity

Solution: Feature matching + bundle adjustment pipeline

Customise every stage to use knowledge about vines

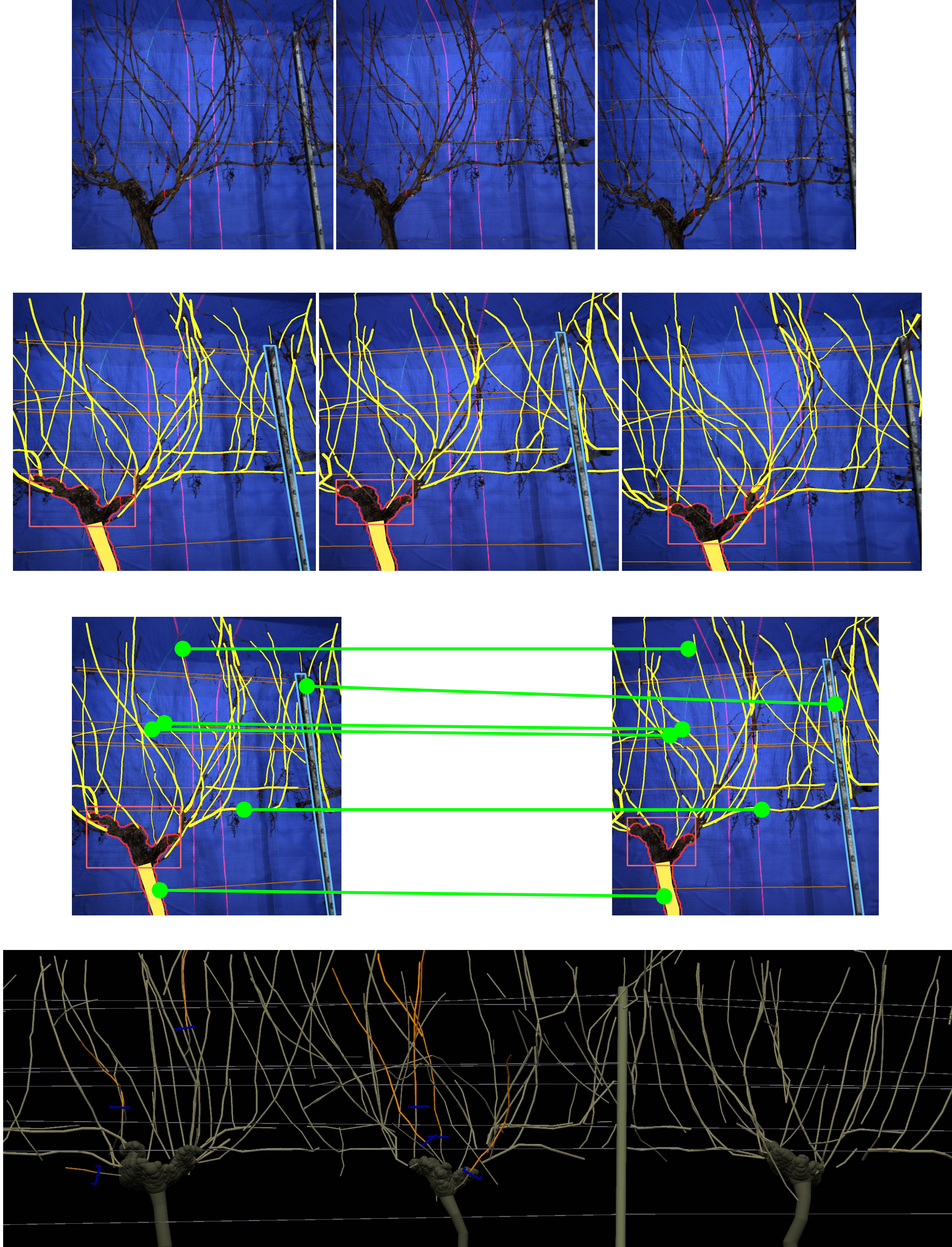
Train to perform well for vines

Stereo images

2D feature extraction
Features are canes, wires, posts

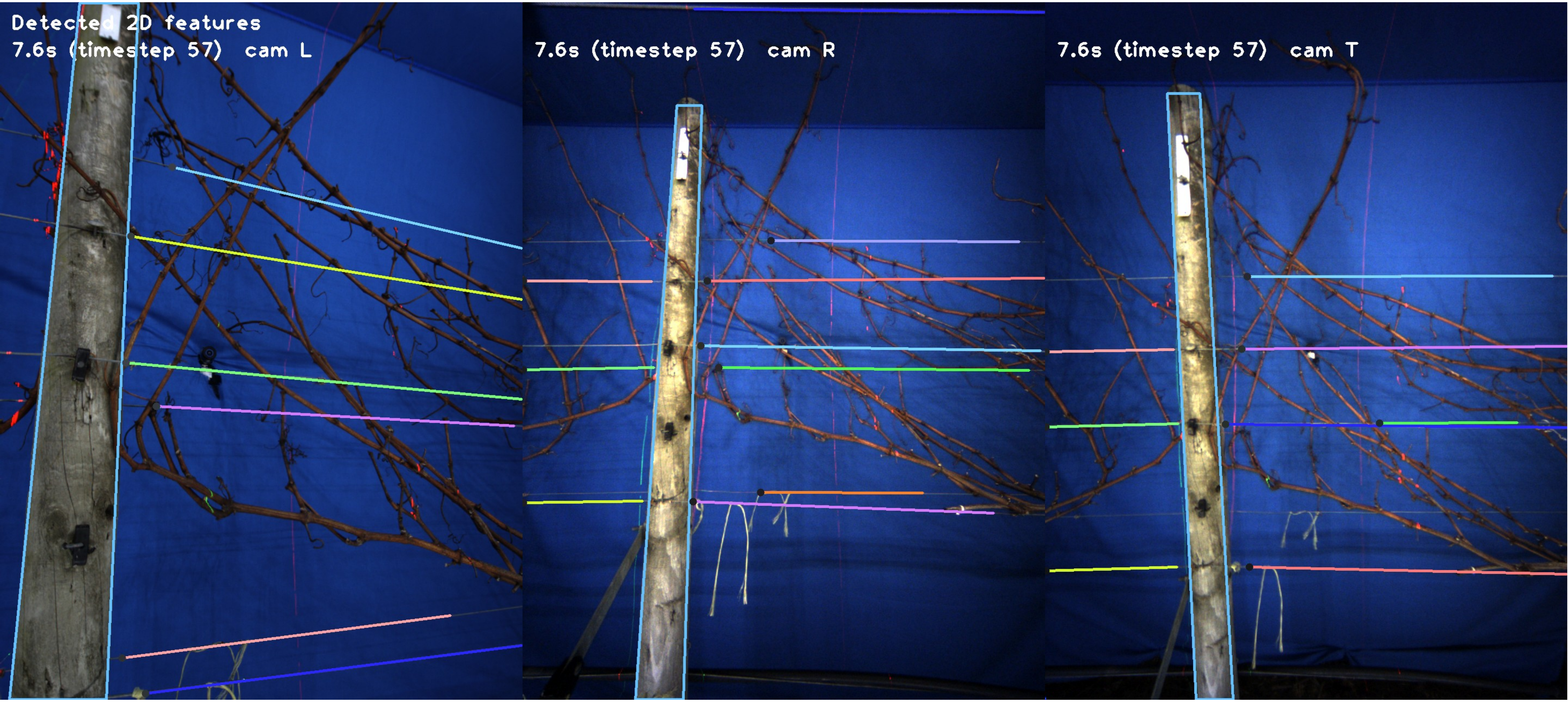
Correspond features between views
Use knowledge of vine's tree structure

Optimise parameters of 3D model to match images



2D feature extraction

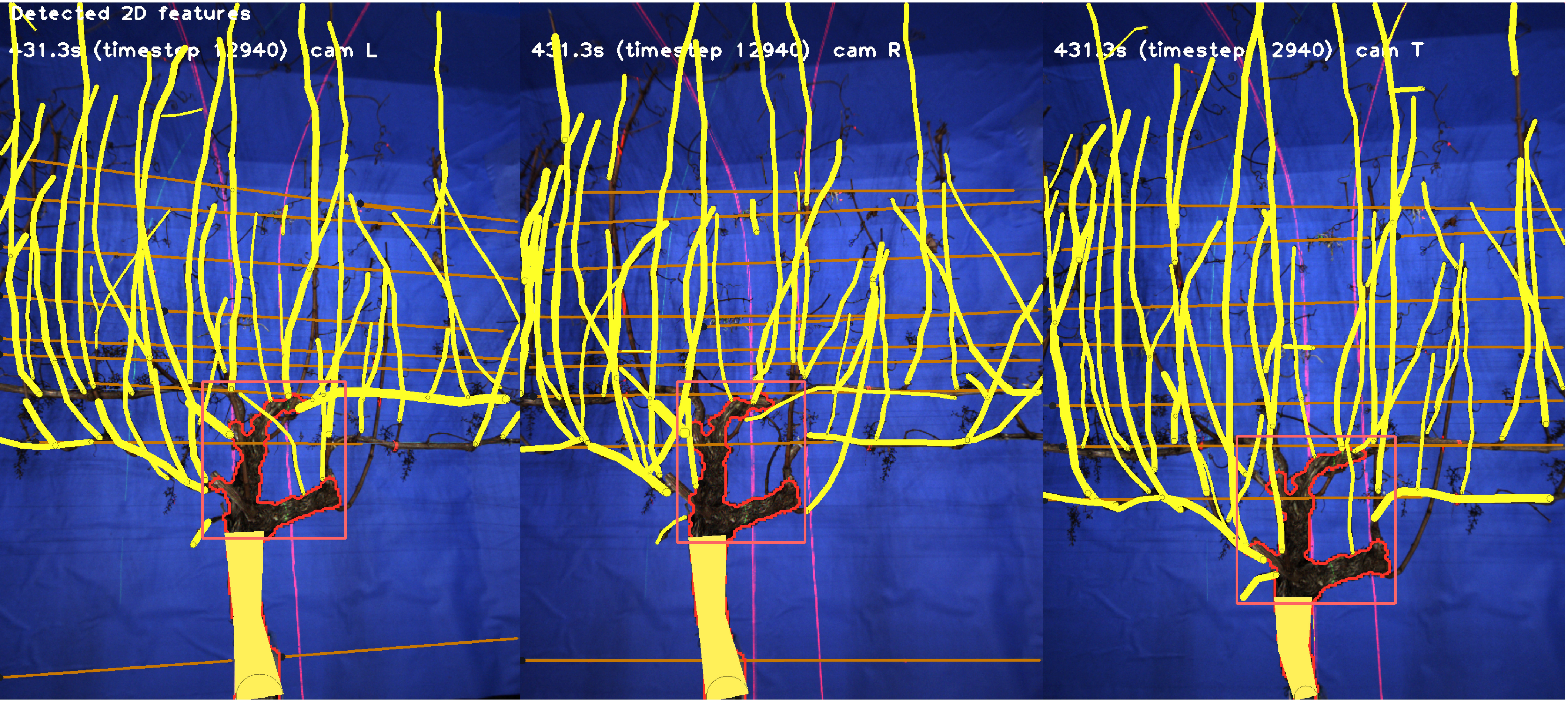
High-level features = posts, wires and vines extracted from each image



2D features → Correspondence/assignment → Bundle adjustment

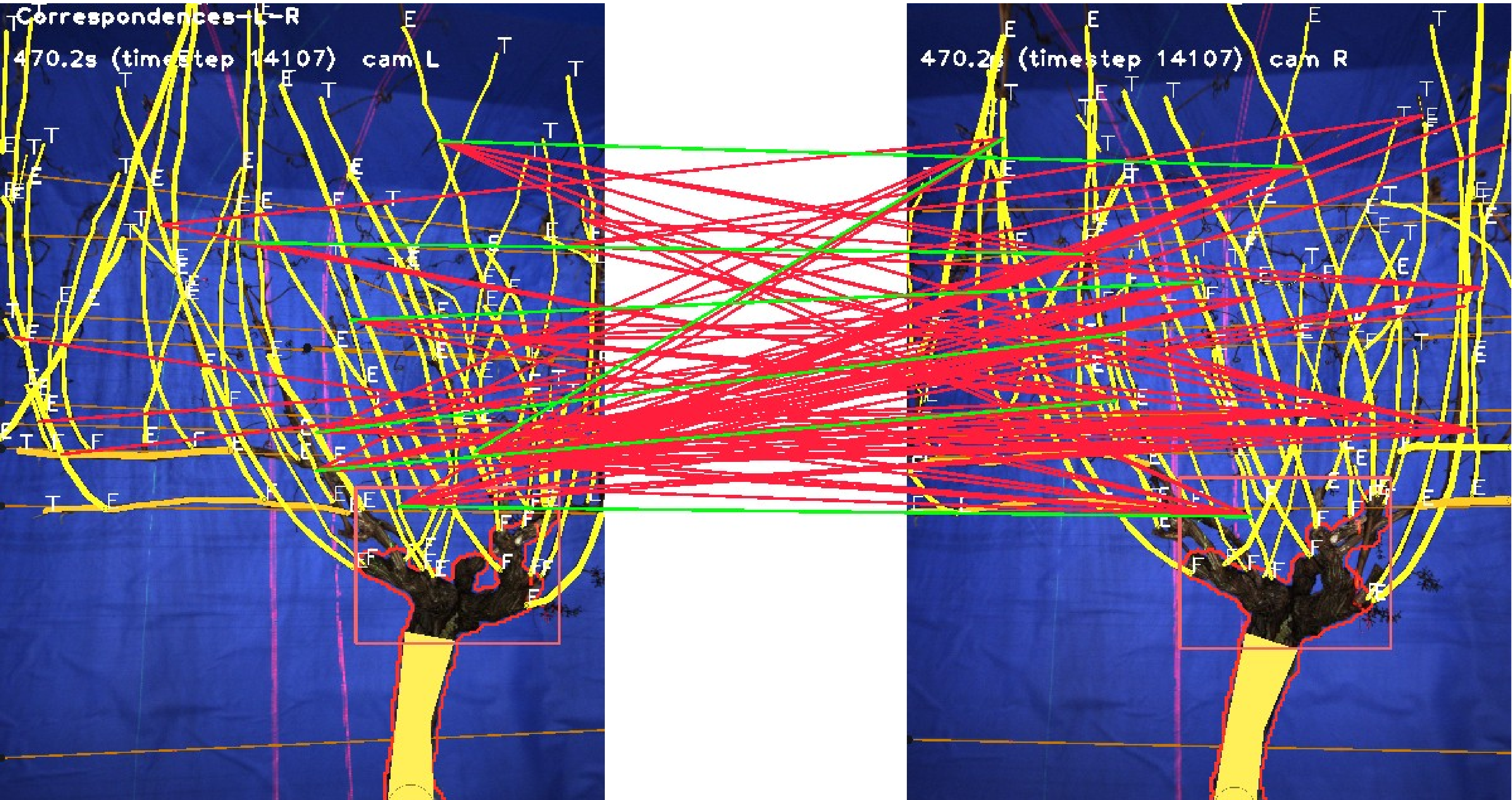
2D vines = features

- Use structure to guide cane extraction and reconstruction (smooth curve, uniform thickness, tree structure).



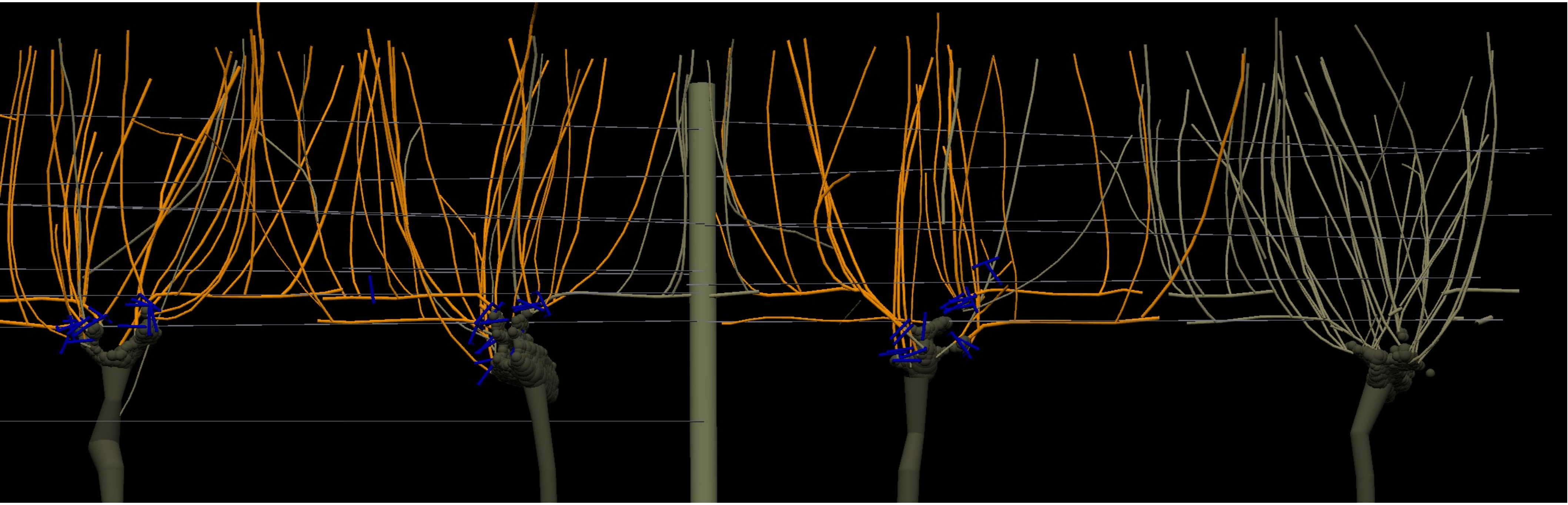
2D features → Correspondence/assignment → Bundle adjustment

Correspondence: match 2D features between images



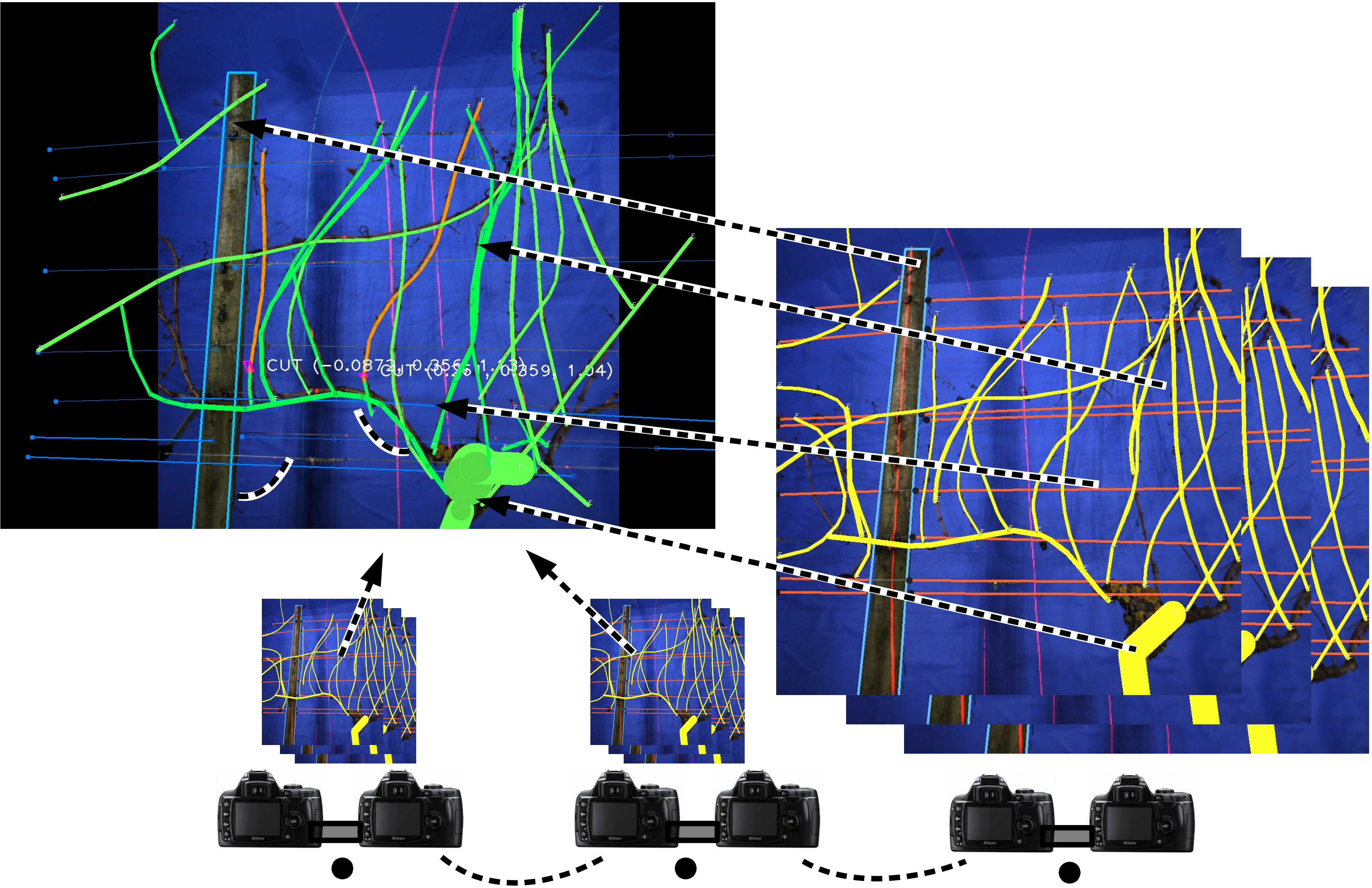
2D features → Correspondence/assignment → **Bundle adjustment**

Correspondence → 3D models



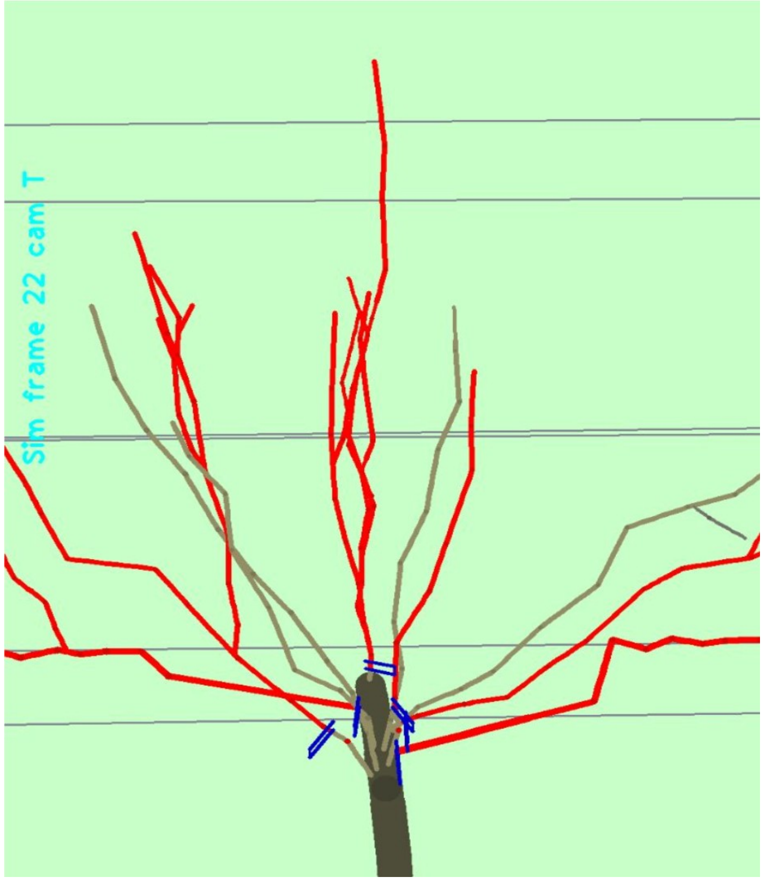
2D features → Correspondence/assignment → **Bundle adjustment**

BA – incorporate constraints from model



2D features → Correspondence/assignment → **Bundle adjustment**

AI – decide where to cut



- Work with simulated 3D vine models

AI – decide which canes to cut

- Brute-force search over all possible pruning schemes
- Need to measure the quality of a pruning scheme
- Learn cost function from hundreds of expertly-pruned vines

Results

[Video]

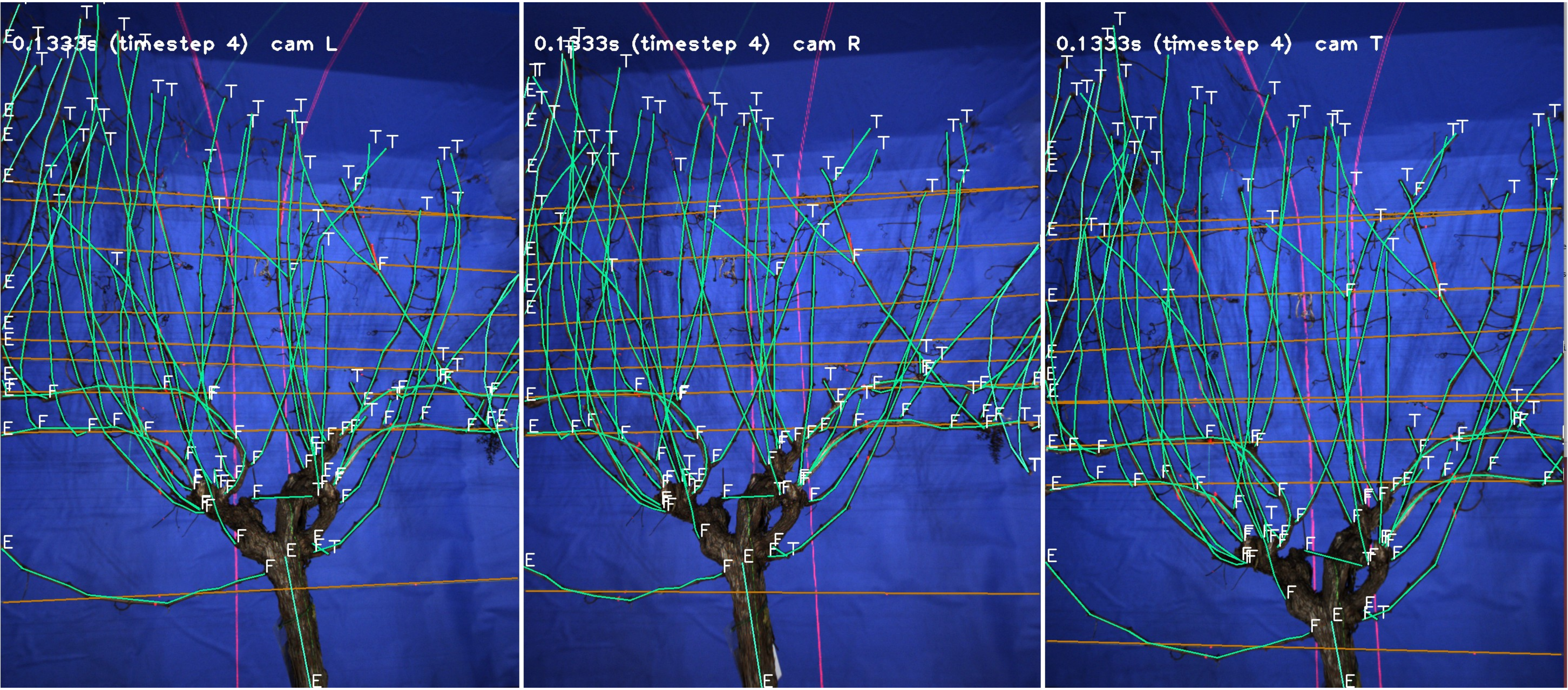
Summary: model-based feature matching + bundle adjustment pipeline

Every stage is customised to work well for vines

- Efficient – rapidly move away from pixels and point clouds to high-level features
- Modular + sequential chain of components can be developed and parametrised *in sequence* and *in isolation*
- Train on hand-labelled ground truth images of vines...

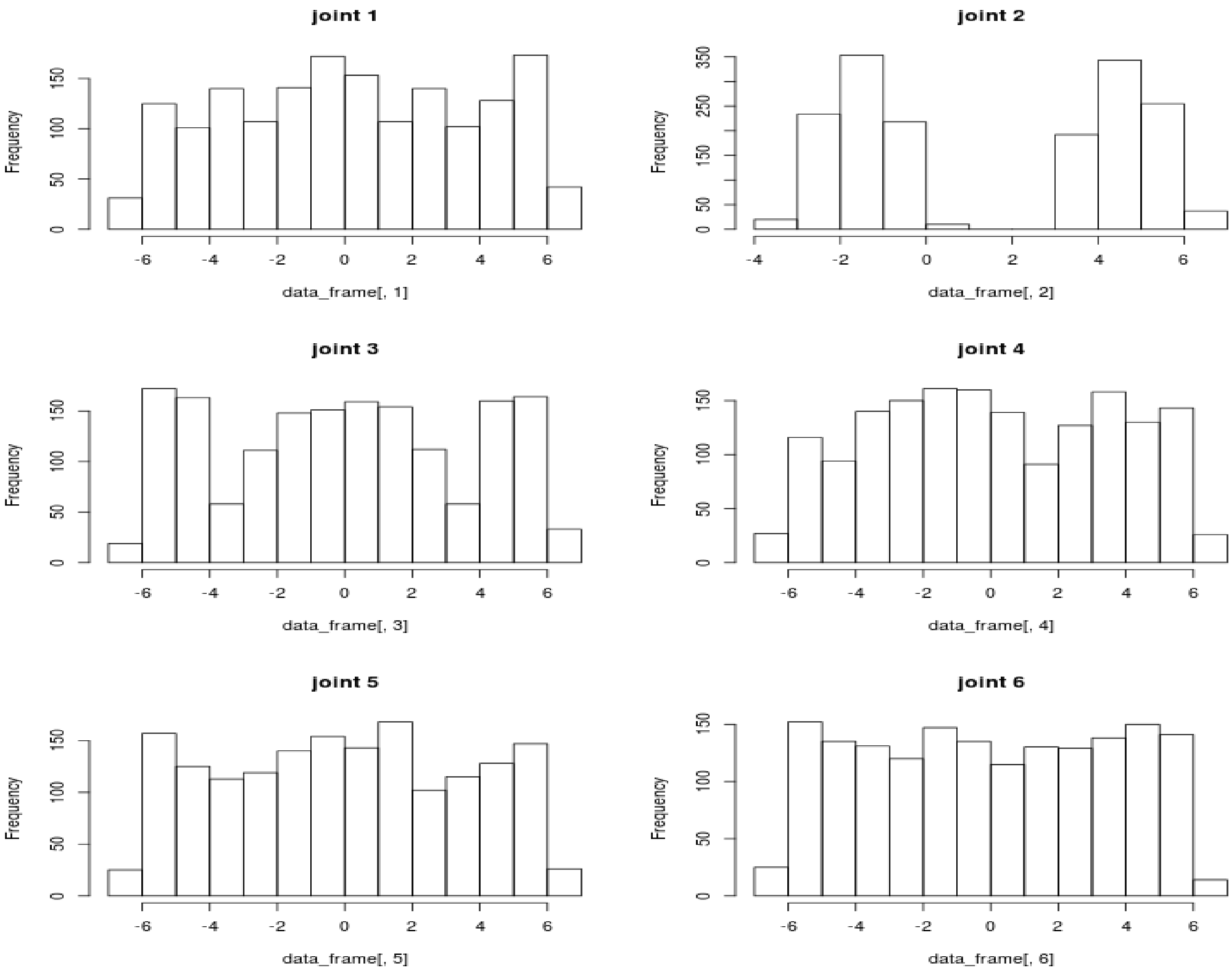
Development

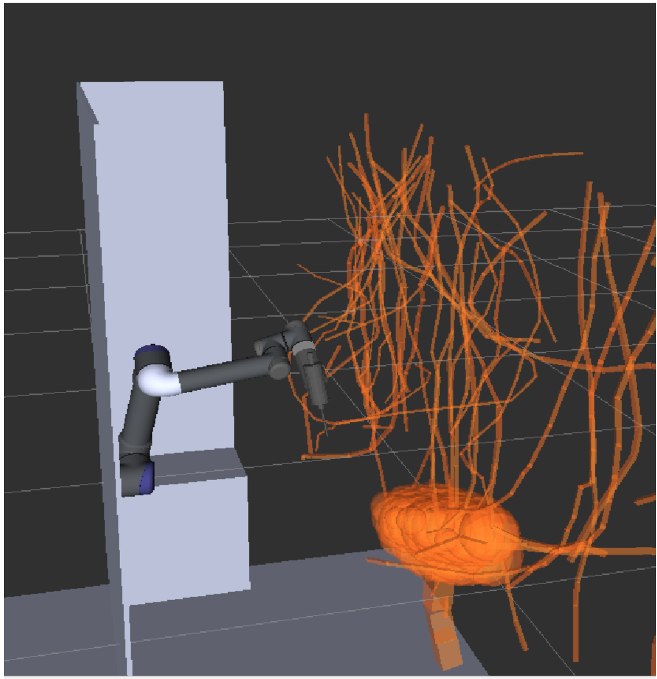
- Ground truth data



Development

- Visual output + statistics output





Main challenges: Complexity + robustness

- Not CV, not 'efficiency', not 'accuracy'!
- High complexity
  - SE challenges
- Cascade of hardware + software modules = many points of failure
  - 10 essential hardware components
  - 50k lines (50% testing code)
  - 1000 different objects
  - 22 nonlinear optimisations (LM), 10 discrete optimisations (brute force or branch+bound search), 6 machine learning problems (SVM)
- 7 calibrations
- Impossible to parameterise everything jointly

HARDWARE

Imaging:
Lighting
Cameras
Computer
Power supplies
Lasers
Screening
Background

Cutting:
Winch
Robot arm
Cutting tool
Cabling

SOFTWARE

2D:
Foreground/background segmentation
Trellis extraction
Vine structure extraction

3D:
Correspondence
Assignment
Bundle Adjustment/Optimisation
Model connection + extension

AI/Path planning:
Decide which canes to remove
Plan path for robot arm

- Technology: C++, clang, valgrind, callgrind, codelite IDE.
- Automatic testing etc. = 50% of code